

Oracle® Key Vault Administrator's Guide



Release 21.10
G12110-03
February 2025



Oracle Key Vault Administrator's Guide, Release 21.10

G12110-03

Copyright © 2014, 2025, Oracle and/or its affiliates.

Primary Author: Monika Sharma

Contributing Authors: Puneet Rai, Sunil Surabhi, Mark Doran, Patricia Huey, Prakash Jashnani

Contributors: Ajay Srivastava, Alexis Abell, Rahil Mir, Bharathi Baskaran, Swati Vijaya Bhaskar, Lalitha Chowdary, Shubham Goyal, Claudia Hüffer, Fahad Ibrar, Srivatsan Kannan, Usha Krishnamurthy, Shirley Kumamoto, Swapna Jawarikapisha, Peter Knaggs, Michael Leong, Hui Li, William Maroulis, Khushal Melana, Dongwon Park, Sunil Pulla, Vipin Samar, Radhika Siravara, Peter Wahl, Arpan Kundu, Prahalad Ragothaman, Hariprasath Mohankumar

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xxvii
Documentation Accessibility	xxvii
Diversity and Inclusion	xxvii
Related Documents	xxvii
Conventions	xxviii

Changes in This Release for Oracle Key Vault

Changes for Oracle Key Vault Release 21.10	xxix
Changes for Oracle Key Vault Release 21.9	xxxii
Changes for Oracle Key Vault Release 21.8	xxxiii
Changes for Oracle Key Vault Release 21.7	xxxv
Changes for Oracle Key Vault Release 21.6	xxxviii
Changes for Oracle Key Vault Release 21.5	xliv
Changes for Oracle Key Vault Release 21.4	xlviii
Changes for Oracle Key Vault Release 21.3	li

1 Introduction to Oracle Key Vault

About Key and Secrets Management in Oracle Key Vault	1-1
Benefits of Using Oracle Key Vault	1-2
Oracle Key Vault Use Cases	1-4
Centralized Management of TDE Master Encryption Keys Using Online Master Encryption Keys	1-4
Remote Server Access Controls and Private Key Governance for Public Key Authentication	1-6
Centralized Storage of Oracle Wallet Files and Java Keystores	1-6
Storage of Credential Files	1-7
Online Management of Endpoint Keys and Secret Data	1-8
Who Should Use Oracle Key Vault	1-8
Major Features of Oracle Key Vault	1-9
Centralized Storage and Management of Security Objects	1-10

Centrally Managed Remote Server Access Controls and Improved Private Key Governance for SSH Public Key Authentication	1-11
Management of the Key Lifecycle	1-11
Reporting and Alerts	1-12
Separation of Duties for Oracle Key Vault Users	1-12
Persistent Master Encryption Key Cache	1-12
Backup and Restore Functionality for Security Objects	1-13
Management of Oracle Key Vault Using RESTful Services Utility	1-13
Support for OASIS Key Management Interoperability Protocol (KMIP)	1-14
Database Release and Platform Support	1-14
Integration with External Audit and Monitoring Services	1-15
Integration of MySQL with Oracle Key Vault	1-15
Oracle Advanced Cluster File System Encryption	1-15
Support for Cloud-Based Oracle Database Deployments	1-15
Oracle Key Vault Hardware Security Module Integration	1-16
Continuous Availability, Fault-tolerance, and High Availability through Oracle Key Vault Clustering	1-16
Oracle Key Vault Interfaces	1-16
Oracle Key Vault Management Console	1-16
Oracle Key Vault okvutil Endpoint Utility	1-17
Oracle Key Vault RESTful Services	1-17
Oracle Key Vault Client SDK	1-17
Overview of an Oracle Key Vault Deployment	1-17

2 Oracle Key Vault Concepts

Overview of Oracle Key Vault Concepts	2-1
Oracle Key Vault Deployment Architecture	2-2
Access Control Configuration	2-3
About Access Control Configuration	2-4
Access Grants	2-4
Access Control Options	2-4
Administrative Roles and Privilege within Oracle Key Vault	2-5
Separation of Duties in Oracle Key Vault	2-5
Administrative Roles	2-6
About Administrative Roles in Oracle Key Vault	2-6
System Administrator Role Duties	2-7
Key Administrator Role Duties	2-8
Audit Manager Role Duties	2-8
Endpoint Privileges	2-8
About Endpoint Privileges in Oracle Key Vault	2-9
Create Endpoint Privilege Duties and Scope	2-9
Manage Endpoint Privilege Duties and Scope	2-10

Create Endpoint Group Privilege Duties and Scope	2-11
Manage Endpoint Group Privilege Duties and Scope	2-12
Monitor Privilege	2-14
Monitoring Privilege Duties and Scope	2-14
Naming Guidelines for Objects	2-15
Emergency System Recovery Process	2-15
Root and Support User Accounts	2-16
Endpoint Managers	2-16
Endpoint Administrators	2-16
FIPS Mode	2-16

3 Oracle Key Vault Multi-Master Cluster Concepts

Oracle Key Vault Multi-Master Cluster Overview	3-1
Benefits of Oracle Key Vault Multi-Master Clustering	3-2
Multi-Master Cluster Architecture	3-3
Oracle Key Vault Cluster Nodes	3-4
Cluster Node Limitations	3-4
Cluster Subgroups	3-5
Critical Data in Oracle Key Vault	3-5
Oracle Key Vault Read/Write Nodes	3-5
Oracle Key Vault Read-Only Nodes	3-6
Cluster Node Mode Types	3-6
Operations Permitted on Cluster Nodes in Different Modes	3-7
Building and Managing a Multi-Master Cluster	3-7
About Building and Managing a Multi-Master Cluster	3-7
Creation of the Initial Node in a Multi-Master Cluster	3-7
Expansion of a Multi-Master Cluster	3-8
About the Expansion of a Multi-Master Cluster	3-9
Management of Cluster Reconfiguration Changes Using a Controller Node	3-9
Addition of a Candidate Node to the Multi-Master Cluster	3-10
Addition of More Nodes to a Multi-Master Cluster	3-10
Migration to the Cluster from an Existing Deployment	3-12
Conversion of an Oracle Key Vault Standalone Server to a Multi-Master Cluster	3-12
Conversion from a Primary-Standby Server to a Multi-Master Cluster	3-12
Oracle Key Vault Multi-Master Cluster Deployment Scenarios	3-13
Cluster Size and Availability in Deployments	3-13
Two-Node Cluster Deployment	3-14
Mid-Size Cluster Across Two Data Centers Deployment	3-15
Multi-Master Cluster Features	3-18
Cluster Inconsistency Resolution in a Multi-Master Cluster	3-18
Name Conflict Resolution in a Multi-Master Cluster	3-18

Endpoint Node Connection Lists (Endpoint Node Scan Lists)	3-19
Cluster Management Information	3-19

4 Managing Oracle Key Vault Multi-Master Clusters

About Managing Oracle Key Vault Multi-Master Clusters	4-2
Setting Up a Cluster	4-2
About Setting Up a Cluster	4-2
Creating the First (Initial) Node of a Cluster	4-2
Adding Nodes to a Cluster	4-4
Adding a Node to Create a Read/Write Pair	4-4
Adding a Node as a Read-Only Node	4-6
Creating an Additional Read/Write Pair in a Cluster	4-8
Terminating the Pairing of a Node	4-8
Disabling a Cluster Node	4-9
Enabling a Disabled Cluster Node	4-10
Deleting a Cluster Node	4-10
Force Deleting a Cluster Node	4-11
Managing Replication Between Nodes	4-12
Restarting Cluster Services	4-12
Disabling Node Replication	4-12
Enabling Node Replication	4-12
Cluster Management Information	4-13
Cluster Monitoring Information	4-14
Naming Conflicts and Resolution	4-15
About Naming Conflicts and Resolution	4-16
Naming Conflict Resolution Information	4-16
Changing the Suggested Conflict Resolution Name	4-17
Accepting the Suggested Conflict Resolution Name	4-17
Multi-Master Cluster Deployment Recommendations	4-17
Adding an Alternate Name or IP Address	4-18

5 Deploying Oracle Key Vault on an Oracle Cloud Infrastructure VM Compute Instance

About Deploying Oracle Key Vault on an Oracle Cloud Infrastructure Compute Instance	5-1
Benefits of Using Oracle Key Vault in Oracle Cloud Infrastructure	5-2
Provisioning an Oracle Key Vault Compute Instance	5-3
About Provisioning an Oracle Key Vault Compute Instance	5-4
Launching the Oracle Key Vault Compute Instance	5-4
About Launching the Oracle Key Vault Compute Instance	5-4
Step 1: Ensure That You Have Prerequisites in Place	5-5

Step 2: Find the Oracle Key Vault Image	5-5
Step 3: Launch the Oracle Key Vault VM Compute Instance	5-5
Step 4: Perform Post-Launch and Post-Installation Tasks	5-6
General Management of an Oracle Key Vault Compute Instance	5-7
Starting, Restarting, or Stopping an Oracle Key Vault Compute Instance	5-7
System Settings in an Oracle Key Vault Compute Instance	5-8
Backup and Restore Operations for Oracle Key Vault Compute Instances	5-8
Terminating an Oracle Key Vault Compute Instance	5-9
Migrating Oracle Key Vault Deployments Between On-Premises and OCI	5-9
About Performing Migrations with Oracle Key Vault Compute Instance Data	5-9
Migrating Oracle Key Vault Deployments into OCI Using Backup and Restore	5-10
Migrating Oracle Key Vault Deployments Out of OCI Using Backup and Restore	5-10
Creating Oracle Key Vault Image in Microsoft Azure	5-11
About Provisioning Oracle Key Vault in Microsoft Azure	5-11
Create an Oracle Key Vault Base Image for Microsoft Azure	5-11
Launching an Oracle Key Vault Cluster Node (Instance) from the Base Image	5-13
Creating Oracle Key Vault Image in Amazon AWS	5-13
About Provisioning Oracle Key Vault in Amazon AWS	5-13
Creating Oracle Key Vault Image on AWS	5-13
Launching an Oracle Key Vault Cluster Node (Instance) from the Base Image	5-16
Creating Oracle Key Vault Image in Google Cloud	5-16
About Provisioning Oracle Key Vault in Google Cloud	5-16
Creating Oracle Key Vault Image for Google Cloud	5-17
Launching an Oracle Key Vault Instance from the Image	5-18
Oracle Key Vault on Oracle VM Virtual Box	5-19
Installing Oracle Key Vault on Oracle VM Virtual Box	5-19

6 Oracle Database Instances in Oracle Cloud Infrastructure

About Managing Oracle Cloud Infrastructure Database Instance Endpoints	6-1
Preparing a Database Instance on OCI to be an Oracle Key Vault Endpoint	6-2
About Preparing a Database Instance on OCI to be an Oracle Key Vault Endpoint	6-2
Configuring a Database Cloud Service Instance	6-2
Creating a Low Privileged Operating System User on Database as a Service	6-3
Using an SSH Tunnel Between Oracle Key Vault and Database as a Service	6-4
Creating an SSH Tunnel Between Oracle Key Vault and a DBaaS Instance	6-5
Managing a Reverse SSH Tunnel in a Multi-Master Cluster	6-8
Managing a Reverse SSH Tunnel in a Primary-Standby Configuration	6-9
Viewing SSH Tunnel Configuration Details	6-9
Disabling an SSH Tunnel Connection	6-10
How the Connection Works if the SSH Tunnel Is Not Active	6-11
Deleting an SSH Tunnel Configuration	6-11

Registering and Enrolling a Database as a Service Instance as an Oracle Key Vault Endpoint	6-12
About Registering and Enrolling a Database as a Service Instance as an Oracle Key Vault Endpoint	6-13
Step 1: Register the Endpoint in the Oracle Key Vault Management Console	6-13
Step 2: Prepare the Endpoint Environment	6-15
Step 3: Install the Oracle Key Vault Software onto the Endpoint for Registration and Enrollment	6-15
Step 4: Perform Post-Installation Tasks	6-17
Suspending Database Cloud Service Access to Oracle Key Vault	6-19
About Suspending Database Cloud Service Access to Oracle Key Vault	6-19
Suspending Access for a Database Cloud Service to Oracle Key Vault	6-20
Resuming Database Cloud Service Access to Oracle Key Vault	6-20
Resuming a Database Endpoint Configured with a Password-Based Keystore	6-21

7 Configuring Single Sign-On in Oracle Key Vault

About Single Sign-On Authentication in Oracle Key Vault	7-1
Configuring SAML Single Sign-On (SSO) Authentication	7-2
About Configuring SAML Single Sign-On Authentication	7-2
Configuring Identity Provider for Single Sign-On for Oracle Key Vault	7-2
SAML SSO Configuration	7-3
SAML Signing Certificate	7-3
User Provisioning and Authorization	7-3
SAML Request Signing	7-3
Configuring Oracle Key Vault for Single Sign-On(SSO)	7-3
Oracle Key Vault SAML SSO Configuration	7-3
Add Single Sign-On Configuration	7-4
Creating Single Sign-On User	7-5
Authenticating Single Sign-On (SSO) User	7-6
Logging in to Oracle Key Vault as an SSO User	7-6
Managing Single Sign-On in Oracle Key Vault	7-7
Download Oracle Key Vault Single Sign-On (SSO) Certificate	7-8
Adding Single Sign-On (SSO) Configuration in Oracle Key Vault	7-8
Enabling Single Sign-On (SSO) Configuration	7-9
Disabling Single Sign-On (SSO) Configuration	7-10
Deleting Single Sign-On Configuration	7-10
Configuring Single Sign-On for Oracle Key Vault and Azure Active Directory	7-11
Adding User for Oracle Key Vault in Azure Active Directory (AD)	7-12
Configuring Single Sign-On for Oracle Key Vault and ADFS	7-13
Guidelines for Managing Single Sign-On Configuration	7-16

8 Managing LDAP User Authentication and Authorization in Oracle Key Vault

About Managing LDAP User Authentication and Authorization in Oracle Key Vault	8-1
Considerations for Granting Privileges to LDAP Users	8-3
Configuring the LDAP Directory Server Connection to Oracle Key Vault	8-4
Step 1: Prepare the LDAP Directory Server	8-4
Step 2: Create the LDAP Connection in Oracle Key Vault	8-5
Step 3: Mapping LDAP Groups in Oracle Key Vault	8-6
Logins to Oracle Key Vault as an LDAP User	8-7
About Logins to Oracle Key Vault as an LDAP User	8-8
Logging in to Oracle Key Vault as an LDAP User	8-8
Managing the LDAP Configuration	8-9
Enabling an LDAP Configuration	8-9
Modifying an LDAP Configuration	8-10
Testing an LDAP Configuration	8-11
Disabling an LDAP Configuration	8-11
Deleting an LDAP Configuration	8-11
Managing LDAP Groups	8-12
About Managing LDAP Groups	8-12
Creating an LDAP Group Mapping	8-13
Modifying an LDAP Group Mapping	8-14
Validating LDAP Group Mappings	8-16
Deleting LDAP Group Mappings	8-17
Managing Oracle Key Vault-Generated LDAP Users	8-17
About Managing LDAP Users	8-17
Finding Information About an Oracle Key Vault-Generated LDAP User	8-18
Validation of Oracle Key Vault-Generated LDAP Users	8-18
About the Validation of Oracle Key Vault-Generated LDAP Users	8-18
Validating Oracle Key Vault-Generated LDAP Users	8-19
Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges	8-19
About Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges	8-19
Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges (Key Administrators)	8-20
Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges (Regular Users)	8-20
Deleting Oracle Key Vault-Generated LDAP Users	8-21

9 Managing Oracle Key Vault Users

Managing User Accounts	9-1
About Oracle Key Vault User Accounts	9-2

User Account Profile Parameters	9-3
About User Account Profile Parameters	9-3
Managing User Account Profile Parameters	9-3
How a Multi-Master Cluster Affects User Accounts	9-4
Multi-Master Cluster Effect on User Account Profile Parameters	9-5
Multi-Master Cluster Effect on System Administrator Users	9-5
Multi-Master Cluster Effect on Key Administrator Users	9-6
Multi-Master Cluster Effect on Audit Manager Users	9-6
Multi-Master Cluster Effect on Administration Users	9-6
Multi-Master Cluster Effect on System Users	9-7
Creating an Oracle Key Vault User Account	9-7
Viewing User Account Details	9-9
Deleting an Oracle Key Vault User Account	9-10
Managing Administrative Roles and User Privileges	9-10
About Managing Administrative Roles and User Privileges	9-11
Granting or Changing an Administrative Role of a User	9-11
Granting the Create Endpoint Privilege	9-12
Granting the Manage Endpoint Privilege	9-13
Granting the Create Endpoint Group Privilege	9-13
Granting the Manage Endpoint Group Privilege	9-14
Granting the Monitor Privilege	9-14
Revoking an Administrative Role or Privilege from a User	9-15
Granting a User Access to a Virtual Wallet	9-16
Enforce Separation of Administrator Roles	9-16
Managing User Passwords	9-17
About Changing User Passwords	9-17
Changing Your Own Password	9-18
Changing Another User's Password	9-18
Changing a Password Manually	9-19
Changing a Password Through Email Notification	9-19
Controlling the Use of Password Reset Methods	9-20
About Controlling the Use of Password Reset Methods	9-20
Configuring the Use of Password Reset Operations	9-21
Unlocking a User Account	9-21
Managing User Email	9-21
Changing the User Email Address	9-22
Changing Your Own Email	9-22
Changing Another User's Email	9-22
Disabling Email Notifications for a User	9-23
Managing User Groups	9-23
About Managing User Groups	9-24
How a Multi-Master Cluster Affects User Groups	9-24

Creating a User Group	9-24
Adding a User to a User Group	9-26
Granting a User Group Access to a Virtual Wallet	9-26
Renaming a User Group	9-27
Changing a User Group Description	9-27
Removing a User from a User Group	9-28
Deleting a User Group	9-28
Managing support and root Password	9-28
Requirements for root and support User Passwords	9-29
Changing the root User Password	9-29
Changing the support User Account Password	9-30

10 Managing Oracle Key Vault Virtual Wallets and Security Objects

Managing Virtual Wallets	10-1
About Virtual Wallets	10-2
Creating a Virtual Wallet	10-2
Modifying a Virtual Wallet	10-4
Adding Security Objects to a Virtual Wallet	10-6
Removing Security Objects from a Virtual Wallet	10-6
Deleting a Virtual Wallet	10-7
Managing Access to Virtual Wallets from Keys & Wallets Tab	10-7
About Managing Access to Virtual Wallets from the Keys & Wallets Tab	10-7
Granting Access to Users, User Groups, Endpoints, and Endpoint Groups	10-8
Modifying Access to Users, User Groups, Endpoints, and Endpoint Groups	10-9
Managing Access to Virtual Wallets from User's Menu	10-9
Granting a User Access to a Virtual Wallet	10-10
Revoking User Access from a Virtual Wallet	10-10
Granting a User Group Access to a Virtual Wallet	10-11
Revoking User Group Access from a Virtual Wallet	10-11
Managing Security Objects	10-11
Creating Keys	10-12
About Creating Keys	10-12
Application Keys	10-13
Creating Symmetric Keys	10-13
Create Public-Private Key Pair	10-16
Create TDE Master Encryption Key	10-19
Create GoldenGate Master Key	10-21
Creating SSH Key Pair	10-22
Managing the State of a Key or a Security Object	10-23
About Managing the State of a Key or a Security Object	10-24
How a Multi-Master Cluster Affects Keys and Security Objects	10-24

Activating a Key or Security Object	10-25
Deactivating a Key or Security Object	10-25
Revoking a Key or Security Object	10-26
Destroying a Key or Security Object	10-26
Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault	10-27
About Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault	10-27
Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys	10-28
Managing Details of Security Objects	10-29
About Managing the Details of Security Objects	10-29
Searching for Security Object Items	10-30
Viewing the Details of a Security Object	10-31
Adding or Modifying Details of a Security Object	10-34

11 Managing Oracle Key Vault Master Encryption Keys

Using the Persistent Master Encryption Key Cache	11-1
About the Persistent Master Encryption Key Cache	11-2
About Oracle Key Vault Persistent Master Encryption Key Cache Architecture	11-2
Caching Master Encryption Keys in the In-Memory and Persistent Master Encryption Key Cache	11-3
Storage Location of Persistent Master Encryption Key Cache	11-3
Persistent Master Encryption Key Cache Modes of Operation	11-4
Oracle Key Vault First Mode	11-4
Persistent Master Encryption Key Cache First Mode	11-4
Persistent Master Encryption Key Cache Refresh Window	11-5
Persistent Master Encryption Key Cache Parameters	11-5
PKCS11_CACHE_TIMEOUT Parameter	11-6
PKCS11_PERSISTENT_CACHE_TIMEOUT Parameter	11-6
PKCS11_PERSISTENT_CACHE_FIRST Parameter	11-7
PKCS11_CONFIG_PARAM_REFRESH_INTERVAL Parameter	11-7
PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW Parameter	11-8
EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN Parameter	11-8
Listing the Contents of the Persistent Master Key Cache	11-9
Oracle Database Deployments and Persistent Master Encryption Key Cache	11-10
Configuring an Oracle Key Vault to a New TDE-Enabled Database Connection	11-11
About Configuring an Oracle Key Vault to a New TDE-Enabled Database Connection	11-11
Limitations to Transparent Data Encryption Endpoint Integration	11-12
Step 1: Configure the Oracle Key Vault Server Environment	11-12
Step 2: Integrate Transparent Data Encryption with Oracle Key Vault	11-15
Migrating Existing TDE Wallets to Oracle Key Vault	11-16
About Migrating Existing TDE Wallets to Oracle Key Vault	11-16
Migrating an Existing TDE Wallet to Oracle Key Vault	11-17

Restoring Database Contents Previously Encrypted by TDE Using an Oracle Wallet	11-19
Uploading and Downloading Oracle Wallets	11-20
About Uploading and Downloading Oracle Wallets	11-20
Uploading Oracle Wallets	11-21
Downloading Oracle Wallets	11-22
Guidelines for Uploading and Downloading Oracle Wallets	11-23
Uploading and Downloading JKS and JCEKS Keystores	11-23
About Uploading and Downloading JKS and JCEKS Keystores	11-23
Uploading JKS or JCEKS Keystores	11-24
Downloading JKS or JCEKS Keystores	11-25
Guidelines for Uploading and Downloading JKS and JCEKS Keystores	11-25
Using a User-Defined Key as the TDE Master Encryption Key	11-26
About Using a User-Defined Key as the TDE Master Encryption Key	11-26
Step 1: Upload the User-Defined Key	11-27
Step 2: Activate the User-Defined Key as a TDE Master Encryption Key	11-28

12 Managing Oracle Key Vault Endpoints

Overview of Managing Endpoints	12-1
About Managing Endpoints	12-1
How a Multi-Master Cluster Affects Endpoints	12-3
Managing Endpoints	12-4
Types of Endpoint Enrollment	12-4
Endpoint Enrollment in a Multi-Master Cluster	12-5
Adding an Endpoint as an Oracle Key Vault System Administrator or Create Endpoint User	12-6
Adding Endpoints Using Self-Enrollment	12-10
About Adding Endpoints Using Self-Enrollment	12-10
Adding an Endpoint Using Self-Enrollment	12-10
Deleting, Suspending, Reenrolling, or Rotating Endpoints	12-11
About Deleting Endpoints	12-12
Deleting One or More Endpoints	12-12
Deleting One Endpoint (Alternative Method)	12-13
Suspending an Endpoint	12-13
Reenrolling an Endpoint	12-14
Rotating Endpoint Certificates	12-15
Managing Endpoint Details	12-18
About Endpoint Details	12-18
Modifying Endpoint Details	12-19
Managing Global and Per-Endpoint Configuration Parameters and Settings	12-21
About Managing Global and Per-Endpoint Configuration Parameters and Settings	12-21
Global Endpoint Configuration Parameters and Settings	12-22

Setting Global Endpoint Configuration Parameters	12-22
Configuring Global Endpoint Settings for Keys and Secrets	12-25
Per-Endpoint Configuration Parameters and Settings	12-25
Modifying Configuration Parameters for an Individual Endpoint	12-25
Configuring Endpoint Settings for Keys and Secrets for an Individual Endpoint	12-26
Default Wallets and Endpoints	12-27
Associating a Default Wallet with an Endpoint	12-27
Setting the Default Wallet for an Endpoint	12-27
Managing Endpoint Access to a Virtual Wallet	12-29
Granting an Endpoint Access to a Virtual Wallet	12-29
Revoking Endpoint Access to a Virtual Wallet	12-30
Viewing Wallet Items Accessed by Endpoints	12-31
Managing Endpoint Groups	12-31
How a Multi-Master Cluster Affects Endpoint Groups	12-32
Creating an Endpoint Group	12-32
Modifying Endpoint Group Details	12-34
Granting an Endpoint Group Access to a Virtual Wallet	12-35
Adding an Endpoint to an Endpoint Group	12-36
Removing an Endpoint from an Endpoint Group	12-37
Deleting Endpoint Groups	12-38

13 Enrolling and Upgrading Endpoints for Oracle Key Vault

About Endpoint Enrollment and Provisioning	13-1
Finalizing Enrollment and Provisioning	13-3
Step 1: Enroll the Endpoint and Download the Software	13-3
Step 2: Prepare the Endpoint Environment	13-4
Step 3: Install the Oracle Key Vault Software onto the Endpoint	13-5
Step 4: Perform Post-Installation Tasks	13-7
Environment Variables and Endpoint Provisioning Guidance	13-8
How the Location of JAVA_HOME Location Is Determined	13-9
Location of the okvclient.ora File and Environment Variables	13-9
Setting OKV_HOME for Non-Database Utilities to Communicate with Oracle Key Vault	13-10
Environment Variables in sqlnet.ora File	13-10
Endpoints That Do Not Use the Oracle Key Vault Client Software	13-10
Transparent Data Encryption Endpoint Management	13-11
Endpoint okvclient.ora Configuration File	13-11
okvclient.ora Parameters That Must Not Be Modified	13-13
Upgrading Endpoint Software	13-14
Step 1: Prepare the Endpoint Environment	13-15
Step 2: Download the Oracle Key Vault Software onto the Endpoint	13-15
Step 3: Install the Oracle Key Vault Software onto the Endpoint	13-16

Step 4: Perform Post-Installation Tasks	13-18
Upgrading Endpoint Software on an Enrolled Endpoint	13-19

14 Managing Keys for Oracle Products

Using a TDE-Configured Oracle Database in an Oracle RAC Environment	14-1
Using a TDE-Configured Oracle Database in an Oracle GoldenGate Environment	14-2
Oracle Wallets in an Oracle GoldenGate Environment	14-2
Configuring Online Master Encryption Keys in an Oracle GoldenGate Deployment	14-3
Migration of TDE Wallets in Oracle GoldenGate to Oracle Key Vault	14-3
Using a TDE-Configured Oracle Database in an Oracle Data Guard Environment	14-4
About Uploading Oracle Wallets in an Oracle Data Guard Environment	14-4
Uploading Oracle Wallets in an Oracle Data Guard Environment	14-5
Performing an Online Master Encryption Key Connection in an Oracle Data Guard Environment	14-5
Migrating Oracle Wallets in an Oracle Data Guard Environment	14-5
Reverse Migrating Oracle Wallets in an Oracle Data Guard Environment	14-6
Migrating an Oracle TDE Wallet to Oracle Key Vault for a Logical Standby Database	14-7
Checking the Oracle TDE Wallet Migration for a Logical Standby Database	14-8
Uploading Keystores from Automatic Storage Management to Oracle Key Vault	14-8
About Uploading Keystores from Automatic Storage Management to Oracle Key Vault	14-9
Uploading a Keystore from Automatic Storage Management to Oracle Key Vault	14-9
Copying a Keystore from Oracle Key Vault to Automatic Storage Management	14-10
MySQL Integration with Oracle Key Vault	14-11
Other Oracle Database Features That Oracle Key Vault Supports	14-11

15 SSH Keys Management Concepts

SSH Protocol	15-1
SSH Public Key Authentication	15-2
OpenSSH Implementation of the SSH Protocol	15-3
Challenges with SSH Public Key Authentication	15-3
Limited Decentralized Access Control	15-4
Client Key Management	15-5
Governance	15-6
No Reporting	15-7
Controlling Access to SSH Server Centrally with Oracle Key Vault	15-7
Deployment	15-8
Limiting Access Control to Root User	15-10
Managing SSH User Keys with Oracle Key Vault	15-10
Deployment	15-10
Oracle Key Vault and SSH Integration	15-12

About Oracle Key Vault and SSH integration	15-13
SSH Admin Managing the SSH User Keys and Access to SSH Servers	15-14
SSH Client Users Manage SSH Keys and SSH Admin Manage Access to SSH Servers	15-15
Supported Platforms for SSH Server and Client Endpoints	15-15

16 Management of SSH Keys - Setup and Configuration

Setup SSH Admin	16-1
Controlling Access to SSH Server Centrally with Oracle Key Vault	16-3
About Controlling Access to SSH Server Centrally with Oracle Key Vault	16-3
Setup in Oracle Key Vault Server	16-4
Setup on SSH Server	16-6
Manage SSH Server Access from Oracle Key Vault	16-10
Managing SSH User Keys with Oracle Key Vault	16-12
About Managing the SSH User Keys with Oracle Key Vault	16-12
Oracle Key Vault Server Setup	16-13
SSH Client Host Setup	16-16
Authorize SSH User's Public Key on SSH Server	16-18
Connect to SSH Servers using Oracle Key Vault PKCS#11 library	16-19
Oracle Key Vault and SSH Integration	16-20
SSH Administrators Managing both the SSH User's Keys and the SSH Server Access	16-20
SSH Client Managing SSH User Keys and SSH Admin Managing the SSH Server Access	16-21
Migrating Existing SSH Deployments to Oracle Key Vault	16-21
Guidelines for OpenSSH SSHD Configuration	16-22
Reports	16-23
Viewing SSH Reports	16-23
SSH Key Details Reports	16-24
SSH Server Access Management Reports	16-25
SSH User Private Key Management Reports	16-28

17 Managing Online and Offline Secrets

Uploading and Downloading Credential Files	17-1
About Uploading and Downloading Credential Files	17-1
Uploading a Credential File	17-2
Downloading a Credential File	17-3
Guidelines for Uploading and Downloading Credential Files	17-3
Managing Secrets and Credentials for SSH	17-4
Integrating Oracle Key Vault with SSH Public Key Authentication	17-4
Step 1: Creating and Uploading the Key Pair	17-5
Step 2: Using an Endpoint to SSH a Remote Host	17-6

Step 3: Incorporating an ssh-agent	17-7
Centrally Managing Passwords in Oracle Key Vault	17-8
About Centrally Managing Passwords in Oracle Key Vault	17-8
Creating and Sharing Centrally Managed Passwords	17-9
Example: Script for Storing and Retrieving Oracle Database User Passwords in and from Oracle Key Vault	17-10
Sharing Secrets with Other Databases	17-12
Changing Passwords for a Large Database Deployment	17-13

18 Oracle Key Vault General System Administration

Overview of Oracle Key Vault General System Administration	18-1
About Oracle Key Vault General System Administration	18-2
Viewing the Oracle Key Vault Dashboard	18-2
Using the Status Panes in the Dashboard	18-2
Configuring Oracle Key Vault in a Non-Multi-Master Cluster Environment	18-4
Configuring the Network Details	18-4
Configuring Network Access	18-6
Configuring DNS	18-9
Configuring the System Time	18-9
Configuring FIPS Mode	18-11
Configuring Syslog	18-12
Changing the Network Interface Mode	18-12
Configuring RESTful Services Utility	18-14
Checking the Oracle Audit Vault Integration Status	18-15
Configuring the Oracle Key Vault Management Console Web Session Timeout	18-15
Restarting or Powering Off Oracle Key Vault	18-16
Configuring Oracle Key Vault in a Multi-Master Cluster Environment	18-16
About Configuring Oracle Key Vault in a Multi-Master Cluster Environment	18-16
Configuring System Settings for Individual Multi-Master Cluster Nodes	18-17
Configuring the Network Details for the Node	18-18
Configuring Network Access for the Node	18-18
Configuring DNS for the Node	18-19
Configuring the System Time for the Node	18-20
Configuring the FIPS Mode for the Node	18-20
Configuring Syslog for the Node	18-21
Changing the Network Interface Mode for the Node	18-21
Configuring Auditing for the Node	18-23
Configuring SNMP Settings for the Node	18-24
Checking the Oracle Audit Vault Integration for the Node	18-25
Restarting or Powering Off Oracle Key Vault from a Node	18-25
Configuring System Settings for an Entire Oracle Key Vault Multi-Master Cluster	18-25

Configuring the System Time for the Cluster	18-26
Configuring DNS for the Cluster	18-27
Configuring the Maximum Disable Node Duration for the Cluster	18-27
Configuring Syslog for the Cluster	18-27
Configuring RESTful Services for the Cluster	18-28
Configuring Auditing for the Cluster	18-28
Configuring SNMP Settings for the Cluster	18-29
Configuring the Oracle Key Vault Management Console Web Session Timeout for the Cluster	18-29
Configuring the Service Ports	18-30
Managing System Recovery	18-32
About Managing System Recovery	18-32
Recovering Credentials for Administrators	18-33
Changing the Recovery Passphrase in a Non-Multi-Master Cluster Environment	18-33
Changing the Recovery Passphrase in a Multi-Master Cluster	18-34
About Changing the Recovery Passphrase for a Multi-Master Cluster	18-34
Step 1: Initiate the Recovery Passphrase Change Across the Nodes	18-34
Step 2: Change the Recovery Passphrase	18-35
Support for a Primary-Standby Environment	18-36
Commercial National Security Algorithm Suite Support	18-37
About Commercial National Security Algorithm Suite Support	18-37
Running the Commercial National Security Algorithm Scripts	18-38
Performing Backup Restore Operations with CNSA	18-39
Upgrading a Standalone Oracle Key Vault Server with CNSA	18-39
Upgrading Primary-Standby Oracle Key Vault Servers to Use CNSA	18-41
Minimizing Downtime	18-42

19 Managing Service Certificates

Overview of Oracle Key Vault Certificates	19-1
Certificates Validity Period	19-3
About Certificates Validity Period	19-4
Setting Validity Period of Self-Signed Root CA Certificate	19-5
Configuring Certificate Validity Period for Server and Node Certificates	19-6
About Configuring Certificate Validity Period for Endpoint Certificates	19-6
Monitoring Certificates Expiry	19-7
Monitoring Certificates Expiry Using Certificate Expiration Alerts	19-7
Finding the Expiration Date of Endpoint Certificates	19-8
CA Certificate Expiration Date on Status Page	19-9
Server and Node Certificate Expiration on Status Page	19-9
Finding the Expiration Date of the CA Certificate	19-10
Finding the Expiration Date of Server Certificates and Node Certificates	19-11

Managing CA Certificate Rotation	19-13
Steps for Managing CA Certificate Rotation	19-14
Checking for Self-Signed Root CA or Intermediate CA Certificate	19-14
Setting the Key Length of the CA Certificate	19-15
Setting the Validity of Self-Signed Root CA Certificate	19-15
Setting Up the Intermediate CA Certificate	19-16
Rotating CA Certificate	19-19
Setting the Endpoint Certificate Rotation Batch Size	19-25
Setting the Endpoint Certificate Rotation Sequence	19-26
Checking Overall Certificate Rotation Status	19-27
Checking Certificate Rotation Status for Endpoints	19-28
Post-CA Certificate Rotation Tasks	19-28
Factors Affecting CA Certificate Rotation Process	19-29
Guidelines for Managing CA Certificate Rotations	19-30
Managing Server Certificates and Node Certificates Rotation	19-32
About Server Certificates and Node Certificates Rotation	19-32
Configuring Certificate Validity Period for Server and Node Certificates	19-33
Rotating Server Certificates and Node Certificates	19-33
Guidelines for Rotating Server Certificates and Node Certificates	19-34
Managing the Oracle Key Vault CA Certificate After Expiry	19-35
Configuring Oracle Key Vault with an Alternate Hostname	19-38
About Configuring Oracle Key Vault with an Alternate Hostname	19-38
Configuring Oracle Key Vault Alternate Hostname on the Management Console	19-39
Choosing the Alternate Hostname to Use in Endpoint Configuration	19-40
Guidelines for Configuring Alternate Hostnames	19-40
Managing Platform Certificates	19-41
About Platform Certificates	19-41
Guidelines for Rotating Platform Certificates	19-42
Rotating Platform Certificates	19-42

20 Managing Console Certificates

About Managing Console Certificates	20-1
Step 1: Download the Certificate Request	20-1
Step 2: Have the Certificate Signed	20-3
Step 3: Upload the Signed Certificate to Oracle Key Vault	20-3
Console Certificates in Special Use Case Scenarios	20-4

21 Backup and Restore Operations

About Backing Up and Restoring Data in Oracle Key Vault	21-1
Oracle Key Vault Backup Destinations	21-2

About the Oracle Key Vault Backup Destination	21-2
Creating a Remote Backup Destination	21-4
Changing Settings on a Remote Backup Destination	21-6
Deleting a Remote Backup Destination	21-7
Scheduled Backups and States	21-7
About Schedule Backup Types and States	21-8
Types of Oracle Key Vault Backups	21-8
Scheduled Backup States in Oracle Key Vault	21-9
Scheduling and Managing Oracle Key Vault Backups	21-9
Scheduling a Backup for Oracle Key Vault	21-9
Changing a Backup Schedule for Oracle Key Vault	21-11
Deleting a Backup Schedule from Oracle Key Vault	21-11
How Primary-Standby Affects Oracle Key Vault Backups	21-11
How Using a Cluster Affects Oracle Key Vault Backups	21-12
Protecting the Backup Using the Recovery Passphrase	21-12
Restoring Oracle Key Vault Data	21-12
About the Oracle Key Vault Restore Process	21-13
Procedure for Restoring Oracle Key Vault Data	21-13
Multi-Master Cluster and the Restore Operation	21-15
Primary-Standby and the Restore Operation	21-15
Certificates and the Restore Operation	21-15
Changes Resulting from a System State Restore	21-16
Scheduling the Purging of Old Oracle Key Vault Backups	21-17
About Scheduling the Purging of Old Oracle Key Vault Backups	21-17
Creating a Backup Destination Policy	21-18
Adding a Backup Destination Policy to a Remote Backup Destination	21-19
Changing a Backup Destination Policy	21-19
Suspending a Backup Destination Policy	21-20
Resuming a Suspended Backup Destination Policy	21-20
Deleting a Backup Destination Policy	21-21
Finding Information about Backup Destination Policies	21-21
Manually Deleting a Local Oracle Key Vault Backup	21-22
Configuring Oracle ZFS Storage Appliance to Store Oracle Key Vault Backups	21-22
Step 1: Create a Storage Project in Oracle ZFS Storage Appliance	21-23
Step 2: Copy the Oracle Key Vault Public Key to the Oracle ZFS Storage Appliance	21-23
Step 3: Complete Creating the Oracle ZFS Storage Appliance Project	21-24
Step 4: Configure Oracle Key Vault to Connect to the Oracle ZFS Storage Appliance Project	21-25
Backup and Restore Best Practices	21-26

22 Monitoring and Auditing Oracle Key Vault

Managing System Monitoring	22-1
Configuring Remote Monitoring to Use SNMP	22-1
About Using SNMP for Oracle Key Vault	22-2
Granting SNMP Access to Users	22-3
Changing the SNMP User Name and Password	22-4
Changing SNMP Settings on the Standby Server	22-4
Remotely Monitoring Oracle Key Vault Using SNMP	22-5
SNMP Management Information Base Variables for Oracle Key Vault	22-6
Example: Simplified Remote Monitoring of Oracle Key Vault Using SNMP	22-8
Configuring SNMP to Work with Old SNMP Clients	22-10
Configuring Email Notification	22-11
About Email Notification	22-11
Configuring Email Settings	22-12
Testing the Email Configuration	22-14
Disabling Email Notifications for a User	22-15
Configuring the Syslog Destination for Individual Multi-Master Cluster Nodes	22-15
Setting the Syslog Destination Setting for the Node	22-15
Clearing the Syslog Destination Setting for the Node	22-16
Capturing System Diagnostics	22-16
About Capturing System Diagnostics	22-16
Configuring the Oracle Key Vault Application Tracing Level	22-17
Downloading the Diagnostics Package	22-18
Unpacking the Diagnostics Package	22-19
Deleting Trace Files	22-21
Monitoring System Metrics	22-21
About Capturing System Metrics	22-22
Viewing System Metrics	22-22
Configuring Oracle Key Vault Alerts	22-25
About Configuring Alerts	22-25
Configuring Alerts	22-40
Alert Configuration at Cluster vs Node Scope	22-42
Viewing Open Alerts	22-43
Managing System Auditing	22-46
About Auditing in Oracle Key Vault	22-47
Oracle Key Vault Audit Trail	22-47
Enabling Auditing and Configuring Syslog to Store Audit Records	22-48
Oracle Key Vault Audit Configuration	22-49
About Oracle Key Vault Audit Configuration	22-49
Configuring All Events Auditing	22-50
Configuring Custom Events Auditing	22-51

Configuring Auditing on Event Categories	22-52
Including Specific Audit Event IDs for Auditing	22-54
Excluding Specific Audit Event IDs from Auditing	22-55
Viewing Audit Records	22-57
Exporting and Deleting Audit Records Manually	22-58
Deleting Audit Records Automatically	22-58
Oracle Key Vault Audit Event IDs	22-59
Configuring Oracle Key Vault with Oracle Audit Vault	22-67
Integrating Oracle Audit Vault with Oracle Key Vault	22-68
Viewing Oracle Key Vault Audit Data Collected by Oracle Audit Vault	22-70
Suspending an Oracle Audit Vault Monitoring Operation	22-71
Resuming an Oracle Audit Vault Monitoring Operation	22-71
Deleting an Oracle Audit Vault Integration	22-71
Guidance for Integrating Oracle Audit Vault in a Multi-Master Cluster or Primary- Standby Environment	22-72
Using Oracle Key Vault Reports	22-73
About Oracle Key Vault Reports	22-73
Viewing Key Management Reports for Oracle Endpoints	22-74
Viewing Keys and Wallets Reports	22-75
Viewing Secrets Management Reports	22-76
Viewing SSH Reports	22-76
Viewing Endpoint Reports	22-77
Viewing User Reports	22-78
Viewing System Reports	22-79

23 Managing an Oracle Key Vault Primary-Standby Configuration

Overview of the Oracle Key Vault Primary-Standby Configuration	23-1
About the Oracle Key Vault Primary-Standby Configuration	23-2
Benefits of an Oracle Key Vault Primary-Standby Configuration	23-3
Difference Between Primary-Standby Configuration and Multi-Master Cluster	23-4
Primary Server Role in a Primary-Standby Configuration	23-4
Standby Server Role in a Primary-Standby Configuration	23-4
Configuring the Primary-Standby Environment	23-5
Step 1: Configure the Primary Server	23-5
Step 2: Configure the Standby Server	23-6
Step 3: Complete the Configuration on the Primary Server	23-8
Switching the Primary and Standby Servers	23-9
Restoring Primary-Standby After a Failover	23-10
Disabling or Unpairing the Primary-Standby Configuration	23-11
Read-Only Restricted Mode in a Primary-Standby Configuration	23-12
About Read-Only Restricted Mode in a Primary-Standby Configuration	23-12

Primary-Standby with Read-Only Restricted Mode	23-13
Primary-Standby without Read-Only Restricted Mode	23-13
States of Read-Only Restricted Mode	23-14
About the States of Read-Only Restricted Mode	23-15
Read-Only Restricted State Functionality During a Primary Server Failure	23-16
Read-Only Restricted Mode Functionality During a Standby Server Failure	23-16
Read-Only Restricted State Functionality During a Network Failure	23-16
Enabling Read-Only Restricted Mode	23-17
Disabling Read-Only Restricted Mode	23-17
Recovering from Read-Only Restricted Mode	23-18
Read-Only Restricted Mode Notifications	23-19
Best Practices for Using Oracle Key Vault in a Primary-Standby Configuration	23-19

24 Oracle Key Vault Integration Accelerator

Key Management for Oracle DBMS_CRYPT0 Integration Accelerator	24-1
Downloading Key Management for Oracle DBMS_CRYPT0 Integration Accelerator	24-1
Using Key Management for Oracle DBMS_CRYPT0 Integration Accelerator	24-2
Prerequisites for Using Key Management for Oracle DBMS_CRYPT0 Integration Accelerator	24-2
Preparing the Environment for Using Key Management for Oracle DBMS_CRYPT0 Integration Accelerator	24-3
Using Key Management for Oracle DBMS_CRYPT0 Integration Accelerator in a PL/SQL Application	24-6
Extending the Functionality of Key Management for Oracle DBMS_CRYPT0 Integration Accelerator	24-7

A Oracle Key Vault Multi-Master Cluster Operations

B Oracle Key Vault okvutil Endpoint Utility Reference

About the okvutil Utility	B-1
okvutil Command Syntax	B-2
okvutil changepwd Command	B-3
okvutil diagnostics Command	B-3
okvutil download Command	B-4
okvutil list Command	B-7
okvutil upload Command	B-9
okvutil sign Command	B-12
okvutil sign-verify Command	B-14
okvutil show Command	B-15

C Troubleshooting Oracle Key Vault

Before You Start Troubleshooting	C-1
Endpoint Related Issues	C-2
Common Oracle Key Vault Tasks	C-2
How to Re-Enroll an Endpoint on an Endpoint Database	C-3
How To Download Diagnostics From Oracle Key Vault Server	C-4
How to Recover the root User Password	C-4
How to Reset the support User Password	C-6
How to Add SAN Details to the Console Certificate	C-6
How to Check for the Patch Updates	C-8
okvutil and Endpoint Issues	C-8
Database Wallet Status Not Open or Not Found, TDE HEARTBEAT Check Failed	C-9
Oracle Key Vault Server Communication or Connection Failed Error	C-14
Could Not Store Private Key Errors on Wallet Upload	C-18
RESTful Services Endpoint Provisioning Command Failure	C-19
Uploading Certificate File Failure	C-19
Error in Uploading the Java Keystore	C-20
SSL layer Error while migrating MYSQL Database Keys to Oracle Key Vault	C-21
Rotation or Set Key Failure in Windows Environment	C-21
Rotation or Set Key Fails with ORA-03113	C-22
Multi-Master Cluster Issues	C-22
Heartbeat Lag or High Replication Lag in Multi-Master Cluster Environment	C-23
Cluster Node Pairing Failure	C-24
Adding a Node to Cluster Fails with Invalid Certificate or Certificate Expired Error	C-25
How to Diagnose Oracle Key Vault Cluster Issues	C-26
Cluster-Node Pairing Issues	C-26
Backup and Restore Issues	C-27
Oracle Key Vault Backup Failed Error	C-28
Unable to Schedule a New Backup	C-28
Remote Backup Failed	C-29
Backup Restore Failure	C-30
Backup Size is Growing Exponentially	C-30
Certificate Related Issues	C-31
Updating to Current Certificate Issuer	C-31
Installation and Upgrade Issues	C-32
Oracle Key Vault Installation Failure	C-32
Oracle Key Vault Upgrade Failure	C-33
Oracle Key Vault Management Console is Not Accessible After Installation	C-34
Oracle Key Vault Upgrade Failure	C-34

Unable to boot after installation of Oracle Key Vault on VMWare VM	C-35
Operation Failed on Network Information Screen After Upgrade from 21.x to 21.5 and Later.	C-36
Primary-Standby Configuration Issues	C-36
Write Operations Fail in Restricted Mode	C-37
Fast-Start Failover Suspended	C-37
How to Verify Primary-Standby Status	C-38
DBCS Endpoint Configuration Issues	C-42
SSH Tunnel Add Failure	C-42
Server and Node Issues	C-43
SSL Client Error Message	C-43
Incorrect Value Returned for Custom Attributes of Integer Type	C-43
Not Receiving Email Alerts	C-45
Oracle Key Vault Server and NTP Server Date and Time Not Synchronized	C-46
Failed To Enable FIPS mode	C-46
Non-Oracle Database Endpoints Fails to Connect to Oracle Key Vault	C-47

D Security Technical Implementation Guides Compliance Standards

About Security Technical Implementation Guides	D-1
Enabling and Disabling STIG Rules on Oracle Key Vault	D-2
Enabling STIG Rules on Oracle Key Vault	D-2
Disabling STIG Rules on Oracle Key Vault	D-2
Current Implementation of STIG Rules on Oracle Key Vault	D-2
Current Implementation of Database STIG Rules	D-3
Additional STIG Guidelines Notes	D-15
DG0116-ORACLE11 STIG Guideline	D-15
Current Implementation of Operating System STIG Rules	D-15

E Managing Oracle Key Vault Platform Certificates

Overview of Oracle Key Vault Platform Certificates	E-1
Monitoring Oracle Key Vault Platform Certificate Expiration	E-1
Finding the Expiration Date of Platform Certificates	E-2
Monitoring Platform Certificates Expiration Using Platform Certificate Expiration Alerts	E-2
Rotating Platform Certificates	E-2
Rotating Platform Certificates on a Standalone Oracle Key Vault Server	E-3
Rotating Platform Certificates in a Multi-Master Cluster Environment	E-5
Rotate Platform CA Certificate on Read/Write Multi-Master Cluster Nodes	E-5
Rotate Platform CA Certificate on Read-Only Multi-Master Cluster Nodes	E-8
Rotate Platform Certificate Used For Redo Shipping On Any One Multi-Master Cluster Node	E-9

Glossary

Index

Preface

Welcome to *Oracle Key Vault Administrator's Guide*. This guide explains how to configure and use Oracle Key Vault.

- [Audience](#)
- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Documents](#)
- [Conventions](#)

Audience

Oracle Key Vault Administrator's Guide is written for Oracle security administrators who are responsible for managing and centralizing encryption keys and other security objects.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For more information, see these Oracle resources:

- *Oracle Key Vault Root of Trust HSM Configuration Guide*
- *Oracle Key Vault RESTful Services Administrator's Guide*

- *Oracle Key Vault Developer's Guide*
- *Oracle Key Vault Licensing Information*
- *Oracle Key Vault Release Notes*
- [Key Management Interoperability Protocol Specification Version 1.1](#)

To download the product data sheet, frequently asked questions, links to the latest product documentation, product download, and other collateral, visit Oracle Technical Resources (formerly Oracle Technology Network). You must register online before using Oracle Technical Services. Registration is free and can be done at

<https://www.oracle.com/technical-resources/>

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Changes in This Release for Oracle Key Vault

In each release, Oracle Key Vault introduces several new features and enhancements to improve performance, configuration, monitoring, and reporting.

- [Changes for Oracle Key Vault Release 21.10](#)
Oracle Key Vault release 21.10 introduces several new features.
- [Changes for Oracle Key Vault Release 21.9](#)
Oracle Key Vault release 21.9 introduces several new features.
- [Changes for Oracle Key Vault Release 21.8](#)
Oracle Key Vault release 21.8 introduces several new features.
- [Changes for Oracle Key Vault Release 21.7](#)
Oracle Key Vault release 21.7 introduces several new features.
- [Changes for Oracle Key Vault Release 21.6](#)
Oracle Key Vault release 21.6 introduces several new features.
- [Changes for Oracle Key Vault Release 21.5](#)
Oracle Key Vault release 21.5 introduces several new features.
- [Changes for Oracle Key Vault Release 21.4](#)
Oracle Key Vault release 21.4 introduces new features that affect this guide.
- [Changes for Oracle Key Vault Release 21.3](#)
Oracle Key Vault release 21.3 introduces new features that affect this guide.

Changes for Oracle Key Vault Release 21.10

Oracle Key Vault release 21.10 introduces several new features.

- [Key Management for DBMS_CRYPTO with Extensible Oracle Key Vault Integration Accelerator](#)
Starting with release 21.10, Oracle Key Vault provides an integration accelerator as an implementation package to demonstrate the use of Oracle Key Vault in a specific use-case.
- [Configurable Oracle Key Vault Ports](#)
Starting with Oracle Key Vault release 21.10, you can now configure the ports used to communicate between read/write peers and connect to Oracle Key Vault remotely using SSH.
- [New Oracle Key Vault Endpoint Platforms - Linux for Arm \(aarch64\) Architecture and IBM: Linux on System z](#)
Starting with Oracle Key Vault release 21.10, you can provision Oracle Key Vault endpoints on the Linux for Arm (aarch64) architecture and IBM: Linux on System z.
- [Manage SSH User Keys with Oracle Key Vault Endpoints on Windows](#)
Oracle Key Vault release 21.10 can manage the SSH user keys of openssh clients on Windows.

- [Monitor Privilege for Users Monitoring Oracle Key Vault](#)
You should grant the **Monitor** privilege added with Oracle Key Vault release 21.10 to users monitoring Oracle Key Vault cluster, nodes, and servers.
- [Automated Rotation of Platform Certificates](#)
In Oracle Key Vault release 21.10, the platform certificate rotation is automated and you can initiate it from the management console.
- [List Recently Active or Inactive Endpoint using the RESTful Services Utility Command](#)
Starting with Oracle Key Vault release 21.10, you can list the endpoints that were active or inactive recently using the RESTful services utility.
- [Raise Alert for Password Expiry of the Oracle Key Vault System Users \(support and root\)](#)
Oracle Key Vault release 21.10 adds a new alert type **System User Password Expiration** to monitor the password expiration of `root` and `support` system users.
- [Monitoring CPU Usage and RESTful Services Memory Usage](#)
Oracle Key Vault release 21.10 provides additional monitoring information about CPU usage and RESTful services memory usage through the `SNMP nsExtendOutputFull MIB` base variable.

Key Management for DBMS_CRYPTO with Extensible Oracle Key Vault Integration Accelerator

Starting with release 21.10, Oracle Key Vault provides an integration accelerator as an implementation package to demonstrate the use of Oracle Key Vault in a specific use-case.

Oracle Key Vault release 21.10 provides **Key Management for DBMS_CRYPTO**. This integration accelerator demonstrates how to manage keys used with DBMS_CRYPTO operations in Oracle Key Vault, using a PL/SQL wrapper built on top of Oracle Key Vault JAVA SDK.

You can download the **Key Management for DBMS_CRYPTO** integration accelerator from the Oracle Key Vault management console and use that without any change or customize the implementation further for your specific needs.

Configurable Oracle Key Vault Ports

Starting with Oracle Key Vault release 21.10, you can now configure the ports used to communicate between read/write peers and connect to Oracle Key Vault remotely using SSH.

Oracle Key Vault uses port 1522 to communicate between read/write peers. The SSH service running on port 22 allows for remote administration of Oracle Key Vault. For certain deployments, it may not be possible to configure the network to allow ingress or egress traffic on these default ports. You can now configure these ports to non-default values for cluster operations and administrative maintenance.

New Oracle Key Vault Endpoint Platforms - Linux for Arm (aarch64) Architecture and IBM: Linux on System z

Starting with Oracle Key Vault release 21.10, you can provision Oracle Key Vault endpoints on the Linux for Arm (aarch64) architecture and IBM: Linux on System z.

Oracle Databases running on Linux for Arm (aarch64) or IBM: Linux on System z can also be enrolled as Oracle Key Vault endpoints for TDE key management. You can set up endpoints for SSH servers or clients on both these platforms. All the endpoint software like `okvutil`,

PKCS#11 library, the RESTful services utility, Java, and C SDK are available on these platforms.

The new endpoint platforms are supported with the Oracle Key Vault 21 RESTful services utility but not available with the deprecated classic RESTful services utility.

Manage SSH User Keys with Oracle Key Vault Endpoints on Windows

Oracle Key Vault release 21.10 can manage the SSH user keys of openssh clients on Windows.

You can use SSH keys generated in Oracle Key Vault with openssh clients on Windows. Oracle Key Vault endpoints deployed on Windows include a new PKCS#11 library for use with openssh clients. Public and private keys of SSH users on Windows can now be managed centrally with private keys set up to be non-extractable.

Monitor Privilege for Users Monitoring Oracle Key Vault

You should grant the **Monitor** privilege added with Oracle Key Vault release 21.10 to users monitoring Oracle Key Vault cluster, nodes, and servers.

The **Monitor** privilege allows grantees to view information relevant for monitoring Oracle Key Vault from the management console. It also enables them to fetch server configuration, status, and metrics information using the RESTful services utility commands. Users monitoring Oracle Key Vault no longer need the powerful system administration role for this purpose as granting them the **Monitor** privilege is sufficient.

Automated Rotation of Platform Certificates

In Oracle Key Vault release 21.10, the platform certificate rotation is automated and you can initiate it from the management console.

Oracle Key Vault raises alerts when the platform certificates are expiring. You can configure how long before the platform certificate expires, the alerts should be generated. Once the alerts are raised, you can initiate the platform certificate rotation for the cluster.

During the rotation process for the cluster, the platform certificates will be rotated in a predefined order. Platform certificates for the nodes with the lowest node ID and their read/write peers will be rotated first followed by other nodes with the next lowest node ID. The read/write peers will operate in read-only mode for a short period while their platform certificates are rotated.

List Recently Active or Inactive Endpoint using the RESTful Services Utility Command

Starting with Oracle Key Vault release 21.10, you can list the endpoints that were active or inactive recently using the RESTful services utility.

To list the endpoint that were recently active, use the following command:

- `okv admin endpoint list --last-active-duration <last-active-duration>`

To list the endpoint with no recent activity, use the following command:

- `okv admin endpoint list --last-inactive-duration <last-inactive-duration>`

You can use the commands to check how often the endpoints connect to the Oracle Key Vault server. You can also use the commands to identify the defunct endpoints for house-keeping.

Raise Alert for Password Expiry of the Oracle Key Vault System Users (support and root)

Oracle Key Vault release 21.10 adds a new alert type **System User Password Expiration** to monitor the password expiration of `root` and `support` system users.

Oracle Key Vault raises an alert when the password for `root` or `support` users is expiring. You can configure how long before the password expires, the alerts should be generated. The default value is 14 days. You must change the passwords for the `root` and `support` users before they expire.

Monitoring CPU Usage and RESTful Services Memory Usage

Oracle Key Vault release 21.10 provides additional monitoring information about CPU usage and RESTful services memory usage through the `SNMP nsExtendOutputFull` MIB base variable.

The `nsExtendOutputFull` MIB base variable now includes CPU usage information. It also shows the Tomcat status as failed if the Tomcat is unresponsive due to Out of Memory (OOM) error.

Changes for Oracle Key Vault Release 21.9

Oracle Key Vault release 21.9 introduces several new features.

- [Configure Auditing for Select Audit Event IDs and Audit Event Categories](#)
When you enable auditing in Oracle Key Vault, all system operations are audited by default. Starting with Oracle Key Vault release 21.9, you can selectively enable auditing of specific operations based on the audit event categories and audit event IDs.
- [Describe Endpoint Details Using `okvutil`](#)
In Oracle Key Vault release 21.8 and earlier, it was not possible for an endpoint to gather the endpoint details like endpoint name, description, or default wallet information about itself. For this information, the endpoint depends on the Oracle Key Vault administrator.
- [Improve Overall Certificate Rotation Time](#)
Starting with Oracle Key Vault release 21.9, endpoints are redirected to the node that generated their new certificates to speed up the CA certificate rotation process.

Configure Auditing for Select Audit Event IDs and Audit Event Categories

When you enable auditing in Oracle Key Vault, all system operations are audited by default. Starting with Oracle Key Vault release 21.9, you can selectively enable auditing of specific operations based on the audit event categories and audit event IDs.

Audit event ID identifies the audit operation type. Audit event IDs for operations that are functionally related are grouped into audit event categories.

You can use custom events configuration to selectively enable auditing for operations of your interest using audit event categories, individual audit event IDs or combination of both. Selecting an audit event category enables auditing of all events from that category. You can also selectively exclude an audit event from the selected audit event category.

You can modify and save custom events audit configuration before making it effective. Once you complete custom events audit configuration, you can switch from auditing all events to auditing custom events to apply the configuration.

**Note:**

Selective auditing may impact the reports that are based on the audit report.

Describe Endpoint Details Using `okvutil`

In Oracle Key Vault release 21.8 and earlier, it was not possible for an endpoint to gather the endpoint details like endpoint name, description, or default wallet information about itself. For this information, the endpoint depends on the Oracle Key Vault administrator.

Starting with Oracle Key Vault release 21.9, endpoint managers can use `okvutil` on the endpoint host to list the endpoint details.

Improve Overall Certificate Rotation Time

Starting with Oracle Key Vault release 21.9, endpoints are redirected to the node that generated their new certificates to speed up the CA certificate rotation process.

Endpoint certificates have to be regenerated when CA certificates are rotated. Endpoint certificates are generated by a designated cluster node and shipped back to the endpoint when the endpoint communicates with the designated cluster node. Considering that the endpoint picks a cluster node at random from the endpoint node scan list to communicate with, it may take some time for the endpoint to connect to the designated node, especially since each successive attempt may only be made after the expiration of the time for the keys to live in persistent cache. CA certificate rotation is incomplete till all the endpoint certificates are rotated.

Starting with Oracle Key Vault release 21.9, when the endpoint certificates are ready and the endpoint communicates with a cluster node, then Oracle Key Vault node redirects the endpoint to the designated cluster node to fetch the certificate. This process eliminates the need for multiple attempts to fetch the endpoint certificate thus accelerating the completion of the CA certificate rotation.

Changes for Oracle Key Vault Release 21.8

Oracle Key Vault release 21.8 introduces several new features.

- [Enforce Separation of Administrator Roles](#)
Starting with Oracle Key Vault release 21.8, you can configure Oracle Key Vault such that an Oracle Key Vault user can have no more than one Oracle Key Vault administrative role.
- [Console Certificate Supports Subject Alternative Name \(SAN\)](#)
Starting with Oracle Key Vault release 21.8, Oracle Key Vault supports a fully qualified domain name (FQDN) for the console certificates without changing the hostname.
- [Alert for Platform Certificate Expiration](#)
Starting with Oracle Key Vault release 21.8, an alert will be generated when the Oracle Key Vault platform certificates are going to expire within a period defined by the alert configuration.

- [Upload of Public and Private Keys through okvutil](#)
With Oracle Key Vault release 21.8, `okvutil` has been enhanced to support Public and Private Keys.
- [Configurable Key Lengths for Service Certificates](#)
Starting with Oracle Key Vault release 21.8, you can configure the key lengths for the service certificates.

Enforce Separation of Administrator Roles

Starting with Oracle Key Vault release 21.8, you can configure Oracle Key Vault such that an Oracle Key Vault user can have no more than one Oracle Key Vault administrative role.

In previous Oracle Key Vault releases, the Oracle Key Vault administrative roles (System Administrator, Key Administrator, and Audit Manager) could be granted to another Oracle Key Vault user by any user who currently has the role with **Allow Forward Grant** option.

Starting with this release, you enable **Enforce Separation of Administrator Roles** check from the **Account Management** tab of the **System Recovery** page to prevent the grant of more than one Oracle Key Vault administrative role to any Oracle Key Vault user. Each user can have at most one administrative role. Enabling the **Enforce Separation of Administrator Roles** option enforces administrative role isolation in Oracle Key Vault.

An Oracle Key Vault user may already have more than one administrative role when the **Enforce Separation of Administrator Roles** option is enabled. In this case, the user cannot exercise any of the administrative roles even though they have been granted to him. The additional roles have to be removed before the remaining one (intended) administrative role can become operational.

Console Certificate Supports Subject Alternative Name (SAN)

Starting with Oracle Key Vault release 21.8, Oracle Key Vault supports a fully qualified domain name (FQDN) for the console certificates without changing the hostname.

Prior to Oracle Key Vault release 21.8, you would have to change the hostname of the Oracle Key Vault to a fully qualified domain name (FQDN) of the Oracle Key Vault server when generating console certificates to avoid HTTPS certificate browser warnings.

Starting with Oracle Key Vault release 21.8, this change is no longer necessary. You can retain the hostname when you add an FQDN as the subject alternative name for the console certificate. Subject alternative name (SAN) is an extension to the X.509 certificate specification that allows the inclusion of additional hostname in the certificate. You can also use this feature to resolve two FQDNs to the same Oracle Key Vault server, if needed. But in this case, you must change the hostname to one of the two FQDNs.

Alert for Platform Certificate Expiration

Starting with Oracle Key Vault release 21.8, an alert will be generated when the Oracle Key Vault platform certificates are going to expire within a period defined by the alert configuration.

Platform certificates are used when a new node is added to the cluster. They are also used when shipping redo between read/write nodes of a cluster. These certificates are different from the Oracle Key Vault service certificates and have different expiration dates. They are also rotated using a different method than the Oracle Key Vault service certificates. You must rotate the platform certificates before they expire. You cannot upgrade an Oracle Key Vault system with the expired platform certificates. The default configuration for the alert is 90 days. You can

change the configured value. The alert will be raised for standalone, multi-master cluster, and primary-standby deployments.

Related Topic

[About Configuring Alerts](#)

Upload of Public and Private Keys through okvutil

With Oracle Key Vault release 21.8, `okvutil` has been enhanced to support Public and Private Keys.

In earlier releases, RESTful services needed to be installed and configured to upload public and private keys to Oracle Key Vault. With Oracle Key Vault release 21.8, `okvutil` has been enhanced to support uploading of public and private key directly, greatly simplifying the enforcement of remote server access controls for public key authentication.

Configurable Key Lengths for Service Certificates

Starting with Oracle Key Vault release 21.8, you can configure the key lengths for the service certificates.

You can choose to set the key length of the certificate authority (CA) certificate to either 2048 bits or 4096 bits. When the next CA certificate rotation is performed, the CA, server/node, and endpoint certificates will be generated with the selected key length. Service certificates with configurable key lengths enable compliance with corporate information security policies.

Changes for Oracle Key Vault Release 21.7

Oracle Key Vault release 21.7 introduces several new features.

- [Controlling Access to SSH Servers Centrally with Oracle Key Vault](#)
Starting with this release, you can use Oracle Key Vault to centrally manage the access to the SSH servers within your enterprise.
- [Improved SSH User Keys Management](#)
Oracle Key Vault release 21.7 further improves the centralized SSH user keys management.
- [RESTful Services Utility Changes to Support SSH Keys Management](#)
Starting with release 21.7, you can use the Oracle Key Vault RESTful services utility to create and register SSH keys and manage SSH Server wallets and SSH Server endpoints.
- [Support Key Creation from Oracle Key Vault Management Console](#)
Starting with release 21.7, you can now create new keys and key pairs from Oracle Key Vault Management console. This allows Oracle Key Vault users to create the keys and key pairs. Previously, only endpoints could create security objects.
- [Support for Node or Cluster Scope for Alerts in Multi-Master Cluster](#)
Starting with Oracle Key Vault release 21.7, in a multi-master cluster, you can set the configuration for certain alerts at the Node or Cluster scope.
- [Setting the Initial Password for the support and root User](#)
Starting with release 21.7, the initial password for the support and root user can no longer be set from the Oracle Key Vault management console during the post-installation steps.

Controlling Access to SSH Servers Centrally with Oracle Key Vault

Starting with this release, you can use Oracle Key Vault to centrally manage the access to the SSH servers within your enterprise.

SSH public key authentication is often the preferred method for accessing remote hosts for operation and administrative purposes. In an enterprise setting, multiple administrators grant or revoke access to enterprise users on large number of hosts within their domain. At this scale, the decentralized nature of the access control architecture quickly becomes overwhelming, making the system generally more prone to errors and thus more exposed to security risks.

Using Oracle Key Vault integration with the OpenSSH, you can now control access to the SSH servers centrally. Centralized access control improves security, enables quicker responses to threats, reduces human error and simplifies the server access management at scale.

With release 21.7, Oracle Key Vault adds a new type of endpoint - the **SSH Server** endpoint. The SSH Server endpoint type must be deployed on the SSH server to facilitate the integration with OpenSSH.

Oracle Key Vault release 21.7 now introduces wallet types – **SSH Server** wallet and **General** wallet. The SSH Server wallet is associated with a host user on the SSH server and is meant to contain the authorized SSH public keys of the host user. You grant or deny access to the SSH servers by adding or removing the user's SSH public keys from the SSH server wallets.

Oracle Key Vault now offers SSH server authorization, SSH server access, and other reports to simplify the management and monitoring of server access.

Improved SSH User Keys Management

Oracle Key Vault release 21.7 further improves the centralized SSH user keys management.

The Oracle Key Vault PKCS#11 library integrates with the OpenSSH to support the SSH public key authentication using a SSH key pair from Oracle Key Vault.

This enables the centralized management of SSH user keys in Oracle Key Vault. You can now create SSH keys explicitly. In the previous releases, generic public-private key pairs were used as the SSH keys, making it difficult to distinguish them from other public-private key pairs not used as SSH keys.

You can now associate an SSH user with the SSH Keys, making it easier to identify and monitor the keys. The SSH user is intended to track the actual consumer of the SSH keys, a human, an application, or a machine.

Oracle Key Vault now offers SSH private key authorization and access reports to simplify the management and monitoring of SSH keys. You can separately configure the expiration alerts for the SSH keys.

RESTful Services Utility Changes to Support SSH Keys Management

Starting with release 21.7, you can use the Oracle Key Vault RESTful services utility to create and register SSH keys and manage SSH Server wallets and SSH Server endpoints.

The following Oracle Key Vault RESTful services utility commands have been updated to support the SSH key pair creation and registration of SSH private and public keys:

- `okv managed-object key-pair create`
- `okv managed-object private-key register`

- `okv managed-object public-key register`

A new option `--ssh-user` is added to these commands. Use of this option makes the underlying public and private key objects identified as the SSH keys.

To support the creation of SSH Server endpoint and SSH Server wallet, following commands have been updated:

- `okv admin endpoint create`
- `okv manage-access wallet create`

Support Key Creation from Oracle Key Vault Management Console

Starting with release 21.7, you can now create new keys and key pairs from Oracle Key Vault Management console. This allows Oracle Key Vault users to create the keys and key pairs. Previously, only endpoints could create security objects.

As an Oracle Key Vault user, you can now create generic or application specific keys and key pairs. As generic keys, Oracle Key Vault supports creation of symmetric keys and public-private key pairs.

The application keys category supports the creation of the TDE Master Encryption Key, Oracle GoldenGate Master Key, and SSH Key Pair. Once the application key is created, the corresponding application must then be configured to make use of the key from Oracle Key Vault.

At the time of key creation, you can select key algorithm and key length, set the key expiration date, and control whether the key is extractable outside of the Oracle Key Vault cluster boundary.

User created keys can be used by endpoints as usual as long as endpoints can access them.

Support for Node or Cluster Scope for Alerts in Multi-Master Cluster

Starting with Oracle Key Vault release 21.7, in a multi-master cluster, you can set the configuration for certain alerts at the Node or Cluster scope.

In earlier releases, in a multi-master cluster, the same alert configuration was always applied to each cluster node. Using the same alert setting across cluster nodes may not always be ideal. For example, if the system is configured to take backup from a particular node, it is not ideal for the other nodes to raise the **System Backup** was never done alert. For such nodes, you can set the node scope and turn off the alerts only on these nodes.

Oracle Key Vault release 21.7 makes the alert configuration even more flexible by allowing node specific configuration for following alerts:

- Fast Recovery Area Space Utilization
- High CPU Usage
- Failed System Backup
- High Memory Usage
- Disk Utilization
- System Backup

You can even enable or disable these alerts at the per node level. On a given node, the node scope configuration overrides the cluster scope configuration for the same alert.

Setting the Initial Password for the support and root User

Starting with release 21.7, the initial password for the support and root user can no longer be set from the Oracle Key Vault management console during the post-installation steps.

With release 21.7, the root user password continues to be the one that is specified during the initial phase of Oracle Key Vault installation.

The initial password for the support user can now be set by logging in to the Oracle Key Vault terminal console as the root user immediately after the install.

Changes for Oracle Key Vault Release 21.6

Oracle Key Vault release 21.6 introduces several new features.

- [Ability to Restrict the Extraction of Private Encryption Keys from Oracle Key Vault](#)
Oracle Key Vault release 21.6 allows private keys uploaded to Oracle Key Vault to be marked as non extractable, so that they do not leave the Oracle Key Vault deployment boundary.
- [Ability to Create Asymmetric Key Pairs in Oracle Key Vault](#)
Starting release 21.6, you can now create an asymmetric key pair in Oracle Key Vault.
- [Ability to Clone an Oracle Key Vault VM](#)
Starting with Oracle Key Vault release 21.6, a fresh installation of an Oracle Key Vault VM guest can be stored as a `template`, and the VM platform cloning capability can be used to clone Oracle Key Vault cluster nodes.
- [Support the Ability to Provide an Alternate Host Name or an IP Address](#)
Starting Oracle Key Vault release 21.6, you can configure Oracle Key Vault with a fully qualified domain name or IP address, such as the public IP address of systems running in cloud infrastructure environments.
- [Support SAMLv2 Based Single Sign-On \(SSO\) Authentication for Oracle Key Vault](#)
Starting with Oracle Key Vault release 21.6, Oracle Key Vault supports SAML based Single Sign-On (SSO) authentication.
- [Support for Unified Application-Level Tracing and Simplified Diagnostics Collection](#)
Starting with Oracle Key Vault release 21.6, a unified application-level tracing is introduced with the ability to centrally control the tracing level. The diagnostics download process is also simplified.
- [Aborting Oracle Audit Vault Integration with Oracle Key Vault](#)
Starting with Oracle Key Vault release 21.6, Oracle Key Vault integration with Oracle Audit Vault can be aborted, when integration is not successful.
- [Event ID Support in Auditing Records](#)
Starting with Oracle Key Vault release 21.6, all the operation events are now categorized using the event IDs.
- [Support for Disk and Network I/O and Application Metrics in Oracle Key Vault Metrics Framework](#)
Starting with Oracle Key Vault release 21.6, Oracle Key Vault expands the metrics framework to include Disk, Network I/O, and Application metrics.
- [Support for Sign and Signature Verify Operations](#)
Starting with Oracle Key Vault release 21.6, C and Java SDKs now provide Sign and Verify capability.

- [Oracle Key Vault Deployments in Microsoft Azure and Amazon AWS](#)
Starting with Oracle Key Vault release 21.6, you can deploy Oracle Key Vault on Microsoft Azure and Amazon AWS cloud platforms.
- [Endpoint IP Address Attribute Added to endpoint get RESTful Command](#)
Oracle Key Vault supports endpoint IP address in the `endpoint get` RESTful command.
- [Improved Audit Record Messages](#)
With Oracle Key Vault release 21.6, audit record messages are modified to improve for consistency and style.
- [Support Endpoint Communication With Oracle Key Vault Using a Secondary IP address or Fully-Qualified Domain Name](#)
Up until Oracle Key Vault release 21.6, the endpoints of an Oracle Key Vault (OKV) system could communicate with it only via its IP address. In particular, for those OKV instances provisioned on Oracle Cloud Infrastructure (OCI) compute instances (which have both a public and private IP), endpoint to Oracle Key Vault communication was via the Oracle Key Vault private IP only.

Ability to Restrict the Extraction of Private Encryption Keys from Oracle Key Vault

Oracle Key Vault release 21.6 allows private keys uploaded to Oracle Key Vault to be marked as non extractable, so that they do not leave the Oracle Key Vault deployment boundary.

You can now restrict private keys, as well as symmetric keys, and make them non-extractable by setting the extractable attribute value to *false*. The false attribute value ensures that the cryptographic objects remain within the Oracle Key Vault boundary.

To control whether private encryption keys can be retrieved (extracted) from Oracle Key Vault, you can use the Oracle Key Vault management console, RESTful services utility commands, the C SDK APIs, and Java SDK APIs.

Related Topics

- [Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault](#)
You can restrict symmetric or private keys from leaving Oracle Key Vault.

Ability to Create Asymmetric Key Pairs in Oracle Key Vault

Starting release 21.6, you can now create an asymmetric key pair in Oracle Key Vault.

You may create the private key as non-extractable, or make it non-extractable afterward to ensure that the private key never leaves the Oracle Key Vault deployment boundary.

You can create an asymmetric key pair using RESTful services utility commands, C and JAVA SDK APIs.

With the support of the non-extractable private keys and the sign operations on-board Oracle Key Vault, you can now implement public key authentication using openssh and Oracle Key Vault's PKCS#11 library such that user's ssh private key never leaves Oracle Key Vault. During public key authentication, the PKCS#11 library now performs the sign operation within Oracle Key Vault itself. This helps you in enforcing centralized key governance and eliminating locally downloaded, vulnerable private keys.

Ability to Clone an Oracle Key Vault VM

Starting with Oracle Key Vault release 21.6, a fresh installation of an Oracle Key Vault VM guest can be stored as a `template`, and the VM platform cloning capability can be used to clone Oracle Key Vault cluster nodes.

With Oracle Key Vault cluster, using the cloned template, the system administrator can significantly shorten the provisioning time, compared to performing a full installation of each individual cluster node.

Oracle Key Vault supports the cloning feature of the underlying virtualization platform. This eliminates the need to go through the full installation process for each individual cluster node. You can clone an Oracle Key Vault system (installed as a VM) after the installation is complete, but before performing post-installation tasks. When a clone is started up for the first time, it goes through a series of steps to regenerate system-specific configuration that makes it unique (and separate from all other clones). The (remote) cloning capability provided by virtualization platforms allows to clone from an Oracle Key Vault **Template**, which is an Oracle Key Vault installation that is stopped before this Oracle Key Vault is made unique. It regenerates all of the system-specific configuration; the clone becomes unique by completing the remaining installation steps.

Support the Ability to Provide an Alternate Host Name or an IP Address

Starting Oracle Key Vault release 21.6, you can configure Oracle Key Vault with a fully qualified domain name or IP address, such as the public IP address of systems running in cloud infrastructure environments.

Oracle Key Vault provides the ability to provide two alternate host names. You can choose whether to have endpoints use one of these alternate host names when communicating with the Oracle Key Vault server or node. The alternate host name is required to be provided for each node and you can also choose if the endpoint should use the provided host name in a multi-master cluster environment.

This feature is not supported in (deprecated) primary-standby deployments.

Support SAMLv2 Based Single Sign-On (SSO) Authentication for Oracle Key Vault

Starting with Oracle Key Vault release 21.6, Oracle Key Vault supports SAML based Single Sign-On (SSO) authentication.

Oracle Key Vault release 21.6 now supports SSO. The SSO feature is SAML based and the user is authenticated via an Identity Provider (IdP). The IdP supported Single Sign-On (SSO) Authentication provides multi-factor authentication (MFA). This provides the ability to minimize the login attempts to one set of credentials hence improving the enterprise security. Single Sign-On (SSO) Authentication is part of an identity and access management (IAM) solution, it utilizes a central directory that controls user access to resources at a more granular level. This allows organizations to comply with regulations that require provisioning users with appropriate permissions. The SSO solution also de-provisions users quickly, another common compliance requirement meant to ensure that former employees, partners, or others cannot access the sensitive data.

Related Topics

- [Configuring SAML Single Sign-On \(SSO\) Authentication](#)
SSO is an access control solution that allows users to authenticate once and get access to all enterprise resources connected to the SSO system. Oracle Key Vault SAML SSO can take advantage of the Multi-Factor-authentication supported by Identity Provider (IDP) if necessary.

Support for Unified Application-Level Tracing and Simplified Diagnostics Collection

Starting with Oracle Key Vault release 21.6, a unified application-level tracing is introduced with the ability to centrally control the tracing level. The diagnostics download process is also simplified.

Previously, to collect the diagnostics, the system administrator logs on to the Oracle Key Vault management console, downloads the readme, logs on to the Oracle Key Vault server as the `root` user, and runs commands manually to install the diagnostics utility, and enable selected log options. The system administrator then logs back on the Oracle Key Vault management console to download the diagnostics bundle.

Oracle Key Vault release 21.6 simplifies the process. Log on to the Oracle Key Vault management console as a system administrator, select the required diagnostics to download, and then click **Download**.

In addition to facilitating centralized trace generation, Oracle Key Vault release 21.6 introduces component-based tracing that allows the system administrator to adjust the trace level for specific Oracle Key Vault components from the Oracle Key Vault management console. These trace levels (in increasing levels of severity) are: MANDATORY, ERROR, WARNING, INFO, and DEBUG.

Related Topics

- [Capturing System Diagnostics](#)
To troubleshoot problems that may arise, you can generate a diagnostics package.

Aborting Oracle Audit Vault Integration with Oracle Key Vault

Starting with Oracle Key Vault release 21.6, Oracle Key Vault integration with Oracle Audit Vault can be aborted, when integration is not successful.

From the Oracle Key Vault management console you can now **Abort** the integration of AVDF. This is helpful when the integration of AVDF takes longer than usual time to get integrated with Oracle Key Vault.

Event ID Support in Auditing Records

Starting with Oracle Key Vault release 21.6, all the operation events are now categorized using the event IDs.

This feature adds a new field, **Event ID**, to Oracle Key Vault audit records. The **Event ID** represents a stable identity that is uniquely associated with an audited operation (type).

- Multiple audit records for the same audit operation use the same **Event ID**.
- Event IDs remains unchanged and remains mapped to the same operation forever as the **Event IDs** are statically baked into the Oracle Key Vault source code.

- The text description of the **Operation**, however, could still change across releases.
- New operations that are added as part of the new functionality will be given new unique Event ID.

Related Topics

- [Managing System Auditing](#)
Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.

Support for Disk and Network I/O and Application Metrics in Oracle Key Vault Metrics Framework

Starting with Oracle Key Vault release 21.6, Oracle Key Vault expands the metrics framework to include Disk, Network I/O, and Application metrics.

This feature adds metrics for Disk, Network I/O, Application metrics. The current available metrics at any time in Oracle Key Vault helps in determining the system capability and resource usage.

- **Disk I/O:** Gives insight into database cache assuming most of the Oracle Key Vault activities are going to be from the database.
- **Network I/O:** Gives insight into number of bytes received/sent during a particular time interval. You can compare the data with the historical date to analyze the usage and activity of endpoints. This also provides data as an average value.
- **Application:** Gives insight into number of KMIP connections accepted to process the connections in an interval.

Related Topics

- [Monitoring System Metrics](#)
You can use the System Metrics Monitoring feature to view and collect data for key system resource usage including CPU and Memory Usage in Oracle Key Vault.

Support for Sign and Signature Verify Operations

Starting with Oracle Key Vault release 21.6, C and Java SDKs now provide Sign and Verify capability.

You can use either RESTful services utility commands, okvutil, or C and Java SDK to perform sign and signature verify operations.

C SDK APIs

- KMIP cryptographic operations are as follows:
 - okvSign
 - okvSignVerify
- Cryptographic utility operations are as follows:
 - okvCryptoContextGetCryptoAlgo
 - okvCryptoContextGetHashingAlgo
 - okvCryptoContextGetDigitalSignAlgo
 - okvCryptoContextSetHashingAlgo

- okvCryptoContextSetCryptoAlgo
- okvCryptoContextSetDigitalSignAlgo
- okvCryptoResponseGetSignatureData
- okvCryptoResponseGetRecoveredData
- okvCryptoResponseGetValidity
- okvSignResponseCreate
- okvSignVerifyResponseCreate
- okvSignResponseFree
- okvSignVerifyResponseFree

Java SDK APIs

- KMIP cryptographic operations are as follows:
 - okvSign
 - okvSignVerify
- Cryptographic utility operations are as follows:
 - getCryptoAlgo
 - getHashingAlgo
 - getDigitalSignAlgo
 - setCryptoAlgo
 - setHashingAlgo
 - setDigitalSignAlgo
 - getSignatureData
 - getRecoveredData
 - getValidity

RESTful APIs

- okv crypto data sign
- okv crypto data sign-verify

okvutil

- okvutil sign
- okvutil sign-verify

Related Topics

- [Oracle Key Vault Client C SDK API Reference](#)
- [Oracle Key Vault Client Java SDK API Reference](#)

Oracle Key Vault Deployments in Microsoft Azure and Amazon AWS

Starting with Oracle Key Vault release 21.6, you can deploy Oracle Key Vault on Microsoft Azure and Amazon AWS cloud platforms.

In addition to on-premises data centers and Oracle Cloud Infrastructure (OCI), Oracle Key Vault release 21.6 can also be deployed in Microsoft Azure and Amazon AWS.

Endpoint IP Address Attribute Added to endpoint get RESTful Command

Oracle Key Vault supports endpoint IP address in the `endpoint get` RESTful command.

The endpoint IP address that was used at enrollment time is now recorded, and displayed with the `okv admin endpoint get --endpoint endpoint_name` command.

Improved Audit Record Messages

With Oracle Key Vault release 21.6, audit record messages are modified to improve for consistency and style.

The object information in the audit record is no longer included in the audit record messages (operation texts). Instead, object information is available under the object column.

Support Endpoint Communication With Oracle Key Vault Using a Secondary IP address or Fully-Qualified Domain Name

Up until Oracle Key Vault release 21.6, the endpoints of an Oracle Key Vault (OKV) system could communicate with it only via its IP address. In particular, for those OKV instances provisioned on Oracle Cloud Infrastructure (OCI) compute instances (which have both a public and private IP), endpoint to Oracle Key Vault communication was via the Oracle Key Vault private IP only.

Starting Oracle Key Vault release 21.6, the Oracle Key Vault system can be configured with a secondary IP address, or a fully-qualified domain name (FQDN) (both of which are referred to as **alternate hostname** in subsequent text). You can configure at most two such alternate hostnames for a given Oracle Key Vault system. Optionally, you can also choose to have endpoints communicate with the Oracle Key Vault server/node using one of these alternate hostnames. In a multi-master cluster environment, the configuration of an alternate hostname is node-specific: each node that is to have an alternate hostname must be separately configured.

Note:

This feature is not supported in primary-standby deployments (deprecated in Oracle Key Vault release 21.5).

Changes for Oracle Key Vault Release 21.5

Oracle Key Vault release 21.5 introduces several new features.

- [Support for SSH Public Key Authentication using SSH User Keys from Oracle Key Vault](#)
Starting in Oracle Key Vault release 21.5, you can use SSH key-based authentication with a key pair stored only in Oracle Key Vault.
- [Automatic Purging of Audit Records Based on a Retention Policy](#)
Starting in Oracle Key Vault release 21.5, you can now purge the old audit records automatically based on a retention policy.

- [Ability to Rotate Endpoint Certificates](#)
Starting in Oracle Key Vault release 21.5, you can rotate an endpoint in order to increase its certificate validity without incurring endpoint downtime.
- [Endpoint and Endpoint Group Privileges Support for LDAP Users](#)
Starting in Oracle Key Vault release 21.5, you can grant the endpoint and endpoint group privileges to LDAP users through the LDAP group mappings.
- [User Account Management](#)
Starting in Oracle Key Vault release 21.5, you can configure the user account profile parameters to meet your corporate user management security policies for the Oracle Key Vault users.
- [Severity based Alert Categorization](#)
Starting in Oracle Key Vault release 21.5, alerts are categorized based on their severity levels to improve ease of administration.
- [Displaying Endpoint Group Membership Column in Endpoint Metadata Report](#)
Starting with Oracle Key Vault release 21.5, additional column for Endpoint Group Membership is available in Endpoint Metadata Report.
- [Ability to Determine Time of Last Endpoint Activity](#)
Starting in Oracle Key Vault release 21.5, you can quickly determine when an endpoint was last active by checking the Endpoints page on the Oracle Key Vault Management Console.
- [UEFI Support for OCI marketplace Image](#)
Starting in Oracle Key Vault release 21.5, the Oracle Key Vault OCI marketplace images are available in UEFI mode only.
- [Separate Alerts for CA Certificate Expiration and Server/Node Certificate Expiration](#)
Starting in Oracle Key Vault release 21.5, you can configure the alerts for the CA certificate expiration and server/node certificate expiration separately.

Support for SSH Public Key Authentication using SSH User Keys from Oracle Key Vault

Starting in Oracle Key Vault release 21.5, you can use SSH key-based authentication with a key pair stored only in Oracle Key Vault.

The Oracle Key Vault PKCS#11 library supports SSH public key authentication using a SSH key pair that is uploaded to Oracle Key Vault. The centralized management of SSH user keys in Oracle Key Vault simplifies key life-cycle management, enables key governance and makes it easier to enforce policies. You can centrally perform actions such as, rotating the keys and revoking them when needed. This also allows you to minimize the risks that are associated with the SSH user keys footprint on local disks.

Related Topics

- [Managing Online and Offline Secrets](#)

Automatic Purging of Audit Records Based on a Retention Policy

Starting in Oracle Key Vault release 21.5, you can now purge the old audit records automatically based on a retention policy.

You can now better manage the disk space consumed by Oracle Key Vault audit records without the need to manually delete them once they are deemed no longer needed. You can configure Oracle Key Vault to automatically purge older audit records based on a retention

policy. For example, you can configure and apply a policy to automatically purge audit records that are older than 180 days.

Ability to Rotate Endpoint Certificates

Starting in Oracle Key Vault release 21.5, you can rotate an endpoint in order to increase its certificate validity without incurring endpoint downtime.

With Oracle Key Vault release 21.5, you can rotate an endpoint in order to increase its certificate validity without incurring endpoint downtime. Previously, this could only be done by re-enrolling the endpoint. You can choose to rotate multiple endpoints at once if required. Rotating an endpoint certificate this way is independent of the CA or server/node certificate rotation processes.

Related Topics

- [Managing Endpoints](#)

Endpoint and Endpoint Group Privileges Support for LDAP Users

Starting in Oracle Key Vault release 21.5, you can grant the endpoint and endpoint group privileges to LDAP users through the LDAP group mappings.

The privileges to the LDAP users are granted through the LDAP groups mappings. You map the endpoint or endpoint group privileges to an LDAP group. LDAP users that are members of this group are granted the mapped endpoint or endpoint group privileges at the time of login.

Related Topics

- [Managing LDAP User Authentication and Authorization in Oracle Key Vault](#)

User Account Management

Starting in Oracle Key Vault release 21.5, you can configure the user account profile parameters to meet your corporate user management security policies for the Oracle Key Vault users.

User account profile parameters govern the rules and requirements for the user passwords, and the account lockout behavior of Oracle Key Vault users. These settings apply to Oracle Key Vault users that are created locally. For LDAP users, the user account management policies are managed in the LDAP directory server.

Oracle Key Vault now also supports unlocking of a user account through a password reset. An Oracle Key Vault administrator can unlock a user account by resetting the user's password.

Related Topics

- [Managing User Accounts](#)

Severity based Alert Categorization

Starting in Oracle Key Vault release 21.5, alerts are categorized based on their severity levels to improve ease of administration.

Oracle Key Vault supports several types of alerts. Oracle Key Vault now categories these alerts to one of the severity levels: CRITICAL, HIGH, MEDIUM, and LOW. The home page of the Oracle Key Vault management console now displays the unresolved alerts in the order of their

severity. Oracle Key Vault administrators can now easily identify most critical alerts that need immediate attention to ensure operational continuity.

Related Topics

- [Configuring Alerts](#)

Displaying Endpoint Group Membership Column in Endpoint Metadata Report

Starting with Oracle Key Vault release 21.5, additional column for Endpoint Group Membership is available in Endpoint Metadata Report.

The Endpoint Metadata Report displays endpoint information and deployment configuration detail. The Metadata Report now displays the Endpoint Group Membership column.

The Endpoint Group Membership information is useful when:

- Granting privileges to Endpoint group
- Performing the Endpoint rotation

Ability to Determine Time of Last Endpoint Activity

Starting in Oracle Key Vault release 21.5, you can quickly determine when an endpoint was last active by checking the Endpoints page on the Oracle Key Vault Management Console.

Starting in Oracle Key Vault release 21.5, you can determine when an endpoint was last active from the Oracle Key Vault Management Console by navigating to the “Endpoints” page and checking the “Last Active Time” column for that endpoint. This information can be useful in quickly determining which endpoints are unused. Previously, the only way to glean this information was from the endpoint activity reports (in particular, in a multi-master cluster, by consolidating all of the endpoint activity reports from all nodes of the cluster).

Related Topics

- [Adding an Endpoint as an Oracle Key Vault System Administrator or Create Endpoint User](#)

UEFI Support for OCI marketplace Image

Starting in Oracle Key Vault release 21.5, the Oracle Key Vault OCI marketplace images are available in UEFI mode only.

The OCI marketplace images of the earlier versions of Oracle Key Vault continue to use the BIOS mode.

Separate Alerts for CA Certificate Expiration and Server/Node Certificate Expiration

Starting in Oracle Key Vault release 21.5, you can configure the alerts for the CA certificate expiration and server/node certificate expiration separately.

You can configure different threshold values for these alerts. The default threshold value for CA certificate expiration is 90 days, while that for server/node certificate expiration is 60 days. Having separate alerts makes it easier to determine when a server/node certificate rotation is to be performed. The server/node certificate rotation is short and quick process performed on a per-node basis, as opposed to a CA certificate rotation which affects the entire Oracle Key

Vault deployment and involves multiple steps. Previously, a single alert type 'Oracle Key Vault Server Certificate expiration' was raised when either the CA or the server/node certificate was expiring within the configured server certificate expiration threshold.

Changes for Oracle Key Vault Release 21.4

Oracle Key Vault release 21.4 introduces new features that affect this guide.

- [Ability to Control the Extraction of Symmetric Encryption Keys from Oracle Key Vault](#)
Starting in Oracle Key Vault release 21.4, to strengthen the protection of symmetric keys, you now can restrict these keys from leaving Oracle Key Vault.
- [Enhancements to Certificate Management](#)
Starting in Oracle Key Vault release 21.4, several enhancements to the management of certificates are available.
- [Support for Policy Based Automatic Purging of Old Oracle Key Vault Backups](#)
Starting in Oracle Key Vault release 21.4, you can manually remove the local Oracle Key Vault backup or create a policy to schedule the removal of one or more remote backups.
- [Ability to Restrict Oracle Key Vault Administrative Role Grants](#)
Starting in Oracle Key Vault release 21.4, you can control whether a grantee of an Oracle Key Vault administrative role can grant the role to other Oracle Key Vault users.
- [Client IP Address in the Oracle Key Vault Audit Trail](#)
Starting in Oracle Key Vault release 21.4, the Oracle Key Vault audit trail has one new field: `Client IP`.
- [Support for Additional Monitoring Information Through SNMP](#)
Starting in Oracle Key Vault release 21.4, additional monitoring information is available through the SNMP `nsExtendOutputFull` MIB base variable.

Ability to Control the Extraction of Symmetric Encryption Keys from Oracle Key Vault

Starting in Oracle Key Vault release 21.4, to strengthen the protection of symmetric keys, you now can restrict these keys from leaving Oracle Key Vault.

This restriction applies to the key material of the symmetric keys, but not its metadata. For example, Transparent Database Encryption (TDE) master encryption keys are stored in Oracle Key Vault. When an endpoint needs to decrypt the key, the PKCS#11 library fetches the TDE master encryption key from Oracle Key Vault to perform the decryption. If your site requires that symmetric keys never leave Oracle Key Vault, then you can configure these keys to remain within Oracle Key Vault during operations. In this case, the PKCS#11 library will send the encrypted data encryption key to Oracle Key Vault. Decryption is then performed within Oracle Key Vault and afterward, the plaintext data encryption key is returned to the PKCS#11 library. The Oracle Key Vault PKCS#11 library performs the encryption and decryption operation within Oracle Key Vault if the TDE master encryption key is restricted to leave Oracle Key Vault, or if it cannot be extracted from Oracle Key Vault.

To control whether symmetric encryption keys can be retrieved (extracted) from Oracle Key Vault, you can use the Oracle Key Vault management console, RESTful services utility commands, the C SDK APIs, and Java SDK APIs.

Related Topics

- [Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault](#)
You can restrict symmetric or private keys from leaving Oracle Key Vault.

Enhancements to Certificate Management

Starting in Oracle Key Vault release 21.4, several enhancements to the management of certificates are available.

The enhancements are as follows:

- **Support for using an Oracle Key Vault certificate authority (CA) certificate that has been signed by an external certificate signing authority:** You can choose to have the CA certificate issued by a third-party signing authority. This option can be exercised by first generating a certificate signing request (CSR), having that CSR signed by the external signing authority, and then uploading that signed CA to Oracle Key Vault. You will then be required to perform a CA certificate rotation so that all certificates on board Oracle Key Vault (endpoint certificates as well as those used for communication between Oracle Key Vault multi-master cluster nodes) are re-issued by the new CA. In previous releases, the Oracle Key Vault CA certificate was always self-signed.
- **Ability to configure a validity period of Oracle Key Vault self-signed root CA certificate:** You can configure the certificate validity period of the Oracle Key Vault self-signed CA. The new validity period would take effect the next time a CA certificate rotation is performed. Previously, this value was fixed and unchangeable.
- **In multi-master cluster environments, the ability to set the order in which endpoints are rotated during the Oracle Key Vault CA certificate rotation process:** This enhancement enables you to configure the order in which endpoints are rotated during a CA certificate rotation. Starting in this release, the endpoints are, by default, rotated in order of endpoint certificate expiry (that is, those expiring soonest are rotated first). You can also choose to order the endpoint rotation by providing a cluster subgroup priority list before initiating a CA certificate rotation. Then, during the CA certificate rotation process, endpoints that belong to cluster subgroups higher in the priority list are rotated before those in lower-priority cluster subgroups. In previous releases, when a CA certificate rotation was performed, the endpoints were rotated in random order.
- **Ability to configure a batch number of endpoints rotated during an Oracle Key Vault CA certificate rotation:** You can configure the number of endpoints that can be in the `Updating to current certificate issuer` state at a given point in the CA certificate rotation process. You can configure this value based on the number of endpoints in the Oracle Key Vault configuration. Previously, this value was static and release dependent (for example, at most, 15 endpoints could be in this state in Oracle Key Vault release 21.3).
- **Ability to rotate Oracle Key Vault server and node certificates:** Starting in this release, the certificates that are used for communication between Oracle Key Vault systems (cluster nodes in a multi-master cluster environment, or primary and standby environments), and for communication between an Oracle Key Vault system and its endpoints are now known as server certificates (in standalone or primary-standby environments) and node certificates (in multi-master cluster environments). This enhancement provides greater operational flexibility, because you now can choose different validity periods for the Oracle Key Vault CA certificate and server and node certificates. You then can rotate the server and node certificates as often as needed, without needing to go through the entire CA certificate rotation process.

Related Topics

- [Managing Service Certificates](#)
This chapter explains about Oracle Key Vault-generated certificates. You will also learn how to manage self-signed and third-party certificates.

Support for Policy Based Automatic Purging of Old Oracle Key Vault Backups

Starting in Oracle Key Vault release 21.4, you can manually remove the local Oracle Key Vault backup or create a policy to schedule the removal of one or more remote backups.

You can now better manage the disk space consumed by Oracle Key Vault backups on remote backup destination servers without the need to manually delete them once they are deemed no longer needed. You can configure Oracle Key Vault to automatically purge older backups from a remote backup destination based on a policy. For example, you can configure and apply a policy to a remote backup destination to automatically purge backups that are older than 30 days unless the backup is among the 10 more recent backups. In addition, you can now manually delete a local Oracle Key Vault backup.

Related Topics

- [Scheduling the Purging of Old Oracle Key Vault Backups](#)
To better manage disk space used by Oracle Key Vault backups on remote destinations, you can schedule the periodic purging of old backups from them.
- [Manually Deleting a Local Oracle Key Vault Backup](#)
You can manually delete a local backup by using the Oracle Key Vault management console.

Ability to Restrict Oracle Key Vault Administrative Role Grants

Starting in Oracle Key Vault release 21.4, you can control whether a grantee of an Oracle Key Vault administrative role can grant the role to other Oracle Key Vault users.

In previous releases, the Oracle Key Vault administrative roles (System Administrator, Key Administrator, and Audit Manager) could be granted to another Oracle Key Vault user by any user who currently has the role. Starting with this release, when an administrator grants the role to another user, the administrator can restrict whether the grantee user can in turn grant the role to other users. This enhancement improves overall user security and helps to adhere to good least privileges practices.

Related Topics

- [About Administrative Roles in Oracle Key Vault](#)
Oracle Key Vault provides the System Administrator, Key Administrator, and Audit Manager roles.

Client IP Address in the Oracle Key Vault Audit Trail

Starting in Oracle Key Vault release 21.4, the Oracle Key Vault audit trail has one new field: `Client IP`.

The Oracle Key Vault audit trail contains fields to capture information such as the name and type of the entity that performed an operation, the time the operation was performed, the node in which an operation was performed, and the result of the operation. The addition of the `Client IP` field enables users to better find where operations were performed, particularly in Cloud environments.

Related Topics

- [Oracle Key Vault Audit Trail](#)
The Oracle Key Vault audit trail captures information about activities that are performed in Oracle Key Vault, such as the name of an action and who performed it.

Support for Additional Monitoring Information Through SNMP

Starting in Oracle Key Vault release 21.4, additional monitoring information is available through the SNMP `nsExtendOutputFull` MIB base variable.

The `nsExtendOutputFull` MIB base variable now returns the following values:

- Oracle Audit Vault monitor status
- Oracle Audit Vault agent status
- Server or CA certificate expiration information (whichever certificate expires sooner)

Related Topics

- [SNMP Management Information Base Variables for Oracle Key Vault](#)
Oracle Key Vault provides a set of SNMP Management Information Base (MIB) variables that you can track.

Changes for Oracle Key Vault Release 21.3

Oracle Key Vault release 21.3 introduces new features that affect this guide.

- [Enhancements for the Oracle Audit Vault Integration with Oracle Key Vault](#)
Starting in Oracle Key Vault release 21.3, the integration of the Oracle Audit Vault component of Oracle Audit Vault with Oracle Key Vault has been made more secure and easier to accomplish.
- [Alert for Fast Recovery Area Space Utilization](#)
Starting in Oracle Key Vault release 21.3, an alert will be generated when the Fast Recovery Area Space utilization of the Oracle Key Vault's embedded database exceeds the configured threshold value.
- [Cluster Redo Shipping Status Alert Message Change](#)
Starting in Oracle Key Vault release 21.3, the `Cluster Redo Shipping Status` alert notification message has changed.

Enhancements for the Oracle Audit Vault Integration with Oracle Key Vault

Starting in Oracle Key Vault release 21.3, the integration of the Oracle Audit Vault component of Oracle Audit Vault with Oracle Key Vault has been made more secure and easier to accomplish.

This enhancement includes the following changes in functionality:

- **Change in System Administrator and Audit Manager roles:** Users who have the System Administrator role no longer can perform the Oracle Audit Vault integration. Instead, for better separation of duty, only a user who has been granted the Audit Manager role can perform the integration. In previous releases, only users with the System Administrator role could perform the integration. However, users who have the System Administrator role can check if the Audit Vault monitoring process is active.
- **Easier integration process:** A user with the Audit Manager role now can use the Oracle Key Vault management console to perform all the Oracle Audit Vault integration steps. In

previous releases, an Oracle Key Vault administrator had to manually perform steps such as downloading and installing the Audit Vault agent to perform this integration.

Related Topics

- [Configuring Oracle Key Vault with Oracle Audit Vault](#)
A user who has the Audit Manager role can configure Oracle Key Vault to send audit records to Oracle Audit Vault for centralized audit reporting and alerting.

Alert for Fast Recovery Area Space Utilization

Starting in Oracle Key Vault release 21.3, an alert will be generated when the Fast Recovery Area Space utilization of the Oracle Key Vault's embedded database exceeds the configured threshold value.

By default, the configured threshold value is 70% and the alert is available for standalone, multi-master cluster, and primary-standby environments. The new alert enables you to better monitor the Fast Recovery Area space usage of the Oracle Key Vault's embedded database.

Related Topics

- [About Configuring Alerts](#)
System administrators can configure alerts from the Oracle Key Vault dashboard, but all users can see alerts for the security objects to which they have access.

Cluster Redo Shipping Status Alert Message Change

Starting in Oracle Key Vault release 21.3, the `Cluster Redo Shipping Status` alert notification message has changed.

In previous releases, users were alerted only when the redo-shipping status was active (up) or inactive (down). The message now, in addition to this information, indicates whether the node in the cluster is operating in read-only mode or is no longer in read-only mode.

Related Topics

- [About Configuring Alerts](#)
System administrators can configure alerts from the Oracle Key Vault dashboard, but all users can see alerts for the security objects to which they have access.

Deprecated Features of Oracle Key Vault

Lists the features that are deprecated in Oracle Key Vault different releases.

- [Deprecated Features Oracle Key Vault 21.8](#)
Lists the features that are deprecated starting Oracle Key Vault release 21.8.
- [Deprecated Features Oracle Key Vault 21.5](#)
Lists the features that are deprecated starting Oracle Key Vault release 21.5.

Deprecated Features Oracle Key Vault 21.8

Lists the features that are deprecated starting Oracle Key Vault release 21.8.

- [Vendor Specific HSM Root of Trust Configuration](#)
Starting with Oracle Key Vault release 21.8, vendor specific integration of Thales, Entrust, and Utimaco is deprecated. Use the instructions for generic vendor integration when configuring Thales, Entrust, and Utimaco integration.

Vendor Specific HSM Root of Trust Configuration

Starting with Oracle Key Vault release 21.8, vendor specific integration of Thales, Entrust, and Utimaco is deprecated. Use the instructions for generic vendor integration when configuring Thales, Entrust, and Utimaco integration.

You can use the instructions provided by the HSM vendor to establish a Root-of-Trust for Oracle Key Vault in the HSM.

Related Topics

- [Vendor Instructions for Integrating an HSM as the Root of Trust for Oracle Key Vault](#)

Deprecated Features Oracle Key Vault 21.5

Lists the features that are deprecated starting Oracle Key Vault release 21.5.

- [SSH Tunnel](#)
SSH Tunnel feature is deprecated in Oracle Key Vault 21.5.
- [Self-enrollment for Endpoints](#)
The self-enrollment process for endpoints is deprecated in Oracle Key Vault 21.5.
- [Primary-Standby](#)
Primary-standby configuration is deprecated in Oracle Key Vault 21.5.
- [Classic RESTful Services](#)
Classic RESTful services are deprecated in Oracle Key Vault 21.5.
- [Oracle Database 11.2.0.4](#)
Wallet uploading or downloading for Oracle Database 11.2.0.4 is desupported in Oracle Key Vault 21.5.

SSH Tunnel

SSH Tunnel feature is deprecated in Oracle Key Vault 21.5.

You can use the Oracle Key Vault hybrid clusters deployment setup using Oracle Key Vault from Marketplace in OCI instead.

Self-enrollment for Endpoints

The self-enrollment process for endpoints is deprecated in Oracle Key Vault 21.5.

You can use the the RESTful services for endpoint enrollment instead.

Primary-Standby

Primary-standby configuration is deprecated in Oracle Key Vault 21.5.

You can use an Oracle Key Vault multi-master cluster deployment instead.

Classic RESTful Services

Classic RESTful services are deprecated in Oracle Key Vault 21.5.

You can use the the new RESTful services introduced with Oracle Key Vault 21 instead.

Oracle Database 11.2.0.4

Wallet uploading or downloading for Oracle Database 11.2.0.4 is desupported in Oracle Key Vault 21.5.

1

Introduction to Oracle Key Vault

Oracle Key Vault is a full-stack, security-hardened software appliance built to centralize the management of keys and security objects within the enterprise.

- [About Key and Secrets Management in Oracle Key Vault](#)
Oracle Key Vault is a fault-tolerant, highly available and scalable, secure and standards-compliant key and secrets management appliance, where you can store, manage, and share your security objects.
- [Benefits of Using Oracle Key Vault](#)
Oracle Key Vault helps you to fight security threats, centralize key storage, and centralize key lifecycle management.
- [Oracle Key Vault Use Cases](#)
The most typical use cases for Oracle Key Vault are centralized storage and management of security objects.
- [Who Should Use Oracle Key Vault](#)
Oracle Key Vault is designed for users who are responsible for deploying, maintaining, and managing security within the enterprise.
- [Major Features of Oracle Key Vault](#)
Oracle Key Vault enhances security in key management with a wide range of features that support different database deployments.
- [Oracle Key Vault Interfaces](#)
Oracle Key Vault provides both a graphical user interface and command-line interfaces.
- [Overview of an Oracle Key Vault Deployment](#)
Oracle Key Vault provides two different deployment options.

About Key and Secrets Management in Oracle Key Vault

Oracle Key Vault is a fault-tolerant, highly available and scalable, secure and standards-compliant key and secrets management appliance, where you can store, manage, and share your security objects.

A security object contains critical data that is provided by a user. Security objects that you can manage with Oracle Key Vault include encryption keys, Oracle wallets, Java keystores (JKS), Java Cryptography Extension keystores (JCEKS), and credential files. Credential files can include SSH private keys used for public key authentication to remote servers. For example, OCI compute instances. Credential files can also include database account passwords for unattended execution of regularly scheduled maintenance scripts.

To increase key and secret availability, you can install Oracle Key Vault as a multi-master cluster with up to 16 (geographically distributed) nodes.

Oracle Key Vault centralizes key and secrets management across your organization quickly and efficiently. Built on Oracle Linux, Oracle Database, Oracle Database security features like Oracle Transparent Data Encryption, Oracle Database Vault, Oracle Virtual Private Database, and Oracle GoldenGate technology, Oracle Key Vault's centralized, highly available, and scalable security solution helps to overcome the biggest key-management challenges facing

organizations today. With Oracle Key Vault you can retain, back up, and restore your security objects, prevent their accidental loss, and manage their lifecycle in a protected environment.

Oracle Key Vault is optimized for the Oracle Stack (database, middleware, systems), and Advanced Security Transparent Data Encryption (TDE). In addition, it complies with the industry standard OASIS Key Management Interoperability Protocol (KMIP).

Oracle Key Vault works with endpoints, which it treats as clients to store and manage security objects, share them with trusted peers, and retrieve them. An endpoint is a computer system such as a database server, an application server, and other information systems, where keys are used to access encrypted data and credentials are used to authenticate to other systems. For database servers hosting one or more Oracle Databases each Oracle Database will be at least one endpoint. You can use Oracle Key Vault to manage a variety of other endpoints, such as master encryption keys for encrypted Oracle GoldenGate trail files, MySQL TDE, encrypted Oracle ACFS file systems, ZFS, and many more KMIP-compliant endpoints, like MongoDB. The Java and C software development kits make it easy to integrate other endpoints as well, for example the Oracle `dbms_crypto` package.

Related Topics

- [Support for OASIS Key Management Interoperability Protocol \(KMIP\)](#)
You can use Oracle Key Vault with a range of OASIS KMIP version 1.1 profiles.

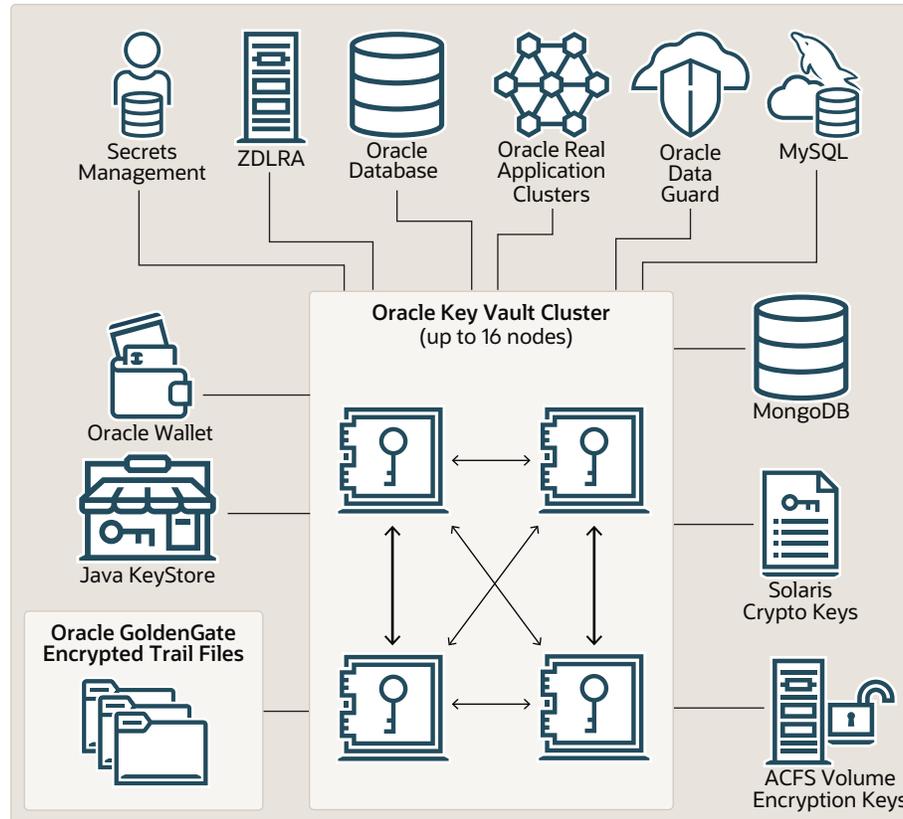
Benefits of Using Oracle Key Vault

Oracle Key Vault helps you to fight security threats, centralize key storage, and centralize key lifecycle management.

Deploying Oracle Key Vault in your organization will help you accomplish the following:

- Manage the lifecycle for endpoint security objects and keys, which includes key creation, rotation, deactivation, and removal.
- Prevent the loss of keys and wallets due to forgotten passwords or accidental deletion.
- Share keys securely between authorized endpoints across the organization.
- Enroll and provision endpoints easily using a single software package that contains all the necessary binaries, configuration files, and endpoint certificates for mutually authenticated connections (mTLS 1.2) between endpoints and Oracle Key Vault.
- Provides key management for encrypted Oracle Databases deployed as Real Application Clusters (RAC), Oracle Data Guard, Oracle Multitenant pluggable Databases, globally distributed (sharded) Databases, encrypted Oracle GoldenGate trail files, encrypted ACFS, encrypted ZFS, ZDLRA, ExaDB-C@C, ADB-C@C, and `dbms_crypto`.
- With Oracle Key Vault deployed in Oracle Cloud Infrastructure (OCI) from the Oracle Cloud Marketplace, in Amazon AWS, Microsoft Azure, or Google GCP, it provides key management for ExaDB-D, ADB-S, ADB-D in OCI as well as Amazon AWS, Microsoft Azure, and Google GCP.
- Work with other products, such as MongoDB, that support integration with external key managers via the Key Management Interoperability Protocol (KMIP).

Figure 1-1 Encrypted Data That Oracle Key Vault Can Protect



This figure illustrates how a multi-master cluster environment can be used to manage different kinds of encrypted data. It has the following components:

- **Oracle Database** refers to Oracle Databases that are connected to the Oracle Key Vault. Typically, these databases are protected with Transparent Data Encryption (TDE).
- **Oracle wallets and Java keystores** are containers for keys and sensitive objects that you upload and download between Oracle Key Vault and endpoints.
- **Secrets Management** refers to other keystore files, which are security objects like certificates, and credential files like Kerberos keytab files, SSH key files, and server password files, that you upload to Oracle Key Vault from endpoints.
- **ZDLRA, MongoDB, MySQL, Oracle GoldenGate, Solaris Crypto Keys, and Oracle ACFS** are all sources of encrypted key data that can be protected by Oracle Key Vault.

You can deploy Oracle Key Vault in the following types of environments:

- Single Oracle Database instance
- Multiple Oracle Databases on the same server
- Oracle Database using a multitenant environment
- Oracle GoldenGate
- Oracle Real Application Clusters (Oracle RAC)
- Oracle Data Guard
- Globally Distributed (sharded) Database

- Oracle Exadata, engineered systems
- Oracle Database deployed on ExaDB-C@C, ExaDB-D, ExaDB-D@Azure, ExaDB-D@AWS, ExaDB-D@AWS and, ExaDB-D@GCP
- Oracle Autonomous Database ADB-C@C, ADB-D, and ADB-S

Other components of Oracle Key Vault can include the following:

- **Oracle Key Vault Management Console** refers to the Oracle Key Vault graphical user interface, where you can log in to manage your security objects and administer the Oracle Key Vault system.
- **Oracle Key Vault Backup** refers to a backup device, where security objects in Oracle Key Vault can be backed up on-demand or on-schedule.

An Oracle Key Vault multi-master cluster provides additional benefits, such as:

- Maximum key availability by providing multiple Oracle Key Vault nodes from which data may be retrieved
- Zero endpoint downtime during Oracle Key Vault multi-master cluster maintenance

Oracle Key Vault Use Cases

The most typical use cases for Oracle Key Vault are centralized storage and management of security objects.

- [Centralized Management of TDE Master Encryption Keys Using Online Master Encryption Keys](#)
You can use an online master encryption key to centralize the management of TDE master encryption keys over a direct network connection.
- [Remote Server Access Controls and Private Key Governance for Public Key Authentication](#)
Creating and managing (non-extractable) private and public keys for public key authentication in Oracle Key Vault provides centralized remote server access controls and private key governance.
- [Centralized Storage of Oracle Wallet Files and Java Keystores](#)
You can store security objects centrally in Oracle Key Vault, and manage them with automatic mechanisms for tracking, backup, and recovery.
- [Storage of Credential Files](#)
Oracle Key Vault can back up credential files other than Oracle wallets and Java keystores for long-term retention and recovery.
- [Online Management of Endpoint Keys and Secret Data](#)
You can use the RESTful key management interface to manage the storage and retrieval of keys.

Centralized Management of TDE Master Encryption Keys Using Online Master Encryption Keys

You can use an online master encryption key to centralize the management of TDE master encryption keys over a direct network connection.

This feature applies only to Oracle databases that use Transparent Data Encryption (TDE). The term [online master encryption key](#) replaces the previous term TDE direct connection.

Online master encryption keys enables you to centrally manage Transparent Data Encryption (TDE) master encryption keys over a network connection as an alternative to using local Oracle wallet files. The connection configuration entails using a PKCS#11 library to connect to Oracle Key Vault. After you perform the configuration, all future TDE master encryption keys will be stored and managed in Oracle Key Vault. There are two scenarios that you can use:

- If the database does not yet have TDE wallets
- If the database has already been configured for TDE

The online master encryption key feature works as follows: TDE generates the master encryption key and stores it in Oracle Key Vault. Oracle Key Vault administrators have full control of the TDE master encryption keys. They can revoke access of the keys from certain endpoints, share the keys with other endpoints, and perform other operations. The online master encryption key is also a convenient alternative to copying local wallet files to multiple endpoints manually. Sharing TDE master encryption keys, rather than maintaining local wallet copies, is especially useful when Oracle Real Application Clusters (Oracle RAC), Oracle Data Guard, or Globally Distributed (sharded) databases are encrypted with TDE. The following comparison illustrates the difference:

- **Local wallet copy**
In a Data Guard scenario, re-key operations on the primary database cause the managed recovery process on the standby databases to fail. You must copy the wallet to the standby database, and then an administrator must open the wallet (if the wallet is not an auto-login wallet). Afterward, you must restart the managed recovery process.
- **Shared TDE key in a virtual wallet in Oracle Key Vault**
In a database cluster, after a key rotation operation, Oracle Key Vault immediately shares the new TDE master encryption key with other nodes in the cluster. There is no need to copy the wallet manually to the other nodes. In a Data Guard configuration, after key rotation, the new keys are immediately available to the standby databases. Sharded databases are independent databases that can have their own TDE implementation. However, all keys of all shards need to be available to all shards to enable data movement across shards. Oracle Key Vault makes key management operations seamless.

Centralized management facilitates copying encrypted data between databases using Oracle Data Pump export, import, and the transportable tablespaces features of Oracle Database when master encryption keys are stored in the wallet.

- In non-centralized management the wallet must be manually copied from source to target databases.
- In centralized management these master encryption keys are easily shared when you place them in a virtual wallet in Oracle Key Vault, and then grant each endpoint access to the virtual wallet.

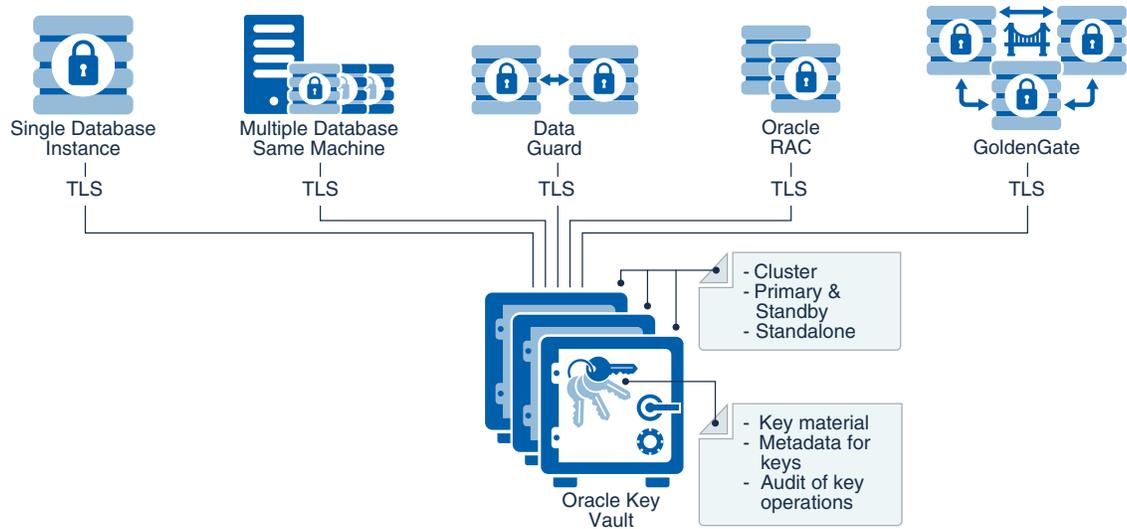
You must open the wallet before encryption and decryption. After you close the wallet, then encrypted data in tables and tablespaces is unavailable to you. You should rotate the TDE master encryption key regularly to remain in compliance with the applicable regulations.

Oracle Key Vault supports the `ADMINISTER KEY MANAGEMENT` SQL statements that are used to manage Transparent Data Encryption in an Oracle Database 12.1.0.2 and later.

Online master encryption keys managed in Oracle Key Vault are supported from Oracle Database 12.1.0.2 and later version. Online master key is deprecated for Oracle Database 11.2.0.4.

The following figure illustrates the centralized management of online master encryption keys.

Figure 1-2 Centralized Management of Online Master Encryption Keys



Remote Server Access Controls and Private Key Governance for Public Key Authentication

Creating and managing (non-extractable) private and public keys for public key authentication in Oracle Key Vault provides centralized remote server access controls and private key governance.

Public key authentication is the easiest way to give a group of administrators management access to a large fleet of remote servers, regardless of where they are deployed: Local or remote data centers, or *any* Cloud provider. Public keys are no longer stored on the remote server but centrally in a special *SSH Server Wallets* in Oracle Key Vault. This functionality allows for individual or all-encompassing access controls in case of an isolated breach attempt or a wide-spread cyber-attack.

The matching private keys are also removed from the administrator's responsibility and are made available by Oracle Key Vault for use by the administrator. The validity of the private and public key pair that is required for granting the access to a remote server is transparently handled by Oracle Key Vault. The remote server no longer has a local `authorized_keys` file and administrators no longer need to balance private key protection with simplicity for daily access to their remote servers.

The fact that private and public key pairs are stored in Oracle Key Vault also allows for complete auditing and reporting. Failed access attempts, non-compliant keys, and keys that have not been rotated for too long will no longer go undetected.

Centralized Storage of Oracle Wallet Files and Java Keystores

You can store security objects centrally in Oracle Key Vault, and manage them with automatic mechanisms for tracking, backup, and recovery.

This will help you address many operational and security challenges posed by the manual tracking and management of security objects dispersed widely across multiple servers.

Oracle Key Vault stores copies of Oracle wallet files, Java keystores, and other security objects in a centralized location for long-term retention and recovery. These security objects can later be downloaded to a new wallet or keystore file and shared with trusted server peer endpoints.

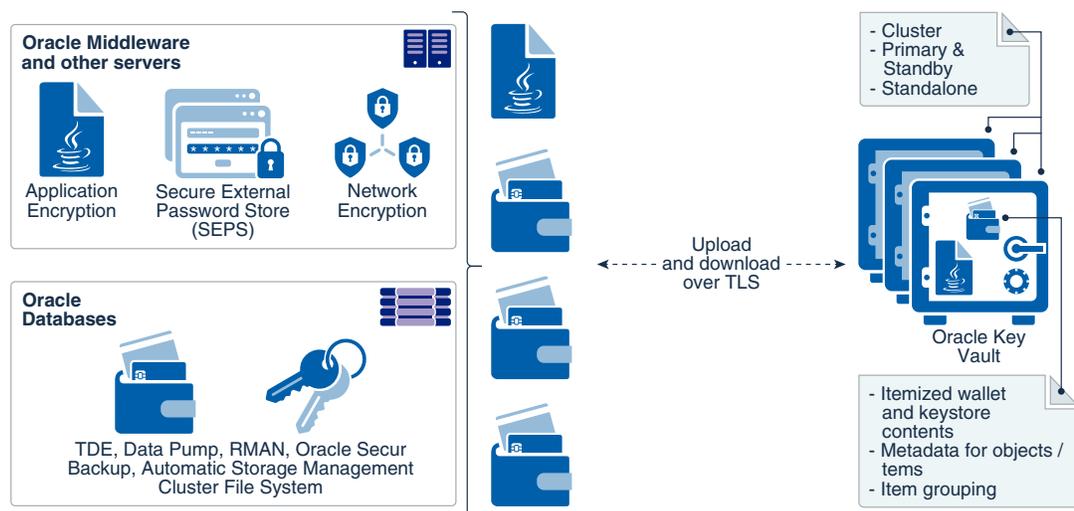
The Oracle Key Vault endpoint software can read the format of Oracle wallet files and Java keystores to store their contents at the granularity of individual security objects. You can upload both password-protected and auto-login wallets, and then download the wallet contents to a new wallet of either type. This enables users to manage security objects individually and add them to virtual wallets for sharing.

Oracle Key Vault can individually store and manage the security objects contained in:

- Oracle wallet files
 - Symmetric keys used for encryption (including TDE master encryption keys), passwords (Secure External Password Store), and X.509 certificates (network encryption).
 - Oracle Key Vault supports wallet files from all supported releases, starting with Oracle Database 12.1.0.2 to Oracle Database 23c.
- Java keystores
 - Symmetric keys, asymmetric keys such as private keys, and X.509 certificates.
 - Oracle Key Vault supports both JKS and JCEKS types of Java keystores.

The following figure illustrates the centralized storage of Oracle wallet files and Java keystores.

Figure 1-3 Centralized Storage of Oracle Wallet Files and Java Keystores



Storage of Credential Files

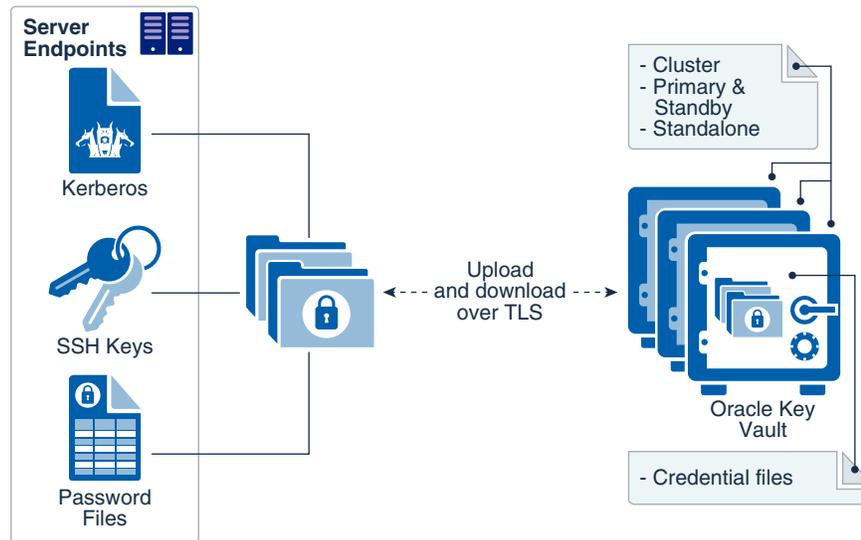
Oracle Key Vault can back up credential files other than Oracle wallets and Java keystores for long-term retention and recovery.

Oracle Key Vault does not interpret the actual content of a credential file. It simply stores the entire file as an opaque object (a file designed to prevent tools such as Oracle Key Vault from interpreting its contents) and provides a handle to the endpoint for retrieval at a later time. A credential file contains security objects such as keys, passwords, SSH keys, Kerberos keytab files, and X.509 certificates.

You can directly upload credential files to Oracle Key Vault, consolidate them in a central repository, and share them across endpoints in a trusted group. Oracle Key Vault backs up all credential files for continued and secure access at any time. Access control to credential files is managed by Oracle Key Vault endpoint administrators.

The following figure illustrates how credential files are backed up in Oracle Key Vault.

Figure 1-4 Backing Up Credential Files



Related Topics

- [Uploading and Downloading Credential Files](#)
The `okvutil upload` and `okvutil download` commands can upload and download credential files.

Online Management of Endpoint Keys and Secret Data

You can use the RESTful key management interface to manage the storage and retrieval of keys.

Applications, scripts, and third-party software can use the new interfaces to manage their keys and secrets in the Oracle Key Vault. They can generate, store, and retrieve secrets and keys at run time to and from Oracle Key Vault. All objects managed by the user or operations executed by the user using the RESTful services utility have the same security and availability attributes and the same access control as those created by other Oracle Key Vault endpoint utilities such as `okvutil`.

Who Should Use Oracle Key Vault

Oracle Key Vault is designed for users who are responsible for deploying, maintaining, and managing security within the enterprise.

These users can be database, system, or security administrators, indeed any information security personnel responsible for protecting enterprise data in database servers, application servers, operating systems, and other information systems. They manage encryption keys, Oracle wallets, Java keystores, and other security objects on a regular basis.

Other users can include personnel responsible for Oracle databases, and servers that interact with Oracle Database, because Oracle Key Vault provides inherently tighter integration with Oracle database. These systems often deploy encryption on a large scale and may have a need to simplify key and wallet management.

Major Features of Oracle Key Vault

Oracle Key Vault enhances security in key management with a wide range of features that support different database deployments.

- [Centralized Storage and Management of Security Objects](#)
You can store and manage security objects, such as TDE master encryption keys, wallets and keystores, and certificates, using Oracle Key Vault.
- [Centrally Managed Remote Server Access Controls and Improved Private Key Governance for SSH Public Key Authentication](#)
By centrally managing the private and public keys needed for public key authentication, Oracle Key Vault enables remote server access control and private key governance.
- [Management of the Key Lifecycle](#)
The management of the key lifecycle is critical for maintaining security and regulatory compliance, and consists of creation, backup, rotation, and expiration.
- [Reporting and Alerts](#)
Oracle Key Vault provides reports and alerts to track system activity in depth.
- [Separation of Duties for Oracle Key Vault Users](#)
Oracle Key Vault provides for separation of duties using three administrative roles and a set of privileges.
- [Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache feature of the endpoint software enables databases to operate when the Oracle Key Vault server is unavailable.
- [Backup and Restore Functionality for Security Objects](#)
Oracle Key Vault enables you to back up all security objects including keys, certificates, and passwords.
- [Management of Oracle Key Vault Using RESTful Services Utility](#)
You can use Oracle Key Vault RESTful services utility to automate many of the configuration, deployment, and administration tasks at scale.
- [Support for OASIS Key Management Interoperability Protocol \(KMIP\)](#)
You can use Oracle Key Vault with a range of OASIS KMIP version 1.1 profiles.
- [Database Release and Platform Support](#)
Oracle Key Vault supports following Database Release and Platform.
- [Integration with External Audit and Monitoring Services](#)
Oracle Key Vault audit records can be collected by Oracle Audit Vault, contributing to a complete picture about security-relevant events in your enterprise.
- [Integration of MySQL with Oracle Key Vault](#)
Oracle Key Vault can manage MySQL TDE encryption keys.
- [Oracle Advanced Cluster File System Encryption](#)
Oracle Key Vault supports key management for Oracle Advanced Cluster File System (Oracle ACFS) encryption.

- [Support for Cloud-Based Oracle Database Deployments](#)
An Oracle Key Vault cluster, deployed on-premises (on dedicated hardware or as a virtual machine) or in your Oracle Cloud Infrastructure (OCI) tenancy from the Oracle Cloud Marketplace, in Microsoft Azure, Amazon AWS, or Google GCP, can provide key management for ExaDB-C@C, ADB-C@C, ExaDB-D, ExaDB-D@Azure, ExaDB-D@AWS, ExaDB-D@GCP, ADB-D, and ADB-S as well as Oracle Databases deployed on-premises. An Oracle Key Vault cluster can span all on-premises and multi-cloud data centers, making encryption keys and secrets available where they are needed.
- [Oracle Key Vault Hardware Security Module Integration](#)
Oracle Key Vault can use a hardware security module (HSM) as a Root of Trust (RoT) that protects encryption keys.
- [Continuous Availability, Fault-tolerance, and High Availability through Oracle Key Vault Clustering](#)
You can use Oracle Key Vault to configure continuous availability and fault-tolerance using node clustering.

Related Topics

- [Benefits of Oracle Key Vault Multi-Master Clustering](#)
The Oracle Key Vault multi-master cluster configuration provides high available, scalable and fault-tolerant key and secrets management.

Centralized Storage and Management of Security Objects

You can store and manage security objects, such as TDE master encryption keys, wallets and keystores, and certificates, using Oracle Key Vault.

- TDE master encryption keys
For Oracle databases that use Transparent Data Encryption (TDE), Oracle Key Vault manages master encryption keys over a direct network connection using an online master encryption key as an alternative to using local wallet files. The keys stored in Oracle Key Vault can be shared across databases according to endpoint access control settings. This method of sharing keys without local wallet copies is useful when TDE is running on database clusters such as Oracle Real Application Clusters (Oracle RAC), Oracle Data Guard, or Oracle GoldenGate. You can easily migrate master encryption keys from Oracle wallets to Oracle Key Vault. Direct connections between TDE and Oracle Key Vault are supported for Oracle Database 12.1.0.2 or later.
- Oracle wallets and Java keystores
Oracle wallets and Java keystores are often widely distributed across servers and server clusters, with backup and distribution of these files performed manually. Oracle Key Vault itemizes and stores contents of these files in a master repository, yet allows server endpoints to continue operating with their local copies, while being disconnected from Oracle Key Vault. After you have archived wallets and keystores, you can recover them to their servers if their local copies are mistakenly deleted or their passwords are forgotten. Oracle Key Vault streamlines the sharing of wallets across database clusters such as Oracle RAC, Oracle Active Data Guard, and Oracle GoldenGate. Sharing wallets also facilitates the movement of encrypted data using Oracle Data Pump and the transportable tablespaces feature of Oracle Database, or when migrating (unplugging or plugging) a PDB. You can use Oracle Key Vault with Oracle wallets from all supported releases of Oracle middleware products and Oracle Database.
- Credential files
Applications store keys, passwords, and other types of sensitive information in credential files that are often widely distributed without appropriate protective mechanisms. Secure

Shell (SSH) key files and Kerberos keytabs are examples of credential files. Oracle Key Vault backs up credential files for long-term retention and recovery, audits access to them, and shares them across trusted server endpoints.

- Certificate files

X.509 certificate files (common file extensions include .pem, .cer, .crt, .der, .p12) used to authenticate and validate user identities and encrypt data on communication channels may also be stored, shared, and managed in Oracle Key Vault.

Centrally Managed Remote Server Access Controls and Improved Private Key Governance for SSH Public Key Authentication

By centrally managing the private and public keys needed for public key authentication, Oracle Key Vault enables remote server access control and private key governance.

Administrators use SSH keys to access servers and IT systems, and that use has exploded with the rise of cloud computing. Unmanaged SSH key pairs used for SSH public key authentication are a security and management challenge. Oracle Key Vault helps organizations better manage their SSH keys in two ways:

- Centralized access control – Administering users' public keys for SSH hosts in Oracle Key Vault makes provisioning and revocation of access to systems by administrators easy to manage. Administrators can provision a user's access to a remote server by uploading the user's public key into an SSH Server wallet in Oracle Key Vault. To deny access to the remote servers, SSH administrators only need to remove the user's public keys from the SSH Server wallets. Centralizing the management of SSH public keys allows administrators to track and report on access attempts.
- Improved SSH key governance – Centralizing both private and public keys in a fault-tolerant, scalable, and continuously available key management system allows for enhanced key governance. With centralized key management, organizations can enforce corporate security policies such as required key length and algorithm, periodic key rotations, and key usage reporting and auditing. Furthermore, administrators can quickly restrict all remote access in case of an ongoing security incident. Security for SSH keys can be enhanced by generating a private/public SSH key pair on-board Oracle Key Vault and by making the private key non-extractable so it cannot leave Oracle Key Vault's cluster boundary. Copying the user's public key into the SSH Server wallet in Oracle Key Vault provides the user with server access. The end-user who attempts to access a remote server can do so as long as the public key is present in the remote server's SSH Server wallet and the user has access to the matching private key in Oracle Key Vault. Managing keys in Oracle Key Vault mitigates risks associated with disk-based private keys, including key theft, unauthorized copying and sharing of keys, and key loss.

Management of the Key Lifecycle

The management of the key lifecycle is critical for maintaining security and regulatory compliance, and consists of creation, backup, rotation, and expiration.

Oracle Key Vault provides mechanisms for facilitating periodic key rotations, backup, and recovery, which ensure that you can stay in regulatory compliance, unlike other systems that create keys and passwords. You can create policies to track the key lifecycle, and configure Oracle Key Vault to report key lifecycle changes as they happen. In this manner, you will know when keys are due to expire, and can ensure that they are properly rotated and backed up.

In addition, you can restrict symmetric and private keys from leaving the Oracle Key Vault cluster boundary. This restriction applies to the key material but not to its metadata. The cryptographic operations using such keys must be performed within Oracle Key Vault.

Key lifecycle tracking is very important to maintain compliance with industry and governmental standards, such as the Payment Card Industry Data Security Standard (PCI DSS), which deal with highly sensitive data, and therefore have stringent requirements regarding the maximum lifetime of encryption keys and passwords.

Reporting and Alerts

Oracle Key Vault provides reports and alerts to track system activity in depth.

- Reports

The Oracle Key Vault audit and management reports provide detailed statistics on system, user, and endpoint activity, certificate, key and password expiry, entitlement and metadata of security objects. Audit reports capture all user and endpoint actions, the objects of the actions, and their final result.

- Alerts

You can configure the types of alerts that you want to receive. These include alerts for the expiration of keys, endpoint certificates, and user passwords, disk utilization, system backup, and the Oracle Key Vault cluster events. You can choose to send alerts to syslog to allow for external monitoring.

Separation of Duties for Oracle Key Vault Users

Oracle Key Vault provides for separation of duties using three administrative roles and a set of privileges.

The roles are System Administrator, Key Administrator, and Audit Manager. Privileges include endpoint privileges (Create Endpoint, Manage Endpoint, Create Endpoint Group, and Manage Endpoint Group) and monitoring privilege (Monitor).

Each user role possesses privileges for a type of task and may be assigned to one user (for a strict separation of duties) or combined so a single user performs multiple user roles according to the needs of the organization.

The user who is responsible for uploading and downloading security objects between Oracle Key Vault and the endpoint is referred to as the endpoint administrator. Only endpoint administrators can directly access security objects provided they have been granted access and only through installing the endpoint software. You cannot retrieve security objects using the Oracle Key Vault management console.

Related Topics

- [Administrative Roles and Privilege within Oracle Key Vault](#)

Oracle Key Vault provides separation of duty compliant administrative roles and privileges that you can combine in various ways to meet enterprise needs.

Persistent Master Encryption Key Cache

The persistent master encryption key cache feature of the endpoint software enables databases to operate when the Oracle Key Vault server is unavailable.

The TDE master encryption key is cached in the persistent master encryption key cache in addition to the in-memory cache, to make the master encryption key available across database

processes. It eliminates the need for databases to contact the Oracle Key Vault server for every new process, redo log switch, or database start-up operations.

The persistent master encryption key cache is not necessary in a multi-master cluster deployment. It is primarily used for standalone or primary-standby (deprecated) Oracle Key Vault deployments.

Related Topics

- [Using the Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache feature enables databases to be operational when the Oracle Key Vault server is unavailable.

Backup and Restore Functionality for Security Objects

Oracle Key Vault enables you to back up all security objects including keys, certificates, and passwords.

It encrypts backups for better protection of the sensitive keys and security objects and supports storing them securely at a remote destination.

This feature prevents loss of your sensitive data in the case of server failure, because you can restore a new Oracle Key Vault server to a previous state from a backup.

Oracle Key Vault can transfer backup files to any remote location that implements the Secure Copy Protocol (SCP) or SSH Secure File Transfer Protocol (SFTP).

Users with the System Administrator role can perform the following backup and restore tasks in Oracle Key Vault:

- Managing incremental and full backups
- Creating, deleting, and modifying remote backup locations
- Setting up, modifying, or disabling the current backup schedule
- Initiating an immediate one-time backup
- Scheduling a future one-time backup

Oracle Key Vault performs hot backup operation which means that the system is not interrupted while the backup is being created.

Related Topics

- [Backup and Restore Operations](#)
Backups provide the ability to restore Oracle Key Vault to a previous state in the case of a failure.

Management of Oracle Key Vault Using RESTful Services Utility

You can use Oracle Key Vault RESTful services utility to automate many of the configuration, deployment, and administration tasks at scale.

A large distributed enterprise deployment often requires automation through scripting to enable mass endpoint deployments, apply configuration changes, and perform routine management operations. The Oracle Key Vault RESTful services utility enables you perform all of these tasks in a way that facilitates faster deployment with less human intervention. You can use Oracle Key Vault RESTful services to automate the management of endpoints, wallets, access control, deployment operations, and backup operations at scale.

The Oracle Key Vault RESTful services utility also enables the automation of most key management functions at scale by providing a simplified interface to Key Management Interoperability Protocol (KMIP) operations. The RESTful services utility allows operations on managed objects such as keys, certificates, and other objects in a simple manner without requiring any client side development.

Related Topics

- [Oracle Key Vault RESTful Services Administrator's Guide](#)

Support for OASIS Key Management Interoperability Protocol (KMIP)

You can use Oracle Key Vault with a range of OASIS KMIP version 1.1 profiles.

OASIS Key Management Interoperability Protocol (KMIP) standardizes key management operations between key management servers and endpoints provided by different vendors.

Oracle Key Vault implements the following OASIS KMIP version 1.1 profiles:

- **Basic Discover versions Server Profile:** Provides the server version to endpoints.
- **Basic Baseline Server KMIP Profile:** Provides core functionality to retrieve objects from the server.
- **Basic Secret Data Server KMIP Profile:** Provides endpoints the ability to create, store, and retrieve secret data (typically passwords) on the server.
- **Basic Symmetric Key Store and Server KMIP Profile:** Provides endpoints the ability to store and retrieve symmetric encryption keys on the server.
- **Basic Symmetric Key Foundry and Server KMIP Profile:** Provides endpoints the ability to create new symmetric encryption keys on the server.

Note:

Oracle Key Vault KMIP Server now uses KMIP protocol version 1.1 as its preferred version. In earlier releases of Oracle Key Vault, even though the KMIP server accepted and processed client requests with KMIP version 1.1, it always sent the server response with the KMIP version 1.0. Now, the KMIP server sends a response with the protocol version with which the KMIP request was made. The KMIP server is also enhanced to return an error for client requests that are made with unsupported KMIP version. Such error responses are returned using the server's preferred KMIP version which is currently set to 1.1.

Related Topics

- [Key Management Interoperability Protocol Specification Version 1.1](#)

Database Release and Platform Support

Oracle Key Vault supports following Database Release and Platform.

Oracle Key Vault supports Oracle Database releases from 12.1.0.2 to 23c on Oracle and RedHat Linux, Linux for Arm (aarch64), IBM: Linux on System z, Solaris (SPARC and x86), SuSE Linux Enterprise Server, AIX, HP-UX (IA) and Windows Server.

Related Topics

-

Integration with External Audit and Monitoring Services

Oracle Key Vault audit records can be collected by Oracle Audit Vault, contributing to a complete picture about security-relevant events in your enterprise.

SNMP and RESTful monitoring commands can monitor Oracle Key Vault cluster health and maintain its availability. Oracle Key Vault audit data and alerts, system activity, and information about Oracle Key Vault cluster operations can be forwarded to a SYSLOG server.

Integration of MySQL with Oracle Key Vault

Oracle Key Vault can manage MySQL TDE encryption keys.



Note:

MySQL Windows databases are not supported.

Related Topics

- [MySQL Integration with Oracle Key Vault](#)
You can manage TDE encryption keys in MySQL with Oracle Key Vault.

Oracle Advanced Cluster File System Encryption

Oracle Key Vault supports key management for Oracle Advanced Cluster File System (Oracle ACFS) encryption.



Note:

Starting with Oracle Database 21c, the name of Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is changed to Oracle Advanced Cluster File System (Oracle ACFS).

This change is only a change of the name. The basic function of Oracle's cluster file system continues to be the same. Oracle continues to develop and enhance Oracle ACFS.

Support for Cloud-Based Oracle Database Deployments

An Oracle Key Vault cluster, deployed on-premises (on dedicated hardware or as a virtual machine) or in your Oracle Cloud Infrastructure (OCI) tenancy from the Oracle Cloud Marketplace, in Microsoft Azure, Amazon AWS, or Google GCP, can provide key management for ExaDB-C@C, ADB-C@C, ExaDB-D, ExaDB-D@Azure, ExaDB-D@AWS, ExaDB-D@GCP, ADB-D, and ADB-S as well as Oracle Databases deployed on-premises. An Oracle Key Vault cluster can span all on-premises and multi-cloud data centers, making encryption keys and secrets available where they are needed.

Related Topics

- [Oracle Database Instances in Oracle Cloud Infrastructure](#)
Oracle Key Vault deployed on-premises can manage the TDE master encryption keys for Oracle Database instances running in Oracle Cloud Infrastructure (OCI).

Oracle Key Vault Hardware Security Module Integration

Oracle Key Vault can use a hardware security module (HSM) as a Root of Trust (RoT) that protects encryption keys.

HSMs are built with specialized tamper-resistant hardware which is harder to access than normal servers. This protects the RoT and makes it difficult to extract sensitive key material, lowering the risk of compromise. In addition, you can use HSMs in FIPS 140-2 Level 3 mode which can help meet certain compliance requirements.

Related Topics

- [Oracle Key Vault Root of Trust HSM Configuration Guide](#)

Continuous Availability, Fault-tolerance, and High Availability through Oracle Key Vault Clustering

You can use Oracle Key Vault to configure continuous availability and fault-tolerance using node clustering.

Oracle Key Vault can be installed in a cluster with up to 16 nodes. Clusters consist of up to eight read/write pairs. These read/write pairs are updated synchronously; when one Oracle Key Vault node creates a key for a TDE-enabled database, the transaction completes only after the key has been replicated to at least one other node. Clustering also allows rolling upgrades to Oracle Key Vault cluster nodes without interruptions to your application availability, even if Oracle Key Vault is operated with non-extractable TDE keys.

Oracle Key Vault Interfaces

Oracle Key Vault provides both a graphical user interface and command-line interfaces.

- [Oracle Key Vault Management Console](#)
The Oracle Key Vault management console is a browser-based graphical user interface that Key Vault administrators use to perform day-to-day tasks.
- [Oracle Key Vault okvutil Endpoint Utility](#)
Endpoint administrators can use the `okvutil` command-line utility to upload and download security objects between Oracle Key Vault and endpoints.
- [Oracle Key Vault RESTful Services](#)
You can use the Oracle Key Vault RESTful Services utility to automate processes for a large distributed enterprise deployment.
- [Oracle Key Vault Client SDK](#)

Oracle Key Vault Management Console

The Oracle Key Vault management console is a browser-based graphical user interface that Key Vault administrators use to perform day-to-day tasks.

It enables Oracle Key Vault administrators to manage keys and sensitive objects, wallets, endpoints, and users. The console can also configure settings for individual Oracle Key Vault servers, as well as multi-master clusters, backup, and recovery.

Related Topics

-

Oracle Key Vault okvutil Endpoint Utility

Endpoint administrators can use the `okvutil` command-line utility to upload and download security objects between Oracle Key Vault and endpoints.

The `okvutil` utility communicates with Oracle Key Vault over a mutually authenticated secure connection.

Related Topics

- [Oracle Key Vault okvutil Endpoint Utility Reference](#)
The `okvutil` utility enables you to perform tasks uploading and downloading security objects.

Oracle Key Vault RESTful Services

You can use the Oracle Key Vault RESTful Services utility to automate processes for a large distributed enterprise deployment.

This utility enables you to automate the management of endpoints, wallets, security objects, deployment operations, and backup operations using RESTful services that are JSON compliant.

Related Topics

- *Oracle Key Vault RESTful Services Administrator's Guide*

Oracle Key Vault Client SDK

Various Oracle and non-Oracle products can use the Oracle Key Vault client SDK to integrate directly with Oracle Key Vault.

The client SDK is available in both Java and C. It has a comprehensive set of high-level and low-level APIs and sample programs.

Related Topics

- *Oracle Key Vault Developer's Guide*

Overview of an Oracle Key Vault Deployment

Oracle Key Vault provides two different deployment options.

- A multi-master cluster configuration allows for up to 16 nodes for scalability and high availability. This is the recommended deployment. Usually, read/write pairs follow the deployment of Oracle Data Guard primary and standby databases, potentially stretching across geographically distributed data centers, or even stretching from on-premises into an Oracle Cloud Infrastructure (OCI), Microsoft Azure, Amazon AWS tenancy, or Google GCP tenancy.

- A standalone deployment is simplest to deploy. However, it does not provide continuous availability of the key service in the event an Oracle Key Vault server becomes unavailable.

You can use the following steps as a guideline to deploying Oracle Key Vault within your organization:

1. Understand important concepts described in [Oracle Key Vault Concepts](#) and [Oracle Key Vault Multi-Master Cluster Concepts](#).
2. Install and configure Oracle Key Vault as described in *Oracle Key Vault Installation and Upgrade Guide*.
3. Create a multi-master cluster by adding up to 16 Oracle Key Vault servers for maximum redundancy and reliability. This is described in [Configuring System Settings for an Entire Oracle Key Vault Multi-Master Cluster](#).

You must have a separate license for each Oracle Key Vault server installation in a multi-master cluster environment.

4. Create users to manage the day-to-day tasks for Oracle Key Vault as described in [Managing Oracle Key Vault Users](#).
5. Register endpoints so that they can use Oracle Key Vault to store and manage their security objects described in [Managing Oracle Key Vault Endpoints](#).
6. Register endpoints in the cloud described in [Oracle Database Instances in Oracle Cloud Infrastructure](#).
7. Enroll endpoints so that you can upload or download security objects between the endpoints and Oracle Key Vault described in [Enrolling and Upgrading Endpoints for Oracle Key Vault](#).
8. Upload or add virtual wallets to Oracle Key Vault described in [Managing Oracle Key Vault Master Encryption Keys](#).
9. Use automating endpoint enrollment and provisioning for large-scale deployments in *Oracle Key Vault RESTful Services Administrator's Guide*.
10. Read about using Oracle Key Vault with other features, such as Oracle GoldenGate, in [Managing Keys for Oracle Products](#).
11. Automate key management to perform online key management with other software using RESTful services utility and client SDK, as described in *Oracle Key Vault RESTful Services Administrator's Guide*.
12. Learn how to perform periodic maintenance tasks such as administering and monitoring the system, as described in [Oracle Key Vault General System Administration](#).
13. Learn how to monitor Oracle Key Vault by performing tasks such as creating alerts, as described in [Monitoring and Auditing Oracle Key Vault](#).

2

Oracle Key Vault Concepts

To successfully deploy Oracle Key Vault, you must understand the deployment architecture, use cases, access control, administrative roles, and endpoints.

- [Overview of Oracle Key Vault Concepts](#)
Endpoints are computer systems such as database and application servers, and other information systems where keys and credentials access data.
- [Oracle Key Vault Deployment Architecture](#)
Oracle Key Vault is packaged as a software appliance preconfigured with an operating system, a database, and the Oracle Key Vault application.
- [Access Control Configuration](#)
Oracle Key Vault enables you to control access to security objects at various access levels.
- [Administrative Roles and Privilege within Oracle Key Vault](#)
Oracle Key Vault provides separation of duty compliant administrative roles and privileges that you can combine in various ways to meet enterprise needs.
- [Naming Guidelines for Objects](#)
The naming guidelines affect the following Oracle Key Vault objects: users, user groups, endpoints, endpoint groups, and virtual wallets.
- [Emergency System Recovery Process](#)
During installation, you will be required to create a special recovery passphrase that Oracle Key Vault uses to recover from emergency situations.
- [Root and Support User Accounts](#)
Both the `root` and `support` user accounts are used with the command-line interface.
- [Endpoint Managers](#)
An endpoint manager owns and manages endpoints, which are entities such as Oracle Databases that use Oracle Key Vault.
- [Endpoint Administrators](#)
An endpoint administrator is an Oracle Key Vault user who is responsible for registering, enrolling, and managing one or more Oracle Key Vault endpoints.
- [FIPS Mode](#)
FIPS mode enables Oracle Key Vault to adhere to FIPS 140-2 compliance.

Overview of Oracle Key Vault Concepts

Endpoints are computer systems such as database and application servers, and other information systems where keys and credentials access data.

These [endpoint](#) systems must store and manage their encryption keys and secrets efficiently, so that data is secure, accessible, and available to meet the day-to-day activities of the enterprise. Endpoints with preexisting keys, or the capability to generate them, can use Oracle Key Vault as secure, external, long-term storage.

You must register and enroll an endpoint so that it can communicate with Oracle Key Vault. Enrolled endpoints can upload their keys, share them with other endpoints, and download them to access their data. Oracle Key Vault keeps track of all enrolled endpoints and audits their actions.

You can group [security objects](#) such as master encryption keys and credential files into a [virtual wallet](#) in Oracle Key Vault. The main purpose of a virtual wallet is to group related security objects so that they can be collectively shared with peers in an easy way. A privileged user can create a virtual wallet, add keys to the empty wallet, and then grant other users, endpoints, [user groups](#), and [endpoint groups](#) various levels of access to the wallet. You can grant these users access to the virtual wallet appropriate to their function within the organization, thus limiting access to security objects to just those users who need it. A user must have access to security objects before being able to grant access on those same security objects to other users. The access level they grant can be equal to or less than their own. This flexibility is designed to meet the multiple and varying needs of any organization.

The owner of a security object is the entity that created the security object with full read, write, and modify access to the security object. The owner can add the security object to any number of wallets to be shared with other users at various access levels.

When an endpoint is registered with Oracle Key Vault, you can specify a default wallet for the endpoint. The default wallet ensures that endpoints are associated with a virtual wallet where the keys will be uploaded if no virtual wallet is specified at the time of wallet or key upload.

Multiple endpoints can have a common default wallet. The contents of this default wallet are shared across all the endpoints that have access to the shared default wallet, without the need to put these endpoints into an endpoint group.

Oracle Key Vault automatically audits all actions performed by users and endpoints.

Oracle Key Vault Deployment Architecture

Oracle Key Vault is packaged as a software appliance preconfigured with an operating system, a database, and the Oracle Key Vault application.

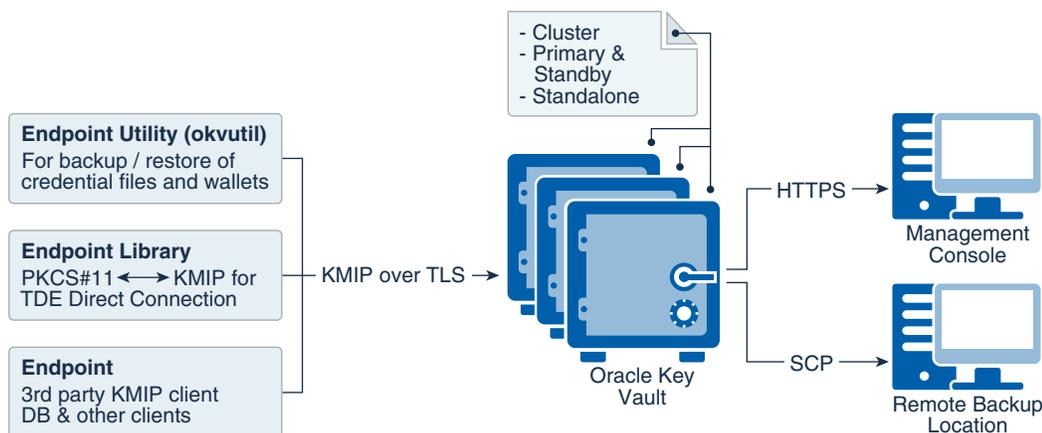
This way, you do not have to install and configure individual components. It is hardened for security according to operating system and database hardening best practices. The installation process does not include any unnecessary packages and software, and it enables only required ports and services.

The [endpoints](#) communicate with Oracle Key Vault over a mutually authenticated Transport Layer Security (mTLS) connection using the OASIS Key Management Interoperability Protocol (KMIP).

The [Oracle Key Vault multi-master cluster](#) configuration can contain up to 16 nodes, two of which must be [read/write nodes](#) with the remaining being a combination of either [read/write pairs](#) or [read-only nodes](#). The multi-master configuration provides several benefits over a primary-standby configuration. The multi-master configuration and the primary-standby configuration are mutually exclusive.

The deprecated Oracle Key Vault primary-standby configuration defined one primary server and one standby server. The primary server is active and services requests from endpoints. If the primary fails to communicate with the standby for a time exceeding a configured time threshold, then the standby server takes over as primary. Communication related to data replication between the primary and standby servers is a mutually authenticated TLS connection. This was referred to as the primary-standby option (previously called high availability) in previous releases of Oracle Key Vault.

The following figure illustrates the deployment architecture of Oracle Key Vault.

Figure 2-1 Oracle Key Vault Deployment Architecture

For multiple geographically distributed data centers with high load and extreme availability requirements, you should deploy Oracle Key Vault in a multi-master cluster configuration. The read/write pairs should span data centers. For single data centers where data does not leave the data center, consider using a 2-node cluster deployment (one read/write pair) of Oracle Key Vault, instead of a primary-standby deployment. It provides better resource utilization (no idle standby server) and is the ideal platform for a growing Oracle Key Vault cluster in case the demand increases over time. A standalone deployment of Oracle Key Vault is useful for testing and development environments.

**Note:**

Oracle Key Vault is packaged as a hardened software appliance. It is strongly discouraged to install any third party software on Oracle Key Vault. Changes to Oracle Key Vault are not supported, may interfere with upgrades, break the appliance and make it unusable.

Related Topics

- [Oracle Key Vault Multi-Master Cluster Overview](#)
The multi-master cluster nodes provide high availability, disaster recovery, load distribution, and geographic distribution to an Oracle Key Vault environment.

Access Control Configuration

Oracle Key Vault enables you to control access to security objects at various access levels.

- [About Access Control Configuration](#)
You can grant users access to security objects in Oracle Key Vault at a level appropriate to their function in the organization.
- [Access Grants](#)
You can grant access to virtual wallets directly or indirectly.
- [Access Control Options](#)
Access control options enable you to set the type of privileges that users have to read, write, and delete security objects.

About Access Control Configuration

You can grant users access to security objects in Oracle Key Vault at a level appropriate to their function in the organization.

You can set access control on [security objects](#) individually, or collectively when you group them into a [virtual wallet](#). Oracle Key Vault uses a virtual wallet to share a set of security objects with others. You can set access levels on a virtual wallet for an [endpoint](#) or [user](#), thus granting simultaneous access to all the security objects contained within the virtual wallet.

In addition to being able to grant access to users or endpoints individually, you can collectively grant access by using [user groups](#) or [endpoint groups](#). If multiple endpoints need access to a virtual wallet, it is simpler to add these endpoints to an endpoint group, and grant the endpoint group access to the virtual wallet. The alternative is to grant access to each endpoint individually. When you grant an endpoint group access to a virtual wallet, you are granting access to all the member endpoints in the endpoint group .

Access Grants

You can grant access to virtual wallets directly or indirectly.

- Grant users and endpoints access directly.
- Grant users and endpoints groups access indirectly through a group membership. When you grant a user group or endpoint group access, you are granting all members of the group access. This is a convenient alternative to individually granting each user or endpoint access.

From the Oracle Key Vault management console, you can grant access mappings on a virtual wallet in the following two ways:

- From the **user**, **endpoint**, or their respective groups. You can start at the user, endpoint, or respective group and add the wallet and access mappings for this user.
- From the **virtual wallet**. You can start from the virtual wallet and add users, endpoints, and their respective groups that can access it at access mappings that you set.

Related Topics

- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.
- [Granting an Endpoint Access to a Virtual Wallet](#)
An endpoint must have the Read, Modify, and Manage Wallet privileges on the wallet before security objects can be uploaded or downloaded.
- [Granting a User Group Access to a Virtual Wallet](#)
You can modify the access level to a virtual wallet for a user group as functional needs change.

Access Control Options

Access control options enable you to set the type of privileges that users have to read, write, and delete security objects.

You can control access to virtual wallets by setting different access levels for users, user groups, endpoints, and endpoint groups corresponding to their role and function in the organization.

There are three access levels:

- **Read Only** grants read privileges on the security object.
- **Read and Modify** grants read and modify privileges on the security object.
- **Manage Wallet** grants the following privileges:
 - Adding or removing security objects from the virtual wallet. The user must have **Read and Modify** access on the security object to be added to the virtual wallet.
 - Granting others access to the wallet
 - Modifying wallet settings, such as its description
 - Deleting the wallet

Administrative Roles and Privilege within Oracle Key Vault

Oracle Key Vault provides separation of duty compliant administrative roles and privileges that you can combine in various ways to meet enterprise needs.

- [Separation of Duties in Oracle Key Vault](#)
When you grant the Oracle Key Vault roles and system privileges to users, ensure that you adhere to separation of duty guidelines.
- [Administrative Roles](#)
The Oracle Key Vault administrative roles are System Administrator, Key Administrator, and Audit Manager.
- [Endpoint Privileges](#)
Oracle Key Vault provides privileges for creating and managing endpoints and endpoint groups.
- [Monitor Privilege](#)
Oracle Key Vault provides a privilege for creating and managing endpoints and endpoint groups.

Separation of Duties in Oracle Key Vault

When you grant the Oracle Key Vault roles and system privileges to users, ensure that you adhere to separation of duty guidelines.

Oracle Key Vault users can be assigned the administrative roles by function, so there is a clear separation of duties between the System Administrator role, the Key Administrator Role, and the Audit Manager role. The Create Endpoint, Manage Endpoint, Create Endpoint Group, and Manage Endpoint Group privileges enable users to manage a subset of endpoints or endpoint groups without requiring these users to be granted more powerful administrative roles. The Monitor privilege enables users to view system information and metrics in read-only mode, both through the Oracle Key Vault management console and the RESTful services utility. You also can create users that have no administrative privileges.

In a strict separation of duties environment, different users are responsible for different functions. For example, for endpoints, the operations to manage an endpoint and grant permissions to the endpoint must be done by different users. Only users with System Administrator role or Manage Endpoint privilege can enroll an endpoint and only users with Key Administrator role or Manage Endpoint Group privilege can add an endpoint to the endpoint group.

You can achieve a separation of duties in two ways:

- For each person who has been granted a role or a privilege, grant them the appropriate privileges for the functional area that they manage. For example, grant the System Administrator role for system-related tasks such as user, backup/recovery, and endpoint management; the Key Administrator role to manage encryption keys, wallets, and endpoint groups (within the Oracle Key Vault interface); and the Audit Manager role to manage auditing-related tasks.
- Grant a user access to one object or function independently of all others using a fine-grained division of access control. For example, grant the Manage Endpoint privilege and Manage Endpoint Group privileges to different users based on their area of responsibility. These users do not need to have any of the administrative roles to perform their function.

You should ensure that every user who interacts with Oracle Key Vault has their own unique user account and password. Because the System Administrator and Key Administrator roles are powerful, also ensure that these users are trustworthy and that they are knowledgeable about the areas that they manage.

Administrative Roles

The Oracle Key Vault administrative roles are System Administrator, Key Administrator, and Audit Manager.

- [About Administrative Roles in Oracle Key Vault](#)
Oracle Key Vault provides the System Administrator, Key Administrator, and Audit Manager roles.
- [System Administrator Role Duties](#)
The Oracle Key Vault System Administrator is responsible for the following general system-related tasks.
- [Key Administrator Role Duties](#)
The Oracle Key Vault Key Administrator is responsible for managing the following security objects:
- [Audit Manager Role Duties](#)
The Oracle Key Vault Audit Manager is responsible for the following audit-related tasks.

About Administrative Roles in Oracle Key Vault

Oracle Key Vault provides the System Administrator, Key Administrator, and Audit Manager roles.

- **System Administrator role** provides privileges for creating and managing users, creating and managing endpoints, configuring system settings and alerts, and generally administering Oracle Key Vault. This is the most powerful role.
- **Key Administrator role** provides privileges for managing the key life cycle and controlling access to all security objects in Oracle Key Vault.
- **Audit Manager role** provides privileges for managing the audit life cycle and audit policies.

The System Administrator, Key Administrator, and Audit Manager roles are designed to be flexible to support various organizational needs and structures. By default, users who have these roles may not grant their roles to other users. Users may grant roles to other users only if they are granted the role with the **Allow Forward Grant** option. If one administrative user is performing two administrative functions, then that user will have two roles in Oracle Key Vault. If the user has received the roles with the **Allow Forward Grant** option, they may grant or revoke one or both roles to other users.

You can enforce separation of administrator roles by enabling the **Enforce Separation of Administrator Roles** check from the **Account Management** tab of the **System Recovery** page. This prevents the grant of more than one Oracle Key Vault administrative role to any Oracle Key Vault user. The restriction holds even if the grantor has the **Allow Forward Grant** option. When the **Enforce Separation of Administrator Roles** check is enabled, an Oracle Key Vault user can have at most one administrative role, enforcing administrative role isolation. It is possible that an existing Oracle Key Vault user may have been granted multiple administrative roles before the **Enforce Separation of Administrator Roles** check is enabled. For example, post upgrade you may not have removed multiple administrative roles from all the users. Such users who hold multiple administrative roles cannot exercise any of those roles after the **Enforce Separation of Administrator Roles** check is enabled. The additional roles have to be removed before the remaining one intended administrative role can become operational.

**Note:**

Users can continue to be granted endpoint and endpoint group privileges when the **Enforce Separation of Administrator Roles** option is enabled.

One of the postinstallation tasks is to create three administrative users for the three administrative roles. The installation process also prompts you to create a recovery passphrase. In a situation where there is no administrative user present, you use the recovery passphrase to repeat the post-installation configuration, and create three administrative users to ensure continued operation and management of Oracle Key Vault.

When you use the management console interface, your access to the various tabs, menus, and actions depends on your role and the objects that you have access to.

System Administrator Role Duties

The Oracle Key Vault System Administrator is responsible for the following general system-related tasks.

- Creating and managing users
- Adding and managing endpoints (you can also grant privileges to individual regular users to handle endpoints)
- Configuring alerts and key rotation reminders
- Scheduling backups
- Starting and stopping Oracle Key Vault
- Configuring SMTP server settings for email notification
- Enabling or disabling FIPS mode
- Configuring Oracle Key Vault to use a hardware security module
- Configuring SNMP for remote monitoring
- Enabling automated endpoint enrollment and key management through RESTful Services Utility
- Checking if the Oracle Audit Vault monitoring process is running
- Creating SSH tunnels for Oracle Cloud Database as a Service endpoints
- Setting up a cluster

- Managing and monitoring a cluster
- Managing the cluster configuration
- Granting and revoking the System Administrator role to and from other users
- Granting and revoking the Create Endpoint and Manage Endpoint privileges to and from other users
- Granting and revoking the Monitor privilege to and from other users

Key Administrator Role Duties

The Oracle Key Vault Key Administrator is responsible for managing the following security objects:

- Managing the lifecycle of security objects
- Having full access on all virtual wallets and security objects
- Adding and managing endpoint groups (you also can grant privileges to individual regular users to handle endpoint groups)
- Controlling access to virtual wallets for users, endpoints, user groups, and endpoint groups
- Granting and revoking the Key Administrator role to and from other users
- Granting and revoking the Create Endpoint Group and Manage Endpoint Group privileges to and from other users
- Managing the extractable attribute settings for private and symmetric keys to allow or disallow them from leaving the Oracle Key Vault cluster boundary

Audit Manager Role Duties

The Oracle Key Vault Audit Manager is responsible for the following audit-related tasks.

- Managing the audit trail as the only user who has privileges to export or delete Oracle Key Vault audit records
- Having read access on all security objects
- Managing audit settings
- Granting the Audit Manager role to other users
- Managing the Oracle Audit Vault integration with Oracle Key Vault

Endpoint Privileges

Oracle Key Vault provides privileges for creating and managing endpoints and endpoint groups.

- [About Endpoint Privileges in Oracle Key Vault](#)
Oracle Key Vault provides fine-grained endpoint-management privileges that enable users who do not have the System Administrator and Key Administrator roles to create and manage endpoints and endpoint groups.
- [Create Endpoint Privilege Duties and Scope](#)
The Create Endpoint system privilege enables a user to create an endpoint.

- [Manage Endpoint Privilege Duties and Scope](#)
The Manage Endpoint object privilege on an endpoint enables a user to perform all endpoint management operations on the endpoint.
- [Create Endpoint Group Privilege Duties and Scope](#)
The Create Endpoint Group system privilege enables a user to create an endpoint group.
- [Manage Endpoint Group Privilege Duties and Scope](#)
The Manage Endpoint Group object privilege on an endpoint group enables a user to perform all endpoint group management operations on the endpoint group.

About Endpoint Privileges in Oracle Key Vault

Oracle Key Vault provides fine-grained endpoint-management privileges that enable users who do not have the System Administrator and Key Administrator roles to create and manage endpoints and endpoint groups.

These privileges are as follows:

- **Create Endpoint system privilege** authorizes a user to create endpoints.
- **Manage Endpoint object privilege** authorizes a user to perform all endpoint management operations on the endpoint.
- **Create Endpoint Group system privilege** authorizes a user to create endpoint groups.
- **Manage Endpoint Group object privilege** authorizes a user to perform all endpoint group management operations on the endpoint group.

Only users who have the System Administrator or Key Administrator role can grant and revoke the endpoint privileges to other users, as follows:

- **Create Endpoint and Manage Endpoint privileges:** System Administrator role
- **Create Endpoint Group and Manage Endpoint Group privileges:** Key Administrator role

The System Administrator and Key Administrator roles are powerful roles that enable users with these roles to modify or delete any objects in Oracle Key Vault that belong to other users. The endpoint and endpoint group privileges enable you to allow regular users who do not have these roles to create and manage a specific set of endpoints and endpoint groups. You can grant management privileges to a user for any endpoint or endpoint group regardless of whether the user created the endpoint or endpoint group. These users cannot perform operations on other endpoints or endpoint groups that they are not authorized to manage. These privileges help implement isolation among different sets of endpoints and endpoint groups that are managed by different users. For example, one set of users can be responsible for managing all cloud database endpoints, and other users can be responsible for managing all on-premises database endpoints.

These privileges are enforced when endpoint or endpoint group operations are run from either the Oracle Key Vault management console or using Oracle Key Vault RESTful services utility command-line interface.

Create Endpoint Privilege Duties and Scope

The Create Endpoint system privilege enables a user to create an endpoint.

When a user who is granted the Create Endpoint privilege creates an endpoint, Oracle Key Vault automatically grants the Manage Endpoint privilege to the user on that endpoint. This allows the user to manage the endpoints that they created without requiring additional privilege

grants. However, the user's Manage Endpoint privilege on endpoints that they created can be revoked later.

Who Can or Cannot Grant or Revoke the Create Endpoint Privilege?

Only a user who has the System Administrator role can grant or revoke the Create Endpoint privilege. Users who have the Create Endpoint privilege cannot grant or revoke this privilege to or from any user.

How the Create Endpoint Privilege Works with the Manage Endpoint Privilege

Even though the Manage Endpoint privilege is automatically granted to a user who has been granted the Create Endpoint privilege, when this user creates an endpoint, the Manage Endpoint privilege can be revoked from this user at any time, so that this user is restricted to creating endpoints only but not modifying or deleting them.

System Administrator Role and the Create Endpoint Privilege

A user with the System Administrator role can create and manage the endpoint. If the System Administrator role is revoked from a user, then the user can no longer manage the endpoint. However, when a user, with the System Administrator role and the Create Endpoint Group privilege, creates an endpoint, Oracle Key Vault grants the Manage Endpoint privilege on the endpoint to the user. If the System Administrator role is revoked from the user afterward, the user can continue to manage the endpoint that the user created.

Revocation of the Create Endpoint Privilege

A revoke of the Create Endpoint privilege prevents the user from being able to create more endpoints. Users cannot revoke the Create Endpoint privilege from themselves. The Create Endpoint privilege can be revoked from a user without affecting the user's Manage Endpoint privilege on one or more endpoints.

Deletion of Users Who Have Been Granted the Create Endpoint Privilege

The endpoint creator does not have perpetual ownership of the endpoint. The Manage Endpoint privilege on that endpoint granted to the endpoint's creator can be revoked, after which the endpoint creator cannot manage that endpoint anymore. Deletion of an endpoint's creator has no special significance other than the one that this user may have been granted the Manage Endpoint privilege at the time of the endpoint creation. If there are no other users with the Manage Endpoint privilege on that endpoint, then the System Administrator becomes responsible for managing that endpoint.

Related Topics

- [Manage Endpoint Privilege Duties and Scope](#)
The Manage Endpoint object privilege on an endpoint enables a user to perform all endpoint management operations on the endpoint.

Manage Endpoint Privilege Duties and Scope

The Manage Endpoint object privilege on an endpoint enables a user to perform all endpoint management operations on the endpoint.

The user who has the Manage Endpoint privilege on an endpoint can perform the following tasks:

- Perform all endpoint management operations (such as reenroll, suspend, resume, or delete) on the endpoint.
- Set the default wallet of that endpoint if the user also has full access on the subject wallet.

Who Can or Cannot Grant or Revoke the Manage Endpoint Privilege?

A user who has the System Administrator role can grant or revoke the Manage Endpoint privilege. If the user with the Create Endpoint privilege creates an endpoint, Oracle Key Vault automatically grants the Manage Endpoint privilege for this endpoint to the user.

System Administrator Role and Users Who Have the Manage Endpoint Privilege

A user who has the System Administrator role can manage any endpoint including those that are created by users with the Create Endpoint privilege. If the System Administrator role is revoked from a user, then the user can no longer manage any endpoint including those that the user created unless the user was also granted the Create Endpoint privilege either at the time of endpoint creation or afterward. The user must be granted either the System Administrator role or the Manage Endpoint privilege on an endpoint to manage that endpoint.

Objects and Wallets Associated with an Endpoint

Objects that are created by an endpoint are always owned by the endpoint. The endpoint creator or users with the Manage Endpoint privilege do not get automatic or implicit access to any objects that were created by the endpoint.

A user with the Manage Endpoint privilege can set the endpoint's default wallet if the user also has full access on the wallet. Revocation of the wallet access from the user afterward does not affect the endpoint's default wallet setting.

Revocation of the Manage Endpoint Privilege

The Manage Endpoint privilege can be revoked on the endpoint from a user without affecting the user's Create Endpoint privilege status. Users cannot revoke the Manage Endpoint privilege from themselves. If there are no users with the Manage Endpoint privilege for the endpoint, then managing that endpoint becomes the responsibility of users with the System Administrator role.

Deletion of Users Who Have Been Granted the Manage Endpoint Privilege

If the user who manages an endpoint is deleted, then the System Administrator or any other users who have the Manage Endpoint privilege for that endpoint can still manage the endpoint.

Related Topics

- [Create Endpoint Privilege Duties and Scope](#)
The Create Endpoint system privilege enables a user to create an endpoint.

Create Endpoint Group Privilege Duties and Scope

The Create Endpoint Group system privilege enables a user to create an endpoint group.

When a user who is granted the Create Endpoint Group privilege creates an endpoint group, Oracle Key Vault automatically grants the user the Manage Endpoint Group privilege on that endpoint group. This allows the user to manage the endpoint groups that they created without requiring additional privilege grants. However, the user's Manage Endpoint Group privilege on endpoint groups that they created can be revoked later.

Who Can or Cannot Grant or Revoke the Create Endpoint Group Privilege?

Only a user who has the Key Administrator role can grant or revoke the Create Endpoint Group privilege. Users who have the Create Endpoint Group privilege cannot grant or revoke this privilege to or from other users.

How the Create Endpoint Group Privilege Works with the Manage Endpoint Group Privilege

Even though the Manage Endpoint Group privilege is automatically granted to a user when the user creates an endpoint group, the Manage Endpoint Group privilege can be revoked for the endpoint group from this user at any time, so that this user can no longer modify or delete that endpoint group.

Key Administrator Role and Users Who Have the Create Endpoint Group Privilege

A user with the Key Administrator role can create and manage the endpoint groups. If the Key Administrator role is revoked from a user, then the user can no longer create and manage the endpoint groups. However, when a user with both the Key Administrator role and the Create Endpoint Group privilege creates an endpoint group, Oracle Key Vault grants the Manage Endpoint Group privilege on the endpoint group to the user. If the Key Administrator role is revoked from the user afterward, the user can continue to manage the endpoint groups that the user created.

Revocation of the Create Endpoint Group Privilege

A revoke of the Create Endpoint Group privilege prevents the user from being able to create more endpoint groups. Users cannot revoke the Create Endpoint Group privilege from themselves. The Create Endpoint Group privilege can be revoked from a user without affecting the user's Manage Endpoint Group privilege on one or more endpoint groups.

Deletion of Users Who Have Been Granted the Create Endpoint Group Privilege

The endpoint group creator does not have perpetual ownership of the endpoint group. The Manage Endpoint Group privilege on that endpoint group granted to the endpoint group's creator can be revoked, after which the endpoint group creator cannot manage that endpoint group anymore. Deletion of an endpoint group's creator has no special significance other than the one that this user may have been granted the Manage Endpoint Group privilege at the time of the endpoint group creation. If there are no other users with the Manage Endpoint Group privilege on that endpoint group, then the Key Administrator becomes responsible for managing that endpoint group.

Related Topics

- [Manage Endpoint Group Privilege Duties and Scope](#)
The Manage Endpoint Group object privilege on an endpoint group enables a user to perform all endpoint group management operations on the endpoint group.

Manage Endpoint Group Privilege Duties and Scope

The Manage Endpoint Group object privilege on an endpoint group enables a user to perform all endpoint group management operations on the endpoint group.

The user who has the Manage Endpoint Group privilege on an endpoint group can perform all endpoint group management operations (such as adding or removing an endpoint to or from an endpoint group, delete an endpoint group, and so on) on the endpoint group. This user inherits access to wallets from the endpoint group. This inheritance of wallet access permissions works in the same way as an endpoint or user inherits wallet permissions from an endpoint group or user group, respectively.

Who Can or Cannot Grant or Revoke the Manage Endpoint Group Privilege?

Only a user who has the Key Administrator role can grant or revoke the Manage Endpoint Group privilege. Users who have the Manage Endpoint Group privilege cannot grant or revoke this privilege to or from other users.

Key Administrator Role and Users Who Have the Manage Endpoint Group Privilege

A user who has the Key Administrator role can manage any endpoint group including those that are created by users with the Create Endpoint Group privilege. If the Key Administrator role is revoked from a user, then the user can no longer manage any endpoint groups including those that the user created unless the user was also granted the Create Endpoint Group privilege either at the time of endpoint group creation or afterward. The user must be granted either the Key Administrator role or the Manage Endpoint Group privilege on an endpoint group to manage that endpoint group.

Inheritance of Wallet Access from an Endpoint Group

- A user who has been granted the Manage Endpoint Group privilege on an endpoint group (this user is also called the endpoint group manager) inherits the access to wallets from the endpoint group. This inheritance of wallet access permissions works in the same way as an endpoint or user inherits wallet permissions from an endpoint group or user group, respectively.
Granting wallet access to an endpoint group results in implicitly giving the same wallet access to all the users who have the Manage Endpoint Group privilege on that endpoint group. These inherited wallet access from an endpoint group are effective as long as the user is the endpoint group manager for the endpoint group. This rule ensures that all endpoint group managers of an endpoint group always have at least the same level of access to wallets that the endpoint group has. This way, when an endpoint is added to the endpoint group, its endpoint group managers remain fully aware of the wallet permissions that the endpoint is implicitly being granted by virtue of adding an endpoint's membership into the endpoint group.
- A user's effective wallet access is the union of the following:
 - Direct wallet access grants to the user
 - Inherited wallet access grants from the user groups of which this user is a member
 - Inherited wallet access grants from the endpoint groups that this user is authorized to manage

How Wallets Are Affected by Revoke Operations

- Revoking a wallet's access from an endpoint group only results in the endpoint group's managers to lose the wallet access rights that were inherited from the respective endpoint group. The endpoint group's managers may continue to have access to the same wallet either from direct grants or from the inheritance of wallet access from user groups or other endpoint groups.
- Revoking the Manage Endpoint Group privilege on an endpoint group from a user (that is, an endpoint group manager) prevents the user from inheriting future wallet access rights from that endpoint group.
- An endpoint group's manager can revoke access of a wallet from the endpoint group (or from any other user, user group, endpoint, or endpoint group) if the user has the Manage Wallet Access privilege on that wallet even if this access was inherited from the endpoint group itself.

Revocation of the Manage Endpoint Group Privilege

The Manage Endpoint Group privilege can be revoked from a user without affecting the user's Create Endpoint Group privilege status. Users cannot revoke the Manage Endpoint Group privilege from themselves. If there are no users with the Manage Endpoint Group privilege for the endpoint, then managing that endpoint group becomes the responsibility of users with the Key Administrator role.

Deletion of Users Who Have Been Granted the Manage Endpoint Group Privilege

Deletion of an endpoint group's creator does not affect the endpoint groups that this user created. If the user who manages the endpoint group is deleted, then any other users who have the Manage Endpoint Group privilege for that endpoint group can still manage the endpoint group. If no such users exist, then a user who has the Key Administrator role becomes responsible for managing the endpoint group.

Related Topics

- [Create Endpoint Group Privilege Duties and Scope](#)
The Create Endpoint Group system privilege enables a user to create an endpoint group.

Monitor Privilege

Oracle Key Vault provides a privilege for creating and managing endpoints and endpoint groups.

- [Monitoring Privilege Duties and Scope](#)
The Monitor privilege enables a user to monitor the system-related information in Oracle Key Vault.

Monitoring Privilege Duties and Scope

The Monitor privilege enables a user to monitor the system-related information in Oracle Key Vault.

A user who holds the Monitor privilege in Oracle Key Vault can view system-related information, such as service status, CA certificate expiration date, cluster monitoring information, system metrics, and so on. Users with the Monitor privilege can only view this information but cannot act upon it to manage Oracle Key Vault.

Who Can or Cannot Grant or Revoke the Monitor Privileges?

Only a user who has the System Administrator role can grant or revoke the Monitor privilege. Users who have the Monitor privilege cannot grant or revoke this privilege to or from other users. The Monitor privilege can be assigned to any LDAP user too.

Revocation of the Monitor Privileges

Revoking the Monitor privilege prevents the user from viewing system-related or metrics information. Users cannot revoke the Monitor privilege from themselves.

Deletion of Users Who Have Been Granted the Monitor Privileges

Deleting a user with the Monitor privilege does not affect the system information or system metrics of Oracle Key Vault. Other users with the Monitor privilege or System Administrator role can continue to perform monitoring functions.

Naming Guidelines for Objects

The naming guidelines affect the following Oracle Key Vault objects: users, user groups, endpoints, endpoint groups, and virtual wallets.

The naming conventions for these objects are as follows:

- You can include the following characters in the names of endpoints, endpoint groups, user groups, and virtual wallets: letters (a–z, A–Z), numbers (0–9), underscores (_), periods (.), and hyphens (-).
- You can include the following characters in the names of users: letters (a–z, A–Z), numbers (0–9), and underscores (_).
- In most environments, the maximum number of bytes allowed for the name length is 120 bytes.
- The names of users, user groups, endpoints, and endpoint groups are not case sensitive. For example, `psmith` and `PSMITH` are considered the same user in Oracle Key Vault.
- The names of virtual wallets are case sensitive. For example, `wallet_hr` and `WALLET_HR` are considered two separate wallets in Oracle Key Vault.

Emergency System Recovery Process

During installation, you will be required to create a special recovery passphrase that Oracle Key Vault uses to recover from emergency situations.

These situations can arise due to administrative users not being immediately available, or something more commonplace such as forgotten passwords.

The recovery passphrase is needed in the following situations:

- If there is no administrative user available to log into Oracle Key Vault, then you can use the recovery passphrase to repeat the post-installation tasks and create new administrative users for system, key, and audit management.
- If you want to restore Oracle Key Vault from a previous backup, then you must have the recovery passphrase that is associated with that backup.
- You will be prompted for the recovery passphrase during the node induction process in the cluster mode.
- You will need it if you want to reset the recovery passphrase periodically.

For these reasons, it is very important to store the recovery passphrase in a safe and accessible place and keep track of older recovery passphrases. The recovery passphrase is the same passphrase that you use when you add nodes to a multi-master cluster.

The only way to recover from a lost recovery passphrase is to reinstall Oracle Key Vault.

Related Topics

- [Managing System Recovery](#)
System recovery includes tasks such as recovering lost administrative passwords.
- [Restoring Oracle Key Vault Data](#)
Oracle Key Vault data from a remote backup destination can be restored onto another Oracle Key Vault server.

Root and Support User Accounts

Both the `root` and `support` user accounts are used with the command-line interface.

The `root` user account is the super user account for the operating system that hosts Oracle Key Vault. You do not need the `root` account for normal Oracle Key Vault administration. Instead, you must use the `root` account when you want to upgrade to a later bundle patch or perform some command-line operations such as adding disk space. The `support` user is the only account that can remotely log in to the operating system hosting the Oracle Key Vault when SSH is enabled.

Be aware that if you enter the `root` or `support` passwords incorrectly three times, then the account is locked for a brief period.

Endpoint Managers

An endpoint manager owns and manages endpoints, which are entities such as Oracle Databases that use Oracle Key Vault.

This user is typically a system, security, or database administrator but can be any personnel charged with deploying, managing, and maintaining security of the endpoint. Endpoint managers are responsible for provisioning endpoints. The endpoint manager user is different from an endpoint administrator who is an Oracle Key Vault user with the Manage Endpoints and Manage Endpoint Group privileges.

The endpoint manager for an Oracle Database endpoint is the security administrator of the database.

Endpoint Administrators

An endpoint administrator is an Oracle Key Vault user who is responsible for registering, enrolling, and managing one or more Oracle Key Vault endpoints.

To ensure isolation among endpoints from different business units, on-premises and cloud deployments, you may designate an endpoint administrator to manage their own endpoints. An Oracle Key Vault system administrator grants Create Endpoint or Manage Endpoint privileges to an endpoint administrator.

For ease-of-management, you may also grant the Create Endpoint Group or Manage Endpoint Group privileges to an endpoint administrator. This way, an endpoint administrator can completely manage the endpoints and their configuration in Oracle Key Vault without requiring any involvement from Oracle Key Vault administrators.

FIPS Mode

FIPS mode enables Oracle Key Vault to adhere to FIPS 140-2 compliance.

Federal Information Processing Standards (FIPS) are standards and guidelines for federal computer systems that are developed by the US National Institute of Standards and Technology (NIST) in accordance with the Federal Information Security Management Act (FISMA). Although FIPS was developed for use by the federal government, many private sector entities voluntarily use these standards.

FIPS 140-2 specifies the security requirements that will be satisfied by a cryptographic module, providing four increasing, qualitative levels that are intended to cover a range of potential applications and environments. Security Level 1 provides the basic security requirements for a cryptographic module. FIPS 140-2 Security Level 1 requires no physical security mechanisms in the module beyond the requirement for production-grade equipment. As a result, this level allows software cryptographic functions to be performed in a general-purpose computer running on a specified operating environment.

FIPS mode enables Oracle Key Vault to adhere to FIPS 140-2 Level 1 compliance. When FIPS 140-2 settings are configured for Oracle Key Vault, it uses FIPS 140-2 Level 1 validated cryptographic libraries to protect keys and data at rest and in transit over the network. Oracle Key Vault clients such as the database PKCS#11 library, `okvutil`, C-SDK, Java-SDK and RESTful services utility make key management requests to the Oracle Key Vault servers. Oracle Key Vault components such as the KMIP service, the Oracle Key Vault application, embedded Oracle Database, and Oracle GoldenGate will process the request and respond to the clients. Administration of Oracle Key Vault is done either using the web console (management console) or REST Admin APIs. The Oracle Key Vault management console is implemented using Oracle APEX backed by Apache and the Oracle Database. The RESTful services utility requests are handled by a Tomcat server. Oracle Key Vault backup is done using secure copy protocol (SCP) or the SSH file transfer protocol (SFTP). The overall Oracle Key Vault system runs on Oracle Linux. These Oracle Key Vault components either use the DELL BSAFE or OpenSSL libraries for their cryptographic operations. Each of these components operates the cryptographic libraries in FIPS mode when FIPS mode is enabled for Oracle Key Vault.

Oracle Key Vault currently uses the following:

- **Dell BSAFE, formerly known as RSA BSAFE, as the FIPS 140-2 level 1 validated cryptography library:** The specific version of the module used is Dell BSAFE Micro Edition Suite 4.6, which integrates Dell BSAFE Crypto-C Micro Edition (CCME) 4.6.1.0.1 as its underlying FIPS.
- **openssl-1.1.1k-14.el8_6.x86_64 :** The specific version of the `openssl` module that is FIPS certified on Oracle Linux is `openssl-libs-1.1.1g-15.el8_3.x86_64.rpm`

 **Note:**

Oracle Key Vault FIPS mode enforces the use of FIPS-approved algorithms for the Oracle Key Vault only. Third-party vendor software that is used with Oracle Key Vault, such as the HSM client software deployed to support root of trust, may operate in FIPS but must use FIPS-approved algorithms, or else the vendor software will not operate when Oracle Key Vault is in FIPS mode or will have failures.

Related Topics

- [Configuring the FIPS Mode for the Node](#)
All multi-master cluster nodes must use the same FIPS mode setting or you will receive an alert.

3

Oracle Key Vault Multi-Master Cluster Concepts

A multi-master cluster is a fully connected network of Oracle Key Vault servers called nodes.

- [Oracle Key Vault Multi-Master Cluster Overview](#)
The multi-master cluster nodes provide high availability, disaster recovery, load distribution, and geographic distribution to an Oracle Key Vault environment.
- [Benefits of Oracle Key Vault Multi-Master Clustering](#)
The Oracle Key Vault multi-master cluster configuration provides high available, scalable and fault-tolerant key and secrets management.
- [Multi-Master Cluster Architecture](#)
An Oracle Key Vault node can be a read/write or a read-only node operating in different modes. Nodes can also form a subgroup.
- [Building and Managing a Multi-Master Cluster](#)
You initialize a multi-master cluster using a single Oracle Key Vault server.
- [Oracle Key Vault Multi-Master Cluster Deployment Scenarios](#)
All multi-master cluster nodes can serve endpoints actively and independently.
- [Multi-Master Cluster Features](#)
Oracle Key Vault provides features that help with inconsistency resolution and name conflict resolution in clusters, and endpoint node scan lists.
- [Cluster Management Information](#)
The Cluster Management page provides a concise overview of the cluster and the status of each node.

Oracle Key Vault Multi-Master Cluster Overview

The multi-master cluster nodes provide high availability, disaster recovery, load distribution, and geographic distribution to an Oracle Key Vault environment.

Oracle recommends that you use the Oracle Key Vault multi-master clusters deployment over other Oracle Key Vault deployments.

An Oracle Key Vault multi-master cluster provides a mechanism to create read/write pairs of Oracle Key Vault nodes for maximum availability and reliability. You can add read-only Oracle Key Vault nodes to the cluster to provide even greater availability to endpoints that need Oracle wallets, encryption keys, Java keystores, certificates, credential files, and other objects.

An Oracle Key Vault multi-master cluster is an interconnected group of Oracle Key Vault [nodes](#). Each node in the cluster is automatically configured to connect with all the other nodes, in a fully connected network. The nodes can be geographically distributed. Oracle Key Vault [endpoints](#) interact with any node in the cluster.

This configuration replicates data to all other nodes, reducing risk of data loss. To prevent data loss, you must configure pairs of nodes called [read/write pairs](#) to enable bidirectional synchronous replication. This configuration enables an update to one node to be replicated to the other node, and verifies this on the other node, before the update is considered successful.

Critical data can only be added or updated within the read/write pairs. All added or updated data is asynchronously replicated to the rest of the cluster.

The Oracle Key Vault multi-master cluster configuration provides highly available, scalable and fault-tolerant key and secrets management. Any additional Oracle Key Vault server that is to join the cluster must be at the same release level as the cluster.

The clocks on all the nodes of the cluster must be synchronized. Consequently, all nodes of the cluster must have the Network Time Protocol (NTP) settings enabled.

Every node in the cluster can serve endpoints actively and independently while maintaining an identical dataset through continuous replication across the cluster. The smallest possible configuration is a two node cluster, and the largest configuration can have up to 16 nodes with several pairs spread across on-premises data centers and/or cloud tenancies in Oracle OCI, Microsoft Azure, Amazon AWS, and Google GCP.

Benefits of Oracle Key Vault Multi-Master Clustering

The Oracle Key Vault multi-master cluster configuration provides high available, scalable and fault-tolerant key and secrets management.

To ensure high availability for geographically distributed [endpoints](#), Oracle Key Vault [nodes](#) that are deployed in different data centers operate in active-active multi-master cluster configurations to create and share keys. With an active-active configuration, there are no passive machines in the cluster, which allows for better resource utilization. An added benefit of the multi-master cluster configuration is load distribution. When multiple Oracle Key Vault nodes in multi-master configuration are deployed in a data center, they can actively share the key requests of the endpoint databases in that data center.

In a typical large scale deployment, Oracle Key Vault must serve a large number of endpoints, possibly distributed in geographically distant data centers.

In comparison to a multi-master deployment, standalone Oracle Key Vault deployments provide the least availability, while primary-standby deployments (deprecated) offered limited availability:

- A primary-standby configuration only has a single primary Oracle Key Vault server that can actively serve clients.
- If the server running in the standby role is unavailable, then the server running in the primary role is in read-only mode and does not allow any write operations.
- The primary-standby mode can support either high availability in the same data center or disaster recovery across data centers.
- If the persistent master encryption key cache is not enabled, then database downtime is unavoidable during maintenance windows.

The Oracle Key Vault multi-master cluster configuration addresses these limitations. You can geographically disperse nodes to provide simultaneous high availability and disaster recovery capability.

An Oracle Key Vault multi-master cluster configuration offers the following significant benefits:

- Data compatibility between multi-master cluster nodes similar to a primary-standby deployment.

Because all the nodes have an identical data set, the endpoints can retrieve information from any node. In a cluster, the unavailability of an Oracle Key Vault node does not affect the operations of an endpoint. If a given node is unavailable, then the endpoint interacts transparently with another node in the cluster.

- **Fault tolerance**

Successfully enrolled clients transparently update their own list of available Oracle Key Vault nodes in the cluster. This enables clients to locate available nodes at any given time, without additional intervention. As such, unexpected failure in nodes or network disruptions do not lead to service interruption for endpoints as long as at least one operational Oracle Key Vault read/write pair remains accessible to the endpoint. If all read/write pairs are unavailable to an endpoint, but a read-only restricted node is available, then the endpoint can still invoke read-only operations.
- **Zero data loss**

Data that has been added or updated at a read/write node is immediately replicated to its read/write peer and must be confirmed at the peer to be considered committed. It is then distributed across the cluster. Therefore, data updates are considered successful only if they are guaranteed to exist in multiple servers.
- **No passive machines in the system**

A primary-standby configuration requires a passive standby server. The Oracle Key Vault multi-master cluster contains only active servers. This allows for better utilization of hardware.
- **Horizontally and vertically scaling up or down**

You can add or remove nodes from the cluster, put cluster nodes on bigger or smaller servers or cloud compute instances, without interrupting any key management services to clients. This means the number of nodes in the cluster can be increased or decreased as required to meet the expected workload.
- **Maintenance**

Whenever hardware or software maintenance is required, Oracle Key Vault nodes can leave the cluster and return back to the cluster after maintenance. The remaining nodes continue to serve the clients. Properly planned maintenance does not cause any service downtime, avoiding interruption of service to endpoints. Unplanned downtime equally does not interfere with database and application availability: If an Oracle Key Vault node suffers irreparable hardware failures, a fresh installed Oracle Key Vault node of the same version can replace it.

Multi-Master Cluster Architecture

An Oracle Key Vault node can be a read/write or a read-only node operating in different modes. Nodes can also form a subgroup.

- **[Oracle Key Vault Cluster Nodes](#)**

An Oracle Key Vault node is an Oracle Key Vault server that operates as a member of a multi-master cluster.
- **[Cluster Node Limitations](#)**

Limitations to cluster nodes depend on whether the node is asynchronous or synchronous.
- **[Cluster Subgroups](#)**

A cluster subgroup is a group of one or more nodes of the cluster.
- **[Critical Data in Oracle Key Vault](#)**

Oracle Key Vault stores critical data that is necessary for the endpoints to operate. Read-only nodes can read critical data as well.

- [Oracle Key Vault Read/Write Nodes](#)
A read/write node is a node in which critical data can be added or updated using the Oracle Key Vault or endpoint software.
- [Oracle Key Vault Read-Only Nodes](#)
In a read-only node, users can add or update non-critical data but not add or update critical data. However, read-only nodes can read critical data.
- [Cluster Node Mode Types](#)
Oracle Key Vault supports two types of mode for cluster nodes: read-only restricted mode or read/write mode.
- [Operations Permitted on Cluster Nodes in Different Modes](#)
In an Oracle Key Vault multi-master cluster, operations are available or restricted based on the node and the operating mode of the node.

Oracle Key Vault Cluster Nodes

An Oracle Key Vault node is an Oracle Key Vault server that operates as a member of a multi-master cluster.

To configure an Oracle Key Vault server to operate as a member of the cluster, you must convert it to be a multi-master cluster node. The process is referred to as node induction.

On induction, Oracle Key Vault modifies the **Cluster** tab to enable management, monitoring, and conflict resolution capabilities on the management console of the node. Cluster-specific features of the management console, such as cluster settings, audit replication, naming resolution, cluster alerts, and so on, are enabled as well.

The Primary-Standby Configuration page is not available on the Oracle Key Vault management console of a node. A node cannot have a passive standby server, nor can it become a passive standby server.

A node runs additional services to enable it to communicate with the other nodes of the cluster. Endpoints enrolled from a node are made aware of the cluster topology.

Each node in the cluster has a user-allocated node identifier. The node identifier must be unique in the cluster.

The Oracle Key Vault release in which the node was created will affect the byte length for the names of users, user groups, endpoints, endpoint groups, and virtual wallets. See [Naming Guidelines for Objects](#).

Cluster Node Limitations

Limitations to cluster nodes depend on whether the node is asynchronous or synchronous.

The nodes of a Oracle Key Vault multi-master cluster replicate data asynchronously between them. The only exception is replicating data to the [read-write peer](#). There are various limitations arising as a result of the asynchronous replicate operations. For example, the IP addresses of a node in the cluster are static and cannot be changed after the node joins the cluster. If you want the node to have a different IP address, then delete the node from the cluster, and either add a new node with the correct IP address, or reimagine the deleted node using the correct IP address before adding it back to the cluster.

You can perform only one cluster change operation, such as adding, disabling, or deleting a node at a time.

Node IDs are unique across the cluster. Ensure that the node ID is unique when you select the node during the induction process.

An Oracle Key Vault cluster does a best-effort job at preventing users from performing unsupported actions. It will try to remove node IDs that are in-use from the node induction drop-down on the user interface and prevent the user from adding a second read/write peer if one already exists. Similarly, a multi-master cluster will prevent the user from adding a second read/write peer if one already exists.

Cluster Subgroups

A cluster subgroup is a group of one or more nodes of the cluster.

A cluster can be conceptually divided into one or more cluster subgroups.

The node is assigned to a subgroup when you add the node to the multi-master cluster. You can change the assignment at any time, so long as a certificate rotation operation is not in progress. A node's cluster subgroup assignment is a property of the individual node, and members of a read/write pair may be in different cluster subgroups.

A cluster subgroup represents the notion of endpoint affinity. A node's cluster subgroup assignment is used to set the search order in the endpoint's node scan list. Nodes in the same cluster subgroup as an endpoint are considered local to the endpoint. You can check which node this refers to by checking the Cluster Subgroup column on the Endpoints page. The nodes within an endpoint's local subgroup are scanned first, before communicating with nodes that are not in the local subgroup.

The cluster topology can change when you add or remove new nodes to and from the cluster. Nodes can also be added or removed from the local cluster subgroup. Each endpoint may get updates to this information along with the response message for any successful non-empty operation which the endpoint initiated. The updated endpoint's node scan list is sent back to the endpoint periodically even if there is no change to cluster topology. This is to make up for any lost messages.

Critical Data in Oracle Key Vault

Oracle Key Vault stores critical data that is necessary for the endpoints to operate. Read-only nodes can read critical data as well.

The loss of this information can result in the loss of data on the [endpoint](#). Endpoint encryption keys, certificates, and similar [security objects](#) that Oracle Key Vault manages are examples of critical data in Oracle Key Vault. Critical data must be preserved in the event of an Oracle Key Vault server failure to ensure endpoint recovery and continued operations.

Oracle Key Vault data that can be re-created or discarded after an Oracle Key Vault server failure is non-critical data. Cluster configuration settings, alert settings, and email settings are examples of non-critical data.

Oracle Key Vault Read/Write Nodes

A read/write node is a node in which critical data can be added or updated using the Oracle Key Vault or endpoint software.

The critical data that is added or updated can be data such as keys, wallet contents, and certificates.

Oracle Key Vault read/write nodes always exist in pairs. Each node in the read/write pair can accept updates to critical and non-critical data, and these updates are immediately replicated to the other member of the pair, the read/write peer. A read/write peer is the specific member of one, and only one, read/write pair in the cluster. There is bi-directional synchronous replication

between read/write peers. Replication to all nodes that are not a given node's read/write peer is asynchronous.

A node can be a member of, at most, one read/write pair. A node can have only one read/write peer. A node becomes a member of a read/write pair, and therefore a read/write node, during the induction process. A read/write node reverts to being a read-only node when its read/write peer is deleted, at which time it can form a new read/write pair.

A read/write node operates in read/write mode when it can successfully replicate to its read/write peer and when both peers are active. A read/write node is temporarily placed in read-only restricted mode when it is unable to replicate to its read/write peer or when its read/write peer is disabled.

An Oracle Key Vault multi-master cluster requires at least one read/write pair to be fully operational. It can have a maximum of 8 read/write pairs.

Oracle Key Vault Read-Only Nodes

In a read-only node, users can add or update non-critical data but not add or update critical data. However, read-only nodes can read critical data.

Critical data is updated only through replication from other nodes.

A read-only node is not a member of a read/write pair and does not have an active read/write peer.

A read-only node can induct a new server into a multi-master cluster. The new node can be another read-only node. However, a read-only node becomes a read/write node if it inducts another node as its read/write peer.

The first node in the cluster is a read-only node. Read-only nodes are used to expand the cluster. A multi-master cluster, after it has been built, does not need to have any read-only nodes. A multi-master cluster with only read-only nodes is not ideal because no useful critical data can be added to such a multi-master cluster.

Cluster Node Mode Types

Oracle Key Vault supports two types of mode for cluster nodes: read-only restricted mode or read/write mode.

- **Read-only restricted mode:** In this mode, only non-critical data can be updated or added to the node. Critical data can be updated or added only through replication in this mode. There are two situations in which a node is in read-only restricted mode:
 - A node is read-only and does not yet have a [read-write peer](#).
 - A node is part of a read/write pair but there has been a breakdown in communication with its read/write peer or if there is a node failure. When one of the two nodes is non-operational, then the remaining node is set to be in the read-only restricted mode. When a read/write node is again able to communicate with its read/write peer, then the node reverts back to read/write mode from read-only restricted mode.
- **Read/Write mode:** This mode enables both critical and non-critical information to be written to a node. A read/write node should always operate in the read/write mode.

You can find the mode type of the cluster node on the Monitoring page of the **Cluster** tab of the node management console. The **Cluster** tab of any node management console displays the mode type of all nodes in the cluster.

Operations Permitted on Cluster Nodes in Different Modes

In an Oracle Key Vault multi-master cluster, operations are available or restricted based on the node and the operating mode of the node.

Related Topics

- [Oracle Key Vault Multi-Master Cluster Operations](#)
There are restrictions and conditions for Oracle Key Vault multi-master cluster operations on cluster nodes.

Building and Managing a Multi-Master Cluster

You initialize a multi-master cluster using a single Oracle Key Vault server.

- [About Building and Managing a Multi-Master Cluster](#)
After the initial cluster is created in the Oracle Key Vault server, you can add the different types of nodes that you need for the cluster.
- [Creation of the Initial Node in a Multi-Master Cluster](#)
The initial node in a multi-master cluster must follow certain requirements before being made the initial node.
- [Expansion of a Multi-Master Cluster](#)
After you initialize the cluster, you can expand it by adding up to 15 more nodes, as either read/write pairs or read-only nodes.
- [Migration to the Cluster from an Existing Deployment](#)
You can migrate an existing Oracle Key Vault deployment to a multi-master cluster node.

About Building and Managing a Multi-Master Cluster

After the initial cluster is created in the Oracle Key Vault server, you can add the different types of nodes that you need for the cluster.

This Oracle Key Vault server seeds the cluster data and converts the server into the first cluster node, which is called the [initial node](#). The cluster is expanded when you induct additional Oracle Key Vault servers, and add them as [read/write nodes](#), or as simple [read-only nodes](#).

A multi-master cluster can contain a minimum of 2 nodes and a maximum of 16 nodes.

Creation of the Initial Node in a Multi-Master Cluster

The initial node in a multi-master cluster must follow certain requirements before being made the initial node.

You create a multi-master cluster by converting a single Oracle Key Vault server to become the initial node. The Oracle Key Vault server can be a freshly installed Oracle Key Vault server, or it can already be in service with existing data. A standalone server, or a primary server of a primary-standby configuration can be converted to the initial node of a cluster.

Before using the primary server of the primary-standby configuration, you must unpair the primary-standby configuration. For a primary-standby configuration, you can use either of the following methods to upgrade to a cluster:

- Method 1:

1. Back up the servers.
 2. Upgrade both the primary and standby servers to the latest release.
 3. Unpair the paired primary and standby servers. (Before you unpair the servers, see *Oracle Key Vault Release Notes* for known issues regarding the unpair process.)
 4. Convert the primary server to be the first node of the cluster.
- Method 2:
 1. Back up the servers.
 2. Unpair the paired primary and standby servers.
 3. Upgrade the former primary server to the latest release.
 4. Convert the primary server to be the first node of the cluster.

The freshly installed servers must be of the same current version as the initial node. The initial node is special in that it provides the entirety of the data with which the cluster is initialized. This happens only once for the cluster when it is created. The data provided by the initial node will include but is not limited to the following components:

- Certificates, keys, wallets, and other security objects
- Users and groups
- Endpoint information
- Audits
- Reports

All other nodes added after the initial node must be created from freshly installed or cloned Oracle Key Vault servers of the same version as the initial node.

The cluster name is chosen when the initial node is created. Once this name is chosen, you cannot change the cluster name.

The cluster subgroup of the initial node is also configured when the initial node is created. You must configure an Oracle Key Vault server that is converted to the initial node to use a valid Network Time Protocol (NTP) setting before you begin the conversion. The initial node always starts as a [read-only node](#) in [read-only restricted mode](#).

Related Topics

- [Creating the First \(Initial\) Node of a Cluster](#)
To create a cluster, you must convert an existing standalone Oracle Key Vault server to become the first node in the cluster.
- [Oracle Key Vault Release Notes](#)

Expansion of a Multi-Master Cluster

After you initialize the cluster, you can expand it by adding up to 15 more nodes, as either read/write pairs or read-only nodes.

- [About the Expansion of a Multi-Master Cluster](#)
Node induction is the process of configuring an Oracle Key Vault server to operate as a multi-master cluster node.
- [Management of Cluster Reconfiguration Changes Using a Controller Node](#)
A controller node is the node that controls or manages a cluster reconfiguration change, such as adding, enabling, disabling, or removing nodes.

- [Addition of a Candidate Node to the Multi-Master Cluster](#)
A freshly installed Oracle Key Vault server that is being added to a cluster is called a candidate node.
- [Addition of More Nodes to a Multi-Master Cluster](#)
You add nodes one at a time, first as a single read-only node, and then later as read/write paired nodes.

About the Expansion of a Multi-Master Cluster

Node induction is the process of configuring an Oracle Key Vault server to operate as a multi-master cluster node.

A [controller node](#) inducts an Oracle Key Vault server that has been converted to a [candidate node](#) into the cluster.

To expand a multi-master cluster, you use the induction process found on the **Cluster** tab of the Oracle Key Vault Management console, or a RESTful API call. Nodes added to the Oracle Key Vault multi-master cluster are initialized with the current cluster data. You can add nodes either as [read/write peers](#), or as [read-only nodes](#).

Related Topics

- [Adding a Node to Create a Read/Write Pair](#)
After you create the initial node, you must add an additional read/write peer to the cluster.
- [Adding a Node as a Read-Only Node](#)
To add a new read-only cluster node, you add a newly configured server from any existing cluster node.

Management of Cluster Reconfiguration Changes Using a Controller Node

A controller node is the node that controls or manages a cluster reconfiguration change, such as adding, enabling, disabling, or removing nodes.

A node is only a controller node during the life of the change. During induction, the controller node provides the server certificate and the data that is used to initialize the candidate node. Another node can be the controller node for a subsequent cluster change. One controller node can only control one cluster configuration change at a time. Oracle Key Vault does not permit multiple cluster operations at the same time.

Oracle recommends that you perform one cluster operation at a time. Each concurrent operation will have its own controller node.

The following table shows the role of the controller and controlled nodes during various cluster configuration.

Cluster Configuration Operation	Controller Node	Node Being Controlled
Induction as the first node	Any server	The controller node itself
Induction as a read-only node	Any node	Any server
Induction as a read-write node	Any node that does not have a read-write peer	Any server
Disable a node	Any node in the cluster	Any node in the cluster
Enable a node	Only the disabled node can re-enable itself	The disabled node itself
Delete a node	Any node in the cluster	Any other node in the cluster

Cluster Configuration Operation	Controller Node	Node Being Controlled
Force Delete a node	Any node in the cluster	Any other node in the cluster
Manage inbound replication	Any node in the cluster	The node itself

Addition of a Candidate Node to the Multi-Master Cluster

A freshly installed Oracle Key Vault server that is being added to a cluster is called a candidate node.

In the process of adding the server to a cluster, Oracle Key Vault converts the server to a candidate node before it becomes a node of the cluster. To induct an Oracle Key Vault server into a cluster, you must provide necessary information such as the controller certificate, the server IP address, and the recovery passphrase so that the new candidate can successfully and securely communicate with the controller node.

You must also make sure that the network ports used for communication between multi-master cluster nodes are open before initiating node addition.

You can convert a freshly installed or cloned Oracle Key Vault server into a candidate node from the **Cluster** tab of the Oracle Key Vault Management console or through a RESTful API call. When a candidate node is inducted into a multi-master cluster, any pre-existing data on that node is wiped out and then replaced by a copy of the data from the cluster.



Note:

Starting with Oracle Key Vault release 21.7, systems that were upgraded from older releases cannot be added to multi-master clusters.

Related Topics

- Network Port Requirements

Addition of More Nodes to a Multi-Master Cluster

You add nodes one at a time, first as a single read-only node, and then later as read/write paired nodes.

When an Oracle Key Vault multi-master cluster is first created, and only contains one node, that **initial node** is a **read-only node**. After you have created the initial node, you can induct additional read-only nodes or a **read/write peer**. After these nodes have been added, you can further add read-only nodes or add a read/write peer.

Because the initial node is in **read-only restricted mode** and no critical data can be added to it, Oracle recommends that you induct a second node to form a **read/write pair** with the first node. You should expand the cluster to have read/write pairs so that both critical and non-critical data can be added to the read/write nodes.

The general process for adding nodes to a cluster is to add one node at a time, and then pair these so that they become read/write pairs:

1. Add the initial node (for example, N1). N1 is a read-only node.
2. Add the second node, N2. N2 will be in read-only restricted mode during the induction process.

If you are adding N2 as the read/write peer of N1, then both N1 and N2 will become read/write nodes when N2 is added to the cluster. Otherwise, N1 and N2 will remain read-only nodes after you add N2 to the cluster. If you want to add N2 as the read/write peer of N1, then you must set **Add Candidate Node as Read-Write Peer** to **Yes** on N1 during the induction process. If you do not want N2 to be paired with N1, then set **Add Candidate Node as Read-Write Peer** to **No**. This example assumes that N2 will be made a read/write peer of N1.

3. Add a third node, N3, which must be a read-only node if N1 and N2 were made read/write peers, because the other two nodes in the cluster already are read/write peers. In fact, you can add multiple read-only nodes to this cluster, but Oracle recommends that you not do this, because when write operations take place, the few read/write nodes that are in the cluster will be overloaded. For optimum performance and load balancing, you must have more read/write pairs.
4. To create a second read/write pair for the cluster, when you add the next node (N4), set **Add Candidate Node as Read-Write Peer** to **Yes** to add node N4 to be paired with node N3. Node N4 must be added from N3 to make the second read/write pair, because N3 is the only node without a read/write peer at this point. After you complete this step, at this stage the cluster has two read/write pairs: the N1-N2 pair, and the N3-N4 pair.
5. To create the next pairing, add the next read-only node (for example, N5), followed by node N6. Be sure to set **Add Candidate Node as Read-Write Peer** to **Yes** when you add N6. Node N6 must be added to N5 because at this point, N5 is the only node without a read/write peer. By the time you complete this step, there will be three read/write pairs in the cluster: the N1-N2 pair, the N3-N4 pair, and the N5-N6 pair.

A freshly installed Oracle Key Vault server (same version as the other nodes in the cluster) can be converted to a [candidate node](#). You should confirm that the candidate node has Network Time Protocol (NTP) configured, as NTP on all existing and incoming Oracle Key Vault nodes is mandatory.

Any node in the cluster can be the [controller node](#) given no other cluster change operations are in progress. The candidate node and the controller node exchange information that enables the controller node to ascertain the viability of induction. Induction replicates the cluster data set to the candidate node. After a successful induction, you can configure the node to use the cluster-wide configuration settings. A cluster data set includes but is not limited to the following components:

- Certificates, keys, wallets, and other security objects
- Users and user groups
- Endpoint and endpoint group information
- Audit data
- Cluster name and cluster node details
- Cluster settings

The controller node assigns the node ID and the cluster subgroup for the candidate node. If the controller node provides an existing cluster subgroup during induction, then the candidate node becomes part of that subgroup. If the controller node provides the name of the cluster subgroup that does not exist, then the cluster subgroup is created as part of the induction process and the candidate node is added to the cluster subgroup. You can have all endpoints associated with one subgroup, if you want. For example, all endpoints in data center A can be in one subgroup, and all endpoints in data center B can be in another subgroup. Endpoints that are in the same subgroup will prioritize connecting to the nodes in that subgroup before connecting to nodes in other subgroups.

A read/write node can become a read-only node if its read/write peer is deleted. This read-only node can be used to form another read/write pair.

Related Topics

- [Adding a Node as a Read-Only Node](#)
To add a new read-only cluster node, you add a newly configured server from any existing cluster node.
- [Creating an Additional Read/Write Pair in a Cluster](#)
Any node can be read/write paired with only one other node, and there can be multiple read/write pairs in a cluster.

Migration to the Cluster from an Existing Deployment

You can migrate an existing Oracle Key Vault deployment to a multi-master cluster node.

- [Conversion of an Oracle Key Vault Standalone Server to a Multi-Master Cluster](#)
You can migrate a standalone Oracle Key Vault deployment that is at an older release to a multi-master deployment.
- [Conversion from a Primary-Standby Server to a Multi-Master Cluster](#)
You can migrate an Oracle Key Vault primary-standby deployment to a multi-master deployment.

Conversion of an Oracle Key Vault Standalone Server to a Multi-Master Cluster

You can migrate a standalone Oracle Key Vault deployment that is at an older release to a multi-master deployment.

First, you must upgrade the server to the latest Oracle Key Vault release. After you complete the upgrade, you then can convert it to an [initial node](#).

If your Oracle Key Vault server deployment is already at the current release, then you can directly convert it to an initial node.

The initial node will retain all the data of the existing Oracle Key Vault standalone deployment. After you create the initial node of the cluster, you can add more nodes to this cluster as necessary. Ensure that the new nodes that you add were installed at the current version of the initial node.

Related Topics

- [Oracle Key Vault Release Notes](#)
- [Creating the First \(Initial\) Node of a Cluster](#)
To create a cluster, you must convert an existing standalone Oracle Key Vault server to become the first node in the cluster.

Conversion from a Primary-Standby Server to a Multi-Master Cluster

You can migrate an Oracle Key Vault primary-standby deployment to a multi-master deployment.

First, you must unpair the primary-standby server configuration. Then you should upgrade the unpaired former primary server to the latest Oracle Key Vault release. (Alternatively, you can perform the upgrade and then unpair the primary-standby server configuration.) After completing these steps, you can convert the upgraded, former primary server to an [initial node](#).

If you have the latest release of Oracle Key Vault in a primary-standby deployment, you must also unpair it before you can move it to a multi-master cluster. Then, you can directly convert the former primary server to an initial node.

Before you perform this kind of migration, you should back up the servers that are currently used in the primary-standby deployment.

Related Topics

- [Disabling or Unpairing the Primary-Standby Configuration](#)
You can disable the primary-standby configuration by unpairing the primary and standby servers.
- [Creating the First \(Initial\) Node of a Cluster](#)
To create a cluster, you must convert an existing standalone Oracle Key Vault server to become the first node in the cluster.
- [Creation of the Initial Node in a Multi-Master Cluster](#)
The initial node in a multi-master cluster must follow certain requirements before being made the initial node.

Oracle Key Vault Multi-Master Cluster Deployment Scenarios

All multi-master cluster nodes can serve endpoints actively and independently.

They can do this while striving to maintain an identical cluster data set through continuous replication across the cluster. Deployment scenarios of the multi-master cluster can range from a small two-node cluster to large 16-node deployments spanning across data centers.

- [Cluster Size and Availability in Deployments](#)
In general, the availability of the critical data to the endpoints increases with the increasing size of the cluster.
- [Two-Node Cluster Deployment](#)
A single read/write pair formed with two Oracle Key Vault nodes is the simplest multi-master cluster.
- [Mid-Size Cluster Across Two Data Centers Deployment](#)
A two-data center configuration provides high availability, disaster recovery, and load distribution.

Cluster Size and Availability in Deployments

In general, the availability of the critical data to the endpoints increases with the increasing size of the cluster.

You may be required to disable both nodes in a read/write pair from the cluster together to undergo maintenance such as patching or upgrade. If you have a cluster of only two nodes, you get continuous read availability; there is no downtime for applications or databases, but re-key operation, or onboarding of new databases into Oracle Key Vault is not possible.

A two-node cluster is suitable for deployments where re-key operations and on-boarding of new databases can be suspended during OKV maintenance, as well as development and test environments. Large scale deployments under heavy load should deploy at least two read/write pairs to ensure endpoint continuity.

A three-node cluster with one read/write pair and one [read-only node](#) fares better than a two-node cluster because it provides endpoint continuity so long as no critical data is added or updated.

A four-node, two read/write pair cluster provides continuous read/write availability for all endpoint operations while one nodes is in maintenance for patching or upgrades.

The cluster should ideally be comprised of read/write pairs. If network latency or network interruptions across data centers is of little concern, then you should deploy read/write pairs across data centers. In case of a disaster, where one node of a read/write pair is lost, the keys are preserved in the read/write peer. However, if the network latency or interruptions are of concern, then you should place the read/write pair in the same data center. A disaster resulting in the loss of read/write pair may result in the loss of keys or other data if the disaster strikes before the created data is asynchronously replicated to other nodes.

Two-Node Cluster Deployment

A single read/write pair formed with two Oracle Key Vault nodes is the simplest multi-master cluster.

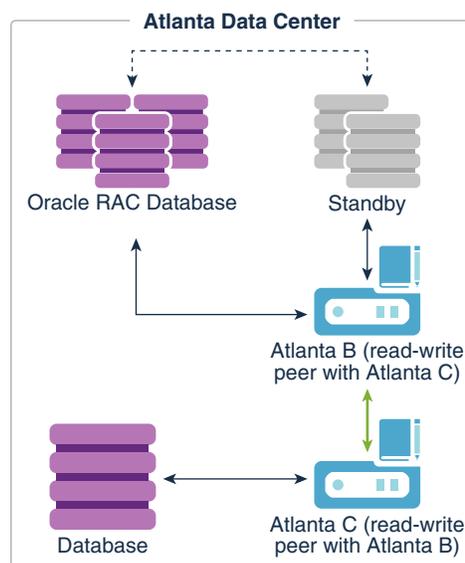
A two-node multi-master cluster looks similar to a standard primary-standby environment, in that there are only two nodes. The significant difference is that unlike the primary-standby configuration where the standby is passive, both nodes are active and can respond to endpoint requests at the same time.

Using a two-node multi-master cluster provides the following advantages over a primary-standby environment:

- Both nodes can be actively queried and updated by endpoints unlike the primary-standby configuration where only the primary server can be queried.
- A multi-master cluster can be expanded to three or more nodes without downtime.
- If the nodes are in separate data centers, then endpoints can prioritize interacting with nodes in the same subgroup, rather than reach across the network to the primary server node.

The following figure describes the deployment used for a two-node, single data center.

Figure 3-1 Oracle Key Vault Multi-Master Cluster Deployment in Single Data Center



In this scenario, the Atlanta Data Center hosts three databases, as follows:

- A single instance database
- A multi-instance Oracle Real Applications Clusters (Oracle RAC) database
- A multi-instance standby database for the Oracle RAC database

The Oracle RAC database and the associated standby database were enrolled using Atlanta B. Because Atlanta B and Atlanta C are in different cluster subgroups, the Oracle RAC database and associated standby database would have Atlanta B at the head of the endpoint node scan list and will prioritize connecting to Atlanta B over Atlanta C. Not shown is that each Oracle RAC instance would also be able to connect to Atlanta C.

There are two Oracle Key Vault servers, labeled Atlanta B and Atlanta C, presenting a read/write pair of Oracle Key Vault nodes. These nodes are connected by a bidirectional line indicating that these are read/write peers. read/write peer nodes are synchronous, which enables an update to one node to be replicated to the other node, and verified this on the other node, before the update is considered successful.

The bottom database instance was enrolled using the Atlanta C node. To illustrate this connection, the database is connected by an arrow to Oracle Key Vault Atlanta C. Because Atlanta B and Atlanta C are in different cluster subgroups, for this database, Atlanta C would be at the head of the endpoint node scan list, meaning that the database would preferentially connect to Atlanta C.

In the event that either Oracle Key Vault node is offline (for example, for maintenance), then all endpoints will automatically connect to the other available Oracle Key Vault node.

Mid-Size Cluster Across Two Data Centers Deployment

A two-data center configuration provides high availability, disaster recovery, and load distribution.

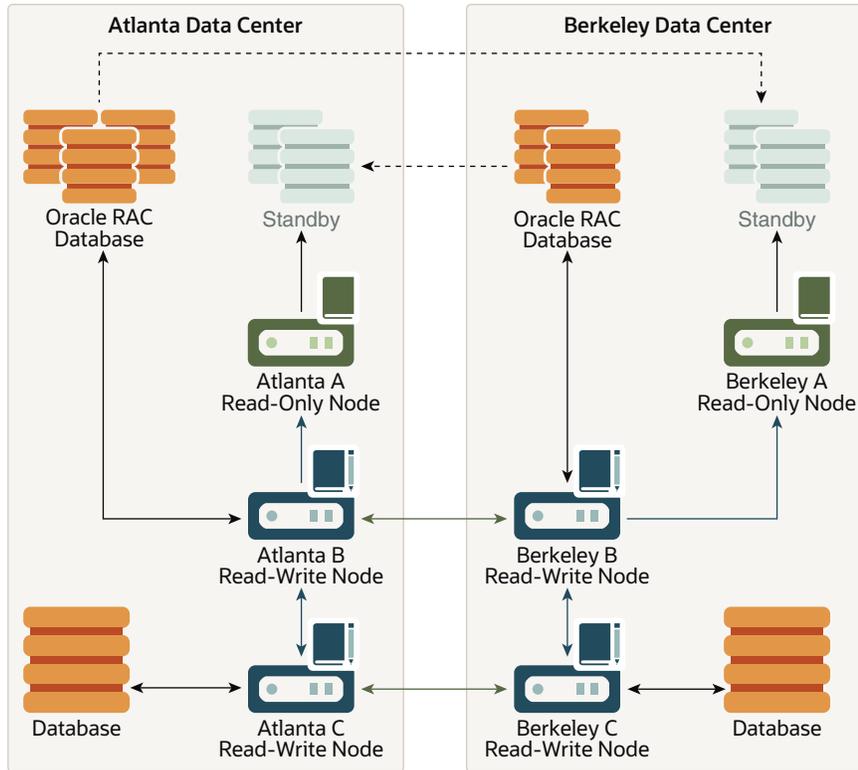
At least two [read/write pairs](#) are required. A read-write pair is only created when you pair a new node with a read-only node as its read-write peer. As a best practice, you could configure the peers in different data centers if you are concerned about disaster recovery, or you could put the [read/write peers](#) in the same data center if you are concerned about network latency or network interruptions. Cluster nodes in the same data center should be part of the same cluster subgroup. You should ensure that all endpoints in a data center are in the same cluster subgroup as the nodes in that data center. This ensures that the nodes within a given data center are at the head of the endpoint node scan list for endpoints in the same data center.

For a large deployment, Oracle recommends that you have a minimum of four Oracle Key Vault servers in a data center for high availability. This enables additional servers to be available for key updates if one of the servers fails. When you register the database endpoints, balance these endpoints across the Oracle Key Vault servers. For example, if the data center has 1000 database endpoints to register, and you have Oracle Key Vault four servers to accommodate them, then enroll 250 endpoints with each of the four servers.

Each endpoint first contacts the Oracle Key Vault nodes in the local data center. If an outage causes all Oracle Key Vault nodes to be unavailable in one data center, then as long as connectivity to another data center is available, the endpoint node scan list will redirect the endpoints to available Oracle Key Vault nodes in another data center.

A possible deployment scenario with two data centers, each containing two read-write nodes, paired with read-write nodes in the other data center is shown in [Figure 3-2](#). A data center can also host one or more [read-only nodes](#) as needed for load balancing, reliability, or expansion purposes. In the scenario described in the following figure, each data center hosts a single read-only node.

Figure 3-2 Oracle Key Vault Multi-Master Cluster Deployment across Two Data Centers



In this scenario, both the Atlanta Data Center and the Berkeley Data Center each hosts three databases, as follows:

- A single instance database
- A multi-instance Oracle RAC database
- A multi-instance standby database for the Oracle RAC database

The dotted lines connecting the Oracle RAC databases to the standby databases represent database transactions. Note that this data is unidirectional, going from the database to the standby only. The Atlanta Data Center Oracle RAC Database sends data to the Berkeley Standby database, and the Berkeley Oracle RAC Database sends data to the Atlanta Standby database.

Atlanta Data Center and Berkeley Data Center each have three Oracle Key Vault nodes, with two being read-write and one being read-only, and all in the same cluster. These nodes are configured as follows:

- Atlanta A Read-Only Node and Berkeley A Read-Only Node are read-only nodes, in which critical data additions or updates are unidirectional, going from the read-write node to the read-only node.
- Atlanta B Read-Write Node is a read-write peer with Berkeley B Read-Write Node. The connection between these two nodes is bidirectional and enables them to be in sync at all times.
- Atlanta C Read-Write Node is a read-write peer with Berkeley C Read-Write Node. The relationship between these two nodes operates in the same way as the relationship between the Atlanta B and Berkeley B nodes.

All of the nodes replicate data to and receive replicated data from all other nodes. To maintain legibility, only some of these connections are shown, specifically:

- The read-write pair connection between Atlanta B Read-Write Node and Berkeley B Read-Write Node across the two data centers, in which the data flow is bidirectional
- The read-write pair connection between Atlanta C Read-Write Node and Berkeley C Read-Write Node across the two data centers, in which the data flow is bidirectional
- The regular connection between Atlanta B Read-Write Node and Atlanta C Read-Write Node in the Atlanta Data Center, in which the data flow is bidirectional
- The regular connection between Berkeley B Read-Write Node and Berkeley C Read-Write Node in the Berkeley Data Center, in which the data flow is bidirectional
- The regular connection between Atlanta B Read-Write Node to Atlanta A Read-Only Node in the Atlanta Data Center, in which the critical data flow is unidirectional from the read-write node to the read-only node
- The regular connection between Berkeley B Read-Write Node and Berkeley A Read-Only Node in the Berkeley Data Center, in which the critical data flow is unidirectional from the read-write node to the read-only node

Nodes in the same data center have been assigned the same cluster subgroups. For example, nodes in the Atlanta Data Center have been given the cluster subgroup Atlanta, and nodes in the Berkeley Data Center have been given the cluster subgroup Berkeley. Endpoints that are in the Atlanta Data Center are enrolled using one of Oracle Key Vault nodes that are also in the Atlanta Data Center, so that the endpoints preferentially connect to the nodes within the same data center. The same is true, respectively, for the endpoints in the Berkeley Data Center. In [Figure 3-2](#), the following connections are shown, which imply the first entry in each client endpoint node scan list. Note that if an endpoint is in the same cluster subgroup as at least one node in the cluster, the first entry in the endpoint node scan list will be randomly chosen from the nodes that are in the same cluster subgroup as the endpoint:

- In Atlanta Data Center:
 - The Oracle RAC Database connects to Atlanta B Read-Write Node, with the data going in a bidirectional flow.
 - The Atlanta A Read-Only Node connect to the Standby, with the data going from the read-only node to the standby in a unidirectional flow.
 - The Database connects to the Atlanta C Read-Write Node, with the data going in a bidirectional flow.
- In Berkeley Data Center:
 - The Oracle RAC Database connects to Berkeley B Read-Write Node, with the data going in a bidirectional flow.
 - The Berkeley A Read-Only Node connects to the Standby, with the data going from the read-only node to the standby in a unidirectional flow.
 - The Database connects to Berkeley C Read-Write Node, with the data going in a bidirectional flow.

In the event that Atlanta C Read-Write node cannot be reached or does not have the necessary key, the database that connected to it will connect to other Oracle Key Vault nodes to fetch the key.

Related Topics

- [Oracle Key Vault Read/Write Nodes](#)
A read/write node is a node in which critical data can be added or updated using the Oracle Key Vault or endpoint software.

- [Oracle Key Vault Read-Only Nodes](#)
In a read-only node, users can add or update non-critical data but not add or update critical data. However, read-only nodes can read critical data.

Multi-Master Cluster Features

Oracle Key Vault provides features that help with inconsistency resolution and name conflict resolution in clusters, and endpoint node scan lists.

- [Cluster Inconsistency Resolution in a Multi-Master Cluster](#)
Network outages can introduce data inconsistency in a cluster, but when the outage is over and the network connection resumes within the Maximum Disable Node Duration (default 24 hours), the data becomes consistent again.
- [Name Conflict Resolution in a Multi-Master Cluster](#)
Naming conflicts can arise when an object has the same name as another object in a different node.
- [Endpoint Node Connection Lists \(Endpoint Node Scan Lists\)](#)
An endpoint node scan list is a list of nodes to which the endpoint can connect.

Cluster Inconsistency Resolution in a Multi-Master Cluster

Network outages can introduce data inconsistency in a cluster, but when the outage is over and the network connection resumes within the Maximum Disable Node Duration (default 24 hours), the data becomes consistent again.

A node can be disconnected from other nodes in the cluster voluntarily or involuntarily. When a voluntarily or involuntarily disconnected node returns to the cluster within the Maximum Disabled Node Duration, any data changes in the cluster are replicated to the node. Network disruptions, power outages, and other disconnects can happen any time for any Oracle Key Vault node, causing an involuntary disconnection from other nodes in the cluster. Such failures interrupt the data replication processes within a multi-master cluster. Temporary failures do not always introduce inconsistency to a cluster. As soon as the problem is addressed, the data replication process will resume from the moment it was halted. This ensures that even after some disconnections, disconnected Oracle Key Vault nodes will be able to synchronize themselves with the other nodes in the cluster eventually.

Any change made in a read-write node is guaranteed to be replicated to the other paired read-write node. Therefore, even if the read-write node suffers a failure, the data is available on at least one other node in the cluster.

Name Conflict Resolution in a Multi-Master Cluster

Naming conflicts can arise when an object has the same name as another object in a different node.

Users must specify names when creating virtual wallets, users, user groups, endpoints, and endpoint groups. A name conflict arises when two or more users create the same type of object with the same name on different nodes before the object has been replicated. If the object has been replicated on other nodes, then the system prevents the creation of objects with the same name. But replication in the Oracle Key Vault cluster is not instantaneous, so there is a possibility that during the replication window (which can be in the order of seconds), another object with the same name may have been created in this cluster. If this happens, it becomes a name conflict. Name conflicts have obvious drawbacks. For example, the system cannot distinguish between the references to two objects with same name. Uniqueness in

names is thus enforced to avoid inconsistencies in the cluster. All other object names must be unique within their object type, such as wallets, endpoint groups, user groups, and any other object type. For example, no two wallets may have the same name within the cluster. User names and endpoint names must not conflict.

While rare, a naming conflict can still arise. When this occurs, Oracle Key Vault detects this name conflict and raises an alert. Oracle Key Vault then will append `_OKVxx` (where `xx` is the node ID of the node on which the renamed object was created) to the name of the conflicting object that was created later. You can choose to accept this suggested object name or rename the object.

To accept or change a conflicting object name, click the **Cluster** tab, then **Conflict Resolution** from the left navigation bar to see and resolve all conflicts.

Endpoint Node Connection Lists (Endpoint Node Scan Lists)

An endpoint node scan list is a list of nodes to which the endpoint can connect.

An [endpoint](#) connects to an Oracle Key Vault server or node to manage or access wallets, keys, certificates, and credentials.

In a standalone situation the endpoint node scan list has one entry. In a primary-standby configuration, the endpoint can connect to one of two servers.

In an Oracle Key Vault multi-master cluster, the endpoint node scan list is the list of all the nodes in the cluster. There is a [read-only node](#) list and [read/write node](#) list. Node subgroup assignments and node modes influence the order of nodes in the endpoint node scan list. The list is made available to the endpoint at the time of endpoint enrollment. The list is maintained automatically to reflect the available nodes in the cluster. This list tracks changes to the cluster and makes them available to the endpoints. The following events will trigger a change to the endpoint node scan list:

- A change of cluster size, for example due to node addition or node removal
- A change to the mode of the node, for example when a node in read-only restricted mode changes to read-write mode
- An hour has passed since the last endpoint update
- A change to an endpoint's or a node's assigned cluster subgroup

The endpoint gets the updated scan list along with the next successful, non-empty response. Once the scan list is sent by the node, it marks the scan list as sent to the endpoint. It is possible that a scan list sent to the endpoint and marked sent in the node, may not be applied at the endpoint. As such the cluster periodically sends the scan list to the endpoint even if there are no changes to the cluster nodes or the modes of any of the cluster nodes.

Cluster Management Information

The Cluster Management page provides a concise overview of the cluster and the status of each node.

If you are logged in as an Oracle Key Vault system administrator then you can also manage the cluster from the Cluster Details section. When a node is performing a cluster operation it becomes the [controller node](#).

Be aware that because the replication across the cluster takes time, there may be a delay before the Cluster Management page refreshes with the new cluster status. The [replication lag](#) in the monitoring page will help estimate the delay.

To view the Cluster Management page, click the **Cluster** tab, and then **Management** from the left navigation bar.

The screenshot displays the Oracle Key Vault Cluster Management interface. It is divided into three main sections:

- Cluster Information:** Shows details for cluster 'okvcluster1', including its subgroups ('okvlgp1'), a maximum disable node duration of 24 hours, and version '21.10.0.0.0'.
- Current Node Information:** Shows details for node 'okv02007040e3c', which is a 'Read-Write' node belonging to subgroup 'okvlgp1'.
- Cluster Details:** A table listing all nodes in the cluster with columns for selection, node ID, name, IP address, mode, status, read-write peer, cluster subgroup, join date, disable date, install date, and version.

Select Node	Node ID	Name	IP Address	Mode	Status	Read-Write Peer	Cluster Subgroup	Join Date	Disable Date	Install Date	Version
<input type="radio"/>	1	okv02007040e3c	100.70.126.142	Read-Write	ACTIVE	okvnode145	okvlgp1	09-OCT-2024 16:45:23	-	05-10-2024 16:01:33	21.10.0.0.0
<input type="radio"/>	2	okvnode145	100.70.126.148	Read-Write	ACTIVE	okv02007040e3c	okvlgp1	09-OCT-2024 16:50:01	-	05-10-2024 09:35:14	21.10.0.0.0
<input type="radio"/>	3	okv145	100.70.126.141	Read-Only Restricted	DELETING	-	okvlgp1	21-OCT-2024 11:45:59	-	-	21.10.0.0.0

Cluster Information

- **Cluster Name:** The name of the cluster.
- **Cluster Subgroups:** All subgroups within the cluster.
- **Maximum Disable Node Duration:** The maximum time, in hours, that a node can be disabled before it can no longer be enabled.
- **Cluster Version:** The version of Oracle Key Vault in which the cluster is operating.

Current Node Information

- **Node Name:** The name of this node.
- **Node Type:** The type of node, such as whether it is read-only or read/write.
- **Cluster Subgroup:** The subgroup to which this node belongs.

Cluster Details

- **Select Node:** Used to select a node for a specific operation, such as delete, force delete, or disable.
- **Node ID:** The ID of the node.
- **Node Name:** The name of the node. Clicking the node name takes you to the Cluster Management page of that node.
- **IP Address:** The IP address of the node.
- **Mode:** The mode in which the node is operating, such as read/write or read-only restricted.
- **Status:** The status of the node, such as active, pairing, disabling, disabled, enabling, deleting, or deleted.
- **Read-Write Peer:** The read/write peer of the node. If blank, it has no read/write peer.
- **Cluster Subgroup:** The subgroup to which the node belongs. When logged in as a user with the System Administrator role, you can change this by 1) checking the check box next to a node, 2) clicking the **Edit** button, which displays a window, 3) entering a new cluster subgroup in the field, and 4) clicking **Save**.
- **Join Date:** The date and time that the node was added to the cluster or most recently enabled
- **Disable Date:** The date and time that the node was disabled.
- **Install Date:** The date and time that the node was installed.
- **Node Version:** The current version of the Oracle Key Vault node.

4

Managing Oracle Key Vault Multi-Master Clusters

You can create, configure, manage, and administer an Oracle Key Vault multi-master cluster by using the Oracle Key Vault management console.

- [About Managing Oracle Key Vault Multi-Master Clusters](#)
You can add or remove nodes from the cluster, disable or enable cluster nodes, and manage activities such as node conflicts and replication.
- [Setting Up a Cluster](#)
After you convert the first stand-alone Oracle Key Vault to the initial node, you can add more nodes, thereby creating read-only nodes, or pairs of read/write nodes.
- [Terminating the Pairing of a Node](#)
On the controller node, you can terminate the pairing process for a new node.
- [Disabling a Cluster Node](#)
You can temporarily disable a cluster node, which is required for upgrades and maintenance.
- [Enabling a Disabled Cluster Node](#)
You can enable any cluster node that was previously disabled. You must perform this operation from the disabled node.
- [Deleting a Cluster Node](#)
You can permanently delete a node from the cluster.
- [Force Deleting a Cluster Node](#)
You can permanently force delete a node from a cluster that is dead, unresponsive, or has exceeded the maximum disabled node time limit.
- [Managing Replication Between Nodes](#)
You can enable and disable node replication from the Oracle Key Vault management console.
- [Cluster Management Information](#)
The Cluster Management page provides a concise overview of the cluster and the status of each node.
- [Cluster Monitoring Information](#)
The Cluster Monitoring page provides the replication health of the cluster and the current node.
- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.
- [Multi-Master Cluster Deployment Recommendations](#)
Oracle provides deployment recommendations for deployments that have two or more nodes.
- [Adding an Alternate Name or IP Address](#)
You can configure two alternate hostnames for a given Oracle Key Vault server or node.

About Managing Oracle Key Vault Multi-Master Clusters

You can add or remove nodes from the cluster, disable or enable cluster nodes, and manage activities such as node conflicts and replication.

Setting Up a Cluster

After you convert the first stand-alone Oracle Key Vault to the initial node, you can add more nodes, thereby creating read-only nodes, or pairs of read/write nodes.

- [About Setting Up a Cluster](#)
You create a multi-master cluster by converting a single Oracle Key Vault server to become the initial node.
- [Creating the First \(Initial\) Node of a Cluster](#)
To create a cluster, you must convert an existing standalone Oracle Key Vault server to become the first node in the cluster.
- [Adding Nodes to a Cluster](#)
After you have converted the initial node to a candidate node, you can start building your cluster. Ensure that the candidate node configuration settings, for example, FIPS settings match the cluster settings as closely as possible.

About Setting Up a Cluster

You create a multi-master cluster by converting a single Oracle Key Vault server to become the initial node.

This Oracle Key Vault server seeds the cluster data and converts the server into the first cluster node, which is called the initial node.

After the Oracle Key Vault server has been converted to the initial node of the multi-master cluster, you can add the different types of nodes that you need to the cluster. The cluster is expanded when you induct additional Oracle Key Vault servers, and add them as read/write nodes, or as simple read-only nodes.

Creating the First (Initial) Node of a Cluster

To create a cluster, you must convert an existing standalone Oracle Key Vault server to become the first node in the cluster.

This first node is called the initial node. The standalone Oracle Key Vault server can be a server that has been upgraded from an earlier release or it can be the primary server that was unpaired from a primary-standby configuration. Check *Oracle Key Vault Release Notes* for known issues about unpair operations and upgrades.

You can use this node to add one or more nodes to the cluster. The node operates in read-only restricted mode until it is part of a read/write pair.

Perform a full backup of the Oracle Key Vault server to a remote destination.

Use the Oracle Key Vault RESTful API to convert the Oracle Key Vault server to the initial node of the cluster:

- Run the `okv cluster node create` RESTful command to convert the Oracle Key Vault server to the initial node of the cluster

```
$ ./bin/okv cluster node create --cluster-name OCEAN11 --cluster-subgroup
WEST_COAST --node-name OKV01
{
  "result" : "Success",
  "value" : {
    "requestId" : "1518"
  }
}
```

Use the Oracle Key Vault Management Console to convert the Oracle Key Vault server to the initial node of the cluster:

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. If the Oracle Key Vault server was upgraded from a release earlier than Oracle Key Vault release 12.2 (bundle patch 8), then generate and activate (rotate) a new certificate for the node.

If this step does not apply to your deployment, then you can bypass it.

3. Select the **Cluster** tab, and then select **Configure** from the left navigation bar.
4. In the Configure as Candidate Node page, enter the following settings:
 - **First Node of Cluster:** Select the **Yes** button.
 - **Node Name:** Enter a unique name for this node. This field is auto-populated with the Oracle Key Vault server name as the node name. You may specify another name. You cannot change this name later.
 - **Cluster Name:** Enter a name for this cluster of nodes. You cannot change this name later.
 - **Cluster Subgroup:** Enter a name for this sub-group of nodes, such as a data center name or a logical group name.
5. Click **Convert to Candidate Node**.

The conversion takes several minutes. After the conversion is complete, the Cluster Management page is displayed and the node is now operating in read-only restricted mode. At this stage, you can add another node (a newly installed system at the same Oracle Key Vault version as the initial node) to start building this cluster. In that case, the initial node will become the controller node and the new node will be the candidate node. The data on the controller node will be replicated to the candidate node.

Related Topics

- [Rotating CA Certificate](#)
Use the Oracle Key Vault management console to rotate CA certificate and enable either a self-signed root CA certificate or an intermediate CA certificate.
- [Managing Service Certificates](#)
This chapter explains about Oracle Key Vault-generated certificates. You will also learn how to manage self-signed and third-party certificates.
- [Oracle Key Vault Release Notes](#)
- [okv cluster node create Command](#)

Adding Nodes to a Cluster

After you have converted the initial node to a candidate node, you can start building your cluster. Ensure that the candidate node configuration settings, for example, FIPS settings match the cluster settings as closely as possible.

- [Adding a Node to Create a Read/Write Pair](#)
After you create the initial node, you must add an additional read/write peer to the cluster.
- [Adding a Node as a Read-Only Node](#)
To add a new read-only cluster node, you add a newly configured server from any existing cluster node.
- [Creating an Additional Read/Write Pair in a Cluster](#)
Any node can be read/write paired with only one other node, and there can be multiple read/write pairs in a cluster.

Adding a Node to Create a Read/Write Pair

After you create the initial node, you must add an additional read/write peer to the cluster.

You can configure any two nodes as a read/write pair. However, any single node can be read/write paired with only one other node.

To create a read/write pair using two nodes in a cluster, you pair a node (referred to as the controller node) with a newly configured server (referred to as the candidate node). Note that this will take some time: an hour or more, depending on the speed of your server, network, and volume of data in the cluster. Be aware that the controller node will be unable to service endpoints during certain parts of this operation. It is also recommended to configure the candidate node system settings, such as, FIPS, NTP, and Root of Trust similar to the controller node before initiating the pairing.

1. Perform a full backup of the controller node to a remote destination before continuing.
2. Ensure that the following network requirements are in place:
 - There is network connectivity between the servers that host the controller node and the candidate node.
 - The ports that are required for Oracle Key Vault are open in the network firewall. These ports are described in Oracle Key Vault Installation and Upgrade Guide. Starting with Oracle Key Vault release 21.9, if the controller and candidate node are unable to communicate with each other over these ports, the node addition process may fail with an error similar to:

```
Operation not allowed, network port is closed
```

You may then be required to abort the pairing on the candidate node, open the required ports between both systems, and attempt the node addition again.

3. As a user who has the System Administrator role, log in to the Oracle Key Vault management console of the node that you want to use as a controller node.

You can use any existing node, including the first node, that does not have a read/write peer to be the controller node for this operation. If necessary, add a read-only node first.

4. Select the **Cluster** tab, and then select **Management** from the left navigation bar.
5. Under Cluster Details, select **Add**.

6. In the Add Candidate Node to Cluster page, under Add Cluster Details, enter the recovery passphrase in the **Recovery Passphrase of the Cluster** field.

This value will be used later when you pair with the candidate node. The recovery passphrase is that of the first node of the cluster and will be used later across all the cluster nodes.

7. Enter the following details under Add Candidate Node Details.

While you enter these details, do not save any of this information or click the **Add Node** button until you reach Step 14.

- **Add Node as Read-Write Peer:** Select **Yes**.
- **Node ID:** Select a unique ID for the candidate node. Remember that after you create this ID, you cannot change it. **Node ID** is auto populated but you may change it. Ensure that the candidate node ID is unique in the cluster.
- **Node Name:** Enter a unique name of the candidate node. After you create this name, you cannot change it.
- **Cluster Subgroup:** Enter the sub-group name for the candidate node. You can provide an existing subgroup name. If you provide a subgroup name that does not exist, it will be created. (This field is auto-populated with the cluster subgroup name of the controller node.)
- **Cluster Name** is auto-populated and cannot be changed.
- **IP Address:** Enter the IP address of the candidate node.
- **Certificate of Candidate Node:** The next steps explain how you can find the certificate of the candidate node.

Do not exit this page.

8. In a new browser window, log into the Oracle Key Vault management console of the candidate node as a user who has the System Administrator role.
9. Select the **Cluster** tab, and then select **Configure** from the left navigation bar.
10. In the Configure as Candidate Node page, enter the following details:

- **First Node of the Cluster:** Select **No**. Selecting **No** shows additional fields to enter.
- **Recovery Passphrase of the Cluster:** enter the recovery passphrase of the cluster that you entered earlier for the controller node.
- **IP Address:** enter the IP address of the controller node.
- **Certificate of Controller Node:** Use these steps to enter certificate of controller node.
 - In the browser window for the controller node, scroll to the bottom of the screen. Select and copy the entire certificate value shown for **Certificate of Controller Node**.
 - In the browser window for the candidate node, paste the copied certificate from the controller node into the **Certificate of the Controller Node** field.
- Check the recovery passphrase, the IP address, and the pasted in certificate very carefully to ensure that you copied it correctly. If there is an error, then after you click **Convert to Candidate Node**, you will need to terminate the pairing process or potentially reinstall Oracle Key Vault on this node.

11. Click **Convert to Candidate Node**.

The conversion can take several minutes. After the conversion is complete, the screen will refresh and the Adding Candidate Node to Cluster page is displayed. The certificate for the candidate node appears on this page.

12. Select and copy the entire candidate node certificate.
13. In the browser window of the controller node, paste the copied certificate from the candidate node into the **Certificate of Candidate Node** box.
14. Click **Add Node**.

This process will take an hour or more, depending on the speed of your server, network, and volume of data in the cluster. During this time, the network management interface of the Oracle Key Vault will be restarted and you might momentarily get a `Server Error 500` or the error `Bad Gateway` on the controller node. On the candidate node, errors may also appear, such as `Bad Gateway`. The candidate node will restart as part of the induction process. This is normal. During the pairing process, the status of the candidate node will display as `PAIRING` on all cluster nodes.

To view the status of any server, view the output on the management console.

After the candidate node restarts and completes the pairing process, you can log in to either cluster node to view the cluster status by selecting the **Cluster** tab. Ensure that the status of the new node shows as `ACTIVE` on all nodes in the cluster. The candidate node may briefly display that it is in read-only restricted mode after it automatically restarts. This node is now a synchronously paired cluster node and no longer a candidate node. After a node is part of a cluster, the console displays the node name, subgroup name, and cluster name in the top right area of the console header.

Adding a Node as a Read-Only Node

To add a new read-only cluster node, you add a newly configured server from any existing cluster node.

The existing cluster node is referred to as the controller node, and the newly configured server is referred to as the candidate node. This process will take an hour or more, depending on the speed of your server, network, and volume of data in the cluster.

1. Perform a server backup before continuing.
2. Ensure that the following network requirements are in place:
 - There is good network connectivity between the servers that host the controller node and the candidate node.
 - The ports that are required for Oracle Key Vault are open in the network firewall. These ports are described in *Oracle Key Vault Installation and Upgrade Guide*. Starting with Oracle Key Vault release 21.9, if the controller and candidate node are unable to communicate with each other over these ports, the node addition process may fail with an error similar to:

```
Operation not allowed, network port is closed
```

You may then be required to abort the pairing on the candidate node, open the required ports between both systems, and attempt the node addition again.

3. As a user who has the System Administrator role, log in to the Oracle Key Vault management console of the node that you want to use as a controller node.

You can use any existing node as a controller for this operation.

4. Select the **Cluster** tab, and then select **Management** from the left navigation bar.
5. In the Cluster Details section click **Add**.

6. In the Add Cluster Details section, enter the cluster recovery passphrase in the **Recovery Passphrase of the Cluster** field.

This value will be used later when pairing with the candidate node.

7. Under **Add Candidate Node Details**, enter the following information:
 - **Add Node as Read-Write Peer:** Select **No**.
 - **Node ID:** Select a unique ID for the candidate node. Remember that after you create this ID, you cannot change it. **Node ID** is auto populated but you may change it. Ensure that the candidate node ID is unique in the cluster.
 - **Node Name:** Enter a unique name of the candidate node. After you create this name, you cannot change it.
 - **Cluster Subgroup:** Enter the subgroup name for the candidate node. You can provide an existing subgroup name. If you provide a subgroup name that does not exist, then it will be created.
 - **Cluster Name:** This name is populated automatically.
 - **IP Address:** Enter the IP address of the candidate node.

Do not exit this page.

8. In a new browser window, log into the Oracle Key Vault management console of the candidate node as a user who has the System Administrator role.
9. Select the **Cluster** tab, and then select **Configure** from the left navigation bar.

The Configure as Cluster Candidate page appears.

10. In the Configure as Candidate Node page, enter the following details:
 - **First Node of the Cluster:** Select **No**. Selecting **No** shows additional fields to enter.
 - **Recovery Passphrase of the Cluster:** enter the recovery passphrase of the cluster that you entered earlier for the controller node.
 - **IP Address:** enter the IP address of the controller node.
 - **Certificate of Controller Node:** Use these steps to enter certificate of controller node.
 - In the browser window for the controller node, scroll to the bottom of the screen. Select and copy the entire certificate value shown for **Certificate of Controller Node**.
 - In the browser window for the candidate node, paste the copied certificate from the controller node into the **Certificate of the Controller Node** field.
 - Check the recovery passphrase, the IP address, and the pasted in certificate very carefully to ensure that you copied it correctly. If there is an error, after you click Convert to Candidate Node, you will need to terminate the pairing process or potentially reinstall Oracle Key Vault on this node.

11. Click **Convert to Candidate**.

The conversion can take several minutes. After the conversion is complete, the screen will refresh and the Adding Candidate Node to Cluster page is displayed. The certificate for the candidate node appears on this page.

12. Select and copy the entire candidate node certificate.
13. In the browser window of the controller node, paste the copied certificate from the candidate node into the **Certificate of Candidate Node** box.
14. Click **Add Node**.

This process will take an hour or more, depending on the speed of your server, network, and volume of data in the cluster. During this time, the Oracle Key Vault console of the controller node will become unresponsive and can display an error such as `Server Error 500`. On the candidate node, errors may also appear, such as `Bad Gateway`. The candidate node will restart as part of the synchronization process. This is normal. During the pairing process, the status of the candidate node will display as `PAIRING` on all other cluster nodes not involved in this pairing process.

To view the status of any server, view the output on the server console.

After the candidate node restarts and completes the pairing process, you can log into either cluster node to view the cluster status by selecting the **Cluster** tab. Ensure that the status of the new node shows as `ACTIVE` on all nodes in the cluster. The candidate node may briefly display that it is in read-only restricted mode after it automatically restarts. This node is now a read-only paired cluster node and no longer a candidate node. After a node is part of a cluster, the console will display the node name, sub-group name, and cluster name in the top right area of the console header.

Creating an Additional Read/Write Pair in a Cluster

Any node can be read/write paired with only one other node, and there can be multiple read/write pairs in a cluster.

1. Select a read-only cluster node as the controller node to pair with a new candidate node.
2. Follow the steps to create a read/write pair of nodes in a cluster.

Related Topics

- [Adding a Node to Create a Read/Write Pair](#)
After you create the initial node, you must add an additional read/write peer to the cluster.

Terminating the Pairing of a Node

On the controller node, you can terminate the pairing process for a new node.

Be aware that when the controller node performs this operation, then it will experience network changes that will temporarily prevent it from serving endpoints.

1. On the controller node, in the **Status** section of the Adding Candidate Node to Cluster page, click the **Abort** button.

Adding Candidate Node to Cluster

Controller Node Information

Node Name	node1
Cluster Subgroup	austin
Cluster Name	oracle

▶ Certificate of Controller Node

Status Abort

Activity Current node processing

Id	Stage	Status
1	Transport channel opened with the candidate node	✓
2	Verified the candidate node details	✓
3	Generated the controller node details	✓
4	Generated backup of the controller node for cloning	✓
5	Clone bundle sent to the candidate node	
6	Data replication (downstream mining configuration) to the candidate node enabled	
7	Data replication to other cluster nodes enabled	
8	The candidate node successfully joined the cluster	

▶ Details

- A dialog with the message **Are you sure you want to ABORT the addition of the new node?** will appear. Select **OK**.

After the pairing process terminates, you will be returned to the **Add Node to Cluster** page on the controller node.

If you terminate the pairing of a candidate node from the controller node, then depending on how far the candidate node is into the induction process, you may or may not be able to return the candidate node to a standalone state. You can attempt to terminate the pairing process from the candidate node. If you are unable to terminate the pairing process from the candidate node, then the candidate node is no longer usable in its current state, and you must re-image the candidate node.

Disabling a Cluster Node

You can temporarily disable a cluster node, which is required for upgrades and maintenance.

However, be aware that a node can only be disabled for a set period of time. When it exceeds that time, it cannot be enabled again. The default maximum disable node duration time is 24 hours, but you can set it for as high as 240 hours. Note that as this value is increased, the average amount of disk space consumed by cluster-related data also increases.

Do not attempt disabling a node unless replication between all nodes is healthy. Failure to do so will result in a node stuck in the `DISABLING` state. To return such a node back to the `ACTIVE` state, you can cancel the disable operation by clicking the **Cancel Disable** button.

1. Log into any cluster Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Cluster** tab, and then select **Management** from the left navigation bar.
3. Under Cluster Details, in the **Select Node** column, select the check box of the node to disable.
4. Click **Disable**.

On the node that is being disabled, the node status will display as `DISABLING` during the disabling process. The other nodes will display the status for this node as `DISABLED`. When the disabling process is complete, the node that you disabled displays the `DISABLED` status.

Related Topics

- [Configuring the Maximum Disable Node Duration for the Cluster](#)
You can set the maximum disable node duration time for the cluster in hours.

Enabling a Disabled Cluster Node

You can enable any cluster node that was previously disabled. You must perform this operation from the disabled node.

1. As a user who has the System Administrator role, log into the Oracle Key Vault management console of any active node in the cluster.
2. Select the **Cluster** tab, and then select **Management** from the left navigation bar.
3. In the Cluster Details section, note the dates in which the nodes have been disabled.

Oracle recommends that you enable the nodes in the reverse order in which they were disabled. Otherwise, the enabling action may not be able to complete.

4. In the Cluster Details section, under Name, click the name of the node that was disabled most recently.

Clicking the node name enables you to log in to the disabled node. You can only enable disabled nodes from the disabled node itself.

5. Select **Enable**.

You do not need to check the check box of the disabled node in the Cluster Details table.

6. Repeat this step for each disabled node, from the most recent to the node that was disabled first.

Deleting a Cluster Node

You can permanently delete a node from the cluster.

When you delete a node from the cluster, its read/write peer transitions into read-only restricted mode. Deleted nodes cannot be added back to any cluster, not just to the current cluster from which they were deleted. However, you can reinstall the Oracle Key Vault appliance software on this server and add the deleted node to a cluster. All data will be synchronized with the cluster before the node is deleted. A node cannot delete itself. Be aware that if a deleted node has a read/write peer, then this read/write peer node will experience network changes that will temporarily prevent it from serving endpoints.

Ensure to shutdown the node before deleting from the cluster.

1. As a user who has the System Administrator role, on a different node, log into the Oracle Key Vault management console.
A node cannot delete itself.
2. Select the **Cluster** tab, and then select **Management** from the left navigation bar.
3. Under Cluster Details, in the **Select Node** column, select the check box of the node to delete.
4. Click **Delete**.

The node status will display as `DELETING`. After it is deleted, it will show as `DELETED`, or later be removed from the cluster management page.

This action is immediate. The node status will display as `DELETING`. Do not shut down the deleted server until it no longer shows in the Cluster Details table on the Cluster Management page. However, Oracle recommends that you wait an hour after deleting a cluster node before reusing the node ID of the node that was deleted.

Force Deleting a Cluster Node

You can permanently force delete a node from a cluster that is dead, unresponsive, or has exceeded the maximum disabled node time limit.

Forcefully deleting a node that is still a part of a cluster may cause inconsistency in the cluster. Be aware that if the read/write peer of the node that was forcefully deleted is also removed from the cluster before confirming that all critical data from the forcefully deleted node has reached other nodes, then data loss can result. **When you forcefully delete a node, ensure that the node to be deleted has first been shut down.** When you force delete a node from the cluster, its read/write peer transitions into read-only restricted mode. A node cannot be deleted. A node cannot be deleted from its own management console. Deleted nodes cannot be added back to the cluster. However, a new Oracle Key Vault appliance can replace the deleted node. Be aware that if a deleted node has a read/write peer, then this read/write peer node will experience network changes that will temporarily prevent it from serving endpoints.

Note:

Ensure to shutdown the node before deleting from the cluster. Before force deleting a node, ensure that the communication among all other nodes is working and they can replicate data among themselves.

1. On a different node, log into the Oracle Key Vault management console as a user who has the System Administrator role.
A node cannot delete itself. Oracle recommends that if the node to be deleted has a read/write peer, to force delete the node from its read/write peer.
2. Select the **Cluster** tab, and then select **Management** from the left navigation bar.
3. Under Cluster Details, in the **Select Node** column, select the check box of the node to disable.
4. Click **Force Delete**.

The node status will display as `DELETING`. After it is deleted, it will show as `DELETED`, or later be removed from the cluster management page. Oracle recommends that you wait an hour after force deleting a cluster node before reusing the node ID of the node that was deleted.

Managing Replication Between Nodes

You can enable and disable node replication from the Oracle Key Vault management console.

- [Restarting Cluster Services](#)
While managing replication between nodes, you can restart a node's cluster services when the cluster service status for the node is down.
- [Disabling Node Replication](#)
You can disable the replication link between the current node and any other node in a cluster.
- [Enabling Node Replication](#)
You can enable the replication link between the current node and any other node in a cluster.

Restarting Cluster Services

While managing replication between nodes, you can restart a node's cluster services when the cluster service status for the node is down.

1. Log into Oracle Key Vault management console of any cluster node as a user who has the System Administrator role.
2. Select the **Cluster** tab, and then **Monitoring** from the left navigation bar.
3. In the Node State pane, click the **Restart Cluster Services** button.

Disabling Node Replication

You can disable the replication link between the current node and any other node in a cluster.

1. Log into Oracle Key Vault management console of any cluster node as a user who has the System Administrator role.
2. Select the **Cluster** tab, and then **Monitoring** from the left navigation bar.
3. Under Cluster Link State, select the check boxes for the nodes for which you want to disable replication.
4. Click **Disable**.
5. Click **OK** to confirm in the dialog box.

Enabling Node Replication

You can enable the replication link between the current node and any other node in a cluster.

1. As a user who has the System Administrator role, log in to the Oracle Key Vault management console of the node for which replication should be managed.
2. Select the **Cluster** tab, and then **Monitoring** from the left navigation bar.
3. Under Cluster Link State, select the check boxes for the nodes for which you want to enable replication.
4. Click **Enable**.
5. Click **OK** to confirm in the dialog box.

Cluster Management Information

The Cluster Management page provides a concise overview of the cluster and the status of each node.

If you are logged in as an Oracle Key Vault system administrator then you can also manage the cluster from the Cluster Details section. When a node is performing a cluster operation it becomes the **controller node**.

Be aware that because the replication across the cluster takes time, there may be a delay before the Cluster Management page refreshes with the new cluster status. The **replication lag** in the monitoring page will help estimate the delay.

To view the Cluster Management page, click the **Cluster** tab, and then **Management** from the left navigation bar.

The screenshot displays the Oracle Key Vault Cluster Management interface. It is divided into three main sections:

- Cluster Information:** Shows cluster details such as Cluster Name (okvcluster1), Cluster Subgroups (okvlgp1), Maximum Disable Node Duration (24 hrs), and Cluster Version (21.10.0.0.0).
- Current Node Information:** Shows details for the current node, including Node Name (okv02007040e3c), Node Type (Read-Write), and Cluster Subgroup (okvlgp1).
- Cluster Details:** A table listing all nodes in the cluster with columns for Select Node, Node ID, Name, IP Address, Mode, Status, Read-Write Peer, Cluster Subgroup, Join Date, Disable Date, Install Date, and Version.

Select Node	Node ID	Name	IP Address	Mode	Status	Read-Write Peer	Cluster Subgroup	Join Date	Disable Date	Install Date	Version
<input type="radio"/>	1	okv02007040e3c	100.70.026.142	Read-Write	ACTIVE	okvnode48	okvlgp1	09-OCT-2024 18:45:23	-	09-10-2024 19:01:33	21.10.0.0.0
<input type="radio"/>	2	okvnode48	100.70.026.143	Read-Write	ACTIVE	okv02007040e3c	okvlgp1	09-OCT-2024 18:50:01	-	09-10-2024 09:35:14	21.10.0.0.0
<input type="radio"/>	3	okv141	100.70.026.141	Read-Only Restricted	DELETING	-	okvlgp1	21-OCT-2024 11:48:59	-	-	21.10.0.0.0

Cluster Information

- **Cluster Name:** The name of the cluster.
- **Cluster Subgroups:** All subgroups within the cluster.
- **Maximum Disable Node Duration:** The maximum time, in hours, that a node can be disabled before it can no longer be enabled.
- **Cluster Version:** The version of Oracle Key Vault in which the cluster is operating.

Current Node Information

- **Node Name:** The name of this node.
- **Node Type:** The type of node, such as whether it is read-only or read/write.
- **Cluster Subgroup:** The subgroup to which this node belongs.

Cluster Details

- **Select Node:** Used to select a node for a specific operation, such as delete, force delete, or disable.
- **Node ID:** The ID of the node.
- **Node Name:** The name of the node. Clicking the node name takes you to the Cluster Management page of that node.
- **IP Address:** The IP address of the node.
- **Mode:** The mode in which the node is operating, such as read/write or read-only restricted.
- **Status:** The status of the node, such as active, pairing, disabling, disabled, enabling, deleting, or deleted.

- **Read-Write Peer:** The read/write peer of the node. If blank, it has no read/write peer.
- **Cluster Subgroup:** The subgroup to which the node belongs. When logged in as a user with the System Administrator role, you can change this by 1) checking the check box next to a node, 2) clicking the **Edit** button, which displays a window, 3) entering a new cluster subgroup in the field, and 4) clicking **Save**.
- **Join Date:** The date and time that the node was added to the cluster or most recently enabled
- **Disable Date:** The date and time that the node was disabled.
- **Install Date:** The date and time that the node was installed.
- **Node Version:** The current version of the Oracle Key Vault node.

Cluster Monitoring Information

The Cluster Monitoring page provides the replication health of the cluster and the current node.

This page also provides a concise overview of the settings enabled in the cluster. You cannot update the settings on this page. Because the replication across the cluster takes time, there may be a delay before the Cluster Monitoring page refreshes with the new cluster state. Replication lag will help estimate the delay.

To view the cluster monitoring page, click the **Cluster** tab, and then **Monitoring** from the left navigation bar.

You can hover the mouse over the checkmarks or X's in the Cluster Settings State pane. It will display one of the following explanations of the state:

- Enabled in Cluster
- Enabled in Node
- Disabled in Cluster
- Disabled in Node

Cluster Link State							Enable	Disable
<input type="checkbox"/>	Node ID	Name	State	Heartbeat Lag (in seconds)	Replication Lag (in seconds)			
<input type="checkbox"/>	2	node2	↑	94.77	63			

1 - 1 of 1

Cluster Settings State								
<input type="checkbox"/>	Node ID	Name	Audit	FIPS	HSM	SNMP	SYSLOG	DNS
	1	node1	✓	✗	✗	✗	✗	✓
	2	node2	✓	✗	✗	✗	✗	✓

1 - 2 of 2

Cluster Link State

- **Select Node:** Used to select nodes for a specific operation, such as enabling or disabling replication. You can click the checkbox on the label row to select all nodes.
- **Node ID:** The ID of the node.
- **Node Name:** The name of the node.
- **State:** The current state of the node. The server is either up or down.
- **Heartbeat Lag:** The amount of time since a heartbeat was last received from this node. This setting should be around 60 seconds or lower. If this value is consistently greater than 60 seconds, then it means that one or more replication links may be broken. After the replication problem is fixed, it will then trend back down towards 60 or lower.
- **Replication Lag:** The average time it takes for data to replicate from this node to the current node.
- **Enable:** Enables the replication between the current node and the node selected (available only when logged in as a user with the System Administrator role).
- **Disable:** Disables the replication between the current node and the node selected (available only when logged in as a user with the System Administrator role).

Cluster Settings State

- **Node ID:** The ID of the node.
- **Node Name:** The name of the node.
- **Audit:** Indicates if auditing is enabled or disabled.
- **FIPS:** Indicates if FIPS mode is enabled or disabled.
- **HSM:** Indicates if HSM integration is enabled or disabled.
- **SNMP:** Indicates if SNMP is enabled or disabled.
- **SYSLOG:** Indicates if syslog is enabled or disabled.
- **DNS:** Indicates if DNS is enabled or disabled.

Naming Conflicts and Resolution

Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

- [About Naming Conflicts and Resolution](#)
If you create an object that has the same name as another object on another node, Oracle Key Vault resolves this conflict.
- [Naming Conflict Resolution Information](#)
The Cluster Conflict Resolution page provides a list of objects with names that conflict with objects created on different nodes.
- [Changing the Suggested Conflict Resolution Name](#)
You can change the suggested name for an object that conflicts with another object of the same type.
- [Accepting the Suggested Conflict Resolution Name](#)
You can accept the suggested name for an object name that conflicts with another object of the same type.

About Naming Conflicts and Resolution

If you create an object that has the same name as another object on another node, Oracle Key Vault resolves this conflict.

You can create a new object with a name that conflicts with an object of the same type created on another node. If a conflict happens, then Oracle Key Vault will make the name of the conflicting object unique by adding `_OKVxx`, where `xx` is the node ID of the node on which the object was created. You can choose to accept this new name or change the object name.

Users who have the System Administrator role can resolve the following naming conflicts:

- User names
- Endpoint names

Users who have the Key Administrator role can resolve the following naming conflicts:

- Endpoint groups
- Security objects
- User groups
- Wallets

If an object is stuck in the `PENDING` state and will not transition to `ACTIVE`, then check for any broken replication links in the cluster. You can find cluster links in the Oracle Key Vault management console by selecting the **Cluster** tab and then selecting **Monitoring**.

Naming Conflict Resolution Information

The Cluster Conflict Resolution page provides a list of objects with names that conflict with objects created on different nodes.

On this page, you can accept the suggested unique name or edit the object name. To view the Cluster Conflict Resolution page, click the **Cluster** tab, and then **Conflict Resolution** from the left navigation bar. Alternatively, you can click the **Click here for details** button on a Naming Conflict alert from the Alerts table on the Home page.

<input type="checkbox"/>	Unique Name	Supplied Name	Name Status	Created By	Creator Node	Description	Rename
<input type="checkbox"/>	Example_Wallet_OKV02	Example_Wallet	ACTIVE	OKVADMIN	node2		

1 - 1 of 1

Wallet Name Conflicts

- **Unique Name:** The unique name assigned to the object by the system.
- **Supplied Name:** The original name of the object that conflicts with another object of this type.

- **Name Status:** The status of the object. The status can be `PENDING` or `ACTIVE`.
- **Created By:** The user that created the conflicting object name.
- **Creator Node:** The node on which the conflicting object was created.
- **Description:** The description of the object as entered by the user.
- **Rename:** The button that links to the object page where it can be renamed. When you click the edit icon, the Wallet Overview page appears. Click **Make Unique** to give the wallet a unique name, and then click **Accept Rename**.
- **Accept:** Allows you to accept the suggested name for the selected objects.

Changing the Suggested Conflict Resolution Name

You can change the suggested name for an object that conflicts with another object of the same type.

1. As a user who has the appropriate administrator role, log in to the Oracle Key Vault management console.
2. Select the **Cluster** tab, and then **Conflict Resolution** from the left navigation bar.
3. Locate the object that requires a name change.
4. Click the edit icon to the right of the object, under **Rename**.
5. On the object overview page, enter the new name for the object.
6. Click **Save**.

Accepting the Suggested Conflict Resolution Name

You can accept the suggested name for an object name that conflicts with another object of the same type.

1. As a user who has the appropriate administrator role, log in to the Oracle Key Vault management console.
2. Select the **Cluster** tab, and then **Conflict Resolution** from the left navigation bar.
3. Select the objects for which you want to accept the suggested name.
4. Click **Accept**.

Multi-Master Cluster Deployment Recommendations

Oracle provides deployment recommendations for deployments that have two or more nodes.

Two-Node Deployment Recommendations

Use a two-node deployments for the following situations:

- Non-critical environments, such as test and development
- Simple deployment of read/write pairs with both nodes active, replacing classic primary-standby
- Single data center environments

Considerations for a two-node deployment:

- Availability is provided by multiple nodes.

- Maintenance will require downtime.
- Good network connectivity between data centers is mandatory.
- Take regular backups to remote destinations for disaster recovery.

Three-Node Deployment Recommendations

Use a three-node deployment for the following situations:

- Single data center environments with minimal downtime requirement
- Single read/write pair with additional read-only node to handle load
- One read-only node is available for zero downtime during maintenance

Considerations for a three-node deployment:

- Take regular backups to remote destinations for disaster recovery.

Four or More Node Deployment Recommendations

Use a deployment of four or more nodes for the the following situations:

- Large data centers distributed across geographical locations
- Deployment of read/write pairs with pair members spanning geography

Considerations for a large deployment:

- Availability is provided by multiple nodes.
- Additional read-only nodes can be used to handle load.
- Good network connectivity between data centers is mandatory.

Adding an Alternate Name or IP Address

You can configure two alternate hostnames for a given Oracle Key Vault server or node.

For more information, see [Configuring Oracle Key Vault with an Alternate Hostname](#)

5

Deploying Oracle Key Vault on an Oracle Cloud Infrastructure VM Compute Instance

You can install Oracle Key Vault on an Oracle Cloud Infrastructure (OCI) VM compute instance from Oracle Cloud Marketplace.

- [About Deploying Oracle Key Vault on an Oracle Cloud Infrastructure Compute Instance](#)
Oracle Key Vault on Oracle Cloud Marketplace is the cloud-based version of Oracle Key Vault and provides flexible, continuous and scalable key management.
- [Benefits of Using Oracle Key Vault in Oracle Cloud Infrastructure](#)
Quick deployments and ease of use are among the benefits of using an Oracle Key Vault Oracle Cloud Infrastructure (OCI) compute instance.
- [Provisioning an Oracle Key Vault Compute Instance](#)
The provisioning process for an Oracle Key Vault compute instance entails launching the compute instance and performing post-launch and post-installation tasks.
- [General Management of an Oracle Key Vault Compute Instance](#)
You can perform many of the Oracle Key Vault compute instance general management tasks in the Oracle Key Vault management console.
- [Migrating Oracle Key Vault Deployments Between On-Premises and OCI](#)
You can migrate an Oracle Key Vault standalone, primary-standby or cluster deployment from an on-premises environment to OCI or back.
- [Creating Oracle Key Vault Image in Microsoft Azure](#)
Oracle Key Vault provides deployment and provisioning in Azure.
- [Creating Oracle Key Vault Image in Amazon AWS](#)
Oracle Key Vault provides deployment and provisioning in AWS.
- [Creating Oracle Key Vault Image in Google Cloud](#)
Oracle Key Vault provides deployment and provisioning in Google Cloud.
- [Oracle Key Vault on Oracle VM Virtual Box](#)
Starting with Oracle Key Vault release 21.6, a fresh installation of an Oracle Key Vault VM guest can be stored as a template, and the VM platform cloning capability can be used to clone Oracle Key Vault. You can form a Oracle Key Vault cluster using multiple cloned VM instances. By using the cloned template, the system administrator can significantly shorten the provisioning time, compared to performing a full installation of each individual cluster node.

About Deploying Oracle Key Vault on an Oracle Cloud Infrastructure Compute Instance

Oracle Key Vault on Oracle Cloud Marketplace is the cloud-based version of Oracle Key Vault and provides flexible, continuous and scalable key management.

Oracle Key Vault is quick and easy to launch on a VM compute instance of any shape or size in your OCI tenancy. This eliminates the need to procure hardware and drastically shortens the time to provision a fully functional Oracle Key Vault deployment. Oracle Key Vault deployed on

an OCI VM compute instance (referred to as an Oracle Key Vault compute instance) is private to your tenancy and is managed by you. After the launch, an Oracle Key Vault compute instance has the same look and feel as an on-premises Oracle Key Vault installation, with the same flexibility in configuration.

An Oracle Key Vault server that is deployed on Oracle Cloud Infrastructure (OCI) VM compute instance can operate in the following situations:

- A standalone environment
- Be paired with other nodes in OCI or on-premises to form a multi-master cluster

The Oracle Key Vault multi-master cluster nodes could be entirely in OCI forming a cloud-only Oracle Key Vault cluster or some of the nodes can exist on-premises, thus forming a hybrid Oracle Key Vault cluster. This flexible deployment provides scalability regardless of whether Oracle Key Vault nodes are deployed in on-premises or cloud environments.

The Oracle Key Vault compute instance deployment enables the use of Oracle Key Vault to manage the encryption keys of your OCI-based database deployments. This enables you to maintain control over your encryption keys in a cloud environment. You can have up to 16 Oracle Key Vault compute instances in a multi-master cluster, distributed across any of the Oracle Cloud regions, to provide key management services to your globally distributed, on-premises, hybrid, or cloud-only Oracle database deployments.

When you enroll endpoints with the Oracle Key Vault compute instance, you must ensure that they are in the same VCN as the Oracle Key Vault compute instance itself. The endpoints will communicate with the Oracle Key Vault compute instance using the private IP of the instance. You can optionally configure the Oracle Key Vault compute instance to have a public IP address that can be used to access the Oracle Key Vault management console.

You can configure Oracle Key Vault to allow endpoints to use this public IP address to communicate with it, by configuring the public IP as an alternate hostname. The Oracle Key Vault instance can also be configured with a fully-qualified domain name (FQDN) as an alternate hostname. Each Oracle Key Vault instance can have up to two alternate hostnames, and endpoints can communicate with the instance using one of them. You must choose which of the two should be used for endpoint communications. You must configure the network to ensure that connectivity exists between Oracle Key Vault compute instances, as well as between endpoints and the Oracle Key Vault compute instances.

Related Topics

- [Oracle Cloud Marketplace](#)
- [About Configuring Oracle Key Vault with an Alternate Hostname](#)
You can configure Oracle Key Vault with an alternate hostname, that is, a fully-qualified domain name (FQDN) or a secondary IP address.

Benefits of Using Oracle Key Vault in Oracle Cloud Infrastructure

Quick deployments and ease of use are among the benefits of using an Oracle Key Vault Oracle Cloud Infrastructure (OCI) compute instance.

- **Key management for OCI-based database environment:** The Oracle Key Vault compute instance deployment provides key management to your OCI-based database environments (ExaDB-D) as well as on-premises and hybrid database environments, including ExaDB-C@C and ADB-C@C). This enables you to own, manage, and maintain control over encryption keys of your database environments in the cloud.
- **Quick deployment:** You can launch the Oracle Key Vault compute instance within minutes, see [Launch Oracle Key Vault Compute Instance](#) and without the need to manage

hardware or set up virtual machines. After it is launched, the Oracle Key Vault compute instance can run stand-alone or be added to a multi-master cluster. You can enroll endpoints with an Oracle Key Vault compute instance. This way, you can quickly set up a production environment. You can also use Oracle Key Vault compute instances to quickly set up a test and development environment to validate and experiment with various use-cases and deployment scenarios of Oracle Key Vault.

- **Scaling out a production environment during peak load or hardware unavailability:** If you use FastConnect or IPSec VPN in OCI, then you can extend the Oracle Key Vault cloud deployments to an on-premises environment. Using FastConnect or IPSec VPN, you can pair Oracle Key Vault nodes on-premises with Oracle Key Vault compute instances in OCI to form a hybrid cluster. You can use a hybrid cluster to run production Oracle Key Vault servers in OCI, or use them to expand the Oracle Key Vault cluster temporarily. Oracle Key Vault compute instances can be added quickly as new nodes to an on-premises, OCI or hybrid Oracle Key Vault cluster. This type of deployment provides spontaneous elasticity to the Oracle Key Vault cluster, and can be used to address any temporary increase of load on nodes of the Oracle Key Vault cluster.
- **Reduced latency for hybrid database environments:** For use cases where the data is shared between on-premises and cloud databases, managing the keys in a hybrid Oracle Key Vault cluster provides for locality of reference. Because the keys are available on all nodes of the cluster, the cluster subgroups can be setup in such a way that the databases in the cloud can primarily fetch the keys from the cluster nodes in OCI and the on-premises databases can primarily fetch the keys from cluster nodes that are provisioned on-premises.
- **Simplified transition of on-premises to OCI-based Oracle Key Vault clusters:** If you are connected to OCI using FastConnect or IPSec VPN, then you can extend your on-premises Oracle Key Vault cluster by adding Oracle Key Vault compute instances to that cluster. The IP addresses of the Oracle Key Vault nodes in OCI are added to the scan lists of your database endpoints. Once you have the appropriate number of Oracle Key Vault nodes in your OCI tenancy, you can remove the on-premises Oracle Key Vault nodes from the cluster. Following the same procedure, it is possible to seamlessly transition from an Oracle Key Vault cluster in OCI back to an on-premises Oracle Key Vault cluster.
- **Engaging OCI infrastructure and services:** You can take advantage of the unique benefits of the Oracle Cloud Infrastructure. If you install multiple Oracle Key Vault compute instances in the same region, you can choose to deploy them in different availability domains (fault domains are selected automatically, but can be changed) to guarantee the highest possible availability of your key management service. Services such as DNS and NTP are also natively available in OCI. You do not have to set them up, thereby simplifying Oracle Key Vault provisioning.

Provisioning an Oracle Key Vault Compute Instance

The provisioning process for an Oracle Key Vault compute instance entails launching the compute instance and performing post-launch and post-installation tasks.

- [About Provisioning an Oracle Key Vault Compute Instance](#)
To provision the Oracle Key Vault compute instance, you choose an Oracle Key Vault image as your custom image.
- [Launching the Oracle Key Vault Compute Instance](#)
The launching process for the Oracle Key Vault compute instance should take roughly two to five minutes.

About Provisioning an Oracle Key Vault Compute Instance

To provision the Oracle Key Vault compute instance, you choose an Oracle Key Vault image as your custom image.

You will launch this image from the OCI Marketplace on a compute shape. After you complete the process, the Oracle Key Vault compute image becomes unique to your environment. The disk size of this image is 4 TB.

After you complete the launch, you can begin to use the Oracle Key Vault compute image immediately. The steps that you must perform after the launch are similar to the steps that you would perform for an on-premises Oracle Key Vault installation.

Launching the Oracle Key Vault Compute Instance

The launching process for the Oracle Key Vault compute instance should take roughly two to five minutes.

- [About Launching the Oracle Key Vault Compute Instance](#)
The launch process requires some minor preparation work on your system.
- [Step 1: Ensure That You Have Prerequisites in Place](#)
Before you can launch an Oracle Key Vault compute instance, you must ensure that you have prerequisites in place in the Oracle cloud.
- [Step 2: Find the Oracle Key Vault Image](#)
The Oracle Key Vault image is available on the Oracle Cloud Marketplace web site.
- [Step 3: Launch the Oracle Key Vault VM Compute Instance](#)
You should perform the entire launching process in the Oracle Cloud Marketplace.
- [Step 4: Perform Post-Launch and Post-Installation Tasks](#)
After you launch Oracle Key Vault in an OCI compute instance, you first perform the post-launch task, followed by post-installation tasks.

About Launching the Oracle Key Vault Compute Instance

The launch process requires some minor preparation work on your system.

Before you begin the launch process, ensure that the endpoints that you plan to use are in the same VCN as the Oracle Key Vault instance will be. The endpoints will communicate with Oracle Key Vault using the private IP of the compute instance. Optionally, the Oracle Key Vault compute instance can have a public IP that can be used to access the Oracle Key Vault management console. You can also optionally configure Oracle Key Vault to allow endpoints to use this public IP address (or an associated fully-qualified domain name) to communicate with it. You will also set up the network and configure it to ensure that network connectivity will exist between the endpoints and the OCI compute instances.

Related Topics

- [About Configuring Oracle Key Vault with an Alternate Hostname](#)
You can configure Oracle Key Vault with an alternate hostname, that is, a fully-qualified domain name (FQDN) or a secondary IP address.

Step 1: Ensure That You Have Prerequisites in Place

Before you can launch an Oracle Key Vault compute instance, you must ensure that you have prerequisites in place in the Oracle cloud.

Ensure that the following are in place:

- You have an Oracle cloud account.
- You have access to your assigned Oracle cloud tenant.
- You have enough Service Limits and Quotas to create new compute resources within Oracle cloud tenant.

Step 2: Find the Oracle Key Vault Image

The Oracle Key Vault image is available on the Oracle Cloud Marketplace web site.

1. Go to the Oracle Cloud Marketplace web site.
<https://cloudmarketplace.oracle.com/marketplace/oci>
2. Log into your OCI tenancy and click **Launch Instance**.
3. From within your OCI tenancy, do the following:
 - a. Click **All Applications**.
 - b. Enter `Key Vault` in the search bar to find the available Oracle Key Vault releases.
 - c. Select the release of Oracle Key Vault that you want from the menu.

Related Topics

- [Oracle Cloud Marketplace](#)

Step 3: Launch the Oracle Key Vault VM Compute Instance

You should perform the entire launching process in the Oracle Cloud Marketplace.

1. In the Oracle Key Vault page, select **Launch Instance**.
2. In the `NAME` field, replace the automatically generated instance name with something more meaningful for your deployments, for example `OKV01`, `OKV02`, and so on.

The VMStandard 2.2 shape has been pre-selected. Larger shapes are recommended for production deployments.

3. For the shape, select VM.Standard 2.2 or bigger. Then click Select **Shape**.

Next, you are ready to configure the network.

4. Select the **Virtual Cloud Network (VCN)**.
5. Select the subnet.
6. Optionally, assign a public IP address, only for access to the Web Interface of Oracle Key Vault.

All communication (including the RESTful services) between endpoints and Oracle Key Vault uses the private IP address. After all post-launch and post-installation steps are completed, you can optionally configure Oracle Key Vault to allow endpoints to communicate with it using its public IP address (or an associated fully-qualified domain name).

7. Click **Advanced Options**, and then choose the **Network** tab.

Here you can replace the default private address with another one. Both of these addresses must be within the range of your current subnet. In addition, you can change the host name to match your naming convention. Otherwise, the host name will be constructed from `okv|MAC-address-of-NIC`.

8. Upload your SSH public key.
9. In the **Boot Volume** area, do not select any settings.
10. Click **Create** to complete the instance creation.

In a moment, the Oracle Key Vault compute image starts and is made available as an Oracle Key Vault server.

At this stage, you must perform the post-launch and post-installation steps.

Related Topics

- [Oracle Key Vault Installation and Upgrade Guide](#)
- [About Configuring Oracle Key Vault with an Alternate Hostname](#)
You can configure Oracle Key Vault with an alternate hostname, that is, a fully-qualified domain name (FQDN) or a secondary IP address.
- [Deploying Oracle Key Vault in OCI](#)

Step 4: Perform Post-Launch and Post-Installation Tasks

After you launch Oracle Key Vault in an OCI compute instance, you first perform the post-launch task, followed by post-installation tasks.

The post-launch task is to set the passwords for the `root` and `support` users. After you set these passwords, you must perform the post-installation tasks, which are the same tasks that are required for an on-premises deployment. After you complete the post-installation tasks, you can start building your Oracle Key Vault cluster or leave Oracle Key Vault in stand-alone mode.

1. Set the passwords for the `root` and `support` users.

- a. In a command prompt, log in as the `opc` user.

```
ssh opc@Oracle_Key_Vault_OCI_IP_address
```

- b. Set the passwords for the `root` and `support` users

```
set_password
```

When prompted, enter and confirm the root password. After you successfully enter the root password, choose `yes` to also set the password of the `support` user. After both passwords have been set, the `opc` account is deleted and SSH into Oracle Key Vault is disabled.

Only during upgrades, or when directed by Oracle Support, you can temporarily enable SSH from the Oracle Key Vault management console. You can then use SSH to log into the Oracle Key Vault server as the `support` user using the same SSH public key as the `opc` user.

2. Perform the following post-installation tasks. For more information, see [Performing Post-Installation Tasks](#) .
 - Create the Oracle Key Vault administrator accounts and set the recovery passphrase.
 - Enter the NTP and DNS addresses, using one of the following choices:

- Use the NTP server address as 169.254.169.254 in Oracle Cloud Infrastructure. Leave the remaining NTP fields empty.

For the DNS settings, consult with your network team because there are multiple options depending how DNS is configured in your subnet and tenancy.

Related Topics

- [Oracle Key Vault Installation and Upgrade Guide](#)

General Management of an Oracle Key Vault Compute Instance

You can perform many of the Oracle Key Vault compute instance general management tasks in the Oracle Key Vault management console.

- [Starting, Restarting, or Stopping an Oracle Key Vault Compute Instance](#)
Depending on the action you need, you can use the Oracle Key Vault management console or the OCI console.
- [System Settings in an Oracle Key Vault Compute Instance](#)
Most system settings in an Oracle Key Vault compute instance are the same as an on-premises deployment, with a few exceptions.
- [Backup and Restore Operations for Oracle Key Vault Compute Instances](#)
You can back up and restore Oracle Key Vault data between OCI environments and on-premises environments.
- [Terminating an Oracle Key Vault Compute Instance](#)
You terminate an Oracle Key Vault compute instance from the OCI console.

Starting, Restarting, or Stopping an Oracle Key Vault Compute Instance

Depending on the action you need, you can use the Oracle Key Vault management console or the OCI console.

You can use the Oracle Key Vault management console or OCI console to restart and stop an Oracle Key Vault compute instance, but to start an already stopped instance, you must use the OCI console.

Select one of the following methods to restart or stop an Oracle Key Vault compute instance:

- From the Oracle Key Vault management console, you can restart or stop the Oracle Key Vault compute instance:
 1. Log into the Oracle Key Vault management console as a user with the System Administrator role.
 2. Select **System**, then **Status** from the left navigation bar.
 3. In the Status page, do one of the following:
 - To restart, click **Reboot**.
 - To stop, click **Power Off**.

 **Note:**

After powering off the Oracle Key Vault from management console, you need to stop the instance from OCI console too as the status on OCI console remain in running state.

- From the OCI console, you can start, restart, or stop the Oracle Key Vault compute instance:
 1. Open the navigation menu. Under Core Infrastructure, go to Compute and click **Instances**.
 2. Select the Oracle Key Vault compute instance that you want to stop or start.
 3. Click one of the following actions:
 - To start a stopped instance, click **Start**.
 - To gracefully shut down the instance by sending a shutdown command to the operating system, click **Stop**.
If the Oracle Key Vault compute instance takes a long time to shut down, it could be improperly stopped, resulting in data corruption. To avoid this, shut down the instance using the commands available in the operating system before you stop the instance using the console.
 - To gracefully restart the Oracle Key Vault compute instance by sending a shutdown command to the operating system, and then power the instance back on, click **Reboot**.

System Settings in an Oracle Key Vault Compute Instance

Most system settings in an Oracle Key Vault compute instance are the same as an on-premises deployment, with a few exceptions.

Settings for system features such as auditing, email, RESTful services, integration Oracle Key Vault with Oracle Audit Vault is the same in both on-premises and OCI deployments.

- You can configure an Oracle Key Vault host name in either the OCI console or in the Oracle Key Vault management console. However, remember that if you set the IP address of the host in the OCI console, later on, you cannot change it in either the OCI console or the Oracle Key Vault management console.
- The SSH tunnel (deprecated) settings are used when on-premises Oracle Key Vault clusters provide key management services to Oracle databases that are deployed in OCI. Do not establish an SSH tunnel in OCI-based Oracle Key Vault deployments.

Backup and Restore Operations for Oracle Key Vault Compute Instances

You can back up and restore Oracle Key Vault data between OCI environments and on-premises environments.

You can back up an Oracle Key Vault compute instance that is stored in an on-premises host: this is the same backup that will be restored. Another on-premises Oracle Key Vault server can be a backup location for a server that is being restored into an Oracle Key Vault compute instance.

Requirements are as follows:

- If you are performing a backup or restore operation from Oracle Key Vault compute instances to an OCI compute instance, then persistent network connectivity to the OCI compute instance from Oracle Key Vault compute instance must exist.
- If you want to perform a backup or restore operation between an Oracle Key Vault compute instance and an on-premises host, ensure that the VCN can span the on-premises hosts.

Terminating an Oracle Key Vault Compute Instance

You terminate an Oracle Key Vault compute instance from the OCI console.

When you terminate the compute instance, all data, including keys that protect endpoints, are permanently lost and cannot be recovered except from a backup. Even backups may not have the most recent keys. Terminating the instances can lead to loss of data for all endpoints. Exercise extreme caution before terminating an instance. Terminate the Oracle Key Vault compute instance only if you are sure that you have a copy of the keys in another, safe location or that you do not need them.

1. Log in to the OCI console.
2. Under Core Infrastructure, go to Compute, and then click **Instances**.
3. Select the name of the Oracle Key Vault compute instance that you want to remove.
4. Click **Terminate**, and then respond to the confirmation prompt.

Terminated instances temporarily remain in the list of instances with the status **Terminated**.

Migrating Oracle Key Vault Deployments Between On-Premises and OCI

You can migrate an Oracle Key Vault standalone, primary-standby or cluster deployment from an on-premises environment to OCI or back.

- [About Performing Migrations with Oracle Key Vault Compute Instance Data](#)
You can transition an Oracle Key Vault deployment from on-premises to OCI, and from OCI back to on-premises.
- [Migrating Oracle Key Vault Deployments into OCI Using Backup and Restore](#)
A user who has the System Administrator role can transition the Oracle Key Vault deployment from on-premises to OCI using backup and restore.
- [Migrating Oracle Key Vault Deployments Out of OCI Using Backup and Restore](#)
A user who has the System Administrator role can transition the Oracle Key Vault deployment from OCI to on-premises.

About Performing Migrations with Oracle Key Vault Compute Instance Data

You can transition an Oracle Key Vault deployment from on-premises to OCI, and from OCI back to on-premises.

You can quickly set up a production Oracle Key Vault deployment in OCI to address your immediate key management needs and then transition to the on-premises deployment. Alternately, Oracle Key Vault compute instances require little to no overhead of hardware and VM management. To eliminate this overhead, you may want to transition your on-premises Oracle Key Vault deployment to OCI.

You can use the Oracle Key Vault backup and restore features to migrate an Oracle Key Vault cluster from on-premises to OCI, and back. You can transition an on-premises Oracle Key Vault cluster deployment to OCI by adding Oracle Key Vault compute instances to the cluster and removing on-premises Oracle Key Vault nodes from the cluster. The cluster is fully transitioned to OCI when no on-premises Oracle Key Vault node is left in the cluster. Similarly, you can also transition an Oracle Key Vault cluster in OCI to on-premises.

Migrating Oracle Key Vault Deployments into OCI Using Backup and Restore

A user who has the System Administrator role can transition the Oracle Key Vault deployment from on-premises to OCI using backup and restore.

1. Log in to the on-premises Oracle Key Vault server as a user who has the System Administrator role.
2. Configure an OCI compute instance as the backup destination.
3. Back up the on-premises Oracle Key Vault server to an OCI compute instance.
4. Launch an Oracle Key Vault compute instance with same Oracle Key Vault version as the on-premises Oracle Key Vault server.
5. Log in to the Oracle Key Vault compute instance as a user who has the System Administrator role.
6. Restore the backup from the OCI compute instance to the newly installed Oracle Key Vault compute instance.
7. To set up an Oracle Key Vault multi-master cluster, convert the restored Oracle Key Vault compute instance as the first (initial) node of the cluster.
8. Configure additional Oracle Key Vault compute instances and add them to the cluster as needed.

Related Topics

- [Creating the First \(Initial\) Node of a Cluster](#)
To create a cluster, you must convert an existing standalone Oracle Key Vault server to become the first node in the cluster.
- [Setting Up a Cluster](#)
After you convert the first stand-alone Oracle Key Vault to the initial node, you can add more nodes, thereby creating read-only nodes, or pairs of read/write nodes.
- [Backup and Restore Operations](#)
Backups provide the ability to restore Oracle Key Vault to a previous state in the case of a failure.

Migrating Oracle Key Vault Deployments Out of OCI Using Backup and Restore

A user who has the System Administrator role can transition the Oracle Key Vault deployment from OCI to on-premises.

1. Log in to the Oracle Key Vault compute instance as a user who has the System Administrator role.
2. Back up the Oracle Key Vault compute instance to an on-premises system.

3. Install a new Oracle Key Vault server on-premises with same Oracle Key Vault version as the Oracle Key Vault compute instance.
4. Log in to the on-premise Oracle Key Vault server as a user who has the System Administrator role.
5. Restore the backup from the on-premises backup destination to the newly installed on-premises Oracle Key Vault server.
6. To set up an Oracle Key Vault multi-master cluster, convert the restored on-premises Oracle Key Vault server as the first (initial) node of the cluster.
7. Configure additional Oracle Key Vault compute instances and add them to the cluster as needed.

Related Topics

- [Creating the First \(Initial\) Node of a Cluster](#)
To create a cluster, you must convert an existing standalone Oracle Key Vault server to become the first node in the cluster.
- [Setting Up a Cluster](#)
After you convert the first stand-alone Oracle Key Vault to the initial node, you can add more nodes, thereby creating read-only nodes, or pairs of read/write nodes.
- [Backup and Restore Operations](#)
Backups provide the ability to restore Oracle Key Vault to a previous state in the case of a failure.

Creating Oracle Key Vault Image in Microsoft Azure

Oracle Key Vault provides deployment and provisioning in Azure.

- [About Provisioning Oracle Key Vault in Microsoft Azure](#)
You can provision Oracle Key Vault in Microsoft Azure.
- [Create an Oracle Key Vault Base Image for Microsoft Azure](#)
You can create an Oracle Key Vault cluster in Microsoft Azure by first creating a Base Image, and then creating Oracle Key Vault cluster nodes from it.
- [Launching an Oracle Key Vault Cluster Node \(Instance\) from the Base Image](#)
Perform the steps to launch an Oracle Key Vault instance.

About Provisioning Oracle Key Vault in Microsoft Azure

You can provision Oracle Key Vault in Microsoft Azure.

To provide familiar, continuously available, extremely scalable, and fault-tolerant key management for your Oracle databases in Azure (including ExaDB-D@Azure), you can install and create an Oracle Key Vault multi-master cluster in Microsoft Azure, or extend on-premises Oracle Key Vault deployments with Oracle Key Vault cluster nodes in Microsoft Azure. You can also move an on-premises Oracle Key Vault cluster to Microsoft Azure by removing the on-premises nodes from the cluster.

Create an Oracle Key Vault Base Image for Microsoft Azure

You can create an Oracle Key Vault cluster in Microsoft Azure by first creating a Base Image, and then creating Oracle Key Vault cluster nodes from it.

- Ensure that you have installed and configured Azure CLI.

- Ensure that you have setup a container under **Storage Accounts** to store the VM disk used for preparing the Oracle Key Vault image for Azure.
- 1. On a separate machine outside, install Oracle VM VirtualBox.
- 2. Create a virtual machine on Oracle VM VirtualBox, with **Linux 8, Oracle (64 bit)** as the Operating System.
 - a. The minimum RAM for the Base Image is **8 GB**.
 - b. The minimum disk size for production deployments is **2 TB (recommended 4 TB)**; ensure to create the virtual hard disk as **Fixed size** and in the **VHD** format.
- 3. Create the empty VM shell with the settings from previous command (OL8, RAM, disk size). Do not install anything at this point.
- 4. Mount the `installer.iso` image to the virtual machine and boot the virtual machine.

 **Note:**

Do not go to the post-install steps; instead, run the script that makes this Oracle Key Vault a **template** to clone.

5. Install an Oracle Key Vault instance on the virtual machine. Do not complete any post installation steps.
6. Log in as `root` on the virtual machine console.
7. Reattach the Oracle Key Vault installation `iso` image to the first drive of the virtual machine before running the script in the next step.
8. Run the following command:

```
/usr/local/okv/bin/okv_export_cloud_image azure
```

This script prepares the system to be exported as a base image that will be later uploaded to Microsoft Azure.

9. Shutdown the virtual machine.
10. Upload the virtual machine's disk file to a container under storage accounts using azure CLI:

```
az storage blob upload --account-name <storage_account_name> --container-name <container_name> --name <blob_name>.vhd --file <VM_DISK_FILE>.vhd
```

11. Create Image from the **VHD** file using these steps:
 - a. Log in to Azure portal and navigate to Images.
 - b. Click **Create**.
 - c. On the **Create an image** page, complete the information as applicable. The required options include:
 - **Storage blob:** Specify `blob_name.vhd` from the storage accounts container uploaded before this step.
 - **OS Type:** Linux
 - **VM generation:** Select **Gen1 (BIOS)** or **Gen2 (UEFI)** to match your VirtualBox virtual machine setting.

- d. Click **Create + Review**. Review the details of the image before it is created.
 - e. Click **Create**.
- Once the deployment is complete, the image is ready for use.

Launching an Oracle Key Vault Cluster Node (Instance) from the Base Image

Perform the steps to launch an Oracle Key Vault instance.

1. Login to Azure portal and navigate to **Images**
2. Click the Oracle Key Vault image.
Consider these settings for the options.
 - **Size** : Select a VM size with at least **16 GB RAM (recommended 32 GB)**.
 - **Authentication type**: Select **SSH public key**.
 - **Username**: Enter **opc**. Do not use the default user name **azureuser**.
 - **Public inbound ports**: Select **Allow selected ports**.
 - **Select inbound ports**: Under **Network settings**, configure security group that allows HTTPS (443) and SSH (22). The required ports are open for external connection. You may need to open additional ports, see Network Port Requirements to allow access of additional services or functionality of Oracle Key Vault instance externally.
3. Complete post-installation steps. For more information, see [Step 4: Perform Post-Launch and Post-Installation Tasks](#).

Creating Oracle Key Vault Image in Amazon AWS

Oracle Key Vault provides deployment and provisioning in AWS.

- [About Provisioning Oracle Key Vault in Amazon AWS](#)
You can provision Oracle Key Vault in Amazon AWS.
- [Creating Oracle Key Vault Image on AWS](#)
Create a Base Image first to launch Oracle Key Vault cluster nodes (instances).
- [Launching an Oracle Key Vault Cluster Node \(Instance\) from the Base Image](#)
Perform the steps to launch an Oracle Key Vault cluster node (instance).

About Provisioning Oracle Key Vault in Amazon AWS

You can provision Oracle Key Vault in Amazon AWS.

Oracle Key Vault deployments in your on-premises data centers can be extended with Oracle Key Vault cluster nodes in Amazon AWS. You can also move an on-premises Oracle Key Vault cluster to Amazon AWS by removing the on-premises nodes from the cluster.

Creating Oracle Key Vault Image on AWS

Create a Base Image first to launch Oracle Key Vault cluster nodes (instances).

- Before proceeding make sure that you have installed AWS CLI. You are also required to create user roles. For more information, see [Create User Roles](#)

- Ensure that you have setup an Amazon S3 bucket to store the VM disk used for preparing the Oracle Key Vault image for AWS.
 - Ensure the AWS user has the **vmimport** role.
1. On a separate machine outside, install Oracle VM VirtualBox.
 2. Create a virtual machine on Oracle VM VirtualBox, with **Linux 8, Oracle (64 bit)** as the Operating System.
 - a. The minimum RAM for the Base Image is **8 GB**.
 - b. The minimum disk size for production deployments is **2 TB (recommended 4 TB)**; ensure to create the virtual hard disk as the **VHD** format.
 3. Create the empty VM shell with the settings from previous command (OL8, RAM, disk size). Do not install anything at this point.
 4. Mount the `installer.iso` image to the virtual machine and boot the virtual machine.

 **Note:**

Do not go to the post-install steps; instead, run the script that makes this Oracle Key Vault a **template** to clone.

5. Install an Oracle Key Vault instance on the virtual machine. Do not complete any post installation steps.
6. Log in as `root` on the virtual machine console.
7. Reattach the Oracle Key Vault installation `iso` image to the first drive of the virtual machine before running the script in the next step.
8. Run the following command:

```
/usr/local/okv/bin/okv_export_cloud_image aws
```

This script prepares the system to be exported as a base image that will be later uploaded to Amazon AWS.

9. Shut down the virtual machine.
10. Upload the VM's disk file to an Amazon S3 bucket using AWS CLI: `aws s3 cp <VM_DISK_FILE>.vhd s3://<s3_bucket_name>/<VM_DISK_FILE>.vhd`
11. Create a disk container JSON file. The following is an example of `container.json`:

```
{
  "Description": "<image_description>",
  "Format": "vhd",
  "UserBucket": {
    "S3Bucket": "<s3_bucket_name>",
    "S3Key": "<VM_DISK_FILE>.vhd"
  }
}
```

12. Import the disk as a snapshot, use the command, `aws ec2 import-snapshot --disk-container file://container.json`.

The output appears similar to the following example:

```
{
  "ImportTaskId": "import-snap-031f7b5abe599ae94",
  "SnapshotTaskDetail": {
    "DiskImageSize": 0.0,
    "Progress": "0",
    "Status": "active",
    "StatusMessage": "pending",
    "UserBucket": {
      "S3Bucket": "<s3_bucket_name>",
      "S3Key": "<VM_DISK_FILE>.vhd"
    }
  },
  "Tags": []
}
```

13. You can monitor the status of the snapshot import by using the `ImportTaskId` from the output of the previous command.

```
aws ec2 describe-import-snapshot-tasks --import-task-ids import-snap-031f7b5abe599ae94
```

```
{
  "ImportSnapshotTasks": [
    {
      "ImportTaskId": "import-snap-031f7b5abe599ae94",
      "SnapshotTaskDetail": {
        "DiskImageSize": 7654604800.0,
        "Format": "VHD",
        "Progress": "43",
        "Status": "active",
        "StatusMessage": "downloading/convertng",
        "UserBucket": {
          "S3Bucket": "<s3_bucket_name>",
          "S3Key": "<VM_DISK_FILE>.vhd"
        }
      },
      "Tags": []
    }
  ]
}
```

14. Create an AMI image from the imported snapshot.
- a. Go to the AWS EC2 Dashboard.
 - b. Select **Snapshots**, under **Elastic Block Store**.
 - c. Search for the *snapshot ID*, for example, *snap-031f7b5abe599ae94*.
 - d. Click the **snapshot ID**.
 - e. Under **Actions**, select **Create image from snapshot**.
 - f. Specify image settings.
 - i. Enter **Image name**.

- ii. Under **Virtualization type**, choose **Hardware-assisted virtualization**.
- iii. Choose Boot mode to match your VM setting provided.
- iv. Maintain the default settings for remaining fields.
- g. Click **Create**. After successful creation, a note with a link to the image appears.

Launching an Oracle Key Vault Cluster Node (Instance) from the Base Image

Perform the steps to launch an Oracle Key Vault cluster node (instance).

1. Go to the AWS EC2 dashboard.
2. Select AMIs under the **Images** folder from the left window pane.
3. Click the Oracle Key Vault image to launch.
4. Click **Launch Instance** from the AMI.
5. Provide the details in **Launch an Instance** page.

Consider these settings for the options:

- a. **Instance type:** Make sure that the selected instance type has enough memory space as configured in the Oracle Key Vault image.
- b. Under **Network settings**, configure security group that allows HTTPS (443) and SSH (22). The required ports are open for external connection. You may need to open additional ports, see Network Port Requirements to allow access of additional services or functionality of Oracle Key Vault instance externally.
6. Click **Launch Instance**.
7. Complete post-installation steps. for more information , see [Step 4: Perform Post-Launch and Post-Installation Tasks](#).

Creating Oracle Key Vault Image in Google Cloud

Oracle Key Vault provides deployment and provisioning in Google Cloud.

- [About Provisioning Oracle Key Vault in Google Cloud](#)
You can provision Oracle Key Vault in Google Cloud.
- [Creating Oracle Key Vault Image for Google Cloud](#)
You can create an Oracle Key Vault cluster in Google Cloud by first creating a Base Image, and then creating Oracle Key Vault cluster nodes from it.
- [Launching an Oracle Key Vault Instance from the Image](#)
Perform the steps to launch an Oracle Key Vault cluster node (instance).

About Provisioning Oracle Key Vault in Google Cloud

You can provision Oracle Key Vault in Google Cloud.

To provide familiar, continuously available, extremely scalable, and fault-tolerant key management for your Oracle Databases in Google Cloud (including ExaDB-D@GCP), you can install and create an Oracle Key Vault multi-master cluster in Google Cloud, or extend on-premises Oracle Key Vault deployments with Oracle Key Vault cluster nodes in Google Cloud. You can also move an on-premises Oracle Key Vault cluster to Google Cloud by removing the on premises nodes from the cluster.

Creating Oracle Key Vault Image for Google Cloud

You can create an Oracle Key Vault cluster in Google Cloud by first creating a Base Image, and then creating Oracle Key Vault cluster nodes from it.

- Ensure that you have installed and configured Google Cloud CLI.
 - Ensure that you have set up a bucket under **Cloud Storage** to store the VM disk used for preparing the Oracle Key Vault image for Google Cloud.
1. On a separate machine outside, install Oracle VM VirtualBox.
 2. Create a virtual machine on Oracle VM VirtualBox, with **Linux 8, Oracle (64 bit)** as the Operating System.
 - a. The minimum RAM for the Base Image is **8 GB**.
 - b. The minimum disk size for production deployments is **2 TB (recommended 4 TB)**; ensure to create the virtual hard disk in the **VMDK** format.
 - c. Unified Extensible Firmware Interface (UEFI) is required if disk size is greater than 2 TB.
 3. Create the empty VM shell with the settings from above (Oracle Linux 8, RAM, disk size). Do not install anything at this point.
 4. Mount the `installer.iso` image to the virtual machine and boot the virtual machine.

 **Note:**

Do not run the post-install steps; instead, run the script that makes this Oracle Key Vault a **template** to clone.

5. Install an Oracle Key Vault instance on the virtual machine. Do not complete any post-installation steps.
6. Log in as `root` on the virtual machine console.
7. Reattach the Oracle Key Vault installation `iso` image to the first drive of the virtual machine before running the script in the next step.
8. Run the following commands:

```
/usr/local/okv/bin/okv_export_cloud_image oci
/usr/local/okv/bin/okv_enable_vm_clone_utility set_cloud_platform_flag
"GCP"
```

 **Note:**

This script prepares the system to be exported as a base image that will be later uploaded to Google Cloud Platform (GCP).

9. Shut down the virtual machine.

10. Upload the disk file of the virtual machine to a bucket under cloud storage using the following Google Cloud CLI command:

```
gcloud storage cp <VM_DISK_FILE>.vmdk gs://<bucket_name>
```

11. Create an image from the **VMDK** file using Google Cloud CLI:

- Run the following command:

```
gcloud compute images import <image_name> --source-file gs://  
<bucket_name>/<VM_DISK_FILE>.vmdk  
--guest-os-features=UEFI_COMPATIBLE --data-disk --async --timeout=12h
```

 **Note:**

You can monitor the import progress using the link available in the output of the command.

Launching an Oracle Key Vault Instance from the Image

Perform the steps to launch an Oracle Key Vault cluster node (instance).

1. Go to the Google Cloud dashboard.
2. Select **Compute Engine, Images** from the left navigation menu.
3. Click the Oracle Key Vault image.
4. Click **CREATE INSTANCE** from the menu bar.
5. Select the required options to create the instance in **CREATE INSTANCE**.

Use the following settings to configure the SSH access:

- a. From the **Advanced options** drop-down list, select **Security** and then **MANAGE ACCESS**.
- b. Click **ADD ITEM** to add the SSH key.
- c. The SSH key must have *opc* user name at the end. For example,

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFZpsXjpc1KJO4i2XZKOTTSL/  
F5r9ZKU+i5aHgcSFngj opc
```

Consider these settings for the options.

- **Machine configuration:** Make sure that the selected Machine type has enough RAM as configured in the Oracle Key Vault image.
 - In the **Firewall** settings, allow HTTPS traffic.
 - Verify if the required ports are open for external connection. You may need to open additional ports, see Network Port Requirements to allow access of additional services or functionality of Oracle Key Vault instance externally.
6. Click **CREATE**.
 7. Complete post-installation steps. For more information, see [Step 4: Perform Post-Launch and Post-Installation Tasks](#).

Oracle Key Vault on Oracle VM Virtual Box

Starting with Oracle Key Vault release 21.6, a fresh installation of an Oracle Key Vault VM guest can be stored as a template, and the VM platform cloning capability can be used to clone Oracle Key Vault. You can form a Oracle Key Vault cluster using multiple cloned VM instances. By using the cloned template, the system administrator can significantly shorten the provisioning time, compared to performing a full installation of each individual cluster node.

- [Installing Oracle Key Vault on Oracle VM Virtual Box](#)
You can create Oracle Key Vault images for Microsoft Azure, Amazon AWS, and Google Cloud by first installing Oracle Key Vault as a VM template.

Installing Oracle Key Vault on Oracle VM Virtual Box

You can create Oracle Key Vault images for Microsoft Azure, Amazon AWS, and Google Cloud by first installing Oracle Key Vault as a VM template.

You can create an Oracle Key Vault VM template on a physical server that has the Oracle VM Virtual Box installed.

After the installation completes, do not complete the post-install steps; instead, run the script that makes this Oracle Key Vault a template to clone.

Note:

The following instructions are based on Oracle VM Virtual Box version 7. Instructions may vary for different versions of Oracle VM Virtual Box.

1. In the Oracle VM Virtual Box, click **New** to create a new virtual machine.
2. On the Create Virtual Machine page, select the **Guided Mode** of installation.

Note:

If you see the **Export Mode**, then you are currently in the **Guided Mode**. To toggle between the modes, click the **Guided Mode**.

3. Enter the following details on :
 - a. Enter the name for the Oracle Key Vault cloning template in the **Name** field.
 - b. Specify the directory location where the VM will be created in the **Folder** field.
 - c. Specify the path to save the `.iso` Oracle Key Vault installation file in **ISO Image** field.
 - d. Select the **Skip Unattended Installation** check box.
 - e. Click **Next**.
4. On the Hardware page, allocate RAM and CPUs to the VM.
 - a. **Base Memory**: Allocate 8GB of RAM.
 - b. **Processors**: Allocate 2 CPUs.

 **Note:**

When you create a clone from the template, you should allocate more RAM and CPUs based on the recommended system requirements.

- c. **Enable EFI** (Special OSEs only) box: Check the allocated size. This is required if you plan to use a disk larger than 2 TB.
 - d. Click **Next**.
5. On the Virtual Hard disk page, specify the disk size. The Virtual Box graphical setup guides only allow disks with up to 2 TB of storage. To create disks larger than 2 TB, you must add the disk after the VM is created.
- a. To create a disk up to 2 TB, select **Create a Virtual Hard Disk Now** button.
 - i. Specify the disk size up to 2 TB.
 - ii. Select the **Pre-allocate Full Size** check box to create the fixed size disk.

 **Note:**

Do not select **Pre-allocate Full Size** check box to create the dynamically allocated disk.

- b. To create a disk larger than 2 TB, use the `VBoxManage createmedium` command.

```
VBoxManage createmedium [disk | dvd | floppy] <--filename=filename> [--size=megabytes | --sizebyte=bytes] [--diffparent= UUID | filename] [--format= VDI | VMDK | VHD] [--variant=Standard|Fixed|Split2G|Stream|ESX|Formatted|RawDisk...] [--property=name=value...] [--property-file=name=/path/to/file/with/value...] [--format VDI | VMDK | VHD] [-variant Standard|Fixed|Split2G|Stream|ESX|Formatted|RawDisk...]
```

Examples:

- Create a 2 TB, fixed size VHD disk as needed for creating the Oracle Key Vault image for Microsoft Azure:

```
VBoxManage createmedium disk --filename <VM-location>/<VM-name>.vhd --size 2097152--format VHD --variant Fixed
```

- Create a 6 TB, dynamically allocated VHD disk as needed for creating the Oracle Key Vault image for Amazon AWS:

```
VBoxManage createmedium disk --filename <VM-location>/<VM-name>.vhd --size 6291456--format VHD --variant Standard
```

- Create a 6 TB, dynamically allocated VMDK disk as needed for creating the Oracle Key Vault image for Google Cloud:

```
VBoxManage createmedium disk --filename <VM-location>/<VM-name>.vmdk --size 6291456--format VMDK --variant Standard
```

- c. Click **Next**.
6. Review the details on the summary page.
7. Click **Finish**.
8. Once the VM ID is added to the Virtual Box main interface page, click **Settings**.
9. Go to **General** page and then select **Basic** tab, and change the OS Version to Oracle Linux (64 bit).
10. Click **OK**.
11. If you added the disk using `VBoxManage createmedium` command, you must attach the disk to the VM.
 - a. Select the **Use Host I/O Cache** check box.
 - b. Click **OK**.
 - c. Highlight the disk that is now attached to the controller, and select the **Solid-state Drive** check box, if the cloned Oracle Key Vault instances run off SSDs.
 - d. Click **OK**.

 **Note:**

Do not go to the post-install steps if you plan to use the Oracle Key Vault installed VM instance as a clone template or use it to generate image for Amazon AWS, Microsoft Azure, or Google Cloud.

6

Oracle Database Instances in Oracle Cloud Infrastructure

Oracle Key Vault deployed on-premises can manage the TDE master encryption keys for Oracle Database instances running in Oracle Cloud Infrastructure (OCI).

- [About Managing Oracle Cloud Infrastructure Database Instance Endpoints](#)
This type of Oracle Key Vault server deployment meets compliance standards for the management of encryption keys.
- [Preparing a Database Instance on OCI to be an Oracle Key Vault Endpoint](#)
Oracle Key Vault supports the use of Oracle Database instances on Oracle Cloud Infrastructure (OCI).
- [Using an SSH Tunnel Between Oracle Key Vault and Database as a Service](#)
An on-premises Oracle Key Vault communicates with an Oracle Cloud Database as a Service instance using a secure SSH tunnel.
- [Registering and Enrolling a Database as a Service Instance as an Oracle Key Vault Endpoint](#)
You can use the command line and the Oracle Key Vault management console to complete this task.
- [Suspending Database Cloud Service Access to Oracle Key Vault](#)
You can suspend one or more enrolled Database as a Service endpoints from access to Oracle Key Vault.
- [Resuming Database Cloud Service Access to Oracle Key Vault](#)
You can reinstate the connection between suspended Database Cloud Service endpoints and Oracle Key Vault.
- [Resuming a Database Endpoint Configured with a Password-Based Keystore](#)
Depending on the configuration, a Database as a Service endpoint can resume either automatically or must be manually resumed.

About Managing Oracle Cloud Infrastructure Database Instance Endpoints

This type of Oracle Key Vault server deployment meets compliance standards for the management of encryption keys.

The Oracle Database instances running in Oracle Cloud Infrastructure (OCI) can be deployed on VMshape, bare metal, or Exadata. This type of deployment provides physical separation of keys from the encrypted data, and gives on-premises administrators control and visibility of how encryption keys are used to access encrypted data in the cloud. This also meets compliance requirements where encryption keys must be managed on-premises or separate from systems containing encrypted data.

Related Topics

- [Managing a Reverse SSH Tunnel in a Multi-Master Cluster](#)
You can reverse an SSH tunnel in a multi-master cluster from more than one node to the cloud-based endpoint for redundancy.

Preparing a Database Instance on OCI to be an Oracle Key Vault Endpoint

Oracle Key Vault supports the use of Oracle Database instances on Oracle Cloud Infrastructure (OCI).

- [About Preparing a Database Instance on OCI to be an Oracle Key Vault Endpoint](#)
To prepare an Oracle Database instance on OCI to be an Oracle Key Vault endpoint, you must first configure the instance, and then create a low-privileged user.
- [Configuring a Database Cloud Service Instance](#)
A Database as a Service (DBaaS) instance must have the correct network configuration.
- [Creating a Low Privileged Operating System User on Database as a Service](#)
The low privileged user account, `okv`, will be responsible for configuring an SSH tunnel and communicating with the DBaaS instances.

About Preparing a Database Instance on OCI to be an Oracle Key Vault Endpoint

To prepare an Oracle Database instance on OCI to be an Oracle Key Vault endpoint, you must first configure the instance, and then create a low-privileged user.

Oracle Databases on Oracle Cloud Infrastructure (OCI) provide fully functional Oracle Database instances that use computing and storage resources provided by Oracle Compute Cloud Service. It eliminates the need to purchase, build, and manage silos of server and storage systems. It also makes database resources and capabilities available online so users can consume them whenever and wherever they are needed.

Configuring a Database Cloud Service Instance

A Database as a Service (DBaaS) instance must have the correct network configuration.

You can find instructions for configuring an Oracle Base Database Service instance in the Oracle Base Database Service documentation.

After you have configured the DBaaS instance, it should have the following default values:

- A public IP address
- Two users: `oracle` and `opc` (Oracle Public Cloud)
- SSH access to the `oracle` and `opc` users

Creating a Low Privileged Operating System User on Database as a Service

The low privileged user account, `okv`, will be responsible for configuring an SSH tunnel and communicating with the DBaaS instances.

By default, Database as a Service instances are provisioned with the `oracle` and `opc` users. These users have more privileges than necessary to create the SSH tunnel, so Oracle recommends that you create another low privileged operating system user named `okv` on the Database as a Service instance. Oracle Key Vault will use user `okv` to configure an SSH tunnel and communicate with the Database as a Service instances.

1. Log in to the Oracle Cloud Infrastructure (OCI) instance using public key authentication (default for Oracle OCI) as user `opc`.

```
$ ssh -i private_key_file opc@node_ip_address
```

In this specification:

- `private_key_file` is the path to your private key file (`~/.ssh/id_rsa`). This key is the counterpart to the public key that you uploaded when you provisioned the Oracle Cloud Infrastructure instance.
- `node_ip_address` is the public IP address of the Database as a Service compute node in `x.x.x.x` format.

If this is the first time you are connecting to the compute node, the SSH utility prompts you to confirm the public key.

2. In response to the prompt asking you to confirm the public key, enter `yes`.
3. Create the Oracle Key Vault user.

```
$ sudo adduser okv
```

4. Append the Oracle Key Vault user `okv` to the `AllowUsers` parameter in the SSH `sshd_config` configuration file in the `/etc/ssh/` directory.

```
$ sudo vi /etc/ssh/sshd_config
```

5. Add the following entry to the end of the file:

```
AllowUsers oracle opc okv
```

6. Restart the SSH daemon:

```
$ sudo /sbin/service sshd restart
```

7. Grant the Oracle Key Vault user `okv` permission to run `/sbin/fuser` by following these steps:

- a. Change the file permission of the `/etc/sudoers` file.

```
sudo chmod 740 /etc/sudoers
```

- b. Edit the `/etc/sudoers` file.

```
sudo vi /etc/sudoers
```

- c. Add the following entry:

```
okv    ALL=(root) NOPASSWD:/sbin/fuser
```

- d. Save the `/etc/sudoers` file. Change the file permission of the `/etc/sudoers` file.

```
sudo chmod 440 /etc/sudoers
```

- e. The `/etc/sudoers` would look similar to the following:

```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
okv     ALL=(root) NOPASSWD: /sbin/fuser
```

8. Become the `okv` user.

```
$ su okv
```

9. Create the `authorized_keys` file and then set appropriate permissions for this file.

```
$ cd $HOME
$ mkdir ~/.ssh
$ chmod 700 ~/.ssh
$ touch ~/.ssh/authorized_keys
$ chmod 640 ~/.ssh/authorized_keys
```

10. Log in to the Oracle Key Vault instance as the `support` user, and switch to `root`, and then switch to `oracle`.

11. Run the following command to upload the Oracle Key Vault public key into the `authorized_keys` file in the Oracle Cloud Infrastructure that you just created.

```
ssh-copy-id ~/.ssh/id_rsa.pub okv@node_ip_address
```

12. Confirm that the `okv` user in Oracle Key Vault can log in to the OCI instance without providing a password:

```
$ ssh okv@node_ip_address
```

Using an SSH Tunnel Between Oracle Key Vault and Database as a Service

An on-premises Oracle Key Vault communicates with an Oracle Cloud Database as a Service instance using a secure SSH tunnel.

- [Creating an SSH Tunnel Between Oracle Key Vault and a DBaaS Instance](#)
You can create a connection between Oracle Key Vault and a Database as a Service (DBaaS) instance by configuring an SSH tunnel.
- [Managing a Reverse SSH Tunnel in a Multi-Master Cluster](#)
You can reverse an SSH tunnel in a multi-master cluster from more than one node to the cloud-based endpoint for redundancy.
- [Managing a Reverse SSH Tunnel in a Primary-Standby Configuration](#)
A reverse SSH tunnel in a primary-standby configuration is similar to a reverse SSH tunnel on a standalone Oracle Key Vault server.
- [Viewing SSH Tunnel Configuration Details](#)
The Oracle Key Vault management console provides information about SSH tunnels that have been configured for Oracle Key Vault.
- [Disabling an SSH Tunnel Connection](#)
You can use the Oracle Key Vault management console to disable the Oracle Key Vault and Database as a Service instance connection.
- [How the Connection Works if the SSH Tunnel Is Not Active](#)
The SSH tunnel is kept alive even if there is no activity between Oracle Key Vault and the Database as a Service instance.

- [Deleting an SSH Tunnel Configuration](#)

You can use the Oracle Key Vault management console to delete the connection between Key Vault and a Database as a Service instance.

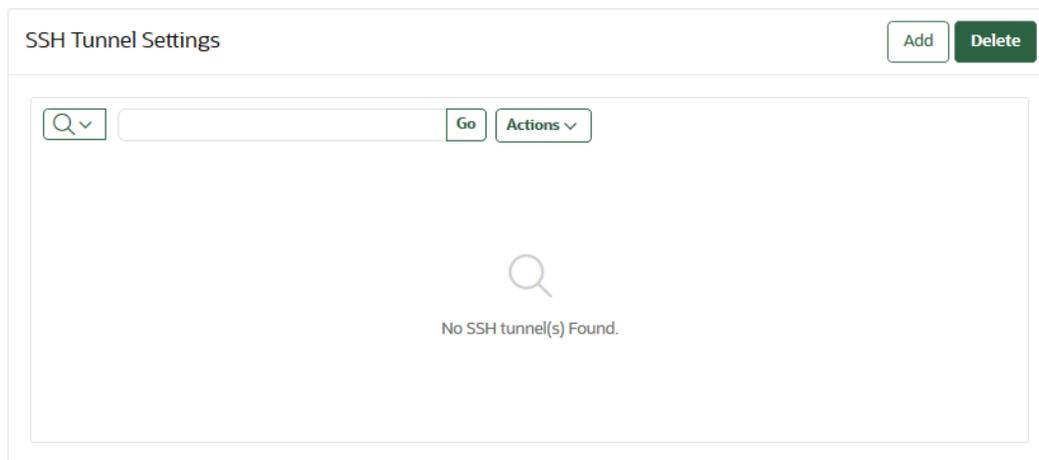
Creating an SSH Tunnel Between Oracle Key Vault and a DBaaS Instance

You can create a connection between Oracle Key Vault and a Database as a Service (DBaaS) instance by configuring an SSH tunnel.

You can configure the SSH tunnel only after you set up the Database as a Service instance. You must have the Database as a Service instance's public IP address and the name of the operating system user that you want to use to establish the tunnel.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** in the left navigation bar.
3. In the Network Details area, select **SSH Tunnel**.

The SSH Tunnel Settings page appears.



4. Click **Add**.

The Add SSH Tunnel page appears.

Add SSH Tunnel
Cancel Add

SSH tunnels are used to connect endpoints on the Oracle Cloud to Oracle Key Vault.
Provide the public IP address, port number and username relating to the remote database cloud instance.

Ensure that the following SSH public key of Oracle Key Vault has been added to the list of trusted keys on the Oracle Database Cloud Service instance before adding the corresponding SSH tunnel here.

SSH Public Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCA137jzB2XvNXHQ2SXihELXLVRYZ30bmzZ6z6n2xyih
Bh3kRp6iprbRDRITHg1KcS5kbvLY+mPFPUFTnAW41DIt2A/7quBjJgTo7DI6gekVIZ+wtgyQM
M46IRToZpzc/Ah06hD0vA3/eh8jzyDzSE3T9SOKL+W61K80MWC726VYska0VghHOKF
/844wIhaaubKE+nTEm0XIXV2nFVLM17eeM99INLzbaXn+HYQT7038IDdZ4j10owcFp0eEN
XQZee7nKCVLXrjN+OvgUHVZQnAQc4VE9jXl+e4x9FF41s0oL1Blt++9zJcUWFp9gS8S17eTD
139UDTQyRUIh5 oracle@okv020017036cZd
```

Remote Host Details

Tunnel Name *	test_tunnel
IP Address *	192.0.2.241
Port *	5697
User Name *	okv_adminj

5. Copy the text in **SSH Public Key** field and save it.

Remember that this is the public key that was copied into the OCI instance for user `okv` and was uploaded when you created a low privileged operating system user the Database as a Service instance. You will need to transport it to the Database as a Service instance and add it to the `authorized_keys` file of the Database as a Service user `okv` at `/home/okv/.ssh/authorized_keys`.

6. In the Remote Host Details page, enter information in the following fields:
- **Tunnel Name:** Choose a descriptive name that identifies the tunnel, based on the Database as a Service instance to be associated with it.
 - **IP Address:** Enter the public IP address of the Database as a Service instance.
 - **Port:** Enter a port number if you want to use a particular port number, or use the displayed default.
 - **User Name:** Enter `okv` for the user name.

You can complete these fields only after you set up the Database as a Service instance and obtained the public IP address and user name.

7. Click **Add**.

The SSH Tunnel Settings page appears. It displays the SSH tunnel that you just created and any preexisting SSH tunnels.

SSH Tunnel Settings Add Delete

Go Actions ▾

<input type="checkbox"/>	Tunnel Name	IP Address	Port	Registration Time
<input type="checkbox"/>	dev_tunnel	192.0.2.63	5697	06-DEC-2020 18:34:26
<input type="checkbox"/>	test_tunnel	192.0.2.241	5697	06-DEC-2020 18:34:46

1 - 2 of 2

It lists the tunnels created with the name, IP address, port, and registration time of each.

- Click a tunnel name to see the **SSH Tunnel Details** page.

SSH Tunnel Details Cancel Disable Delete

Tunnel Name dev_tunnel

IP Address 192.0.2.63

Port 5697

User Name okv

SSH Tunnel Status ↑

- To delete a tunnel, check the box by the tunnel that you want to delete and then click **Delete**.

You can delete more than one tunnel by selecting multiple boxes.

SSH Tunnel Settings Add Delete

Go Actions ▾

<input type="checkbox"/>	Tunnel Name	IP Address	Port	Registration Time
<input type="checkbox"/>	dev_tunnel	192.0.2.63	5697	06-DEC-2020 18:34:26

1 - 1 of 1

- Click **Disable** to disable the tunnel.

When you disable the tunnel, the endpoints that are associated with this tunnel will no longer be able to communicate with Oracle Key Vault.

SSH Tunnel Details		Cancel	Enable	Delete
Tunnel Name	dev_tunnel			
IP Address	192.0.2.63			
Port	5697			
User Name	okv			
SSH Tunnel Status	↓			

11. In the confirmation dialog box, click **Yes**.

The **Disable** button is replaced by an **Enable** button.

Related Topics

- [Creating a Low Privileged Operating System User on Database as a Service](#)
The low privileged user account, `okv`, will be responsible for configuring an SSH tunnel and communicating with the DBaaS instances.

Managing a Reverse SSH Tunnel in a Multi-Master Cluster

You can reverse an SSH tunnel in a multi-master cluster from more than one node to the cloud-based endpoint for redundancy.

Oracle recommends that you configure three tunnels. Ideally, the cloud-based reverse SSH tunnels should be from different read/write pairs. Multiple SSH tunnels to the same endpoint are distinguished by the port number used. Oracle Key Vault suggests unique port numbers based on node ID. If you want to specify different port numbers, make port numbers for SSH tunnels from different nodes to the same endpoint unique.

In a multi-master cluster, multiple SSH tunnels are created from multiple nodes to the same endpoint. However, when you register and enroll endpoints, you will only see the tunnel from that node.

Be aware of the following:

- You should register and enroll the endpoint where there is a SSH tunnel created to that endpoint.
- You only see the tunnel from that node to endpoint in the following places:
 - During the registration, the option to select the SSH tunnel.
 - After registration, when you view endpoint details, only that tunnel is displayed.
 - When you submit the enrollment token and download the endpoint software, only that tunnel is displayed. However, the endpoint software downloaded has information about all tunnels to the endpoint. This means that the endpoint is able to use all the tunnels that were created before the endpoint is created.

All nodes which have an SSH tunnel created display their tunnel to the endpoint on the Endpoint Details page. They also list all tunnels that were created from that node on the SSH Tunnels page in the Oracle Key Vault management console.

Managing a Reverse SSH Tunnel in a Primary-Standby Configuration

A reverse SSH tunnel in a primary-standby configuration is similar to a reverse SSH tunnel on a standalone Oracle Key Vault server.

The SSH key of the primary and standby servers are the same after pairing. Tunnels created on an Oracle Key Vault server before primary-standby pairing as well as tunnels created on the primary after the primary-standby pairing are valid after primary-standby operations such as switchover, and failover, although the tunnels may be unavailable during the execution of these operations.

Viewing SSH Tunnel Configuration Details

The Oracle Key Vault management console provides information about SSH tunnels that have been configured for Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** in the left navigation bar.
3. In the Network Details area, select **SSH Tunnel**.

The SSH Tunnel Settings page appears.

<input type="checkbox"/>	Tunnel Name	IP Address	Port	Registration Time
<input type="checkbox"/>	dev_tunnel	192.0.2.63	5697	06-DEC-2020 18:34:26
<input type="checkbox"/>	test_tunnel	192.0.2.241	5697	06-DEC-2020 18:34:46

1 - 2 of 2

4. Click a tunnel name to see the **SSH Tunnel Details** page.

Tunnel Name	dev_tunnel
IP Address	192.0.2.63
Port	5697
User Name	okv
SSH Tunnel Status	↑

Disabling an SSH Tunnel Connection

You can use the Oracle Key Vault management console to disable the Oracle Key Vault and Database as a Service instance connection.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** in the left navigation bar.
3. In the Network Details area, select **SSH Tunnel**.

The SSH Tunnel Settings page appears.

<input type="checkbox"/>	Tunnel Name	IP Address	Port	Registration Time
<input type="checkbox"/>	dev_tunnel	192.0.2.63	5697	06-DEC-2020 18:34:26
<input type="checkbox"/>	test_tunnel	192.0.2.241	5697	06-DEC-2020 18:34:46

4. Click a tunnel name to see the **SSH Tunnel Details** page.

Tunnel Name	dev_tunnel
IP Address	192.0.2.63
Port	5697
User Name	okv
SSH Tunnel Status	↑

5. Click **Disable** to disable the tunnel.

When you disable the tunnel, the endpoints that are associated with this tunnel will no longer be able to communicate with Oracle Key Vault. A red down arrow appears next to the SSH Tunnel Status label.

SSH Tunnel Details		Cancel	Enable	Delete
Tunnel Name	dev_tunnel			
IP Address	192.0.2.63			
Port	5697			
User Name	okv			
SSH Tunnel Status	↓			

6. In the confirmation dialog box, click **Yes**.
The **Disable** button is replaced by an **Enable** button.

How the Connection Works if the SSH Tunnel Is Not Active

The SSH tunnel is kept alive even if there is no activity between Oracle Key Vault and the Database as a Service instance.

If the tunnel stops, then it is automatically restarted. An alert will be sent if the tunnel is not available for any reason. An administrative user may elect to receive these alerts by email by configuring SMTP settings on Oracle Key Vault.

Related Topics

- [Configuring Email Settings](#)
You can configure the Simple Mail Transfer Protocol (SMTP) server properties to receive email notifications from Oracle Key Vault.

Deleting an SSH Tunnel Configuration

You can use the Oracle Key Vault management console to delete the connection between Key Vault and a Database as a Service instance.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Click **System**.
The Status page appears.
3. Select **SSH Tunnel Settings** from the left side bar.
The SSH Tunnel Settings page appears.

SSH Tunnel Settings

Search: Go Actions ▾

<input type="checkbox"/>	Tunnel Name	IP Address	Port	Registration Time
<input type="checkbox"/>	dev_tunnel	192.0.2.63	5697	06-DEC-2020 18:34:26
<input type="checkbox"/>	test_tunnel	192.0.2.241	5697	06-DEC-2020 18:34:46

1 - 2 of 2

- To delete a tunnel, check the box by the tunnel that you want to delete and then click **Delete**.

You can delete more than one tunnel by selecting multiple boxes.

SSH Tunnel Settings

Search: Go Actions ▾

<input type="checkbox"/>	Tunnel Name	IP Address	Port	Registration Time
<input type="checkbox"/>	dev_tunnel	192.0.2.63	5697	06-DEC-2020 18:34:26

1 - 1 of 1

Registering and Enrolling a Database as a Service Instance as an Oracle Key Vault Endpoint

You can use the command line and the Oracle Key Vault management console to complete this task.

- [About Registering and Enrolling a Database as a Service Instance as an Oracle Key Vault Endpoint](#)
 You must enroll the Oracle Database as a Service instance before it can communicate with an Oracle Key Vault server.
- [Step 1: Register the Endpoint in the Oracle Key Vault Management Console](#)
 The endpoint registration process downloads an `okvclient.jar` file, which contains the Oracle Key Vault software that the endpoint needs, to the local system.
- [Step 2: Prepare the Endpoint Environment](#)
 The endpoint must have a compatible version of the Java Development Toolkit (JDK) and the Oracle Database environment variables must be set.
- [Step 3: Install the Oracle Key Vault Software onto the Endpoint for Registration and Enrollment](#)
 To install the Oracle Key Vault software installation, you run the `okvclient.jar` file on the endpoint.

- [Step 4: Perform Post-Installation Tasks](#)
Post-installation tasks are important for a fully functioning Oracle Key Vault installation.

About Registering and Enrolling a Database as a Service Instance as an Oracle Key Vault Endpoint

You must enroll the Oracle Database as a Service instance before it can communicate with an Oracle Key Vault server.

The enrollment of Database as a Service endpoints is similar to the enrollment of on-premises endpoints with the following exceptions:

- Database as a Service endpoints should be registered with an endpoint type of "Oracle Database Cloud Service".
- Database as a Service endpoints have a primary tunnel IP associated with them. You must select the SSH tunnel with the same public IP address of the Database as a Service instance.
- The platform must be Linux. This is automatically selected and cannot be modified.
- You must download the jar file on-premises and transfer it to the Database as a Service instance using an out-of-band method like SCP or FTP.

Step 1: Register the Endpoint in the Oracle Key Vault Management Console

The endpoint registration process downloads an `okvclient.jar` file, which contains the Oracle Key Vault software that the endpoint needs, to the local system.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Endpoints** tab, and then **Endpoints** in the left navigation bar.
3. Click **Add**.

The Register Endpoint page appears.

The screenshot shows the 'Register Endpoint' form in the Oracle Key Vault Management Console. The form has a title bar with 'Register Endpoint' and two buttons: 'Cancel' and 'Register'. The form contains the following fields and controls:

- Endpoint Name ***: A text input field with a 'Make Unique' checkbox and a refresh icon to its right.
- Type ***: A dropdown menu with 'Oracle Database' selected.
- Platform ***: A dropdown menu with 'Linux' selected.
- Description**: A text area with a small icon in the bottom right corner.
- Administrator Email**: A text input field.
- Cluster Subgroup ***: A dropdown menu with 'austin (from Creator Node)' selected.

4. Enter the following endpoint details:
 - **Endpoint Name**: Enter a unique name for the endpoint.
 - **Make Unique**: If you are using a multi-master cluster, then choose whether to select the **Make Unique** check box. **Make Unique** helps to control naming conflicts with endpoint names across the multi-master cluster environment.

- If you select **Make Unique**, then the endpoint will be active immediately for Oracle Key Vault operations.
 - If you do not select **Make Unique**, then the endpoint account will be created in the `PENDING` state. Oracle Key Vault will then begin a name resolution operation and may rename the endpoint name to a name that is unique across the clusters. If there is a naming collision, then you must recreate the endpoint with a unique name. An endpoint in the `PENDING` state cannot be used in any Oracle Key Vault operations.
- **Type:** Select **Oracle Database Cloud Service**. When you select this option, the **Click here to add a SSH tunnel link** appears. Click this link to go to the Add SSH Tunnel page where you can add an SSH tunnel.
 - **Platform:** Linux is automatically selected.
 - **Description:** Enter a meaningful description to identify the endpoint.
 - **Administrator Email** Optionally, enter the email address of an administrator who should receive endpoint-related alerts.
 - **Cluster Subgroup:** For a multi-master cluster environment, select the cluster subgroup for this endpoint.
5. Click **Register**.
After a short delay the Endpoints page displays the new endpoint in the **Registered** state with an **Enrollment Token**.
 6. Click **Endpoint Name**. The **Endpoint Details** page appears.
Associate a default wallet with the registered endpoint now before enrolling the endpoint.
 7. Copy the **Enrollment Token**.
You will need it to download the endpoint software and then enroll the endpoint (next step).
 8. Log out of Oracle Key Vault and open a new session.
The login page appears. **Do not log in**.
 9. Click **Endpoint Enrollment and Software Download** immediately after the **Login** button.
The Enroll Endpoint & Download Software page appears.

The fields are populated with the values that were chosen by the Oracle Key Vault system administrator while registering the endpoint. You can change these values while completing the enrollment of the endpoint. Note that you must select the **Primary SSH Tunnel** for Database as a Service endpoints from the drop-down list. This is the only difference in the enrollment process from on-premises endpoints.

10. In the **Enrollment Token** field, enter the endpoint token and then click **Submit Token** to validate the token.
11. Click **Enroll** to download the `okvclient.jar` file to your local system.
12. Move the `okvclient.jar` file to a secure directory on the Cloud Database as a Service instance with appropriate permissions in place so it cannot be read or copied by others.

```
$ scp -i path_to_private_key-file path_to_okvclient.jar_on_local_computer
oracle@node_ip_address:path_to_okvclient.jar_on_cloud_db_instance
```

In this specification:

- `path_to_okvclient.jar_on_local_computer` refers to the location of `okvclient.jar` on an on-premises local computer.
- `path_to_okvclient.jar_on_cloud_db_instance` refers to the location of `okvclient.jar` on the Oracle Cloud Database as a service instance.

Related Topics

- [Types of Endpoint Enrollment](#)
The first step in enrolling an endpoint is to add the endpoint to Oracle Key Vault.
- [Setting the Default Wallet for an Endpoint](#)
Setting a default wallet for an endpoint automatically uploads the endpoint's security objects to the wallet if another wallet is not explicitly specified.

Step 2: Prepare the Endpoint Environment

The endpoint must have a compatible version of the Java Development Toolkit (JDK) and the Oracle Database environment variables must be set.

1. Ensure that you have the necessary administrative privileges to install software on the endpoint.
2. Ensure that you have JDK 1.6 or later installed, and that the `PATH` environment variable includes the `java` executable (in the `JAVA_HOME/bin` directory).

Oracle Key Vault supports JDK versions, 1.6, 7, and 8. The 64-bit version of Java is required.

3. Run the shell utility `oraenv` or `source oraenv` command to set the correct environment variables on Oracle Database servers.
4. Check that the environment variables `ORACLE_BASE` and `ORACLE_HOME` are correctly set.
 - If you used `oraenv` to set these variables, then you must verify that `ORACLE_BASE` points to the root directory for Oracle Databases, and that `ORACLE_HOME` points to a sub-directory under `ORACLE_BASE` where an Oracle database is installed.
5. Shut down the database if you are installing the endpoint software for an Oracle database configured for online TDE master encryption key management.
6. As an endpoint administrator, shutdown the Oracle database server.

Step 3: Install the Oracle Key Vault Software onto the Endpoint for Registration and Enrollment

To install the Oracle Key Vault software installation, you run the `okvclient.jar` file on the endpoint.

1. Ensure that you are logged in to the endpoint server as the endpoint administrator.
2. Confirm that the target directory exists, and that it is empty.
3. If you are installing the endpoint software for an Oracle database configured for online TDE master encryption key management, then shut down the database.
4. Run the `java` command to install the `okvclient.jar` file.

```
$ java -jar /tmp/okvclient.jar -d /etc/ORACLE/KEYSTORES/okv
```

In this specification, `-d` specifies the directory location for the endpoint software and configuration files, in this case `/home/oracle/okvutil`.

`-o` is an optional argument that enables you to overwrite the symbolic link reference to `okvclient.ora` when `okvclient.jar` is deployed in a directory other than the original directory. This argument is used only when you reenroll an endpoint.

Later on, an administrator will need to set the `WALLET_ROOT` parameter to point to the `/etc/ORACLE/KEYSTORES` directory when the Oracle database must be configured to communicate with Oracle Key Vault.

5. When you are prompted for a password, then perform either of the following two steps.

The optional password goes into two places: `okvutil` and in `ADMINISTER KEY MANAGEMENT`. With `okvutil`, only users who know that password can upload or download content to and from Oracle Key Vault. With `ADMINISTER KEY MANAGEMENT`, it becomes the password that you must use in the `IDENTIFIED BY password` clause. If you choose not to give a password, then `okvutil upload` and `download` commands will not prompt for a password, and the password for `ADMINISTER KEY MANAGEMENT` becomes `NULL`.

The choices for handling the password are as follows:

- If you want to create a password-protected wallet, at minimum enter a password between 8 and 30 characters and then press **Enter**. For better security, Oracle recommends that you include uppercase letters, lowercase characters, special characters, and numbers in the password. Use of following special characters is recommended: period (`.`), comma (`,`), underscore (`_`), plus sign (`+`), colon (`:`), and exclamation mark (`!`).
- Enter new Key Vault endpoint password (<enter> for auto-login):
`Key_Vault_endpoint_password`
 Confirm new endpoint password: `Key_Vault_endpoint_password`

A password-protected wallet is an Oracle wallet file that store the endpoint's credentials to access Oracle Key Vault. This password will be required whenever the endpoint connects to Oracle Key Vault.

- Alternatively, enter no password and then press **Enter**.

A successful installation of the endpoint software creates the following directories:

- `bin` directory, with these contents:
 - `okveps.x64` binary file
 - `okvutil.bat` program (for Windows platforms)
 - `okvutil` program (for UNIX platforms)
 - `root.sh` (for UNIX platforms)
 - `root.bat` (for Windows platforms)
- `conf` directory, with these contents:

- ewallet.p12 wallet file (Note that this wallet is the optional persistent cache. It is an auto-open wallet when the okvclient.jar file is installed without a password. It is protected with the Oracle Key Vault password if one is defined during the Oracle Key Vault client installation. It is protected with a random (unknown) password if that selection is made for this endpoint in the Oracle Key Vault management console.)
- logging.properties configuration file
- okvclient.lck lock file
- okvclient.ora configuration file
- okv.pc.lck lock file
- csdk directory, with this subdirectory:
 - lib
- jlib directory, with the following file:
 - okvutil.jar Java library jar file
- lib directory with the following file.
- For UNIX Platforms:
 - liborapkcs.so (library that the Oracle Database uses to communicate with Oracle Key Vault)
- For Windows Platforms:

Different versions of Oracle Database use different libraries to communicate with Oracle Key Vault.:

 - liborapkcs.dll
 - liborapkcs11204.dll (for 11gr2)
 - liborapkcs12102.dll (for 12.2)
 - liborapkcs12101.dll (for 12.1)
 - liborapkcs18.dll (for 18c)
 - liborapkcs19.dll (for 19c)
 - liborapkcs21.dll (for 21c)
 - libsshpkcs.dll (for SSH clients)
- log directory, which contains the following file:
 - okvutil.deploy.log log file
- ssl directory, with the following file:
 - ewallet.p12, which refers to a password-protected wallet. The cwallet.sso file refers to an auto-login wallet. These are TLS-related files and wallet files. The wallet files contain the endpoint credentials to connect to Oracle Key Vault.

Step 4: Perform Post-Installation Tasks

Post-installation tasks are important for a fully functioning Oracle Key Vault installation.

After you complete the installation, you can optionally configure a TDE connection for the endpoint, check the installation contents, and then delete the okvclient.jar file.

1. Optionally, configure a TDE connection for the endpoint.

The `liborapkcs.so` file contains the library that the Oracle database uses to communicate with Oracle Key Vault. If an endpoint uses online TDE master encryption key management by Oracle Key Vault, then you must install the PKCS#11 library by using `root.sh` or `root.bat` script.

 **Note:**

- You must run `root.sh` or `root.bat` script to install the latest Oracle Key Vault PKCS#11 library only once on a host machine that has multiple TDE-enabled Oracle databases that use Oracle Key Vault for master encryption key management.
- Ensure that you run the `root.sh` or `root.bat` script only after the upgrade of the Oracle Key Vault endpoints for all of the TDE-enabled databases on the same host machine is complete.
- Ensure that all of the TDE-enabled Oracle databases on this host are shutdown.

Log in as the `root` user and then run either of the following commands:

```
$ sudo /etc/ORACLE/KEYSTORES/okv/bin/root.sh
```

This command creates the directory tree `/opt/oracle/extapi/64/hsm/oracle/1.0.0`, changes ownership and permissions, then copies the PKCS#11 library into this directory.

2. Use a command such as `namei` or `ls -l` to confirm that a softlink was created in `$ORACLE_BASE/okv/$ORACLE_SID/okvclient.ora` to point to the real file in the `/conf` subdirectory of the installation target directory.

If the `ORACLE_BASE` environment variable has not been set, then the softlink was created in `$ORACLE_HOME/okv/$ORACLE_SID`.

3. Start the Oracle databases if the upgrade of the Oracle Key Vault endpoints for all of the TDE-enabled databases on this host machine is complete.
4. Run the `okvutil list` command to verify that the endpoint software installed correctly, and that the endpoint can connect to the Oracle Key Vault server.

```
$ ./okvutil list
```

If the endpoint is able to connect to Key Vault, then the `No objects found` message appears. If a `Server connect failed` message appears, then you must troubleshoot the installation for possible issues. Check that environment variables are correctly set. To get help on the endpoint software, run the following command:

```
$ java -jar okvclient.jar -h
```

Output similar to the following appears:

```
Production on Fri Apr 12 15:03:01 PDT 2019
Copyright (c) 1996, 2019 Oracle. All Rights Reserved.
Usage:
  java -jar okvclient.jar [-h | -help] [[-v | -verbose] [-d <destination directory>]
  [-o]]
```

```
Options:
-h or -help : Display command help.
-v or -verbose : Turn on the verbose mode. Logs will be written to files under
                 <destination directory>/log/ directory.
-d <destination directory> : Specify the software installation directory.
-o : Overwrite the current symbolic link to okvclient.ora.
```

Suspending Database Cloud Service Access to Oracle Key Vault

You can suspend one or more enrolled Database as a Service endpoints from access to Oracle Key Vault.

- [About Suspending Database Cloud Service Access to Oracle Key Vault](#)
When the DBaaS service is suspended, the Oracle Key Vault Server rejects all requests from the suspended endpoints.
- [Suspending Access for a Database Cloud Service to Oracle Key Vault](#)
After you suspend the Database as a Service access to Oracle Key Vault, you can resume the access when needed.

About Suspending Database Cloud Service Access to Oracle Key Vault

When the DBaaS service is suspended, the Oracle Key Vault Server rejects all requests from the suspended endpoints.

When you use an on-premises Oracle Key Vault to manage the online master encryption keys for Database as a Service endpoints, the master encryption keys are never stored persistently in Oracle Cloud. This way, the on-premises Oracle Key Vault administrator can control access to the encrypted data in the cloud.

An on-premises Oracle Key Vault administrator can suspend Database as a Service endpoints with a single click. This means that the Oracle Key Vault Server rejects all requests from the suspended endpoints. Because the endpoint cannot request keys from the Oracle Key Vault server, its ability to access encrypted data is lost after the key cached in memory times out. For Oracle Database Cloud Service endpoints, this time out is 5 minutes by default.

The on-premises Oracle Key Vault administrator can resume a suspended endpoint. This means that the Oracle Key Vault server can start servicing requests from the reinstated endpoint. The reinstated endpoint can now retrieve keys from the Oracle Key Vault server and access sensitive data.

In a multi-master cluster, when a node is being enabled or disabled, the information may not yet have reached all nodes in the cluster. If an endpoint attempts to contact a node whose information has not yet propagated throughout the cluster, an error may be returned.

Caution:

The suspend operation is a disruptive operation as it results in operational discontinuity. Therefore, you should use it with care. Usually, you should suspend the database only if there is a strong indication of abnormal activity in the Database as a Service instance.

You can only suspend enrolled endpoints. You cannot suspend endpoints that are in the **Registered** state. If you try to suspend endpoints that are already suspended, no operation will be performed. The endpoints will continue to be in suspended state.

Suspending Access for a Database Cloud Service to Oracle Key Vault

After you suspend the Database as a Service access to Oracle Key Vault, you can resume the access when needed.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Endpoints** tab, and then **Endpoints** in the left navigation bar.
3. In the Endpoints page, check the boxes by the endpoints that you want to suspend.
4. Click **Suspend**.
5. In the confirmation dialog box, click **Yes**.
6. Click **Endpoints** to see the suspended endpoints.

The status of suspended endpoints is highlighted in red.

<input type="checkbox"/>	Endpoint Name	Name Status	Status	Enrollment Token	Type	Platform	Default Wallet
<input type="checkbox"/>	HARRYS_HR_DATABASE	ACTIVE	REGISTERED	cFZKCM45gZT4deDM	Oracle Database	Linux	
<input type="checkbox"/>	PAULS_PEOPLESOFT_APPLICATION	ACTIVE	SUSPENDED	-	Other	Linux	App_Wallet
<input type="checkbox"/>	JACKS_COMPUTE_NODE_CLIENT	ACTIVE	SUSPENDED	-	Oracle Database Cloud Service	Linux	Cloud_App_Wallet
<input type="checkbox"/>	RAMS_DBAAS_CLIENT	ACTIVE	ENROLLED	-	Oracle Database Cloud Service	Linux	Cloud_App_wallet

Resuming Database Cloud Service Access to Oracle Key Vault

You can reinstate the connection between suspended Database Cloud Service endpoints and Oracle Key Vault.

When you resume these endpoints, their status will change to **Enrolled**. Resuming enrolled endpoints does not change their enrolled status.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Endpoints** tab, and then **Endpoints** in the left navigation bar.
3. In the Endpoints page, check the boxes by the endpoints that you want to resume.
4. Click **Resume**.

5. In the confirmation dialog box, click **Yes**.
6. Click **Endpoints** to see the reenrolled endpoints. Their status is **Enrolled**.

Endpoints Add Suspend Resume Reenroll Delete Rotate ↗

Q Go Actions ▾

<input type="checkbox"/>	Endpoint Name	Name Status	Status	Enrollment Token	Type	Platform	Default Wallet
<input type="checkbox"/>	HARRYS_HR_DATABASE	ACTIVE	REGISTERED	cFZKCM45gZT4deDM	Oracle Database	Linux	
<input type="checkbox"/>	PAULS_PEOPLESOFT_APPLICATION	ACTIVE	ENROLLED	-	Other	Linux	App_Wallet
<input type="checkbox"/>	JACKS_COMPUTE_NODE_CLIENT	ACTIVE	ENROLLED	-	Oracle Database Cloud Service	Linux	Cloud_App_Wallet
<input type="checkbox"/>	RAMS_DBAAS_CLIENT	ACTIVE	ENROLLED	-	Oracle Database Cloud Service	Linux	Cloud_App_wallet

Resuming a Database Endpoint Configured with a Password-Based Keystore

Depending on the configuration, a Database as a Service endpoint can resume either automatically or must be manually resumed.

A Database as a Service endpoint that is configured with auto-login keystore support will begin operations as soon as one of the nodes configured with reverse SSH access restores connectivity to the DBCS endpoint. On the other hand, the Database as a Service endpoint configured with password keystore will not resume operations after the endpoint is resumed on the Oracle Key Vault server. The keystore on the Database as a Service instance was closed because Oracle Key Vault suspended the endpoint. You should open the password-based keystore on the Database as a Service instance to resume operations.

7

Configuring Single Sign-On in Oracle Key Vault

You can configure Oracle Key Vault for Single Sign-On (SSO) once you have completed the configuration in Identity provider and Service provider.

- [About Single Sign-On Authentication in Oracle Key Vault](#)
Oracle Key Vault supports SAML based Single Sign-On (SSO) authentication. Single sign-on (SSO) is an authentication scheme that allows a user to log in with a single ID to any of several related, yet independent, software systems. The SSO feature allows Oracle Key Vault to join in the systems supporting SSO.
- [Configuring SAML Single Sign-On \(SSO\) Authentication](#)
SSO is an access control solution that allows users to authenticate once and get access to all enterprise resources connected to the SSO system. Oracle Key Vault SAML SSO can take advantage of the Multi-Factor-authentication supported by Identity Provider (IDP) if necessary.
- [Managing Single Sign-On in Oracle Key Vault](#)
You can easily manage the SSO configuration using the Oracle Key Vault management console.
- [Configuring Single Sign-On for Oracle Key Vault and Azure Active Directory](#)
You can configure Oracle Key Vault and Azure Active Directory for SAML based Single Sign-On (SSO).
- [Configuring Single Sign-On for Oracle Key Vault and ADFS](#)
Oracle Key Vault supports SSO on self-hosted platform Active Directory Federation Service (ADFS) server.
- [Guidelines for Managing Single Sign-On Configuration](#)
Consider these Oracle Key Vault guidelines for managing SSO configuration.

About Single Sign-On Authentication in Oracle Key Vault

Oracle Key Vault supports SAML based Single Sign-On (SSO) authentication. Single sign-on (SSO) is an authentication scheme that allows a user to log in with a single ID to any of several related, yet independent, software systems. The SSO feature allows Oracle Key Vault to join in the systems supporting SSO.

You are required to configure Identity Provider and Oracle Key Vault for SSO functionality. Oracle Key Vault SSO provides you the following features:

- SSO authentication helps you in complying with the laid-out regulations, ensuring effective access reporting.
- SSO helps in securing the user credential data by reducing the number of login's required for each application.
- The IDP supported SSO provides Multi-Factor Authentication (MFA). In this case, user needs to figure out how to configure their Identity provider.

Configuring SAML Single Sign-On (SSO) Authentication

SSO is an access control solution that allows users to authenticate once and get access to all enterprise resources connected to the SSO system. Oracle Key Vault SAML SSO can take advantage of the Multi-Factor-authentication supported by Identity Provider (IDP) if necessary.

- [About Configuring SAML Single Sign-On Authentication](#)
You can configure the Single Sign-On (SSO) in Oracle Key Vault to enable users to log into Oracle Key Vault using their Identity Provider (IDP) credentials.
- [Configuring Identity Provider for Single Sign-On for Oracle Key Vault](#)
You can configure the connection between Oracle Key Vault and IDP's to enable the Single Sign-On (SSO) in Oracle Key Vault.
- [Configuring Oracle Key Vault for Single Sign-On\(SSO\)](#)
The configurations created in IDP's are required to be configured in Oracle Key Vault.
- [Logging in to Oracle Key Vault as an SSO User](#)
An SSO user who is configured in Oracle Key Vault can log in to the Oracle Key Vault management console.

About Configuring SAML Single Sign-On Authentication

You can configure the Single Sign-On (SSO) in Oracle Key Vault to enable users to log into Oracle Key Vault using their Identity Provider (IDP) credentials.

As a System Administrator you need to configure the IDP in Oracle Key Vault before using the SAML based SSO. Oracle Key Vault supports the following IDP's:

1. Active Directory Federation Services (ADFS)
2. Microsoft Azure Active Directory
3. Other

The user must be provisioned as SSO user type in Oracle Key Vault. In a multi-master cluster environment each node is required to be configured for enabling SSO.

Configuring Identity Provider for Single Sign-On for Oracle Key Vault

You can configure the connection between Oracle Key Vault and IDP's to enable the Single Sign-On (SSO) in Oracle Key Vault.

Oracle Key Vault supports SAML based SSO authentication. You can authenticate at one application and access the service providers at different locations without the need to login multiple times.

- [SAML SSO Configuration](#)
You need to configure the Identity Provider (IDP) before starting the SSO configuration in Oracle Key Vault.
- [SAML Signing Certificate](#)
The SAML signing certificate is required for SSO user authentication.
- [User Provisioning and Authorization](#)
Identity Provider (IDP) does the user provisioning and authorization for providing the user credentials to the Service Provider (SP).
- [SAML Request Signing](#)
SAML Request Signing is required for user signature authentication.

SAML SSO Configuration

You need to configure the Identity Provider (IDP) before starting the SSO configuration in Oracle Key Vault.

The IDP shares the SAML authentication with Oracle Key Vault after receiving a request from Oracle Key Vault. For the SAML request authentication, IDP validates the signatures using the public certificate received from Oracle Key Vault.

SAML Signing Certificate

The SAML signing certificate is required for SSO user authentication.

The SAML signing certificate is one of the important steps during SSO configuration by IDP. The SAML certificate authenticates the IDP to pass the user data to the service provider for using the SSO functionality.

User Provisioning and Authorization

Identity Provider (IDP) does the user provisioning and authorization for providing the user credentials to the Service Provider (SP).

Before using SAML based SSO, the IDP user needs to be provisioned as **SSO user** type in Oracle Key Vault and needs to add proper roles or privileges.

SAML Request Signing

SAML Request Signing is required for user signature authentication.

The SAML request signature authenticates the signatures received in signed request.

Configuring Oracle Key Vault for Single Sign-On(SSO)

The configurations created in IDP's are required to be configured in Oracle Key Vault.

- [Oracle Key Vault SAML SSO Configuration](#)
Oracle Key Vault SAML SSO configuration is the next step once the SSO configuration is completed in Identity Provider (IDP).
- [Add Single Sign-On Configuration](#)
The configurations created in IDP's are required to be added in Oracle Key Vault.
- [Creating Single Sign-On User](#)
You need to create a SSO user in Oracle Key Vault by using the user name provided from Identity Provider (IDP).
- [Authenticating Single Sign-On \(SSO\) User](#)
The Oracle Key Vault Single Sign-On user requires to be validated before using the SSO functionality.

Oracle Key Vault SAML SSO Configuration

Oracle Key Vault SAML SSO configuration is the next step once the SSO configuration is completed in Identity Provider (IDP).

Oracle Key Vault uses the public certificate from the IDP to validate the signature for incoming SAML response. Oracle Key Vault accepts the SAML response and redirect the response to the Oracle Key Vault management console.

Add Single Sign-On Configuration

The configurations created in IDP's are required to be added in Oracle Key Vault.

You need to add the SSO configuration in Oracle Key Vault before using the SSO functionality. Once you have configured the IDP's, perform the following steps to configure Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **Systems** tab, then **Single Sign-On Configuration** navigation bar.
3. In the **Manage Sign-On Configuration** page, click **Add**. The **Add Sign-On Configuration** page displays.

The screenshot shows a web form titled "Add Single Sign-On Configuration". The form has several input fields and dropdown menus. The fields are: "Identity Provider" (text input), "Provider Type" (dropdown menu showing "Microsoft Active Directory Federation Services"), "Protocol" (dropdown menu showing "SAML 2.0"), "SAML Sign In URL" (text input), "SAML Sign Out URL" (text input), "Identity Provider Issuer" (text input), "Identity Provider Domain" (text input with a refresh icon), and "Identity Provider Signing Certificate" (text input with a "Choose File" button). There are "Cancel" and "Add" buttons at the top right, and a "Clear" button on the right side of the form.

4. Enter the identity provider name to save the SSO configuration in the **Identity Provider** field.
5. From the **Provider Type** drop-down list, select the service provider.
6. Select **SAML 2.0** from the **Protocol** drop-down.
7. Provide the information in the **SAML Sign in URL**, **SAML Sign Out URL**, and **Identity Provider Issuer**.
8. Click **Choose File** and upload the signing certificate issued by the identity provider in the **Identity Provider Signing Certificate** field.

 **Note:**

The information required for the **SAML Sign in URL**, **SAML Sign Out URL**, **Identity Provider Issuer**, and **Identity Provider Signing Certificate** fields are obtained from the IDP. For more information, see, [Configuring Single Sign-On for Oracle Key Vault and Azure Active Directory](#) or [Configuring Single Sign-On for Oracle Key Vault and ADFS](#).

9. Click **Add** to save the provided information.

 **Note:**

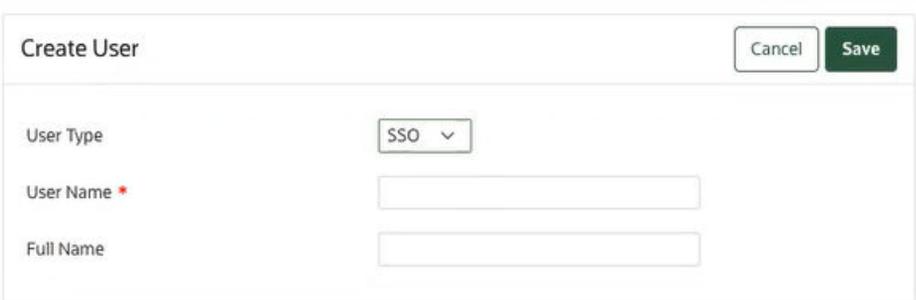
When you edit the already existing Single Sign-On configuration the **Edit Sign-On Configuration** page gets displayed. After updating the existing information , you need to click **Save**.

Creating Single Sign-On User

You need to create a SSO user in Oracle Key Vault by using the user name provided from Identity Provider (IDP).

Oracle Key Vault needs the SSO user information to validate the user principal extracted from the SAML response. To create an SSO user in Oracle Key Vault perform the following steps.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **Users** tab, and then **Manage Users** from the left navigation bar.
3. Click **Create**. The **Create User** page displays.



The screenshot shows a 'Create User' form with the following elements:

- Title: Create User
- Buttons: Cancel, Save
- User Type: A dropdown menu with 'SSO' selected.
- User Name *: A text input field.
- Full Name: A text input field.

4. From the **User Type** drop-down list, select **SSO**.

 **Note:**

In order to see **User Type** in creating user, SSO configuration should be enabled first.

5. Enter the user name in **User Name** field.

 **Note:**

Ensure that you enter the user name along with the domain name. The user name should match the Active Directory (AD) user along with the domain name.

6. Enter **Full Name**.
7. Click **Save**.

Authenticating Single Sign-On (SSO) User

The Oracle Key Vault Single Sign-On user requires to be validated before using the SSO functionality.

The System Administrator assigns the different roles and privileges to the SSO user based on the requirement. By default, all SSO type users have no role or privilege assigned to them.

To authenticate an SSO user in Oracle Key Vault perform the following steps:

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **Users** tab, and then **Manage Users** from the left navigation bar.
3. Select the **User Type** drop-down list, select **SSO**.
4. Enter the user name in **User Name** field.

 **Note:**

The user name in case of SSO should be the user name as created in IDP. The SSO user name is an email address when the user type in Oracle Key Vault is SAML based SSO.

5. Enter **Full Name**.
6. Click **Save**.

Logging in to Oracle Key Vault as an SSO User

An SSO user who is configured in Oracle Key Vault can log in to the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role
2. Select the **Systems** tab, then **Single Sign-On Configuration** from the left navigation bar.
3. In the **Manage Sign-On Configuration** page, click **Add**. The **Add Sign-On Configuration** page appears.
4. From the **Select Provider**, select the Identity Provider to enable for SSO login.

Manage Single Sign-On Configuration

Download Certificate Add Enable Disable Delete

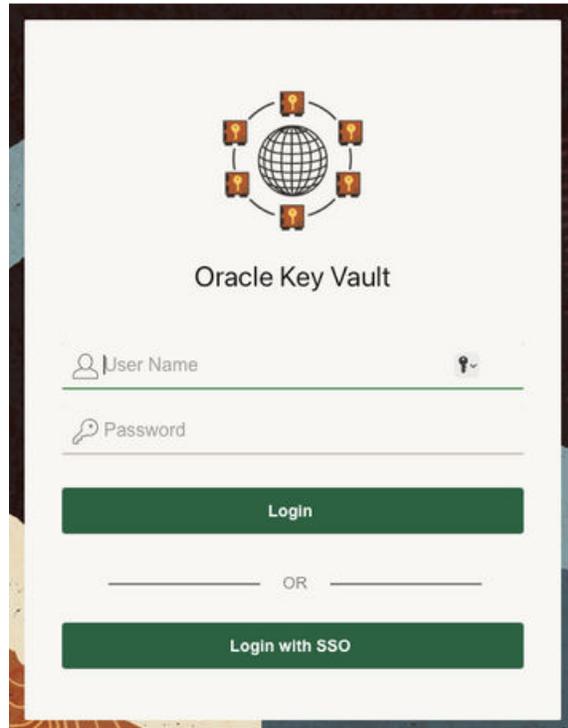
Go Actions

Select Provider	Identity Provider	Creation Time	Type	State	Protocol
<input type="radio"/>	ADFS	12-APR-2023 19:27:31	Microsoft Active Directory Federation Services	Active	SAML 2.0
<input type="radio"/>	Azure	13-APR-2023 12:16:42	Microsoft Azure Active Directory	Disabled	SAML 2.0

1 - 2 of 2

5. Log out from the Oracle Key Vault management console.

If SSO is enabled for the user, the Oracle Key Vault login screen display the **Login with SSO** button.



6. Select **Login with SSO**. You will be redirected to IDP's login screen.
7. Enter credential in the IDP's login screen.

Managing Single Sign-On in Oracle Key Vault

You can easily manage the SSO configuration using the Oracle Key Vault management console.

- [Download Oracle Key Vault Single Sign-On \(SSO\) Certificate](#)
You need to download the SSO certificate from the Oracle Key Vault management console for completing the SSO configuration with the service provider.
- [Adding Single Sign-On \(SSO\) Configuration in Oracle Key Vault](#)
Oracle Key Vault requires the configurations created in IDP's to be added.
- [Enabling Single Sign-On \(SSO\) Configuration](#)
You are required to enable the SSO in Oracle Key Vault once the SSO configuration is complete.

- [Disabling Single Sign-On \(SSO\) Configuration](#)
You can deactivate the existing SSO functionality using the disable function.
- [Deleting Single Sign-On Configuration](#)
You can permanently delete the existing SSO functionality from the Oracle Key Vault management console using the delete function.

Download Oracle Key Vault Single Sign-On (SSO) Certificate

You need to download the SSO certificate from the Oracle Key Vault management console for completing the SSO configuration with the service provider.

The service provider requires the SSO certificate to complete the configuration with the Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **Systems** tab, then **Single Sign-On Configuration** navigation bar.

The **Manage Single Sign-On Configuration** page displays.

Select Provider	Identity Provider	Creation Time	Type	State	Protocol
<input type="radio"/>	ADFS	12-APR-2023 19:27:31	Microsoft Active Directory Federation Services	Active	SAML 2.0
<input type="radio"/>	Azure	15-APR-2023 12:16:42	Microsoft Azure Active Directory	Disabled	SAML 2.0

3. On the **Manage Single Sign-On Configuration** page, click **Download Certificate** to download the certificate on your machine.
4. Choose the location to download and save the certificate on your machine.
5. Click **Save**.

The certificate gets downloaded at the provided location.

Adding Single Sign-On (SSO) Configuration in Oracle Key Vault

Oracle Key Vault requires the configurations created in IDP's to be added.

You need to add the SSO configuration in Oracle Key Vault before using the SSO functionality. Once you have configured the IDP's, perform the following steps to configure Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **Systems** tab, then **Single Sign-On Configuration** navigation bar.
3. In the **Manage Single Sign-On Configuration** page, click **Add**. The **Add Single Sign-On Configuration** page appears.

The screenshot shows a web form titled "Add Single Sign-On Configuration". The form has several input fields and buttons. At the top right, there are "Cancel" and "Add" buttons. The fields are: "Identity Provider" (text input), "Provider Type" (dropdown menu showing "Microsoft Active Directory Federation Services"), "Protocol" (dropdown menu showing "SAML 2.0"), "SAML Sign In URL" (text input), "SAML Sign Out URL" (text input), "Identity Provider Issuer" (text input), "Identity Provider Domain" (text input with a refresh icon), and "Identity Provider Signing Certificate" (text input with a "Choose File" button and a "Clear" button). Below the "Identity Provider Signing Certificate" field is a large empty rectangular area for file upload.

4. Enter the name of the identity provider name to save the SSO configuration in the **Identity Provider** field.
5. From the **Provider Type** drop-down list, select the service provider.
6. Provide the information in the **SAML Sign in URL**, **Same Sign Out URL**, and **Identity Provider Issuers**.
7. Enter the URL for **Identity Provider Domain**.
8. Click **Choose File** and upload or paste the signing certificate issues by the identity provider in the **Identity Provider Signing Certificate** field.

 **Note:**

The information required for the **SAML Sign in URL**, **Same Sign Out URL**, **Identity Provider Issuers**, and **Identity Provider Signing Certificate** fields are obtained from the IDP. For more information, see, [Configuring Single Sign-On for Oracle Key Vault and Azure Active Directory](#) or [Configuring Single Sign-On for Oracle Key Vault and ADFS](#).

9. Click **Add** to save the provided information.

 **Note:**

For the SSO configuration to come into effect, user needs to perform the enable function after saving the configuration.

Enabling Single Sign-On (SSO) Configuration

You are required to enable the SSO in Oracle Key Vault once the SSO configuration is complete.

The SSO functionality must be configured in Oracle Key Vault before using the enable or disable.

1. Select the **Service Provider** to enable from the **Manage Single Sign-On Configuration** page.
2. Click **Enable**.
3. Click **OK** in the displayed dialogue box.
4. Log out from the Oracle Key Vault management console.
5. On the Oracle Key Vault login screen, Click **Login with SSO**. This opens the service provider login screen.
6. Login using the Single Sign-On credentials. For more information, see [Configuring Single Sign-On for Oracle Key Vault and Azure Active Directory](#) or [Configuring Single Sign-On for Oracle Key Vault and ADFS](#).

Successful login ensures the selected service provider's SSO Configuration is enabled for you in Oracle Key Vault.

Disabling Single Sign-On (SSO) Configuration

You can deactivate the existing SSO functionality using the disable function.

The SSO configuration is enabled in Oracle Key Vault before using the disable functionality.

1. Select the **Service Provider** to disable from the **Manage Single Sign-On Configuration** page.
2. Click **Disable**.
3. Click **OK** in the displayed dialogue box.

The selected service provider's SSO configuration is disabled in Oracle Key Vault.

Deleting Single Sign-On Configuration

You can permanently delete the existing SSO functionality from the Oracle Key Vault management console using the delete function.

The SSO configuration is existing in the **Manage Single Sign-On Configuration** page in Oracle Key Vault before using the delete functionality.

1. Select the **Service Provider** to delete from the **Manage Single Sign-On Configuration** page.
2. Click **Delete**.
3. Click **OK** in the displayed dialogue box.

The selected service provider's SSO configuration is deleted from the Oracle Key Vault management console.

Configuring Single Sign-On for Oracle Key Vault and Azure Active Directory

You can configure Oracle Key Vault and Azure Active Directory for SAML based Single Sign-On (SSO).

1. In the **Azure** portal, select **Azure Active Directory** on the left navigation pane.
2. Select **Enterprise applications** in Azure Active Directory.
The **Enterprise applications** page appears.
3. Select **New application** from the menu bar. The **Browse Azure AD Gallery** page appears.
4. Select **Create your own application**.
5. Provide the application name for Oracle Key Vault and select **Integrate any other application you don't find in the gallery (Non-gallery)**.
6. Select your application to configure single sign-on .
7. Navigate to **Set up single sign on** under **Getting Started**.
8. Select **SAML** as the single sign-on method.
9. In the **Set up Single Sign-On with SAML** - preview page, navigate to the **Basic SAML Configuration** section.
10. Click **Edit** for **Basic SAML Configuration**.
11. Enter the following values, based on *apex_authentication.saml_metadata* from your APEX server:
 - Identifier (Entity ID): *https://<okv IP address>/ords/apex_authentication.saml_callback*
 - Reply URL: *https://<okv IP address>/ords/apex_authentication.saml_callback Logout*
 - URL: *https://<okv IP address>/ords/apex_authentication.saml_callback*
12. Click **Save**.
13. Click **Edit** for **SAML Signing Certificate**.
14. Verify that Azure signs both the response and the assertion in **SAML Singing Certificate**.
15. Save the changes.

 **Note:**

If you later need to change the Signing Option, for example, because this step was forgotten, make sure to verify the certificate. Azure might serve a different one after changing this option.

16. Click **Download** for **Certificate (Base64)**. Oracle Key Vault use this for **Identity Provider Signing Certificate**.
17. Copy the Login URL, Azure AD Identifier, and Logout URL to configure SAML SSO in Oracle Key Vault.
18. In the Oracle Key Vault management console, go to **System, Setting** and then **Single Sign-on**.

19. Save the SAML configuration and enable SAML Authentication. The **SAML Sign In URL**, **SAML Sign Out URL**, and **Identity Provider Issuer** and the **downloaded Identity Provider Signing Certificate** from Azure are required.
- [Adding User for Oracle Key Vault in Azure Active Directory \(AD\)](#)
You need to add the user to Azure AD before using the Single Sign-On in Oracle Key Vault.

Adding User for Oracle Key Vault in Azure Active Directory (AD)

You need to add the user to Azure AD before using the Single Sign-On in Oracle Key Vault.

1. Login to the Azure portal using Azure credentials.
2. Click **Azure Active Directory**.
3. In the left pane, click **Enterprise applications**.
4. Select **+ New Application**.
5. Select **+ Create your own application**.
6. Provide a name for your application in the name field. For example, *Oracle Key Vault Service*.
7. Select the **Integrate any other application you don't find in the galley (Non-gallery)**.
8. Click **Create**. Azure will take few moments to create the application.
9. On application's overview page, click **Single Sign-On** in the left pane under **Manage**.
10. Select **SAML** for the single sign-on method.
11. In the **Set up Single Sign-On with SAML** page, navigate to the **Basic SAML Configuration** section.
12. Click **Edit**. Enter the following values, based on *apex_authentication.saml_metadata* from your APEX server:
 - Identifier (Entity ID): `https://<okv IP address>/ords/apex_authentication.saml_callback`
 - Reply URL: `https://<okv IP address>/ords/apex_authentication.saml_callback`
 - Logout Url: `https://<okv IP address>/ords/apex_authentication.saml_callback`
13. Click **Save**.
14. Click **Edit** for SAML Signing Certificate.
15. Verify the **Signing Option** field display **Sign SAML response and assertion** in **SAML Singing Certificate**.
16. Click **Save**.

Note:

Make sure to verify the certificate if you later wants to change the **Signing Option**, Azure generates a new certificate, after changing this option.

17. Click **Download** at **Certificate (Base64)** option. This certificate is **Identity Provider Signing Certificate** used in Oracle Key Vault.

Optional, click **Edit** in **Verification certificates(optional)** and upload the certificate downloaded from SSO configuration page in Oracle Key Vault management console. Check on the **Require verification certificates** and then **Save**.

18. From the **Set up Node Name** page, copy the **Login URL**, **Azure AD Identifier**, **Logout URL**. You can use this information while configuring the SAML SSO in Oracle Key Vault.

 **Note:**

Go to Oracle key Vault to create, save, and enable SAML SSO configuration. See [Configuring SAML Single Sign-On \(SSO\) Authentication](#)

19. Under **Enterprise Application**, go to Oracle Key Vault.
20. Select **Assign users and groups**.
21. In the left pane, Select **Users and Groups**.
22. Click **+ Add user/group**.
23. On the **Add Assignment** pane, select **None Selected** under Users and groups or **Users**.
24. Click **Save**.
25. From the right-side **Users** pane, select the user.
26. On the **Add Assignment** pane, assign the selected user to the Oracle Key Vault enterprise application.
27. Select **Assign** at the bottom of the pane.
28. Navigate to the **User and groups** option.
29. The **User and Groups** page displays the assigned user information with access to the Oracle Key Vault application.

Configuring Single Sign-On for Oracle Key Vault and ADFS

Oracle Key Vault supports SSO on self-hosted platform Active Directory Federation Service (ADFS) server.

You need to configure ADFS for using SSO in Oracle Key Vault.

1. From the windows menu, click **Server Manager**.
2. Click **AD FS Management** in the **Tools** menu.
3. From the **Service** folder, select **Certificates**.
4. Double click **Token-signing** option in **Certificates** window pane.
5. From **Certificates**, select the **Details** tab.
6. Select **Copy to File...** The certificate Export Wizard page displays.
7. Select the **Base-64 encoded X.509 (.CER)**.
8. Click **Next**, and save the certificate as `adfs_for_okv.cer`.
9. Go to the **Service** folder again and select the **Relying Party Trust** folder.
10. Right click to open the menu options, select the **Add Relying Party Trust**.

11. From the **Add Relaing Party Trust Wizard**, select **Claims aware**.

The screenshot shows the 'Add Relying Party Trust Wizard' dialog box. The title bar reads 'Add Relying Party Trust Wizard' with a close button (X) on the right. The main heading is 'Select Data Source'. On the left, a 'Steps' pane lists: Welcome, Select Data Source (highlighted), Specify Display Name, Configure Certificate, Configure URL, Configure Identifiers, Choose Access Control Policy, Ready to Add Trust, and Finish. The main area contains the instruction: 'Select an option that this wizard will use to obtain data about this relying party:'. There are three radio button options: 1. 'Import data about the relying party published online or on a local network' (unselected). Below it is a text box for 'Federation metadata address (host name or URL):' with the example 'fs.contoso.com or https://www.contoso.com/app'. 2. 'Import data about the relying party from a file' (unselected). Below it is a text box for 'Federation metadata file location:' with a 'Browse...' button. 3. 'Enter data about the relying party manually' (selected). Below it is the instruction: 'Use this option to manually input the necessary data about this relying party organization.'. At the bottom right are buttons for '< Previous', 'Next >' (highlighted), and 'Cancel'.

12. Go to the **Select Data Source** option in the left.
13. Select the **Enter data about the replying party manually**.
14. For **Specify Display Name** option, type the name for the relying party.

 **Note:**

The relying party name will be used as a target name in ADFS command.

15. For the **Configure Certificate**, click **Next**.
16. For **Configure URL**, select **Enable support for the SAML 2.0 Web SSO protocol**.
17. Enter `https://<OKV IP address>/ords/apex_authentication.saml_callback` in **Relying party SAML 2.0 SSO service URL:**.
18. For **Configure Identifier**, enter `https://<OKV IP address>/ords/apex_authentication.saml_callback` in **Relying party trust identifier:**.

 **Note:**

This should be the unique identifier in the given ADFS.

19. For **Choose Access Control Policy**, select the required access policy from the **Choose the access control policy** field.
20. For **Ready Add Trust**, click **Next**.
21. On the **Finish** option, make sure the **Configure claim insurance policy for the application** is selected.
22. Click **Close**.
The **Edit Claim Issuance Policy** window displays.
23. In the **Issuance Transform Rules**, click **Add Rule**.
24. In the **Add transform Claim Rule Wizard**, for **Claim Rule Template**, select **Send LDAP Attribute as Claims**.
25. Click **Next**.
26. In **Configure Claim Rule**, for **Claim rule name:**, provide the name and select **Active Directory** from the **Attribute Store** drop-down.
27. In the **Mapping of LDAP attributes to outgoing claim types**, select **E-Mail-Address** on **LDAP Attribute** and **Name ID** on **Outgoing Claim Type**.
28. Click **Add Rule** to add another rule.
29. Again from **Add Transform Claim Rule Wizard**, for **Claim rule template**, select **Transform an Incoming Claim**.
30. Click **Next**.
31. In **Configure Claim Rule** for the **Claim rule name:**, provide the claim rule name.
32. Select the **E-mail Address** from the **Incoming claim type** drop-down.
33. Select **Name ID** from the **Outgoing claim type** drop-down .
34. Select **Email** from the **Outgoing name ID format** drop-down.
35. Select **Pass through all claim values** .
36. Click **Finish**.

 **Note:**

Before proceeding, make sure that the configured rules are added and available.

37. Click **Apply**.
38. In **Server Manager**, Select **AD FS** management, from the **Tools** menu.
39. Go to **Relying Party Trusts**.
40. Double-click the newly added relying party trust.
41. From the displayed **Properties** window, select the **Signature** tab.
42. Click **Add** to upload the server certificate downloaded from the Oracle Key Vault management console.
43. Click **OK**.
44. Select the **Endpoints** tab and click **Add SAML....**
45. Select **SAML Logout** from the **Endpoint type:** drop-down.
46. Select **Redirect** from the **Binding** drop-down.

47. Enter `https://<OKV IP address>/ords/f?p=7700:LOGIN` in the **Trusted URL** field.
48. Click **OK**, and then **Apply**.
49. In the displayed windows powershell, run the following the commands.
 - a. `Set-ADFSRelyingPartyTrust -TargetName <target name> -SigningCertificateRevocationCheck None`
 - b. `Set-AdfsRelyingPartyTrust -targetname <target name> -SamlResponseSignature MessageAndAssertion`where the `target name` is the name provided as display name in **Specify Display Name** option.
50. In Oracle Key Vault, make sure to provide the NTP server and populates SAML configuration and enable SAML authentication.
51. The logout URL is required for ADFS.

Guidelines for Managing Single Sign-On Configuration

Consider these Oracle Key Vault guidelines for managing SSO configuration.

Guidelines for Microsoft Azure Active Directory / Active Directory Federation Services

- The user must be provisioned as SSO user type in Oracle Key Vault.
- In a multi-master cluster environment, each node has its own SSO configuration. Each node should has its own enterprise application(Azure) or target(ADFS) because of different IP address for login or log out URL and SAL metadata.
- In a multi-master cluster environment, each node has its own SSO configuration. Each node should has its own enterprise application(Azure) or target(ADFS) because of different IP address for login or log out URL and SAL metadata.

8

Managing LDAP User Authentication and Authorization in Oracle Key Vault

You can configure a connection between Oracle Key Vault and an LDAP server (currently Microsoft Active Directory) so that their users can access Oracle Key Vault.

- [About Managing LDAP User Authentication and Authorization in Oracle Key Vault](#)
You can configure Oracle Key Vault users to be centrally managed in the configured LDAP directory server.
- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.
- [Configuring the LDAP Directory Server Connection to Oracle Key Vault](#)
Both the LDAP administrator and Oracle Key Vault administrator play a role in configuring the LDAP directory server connection to Oracle Key Vault.
- [Logins to Oracle Key Vault as an LDAP User](#)
An LDAP user who has been properly configured can log in to the Oracle Key Vault management console.
- [Managing the LDAP Configuration](#)
You can enable, validate, modify, disable, and delete the LDAP configuration.
- [Managing LDAP Groups](#)
You can modify or delete LDAP group mappings.
- [Managing Oracle Key Vault-Generated LDAP Users](#)
You cannot administer the actual LDAP user account in the LDAP directory server but you can administer the Oracle Key Vault generated user account that is created the first time the LDAP user logs in to Oracle Key Vault.

About Managing LDAP User Authentication and Authorization in Oracle Key Vault

You can configure Oracle Key Vault users to be centrally managed in the configured LDAP directory server.

Oracle Key Vault supports only Microsoft Active Directory as an LDAP provider. This type of configuration enables you to manage authentication and authorization of Oracle Key Vault users in an LDAP directory server so that LDAP users can perform the following operations:

- Log in to the Oracle Key Vault management console and perform administrative tasks for which they are authorized.
- Run Oracle Key Vault RESTful services commands at the command line.

In a large enterprise, centrally managing users and their authorization not only brings operational efficiencies in user management but also significantly improves compliance, control, and security. For example, when terminating an employee, an LDAP administrator can

lock the user's account in the LDAP directory server to end the employee's access to various systems, including Oracle Key Vault.

By centrally managing Oracle Key Vault users in an LDAP directory server, you eliminate the need to maintain user account policies and password policies for LDAP users in each Oracle Key Vault instance. Instead, you can manage these policies centrally in the LDAP directory server.

This feature implements automatic provisioning of LDAP users in Oracle Key Vault. When an LDAP user successfully logs in to Oracle Key Vault the first time, Oracle Key Vault automatically creates an Oracle Key Vault user account for this user, based on the user account information from the LDAP directory server. You cannot modify this user account except for granting or revoking Oracle Key Vault privileges. Other changes to the user account, such as changing the user's password, must be performed to the actual account in its LDAP directory server. The automatic provisioning of users is not only beneficial for new Oracle Key Vault deployments but also when access to an existing Oracle Key Vault deployment must be granted to other employees, including provisioning of new employees.

To enable authentication and authorization of LDAP users with Oracle Key Vault, an Oracle Key Vault administrator must perform the following configuration in Oracle Key Vault:

1. Configure a connection to LDAP directory server.
2. Map one or more Oracle Key Vault administrative roles, endpoint or endpoint group privileges, or user groups with LDAP groups.

Most of the configuration work is performed by an Oracle Key Vault administrator using the Oracle Key Vault management console.

To enable the Oracle Key Vault administrator to configure a connection to the LDAP directory server, the LDAP administrator creates an LDAP user account (called service directory user). Oracle Key Vault uses this user account to connect to the LDAP directory server and fetch the necessary information from the LDAP directory server during the user login process. The LDAP administrator provides the details of this LDAP service directory user as well as the trust certificate of the LDAP directory server to an Oracle Key Vault administrator.

The general process for using Oracle Key Vault with an LDAP directory server is as follows:

1. An administrator for the LDAP directory server identifies the LDAP users who need access to Oracle Key Vault, along with their authorization requirements in Oracle Key Vault. This administrator configures one or more LDAP groups, depending on the required separation of roles and duties of these users. This administrator then assigns specific users to respective LDAP groups.
2. Authorization for LDAP users is through the LDAP mappings from LDAP groups to administrative roles, endpoint or endpoint group privileges, and the user groups.
3. The Oracle Key Vault administrator uses the Oracle Key Vault management console to configure the connection between Oracle Key Vault and the LDAP directory server.
4. To configure the authorization for LDAP users, the Oracle Key Vault administrator then maps each LDAP group to one or more of these roles or privileges:
 - Administrative roles
 - Endpoint or Endpoint group privileges
 - User groups (for granting access to wallets)

You cannot map an LDAP group to a virtual wallet directly. To grant the wallet access to an LDAP group member(s) you need to first configure a user group with the appropriate wallet access permissions. Afterward, map the LDAP group with the user group. In Oracle Key Vault, the authorization of an LDAP user is determined on the basis of mappings of

administrative roles, endpoint or endpoint group privileges, or the user groups to the user's LDAP groups. For example, if an LDAP group is mapped to the Audit Manager role, then Oracle Key Vault assigns the Audit Manager role to the LDAP group member's session.

5. The LDAP users are now able to log in to Oracle Key Vault and perform tasks for which they are authorized. After first successful login, a new user account is automatically created in Oracle Key Vault.
6. In addition to the administrative roles and privileges granted to the user through LDAP group mappings, you can directly grant wallet access permissions to an LDAP user account after it has been created in Oracle Key Vault. You cannot directly grant Oracle Key Vault administrative roles, endpoint or endpoint group privileges to an LDAP user.

Authorization for an LDAP user session is a combination of the authorization granted through the LDAP groups as well as the authorization that is granted to the LDAP user locally. Authorization through LDAP groups is granted at the login time and is effective only for that session. During logon of an LDAP user, Oracle Key Vault fetches the user's LDAP groups from the directory server and determines mapped administrative roles, endpoint and endpoint group privileges, and user groups that are effective for the current user session. The set of these mapped user groups is referred to as **effective user group membership** of the LDAP user.

Note that you cannot add an LDAP user as a member of an Oracle Key Vault user group directly.

Any changes to the user's membership in the LDAP groups or to the LDAP group mappings do not affect the administrative roles, privileges, and user group memberships that are currently effective for the existing user sessions. However, any changes to the privileges that are granted to or revoked from the Oracle Key Vault user groups take effect immediately and apply to all existing sessions.

Note the following:

- You can perform the LDAP configuration with a Microsoft Active Directory version that supports the LDAP-v3 protocol.
- You can perform the LDAP configuration in a primary-standby environment. No special configuration is necessary.
-
- In multi-master cluster environments, the LDAP configuration is effective on all cluster nodes. You can configure node-specific configuration of LDAP directory server and hosts.
- For LDAP directory servers that support multiple domains, access to users from different domains is enabled by setting up multiple LDAP configurations, one for each domain.

Considerations for Granting Privileges to LDAP Users

You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Note the following considerations with regard to LDAP users, Oracle Key Vault admin roles, user groups, endpoint privileges, and wallet privileges:

- You cannot directly add an LDAP user as a member of an Oracle Key Vault user group.
- When a local Oracle Key Vault user with the Create Endpoint privilege creates an endpoint, Oracle Key Vault grants the Manage Endpoint privilege on that endpoint to the local user. For LDAP users, however, when creating an endpoint, Oracle Key Vault grants the Manage Endpoint privilege to the LDAP group through which the LDAP user inherited the Create Endpoint privilege. When the LDAP user inherits the Create Endpoint privilege

from more than one LDAP groups, the Manage Endpoint privilege is granted to the LDAP group with the earliest creation timestamp in Oracle Key Vault. Similar behavior applies for the Endpoint Group privileges.

- After the LDAP user account is created in Oracle Key Vault, you can directly grant this user wallet privileges locally. However, it is recommended that you grant the wallet privileges to LDAP users through the LDAP group mappings
-
- You cannot directly grant the administrator roles, endpoint, endpoint group, and monitor privilege to an LDAP user account in Oracle Key Vault. An LDAP user account in Oracle Key Vault inherits these privileges through LDAP group mappings only.

Configuring the LDAP Directory Server Connection to Oracle Key Vault

Both the LDAP administrator and Oracle Key Vault administrator play a role in configuring the LDAP directory server connection to Oracle Key Vault.

- [Step 1: Prepare the LDAP Directory Server](#)
Before the Oracle Key Vault administrator can create a connection to an LDAP directory server, the LDAP administrator must perform preparation tasks.
- [Step 2: Create the LDAP Connection in Oracle Key Vault](#)
An Oracle Key Vault user who has the System Administrator role uses the Oracle Key Vault management console to create the LDAP connection.
- [Step 3: Mapping LDAP Groups in Oracle Key Vault](#)
You can configure authorization for LDAP users in Oracle Key Vault by creating LDAP group mappings and granting them appropriate roles and privileges.

Step 1: Prepare the LDAP Directory Server

Before the Oracle Key Vault administrator can create a connection to an LDAP directory server, the LDAP administrator must perform preparation tasks.

1. As the LDAP administrator (or a user who has the appropriate privileges), log in to the LDAP directory server.
2. Create or designate existing LDAP groups that you want to map to Oracle Key Vault.
3. Assign users to these LDAP groups.

The group will determine the privileges that its member users will have in Oracle Key Vault when the connection configuration is complete. If a user must have specific privileges that are not covered by any existing LDAP groups, then create a specific group for this user.

4. Create a service directory user account if such an account does not yet exist, and then provide this account name and its password to the Oracle Key Vault administrator.

This service directory user account will be used in the LDAP configuration that the Oracle Key Vault administrator will create. Oracle Key Vault will use this account to perform necessary LDAP actions, such as searches. If this user account changes in the future, then notify the Oracle Key Vault administrator immediately.

5. Obtain the trust certificate for the LDAP directory server and then provide this certificate to the Oracle Key Vault administrator.

This certificate will be used in the LDAP configuration that the Oracle Key Vault administrator will create.

Step 2: Create the LDAP Connection in Oracle Key Vault

An Oracle Key Vault user who has the System Administrator role uses the Oracle Key Vault management console to create the LDAP connection.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** from the left navigation bar.
3. In Network Services, click **LDAP** to display the Manage LDAP Configuration page.
4. Click **Add** to display the Add LDAP Configuration page.
5. Enter the following settings:
 - **Configuration Name:** Enter a name for the LDAP configuration. The maximum character length is 120 bytes.
 - **Directory Type:** Ensure that Microsoft Active Directory is selected from this list.
 - **Service Directory User Name:** Enter the service directory user account that the LDAP administrator provided. You can enter this name using any of the following formats:
 - **NetBIOS Domain Name\Account Name:** For example, `global\johndoe`
 - **User principal name:** For example, `johndoe@example.com`
 - **Service Directory User Password:** Enter the password that the LDAP administrator provided with the service directory user account. If the password changes in the LDAP directory server, you must update it in Oracle Key Vault.
 - **Hostname:** Enter either a host name or an IP address for the Microsoft Active Directory domain controller (server) that will service the client requests.
 - **LDAPS Port:** Enter the port number. 636, the default, is the standard port number for LDAP connections for Secure Sockets Layer (SSL) connections.
 - **Trusted Certificate:** Either paste the contents of the server trust certificate of the LDAP directory server that the LDAP administrator provided, or upload the certificate as a file.
 - **Domain Name:** This setting is auto-populated when you complete the preceding settings and click the **Get Domain Name** button. This is the name of the Microsoft Active Directory domain of which the specified host (domain controller) is a member. You cannot change this setting.
 - **Search Base DN:** This setting is auto-populated when you complete the preceding settings and click the **Get Domain Name** button. It represents a distinguished name of the search base object, which defines the location in the directory from which the LDAP search begins. This setting is useful for environments where the number of users and groups in a directory is very large, and if the users and groups that are relevant for managing Oracle Key Vault access are placed under this directory container. Setting the base DN to this directory container can help to improve performance for user and group searches. Optionally, modify this search base.
 - **Defunct LDAP Users Grace Period (in days):** Enter the duration in days after which the users deleted in LDAP directory server are automatically deleted from Oracle Key Vault. This value also defines the duration after which the users whose LDAP

configuration no longer exists are deleted from Oracle Key Vault automatically. Enter a positive integer of 0 or higher. The default is 15 days.

6. Click **Test Connection** to ensure that the connection works.
7. Click **Add** to complete the configuration.

The Manage LDAP Configuration page appears, with the new configuration listed under LDAP Configuration Name.

Related Topics

- [Step 3: Mapping LDAP Groups in Oracle Key Vault](#)
You can configure authorization for LDAP users in Oracle Key Vault by creating LDAP group mappings and granting them appropriate roles and privileges.

Step 3: Mapping LDAP Groups in Oracle Key Vault

You can configure authorization for LDAP users in Oracle Key Vault by creating LDAP group mappings and granting them appropriate roles and privileges.

An Oracle Key Vault user who has the Key Administrator role can create an LDAP group mapping. At the creation time, the Key Administrator can also optionally map the LDAP group to certain roles and privileges that the Key Administrator is authorized to grant. Depending upon the specific roles and privileges that you want to grant to the LDAP group, users with different administrator roles may need to perform the authorization grant to the LDAP group after the LDAP group mapping is created.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Configure the Oracle Key Vault user groups that you want to map to the LDAP groups. You can grant wallet privileges to an LDAP group through Oracle Key Vault user group.

Ensure that these user groups have the appropriate Oracle Key Vault privileges, and that you understand how privilege grants and revokes work for LDAP users.

3. Select the **Users** tab, and then **Manage LDAP Mappings** from the left navigation bar.
4. Click **Create** to display the **Create LDAP Group Mapping** page.
5. Enter the following settings:
 - **Domain:** From the list, select the domain that is associated with the LDAP directory server configuration for which you want to define the mapping.
 - **LDAP Group:** From the list, select the LDAP group. You can choose from the following:
 - **Fetch All Groups** provides a listing of all available groups that you can add. Select the group from the menu following this label. If the number of LDAP groups exceeds the number of objects that can be returned in a single search result, then an error is returned. Narrow down your LDAP group search by using either the **Filter by CN (Common Name)** or **Specify DN (Distinguished Name)** options.
 - **Filter by CN (Common Name)** enables you to select from a filtered list by CN. In the field that appears when you select this choice, enter the CN prefix, and in the list following it, select as needed. For example, to search for all the LDAP groups that start with the common name **Admin**, enter **Admin**. If the number of LDAP groups that match the **Filter by CN (Common Name)** criteria exceeds the number of objects that can be returned in a single search result, then an error is returned. Narrow down your LDAP group search by further refining the **Filter by CN (Common Name)** search or use the **Specify DN (Distinguished Name)** option.

- **Specify DN (Distinguished Name)** enables you to select from a filtered list by DN. In the field that appears when you select this choice, enter the complete DN for the group, and in the list following it, select as needed.
 - **Roles:** Optionally, grant the **Key Administrator** role to the LDAP group. (Remember that you cannot grant a role that is different from the role you currently have. Also, you must have received the role with the **Allow Forward Grant** option.)
If you want to grant the user the System Administrator or Audit Manager role, then a user who has the Key Administrator role first must create an LDAP group mapping (with or without mapping to any roles, privileges, and user groups). After the LDAP group mapping is created, a user who is granted the System Administrator or Audit Manager role with the **Allow Forward Grant** option then can edit the existing LDAP group mapping to map it with the corresponding administrative role.
 - **Privileges:** Optionally, select the **Create Endpoint Group** to assign privileges to LDAP group.
If you want to grant the LDAP group the **Create Endpoint** privilege, then a user who has the Key Administrator role must first create an LDAP group mapping. After the LDAP group mapping is created, a user who is granted the System Administrator role then can edit the existing LDAP group mapping to map it with the **Create Endpoint** privilege.
 - Similarly, if you want to grant the LDAP group the Monitor privilege, then a user who has the Key Administrator role must first create an LDAP group mapping. After the LDAP group mapping is created, a user with the System Administrator role can then edit the existing LDAP group mapping to grant the Monitor privilege.
 - **Access on Endpoint Groups** Optionally, select the required endpoint group to grant the Manage Endpoint Group privilege for it to the LDAP group.
If you want to grant the LDAP group the Manage Endpoint privilege on an endpoint, then a user who has the Key Administrator role must first create an LDAP group mapping. After the LDAP group mapping is created, a user who is granted the System Administrator role then can edit the existing LDAP group mapping to grant the Manage Endpoint privilege on an endpoint by selecting the endpoint from the **Access on Endpoints** area.
 - **User Groups:** Optionally, select the Oracle Key Vault user groups that you want to map to the LDAP group.
You can find information about a user group by clicking its **Details** button.
6. Click **Create**.
The LDAP Access Mappings page appears, where the new mapping is included in the list.
 7. Define more LDAP group mappings as necessary.

At this stage, the configuration is complete and LDAP users can log in to Oracle Key Vault.

Related Topics

- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Logins to Oracle Key Vault as an LDAP User

An LDAP user who has been properly configured can log in to the Oracle Key Vault management console.

- [About Logins to Oracle Key Vault as an LDAP User](#)
After the LDAP directory server configuration with Oracle Key Vault is complete, LDAP users can log in to Oracle Key Vault if they have valid authorization.
- [Logging in to Oracle Key Vault as an LDAP User](#)
An LDAP user who is a member of an LDAP group that has been mapped to any Oracle Key Vault administrative role, user group, or endpoint or endpoint group privileges can log in to the Oracle Key Vault management console.

About Logins to Oracle Key Vault as an LDAP User

After the LDAP directory server configuration with Oracle Key Vault is complete, LDAP users can log in to Oracle Key Vault if they have valid authorization.

The login is successful if:

- The user provides the correct LDAP credential.
- The user's LDAP groups from the LDAP directory server map to at least one of the Oracle Key Vault administrative roles, endpoint or endpoint group privileges, or user groups.

At the login time, user's authorization is determined based on the LDAP groups of which this user is a member. The user is granted administrative roles, endpoint or endpoint group privileges, and the privileges of the user groups that are mapped to user's LDAP groups. When a user successfully logs into Oracle Key Vault for the first time, a new user account is automatically created in Oracle Key Vault. (Ensure that you understand how privilege grants work for LDAP users.)

In a multi-master cluster environment, an LDAP user can log in to any node in the cluster. The first time that the LDAP user logs in to a node, a single Oracle Key Vault-generated user account is created for this user. This account will apply to all nodes in the cluster.

Valid LDAP users can run the Oracle Key Vault RESTful services commands. *Oracle Key Vault RESTful Services Administrator's Guide* describes how to use the RESTful services utility.

Related Topics

- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Logging in to Oracle Key Vault as an LDAP User

An LDAP user who is a member of an LDAP group that has been mapped to any Oracle Key Vault administrative role, user group, or endpoint or endpoint group privileges can log in to the Oracle Key Vault management console.

1. Open a web browser.
2. Connect using an HTTPS connection and the IP address of Oracle Key Vault.

For example, to log in to a server whose IP address is 192.0.2.254, enter:

```
https://192.0.2.254
```

3. After the login screen appears, enter the following credentials:
 - **Domain:** From the list, select the domain of the LDAP user.
 - **User Name:** Enter your user name using one of the following formats:
 - **NetBIOS Domain Name\Account Name:** For example, `global\johndoe`

For convenience, when a user specifies the user name in the NetBIOS Domain Name or User Principal Name format (described next), the domain name in the drop-down list is automatically selected based on the pattern matching of the NetBIOS domain name or the domain name of the user principal name with the names of the configured domains. You can select the domain manually, as needed. The domain name Local is not an Active Directory domain. The use of the domain Local indicates that the user account was created locally.

- **User principal name:** For example, johndoe@example.com
 - **Login name:** For example, johndoe. This name must match the `sAMAccountName` attribute of the LDAP user account.
 - **Password:** Enter your password.
4. Click **Login**.

LDAP users who have the appropriate Oracle Key Vault authorization can also run the Oracle Key Vault RESTful services utility commands.

Related Topics

- [About Managing LDAP Users](#)
The LDAP user account in Oracle Key Vault is an automatically created account that is based on the LDAP user account in the configured LDAP directory server.
- *Oracle Key Vault RESTful Services Administrator's Guide*

Managing the LDAP Configuration

You can enable, validate, modify, disable, and delete the LDAP configuration.

- [Enabling an LDAP Configuration](#)
A user who has the System Administrator role can enable an LDAP configuration.
- [Modifying an LDAP Configuration](#)
A user who has the System Administrator role can modify an LDAP configuration.
- [Testing an LDAP Configuration](#)
A user who has the System Administrator role can test an LDAP configuration.
- [Disabling an LDAP Configuration](#)
A user who has the System Administrator role can disable an LDAP configuration.
- [Deleting an LDAP Configuration](#)
A user who has the System Administrator role can delete an LDAP configuration.

Enabling an LDAP Configuration

A user who has the System Administrator role can enable an LDAP configuration.

An LDAP configuration is effective only when it has been enabled. By default, after an LDAP configuration is created, it is enabled. In a multi-master cluster environment, the enablement of an LDAP configuration apply to all nodes in the cluster and can be performed in any node.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** from the left navigation bar.
3. In Network Services, click **LDAP** to display the Manage LDAP Configuration page.
4. Select the check box for the LDAP configuration and then click the **Enable** button.

5. In the confirmation window, click **OK**.

Modifying an LDAP Configuration

A user who has the System Administrator role can modify an LDAP configuration.

In a multi-master cluster environment, changes to an LDAP configuration apply to all nodes in the cluster and can be performed in any node. However, be aware that a node-specific host configuration takes precedence over cluster-wide host configuration.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** from the left navigation bar.
3. In Network Services, click **LDAP** to display the Manage LDAP Configuration page.
4. Select the LDAP configuration name to display the Edit LDAP Configuration page.
5. Modify the following settings as necessary:
 - **Configuration Name:** Update the name of the configuration.
 - **Service Directory User Name:** Update the service directory user name.
 - **Service Directory User Password:** Update the service directory user's password.
 - **Trusted Certificate:** Either paste the contents of the server trust certificate of the LDAP directory server that the LDAP administrator provided, or upload the certificate as a file.
 - **Search Base DN:** Update the base DN that is used for searching users and groups
 - **Defunct LDAP Users Grace Period (in days):** Enter a new grace period value. The default is 15.
 - Under Servers, do the following:
 - To add a new server, click **Add** and then provide the **Hostname**, **Port**, and **Service Directory User Password**. To test the connection, click **Test Server**. Then click **Add**. You can only select a server that is in the same domain as the current server.
 - To remove a server, select its check box and then click **Delete**.
6. Click **Test Connection(s)** to ensure that the new configuration settings work.
7. Click **Save**.

Note:

Do not add the same LDAP server (domain controller) more than once using different host names and IP addresses. You should add an LDAP server only once. At the cluster or node-specific configuration level, Oracle Key Vault does not allow the addition of a server with the duplicate host name. However, it cannot prevent the addition of the multiple server entries for the same server using different values for the host name. For example, you may add the server once using its host name and another time using its IP address. Such configuration is not recommended.

Testing an LDAP Configuration

A user who has the System Administrator role can test an LDAP configuration.

In a multi-master cluster environment, you can test the LDAP configuration from any node in the cluster. The test connection validates the connection to LDAP hosts that are effective for the current cluster node. If node-specific LDAP hosts are configured, then the connection to those hosts is validated.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** from the left navigation bar.
3. In Network Services, click **LDAP** to display the Manage LDAP Configuration page.
4. Select the LDAP configuration name to display the Edit LDAP Configuration page.
5. Click **Test Connection(s)**.

Disabling an LDAP Configuration

A user who has the System Administrator role can disable an LDAP configuration.

Disabling an LDAP configuration effectively makes the configuration unavailable for use. Users from the disabled LDAP configuration are denied access when they try to log in into Oracle Key Vault. However, disabling an LDAP configuration does not affect users who are currently logged in using the configuration. In a multi-master cluster environment, the disablement of an LDAP configuration applies to all nodes in the cluster and can be performed in any node.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** from the left navigation bar.
3. In Network Services, click **LDAP** to display the Manage LDAP Configuration page.
4. Select the check boxes for the configurations to disable, and then click **Disable**.
5. In the confirmation window, click **OK**.

Deleting an LDAP Configuration

A user who has the System Administrator role can delete an LDAP configuration.

In a multi-master cluster environment, the deletion of an LDAP configuration applies to all nodes in the cluster and can be performed from any node. However, deleting an LDAP configuration does not log out users who are currently logged in using the configuration.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** from the left navigation bar.
3. In Network Services, click **LDAP** to display the Manage LDAP Configuration page.
4. Select the check boxes for the LDAP configurations that you want to delete and then click one of the following buttons:
 - Click **Delete** if there are no mappings defined in Oracle Key Vault for any LDAP groups associated with the LDAP configuration.

- Click **Force Delete** if there are group mappings defined for this LDAP configuration. You must have both the System Administrator and Key Administrator role to perform this operation. Otherwise, first delete all LDAP group mappings defined for the LDAP configuration before a user with the System Administrator role deletes the LDAP configuration.

5. In the confirmation window, select **OK**.

When you delete an LDAP configuration, the associated LDAP user accounts are not deleted immediately. Oracle Key Vault deletes these accounts after number of days that were specified in the **Defunct LDAP Users Grace Period** setting in the Edit LDAP Configuration page have passed. You can delete an LDAP user account any time. If you recreate the identical LDAP configuration before associated LDAP user accounts are deleted, then Oracle Key Vault will make these user accounts valid again.

Managing LDAP Groups

You can modify or delete LDAP group mappings.

- [About Managing LDAP Groups](#)
The LDAP group can be mapped to Oracle Key Vault administrator roles, endpoint and endpoint group privileges, and user groups.
- [Creating an LDAP Group Mapping](#)
After you have created an LDAP configuration, you can create one or more LDAP group mappings.
- [Modifying an LDAP Group Mapping](#)
You can modify an LDAP group mapping to change its mapped roles, endpoint or endpoint group privileges, and user groups.
- [Validating LDAP Group Mappings](#)
In the event that LDAP groups change in the LDAP directory server, a user who has the Key Administrator role can validate their mappings in Oracle Key Vault.
- [Deleting LDAP Group Mappings](#)
A user who has the Key Administrator role can delete one or more LDAP groups and associated mappings from Oracle Key Vault.

About Managing LDAP Groups

The LDAP group can be mapped to Oracle Key Vault administrator roles, endpoint and endpoint group privileges, and user groups.

An Oracle Key Vault user who has the Key Administrator role can create an LDAP group mapping. At the creation time, the Key Administrator can also optionally map the LDAP group to certain roles and privileges that the Key Administrator is authorized to grant. Depending upon the specific roles and privileges that you want to grant to the LDAP group, users with different administrator roles may need to perform the authorization grant to the LDAP group after the LDAP group mapping is created.

Local Oracle Key Vault users cannot be members of an LDAP group.

Creating an LDAP Group Mapping

After you have created an LDAP configuration, you can create one or more LDAP group mappings.

You can map LDAP groups to one or more Oracle Key Vault administrative roles, user groups, endpoint or endpoint groups privileges.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, then **Manage LDAP Mappings** from the left navigation bar.
3. In the LDAP Group Mappings page, click **Create**.

The Create LDAP Group Mapping page appears.

4. Enter the following settings:
 - **Domain:** From the list, select the domain that is registered with Oracle Key Vault.
 - **LDAP Group:** From the list, select the LDAP group. You can choose from the following:
 - **Fetch All Groups** provides a listing of all available groups that you can add. Select the group from the menu following this label. If the number of LDAP groups exceeds the number of objects that can be returned in a single search result, then an error is returned. Narrow down your LDAP group search by using either the **Filter by CN (Common Name)** or **Specify DN (Distinguished Name)** options.
 - **Filter by CN (Common Name)** enables you to select from a filtered list by CN. In the field that appears when you select this choice, enter the CN prefix, and in the list following it, select as needed. For example, to search for all the LDAP groups that start with the common name **Admin**, enter **Admin**. If the number of LDAP groups that match the **Filter by CN (Common Name)** criteria exceeds the number of objects that can be returned in a single search result, then an error is returned. Narrow down your LDAP group search by further refining the **Filter by CN (Common Name)** search or use the **Specify DN (Distinguished Name)** option.
 - **Specify DN (Distinguished Name)** enables you to select from a filtered list by DN. In the field that appears when you select this choice, enter the complete DN for the group, and in the list following it, select as needed.
 - **Roles:** Optionally, select one or more Oracle Key Vault roles to map to the LDAP group.
 - **Audit Manager**
 - **Key Administrator:**
 - **System Administrator**

Note the following:

- **Allow Forward Grant** check box: For each role that you select, a check box for Allow Forward Grant appears. If you want the LDAP group members to be able to grant or revoke the administrative role to and from other users, then select the Allow Forward Grant check box. Only the users who are granted a specific role with the Allow Forward Grant option can grant this role to others including granting it to LDAP users through an LDAP group mapping. Remember that you cannot grant a role that is different from the role you currently have. Also, you must have received the role with the Allow Forward Grant option to be able to grant it to an LDAP group mapping.

- If your site follows strict separation-of-duty guidelines and the user with the Key Administrator role does not have the System Administrator and Audit Manager roles, then granting the System Administrator or Audit Manager role to an LDAP group is a two step-process:
 - a. A user who has the Key Administrator role must first create an LDAP group mapping (with or without mapping any roles or privileges).
 - b. After the LDAP group mapping is created, a user who is granted the System Administrator or Audit Manager role with the Allow Forward Grant option then can edit the existing LDAP group mapping to map it with the corresponding administrative role and also optionally select the Allow Forward Grant option.
- **Privileges:** Optionally, select **Create Endpoint** or **Create Endpoint Group** privilege to map to the LDAP group.

You need to have the System Administrator role to grant the Create Endpoint privilege. If the user creating the LDAP group mapping does not have the System Administrator role, then granting the Create Endpoint privilege to an LDAP group is a two step-process:

 - a. A user who has the Key Administrator role must first create an LDAP group mapping.
 - b. After the LDAP group mapping is created, a user who is granted the System Administrator role then can edit the existing LDAP group mapping to map it with the Create Endpoint privilege.
- **Access on Endpoints and Endpoint Groups:** Optionally, select the required endpoints or the endpoint groups to grant the Manage Endpoint or Manage Endpoint Group privilege on them to the LDAP group. You need to have the System Administrator role to grant the Manage Endpoint privilege on an endpoint. If the user creating the LDAP group mapping does not have the System Administrator role, then granting the Manage Endpoint privilege on an endpoint to an LDAP group is a two step-process:
 - a. A user who has the Key Administrator role must first create an LDAP group mapping.
 - b. After the LDAP group mapping is created, a user who is granted the System Administrator role then can edit the existing LDAP group mapping to grant the Manage Endpoint privilege on an endpoint by selecting the endpoint from the Access on Endpoints.
- **User Groups:** Under user groups select the Oracle Key Vault user groups that you want to map to the LDAP group.

You can find information about a user group by clicking its Details button.

5. Click **Create**.

After the LDAP group mapping is created, mapped roles, privileges and user group mappings take effect when an LDAP group member user logs into Oracle Key Vault. Changes in the LDAP group mapping do not affect the authorization granted to the users in their existing sessions.

Modifying an LDAP Group Mapping

You can modify an LDAP group mapping to change its mapped roles, endpoint or endpoint group privileges, and user groups.

Depending upon the specific roles and privileges that you want to grant or revoke to and from the LDAP group, users with different administrator roles may need to perform the authorization grant to the LDAP group.

1. Log in to the Oracle Key Vault management console as a user who has the necessary authorization to grant the role or privileges to others. For example, to map the Audit Manager role to an LDAP group, you must have the Audit Manager role with the **Allow Forward Grant** option.
2. Select the **Users** tab, then **Manage LDAP Mappings** from the left navigation bar.
3. Under **LDAP Group Mappings**, find the LDAP group whose privileges you want to change.
4. Select the **Edit** button for this LDAP group to display the **Edit LDAP Group Mapping** page
5. Modify the following settings, as necessary:
 - **Roles:** Optionally, select one or more Oracle Key Vault roles to map to the LDAP group.
Remember that you cannot grant or revoke a role that is different from the role you currently have. Also, you must have received the role with the Allow Forward Grant option to be able to grant it to an LDAP group mapping.

To grant the admin roles to the LDAP group mapping, select the corresponding check boxes from the available roles:
 - **Audit Manager**
 - **Key Administrator**
 - **System Administrator**

Note the following:

- **Allow Forward Grant check box:** For each role that you select, a check box for **Allow Forward Grant** appears. If you want the LDAP group members to be able to grant or revoke the administrative role to and from other users, then select the **Allow Forward Grant** check box. Only the users who are granted a specific role with the **Allow Forward Grant** option can grant this role to others including granting it to LDAP users through an LDAP group mapping.
To revoke the admin roles previously granted to the LDAP group mapping, deselect the check boxes for the corresponding roles. If you want to only revoke the Allow Forward Grant option, then deselect the corresponding Allow Forward Grant check box.
- **Privileges:** Optionally, you may grant or revoke the **Create Endpoint**, **Create Endpoint Group**, or **Monitor privilege** to and from the LDAP group mapping. You need the System Administrator role to grant or revoke the Create Endpoint privilege or Monitor privilege, and the Key Administrator role to grant, or revoke the Create Endpoint Group privilege.

To grant the Create Endpoint, Create Endpoint Group, or Monitor privilege, select the check box for the corresponding privilege.

To revoke the Create Endpoint, Create Endpoint Group, or Monitor privilege, deselect the check box for the corresponding privilege.
- **Access on Endpoints and Endpoint Groups:** Optionally, you may grant or revoke the Manage Endpoint or Manage Endpoint Group privilege on one or more endpoints or endpoint groups to and from the LDAP group mapping.

You need the System Administrator role to grant or revoke the Manage Endpoint privilege and the Key Administrator role to grant or revoke the Manage Endpoint Group privilege.

To grant the Manage Endpoint privilege on endpoints:

- a. In the **Access on Endpoints** area, select the endpoints for which you want to revoke the Manage Endpoint privilege.

- b. Click **Remove**.
- c. Click **OK** to confirm in the dialog box. After changes are successfully saved, the control stays in the **Edit LDAP Group Mapping** page.

To revoke the Manage Endpoint Group privilege on endpoint groups:

- a. In the Access on Endpoint Groups area, select the endpoint groups for which you want to revoke the Manage Endpoint Group privilege.
- b. Click **Remove**.
- c. Click **OK** to confirm in the dialog box. After changes are successfully saved, the control stays in the Edit LDAP Group Mapping page.

- **User Groups:** Optionally, you may grant or revoke the wallet access by adding or removing the user groups mapping of the LDAP group. You need the Key Administrator role to map the user groups.

To add the Oracle Key Vault user group mapping:

- a. In the **User Groups** area, select from the available Oracle Key Vault user groups to associate with this LDAP group.
- b. Click **Save**

To revoke the Manage Endpoint Group privilege on endpoint groups:

- a. In the **User Groups Mapped** area, select the user groups for which you want to remove the mapping.
- b. Click **Remove User Groups**.

Optionally, select the Details button of the Oracle Key Vault group to modify the user group's settings and privileges.

Oracle Key Vault determines the LDAP user's authorization for the current session only at the login time. During the login process, Oracle Key Vault fetches the user's LDAP groups from the LDAP directory server and then determines the mapped Oracle Key Vault administrative roles, endpoint or endpoint group privileges, and user groups for the current session. Any changes to the user's membership in the LDAP groups or changes to the LDAP group mapping do not affect user's authorization for existing sessions. Note, however, that any changes to the privileges that are granted to the Oracle Key Vault user groups take effect immediately and apply to all existing sessions.

Validating LDAP Group Mappings

In the event that LDAP groups change in the LDAP directory server, a user who has the Key Administrator role can validate their mappings in Oracle Key Vault.

In a multi-master cluster environment, the validation of an LDAP group mapping applies to all nodes in the cluster and can be performed in any node.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, then **Manage LDAP Mappings** from the left navigation bar.
3. Under LDAP Group Mappings, select the check boxes for the group mappings that you want to validate.
4. Select the **Validate** button.
5. In the confirmation window, click **OK**.

Deleting LDAP Group Mappings

A user who has the Key Administrator role can delete one or more LDAP groups and associated mappings from Oracle Key Vault.

In a multi-master cluster environment, the LDAP group mapping deletion applies to all nodes in the cluster and can be performed in any node.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, then **Manage LDAP Mappings** from the left navigation bar.
3. In the LDAP Group Mappings page, select the check boxes for the LDAP groups that you want to delete.
4. Click **Delete**.
5. In the confirmation window, click **OK**.

Managing Oracle Key Vault-Generated LDAP Users

You cannot administer the actual LDAP user account in the LDAP directory server but you can administer the Oracle Key Vault generated user account that is created the first time the LDAP user logs in to Oracle Key Vault.

- [About Managing LDAP Users](#)
The LDAP user account in Oracle Key Vault is an automatically created account that is based on the LDAP user account in the configured LDAP directory server.
- [Finding Information About an Oracle Key Vault-Generated LDAP User](#)
You can find information about the Oracle Key Vault-generated LDAP user accounts.
- [Validation of Oracle Key Vault-Generated LDAP Users](#)
You can find if an LDAP user account that is associated with the Oracle Key Vault-generated LDAP user account is a valid account.
- [Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges](#)
Users who have either the Key Administrator role or regular users who have privileges to manage wallets can modify the wallet privileges of Oracle Key Vault-generated LDAP user account.
- [Deleting Oracle Key Vault-Generated LDAP Users](#)
A user who has the System Administrator role can delete an LDAP user account from Oracle Key Vault.

About Managing LDAP Users

The LDAP user account in Oracle Key Vault is an automatically created account that is based on the LDAP user account in the configured LDAP directory server.

Oracle Key Vault creates this account the first time that the LDAP user logs in to Oracle Key Vault, capturing the first name, last name, and email attributes of the user. These values cannot be changed in Oracle Key Vault; they can only be changed in their LDAP directory server corresponding account by a privileged LDAP administrator. If these values change, then Oracle Key Vault updates the user account with these values the next time the LDAP user logs in to Oracle Key Vault. Except for granting and revoking wallet privileges to and from this user from Oracle Key Vault, the Oracle Key Vault administrator cannot make any changes to this account.

In a multi-master cluster environment, there is no need for user name conflict resolution because the uniqueness of the account is guaranteed by the LDAP directory server where the LDAP user account exists. If the LDAP user logs in to different nodes in the cluster, then an identical user account is created, and this account is uniform across the cluster. Each of these account creations is timestamped. The Oracle Key Vault synchronization process keeps the most recent account creation timestamp value (that is, from the node where this user was created last). Hence, throughout the cluster environment, the timestamp value is the same as the most recent user account creation timestamp.

Finding Information About an Oracle Key Vault-Generated LDAP User

You can find information about the Oracle Key Vault-generated LDAP user accounts.

You cannot change the LDAP user account in Oracle Key Vault; instead you must modify the account in the LDAP directory server where the user account exists. If you want to move the user to a different LDAP group, you must do this in the LDAP directory server.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator, Key Administrator, or Audit Manager role.
2. Select the **Users** tab, and then **Manage Users** from the left navigation bar.
3. In the Manage Users page, scroll down to Manage LDAP Users area. You can find the following information for each LDAP user.

You can find the following information:

- The user's distinguished name (DN)
 - User status
 - Setup Time
 - LDAP Configuration Name
 - Domain Name
4. Click the **user name** of an LDAP user to see its detailed information including granted administrative roles, endpoint or endpoint group privileges, and user group mappings.

Validation of Oracle Key Vault-Generated LDAP Users

You can find if an LDAP user account that is associated with the Oracle Key Vault-generated LDAP user account is a valid account.

- [About the Validation of Oracle Key Vault-Generated LDAP Users](#)
An Oracle Key Vault-generated user account still exists in Oracle Key Vault if the LDAP user account has been deleted in the source LDAP directory server.
- [Validating Oracle Key Vault-Generated LDAP Users](#)
A user who has the System Administrator role can manually validate Oracle Key Vault-Generated LDAP users.

About the Validation of Oracle Key Vault-Generated LDAP Users

An Oracle Key Vault-generated user account still exists in Oracle Key Vault if the LDAP user account has been deleted in the source LDAP directory server.

A user who has the System Administrator role can find if the Oracle Key Vault-generated user account still exists in the source LDAP directory server by validating it in Oracle Key Vault. In a

multi-master cluster environment, the validation of an Oracle Key Vault-Generated LDAP user account applies to all nodes in the cluster.

Oracle Key Vault periodically checks the validity of the LDAP user accounts and marks them as `NOT FOUND` if the following events take place:

- The LDAP user account does not exist in the LDAP directory server.
- The LDAP configuration that is associated with the LDAP user account is deleted.

Oracle Key Vault automatically deletes invalid LDAP user accounts after the number of days configured in the **Defunct LDAP Users Grace Period** setting (in the Edit LDAP Configuration page) have passed. You can delete an LDAP user account from Oracle Key Vault any time.

Validating Oracle Key Vault-Generated LDAP Users

A user who has the System Administrator role can manually validate Oracle Key Vault-Generated LDAP users.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Users** tab, then **Manage Users** from the left navigation bar.
3. In the Manage Users page, scroll down to the Manage LDAP Users section.
4. Select the check box of the LDAP users that you want to validate.
5. Click **Validate**.

If an LDAP account is not valid, then a `NOT FOUND` message appears.

Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges

Users who have either the Key Administrator role or regular users who have privileges to manage wallets can modify the wallet privileges of Oracle Key Vault-generated LDAP user account.

- [About Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges](#)
The wallet privileges that you can change are Read Only, Read and Modify, or Manage Wallet.
- [Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges \(Key Administrators\)](#)
A user who has the Key Administrator role can grant and revoke wallet privileges for any wallet to LDAP users in Oracle Key Vault.
- [Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges \(Regular Users\)](#)
A regular user who has privileges to manage wallets can grant and revoke privileges for these wallets to LDAP users in Oracle Key Vault.

About Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges

The wallet privileges that you can change are Read Only, Read and Modify, or Manage Wallet.

You cannot change the corresponding LDAP account in the LDAP directory, but you can change the wallet privileges of the Oracle Key Vault-generated LDAP user account. Changes to the privileges granted directly to the LDAP user account in Oracle Key Vault are applied immediately, even to the existing sessions of the same user. If the LDAP user account is modified on the LDAP server (such as a change in LDAP group membership of the user), then the changes take effect from the next user login. In a multi-master cluster environment, changes to an LDAP user apply to all nodes in the cluster and can be performed in any node.

Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges (Key Administrators)

A user who has the Key Administrator role can grant and revoke wallet privileges for any wallet to LDAP users in Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, then **Manage Users** from the left navigation bar.
3. Under Manage Users, scroll down to the Manage LDAP Users section.
4. Select the name of the LDAP user account to display the LDAP User Details page.
5. In the Access to Wallets pane, do the following:
 - a. Click **Add** to display the Add Access to User page.
 - b. Under Select Wallet, select the wallets to which you want to grant privileges to the LDAP user.
 - c. Under Select Access Level, select **Read Only**, **Read and Modify**, or **Manage Wallet**.
 - d. Click **Save**.

The wallet privilege is added as a direct privilege for this user, as opposed to a wallet privilege that is available through an LDAP group. If the user already has a wallet privilege from the LDAP group to which they are assigned, then the user has a union of the privileges from both the direct privilege grant and the LDAP group privilege grant.

6. Click **Save**.

Modifying an Oracle Key Vault-Generated LDAP User Account Wallet Privileges (Regular Users)

A regular user who has privileges to manage wallets can grant and revoke privileges for these wallets to LDAP users in Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user who has privileges to manage wallets.
2. Select the **Keys & Wallets** tab, and then **Wallets** from the left navigation bar.

The Wallets page lists the wallets for which this user has privileges.
3. Select the **Edit** icon for the wallet whose privileges you want to modify.

The Wallet Access Settings area lists all the users who have privileges for this wallet.
4. In the Wallet Access Settings area, click **Add**.
5. In the Add Access to Wallets page, under Select Endpoint/User Group, select **Users** from the **Type** menu.

6. Select the check boxes for the user to whom you want to grant privileges.
7. In the Select Access Level area, select **Read Only**, **Read and Modify**, or **Manage Wallet**.
8. Click **Save**.

The wallet privilege is added as a direct privilege for this user, as opposed to a wallet privilege that is available through an LDAP group. If the user already has a wallet privilege from the LDAP group to which they are assigned, then the user has a union of the privileges from both the direct privilege grant and the LDAP group privilege grant.

Deleting Oracle Key Vault-Generated LDAP Users

A user who has the System Administrator role can delete an LDAP user account from Oracle Key Vault.

If an LDAP user account is deleted from the LDAP directory server, during a periodic check, Oracle Key Vault automatically first marks such user accounts as invalid (`NOT FOUND`) and then deletes these accounts after the **Defunct LDAP Users Grace Period** setting (on the Edit LDAP Configuration page) passes.

If you inadvertently delete an Oracle Key Vault-generated LDAP user account, then the next time the user logs in, the account is recreated. However, the user would no longer own any objects that they created before the deletion. In a multi-master cluster environment, the removal of an Oracle Key Vault-Generated LDAP user account applies to all nodes in the cluster and can be performed in any node.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Users** tab, then **Manage Users** from the left navigation bar.
3. In the Manage Users page, scroll down to the Manage LDAP Users section.
4. Select the check box of the LDAP users that you want to remove.
5. Click **Delete**.
6. In the confirmation window, click **OK**.

The user account is deleted immediately.

9

Managing Oracle Key Vault Users

Oracle Key Vault users administer the system, enroll endpoints, manage users and endpoints, control access to security objects, and grant other users administrative roles.

- [Managing User Accounts](#)
You can create Oracle Key Vault user accounts, grant these users Key Vault administrative roles, endpoint and endpoint group privileges, and add the users to user groups. You can also grant users privileges for managing endpoints and endpoint groups.
- [Managing Administrative Roles and User Privileges](#)
Oracle Key Vault has predefined roles and privileges that you can grant to (or change) or revoke from users.
- [Managing User Passwords](#)
You or the user can change the user's password. You also can have passwords reset automatically.
- [Managing User Email](#)
Oracle Key Vault users should have their current email on file so that they can receive alerts such as system changes.
- [Managing User Groups](#)
You can organize users who have a common purpose into a named user group.
- [Managing support and root Password](#)
Using SSH on Oracle Key Vault server, you can change the support or `root` password.

Managing User Accounts

You can create Oracle Key Vault user accounts, grant these users Key Vault administrative roles, endpoint and endpoint group privileges, and add the users to user groups. You can also grant users privileges for managing endpoints and endpoint groups.

- [About Oracle Key Vault User Accounts](#)
Oracle Key Vault user functionality provides multiple functionalities, such as registering and enrolling endpoints.
- [User Account Profile Parameters](#)
You can configure user account profile parameters to apply certain rules for the user passwords and user account lockout behavior of Oracle Key Vault local users. For LDAP users, the user account management policies are managed in the LDAP directory server.
- [How a Multi-Master Cluster Affects User Accounts](#)
An Oracle Key Vault multi-master cluster environment affects users in various ways.
- [Creating an Oracle Key Vault User Account](#)
A user with the System Administrator role can create user accounts from the Oracle Key Vault management console.
- [Viewing User Account Details](#)
All administrative users can view the list of Oracle Key Vault user accounts and their details.

- [Deleting an Oracle Key Vault User Account](#)
Deleting an Oracle Key Vault user removes the user from any user groups the user was part of in Oracle Key Vault.

About Oracle Key Vault User Accounts

Oracle Key Vault user functionality provides multiple functionalities, such as registering and enrolling endpoints.

An important user function is to register and enroll Oracle Key Vault endpoints, enabling the user to manage his or her security objects by using Oracle Key Vault.

There are three types of Oracle Key Vault users:

- Administrative users who have one or more of the three administrative roles: System Administrator, Key Administrator, or Audit Manager
- Users who have any of the following privileges: Create Endpoint, Manage Endpoint, Create Endpoint Group, Manage Endpoint Group, or Monitor privilege
- Ordinary users who have none of the administrative roles, but who have access to security objects

Separation of duties in Oracle Key Vault means that users with an administrative role or privilege have access to functions pertaining to their role or privilege, but not other roles or privileges. For example, only a user with the System Administrator role has access to the full **System** tab, not users with the Key Administrator or Audit Manager roles. A user who has the Key Administrator role or a user with the Create Endpoint Group privilege can create endpoint groups (but cannot create endpoints). The user interface elements required to create endpoint groups are visible only to the users who have the privileges for creating endpoint groups.

Users who have no administrative role can be granted access to security objects that are specific to their function. For example, you can grant a user access to a specific virtual wallet. This user can log into the Oracle Key Vault management console and add, manage, and delete his or her own security objects, but he or she cannot see system menus, details of other users and endpoints, their wallets, or audit reports.

Although the separation of user duties is recommended, you can have a single user perform all the administrative functions by granting that user all the administrative roles.



Note:

You can enable the **Enforce Separation of Administrator Roles** option if you wish to enforce the separation of user duties.

Oracle Key Vault does not permit the user name to be the same as the name of another user or an endpoint. If you are creating users in a multi-master cluster environment, then there is a chance that user with the same name will be created in another node at the same time. In that case, Oracle Key Vault checks for naming conflicts and will automatically rename the user account that was created after the first user account of that name. You must either accept the generated name for the second user or drop the user and then recreate it with a different name.

Oracle Key Vault allows you to configure certain parameters affecting the user passwords and user lockout behavior.

User Account Profile Parameters

You can configure user account profile parameters to apply certain rules for the user passwords and user account lockout behavior of Oracle Key Vault local users. For LDAP users, the user account management policies are managed in the LDAP directory server.

- [About User Account Profile Parameters](#)
User account profile parameters govern the rules and requirements for the user passwords, and account lockout behavior of Oracle Key Vault local users.
- [Managing User Account Profile Parameters](#)
You can manage the user account profile parameters in Oracle Key Vault.

About User Account Profile Parameters

User account profile parameters govern the rules and requirements for the user passwords, and account lockout behavior of Oracle Key Vault local users.

You can configure user account profile parameters, as described in the following table, to best meet your corporate security requirements:

Table 9-1 User Account Profile Parameters

Parameter	Description	Default Value
Failed Login Attempts	Number of consecutive failed attempts to log in to the user account before the account is locked.	3
Password Life Time (in days)	Number of days the same password can be used for authentication.	180
Password Grace Time (in days)	Number of days after the grace period begins during which a warning is issued, and login is allowed.	5
Password Reuse Max	Number of password changes required before the current password can be reused.	UNLIMITED
Password Reuse Time (in days)	Number of days before which a password cannot be reused.	365
Password Lock Time (in days)	Number of days an account will be locked after the specified number of consecutive failed login attempts. After the time passes, then the account becomes unlocked.	1

Managing User Account Profile Parameters

You can manage the user account profile parameters in Oracle Key Vault.

You can modify the User Accounts profile parameters using Oracle Key Vault management console.

1. From a web browser using HTTPS, enter the IP address of the Oracle Key Vault server.
2. Do not log in to Oracle Key Vault.
3. Click the **System Recovery** link at the lower right corner of the login page.
A new login page appears with a single field: **Recovery Passphrase**.
4. In the **Recovery Passphrase** field, enter the *recovery passphrase* .
5. Click **Login**.
6. Select the **Account Management** tab from the page displayed.

The **User Account Profile Parameters** pane appears along with the **Administrator Management** and **Manage User Administrator Roles** pane.

The screenshot shows the Oracle Key Vault Administrator Management console. It features three main panes:

- Administrator Management:** Contains a toggle for "Enforce Separation of Administrator Roles" and a "Save" button.
- Manage User Administrator Roles:** Includes a search field for "User Name", a "Search" button, and radio buttons for roles: "Audit Manager", "Key Administrator", and "System Administrator". It also has a "STIG Mode" dropdown set to "Disabled" and a "Notify" button.
- User Account Profile Parameters:** A table of parameters with input fields and "Save Defaults" and "Save" buttons.

Parameter	Value
Failed Login Attempts	3
Password Life Time (in days)	90
Password Grace Time (in days)	5
Password Reuse Max	UNLIMITED
Password Reuse Time (in days)	365
Password Lock Time (in days)	1

7. Enter the desired parameter values for the required fields.
8. Click **Save**.

Clicking **Save Defaults** in the **User Account Profile Parameters** pane restores the profile parameters to their default values.

Note:

Profile parameter values cannot be modified when the recovery passphrase change is in progress.

How a Multi-Master Cluster Affects User Accounts

An Oracle Key Vault multi-master cluster environment affects users in various ways.

These can include expanding the activities that they can perform and ensuring that their names do not conflict with other objects in the cluster environment.

- [Multi-Master Cluster Effect on User Account Profile Parameters](#)
In a multi-master cluster environment, when you modify user account profile parameters from the Oracle Key Vault management console, the change is applied to all the cluster nodes.
- [Multi-Master Cluster Effect on System Administrator Users](#)
The user who is granted the System Administrator role is responsible for managing the cluster configuration.

- [Multi-Master Cluster Effect on Key Administrator Users](#)
The user who is granted the Key Administrator role manages endpoint groups, user groups, wallets, and objects.
- [Multi-Master Cluster Effect on Audit Manager Users](#)
The user who is granted the Audit Manager role is responsible for configuring audit settings and integration with Oracle Audit Vault.
- [Multi-Master Cluster Effect on Administration Users](#)
Administrative users can have any combination of the administration roles, including the System Administrator, Key Administrator, and Audit Manager roles.
- [Multi-Master Cluster Effect on System Users](#)
System users are responsible for the operating system of each Oracle Key Vault appliance, server, and node.

Multi-Master Cluster Effect on User Account Profile Parameters

In a multi-master cluster environment, when you modify user account profile parameters from the Oracle Key Vault management console, the change is applied to all the cluster nodes.

User account profile parameters govern the rules and requirements for the user passwords, and account lockout behavior of Oracle Key Vault local users.

In a multi-master cluster environment, when you modify user account profile parameters from the Oracle Key Vault management console, the change is applied to all the cluster nodes. However, these account profile settings are enforced on each cluster node independently. When a user account status changes on a cluster node, the user account status on other nodes remains unaffected. For example, if a user account gets locked on a cluster node, it may still remain unlocked on other cluster nodes.

This behavior applies only to the Oracle Key Vault local users. For LDAP users, the account status is managed externally in the LDAP Directory and remains consistent across cluster nodes.



Note:

You cannot modify user account profile parameters when the cluster upgrade is in-progress.

Multi-Master Cluster Effect on System Administrator Users

The user who is granted the System Administrator role is responsible for managing the cluster configuration.

The System Administrator role in a multi-master cluster includes the following responsibilities:

- All system administrator responsibilities for a single Oracle Key Vault server
- Cluster initialization, converting the first Oracle Key Vault server to the initial node
- Adding and removing nodes from the cluster
- Disabling and enabling nodes in the cluster
- Managing cluster-wide system settings
- Monitoring cluster operations and cluster health indicators
- Enabling and disabling replication between nodes

- Monitoring and resolving data and naming conflicts
- Monitoring and reacting to cluster alerts
- Managing cluster settings

The user who has the System Administrator privilege can also create and then manage endpoints. A user with the Create Endpoint privilege can create his or her own endpoints, and a user with the Manage Endpoint privilege can manage his or her own endpoints.

Multi-Master Cluster Effect on Key Administrator Users

The user who is granted the Key Administrator role manages endpoint groups, user groups, wallets, and objects.

In a multi-master cluster, when these items are uploaded in separate nodes and in separate data centers, name conflicts can occur. The key administrator provides input to the system administrator to resolve these conflicts for wallets, KMIP objects, endpoint groups, and user groups.

A user with the Create Endpoint Group privilege can create his or her own endpoint groups, and a user with the Manage Endpoint Group privilege can manage his or her own endpoint groups.

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

Multi-Master Cluster Effect on Audit Manager Users

The user who is granted the Audit Manager role is responsible for configuring audit settings and integration with Oracle Audit Vault.

In a multi-master cluster environment, this user can configure audit settings for the entire cluster and for individual nodes. The audit manager user can use different setting for different nodes, if necessary. However, this user can also unify audit settings across the entire cluster.

The audit manager can replicate audit trails between nodes, if necessary. However, this can result in significant traffic between nodes, so the audit manager can turn on or off the audit trail replication. By default, the audit trails replication is turned off.

Multi-Master Cluster Effect on Administration Users

Administrative users can have any combination of the administration roles, including the System Administrator, Key Administrator, and Audit Manager roles.

Administrative user information created in the Oracle Key Vault server that is used as the initial node seeds the cluster.

New servers added to a cluster will get administrative user information from the cluster. Administrator information that is created on the server for the purpose of inducting the server into the cluster will be removed.

Administrative users that are created in a node after the node joins an Oracle Key Vault cluster will have a cluster-wide presence. New administrative users that are added to the Oracle Key Vault cluster on different Oracle Key Vault nodes may have name conflicts. When the user account is created, Oracle Key Vault automatically resolves the administrative user name conflicts. User and endpoint conflicts will be displayed in the Conflicts Resolution page and

administrators can choose to rename endpoint conflicts. If there is a user name conflict, then you must either accept the automatically generated user name, or delete and recreate the user. User accounts will not be available for use and will be placed in a `PENDING` state until the name resolution is completed. You cannot delete the user accounts in `PENDING` state.

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

Multi-Master Cluster Effect on System Users

System users are responsible for the operating system of each Oracle Key Vault appliance, server, and node.

Oracle Key Vault servers are first installed or later configured to become nodes of an Oracle Key Vault cluster. As part of the server configuration, the operating system users (`support` and `root`) are created. Those users will remain unchanged after the server joins a cluster.

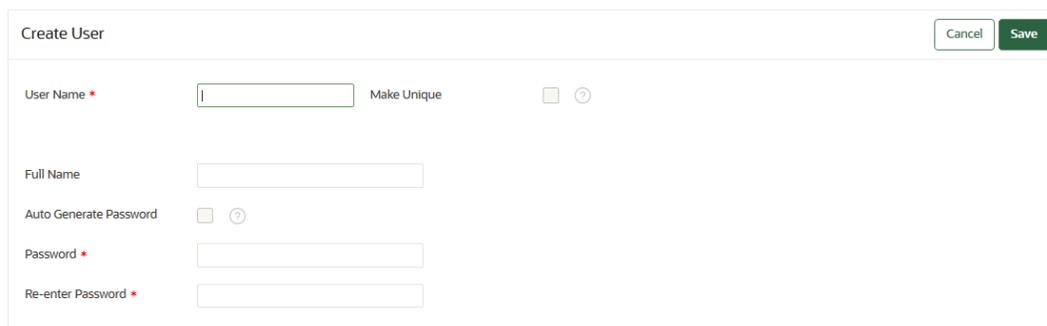
Unless security requirements specify otherwise, the same `support` and `root` passwords should be used for all the Oracle Key Vault nodes. Unlike Oracle Key Vault administrative accounts that are replicated, the `support` and `root` accounts are operating system users, and their passwords are not automatically synchronized across the cluster. Therefore, each node can potentially have a different `support` or `root` user password, making it difficult to manage multiple nodes of the cluster.

Creating an Oracle Key Vault User Account

A user with the System Administrator role can create user accounts from the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.
The Manage Users page appears with a list of existing users.
3. In the Manage Users page, click **Create**.

The Create User page appears.



The screenshot shows a 'Create User' form with the following fields and options:

- User Name ***: A text input field with a 'Make Unique' checkbox and a refresh icon.
- Full Name**: A text input field.
- Auto Generate Password**: A checkbox with a help icon.
- Password ***: A text input field.
- Re-enter Password ***: A text input field.

Buttons for 'Cancel' and 'Save' are located in the top right corner.

4. Enter a user name in **User Name**.
See [Naming Guidelines for Objects](#). Ensure that the user name is not the same as an Oracle Key Vault endpoint name.

5. If you are using a multi-master cluster, then choose whether to select the **Make Unique** check box.

Make Unique helps to control naming conflicts with user names across the multi-master cluster environment. When a server is converted to a cluster node, then the character limit for user names drops from 128 to 120 to allow for automatic renaming in case of a conflict. Users that were created before an Oracle Key Vault conversion to a cluster node are not affected by naming conflicts.

- If you select **Make Unique**, then the user account will be active immediately and this user can perform operations.
 - If you do not select **Make Unique**, then the user account will be created in the `PENDING` state. Oracle Key Vault will then begin a name resolution operation and may rename the user account to a name that is unique across the cluster. If there is a naming collision, then the collision will be reported on the Conflicts page on any node in the cluster. The user account will then be renamed to a unique name. You will need to go to a read/write node of the cluster and either accept the renamed user account or change the user account name. If you change the user account name, then this will restart the name resolution operation and the user account will return to a `PENDING` state. A user account in the `PENDING` state cannot be used to perform most operations.
6. Optionally, add the user's full name in **Full Name**.
 7. For the password, do one of the following:

- **Auto Generate Password:** Select this option to have a password automatically generated and sent to the user. The user will receive an email message with `Oracle Key Vault: System Generated User Password` in the subject line. When the user logs in to the Oracle Key Vault management console for the first time, he or she will be asked to change the password.

The SMTP server configuration must be configured to use this option.

- **Password and Re-enter password:** Enter a valid password. The password must be between 8 and 30 characters in length and contain at least one of each of the following: an uppercase letter, a lowercase letter, a number, and a special character from the set: period (`.`), comma (`,`), underscore (`_`), plus sign (`+`), colon (`:`), exclamation mark (`!`). In addition, the password may include a space character () provided it is not used as the first or last character of the password.
8. Click **Save**.

The **Manage Users** page appears and lists the new user. If the user is in the `PENDING` state, then it remains in the Users being created section until it transitions to the `ACTIVE` state, similar to the following example.

Manage Users Create

Q Go Actions

<input type="checkbox"/>	User Name	Full Name	System Admin	Key Admin	Audit Manager	Creation Time	Created By	Creator Node
<input type="checkbox"/>	RICKY					09-NOV-2020 15:12:40	OKVADMIN	node1
<input type="checkbox"/>	ANDY					09-NOV-2020 15:12:15	OKVADMIN	node1
<input type="checkbox"/>	SCOTT					09-NOV-2020 15:11:51	OKVADMIN	node1
<input type="checkbox"/>	SIMRAN					09-NOV-2020 15:11:17	OKVADMIN	node1
<input type="checkbox"/>	MIKE					09-NOV-2020 15:10:58	OKVADMIN	node1
<input type="checkbox"/>	BOB					09-NOV-2020 15:10:32	OKVADMIN	node1
<input type="checkbox"/>	OKVADMIN		✓	✓	✓	09-NOV-2020 08:53:47	SAM_SYSADMIN	node1
<input type="checkbox"/>	ALICE_AUDADMIN				✓	09-NOV-2020 08:49:28		node1
<input type="checkbox"/>	SAM_SYSADMIN		✓			09-NOV-2020		node1

1 - 14 of 14

Pending Users Check Conflict Status

User Name	Full Name	Created By	Creator Node
NANCY	Nancy Smith	OKVADMIN	node1

row(s) 1 - 1 of 1

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.
- [Managing Administrative Roles and User Privileges](#)
Oracle Key Vault has predefined roles and privileges that you can grant to (or change) or revoke from users.
- [Configuring Email Notification](#)
You can use email notifications to directly notify administrators of Key Vault status changes without logging into the Oracle Key Vault management console.

Viewing User Account Details

All administrative users can view the list of Oracle Key Vault user accounts and their details.

Users without any of the three administrative roles can only see their own user details. The **User Details** page provides a consolidated view of the Oracle Key Vault user. This is the page where all user management tasks are performed.

1. Log in to the Oracle Key Vault management console.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.

The **Manage Users** page appears with a list of existing users. You can sort and search the list by the column user name, full name, or roles.

3. Click a user name to display the **User Details** page.

Related Topics

- [Administrative Roles and Privilege within Oracle Key Vault](#)
Oracle Key Vault provides separation of duty compliant administrative roles and privileges that you can combine in various ways to meet enterprise needs.

- *Oracle Key Vault Installation and Upgrade Guide*

Deleting an Oracle Key Vault User Account

Deleting an Oracle Key Vault user removes the user from any user groups the user was part of in Oracle Key Vault.

The operation does not delete any security objects managed by the user. Administrators can only delete users that are not in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role and the same roles as the user being deleted, should that user have any administrative roles.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.
The **Manage Users** page appears with a list of existing users.
3. Select the check boxes for the users that you want to delete.
4. Click **Delete**.
5. In the confirmation dialog box, click **OK**.
6. Click **Save**.

Managing Administrative Roles and User Privileges

Oracle Key Vault has predefined roles and privileges that you can grant to (or change) or revoke from users.

- [About Managing Administrative Roles and User Privileges](#)
You can grant or change an administrative role or user privileges for a user account that you have added.
- [Granting or Changing an Administrative Role of a User](#)
You can use the Manage Users page to grant or change a user administrative role.
- [Granting the Create Endpoint Privilege](#)
The Create Endpoint privilege enables a user to create the user's own endpoints.
- [Granting the Manage Endpoint Privilege](#)
The Manage Endpoint privilege enables a user to manage the user's own endpoints.
- [Granting the Create Endpoint Group Privilege](#)
The Create Endpoint Group privilege enables a user to create the user's own endpoint groups.
- [Granting the Manage Endpoint Group Privilege](#)
The Manage Endpoint Group privilege enables a user to manage the user's own endpoint groups.
- [Granting the Monitor Privilege](#)
The Monitor privilege enables a user to view system information and metrics of an Oracle Key Vault server or node. You can grant the Monitor privilege to a user when all systems in the Oracle Key Vault deployment are at release 21.10.0.0.0 or later.
- [Revoking an Administrative Role or Privilege from a User](#)
You can use the Manage User page to revoke a role or a privilege from a user.

- [Granting a User Access to a Virtual Wallet](#)
A user with the Key Administrator role controls access to security objects for users, endpoints, and their respective groups.
- [Enforce Separation of Administrator Roles](#)
You can use the recovery passphrase option to enforce the separation of administrator roles.

About Managing Administrative Roles and User Privileges

You can grant or change an administrative role or user privileges for a user account that you have added.

You must be a user with the System administrative role to grant, change, or revoke the Create Endpoint, Manage Endpoint, or Monitor privileges to or from other users. You must be a user with the Key Administrative role to grant, change, or revoke the Create Endpoint Group and Manage Endpoint Group privileges to or from other users. You can also revoke the privilege when it is no longer needed. Users with the Create Endpoint, Manage Endpoint, Monitor, Create Endpoint Group, and Manage Endpoint Group privilege cannot grant this privilege to (or revoke it from) other users.

If you are using a multi-master cluster environment, then you cannot grant, change, and revoke administrative roles for users in the `PENDING` state.

If you are using a multi-master cluster environment, then you cannot grant, change, and revoke user privileges for users in the `PENDING` state.



Note:

If the **Enforce Separation of Administrator Roles** option is enabled in your Oracle Key Vault environment, then you cannot grant more than one administrative role to a user. You must revoke all but one administrative role from any user who holds multiple roles when this option is enabled. Users who hold multiple administrative roles while the **Enforce Separation of Administrator Roles** option is enabled cannot exercise any of those roles until all but one have been revoked. The **Enforce Separation of Administrator Roles** option is disabled by default.

Granting or Changing an Administrative Role of a User

You can use the Manage Users page to grant or change a user administrative role.

1. Log in to the Oracle Key Vault management console as a user who has the same role that was granted to them with the **Allow Forward Grant** option.

For example, if the user needs the System Administrator role, the granting user should have the same role with the **Allow Forward Grant** option.

2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.

The Manage Users page appears displaying the list of users.

	User Name	User Type	Full Name	System Admin	Key Admin	Audit Manager	Creation Time
<input type="checkbox"/>	OKVADMIN	Local		✓	✓	✓	08-JAN-2024 10:49:39

3. Click the name of the user in the **User Name** column.

The User Details page appears. The User Details page provides a consolidated view of the Oracle Key Vault user. It displays the following user information: user name, email, administrative role, user privileges, membership in user groups, endpoints that the user has the Manage Endpoint privilege on, endpoint groups that the user has Manage Endpoint Group privilege on, and access to wallets.

The screenshot displays the 'User Details' page for the user 'OKV_MONITOR'. At the top right, there are buttons for 'Cancel', 'Reset Password', and 'Save'. The 'User Details' section includes:

- User Name: OKV_MONITOR
- Roles: Audit Manager, Key Administrator, System Administrator (all unchecked)
- Privileges: Create Endpoint, Create Endpoint Group (unchecked), Monitor (checked)
- Full Name: OKV_Monitor

 Below this are two panels: 'User Group Membership' (showing 'Not part of any user group.') and 'Access to Wallets' (showing 'No access to wallets.'). Both panels have search and action buttons.

4. To grant a role, check the **Roles** box for the role that you want to grant.

To change a role, uncheck the box for the previous role and check the box by the new role. If you do not see the role listed that you want to grant, then you are logged in as a user who does not have that role and therefore do not have the privilege to grant it.

5. To allow this user to grant the role to other users, check the **Allow Forward Grant** box. This option only shows after you select the role.

By default, a user cannot grant roles to or revoke roles from other users. In order for a user to grant or revoke a role from another user, you must select the **Allow Forward Grant** option.

6. Click **Save**.

You cannot grant or revoke administrator roles to an LDAP user directly. LDAP users can be granted administrative roles only through LDAP group mappings.

Related Topics

- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Granting the Create Endpoint Privilege

The Create Endpoint privilege enables a user to create the user's own endpoints.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

Users who have the Create Endpoint privilege cannot grant it to other users.

2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.
The Manage Users page appears displaying the list of users.
3. Select the user to whom you want to grant the Create Endpoint privilege.
4. Under User Details, select the **Create Endpoint** check box.
5. Click **Save**.

 **Note:**

When a local Oracle Key Vault user with the Create Endpoint privilege creates an endpoint, Oracle Key Vault grants the Manage Endpoint privilege on that endpoint to the local user.

Related Topics

- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Granting the Manage Endpoint Privilege

The Manage Endpoint privilege enables a user to manage the user's own endpoints.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
Users who have the Manage Endpoint privilege cannot grant it to other users.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.
The Manage Users page appears displaying the list of users.
3. Select the user to whom you want to grant the Manage Endpoint privilege.
4. In the Access on Endpoints area, click **Add**.
5. In the Add Endpoint Access to User page, under **Select Endpoint**, select the endpoint for which you want to grant the user the Manage Endpoint privilege.
6. Click **Save**.

Related Topics

- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Granting the Create Endpoint Group Privilege

The Create Endpoint Group privilege enables a user to create the user's own endpoint groups.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
Users who have the Create Endpoint Group privilege cannot grant it to other users.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.

The Manage Users page appears displaying the list of users.

3. Select the user to whom you want to grant the Create Endpoint Group privilege.
4. Under User Details, select the **Create Endpoint Group** check box.
5. Click **Save**.

Related Topics

- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Granting the Manage Endpoint Group Privilege

The Manage Endpoint Group privilege enables a user to manage the user's own endpoint groups.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.

Users who have the Manage Endpoint Group privilege cannot grant it to other users.

2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.

The Manage Users page appears displaying the list of users.

3. Select the user to whom you want to grant the Manage Endpoint Group privilege.
4. In the Access on Endpoint Groups area, click **Add**.
5. In the Add Endpoint Group Access to User page, in the Select Endpoint Group area, select the endpoint group to which you want to grant the user the Manage Endpoint Group privilege.
6. Click **Save**.

Related Topics

- [Considerations for Granting Privileges to LDAP Users](#)
You can grant privileges to the LDAP users in Oracle Key Vault subject to certain rules and considerations.

Granting the Monitor Privilege

The Monitor privilege enables a user to view system information and metrics of an Oracle Key Vault server or node. You can grant the Monitor privilege to a user when all systems in the Oracle Key Vault deployment are at release 21.10.0.0.0 or later.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

2. Select the **Users** tab and then select **Manage Users** in the left navigation bar.

The Manage Users page appears displaying the list of users.

3. Select the user to grant the Monitor privilege. The User Details page appears.
4. Select the **Monitor** check box in the **Privilege** section.
5. Click **Save**.

Revoking an Administrative Role or Privilege from a User

You can use the Manage User page to revoke a role or a privilege from a user.

- Depending on the administrative role or privilege that you want to revoke, log in to the Oracle Key Vault management console as follows:
 - Administrative roles:** Log in as a user who has the same role with the **Allow Forward Grant** option. You can only grant and revoke roles for which you are an administrator and were given the **Allow Forward Grant** option.
 - Create Endpoint or Manage Endpoint privilege:** Log in as a user who has the System Administrator role.
 - Create Endpoint Group or Manage Endpoint Group privilege:** Log in as a user who has the Key Administrator role.
 - Monitor privilege:** Log in as a user who has the System Administrator role.
- Select the **Users** tab, and then **Manage Users** in the left navigation bar.

The Manage Users page appears displaying the list of users.

	User Name	User Type	Full Name	System Admin	Key Admin	Audit Manager	Creation Time
<input type="checkbox"/>	OKVADMIN	Local		✓	✓	✓	08-JAN-2024 10:49:39

- Click the user name whose role or privilege you want to revoke. The **User Details** page appears.
- Revoke privileges as follows:
 - Administrative roles, or Create Endpoint, Create Endpoint Group, or Monitor privilege:** Deselect the box for the role, endpoint or monitor privilege.
 - Manage Endpoint privilege:** When logged in as a user with the System Administrator role, scroll down to the Access on Endpoint area, select the check box for the endpoint, and then click **Remove**.
 - Manage Endpoint Group privilege:** When logged in as a user with the Key Administrator role, scroll down to the Access on Endpoint Group area, select the check box for the endpoint group, and then click **Remove**.
- Click **Save**.

If you upgraded Oracle Key Vault from a version prior to release 21.4, all administrative users (users with the Audit Manager, Key Administrator, or System Administrator roles) will have the **Allow Forward Grant** option selected. If you do not want these users to have the ability to grant their role to other users, you should immediately revoke this option from that user. From the normal user management pages, you can remove the **Allow Forward Grant** option from all users but one. You must use the **Administrative Management** page to remove the **Allow Forward Grant** option from the last user. The **Administrative Management** page can be accessed using the **System Recovery** option which requires the recovery passphrase.

Related Topics

- Configuring the Use of Password Reset Operations

Granting a User Access to a Virtual Wallet

A user with the Key Administrator role controls access to security objects for users, endpoints, and their respective groups.

Any user can be granted access to security objects in Oracle Key Vault at a level that is appropriate to their function in the organization.

You cannot grant access to a virtual wallet if the wallet is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.
The Manage Users page appears displaying the list of users.
3. Click the name of the user you want to grant access.
The **User Details** page appears.
4. Click **Add** in the **Access to Wallets** section.
The **Add Access to User** page appears.
5. Select the wallet under **Select Wallet**.
6. Set the access level to the selected wallet under **Select Access Level: Read Only, Read and Modify, or Manage Wallet**.

Set access levels when you grant access to the wallet, if you know the level to grant. You can also set or modify access levels from the wallet menu.

7. Click **Save**.

Related Topics

- [Access Control Configuration](#)
Oracle Key Vault enables you to control access to security objects at various access levels.
- [Managing Access to Virtual Wallets from Keys & Wallets Tab](#)
You can grant virtual wallet access to and revoke virtual wallet access from endpoint by using the **Keys & Wallets** tab.

Enforce Separation of Administrator Roles

You can use the recovery passphrase option to enforce the separation of administrator roles.

You can enforce separation of administrator roles and limit users to holding at most one administrator role using the Oracle Key Vault management console.

1. From a web browser using HTTPS, enter the IP address of the Oracle Key Vault server.
2. Do not log in to Oracle Key Vault.
3. Click the **System Recovery** link at the lower right corner of the login page.
A new login page appears with a single field: **Recovery Passphrase**.
4. In the **Recovery Passphrase** field, enter the *recovery passphrase* .
5. Click **Login**.
6. Select the **Account Management** tab from the page displayed.

The **Enforce Separation of Administrator Roles** pane appears.

7. Select the **Enforce Separation of Administrator Roles** option to enable or disable this option.
8. Click **Save**.

Managing User Passwords

You or the user can change the user's password. You also can have passwords reset automatically.

- [About Changing User Passwords](#)
Any valid Oracle Key Vault user can change his or her own password.
- [Changing Your Own Password](#)
Any user can change his or her own Oracle Key Vault account password.
- [Changing Another User's Password](#)
You can change another user's password if you have the System Administrator role or the identical administrative roles (at a minimum) as the user whose password you want to reset.
- [Controlling the Use of Password Reset Methods](#)
You can restrict the ability of users to reset another user's password manually so that only password reset operations through email notifications are allowed.
- [Unlocking a User Account](#)
You can unlock a user account by resetting the user's password.

About Changing User Passwords

Any valid Oracle Key Vault user can change his or her own password.

You can reset the password of another user if you have at a minimum the same administrative roles as that user. For example, if you want to change the password of a user who has the Audit Manager role, then you also must have the Audit Manager role before you can change the password.

Consider the following users and roles:

User	System Admin	Key Admin	Audit Manager
OKV_ALL_JANE	Yes	Yes	Yes
OKV_SYS_KEYS_JOE	Yes	Yes	-
OKV_SYS_SEAN	Yes	-	-
OKV_KEYS_KATE	-	Yes	-
OKV_AUD_AUDREY	-	-	Yes
OKV_OLIVER	-	-	-

Suppose that user `OKV_SYS_KEYS_JOE`, who has the System Administrator and Key Administrator roles, is logged in and wants to change the other users' passwords. The following happens:

- `OKV_KEYS_KATE`: `OKV_SYS_KEYS_JOE` can change the password for `OKV_KEYS_KATE` because they have the Key Administrator role in common.

- OKV_AUD_AUDREY: OKV_SYS_KEYS_JOE cannot change OKV_AUD_AUDREY's password because OKV_SYS_KEYS_JOE does not have the Audit Manager role.
- OKV_ALL_JANE: OKV_SYS_KEYS_JOE cannot change the password for user OKV_ALL_JANE because he does not have the Audit Manager role.
- OKV_OLIVER: OKV_SYS_KEYS_JOE can change the password for user OKV_OLIVER, who has no roles at all.

Any user can change his or her own password.

Assuming you have privileges to do so, you can change the password of another user by using either of the following methods:

- Specify a new password for the other user and then notify this user of the new password by using any out-of-band method.
- Send the user a randomly generated one-time password to their email account.

Related Topics

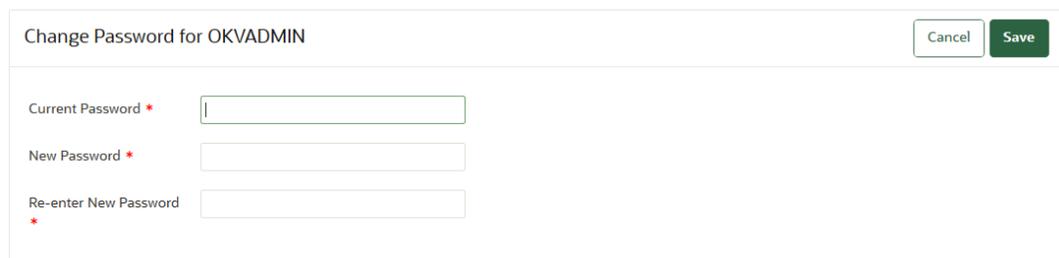
- [Controlling the Use of Password Reset Methods](#)
You can restrict the ability of users to reset another user's password manually so that only password reset operations through email notifications are allowed.

Changing Your Own Password

Any user can change his or her own Oracle Key Vault account password.

1. Log in to the Oracle Key Vault management console.
2. Select the **Users** tab, and then **Change Password** in the left navigation bar.

The Change Password page appears for your account.



Change Password for OKVADMIN

Cancel Save

Current Password *

New Password *

Re-enter New Password *

3. Enter your current password in **Current Password**.
4. Enter the new password in **New Password** and **Re-enter New Password**.
5. Click **Save**.

Changing Another User's Password

You can change another user's password if you have the System Administrator role or the identical administrative roles (at a minimum) as the user whose password you want to reset.

- [Changing a Password Manually](#)
You can change the password manually for a user and then use any out-of-band method to notify the user of the new password.

- [Changing a Password Through Email Notification](#)
You can change a user's password by sending them a randomly generated one-time password to their email account.

Changing a Password Manually

You can change the password manually for a user and then use any out-of-band method to notify the user of the new password.

This method of changing a password is available only when the `Reset passwords using email only` option in the User Password Recovery tab of the System Recovery page is not selected.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.

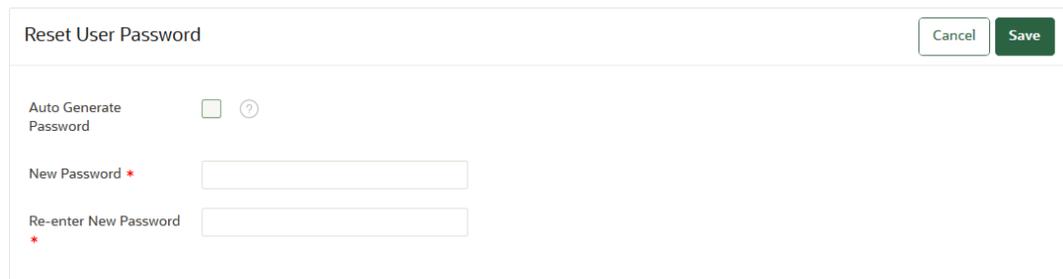
The Manage Users page displays the list of users.

3. Click the user name whose password you want to change.

The **User Details** page appears.

4. Click **Reset Password**.

By default, the Auto Generate Password option is selected. Deselect it so that you can manually change the password.



Reset User Password

Auto Generate Password ?

New Password *

Re-enter New Password *

Cancel Save

If the **Email Address** field appears instead of the password prompts, then the system has been configured to change passwords through email notification only.

5. Enter the new password in **New Password** and **Re-enter New Password**.
6. Click **Save**.

Related Topics

- [About Controlling the Use of Password Reset Methods](#)
You can configure Oracle Key Vault to allow users to change another user's password only by sending them a randomly generated one-time password through email.

Changing a Password Through Email Notification

You can change a user's password by sending them a randomly generated one-time password to their email account.

This one-time password can be sent directly from Oracle Key Vault to the user. You must configure SMTP in email settings in order to use this feature. Oracle recommends that you restrict the password recovery functionality to use this method by selecting the `Reset passwords using email only` option in the User Password Recovery tab of the System Recovery page.

1. Log in to the Oracle Key Vault management console.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.
The Manage Users page appears displaying the list of users.
3. Click the user name of the user whose password you want to change.
The User Details page appears.
4. Click **Reset Password**.
The Reset User Password page appears.
5. Check the box by **Auto Generate Password**.
If SMTP is configured, then an email address field appears.
6. Enter the email address of the user.
7. Click **Send One-Time Password**.

If you check **Auto Generate Password** without configuring SMTP, the link **Click here to configure SMTP** appears. Click the link to configure email settings and repeat the steps in this topic.

Related Topics

- [Controlling the Use of Password Reset Methods](#)
You can restrict the ability of users to reset another user's password manually so that only password reset operations through email notifications are allowed.
- [Configuring Email Settings](#)
You can configure the Simple Mail Transfer Protocol (SMTP) server properties to receive email notifications from Oracle Key Vault.

Controlling the Use of Password Reset Methods

You can restrict the ability of users to reset another user's password manually so that only password reset operations through email notifications are allowed.

- [About Controlling the Use of Password Reset Methods](#)
You can configure Oracle Key Vault to allow users to change another user's password only by sending them a randomly generated one-time password through email.
- [Configuring the Use of Password Reset Operations](#)
A user who has access to the system recovery passphrase can configure the use of password reset operations

About Controlling the Use of Password Reset Methods

You can configure Oracle Key Vault to allow users to change another user's password only by sending them a randomly generated one-time password through email.

The user performing a password change for another user must be either an Oracle Key Vault administrator or have the same or higher privileges as the user whose password needs to be reset.

By default, there are two ways to change another user's password:

- Manually, in which you create a new password for the user. In this scenario, both you and the user will know the password (until this user manually changes his or her own password)

- Automatically, in which you trigger an automatically-generated password for the user, who is then emailed the new password on a one-time basis. In this scenario, only the user knows his or her new password.

You can enable automatic password generation only through email notification and disable manual password reset operations. The email notification uses the email ID that is associated with the user's account. The benefit of this feature is that the newly generated password is known only to the user whose password was reset, not to the user who initiated the user's password change. Users can still change their own passwords when this feature is enabled.

When this feature is disabled, then both methods of user creation are allowed: manual password reset operations and automatic password reset operations.

Configuring the Use of Password Reset Operations

A user who has access to the system recovery passphrase can configure the use of password reset operations

1. Navigate to the Oracle Key Vault management console, but do not log in.
2. Click the **System Recovery** button.
3. When prompted, enter the system recovery passphrase.
4. Select the **User Password Recovery** tab.
5. In the User Password Recovery page, select the **Reset passwords using email only** option to enable or disable this option.
6. Click **Save**.

Related Topics

- [Changing a Password Through Email Notification](#)
You can change a user's password by sending them a randomly generated one-time password to their email account.

Unlocking a User Account

You can unlock a user account by resetting the user's password.

A user account may become locked after multiple failed login attempts.

In a multi-master cluster, the user account profile parameter **Failed Login Attempts** is enforced at each node separately. If a user account becomes locked on a cluster node, the user account may still remain unlocked on other cluster nodes. In such a case, the user may reset their own password from another node or another administrator may reset the user's password. Resetting of the user password unlocks the user account on all cluster nodes.

Related Topics

- [Managing LDAP User Authentication and Authorization in Oracle Key Vault](#)
You can configure a connection between Oracle Key Vault and an LDAP server (currently Microsoft Active Directory) so that their users can access Oracle Key Vault.

Managing User Email

Oracle Key Vault users should have their current email on file so that they can receive alerts such as system changes.

- [Changing the User Email Address](#)
After creating a user account, you can add or change the user's email address.
- [Disabling Email Notifications for a User](#)
You can disable email notifications for a user on the User Details page.

Changing the User Email Address

After creating a user account, you can add or change the user's email address.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar.
The Manage Users page appears displaying the list of users.
3. Click the user's name in the **User Name** column.
The User Details page appears.
4. Enter the email address in **Email**.
5. Click **Save**.

- [Changing Your Own Email](#)
Any user can change their own Oracle Key Vault account email.
- [Changing Another User's Email](#)
After creating a user account, you can change another user's email if you have the System Administrator role and the **Reset passwords using email only** option in the User Password Recovery tab of the System Recovery page is not selected.

Changing Your Own Email

Any user can change their own Oracle Key Vault account email.

1. Log in to the Oracle Key Vault management console.
2. Select the **Users** tab and then **Details** in the left navigation bar.
The Users Details page appears for your account.
3. Enter your new email address in the **Email** field.
4. Click **Save**.

Changing Another User's Email

After creating a user account, you can change another user's email if you have the System Administrator role and the **Reset passwords using email only** option in the User Password Recovery tab of the System Recovery page is not selected.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Users** tab and then **Manage Users** in the left navigation bar.
The Manage Users page appears displaying the list of users.
3. Click the user's name in the **User Name** column.

- The User Details page appears.
4. Enter the email address in the **Email** field.
 5. Click **Save**.

Disabling Email Notifications for a User

You can disable email notifications for a user on the User Details page.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Users** tab and then **Manage Users** in the left navigation bar.
The Manage Users page appears displaying the list of users.
3. Click the user's name in the **User Name** column.
The **User Details** page appears:
4. Select the **Do not receive email alerts** check box.
5. Click **Save**.

Managing User Groups

You can organize users who have a common purpose into a named user group.

- [About Managing User Groups](#)
Users who have the Key Administrator role can create, modify, and delete user groups.
- [How a Multi-Master Cluster Affects User Groups](#)
User groups are used at the Oracle Key Vault server and cluster level to group user roles and permissions.
- [Creating a User Group](#)
You can create a user group when a set of users must manage a set of common security objects.
- [Adding a User to a User Group](#)
You can add an existing user to a user group if that user must manage the same security objects as the group.
- [Granting a User Group Access to a Virtual Wallet](#)
You can modify the access level to a virtual wallet for a user group as functional needs change.
- [Renaming a User Group](#)
Depending on its status, you can change the name of a user group.
- [Changing a User Group Description](#)
A group description is useful for identifying the purpose of the group.
- [Removing a User from a User Group](#)
Depending on the requirement, you can remove a user from a user group.
- [Deleting a User Group](#)
You can delete a user group when the users in the group do not need to access the same security objects.

About Managing User Groups

Users who have the Key Administrator role can create, modify, and delete user groups.

This enables them to manage their access to [virtual wallets](#). After a user group is created, you can modify its details.

The main purpose of a user group is to simplify access control to [security objects](#). If a set of users need access to a common set of security objects, then you can assign these users to a group and grant the group access instead of granting access to each user or each security object. When certain users do not need access to the security objects any longer, you can remove them from the group. You can add new users to the group. You can modify the group's access level to security objects at any time.

How a Multi-Master Cluster Affects User Groups

User groups are used at the Oracle Key Vault server and cluster level to group user roles and permissions.

When new servers are introduced into the cluster, Oracle Key Vault replicates any user group information that is in the cluster. You can create new user groups in the cluster from a read/write pair.

User groups created in a node after the node is added to an Oracle Key Vault cluster will have a cluster-wide presence. User groups created on two different nodes could have name conflicts. Oracle Key Vault automatically resolves the user group name conflicts. These conflicts will be displayed in the Conflicts Resolution page and administrators can choose to rename them.

Note:

- You cannot change membership by adding or removing users when the user group is in a `PENDING` state. Similarly, users in a pending state cannot be added to, or removed from a user group in the `ACTIVE` state.
- You cannot change access mapping for users and user groups if a wallet is in the `PENDING` state. Similarly, users and user groups in a `PENDING` state cannot be added to, or removed from a wallet access mapping even when the wallet is in the `ACTIVE` state.

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

Creating a User Group

You can create a user group when a set of users must manage a set of common security objects.

You can add users to the group when you create the group or later after creating the group.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, and then **Manage Access** in the left navigation bar.

The User Groups page appears displaying existing user groups.

<input type="checkbox"/>	User Group Name	Name Status	Description	Creation Time	Creator Node	Created By	Edit
<input type="checkbox"/>	AUDIT_ADMIN_USER_GROUP	ACTIVE	This group contains the list of users who have been granted AUDIT MANAGER role to them	09-NOV-2020 15:17:08	node1	OKVADMIN	
<input type="checkbox"/>	SYSTEM_ADMIN_USER_GROUP	ACTIVE	This group contains the list of users who have been granted SYSTEM ADMINISTRATOR role to them	09-NOV-2020 15:16:42	node1	OKVADMIN	
<input type="checkbox"/>	KEY_ADMIN_USER_GROUP	ACTIVE	This group contains the list of users who have been granted KEY ADMINISTRATOR role to them	09-NOV-2020 15:16:17	node1	OKVADMIN	
<input type="checkbox"/>	ALL_ADMIN_USER_GROUP	ACTIVE	This group contains the list of users who have been granted KEY ADMINISTRATOR, SYSTEM ADMINISTRATOR and AUDIT MANAGER roles to them	09-NOV-2020 15:15:52	node1	OKVADMIN	

1 - 4 of 4

Group Members

Access to Wallets

3. Click **Create**.

The **Create User Group** page appears.

Create User Group

Name * Make Unique

Description

Cancel Save

4. In the **Name** field, enter the name of the new group.
See [Naming Guidelines for Objects](#).
5. If you are using a multi-master cluster, then choose whether to select the **Make Unique** check box.

Make Unique helps to control naming conflicts with names across the multi-master cluster environment. User groups that were created before an Oracle Key Vault conversion to a cluster node are not affected by naming conflicts.

- If you select **Make Unique**, then the group name will be active immediately and this user group can be used in user operations. Clicking **Make Unique** also displays a list of users that you can add to the group.
- If you do not select **Make Unique**, then the user group will be created in the `PENDING` state. Oracle Key Vault will then begin a name resolution operation and may rename the user group to a name that is unique across the cluster. If there is a naming collision, then the collision will be reported on the Conflicts page on any node in the cluster. The user group will then be renamed to a unique name. You will need to go to a read/write node of the cluster and either accept the renamed user group or change the user group name. If you change the user group name, then this will restart the

name resolution operation and the user group will return to a `PENDING` state. A user group in the `PENDING` state cannot be used to perform most operations.

6. In **Description**, optionally, enter a description for the user group.
7. Click **Save**.

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

Adding a User to a User Group

You can add an existing user to a user group if that user must manage the same security objects as the group.

If both the user and user group are in the `ACTIVE` state, then you can add users to a group when you create the group or later after creating the groups.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, and then **Manage Access** in the left navigation bar.
The User Groups page appears displaying a list of existing user groups.
3. Click the pencil icon in the **Edit** for the user group.
The User Group Details page appears.
4. Click **Add** in the **User Group Members** pane.
The Add User Group Members page appears displaying the list of existing users who are not in the user group.
5. Check the boxes for the users you want to add.
6. Click **Save**.

Granting a User Group Access to a Virtual Wallet

You can modify the access level to a virtual wallet for a user group as functional needs change.

However, you can only modify the access level if the user group and wallet are in the `ACTIVE` state.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, and then **Manage Access** in the left navigation bar.
The User Groups page appears displaying a list of existing user groups.
3. Click the pencil icon in the **Edit** column, for the user group that you want to modify.
The User Group Details page appears.
4. Click **Add** in the **Access to Wallets** section.
The **Add Access to User Group** page appears.
5. Select the wallet in **Select Wallet**.
6. Set the access level to the selected wallet in **Select Access Level**.

Select **Read Only**, **Read and Modify**, or **Manage Wallet**.

7. Click **Save**.

Related Topics

- [Access Control Configuration](#)
Oracle Key Vault enables you to control access to security objects at various access levels.

Renaming a User Group

Depending on its status, you can change the name of a user group.

In a multi-master cluster, if the user group is in the `PENDING` state, then only the creator user can rename the user group.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, and then **Manage Access** in the left navigation bar.
The User Groups page appears.
3. On the **User Groups** page, select the pencil icon in the **Edit** column for the user group that you want to modify.
The User Group Details page appears.
4. Enter a new name in the **Name** field.
See [Naming Guidelines for Objects](#). If this node is part of a multi-master cluster and you do not select **Make Unique**, then the user group will enter the `PENDING` state after being renamed.
5. Click **Save**.

Changing a User Group Description

A group description is useful for identifying the purpose of the group.

You can change this description at any time to match the purpose of the group. In a multi-master cluster, if the user group is in the `PENDING` state, then only the creator can modify the user group description.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, and then **Manage Access** in the left navigation bar.
The User Groups page appears.
3. On the **User Groups** page, select the pencil icon in the **Edit** column, for the user group that you want to modify.
The User Group Details page appears.
4. Enter a new description in the **Description** field.
5. Click **Save**.

Removing a User from a User Group

Depending on the requirement, you can remove a user from a user group.

In a multi-master cluster, if both the user and the user group are in the `ACTIVE` state, then you can remove users from a user group. You may want to remove these users when their function in the organization changes and they no longer need to manage the same security objects as the group.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab and then **Manage Access** in the left navigation bar.
The User Groups page appears displaying a list of existing user groups.
3. Click the pencil icon in the **Edit** for the user group.
The **User Group Details** page appears.
4. In the User Group Members area, select the users that you want to remove.
5. Click **Remove**.
6. Click **OK** to confirm.

Deleting a User Group

You can delete a user group when the users in the group do not need to access the same security objects.

Removing a user group automatically deletes the group's access to wallets and security objects. In a multi-master cluster, if a user group is in the `PENDING` state, then only the creator can delete it.

1. Log in to the Oracle Key Vault management console to Oracle Key Vault as a user who has been granted the Key Administrator role.
2. Select the **Users** tab and then **Manage Access** in the left navigation bar.
The User Groups page appears.
3. Select the users groups that you want to delete.
4. Click **Delete**.
5. Click **OK** to confirm.

Managing support and root Password

Using SSH on Oracle Key Vault server, you can change the support or `root` password.

- [Requirements for root and support User Passwords](#)
Ensure that you meet these requirements for `root` and `support` user passwords.
- [Changing the root User Password](#)
You can change the `root` user password using the provided information.
- [Changing the support User Account Password](#)
Before you perform the post-installation configuration task after the Oracle Key Vault installation, you can change the password for the `support` account in the server terminal console.

Requirements for root and support User Passwords

Ensure that you meet these requirements for `root` and `support` user passwords.

- The password must have at least 15 characters.
- The password must contain at least one uppercase letter, one lowercase letter, one digit, and one special character.
- The same character cannot repeat consecutively more than 3 times in the password.
- Characters from the same class cannot repeat consecutively more than 4 times in the password. For example, more than 4 lowercase letters in a row.
- The new password must have at least 8 characters that are different from the old password.

Changing the root User Password

You can change the `root` user password using the provided information.

To change the operating system user `root` password:

1. Enable SSH.

Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Select the **System** tab, then **Settings**. In the Network Details area, click **SSH Access**. Select **IP address(es)** and then enter only the IP addresses that you need, or select **All**. Click **Save**.

2. Log in to the Oracle Key Vault server through SSH as user `support`, then switch user to `root`.

```
ssh support@okv_server_IP_address
su - root
```

3. Use `passwd` command. Enter and then reenter the new password for the `root` user when prompted.

```
passwd
```

```
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Once the password is set successfully, the following message is displayed on the console:

```
All authentication tokens updated successfully.
```

Changing the support User Account Password

Before you perform the post-installation configuration task after the Oracle Key Vault installation, you can change the password for the `support` account in the server terminal console.

After setting the password for the `support` account during the post-installation task, you can use SSH to change the `support` password. (When you install Oracle Key Vault, you create this account as part of the process.) The `support` user will be prompted to change their password when the next time they log in is past the expiration time of their passwords. The expiration times are 365 days with a warning at 120 days, and with STIG enabled, it is 60 days with a warning at 60 days.

To change the `support` user password:

1. Enable SSH.

Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Select the **System** tab, then **Settings**. In the Network Details area, click **SSH Access**. Select **IP address(es)** and then enter only the IP addresses that you need, or select **All**. Click **Save**.

2. Log in to the Oracle Key Vault server through SSH as user `support`.

```
ssh support@okv_server_IP_address
```

3. Use `passwd` command. Enter and then re-enter the new password for the `support` user when prompted.

```
passwd
```

```
Changing password for user support.
```

```
New password:
```

```
Retype new password:
```

```
passwd: all authentication tokens updated successfully.
```

Once the password is set successfully, the following message is displayed on the console:

```
All authentication tokens updated successfully.
```

10

Managing Oracle Key Vault Virtual Wallets and Security Objects

You can create a virtual wallet to store security objects, and then share this wallet with trusted peers at different access levels.

- [Managing Virtual Wallets](#)
A virtual wallet is a container for security objects that you can create and then grant access to users.
- [Managing Access to Virtual Wallets from Keys & Wallets Tab](#)
You can grant virtual wallet access to and revoke virtual wallet access from endpoint by using the **Keys & Wallets** tab.
- [Managing Access to Virtual Wallets from User's Menu](#)
To manage access control on virtual wallets for users, endpoints, and their respective groups, you can use the **Users** menu or **Endpoints** menu.
- [Managing Security Objects](#)
You can manage the security objects in Oracle Key Vault using the Oracle Key Vault management console.
- [Managing the State of a Key or a Security Object](#)
You can set the date to activate or deactivate keys or security objects, and change the state of some virtual wallet security objects.
- [Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault](#)
You can restrict symmetric or private keys from leaving Oracle Key Vault.
- [Managing Details of Security Objects](#)
You can manage details about security objects, such as find details about these objects and modifying these details.

Managing Virtual Wallets

A virtual wallet is a container for security objects that you can create and then grant access to users.

- [About Virtual Wallets](#)
A virtual wallet is a container for security objects.
- [Creating a Virtual Wallet](#)
You can create a virtual wallet and add security objects to it at the same time.
- [Modifying a Virtual Wallet](#)
You can modify a virtual wallet and add security objects to it at the same time.
- [Adding Security Objects to a Virtual Wallet](#)
You can add new security objects to a virtual wallet at any time as needed.
- [Removing Security Objects from a Virtual Wallet](#)
You cannot remove security objects from virtual wallets at any time as needed.

- [Deleting a Virtual Wallet](#)
Deleting a virtual wallet removes the wallet as a container, but does not delete the security objects that were contained in it.

About Virtual Wallets

A virtual wallet is a container for security objects.

These security objects can be public and private encryption keys, including Transparent Data Encryption (TDE) keystores, Oracle wallets, Java keystores, certificates, secret data, and credential files. You can use a virtual wallet to group security objects for sharing with multiple users who need them to access encrypted data.

Any user can create a virtual wallet. After you create a virtual wallet, you can add keys and other security objects to the wallet. You can then grant other users, endpoints, user groups, and endpoint groups access to the virtual wallet at various levels of access. You can modify a virtual wallet and its wallet contents at any time. You can also modify virtual wallet user lists and their respective access level.

Other than the Key Administrator, access to the virtual wallet must be granted explicitly to users. Read, modify, and manage wallet permissions are required to add and remove objects from the wallet, and to grant or modify wallet access to other users and groups.

Creating a Virtual Wallet

You can create a virtual wallet and add security objects to it at the same time.

However, you can also create an empty virtual wallet, and add security objects to it later. You can modify access mappings on a virtual wallet at any time.

1. Log in to the Oracle Key Vault management console as a user with the Key Administrator role.
2. Select the **Keys & Wallets** tab, then **Wallets** in the left navigation bar.
3. In the Wallets page, click **Create**.

Create Wallet
Cancel Save

Name * Make Unique

Description

Wallet Type General SSH Server

Add Wallet Contents

Go Actions

	Display Name	Type	State	Creation Time	Creator	
<input type="checkbox"/>	TDE Master Encryption Key: MKID 06BDD8E1F43E8E2BB9AB6E2EE807F65996	Symmetric Key	Active	05-OCT-2023 16:36:15	OKVADMIN	^
<input type="checkbox"/>	TDE Master Encryption Key: MKID 068FC0667F137312E37B0CCB55A6360820	Symmetric Key	Active	05-OCT-2023 10:03:48	HR_DB_EP	
<input type="checkbox"/>	Default template for JANE	Template	No State	05-OCT-2023 09:56:36	JANE	
<input type="checkbox"/>	SSH Key for user: Sam, Fingerprint: SHA256:FULo4TaZ0uKG2vjRKM5cpM5UOCqyFvF94HSJB4TJMhl	Private Key	Active	05-OCT-2023 10:03:58	SSHADMIN	v

1 - 22 of 22

- Enter a name for the wallet in the **Name** field and an identifying description in **Description**. See [Naming Guidelines for Objects](#).

- Select the **Wallet Type** as **General** or **SSH Server**.

You need to provide SSH Server Host User name as well if you select **SSH Server**. SSH Server Host User name is the user on the SSH server host for whom this wallet is intended to authorize SSH access.

- If you are using a multi-master cluster, then choose whether to select the **Make Unique** check box.

Make Unique helps to control naming conflicts with virtual wallet names across the multi-master cluster environment. Virtual wallets that were created before an Oracle Key Vault conversion to a cluster node are not affected by naming conflicts.

- If you select **Make Unique**, then the virtual wallet will be active immediately and this wallet can be used in operations.
- If you do not select **Make Unique**, then the wallet will be created in the `PENDING` state. Oracle Key Vault will then begin a name resolution operation and may rename the wallet to a name that is unique across the cluster. If there is a naming collision, then the collision will be reported on the Conflicts page on any node in the cluster. The wallet will then be renamed to a unique name. You will need to go to a read/write node

of the cluster and either accept the renamed wallet name or change the wallet name. If you change the wallet name, then this will restart the name resolution operation and the wallet will return to a `PENDING` state. A wallet in the `PENDING` state cannot be used to perform most operations.

7. In the Add Wallet Contents pane, check the boxes by the names of the listed security objects that you want to add to the wallet.

The Add Wallet Contents pane lists the security objects you have Read and Modify access to. If the list is empty, then you have no access to the security objects already in Oracle Key Vault. In this case, you would add security objects to the wallet after you upload them to Oracle Key Vault.

You can modify the columns in the table in the Wallet Contents pane to show more information. From the **Actions** menu, select **Select Columns**. In the Select Columns dialog box, move the columns that you want to see to the Display in Report list, and then click **Apply**.

8. Click **Save** to create the new wallet with any associated security objects.

A **Wallet created successfully** message appears. The **Wallets** page appears and displays the new wallet in the list.

To see the contents in the wallet click the wallet name as the following figure shows.

The screenshot displays the Oracle Key Vault interface. At the top, there are 'Create' and 'Delete' buttons. Below is a search bar and an 'Actions' dropdown menu. The main table lists wallets with columns for 'Wallet Name', 'Name Status', 'Type', 'Description', 'Creation Time', and 'Edit'. The 'Wallet Contents' pane shows a table with columns for 'Display Name', 'Type', and 'Details'. The 'Access Settings' pane shows a table with columns for 'Subject Name' and 'Access'.

Wallet Name	Name Status	Type	Description	Creation Time	Edit
Cloud_App_Wallet	ACTIVE	GENERAL	This wallet contains the SSH private and public keys of the entities being managed on Oracle public cloud.	10-OCT-2023 16:51:34	
Peoplesoft_DB_Connect_Client_Wallet	ACTIVE	GENERAL	This wallet contains the password of DB users who login to Oracle DB.	10-OCT-2023 16:49:49	
Data_Recovery_Manager_Wallet	ACTIVE	GENERAL	This wallet contains the various client user credentials that can be downloaded as SEPS (Secure External Password Store) wallet.	10-OCT-2023 16:48:40	
root_ssh_wallet_for_austin	ACTIVE	SSH Server	SSH server wallet for host user 'root' on SSH server 'austin'	06-OCT-2023 18:25:51	

Display Name	Type	Details
Secret Data: Name Password_Name-Mon Oct 9 19:28:59 2023	Secret Data	0463E846-F9D8-4F5D-BFFB-9FECBD40373E
Default template for PEOPLESOFT_DB_CONNECT_CLIENT	Template	A86F60F1-9568-4E5D-8D3B-2EDD28661861

Subject Name	Access
PEOPLESOFT_DB_CONNECT_CLIENT	Read, Write, Manage Wallet
ALL_ADMIN_USER_GROUP	Read
KEY_ADMIN_USER_GROUP	Read

Related Topics

- [Name Conflict Resolution in a Multi-Master Cluster](#)
Naming conflicts can arise when an object has the same name as another object in a different node.

Modifying a Virtual Wallet

You can modify a virtual wallet and add security objects to it at the same time.

1. Log in to the Oracle Key Vault management console as a user with the Key Administrator role.
2. Select the **Keys & Wallets** tab, then **Wallets** in the left navigation bar.
3. In the Wallets page, click the **Edit** button for the wallet that you want to modify.
4. In the Wallet Overview pane, enter a new name for the wallet in the **Name** field and an identifying description in **Description**.

See [Naming Guidelines for Objects](#).

5. If you are using a multi-master cluster, then choose whether to select the **Make Unique** check box.

Make Unique helps to control naming conflicts with virtual wallet names across the multi-master cluster environment. Virtual wallets that were created before an Oracle Key Vault conversion to a cluster node are not affected by naming conflicts.

- If you select **Make Unique**, then the virtual wallet will be active immediately and this wallet can be used in operations.
 - If you do not select **Make Unique**, then the wallet will be created in the `PENDING` state. Oracle Key Vault will then begin a name resolution operation and may rename the wallet to a name that is unique across the cluster. If there is a naming collision, then the collision will be reported on the Conflicts page on any node in the cluster. The wallet will then be renamed to a unique name. You will need to go to a read/write node of the cluster and either accept the renamed wallet name or change the wallet name. If you change the wallet name, then this will restart the name resolution operation and the wallet will return to a `PENDING` state. A wallet in the `PENDING` state cannot be used to perform most operations.
6. To modify endpoint access settings, in the Wallet Access Settings pane, click **Add** to add new endpoints or click **Remove** to remove existing endpoints.
 7. In the Wallet Contents pane, check the boxes by the names of the listed security objects that you want to remove from the wallet.

The Wallet Contents pane lists the security objects you have added to the wallet. If the list is empty, then you have no access to the security objects already in Oracle Key Vault. In this case, you would add security objects to the wallet after you upload them to Oracle Key Vault.

You can modify the columns in the table in the Wallet Contents pane to show more information. From the **Actions** menu, select **Select Columns**. In the Select Columns dialog box, move the columns that you want to see to the Display in Report list, and then click **Apply**.

Select **Add Objects** or **Remove Objects**.

8. Click **Save** to create the new wallet with any associated security objects.
9. To view the status of the modified wallet, click **Wallets** in the left navigation bar.

The Wallets page appears and displays the modified wallet in the list, with the status of `PENDING`.

Related Topics

- [Name Conflict Resolution in a Multi-Master Cluster](#)
Naming conflicts can arise when an object has the same name as another object in a different node.

Adding Security Objects to a Virtual Wallet

You can add new security objects to a virtual wallet at any time as needed.

In a multi-master cluster, you cannot add **security objects** to a virtual wallet when it is in the **PENDING** state.

1. Log in to the Oracle Key Vault management console as a user who has the **Manage Wallet** access on the virtual wallet or as a user with the Key Administrator role.
2. Select the **Keys & Wallets** tab, then **Wallets** in the left navigation bar.
3. In the Wallets page, click the pencil icon in the **Edit** column corresponding to the wallet you want to work with.

The Wallet Overview page appears. The Wallet Contents pane lists the security objects already in the wallet.

4. In the Wallet Contents page, click **Add Objects** to display the Add Wallet Contents pane.

The Add Wallet Contents page lists the security objects you have **Read and Modify** access to. If the list is empty, then you have no access to the security objects already in Oracle Key Vault. In this case, you would add security objects to the wallet after you upload them to Oracle Key Vault.

You can modify the columns in the table in the Add Wallet Contents page to show more information. From the **Actions** menu, select **Select Columns**. In the Select Columns dialog box, move the columns that you want to see to the Display in Report list, and then click **Apply**.

5. Check the boxes by the security objects that you want to add to the wallet.
6. Click **Save**.

A confirmation message appears, then the **Wallet Overview** page appears. **Wallet Contents** lists the new security objects added.

Removing Security Objects from a Virtual Wallet

You cannot remove security objects from virtual wallets at any time as needed.

In a multi-master cluster, you can remove security objects from a virtual wallet when it is in the **PENDING** state.

1. Log in to the Oracle Key Vault management console as a user who has the **Manage Wallet** access on the virtual wallet or as a user with the Key Administrator role.
2. Select the **Keys & Wallets** tab, then **Wallets** in the left navigation bar.
3. In the Wallets pane, click the pencil icon in the **Edit** column corresponding to the wallet that you want to work with.

The Wallet Overview page appears. The Wallet Contents pane lists the security objects already in the wallet.

4. Check the boxes by the security objects you want to remove from the wallet.
5. Click **Remove Objects**.

The Wallet Contents pane in the Wallet Overview page displays the revised list.

6. Click **OK** to confirm.

Deleting a Virtual Wallet

Deleting a virtual wallet removes the wallet as a container, but does not delete the security objects that were contained in it.

These security objects will continue to remain in Oracle Key Vault. Endpoints that have downloaded this virtual wallet will continue to retain their local copy. In a multi-master cluster, you cannot delete a virtual wallet when it is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Manage Wallet access on the virtual wallet, or as a user with the Key Administrator role.
2. Select the **Keys & Wallets** tab, then **Wallets** in the left navigation bar.
3. In the Wallets page, check the boxes next to the name of the wallet that you want to delete.

You can delete more than one virtual wallet at the same time.

4. Click **Delete**.
5. Click **OK** to confirm.

Managing Access to Virtual Wallets from Keys & Wallets Tab

You can grant virtual wallet access to and revoke virtual wallet access from endpoint by using the **Keys & Wallets** tab.

- [About Managing Access to Virtual Wallets from the Keys & Wallets Tab](#)
Access control is deciding which users and endpoints share virtual wallets and security objects, and what operations they can perform on those virtual wallets.
- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.
- [Modifying Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can modify access settings on a virtual wallet for users, user groups, endpoints, and endpoint groups from the **Keys & Wallets** tab.

About Managing Access to Virtual Wallets from the Keys & Wallets Tab

Access control is deciding which users and endpoints share virtual wallets and security objects, and what operations they can perform on those virtual wallets.

You must have Manage Wallet access to a virtual wallet or be a Key Administrator to manage access control for users, endpoints, and their respective groups.

To manage access to virtual wallets, you can use the **Keys & Wallets** tab, where you select the wallet, you grant an endpoint, endpoint group, user, or user group access to the wallet.

Related Topics

- [Managing Access to Virtual Wallets from User's Menu](#)
To manage access control on virtual wallets for users, endpoints, and their respective groups, you can use the **Users** menu or **Endpoints** menu.

Granting Access to Users, User Groups, Endpoints, and Endpoint Groups

You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.

After they have access to the wallet, they will have access to all the [security objects](#) in the wallet. In a multi-master cluster, you cannot grant access to endpoints, endpoint groups, users, or user groups while the virtual wallet is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the **Manage Wallet** access on the virtual wallet, or as a user with the Key Administrator role.
2. Select the **Keys & Wallets** tab, then **Wallets** in the left navigation bar.
3. In the Wallets pane, click the pencil icon in the **Edit** column corresponding to the wallet to which you want to grant access.

The Wallet Overview page appears.

4. In the Wallet Access Settings pane, click **Add**.

The Add Access to Wallet page appears.

5. In the Add Access to Wallet page, under Select Endpoint/User Group, from the **Type** menu, select the entity type you want to grant access.

Possible values for **Type** are **Endpoint Groups**, **Endpoints**, **User Groups**, and **Users**. The type you select determines the list that is displayed. For example, if you select **Endpoint Groups** as the **Type**, the list of Oracle Key Vault endpoint groups is displayed under the heading **Endpoint Groups**. If you select **Users**, the list of Oracle Key Vault users are displayed under the heading **Users**.

6. Select the check box in the **Name** table corresponding to the entity you want to grant access.
7. In the Select Access Level pane, select one of the following access levels: in the **Select Access Level** pane.
 - **Read Only** or **Read and Modify**

- **Manage Wallet**
8. Click **Save**.
The Wallet Access Settings pane displays the new entity.

Modifying Access to Users, User Groups, Endpoints, and Endpoint Groups

You can modify access settings on a virtual wallet for users, user groups, endpoints, and endpoint groups from the **Keys & Wallets** tab.

In a multi-master cluster, you cannot modify access to endpoints, endpoint groups, users, or user groups while the virtual wallet is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Manage Wallet permission on the virtual wallet or as a user with the Key Administrator role.
2. Select the **Keys & Wallets** tab, then **Wallets** in the left navigation bar.
3. In the Wallets pane, click the pencil icon in the **Edit** column corresponding to the wallet name.

The Wallet Overview page appears, with Wallet Access Settings listing the entities that have access to the wallet and their access levels.

4. In Wallet Access Settings, click the pencil icon corresponding to the entity under **Subject Name**.

A Modify Access window appears. Wallet Access Settings lists all the entities that have access to this wallet under Subject Name, and can include users, endpoints, user groups, and endpoint groups.

5. Select the access settings that you want to modify, then click **Save**.

A message appears: **Successfully updated**. The Wallet Overview page appears and Wallet Access Settings displays the new access mapping for the entity.

6. Click **Save** in the Wallet Overview page.

Managing Access to Virtual Wallets from User's Menu

To manage access control on virtual wallets for users, endpoints, and their respective groups, you can use the **Users** menu or **Endpoints** menu.

- [Granting a User Access to a Virtual Wallet](#)
You can grant access to a virtual wallet by using the **Users** tab.
- [Revoking User Access from a Virtual Wallet](#)
You can revoke access to a virtual wallet for a user by using the **Users** tab.
- [Granting a User Group Access to a Virtual Wallet](#)
You can grant user group access to a virtual wallet by using the **Users** tab.
- [Revoking User Group Access from a Virtual Wallet](#)
You can remove user group access to a virtual wallet by using the **Users** tab.

Related Topics

- [Managing Endpoint Access to a Virtual Wallet](#)
You can grant an endpoint access to a virtual wallet, and revoke or modify access when it is no longer necessary.

- [Managing Access to Virtual Wallets from Keys & Wallets Tab](#)
You can grant virtual wallet access to and revoke virtual wallet access from endpoint by using the **Keys & Wallets** tab.

Granting a User Access to a Virtual Wallet

You can grant access to a virtual wallet by using the **Users** tab.

In a multi-master cluster, you cannot grant a user access to a virtual wallet while the virtual wallet is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Manage Wallet permission on the virtual wallet, or as a user with the Key Administrator role.
2. Select the **Users** tab, then **Manage Users** in the left navigation bar.
3. In the Manage Users pane, click the user's name in the **User Name** column.
4. In the Access to Wallets pane, click **Add**.
The Add Access to User page appears.
5. Select a virtual wallet from the available list.
6. In the Select Access Level pane select the desired access levels.
7. Click **Save**.

A message appears: **Access mapping successfully added**. You can check **Access to Wallets** in User Details for the user to see the wallet added.

Related Topics

- [Access Control Options](#)

Revoking User Access from a Virtual Wallet

You can revoke access to a virtual wallet for a user by using the **Users** tab.

In a multi-master cluster, you cannot revoke user access from a virtual wallet while the virtual wallet is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Manage Wallet access on the virtual wallet, or as a user with the Key Administrator role.
2. Select the **Users** tab, then **Manage Users** in the left navigation bar.
3. In the Manage Users pane, click the user's name under the **User Name** column.
4. In the Access to Wallets pane, check the box by the virtual wallet that you want to revoke access to.
5. Click **Remove**.
6. In the confirmation window, click **OK**.

A message appears: **Access Mapping(s) deleted successfully**. You can check **Access to Wallets** in User Details for the user to see the wallet deleted.

Granting a User Group Access to a Virtual Wallet

You can grant user group access to a virtual wallet by using the **Users** tab.

When you grant a user group access to a virtual wallet all members of the group will have access to the security objects within the wallet. In a multi-master cluster, you cannot grant a user group access to a virtual wallet while the virtual wallet is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, then **Manage Access** in the left navigation bar.
3. Click the pencil icon in the **Edit** column corresponding to the user group.
4. In the Access to Wallets pane, click **Add**.

The Add Access to User Group page appears.

5. In the Select Wallet pane, select the check boxes for one or more wallets.
6. In the Select Access Level pane, select the desired access levels.
7. Click **Save**.

A message appears: **Access mapping successfully added**. You can check **Access to Wallets** in User Groups for the user to see the wallet added.

Revoking User Group Access from a Virtual Wallet

You can remove user group access to a virtual wallet by using the **Users** tab.

In a multi-master cluster environment, you cannot revoke user group access from a virtual wallet while the virtual wallet is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Users** tab, and then select **Manage Access** in the left sidebar.
The **User Groups** page appears.
3. Click the pencil icon in the **Edit** column corresponding to the user group.
The **User Group Details** page appears.
4. In the **Access to Wallets** pane, check the box by the virtual wallet you want to revoke access to.
5. Click **Remove**.
6. Click **OK** to confirm.

A message appears: **Access Mapping(s) deleted successfully**. You can check **Access to Wallets** in **User Groups** to see the wallet removed from the list.

Managing Security Objects

You can manage the security objects in Oracle Key Vault using the Oracle Key Vault management console.

- [Creating Keys](#)
You can create a regular or application specific keys and key pairs.

Creating Keys

You can create a regular or application specific keys and key pairs.

- [About Creating Keys](#)
As an Oracle Key Vault user, you can create keys for Oracle TDE and Oracle GoldenGate, and key pairs for SSH key management.
- [Application Keys](#)
You can create feature specific keys called application keys from the Oracle Key Vault management console. You can create keys for TDE, keys for Oracle GoldenGate and key pairs for SSH key management.
- [Creating Symmetric Keys](#)
You can create symmetric keys from the Oracle Key Vault management console. The key material can be system generated in Oracle Key Vault or can be uploaded from a file. Symmetric keys can be used for custom applications using Java, C SDK, or RESTful API.
- [Create Public-Private Key Pair](#)
You can create public-private key pairs from the Oracle Key Vault management console. The public-private key pairs can be used for sign and verify operations besides encryption and decryption by custom applications using Java, C SDK. or RESTful API
- [Create TDE Master Encryption Key](#)
You can create a TDE master encryption key from the Oracle Key Vault management console. The key material can be system generated in Oracle Key Vault or can be uploaded from a file. The key has to be put into use on the database for which it was created.
- [Create GoldenGate Master Key](#)
You can create a GoldeGate master encryption key from the management console. The key material can be system generated in Oracle Key Vault or can be uploaded from a file. The user must configure and create the key for the GoldenGate deployment.
- [Creating SSH Key Pair](#)
You can create an Secure Shell (SSH) key pair from the Oracle Key Vault management console. The keys can be granted access to SSH endpoint to setup connections to SSH deployment or they can be used to rotate the SSH keys of endpoints of an existing deployment.

About Creating Keys

As an Oracle Key Vault user, you can create keys for Oracle TDE and Oracle GoldenGate, and key pairs for SSH key management.

You can define the life time of the key using the activation and de-activation dates and control whether a key is extractable or not.

Based on how your alerts and emails are configured, you will be notified when the keys are expiring. You must also specify the usage for the key. If necessary, you can add the name attribute. Name attribute would be unique in the system so should be used only if the object needs to have a unique human readable name in the Oracle Key Vault cluster.

You can also add custom attributes to the key. You can use the custome attributes to attach the tags to the keys. For example, if you are creating the keys for a specific department, you can add the department name as the custom attribute for those keys.

For endpoints to exercise these keys, you need to add them to the wallets where the endpoints can access them.

To enable endpoints to use the key, you can add the keys to the wallet where the endpoint has at least the read access.

Related Topics

- [How a Multi-Master Cluster Affects User Accounts](#)
An Oracle Key Vault multi-master cluster environment affects users in various ways.

Application Keys

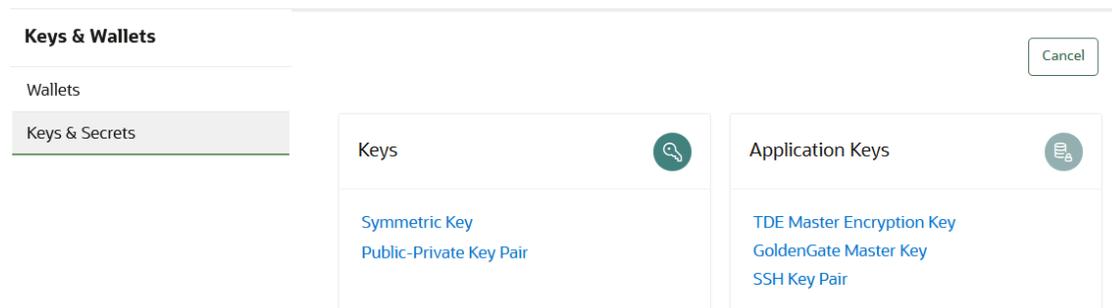
You can create feature specific keys called application keys from the Oracle Key Vault management console. You can create keys for TDE, keys for Oracle GoldenGate and key pairs for SSH key management.

Applications require keys have certain basic or customer attributes set with pre-defined names and formats, like TDE keys should have the name attribute with TDE master key identifier in hex format or the cryptographic algorithms for Oracle GoldenGate keys are set to AES and cryptographic length to 256.

Application keys are preset to work with the specific features. The Oracle Key Vault management console supports creation of these application keys:

- TDE master encryption key
- GoldenGate master key
- SSH key pair

In each case, once the key is created the corresponding application needs to be setup to make use of the keys in Oracle Key Vault. In case of TDE master encryption key, the database needs to use or activate the key. And for that the database must be setup with Oracle Key Vault and have read and write access on the created key. Similarly, the key management service (KMS) global parameters need to be setup appropriately besides the Oracle Key Vault endpoints to consume the Oracle GoldenGate keys from Oracle Key Vault.



Creating Symmetric Keys

You can create symmetric keys from the Oracle Key Vault management console. The key material can be system generated in Oracle Key Vault or can be uploaded from a file. Symmetric keys can be used for custom applications using Java, C SDK, or RESTful API.

You can create AES and 3DES keys and the keys can be marked extractable or non-extractable. You can either bring your own key material or let the system generate the key material.

To enable the use of the keys by endpoints, you add them to the wallets that the endpoints can access.

Creating Symmetric Keys

1. Log in to the Oracle Key Vault management console.
2. Select **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the **Keys & Secrets** page, click **Create**.
4. Under the Keys area of the page that appears, click **Symmetric Keys**. The **Create Symmetric Key** page appears.

The screenshot shows the 'Create Symmetric Key' form with the following fields and values:

- Cryptographic Algorithm:** AES
- Cryptographic Length:** 256
- Key Material:** System-Generated (selected), Bring You Own Key
- Extractable:** False
- Date of Activation:** 19-Sep-2023 17:55:00
- Date of Deactivation:** (empty)
- Wallet Membership:** Crypto_app_wallet
- Select Wallet:** (button)
- Advanced:** (collapsed section)

Create symmetric key

5. From the **Cryptographic Algorithm** drop-down list, select the algorithm **AES** or **3DES**.
6. Select the Cryptographic Length.
7. Choose **System-Generated**, if you want the key material to be system generated or **Bring Your Own Key** if you are supplying the key material for the key. If you choose the **Bring Your Own Key** option, choose a file that includes the key material in hex.
8. Select the **Extractable** setting from the drop-down list:
 - a. Selecting **FALSE** prevents the key from leaving the Oracle Key Vault cluster boundary.
 - b. Selecting **TRUE** allows the key to leave Oracle Key Vault cluster boundary. Default value is **FALSE**.
9. Enter the **Date of Activation**.
10. Enter the **Date of Deactivation**.
11. Enter an existing wallet name for the **Wallet Membership**. The newly created key gets added to this wallet. You can also click **Select Wallet** and then select the wallet from the pop-up.
12. Click **Create** to create the key.
You can set advanced attributes for the symmetric keys. You can set the human readable **Name** attribute which is unique across the cluster. You can also set three custom attributes of type text or number. You can edit the key usage as well.

Advanced Attributes for Symmetric Keys

Before you click Create to create the key, you can set the advanced attributes of the key.

1. Expand the **Advanced** section.

The screenshot shows the 'Create Symmetric Key' form. The 'Advanced' section is expanded, revealing the 'Key Usage' section with various checkboxes. The 'Name' section has 'Name Value' set to 'Crypto_App_Key_10' and 'Name Type' set to 'Text String'. The 'Custom Attribute 1' section has 'Name' set to 'x-Department', 'Value' set to 'HR', and 'Type' set to 'Text'.

Advanced attributes for symmetric key

2. Enter the following information in the **Advanced** section,
 - **Key Usage:** Select operations for the key usage.
 - **Name:** Add the **Name Value** to identify the key. Add the **Name Type**.
 - **Custom Attribute 1:** Add **Name**, **Value**, and **Type** for the custom attribute. Name should begin with x- and cannot begin with x-OGG and x-OKV.
 - **Custom Attribute 2:** Set the custom attribute like **Custom Attribute 1**.
 - **Custom Attribute 3:** Set the custom attribute like **Custom Attribute 1**.
3. After adding the details, click **Create** to create the key with advanced attributes.

Create Public-Private Key Pair

You can create public-private key pairs from the Oracle Key Vault management console. The public-private key pairs can be used for sign and verify operations besides encryption and decryption by custom applications using Java, C SDK, or RESTful API

You can create RSA key pairs of length 2048, 3072 and 4096 bits and the private keys can be marked extractable or non-extractable.

You can either bring your own key material or let the system generate the key material.

Creating Public-Private Key Pair

1. Log in to the Oracle Key Vault management console.
2. Select **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the **Keys & Secrets** page, click **Create**.
4. Under the Keys area of the page that appears, click **Public-Private Key Pair**. The **Public-Private Key Pair** page appears.

The screenshot shows the 'Create Public-Private Key Pair' form. At the top right, there are 'Cancel' and 'Create' buttons. The form contains the following fields:

- Cryptographic Algorithm ***: A dropdown menu with 'RSA' selected.
- Cryptographic Length ***: A dropdown menu with '2048' selected.
- Private Key Extractable ***: A dropdown menu with 'False' selected.
- Date of Activation**: A text input field with a calendar icon.
- Date of Deactivation**: A text input field with a calendar icon.
- Wallet Membership**: A text input field with a 'Select Wallet' button next to it.

At the bottom left, there is an 'Advanced' toggle button.

5. From the **Cryptographic Algorithm** drop-down list, select the algorithm **RSA**.
6. Select the **Cryptographic Length**.
7. Select the **Private Key Extractable** setting from the drop-down list:
 - a. Selecting **FALSE** prevents the key from leaving the Oracle Key Vault cluster boundary.
 - b. Selecting **TRUE** allows the key to leave Oracle Key Vault cluster boundary.Default value is **FALSE**.
8. Enter the **Date of Activation**.
9. Enter the **Date of Deactivation**.
10. Enter an existing wallet name for the **Wallet Membership**. The newly created keys gets added to this wallet. You can also click **Select Wallet** and then select the wallet from the pop-up.

11. Click **Create** to create the key pair.

You can set advanced attributes that are common to both public and private keys and attributes specific to either the public or private key. You can set the human readable **Name** attribute which is unique across the cluster for the private key and for the public key. You can set the key usage for public and private key as well. You can also set up to two custom attributes of type text or number that is for public, private keys and for both (common attributes).

Advanced Creating Public-Private Key Pair

Before you click Create to create the key pair, you can set the advanced attributes of the public and private keys.

1. Expand the **Advanced** section.

▼ Advanced

Common Attributes

Custom Attribute 1

Name

Value Type

Custom Attribute 2

Name

Value Type

Private Key Attributes

Key Usage *

<input checked="" type="checkbox"/> Decrypt	<input type="checkbox"/> Derive Key
<input type="checkbox"/> Encrypt	<input type="checkbox"/> Export
<input type="checkbox"/> Generate Cryptogram	<input checked="" type="checkbox"/> Sign
<input type="checkbox"/> Translate Decrypt	<input type="checkbox"/> Translate Encrypt
<input type="checkbox"/> Translate Unwrap	<input type="checkbox"/> Translate Wrap
<input checked="" type="checkbox"/> Unwrap Key	<input type="checkbox"/> Validate Cryptogram
<input type="checkbox"/> Verify	<input type="checkbox"/> Wrap Key

Name

Name Value Name Type

Custom Attribute 1

Name

Value Type

Custom Attribute 2

Name

Value Type

Public Key Attributes

2. Enter the following information for the **Common Attributes** in the Advanced section.
 - **Custom Attribute 1:** Add **Name**, **Value** and **Type** for the custom attribute of the common attributes. Name should begin with x- and cannot begin with x-OGG and x-OKV.
 - **Custom Attribute 2:** Set the custom attribute like **Custom Attribute 1**.
3. Enter the following information for the **Private Key Attributes** in the Advanced section,
 - **Key Usage:** Select operations for the key usage.
 - **Name:** Add the **Name Value** to identify the private key. Add the **Name Type**.
 - **Custom Attribute 1:** Add **Name**, **Value** , and **Type** for the custom attribute of the private key. Name should begin with x- and cannot begin with x-OGG and x-OKV.
 - **Custom Attribute 2:** Set the custom attribute like **Custom Attribute 1**.
4. Enter the following information for the **Public Key Attributes** in the **Advanced** section,
 - **Key Usage:** Select operations for the key usage.
 - **Name:** Add the **Name Value** to identify the public key. Add the **Name Type**.
 - **Custom Attribute 1:** Add Name, Value and Type for the custom attribute of the public key. Name should begin with x- and cannot begin with x-OGG and x-OKV.
 - **Custom Attribute 2:** Set the custom attribute like **Custom Attribute 1**.
5. Click **Create** to create the key pair with advanced attributes.

Create TDE Master Encryption Key

You can create a TDE master encryption key from the Oracle Key Vault management console. The key material can be system generated in Oracle Key Vault or can be uploaded from a file. The key has to be put into use on the database for which it was created.

You need to supply the master key identifier when creating the TDE master encryption key. The master key identifier is a 32 byte random string that should be unique in the database ecosystem. Either you can use the one that Oracle Key Vault has generated for you or can supply your own.

Activation and deactivation dates are preset to activate the key as soon as it is created and expire the key in 2 years. You can choose to activate the key at a later date and also clear the de-activation dates so the TDE master encryption keys never expires. You can bring in your own key material or have the key material system generated by Oracle Key Vault. You can make the TDE master encryption key not extractable.



Note:

Setting the TDE master encryption key to non-extractable may cause scale and performance issues.

You should add the TDE master encryption key to the wallet of the database endpoint where it will be activated.

1. Log in to the Oracle Key Vault management console.
2. Select **Keys & Wallets**, then **Keys & Secrets** in the left navigation bar.

3. Click **Create**.
4. In the **Create Keys** page, click **Public-Private Key Pair**. The **The Create TDE Master Encryption Key** page appears.
5. Under the *Application Keys* area of the page that appears, click **TDE Master Encryption Key**.

Create TDE master encryption key

6. Enter the **Master Encryption Key Identifier** or choose the one that is system generated.
7. Choose System-Generated if you want the key material to be system generated or **Bring Your Own Key** if you are supplying the key material for the key. If you choose the **Bring Your Own Key** option, choose a file that includes the key material in hex.
8. Select the **Extractable** setting from the drop-down list:
 - a. Selecting **FALSE** prevents the key from leaving the Oracle Key Vault cluster boundary.
 - b. Selecting **TRUE** allows the key to leave Oracle Key Vault cluster boundary.
Default value is **FALSE**.
9. Enter the **Date of Activation**. Activation date is auto-populated to current date and time. You can edit the activation date.
10. Enter the **Date of Deactivation**. Deactivation date is set to 2 years from current date.
11. Enter an existing wallet name for the **Wallet Membership**. The newly created key gets added to this wallet. You can also click **Select Wallet** and then select the wallet from the pop-up.
12. Click **Create** to create the TDE master encryption key.

Create GoldenGate Master Key

You can create a GoldenGate master encryption key from the management console. The key material can be system generated in Oracle Key Vault or can be uploaded from a file. The user must configure and create the key for the GoldenGate deployment.

For the Oracle GoldenGate Master Key you will need to supply the master key name and the master key version. The user must ensure that the master key name is unique within the cluster. And the key version supplied is numeric and larger than a previous value for the given master key name. See [Configuring an Encryption Profile](#)

As with the TDE master encryption keys, the activation and deactivation dates are preset to activate the key as soon as it is created and expire the key in 2 years. You can choose to activate the key at a later date and also clear the de-activation dates so the Oracle GoldenGate master keys never expires. As before you can bring in your own key material or have the key material system generated by Oracle Key Vault. You can make the GoldenGate master encryption key not extractable if you are using GoldenGate deployment of version or higher.

Rotating the GoldenGate master key is as easy setting a new version larger than any previous version for the given master key name.

You should add the GoldenGate master key to the wallet of all the endpoints of a given GoldenGate deployment.

1. Log in to the Oracle Key Vault management console.
2. Select **Keys & Wallets**, then **Keys & Secrets** in the left navigation bar.
3. Click **Create**.
4. In the **Create Keys** page, click **GoldenGate Master Key**. The **Create GoldenGate Master Key** page appears.
5. Under the *Application Keys* area of the page that appears, click **GoldenGate Master Key**.

Create GoldenGate Master Key

Cancel Create

Create a new Oracle GoldenGate master key with a master key name and version and add it to the wallet of the Oracle GoldenGate endpoint.
Ensure to configure the Key Management Service (KMS) global parameters in Oracle GoldenGate for Oracle Key Vault.

Master Key Name * OGG_Masterkey ⓘ

Master Key Version * 1 ⓘ

Cryptographic Algorithm AES

Cryptographic Length 256

Key Material * System-Generated Bring Your Own Key

OGG_MasterKey_Material_AES256.txt ⓘ
Select file with hex encoded key material

Extractable * False ▾

Date of Activation 19-SEP-2023 15:07:20 ⓘ

Date of Deactivation 19-SEP-2025 15:07:20 ⓘ

Wallet Membership OGG_WALLET ⓘ

Create GoldenGate master key

6. Enter the **Master Key Name**.
7. Enter the **Master Key Version**.
8. Choose **System-Generated**, if you want the key material to be system generated or **Bring Your Own Key** if you are supplying the key material for the key. If you choose the **Bring Your Own Key** option, choose a file that includes the key material in hex.
9. Select the **Extractable** setting from the drop-down list:
 - a. Selecting **FALSE** prevents the key from leaving the Oracle Key Vault cluster boundary.
 - b. Selecting **TRUE** allows the key to leave Oracle Key Vault cluster boundary.
Default value is **FALSE**.
10. Enter the **Date of Activation**. Activation date is auto-populated to current date and time. You can edit it or clear it.
11. Enter the Date of Deactivation. Deactivation date is set to 2 years from now.
12. Enter an existing wallet name for the **Wallet Membership**. The newly created key will be added to this wallet. You can also click **Select Wallet** and then select the wallet from the pop-up.
13. Click **Create** to create the GoldenGate master key.

Related Topics

- [Managing Encryption Using a Key Management Service in Oracle GoldenGate](#)
- [Managing Encryption Using a Key Management Service in Oracle GoldenGate Classic Architecture](#)

Creating SSH Key Pair

You can create an Secure Shell (SSH) key pair from the Oracle Key Vault management console. The keys can be granted access to SSH endpoint to setup connections to SSH deployment or they can be used to rotate the SSH keys of endpoints of an existing deployment.

You can create SSH key pairs that are RSA key pairs of length 2048, 3072 and 4096 bits and the private keys can be marked extractable or not extractable. You should mark the private key SSH private key as not extractable. You must supply the name of the SSH client for which these keys are created in the SSH user field and at the time of creation the SSH key pairs should be added to the SSH users general wallet. The public key may be added to the 'SSH Server' wallet of the hosts where you want to give the client access.

As with the TDE master encryption keys, the activation and deactivation dates are preset to activate the key as soon as it is created and expire the key in 2 years. You can choose to activate the key at a later date and also clear the de-activation dates so the SSH keys do not expire. This is not recommended.

1. Log in to the Oracle Key Vault management console.
2. Select **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. Click **Create**.
4. In the **Create Keys** page, click **SSH Key Pair**. The **Create SSH Key Pair** page appears.
5. Under the Application Keys area of the page that appears, click **SSH Key Pair**.

Create SSH Key Pair

Cancel Create

Create a new SSH key pair for an SSH user and add it to the wallet of the SSH user endpoint.
You can authorize the SSH user to access an SSH server by adding the SSH user's public key to the SSH server wallet.

SSH User * Sam_sysadmin ⓘ

Cryptographic Algorithm * RSA ▾

Cryptographic Length * 2048 ▾

Private Key Extractable * False ▾

Date of Activation 19-SEP-2023 14:57:39 ⓘ

Date of Deactivation 19-SEP-2025 14:57:39 ⓘ

Wallet Membership Sam_ssh_keypair_wallet ⓘ Select Wallet

Create SSH key pairs

6. Enter the **SSH User**. The **SSH User** is intended to track the actual consumer of the SSH keys, a human, an application, or a machine.
7. Enter the name or the identity of the **SSH User**.
8. Enter the **Cryptographic Algorithm** drop-down list, select the algorithm **RSA**.
9. Select the **Cryptographic Length**.
10. Select the **Private Key Extractable** setting from the drop-down list:
 - a. Selecting **FALSE** prevents the private key from leaving the Oracle Key Vault cluster boundary.
 - b. Selecting **TRUE** allows the private key to leave Oracle Key Vault cluster boundary.
Default value is **FALSE**.
11. Enter the **Date of Activation**. Activation date is auto-populated to current date and time. You can edit it or clear it.
12. Enter the Date of Deactivation. Deactivation date is set to 2 years from now.
13. Enter an existing wallet name for the **Wallet Membership**. The newly created keys will be added to this wallet. You can also click **Select Wallet** and then select the wallet from the pop-up.
14. Click **Create** to create the SSH key pair.

Related Topics

- [SSH Keys Management Concepts](#)
Secure Shell (SSH) is the protocol used for remote administration and operations of hosts.
- [Management of SSH Keys - Setup and Configuration](#)
You can use Oracle Key Vault to centrally manage Secure Shell (SSH) private and public keys and control access to SSH servers.

Managing the State of a Key or a Security Object

You can set the date to activate or deactivate keys or security objects, and change the state of some virtual wallet security objects.

- [About Managing the State of a Key or a Security Object](#)
You can control the dates when a key or a security object is active, that is, when it can be used.
- [How a Multi-Master Cluster Affects Keys and Security Objects](#)
Keys that you create on one node of a multi-master cluster will take some time to appear on other nodes in the cluster.
- [Activating a Key or Security Object](#)
Keys can be in the **Active** or **Pre-Active** state.
- [Deactivating a Key or Security Object](#)
A key deactivates or expires when it passes the date that has been set for deactivation.
- [Revoking a Key or Security Object](#)
When you revoke a key, you can set its state to **Deactivated** or **Compromised**.
- [Destroying a Key or Security Object](#)
When a key is no longer used or compromised in some way, then you can destroy it.

About Managing the State of a Key or a Security Object

You can control the dates when a key or a security object is active, that is, when it can be used.

You also can revoke and destroy keys and security objects. Be aware that a multi-master cluster affects the activation or deactivation times of keys and security objects on different nodes, and that naming conflicts can arise.

Related Topics

- [How a Multi-Master Cluster Affects Keys and Security Objects](#)
Keys that you create on one node of a multi-master cluster will take some time to appear on other nodes in the cluster.

How a Multi-Master Cluster Affects Keys and Security Objects

Keys that you create on one node of a multi-master cluster will take some time to appear on other nodes in the cluster.

The time is defined by the replication lag between nodes. The replication lag value is displayed on the Cluster Link State pane of the Monitoring page, which can be accessed by choosing the **Cluster** tab.

If you add a Transparent Data Encryption (TDE) master encryption key to two different keystores on two different nodes, then it will be shown in both keystores.

Adjusting the activation date, deactivation date, process start date, and protect stop date has restrictions. For these dates, if changes are made to the [security object](#) very close to the current time, then state changes can happen because of replication lag.

As with the creation of any object in a multi-master cluster, a security object can have a name conflict with an object created on a different node. If there is a conflict, then Oracle Key Vault will suggest a unique name or allow you to rename it.

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

Activating a Key or Security Object

Keys can be in the **Active** or **Pre-Active** state.

Keys are in the **Pre-Active** state when they are created. However, for a key that will be used for securing data at a date later than its creation date, you can set the **Process Start Date**. Currently, keys uploaded with a third-party KMIP clients, RESTful service utility, C and Java SDKs are in a **Pre-Active** state and do not have the **Date of Activation** field set. For all other keys, the **Date of Activation** is system generated and cannot be set.

1. Log in to the Oracle Key Vault management console as a user who has read and modify access on this key.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the Keys & Secrets page, click the edit pencil icon under **Edit** corresponding to the item for which you want to set.
4. On the Object Details page for the item, click **Activate**.
5. Click **OK** to confirm.

Note:

- You can set the activation date at the time of creating the security object from Oracle Key Vault 21.3 onwards.
- You can set the date of activation of a security object after its creation by setting the activation date attribute of the security object using third-party KMIP clients, RESTful services utility, C and Java SDKs.

Related Topics

- Enhancements for RESTful Services Utility Commands Used for registration
- okv managed-object attribute add Command
- [okvAttrAddActivationDate](#)

Deactivating a Key or Security Object

A key deactivates or expires when it passes the date that has been set for deactivation.

1. Log in to the Oracle Key Vault management console as a user who has read and modify access on this key.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the Keys & Secrets page, click the edit pencil icon under **Edit** corresponding to the item to be deactivated.
4. In the Object Details page for the item, set the **Date of Deactivation** to the date by which you want the key to be deactivated.
5. Click **Save**.

 **Note:**

- You can set the deactivation date at the time of creating the security object from Oracle Key Vault 21.3 onwards.
- You can set the date of deactivation of a security object after its creation by setting the deactivation date attribute of the security object using third-party KMIP clients, RESTful services utility, C and Java SDKs.

Related Topics

- Enhancements for RESTful Services Utility Commands Used for registration
- okv managed-object attribute add Command
- [okvAttrAddDeactivationDate](#)

Revoking a Key or Security Object

When you revoke a key, you can set its state to **Deactivated** or **Compromised**.

At this point, the key should no longer be used to encrypt new data. However, you can download and use the deactivated keys to decrypt old data.

1. Log in to the Oracle Key Vault management console as a user who has read and modify access on this key.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the Keys & Secrets page, click the edit pencil icon under **Edit** corresponding to the item that you want to revoke.
4. In the Object Details page, click **Revoke**.
5. In the Revoke Object page, from the **Revocation Reason** drop-down list, select a reason for the revocation.
6. Optionally, add more details in **Revocation Message**
7. Click **Save**.

Destroying a Key or Security Object

When a key is no longer used or compromised in some way, then you can destroy it.

Metadata for destroyed keys and [security objects](#) are kept in Oracle Key Vault even after they have been destroyed.

1. Log in to the Oracle Key Vault management console as a user who has read and modify access on this key.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the Keys & Secrets page, click the edit pencil icon under **Edit** corresponding to the item that you want to destroy.
4. On the Object Details page, click **Destroy**.
5. In the confirmation window, click **OK**.

Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault

You can restrict symmetric or private keys from leaving Oracle Key Vault.

- [About Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault](#)
The ability to restrict symmetric or private keys (extraction) from leaving Oracle Key Vault ensures a higher level of security for these objects.
- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

About Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault

The ability to restrict symmetric or private keys (extraction) from leaving Oracle Key Vault ensures a higher level of security for these objects.

Many operations that use symmetric and private keys perform these operations outside of Oracle Key Vault and by default, symmetric and private keys within Oracle Key Vault can be extracted for this purpose. Consider the example with Transparent Database Encryption (TDE) master encryption keys that are stored in Oracle Key Vault. When an Oracle Database endpoint needs to decrypt the data encryption key, the PKCS#11 library fetches the TDE master encryption key from Oracle Key Vault to perform the decryption. If your site requires that symmetric or private keys to never leave Oracle Key Vault, then you can configure the symmetric and private keys to remain within Oracle Key Vault by setting their extractable attribute value to false. Setting the extractable attribute value to false prevents the key material of the symmetric and private key from being extracted from Oracle Key Vault, but still allows other object metadata (including object attributes, state, and so on) to be retrieved from Oracle Key Vault. If the TDE master encryption key is restricted from leaving Oracle Key Vault, the PKCS#11 library sends a request to Oracle Key Vault to decrypt the encrypted data encryption key. Decryption is then performed within Oracle Key Vault and afterward, the plaintext data encryption key is returned to the PKCS#11 library. To allow a symmetric or private key to leave Oracle Key Vault, you would set its extractable attribute value to true.

You can set the extractable attribute of symmetric or private keys in the following ways:

- **Setting the extractable attribute value for an existing symmetric or private key:** A user who has the Key Administrator role can modify the extractable attribute value of an existing symmetric or private key to be either true or false. A user or an endpoint with read-write access on an existing symmetric or private key can also modify its extractable attribute setting. However, this is allowed only to apply the stricter setting (that is, to set the value to false to make the symmetric or private key non-extractable). Such users or endpoints cannot modify the extractable attribute setting to make a symmetric or private key extractable if it is currently non-extractable.
- **Setting the default value of the extractable attribute globally for all endpoints:** You can set the default value of the extractable attribute in the global endpoint settings. This setting applies to all endpoints. This setting is used when an endpoint creates or registers a new symmetric or private key unless either of the following conditions occur:
 - The extractable attribute is set for the symmetric or private key at the time of its creation or registration.

- The default extractable attribute value has been set for that endpoint specifically (that is, the endpoint does not inherit this setting from the global endpoint).

This global endpoint setting does not apply to existing symmetric or private keys; it only applies to new symmetric or private keys that are created or registered after this setting has been configured.

- **Setting the default value of the extractable attribute for an individual endpoint:** You can set the default value of the extractable attribute for an individual endpoint. The endpoint specific setting takes precedence over the global endpoint setting. This endpoint specific extractable attribute setting applies when the endpoint creates or registers a new symmetric or private key unless the extractable attribute is set for the key at the time of its creation or registration itself. This individual endpoint setting does not apply to existing symmetric or private keys; it only applies to new symmetric or private keys that are created or registered by the endpoint after this setting has been configured.
- **Setting the extractable attribute value when you create or register a symmetric or private key:** You can set the extractable attribute value for a new symmetric key or private key at the time of its creation or registration using the C SDK, the Java SDK, or the RESTful services utility. The extractable attribute value specified at the time of key creation takes precedence over the endpoint's effective setting for the extractable attribute. However, this is subject to an additional restriction: You cannot set the extractable attribute of a new symmetric or private key to true, that is, create the new key as extractable, if the endpoint's effective setting for the extractable attribute is set to false (that is, the new symmetric keys or private are not extractable).

Be aware that setting the extractable attribute value to false may affect the performance of Oracle Key Vault. The performance impact may not be limited to Oracle Key Vault. The endpoint performance may be impacted as well.

Related Topics

- [About Managing Global and Per-Endpoint Configuration Parameters and Settings](#)
Users who have the System Administrator role or the Key Administrator role can centrally update certain endpoint configuration parameters and settings in the Oracle Key Vault management console.
- [Configuring Global Endpoint Settings for Keys and Secrets](#)
You can set the default extractable attribute value for new symmetric keys that you create or register in the endpoint configuration.
- [Configuring Endpoint Settings for Keys and Secrets for an Individual Endpoint](#)
A user who has the Key Administrator role can set values for keys and secrets in an individual endpoint.

Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys

You can configure the extractable attribute value of existing symmetric or private keys.

1. Log in to the Oracle Key Vault management console as one of the following types of users:
 - A user who has the Key Administrator role can modify the extractable attribute value of any symmetric or private key.
 - A user with read-modify access on a symmetric or private key can modify its extractable attribute value to only apply a stricter setting (that is, to set the value to false to make the object non-extractable).

2. Select the **Keys & Wallets** tab, and then **Keys & Secrets** from the left navigation bar.
The Keys & Secrets page appears.
3. For the key whose extraction that you want to configure, click the **Edit** icon, and then scroll down the Object Details page to the Advanced section.

Advanced

Cryptographic Algorithm -

Cryptographic Length -

Retrieved at least once No

Key Usage

- Decrypt
- Derive Key
- Encrypt
- Export
- Generate Cryptogram
- Translate Decrypt
- Translate Encrypt
- Translate Unwrap
- Translate Wrap
- Unwrap Key
- Validate Cryptogram
- Wrap Key

Extractable True

Never Extractable False

Contact Information

4. In the **Extractable** menu, select **True** or **False**.
 - **True** allows the object value to be extracted from Oracle Key Vault.
 - **False** prevents the object value from being extracted from Oracle Key Vault.
5. Select **Save**.

Managing Details of Security Objects

You can manage details about security objects, such as find details about these objects and modifying these details.

- [About Managing the Details of Security Objects](#)
You can search for security objects within a virtual wallet, and add, modify, or remove these security objects.
- [Searching for Security Object Items](#)
You can search for individual security objects if you have privileges to view these objects.
- [Viewing the Details of a Security Object](#)
An administrative user with the Key Administrator role can view, add, and modify the details of a security object.
- [Adding or Modifying Details of a Security Object](#)
Only users who have the appropriate privileges can add or modify the details of a security object.

About Managing the Details of Security Objects

You can search for security objects within a virtual wallet, and add, modify, or remove these security objects.

Security objects are managed by Oracle Key Vault administrative users with a clear separation of duties. You must be an administrative user with the Key Administrator role to manage wallet privilege on the virtual wallet containing the security objects. A user with the Audit Manager role can view security objects, but cannot modify them, whereas individual security objects are not even viewable to a user with the System Administrator role.

You can set the deactivation date for security objects and have an alert notify you when the security object will expire. For example, if you configure an alert for an object expiration with a threshold of 7 days, its expiration alert will be raised when object's deactivation date is within the 7 days of its deactivation date. An email notification will be sent every 24 hours during this threshold period. The alert is raised only when the security object is in the `PRE-ACTIVE` or `ACTIVE` state. Oracle Key Vault deletes the expiration alerts for the security objects when the security object is revoked or destroyed.

Related Topics

- [About Configuring Alerts](#)
System administrators can configure alerts from the Oracle Key Vault dashboard, but all users can see alerts for the security objects to which they have access.

Searching for Security Object Items

You can search for individual security objects if you have privileges to view these objects.

1. Log in to the Oracle Key Vault management console as a user with the Key Administrator role, an Audit Manager role, or as a user with access to a virtual wallet.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.

The Keys & Secrets page appears displaying all the security objects in a table.

The screenshot shows the 'Keys & Secrets' page in the Oracle Key Vault management console. At the top right, there is a 'Delete' button and a refresh icon. Below the title, there is a search bar with a magnifying glass icon, a 'Go' button, and an 'Actions' dropdown menu. The main content is a table with the following columns: 'Display Name', 'Type', 'Wallet Membership', 'Creating Endpoint', and 'State'. Each row has a checkbox in the 'Display Name' column. The table contains five rows of data:

<input type="checkbox"/>	Display Name	Type	Wallet Membership	Creating Endpoint	State
<input type="checkbox"/>	Default template for PEOPLESOFT_DB_CONNECT_CLIENT	Template	Peoplesoft_DB_Connect_Client_Wall	PEOPLESOFT_DB_CONNECT_CLIENT	No State
<input type="checkbox"/>	GoldenGate Master Key Name: OGG_MASTER_KEY_NAME, Version: 201	Symmetric Key		PRIMARY_OGG	Pre-Active
<input type="checkbox"/>	Default template for EXPENSE_DB_RAC_INST1	Template	HR_Application_Wallet	EXPENSE_DB_RAC_INST1	No State
<input type="checkbox"/>	Default template for WEBAPP_MYSQL	Template	WebApp_Wallet	WEBAPP_MYSQL	No State
<input type="checkbox"/>	Default template for ENCRYPTED_FILE_SYSTEM	Template	ASM_Cluster_File_System_Wallet	ENCRYPTED_FILE_SYSTEM	No State

By default, the table has the following columns for each security object:

- **Display Name** lists the name of the object.
- **Type:** Indicates the object type of security object. Valid values are **Symmetric Key**, **Public Key**, **Private Key**, **Template**, **Opaque Object**, **Certificate**, and **Secret Data**.
- **Wallet Membership:** The virtual wallet that contains the security object.
- **Creating Endpoint:** The endpoint that owns the security object.

- **State:** Indicates the state of the object. Valid values are **Active**, **Compromised**, **Deactivated**, **Destroyed**, **Destroyed Compromised**, and **Pre-Active**.
- **Extractable:** The extractable attribute setting of the security object.
- **Creation Date:** Date and time that the security object was added to Oracle Key Vault.
- **Deactivation Date:** Date and time that the security object was deactivated.
- **Name:** Actual name of the object.
- **Unique Identifier:** A globally unique ID that identifies an item.
- **Edit:** A pencil icon links to the Object Details page for the security object.

You can modify these columns to show more information. From the **Actions** menu, select **Select Columns**. In the Select Columns window, move the columns that you want to see to the Display in Report list, and then click **Apply**.

3. If the security object does not appear, then search for it using the Search bar or the **Actions** menu.

Viewing the Details of a Security Object

An administrative user with the Key Administrator role can view, add, and modify the details of a security object.

The administrative user can perform these actions on the security object from its corresponding Object Details page. Object details are attributes of a specific security object and depend on the type of security object.

1. Log in to the Oracle Key Vault management console as a user with the Key Administrator role or as a user with access to the virtual wallet.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the Keys & Secrets page, search for the security object that you want.

The Keys & Secrets page shows a table that displays the security objects in Key Vault.

You can modify the columns in this table to show more information. From the **Actions** menu, select **Select Columns**. In the Select Columns dialog box, move the columns that you want to see to the Display in Report list, and then click **Apply**.

4. Click the pencil icon in the **Edit** column corresponding to the security object.

The Object Details page appears displaying the attributes of the security object. The following screen shows a partial view of all the activities that you can perform on this object.

Object Details

Display Name: TDE Master Encryption Key: MKID Aa7JQ/p5I09zv5RK0ticuG0AAAAAAAAAAAA

Unique Identifier: 008512D5-9A5B-5339-B602-2DB5E48B5F22

Type: Symmetric Key

State: Active

Creator: XYZAPP_WEBAPP_MYSQL

Last Modified: 09-NOV-2020 19:00:07

Date of Creation: 09-NOV-2020 19:00:07

Date of Activation: 09-NOV-2020 19:00:07

Process Start Date:

Protect Stop Date:

Date of Deactivation:

Wallet Membership

	Wallet Name	Description	Creation Time
<input type="checkbox"/>	XYZ_MYSQL_WALLET	This wallet contains the MySQL DB Symmetric keys for the management of XYZ Application.	09-NOV-2020 15:53:37

You can set the dates when the security object should be deactivated or not used on the Object Details page. The attributes shown in Object Details depend on the type of security object. The attributes for a **Symmetric Key** are different from those of **Private Key** or **Opaque Object**.

You can revoke or destroy a security object, and add or remove it to and from a wallet from the Object Details page.

The Wallet Membership pane in the Object Details page enables you to add the security object to a wallet or delete the security object from a wallet.

The Object Details page contains the following attributes:

- **Display Name:** A summary description to help identify the item to the user. For example, if the item is a TDE master encryption key, then the **Identifier** shows the prefix TDE master encryption key followed by the identifier used by the database to identify the key.
- **Unique Identifier:** This is a globally unique ID that identifies an item.
- **Type:** Indicates the object type of the item. Valid values are **Symmetric Key**, **Public Key**, **Private Key**, **Template**, **Opaque Object**, **Certificate**, and **Secret Data**.
- **State:** Indicates the status of the security objects. Values are as follows:
 - **Pre-active:** The object exists but is not yet usable for any cryptographic purpose.
 - **Active:** The object is available for use. Endpoints should examine the Cryptographic Usage Mask attribute to determine which uses are appropriate for this object.
 - **Deactivated:** The object is no longer active and should not be used to apply cryptographic protection (for example, encryption or signing). It may still be appropriate to use for decrypting or verifying previously protected data.
 - **Compromised:** The object is believed to be compromised and should not be used.
 - **Destroyed:** The object is no longer usable for any purpose.

- **Destroyed Compromised:** The object was compromised and destroyed. It is no longer usable for any purpose.
 - **Creator:** The endpoint that created the security object.
 - **Last Modified:** The date last modified.
 - **Date of Creation:** The date created.
 - **Date of Activation:** The date of activation.
 - **Process Start Date:** The date when the key may start to be used to encrypt data. It can be equal or later than the **Date of Activation** setting but cannot precede it.
 - **Protect Stop Date:** When this date is passed, the key should not be used to encrypt any more data. It cannot be later than the **Date of Deactivation** setting.
 - **Date of Deactivation:** The date of deactivation.
5. Click **Advanced** to view the attributes of the security object.

Attribute information and queries will vary depending on the item type. Examples of attributes are as follows:

- **Cryptographic Algorithms:** The encryption algorithm used by the item
- **Key Usage:** Operations that the key can be used for. Clients may or may not use these attributes. For example, Transparent Data Encryption does not consult the key usage attributes.
- **Extractable:** Indicates if the symmetric or private key security object can be extracted. **TRUE** means that it can be extracted; **FALSE** means that it cannot be extracted.
- **Never Extractable:** Indicates if a security object (in this case, symmetric or private keys only) was never allowed to be extracted from Oracle Key Vault. **TRUE** means that the extractable attribute of the symmetric key has always been set to **FALSE**. If the **Extractable** attribute was ever (even once) set to **TRUE**, then the **Never Extractable** attribute becomes (and remains set to) **FALSE**.
- **Names:** Labels attached by a user or endpoint to identify the key
- **Custom Attributes:** Additional attributes defined by the endpoint and not interpreted by Oracle Key Vault
- **Cryptographic Parameters:** Optional parameters for the encryption algorithm used by the item, such as block cipher mode and padding method
- **Cryptographic Length:** The length in bits of the key
- **Retrieved at Least Once:** Indicates if the object has been served to the client
- **Contact Information:** Used for contact purposes only
- **Digests:** Digest values of the security object
- **Link Details:** Links to related objects

Related Topics

- [Key Management Interoperability Protocol Specification Version 1.1](#)

Adding or Modifying Details of a Security Object

Only users who have the appropriate privileges can add or modify the details of a security object.

To modify the attributes of a [security object](#) you must be a user with the Key Administrator role, or you must have **Read and Modify** access on the security object. For example, a user who has the Key Administrator role can modify the extractable attribute to apply its settings to all security objects in Oracle Key Vault. A user who has **Read and Modify** can set the extractable attribute for only objects that they create. You can get **Read and Modify** access on a security object if you own the security object or if you have access to a virtual wallet that contains the security object.

1. Log in to the Oracle Key Vault management console as a user with the Key Administrator role or as a user with access to a virtual wallet.

2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.

The Keys & Secrets page shows a table listing all the security objects in a table. You can modify the columns in this table to show more information. From the **Actions** menu, select **Select Columns**. In the Select Columns dialog box, move the columns that you want to see to the Display in Report list, and then click **Apply**.

3. Click the pencil icon corresponding to the security object.

The Object Details page appears.

4. In the Advanced pane, make the necessary changes.
5. Click **Save** in the top right corner of the pane.

Note:

Setting the date-time attributes (such as `Activation Date`, `Deactivation Date`, `Process Start Date`, and `Protect Stop Date`) for a security object to the epoch time (January 1st, 1970 at UTC) has the same effect of not setting the attribute at all. For example, if you set the `Activation Date` attribute of a security object to the epoch time to immediately activate the object, then the object remains in the Pre-Active state. This is because Oracle Key Vault treats the epoch value of the attribute as if the attribute is not set at all.

11

Managing Oracle Key Vault Master Encryption Keys

Managing security objects includes uploading and downloading security objects, managing the persistent master encryption key cache, and using user-defined TDE keys.

- [Using the Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache feature enables databases to be operational when the Oracle Key Vault server is unavailable.
- [Configuring an Oracle Key Vault to a New TDE-Enabled Database Connection](#)
You can configure a connection between Oracle Key Vault and a database that has not yet been configured for Transparent Data Encryption.
- [Migrating Existing TDE Wallets to Oracle Key Vault](#)
A migrated TDE wallet can be used to restore database contents that were previously encrypted by TDE.
- [Uploading and Downloading Oracle Wallets](#)
To store and share Oracle wallets, you must upload them to Oracle Key Vault.
- [Uploading and Downloading JKS and JCEKS Keystores](#)
The `okvutil upload` and `okvutil download` commands can upload and download JKS and JCEKS keystores.
- [Using a User-Defined Key as the TDE Master Encryption Key](#)
You can import a generated key to be used as the Transparent Data Encryption (TDE) master encryption key in Oracle Key Vault.

Using the Persistent Master Encryption Key Cache

The persistent master encryption key cache feature enables databases to be operational when the Oracle Key Vault server is unavailable.

- [About the Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache ensures the availability of TDE master encryption keys.
- [About Oracle Key Vault Persistent Master Encryption Key Cache Architecture](#)
The Oracle Key Vault persistent master encryption key cache is implemented in Oracle Key Vault's `PKCS#11` library.
- [Caching Master Encryption Keys in the In-Memory and Persistent Master Encryption Key Cache](#)
After a master encryption key is created or fetched from a different location, it is stored in an Oracle Key Vault cache.
- [Storage Location of Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache is created in the same location as the configuration file `okvclient.ora`.

- [Persistent Master Encryption Key Cache Modes of Operation](#)
The persistent master encryption key cache operates in two modes.
- [Persistent Master Encryption Key Cache Refresh Window](#)
The persistent master encryption key cache refresh window helps to extend the availability of the master encryption key.
- [Persistent Master Encryption Key Cache Parameters](#)
Oracle Key Vault provides parameters to configure the persistent master encryption key cache.
- [Listing the Contents of the Persistent Master Key Cache](#)
The `okvutil` list command can be used to list the master encryption keys that are cached in the persistent master encryption key cache.
- [Oracle Database Deployments and Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache affects the integration of other Oracle features with Oracle Key Vault.

About the Persistent Master Encryption Key Cache

The persistent master encryption key cache ensures the availability of TDE master encryption keys.

It accomplishes this by reducing dependence on the state of the Oracle Key Vault server.

The TDE master encryption key is cached in the persistent master encryption key cache in addition to the in-memory cache, to make the master encryption key available across database processes. It eliminates the need for databases to contact the Oracle Key Vault server for every new process, redo log switch, or database startup operation.

The following are the benefits of ensuring availability of TDE master encryption keys:

- Continuous operation of endpoints during upgrade, primary-standby configuration, switchover, failover, and other procedures that require an Oracle Key Vault restart operation
- Less load on the Oracle Key Vault server when multiple sessions of a single database request the same master encryption key
- Improved scalability of Oracle Key Vault

Note that the extractable attribute of the TDE master encryption key must be set to true, otherwise, the TDE master encryption key cannot leave Oracle Key Vault and thus it cannot be cached in either in-memory cache or the persistent master encryption key cache.

About Oracle Key Vault Persistent Master Encryption Key Cache Architecture

The Oracle Key Vault persistent master encryption key cache is implemented in Oracle Key Vault's `PKCS#11` library.

When the persistent master encryption key cache feature is configured, the Oracle Key Vault `PKCS#11` library will create the persistent master encryption key cache when the first master encryption key is retrieved from Oracle Key Vault.

The persistent master encryption key cache is an auto-login wallet or a password-based wallet, depending on how Oracle Key Vault is installed:

- If the Oracle Key Vault client is installed with a password specified, then the persistent master encryption key cache is a password-based wallet.
- If the Oracle Key Vault client is installed without a password specified, then the persistent master encryption key cache is an auto-login wallet.

The `PKCS#11` library also implements an in-memory master encryption key cache. When the in-memory master encryption key cache feature is configured, the master encryption key is cached in the process memory of the process that loaded the library into memory. The in-memory and persistent master encryption key caches are independent of each other. You can enable and disable these caches independently.

For operations that involve encryption and decryption, `PKCS#11` attempts to look up the master encryption key in the in-memory master encryption key cache. If it does not find it, `PKCS#11` then it looks up the master encryption key in the persistent master encryption key cache. If the master encryption key is not found in the in-memory or the persistent master encryption key cache, then it is retrieved from the Oracle Key Vault server, if the server is online.

Related Topics

- [Step 3: Install the Oracle Key Vault Software onto the Endpoint](#)
You can install the endpoint using downloaded `okvclient.jar` file.

Caching Master Encryption Keys in the In-Memory and Persistent Master Encryption Key Cache

After a master encryption key is created or fetched from a different location, it is stored in an Oracle Key Vault cache.

When the master encryption key is first fetched from the Oracle Key Vault server, or created in the Oracle Key Vault server, the master encryption key is stored in the in-memory master encryption key cache and in the persistent master encryption key cache.

Master encryption keys stored in the in-memory master encryption key cache are available for a limited time from the moment the key is placed into the persistent cache. The duration is defined by the `PKCS11_CACHE_TIMEOUT` parameter in the `okvclient.ora` file.

If the persistent cache exists, then it will be used. If the persistent cache does not exist, then Oracle Key Vault creates it. When the key is created, all future sessions will retrieve it from the in-memory master encryption key cache or persistent master encryption key cache.

Persistent master encryption keys that are stored in the persistent master encryption key cache are available for a limited time from the moment the key is placed into the persistent cache. You can define this time by setting the `PKCS11_PERSISTENT_CACHE_TIMEOUT` parameter in the `okvclient.ora` file.

When the endpoint deletes the master encryption key, the key will be removed from the in-memory master encryption key cache and persistent master encryption key cache.

Storage Location of Persistent Master Encryption Key Cache

The persistent master encryption key cache is created in the same location as the configuration file `okvclient.ora`.

The default location for the `okvclient.ora` file is the directory `$OKV_HOME/conf`.

It is important that the `ORACLE_HOME`, `ORACLE_BASE`, and `OKV_HOME` environment variables are consistently set across the deployment. If they are not consistent, then operations requiring the persistent cache may fail, and the persistent cache may be created in multiple locations.

If the environment variable `OKV_HOME` is set, then the persistent cache is created in `$(OKV_HOME)/conf`.

If `OKV_HOME` is not set, but `ORACLE_BASE` is set, then the persistent cache is created in `$(ORACLE_BASE)/okv/$(ORACLE_SID)`.

If neither `OKV_HOME` nor `ORACLE_BASE` is set, but `ORACLE_HOME` is set, then the persistent cache is created in `$(ORACLE_HOME)/okv/$(ORACLE_SID)`.



Note:

Ensure that the directory in which the persistent cache is created is secure and has restricted permissions.

Persistent Master Encryption Key Cache Modes of Operation

The persistent master encryption key cache operates in two modes.

The difference between the two modes is the order in which the persistent master encryption key cache and Oracle Key Vault are looked up to retrieve the master encryption key.

- [Oracle Key Vault First Mode](#)
In Oracle Key Vault first mode, the endpoints attempt to retrieve the master encryption key from the Oracle Key Vault server.
- [Persistent Master Encryption Key Cache First Mode](#)
In persistent master encryption key cache first mode, the endpoints retrieve the master encryption key from the persistent master encryption key cache.

Oracle Key Vault First Mode

In Oracle Key Vault first mode, the endpoints attempt to retrieve the master encryption key from the Oracle Key Vault server.

If the Oracle Key Vault server is offline, then the endpoints attempt to retrieve the master encryption key from the persistent master encryption key cache.

The endpoints must determine the status of the Oracle Key Vault server, and if it is offline, then the endpoints attempt to retrieve the master encryption key from the persistent master encryption key cache. Hence, database operations that require access to master encryption keys will experience a delay.

Persistent Master Encryption Key Cache First Mode

In persistent master encryption key cache first mode, the endpoints retrieve the master encryption key from the persistent master encryption key cache.

If the master encryption key is not available in the persistent master encryption key cache, then the endpoints attempt to retrieve the master encryption key from the Oracle Key Vault server.

The modifications to the master encryption keys on the Oracle Key Vault server are not applied until the key expires in the persistent master encryption key cache.

Persistent Master Encryption Key Cache Refresh Window

The persistent master encryption key cache refresh window helps to extend the availability of the master encryption key.

The refresh window feature of the persistent master encryption key cache enables the database endpoint to make multiple attempts to refresh the expired master encryption key from the Oracle Key Vault server. In that sense, the endpoint waits for the Oracle Key Vault server to be back online for the master encryption key refresh to complete. Meanwhile, if the master encryption key refresh attempt fails, then the keys are retrieved from the persistent cache for the duration of the refresh window.

The refresh window feature of the persistent master encryption key cache therefore extends the duration for which the master encryption key is available after it is cached in the persistent master encryption key cache. At the same time, the endpoints can refresh the key during the refresh window instead of once at the end of the cache time. This addresses the possibility that persistent cache expires during the time when the Oracle Key Vault is unavailable, such as when a primary-standby switchover is in progress. The refresh window terminates and then the cache period begins as soon as the key is refreshed.

In the `okvclient.ora` file, you can use the `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` parameter to extend the duration for which the master encryption key is available after it is cached in the persistent master encryption key cache. This value reflects the amount of time it takes for the Oracle Key Vault server to recover and return online. You must specify this value in minutes. The default value for `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` is 30 (minutes).

Related Topics

- [PKCS11_PERSISTENT_CACHE_TIMEOUT Parameter](#)
The `PKCS11_PERSISTENT_CACHE_TIMEOUT` parameter sets how long the master encryption is available in the persistent cache.

Persistent Master Encryption Key Cache Parameters

Oracle Key Vault provides parameters to configure the persistent master encryption key cache.

- [PKCS11_CACHE_TIMEOUT Parameter](#)
The `PKCS11_CACHE_TIMEOUT` parameter sets how long a master encryption key is available in the in-memory cache.
- [PKCS11_PERSISTENT_CACHE_TIMEOUT Parameter](#)
The `PKCS11_PERSISTENT_CACHE_TIMEOUT` parameter sets how long the master encryption is available in the persistent cache.
- [PKCS11_PERSISTENT_CACHE_FIRST Parameter](#)
The `PKCS11_PERSISTENT_CACHE_FIRST` parameter sets the persistent master encryption key cache operation mode.
- [PKCS11_CONFIG_PARAM_REFRESH_INTERVAL Parameter](#)
The `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL` parameter describes the frequency at which a long-running process will re-read the `okvclient.ora` configuration file.
- [PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW Parameter](#)
The `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` parameter extends time the master encryption key is available after it is cached in the persistent master encryption key cache.

- [EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN Parameter](#)
The `EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN` parameter ensures that the PKCS#11 persistent cache for a given endpoint database automatically expires upon shutdown of the endpoint database.

PKCS11_CACHE_TIMEOUT Parameter

The `PKCS11_CACHE_TIMEOUT` parameter sets how long a master encryption key is available in the in-memory cache.

You must specify the value in minutes. When the specified duration of time elapses, the master encryption key expires. Keys are deleted from the in-memory cache.

The default value for `PKCS11_CACHE_TIMEOUT` is 60 (minutes). Oracle recommends that you set the `PKCS11_CACHE_TIMEOUT` parameter in the Oracle Key Vault management console.

Related Topics

- [Setting Global Endpoint Configuration Parameters](#)
You can set global endpoint configuration parameters in the Oracle Key Vault management console.

PKCS11_PERSISTENT_CACHE_TIMEOUT Parameter

The `PKCS11_PERSISTENT_CACHE_TIMEOUT` parameter sets how long the master encryption is available in the persistent cache.

This time starts when the database retrieves the key from the Oracle Key Vault server and puts it in the cache. After this duration has elapsed, the master encryption key expires. At this time, the endpoint will attempt to contact the Oracle Key Vault server in order to retrieve the key, and if it succeeds, the key remains available for another duration specified by this parameter. If it is unable to retrieve the key, the key remains available for the amount of time dictated by the `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` parameter, after which the database can no longer use the key without successfully retrieving it again from the Oracle Key Vault server. Encryption keys are deleted from persistent master encryption key cache.

The `Cache Start Time` and `Maximum Use Time` values displayed in the OKV Persistent Cache entries list is updated when the master encryption key is refreshed.

The default value for `PKCS11_PERSISTENT_CACHE_TIMEOUT` is 1440 (minutes).

You can disable the persistent master encryption key cache by setting both the `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` and the `PKCS11_PERSISTENT_CACHE_TIMEOUT` parameters to 0 (zero).

Oracle recommends that you set this global parameter in the Oracle Key Vault management console.

Note:

The parameter `PKCS11_PERSISTENT_CACHE_TIMEOUT` and its default value are included by default in the `okvclient.ora` file.

Related Topics

- [Setting Global Endpoint Configuration Parameters](#)
You can set global endpoint configuration parameters in the Oracle Key Vault management console.

PKCS11_PERSISTENT_CACHE_FIRST Parameter

The `PKCS11_PERSISTENT_CACHE_FIRST` parameter sets the persistent master encryption key cache operation mode.

You set the `PKCS11_PERSISTENT_CACHE_FIRST` parameter in the `okvclient.ora` file.

The following are the modes of operation:

- **Oracle Key Vault First Mode:** To enable Oracle Key Vault first mode, set the value of the `PKCS11_PERSISTENT_CACHE_FIRST` parameter to 0 (zero).
- **Persistent Master Encryption Key Cache First Mode:** Persistent master encryption key cache first mode is the default mode.

To enable persistent master encryption key cache first mode, set the value of the `PKCS11_PERSISTENT_CACHE_FIRST` parameter to 1.

Related Topics

- [Oracle Key Vault First Mode](#)
In Oracle Key Vault first mode, the endpoints attempt to retrieve the master encryption key from the Oracle Key Vault server.
- [Persistent Master Encryption Key Cache First Mode](#)
In persistent master encryption key cache first mode, the endpoints retrieve the master encryption key from the persistent master encryption key cache.

PKCS11_CONFIG_PARAM_REFRESH_INTERVAL Parameter

The `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL` parameter describes the frequency at which a long-running process will re-read the `okvclient.ora` configuration file.

This parameter also describes the frequency at which a process will attempt to remove expired master encryption keys from the persistent cache.

When the process cannot use a key from the in-memory cache and instead reaches out to the persistent cache or the Oracle Key Vault server, if it has been longer than the value specified by `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL` since `okvclient.ora` was last read, the process will re-read `okvclient.ora`, start using any changed parameters, re-read the `okvclient.ora` configuration file, start using any changed parameters, and remove expired master encryption keys from the persistent cache. Note that if the parameter for the in-memory cache and if `PKCS11_CACHE_TIMEOUT`, is larger than `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL`, then these operations will be performed at intervals described by the `PKCS11_CACHE_TIMEOUT` parameter instead.

Oracle recommends that you set the `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL` parameter in the Oracle Key Vault management console. You must specify this value in minutes. The default value for `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL` is 10 (minutes).

You can disable this parameter by setting the `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL` parameter to 0 (zero).

Related Topics

- [Setting Global Endpoint Configuration Parameters](#)
You can set global endpoint configuration parameters in the Oracle Key Vault management console.

PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW Parameter

The `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` parameter extends time the master encryption key is available after it is cached in the persistent master encryption key cache.

Oracle recommends that you set these global parameters in the Oracle Key Vault management console. You must specify the value in minutes. The default value for `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` is 30 (minutes).

You can disable the persistent master encryption key cache by setting the `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` and `PKCS11_PERSISTENT_CACHE_TIMEOUT` parameters to 0 (zero).

Related Topics

- [Setting Global Endpoint Configuration Parameters](#)
You can set global endpoint configuration parameters in the Oracle Key Vault management console.

EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN Parameter

The `EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN` parameter ensures that the `PKCS#11` persistent cache for a given endpoint database automatically expires upon shutdown of the endpoint database.

When enabled, the `EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN` protects the persistent cache by using a system-generated random password that is independent of the password that was set when an endpoint database was enrolled in Oracle Key Vault, even if an auto-login wallet was used. Having the persistent cache password protected provides better security.

Before you can use the `EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN` parameter, ensure that the endpoint database has had the patch for bug 29869906: `AUTO-LOGIN OKV NEEDS PERSISTENT CACHE PROTECTION KEY FROM RDBMS` applied to it. This patch applies to Oracle Database releases 12.1 through 19c. Contact Oracle Support for more information.

You can set `EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN` for individual endpoint databases that have been enrolled with Oracle Key Vault, or globally for all endpoint databases that have been enrolled in Oracle Key Vault. This parameter is not available in the `okvclient.ora` configuration file for the database endpoint. To set this parameter, use the Oracle Key Vault management console.

After you have enabled `EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN`, the `PKCS#11` persistent cache is created when keys are fetched from Oracle Key Vault. The cache remains available to the endpoint database only as long as the database instance is mounted or open. When the endpoint database is shut down, the `PKCS#11` persistent cache is no longer available, but is recreated the next time the endpoint database is started. The persistent cache does, however, remain available when an endpoint pluggable database (PDB) is closed and then re-opened.

Be aware that after you have enabled `EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN` for a given endpoint database, you can no longer use the `okvutil list -t`

`okv_persistent_cache` command to view the contents of the persistent cache. In addition, you must ensure that Oracle Key Vault is available when keys are fetched after the endpoint database is started.

Setting EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN for Individual Endpoint Databases

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Click the **Endpoints** tab.
3. On the Endpoints page, select the endpoint for which you want to set `EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN`.
4. On the Endpoint Details page, scroll to the bottom and then set the **Expire PKCS11 Persistent Cache on Database Shutdown** checkbox.
5. Click **Save**.

Setting EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN Globally

The following procedure will apply the `EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN` to all current and future endpoint databases that have been enrolled with Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Endpoints** tab, and then **Settings** from the left side bar.
3. In the Global Endpoint Configuration Parameters page, set the **Expire PKCS11 Persistent Cache on Database Shutdown** checkbox.
4. Click **Save**.

Listing the Contents of the Persistent Master Key Cache

The `okvutil list` command can be used to list the master encryption keys that are cached in the persistent master encryption key cache.

Oracle Key Vault automatically removes master encryption keys that are expired after you run the `okvutil list` command on the persistent cache.

- Run the `okvutil list` command, similar to the following example:

```
$ ./okvutil list -t okv_persistent_cache -l $ORACLE_HOME/okv/$ORACLE_SID
Enter Oracle Key Vault endpoint password: password
```

Output similar to the following appears:

```
OKV Persistent Cache entries:
Current Persistent Cache Timeout is 600 seconds
Version Unique ID                               TDE Master Key Identifier
Cache Start Time   Maximum Use Time  Maximum Refresh Window  Status
02      55D745B1-2F30-667F-E053-0100007FAFDB  0636846AAF88F74FC6BF1DB68538797B69
22:38:12 2019-08-03 600 seconds      0 seconds                Expired
02      55D745B1-2F2E-667F-E053-0100007FAFDB  063AC48E9433734F7EBF97180276E719C4
22:37:10 2019-08-03 600 seconds      180 seconds              Available
02      55D745B1-2F2D-667F-E053-0100007FAFDB  0604425983989C4F6ABF7BD9E1D55459C4
22:37:00 2019-08-03 600 seconds      180 seconds              Available
02      55D70FA4-81D1-5C8A-E053-0100007F8217  06172EACB79F4C4F32BFB7D50B0ACA7101
03:44:22 2019-08-03 300 seconds      0 seconds                Expired
```

```

02      55D745B1-2F2B-667F-E053-0100007FAFDB 06983C4664FFC04F6ABF72F961A15AD943
22:36:49 2019-08-03 600 seconds      300 seconds      Available
02      55D745B1-2F29-667F-E053-0100007FAFDB 0639E05D58B27B4FFDBFAEC5EAA08DB301
03:26:40 2019-08-03 300 seconds      0 seconds        Expired
02      55D745B1-2F28-667F-E053-0100007FAFDB 06A29F4039E1B74FDCBFA687E0608EEEEBA
03:19:17 2019-08-03 300 seconds      0 seconds        Expired
02      55D745B1-2F27-667F-E053-0100007FAFDB 0678287C2877B74FF3BF0BA33A17A59F94
03:19:21 2019-08-03 300 seconds      0 seconds        Expired

```

The following table describes the columns in the OKV Persistent Cache entries list:

Column Name	Description
Version	Persistent master encryption key cache version
Unique ID	KMIP identifier assigned to the master encryption key
TDE Master Key Identifier	Database ID assigned to the master encryption key
Cache Start Time	Time at which the master encryption key was cached
Maximum Use Time	Time until the master encryption key expires, in seconds, from the moment that the key was placed into the persistent master encryption key cache
Maximum Refresh Window	Extended duration for which the master encryption key is available after it is cached in the persistent master encryption key cache
Status	Indicates whether the master encryption key is available, refreshing or expired

Oracle Database Deployments and Persistent Master Encryption Key Cache

The persistent master encryption key cache affects the integration of other Oracle features with Oracle Key Vault.

- **Database restart when the Oracle Key Vault Server is offline:** When you configure Oracle Key Vault to use an auto-login wallet, the database connects to the Oracle Key Vault server when the database is restarted. If the Oracle Key Vault server is offline when the database restarts, then the database retrieves master encryption keys from the persistent master encryption key cache. Database operations resume normally if the master encryption keys are active and have not expired.

Ensure that the passwords of the persistent master encryption key cache and the Oracle Key Vault endpoint wallet are synchronized.

 **Note:**

The persistent master encryption key cache must be deleted when the endpoint wallet credentials are modified.

- **Using the persistent master encryption key cache in an Oracle Real Application Cluster (Oracle RAC) environment:** In an Oracle RAC environment, each Oracle RAC

node is a unique database endpoint, and uses a unique persistent master encryption key cache.

In an Oracle RAC Environment, you must query the database from each Oracle RAC node to cache the most recent version of the master encryption key in the persistent master encryption key cache of each Oracle RAC node.

- **Using persistent master encryption key cache in an Oracle Data Guard Environment:** Rotation of the master encryption key in the primary server's database caches the master encryption key in the persistent master encryption key cache of the primary server's database.

The standby server retrieves and caches the new master encryption key in the persistent master encryption key cache of the standby server's database after the new `REDO` logs from the primary server are applied on the standby server. To avoid disruptions, you should synchronize the primary and standby servers immediately after the rotation of the master encryption key in the primary server's database.

Configuring an Oracle Key Vault to a New TDE-Enabled Database Connection

You can configure a connection between Oracle Key Vault and a database that has not yet been configured for Transparent Data Encryption.

- [About Configuring an Oracle Key Vault to a New TDE-Enabled Database Connection](#)
You can configure a connection between Oracle Key Vault and a database that has not yet been configured for TDE.
- [Limitations to Transparent Data Encryption Endpoint Integration](#)
This type of Transparent Data Encryption (TDE) endpoint integration can have problems if the versions are incompatible.
- [Step 1: Configure the Oracle Key Vault Server Environment](#)
Before you can configure the connection, you must ensure that Oracle Database and Oracle Key Vault settings are correct.
- [Step 2: Integrate Transparent Data Encryption with Oracle Key Vault](#)
This integration enables Oracle Key Vault to directly manage the TDE master encryption keys.

About Configuring an Oracle Key Vault to a New TDE-Enabled Database Connection

You can configure a connection between Oracle Key Vault and a database that has not yet been configured for TDE.

Before you start configuring the connection, ensure that the Oracle Key Vault client installation environment is the same as the database runtime environment. The environment variables `ORACLE_HOME`, `ORACLE_BASE`, and `ORACLE_SID` must be set to the same values in `svrctl` and operating system environment variables. This also applies if you are using the Oracle Key Vault RESTful services utility to enroll endpoints.

For information about configuring Oracle Data Guard so that it can work with TDE and Oracle Key Vault:

- [Oracle Database Advanced Security Guide](#), release 19c

- [Oracle Database Advanced Security Guide](#), release 18c
- [Oracle Database Advanced Security Guide](#), release 12.2.0.1

For information about configuring Oracle Real Application Clusters (Oracle RAC) to work with TDE and Oracle Key Vault:

- [Oracle Database Advanced Security Guide](#), release 19c
- [Oracle Database Advanced Security Guide](#), release 18c
- [Oracle Database Advanced Security Guide](#), release 12.2.0.1

For more information on the steps to clone PDB when PDB keys are stored in Oracle Key Vault, see in ASO admin guide:

- [Managing Cloned PDBs with Encrypted Data in United Mode](#)
- [Cloning a PDB with Encrypted Data in a CDB in Isolated Mode](#)

Limitations to Transparent Data Encryption Endpoint Integration

This type of Transparent Data Encryption (TDE) endpoint integration can have problems if the versions are incompatible.

The limitations to TDE endpoint integration are as follows:

- All endpoints on the same computer must use the same version of the Oracle Key Vault library. There is only one location per computer for the `liborapkcs.so` file, which is `/opt/oracle/expapi/64/hsm/oracle/1.0.0/liborapkcs.so`.
- If you have multiple databases on the same host, and those databases share the same `sqlnet.ora` file, then you must migrate all databases at the same time. If each database has its own `sqlnet.ora` file (controlled by individual settings for the `TNS_ADMIN` parameter), then you can migrate the databases into Oracle Key Vault at independent points in time.

Caution:

Oracle strongly recommends that you never remove keys from a wallet or the wallet itself after TDE is configured. Loss of keys will result in the loss of the database. This is true even in the following scenarios:

- If you had created or opened a keystore (wallet or Oracle Key Vault), even before setting the first key and before encrypting data in your database
- If all of the encrypted data has been decrypted
- If you have migrated your keys and wallets to a hardware security module

Step 1: Configure the Oracle Key Vault Server Environment

Before you can configure the connection, you must ensure that Oracle Database and Oracle Key Vault settings are correct.

These settings include Oracle environment variables and the Oracle `COMPATIBILITY` parameter.

1. Log in to the server where the database endpoint is installed.

2. Ensure that the `ORACLE_BASE` environment variable is set for the current operating system session, as well as in `srvctl` for Oracle Real Application Clusters (Oracle RAC)-enabled databases, before you start the database.

This step is very important. If the `ORACLE_BASE` environment variable is not present, then create a soft link from the `$ORACLE_BASE/okv/$ORACLE_SID/okvclient.ora` file to the `key_vault_endpoint_installation_dir/conf/okvclient.ora` file. In an Oracle Real Application Clusters environment, you must perform this step on all database instances.

3. Ensure that the `COMPATIBILITY` initialization parameter is set to `12.1.0.0` or later.
4. Enroll and provision the endpoint for the Transparent Data Encryption (TDE)-enabled database that contains the TDE data.

When you initially enroll the endpoint, select `ORACLE_DB` as the endpoint type for integration with TDE.

5. Ensure that the endpoint has access to the virtual wallet that you want to use by defining the wallet as the **default wallet** for that endpoint.

In Oracle RAC and Oracle Data Guard environments, the endpoints of all primary and standby instances share the same default wallet.

6. Depending on the Oracle Database release, configure the encryption wallet location.
 - For Oracle Database release 12.1.0.2, configure the `sqlnet.ora` file on this database to point to Oracle Key Vault as follows:

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=HSM))
```

- For Oracle Database release 12.2.0.1, configure the `sqlnet.ora` file on this database to point to Oracle Key Vault as follows:

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=OKV))
```

By default, the `sqlnet.ora` file is located in the `ORACLE_HOME/network/admin` directory or in the location set by the `TNS_ADMIN` environment variable.

- For Oracle Database release 18c or later, configure the following parameters:
 - Set `WALLET_ROOT` to define the installation directory of the Oracle Key Vault client. The directory that will be defined as `WALLET_ROOT` must have the following sub-directories (in lower case) in order for the database to auto-discover the correct keystore configuration:
 - * `/okv`: The Oracle Key Vault client software is installed into this directory.
 - * `/tde`: The TDE wallet (or the auto-open wallet for Oracle Key Vault) goes into this sub directory.
 - * `/tde_seps`: This directory is for a wallet that will hide the keystore password (wallet or Oracle Key Vault) from the `ADMINISTER KEY MANAGEMENT` command when the `EXTERNAL STORE` clause is used to replace the password.
 - Set `TDE_CONFIGURATION` to `OKV` for Oracle Key Vault, or `OKV|FILE` for an auto-open connection into Oracle Key Vault.

Endpoints use the PKCS#11 library support to manage TDE master encryption keys. Run the `root.sh` script (in the `okvclient_installation_directory/bin` directory) to deploy the PKCS#11 library that facilitates the communication between the endpoint and Oracle Key Vault.

At this stage, the Oracle database will use Oracle Key Vault for centralized key management. For all TDE commands and statements, use the Oracle Key Vault endpoint password that was specified during the installation of the endpoint software, unless the keystore password is hidden from the `ADMINISTER KEY MANAGEMENT` commands in a wallet that was created in the `WALLET_ROOT/tde_seps` directory.

7. Reconnect to the database if you are in SQL*Plus.

The changes will appear after you log out of the current SQL*Plus session and then connect again.

8. Query the `V$ENCRYPTION_WALLET` dynamic view to ensure that the `METHOD_DATA` setting in the `sqlnet.ora` file changed.

```
SELECT * FROM V$ENCRYPTION_WALLET;
```

The output of the query should now show `OKV`.

9. Configure the **Extractable** attribute value for the TDE master encryption key.

This setting determines whether the symmetric key can be retrieved (extracted) from Oracle Key Vault during certain operations.

As a user who has the Key Administrator role, you can set the default value of the **Extractable** attribute for the new symmetric key in the Endpoint Settings for Keys and Secrets page. Per-endpoint setting, if any, overrides the global endpoint setting. You can also set this attribute value explicitly at the time of creating or registering a new symmetric key (using the C or JAVA SDK, or the RESTful services utility). However, this is subject to an additional restriction: You cannot set the extractable attribute of a new symmetric key to **TRUE** (that is, create the new key as extractable) if the effective endpoint setting for the extractable attribute is set to **FALSE** (that is, the new symmetric key are not extractable).

As a user who has the Key Administrator role, you can modify the extractable attribute setting of an existing symmetric key to either **TRUE** or **FALSE**.

A user or an endpoint with read/write access on an existing symmetric key can also modify its extractable attribute setting, however, this is allowed only to apply the stricter setting, that is, to set the value to **FALSE** to make the object non-extractable. Such users or endpoints cannot modify the **Extractable** attribute setting to make a symmetric key extractable if it is currently non-extractable.

The TDE configuration is complete at this stage. You can now encrypt existing table spaces or create new encrypted table spaces in the database. Oracle does not recommend using TDE column encryption with any external key manager. If you have configured the `sqlnet.ora` file correctly along with the rest of the TDE configuration, then a TDE master encryption key is created in Oracle Key Vault when you set the encryption key by using one of the following SQL statements:

- `ALTER SYSTEM SET [ENCRYPTION] KEY IDENTIFIED BY "password";`
- `ADMINISTER KEY MANAGEMENT SET [ENCRYPTION] KEY IDENTIFIED BY "password"`

With both of these SQL statements, the password was defined when the Oracle Key Vault client software was installed. If no password was defined at that time, then the password for these two statements is `NULL`.

Related Topics

- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

- [Configuring Global Endpoint Settings for Keys and Secrets](#)
You can set the default extractable attribute value for new symmetric keys that you create or register in the endpoint configuration.

Step 2: Integrate Transparent Data Encryption with Oracle Key Vault

This integration enables Oracle Key Vault to directly manage the TDE master encryption keys.

1. Ensure that you are logged into the server where the Oracle Key Vault client is installed.

Note:

Consider the following points when executing `root.sh` script:

- You must run `root.sh` to install the Oracle Key Vault library only once on a host with multiple Oracle databases.
- Ensure that you execute the `root.sh` script only after installation of Oracle Key Vault clients for all of the TDE-enabled databases on the same host is complete.
- Ensure that all of the Oracle databases on this host are shutdown.

2. Shutdown the database server.

This script implements the persistent master encryption key cache feature in the Oracle Key Vault PKCS#11 library, which improves the availability of the database during intermittent network disruptions or Oracle Key Vault upgrade.

- **UNIX:** The root script copies the PKCS#11 library into `/opt/oracle/extapi/64/hsm/oracle/1.0.0` and minimizes the file access privileges.
- **Windows:** Run the `root.bat` script to copy the `liborapkcs.dll` file (located in the `lib` directory) to the `C:\oracle\extapi\64\hsm\oracle\1.0.0` directory. Provide the database version when prompted.

3. Run the `root` script from `okvclient_installation_directory/bin`.

4. Start the Oracle database.

5. For password-protected wallets on the database, open the wallet. (Auto-login wallets are automatically opened.)

- **For Oracle Database release 12.1.0.2 or later:** As a user who has been granted the SYSKM administrative privilege:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
"Oracle_Key_Vault_endpoint_password";
```

6. Set the master encryption key.

- **For Oracle Database release 12.1.0.2 or later:**

```
ADMINISTER KEY MANAGEMENT SET [ENCRYPTION] KEY IDENTIFIED BY
"Oracle_Key_Vault_endpoint_password" WITH BACKUP;
```

Related Topics

- [Enrolling and Upgrading Endpoints for Oracle Key Vault](#)
After an endpoint is registered in Oracle Key Vault, an endpoint administrator enrolls and provisions the endpoint to manage security objects in Key Vault.

- [Managing Endpoints](#)
You can enroll, reenroll, suspend, rotate, and delete endpoints.
- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.

Migrating Existing TDE Wallets to Oracle Key Vault

A migrated TDE wallet can be used to restore database contents that were previously encrypted by TDE.

- [About Migrating Existing TDE Wallets to Oracle Key Vault](#)
The `sqlnet.ora` file enables the migration of existing TDE wallets to Oracle Key Vault.
- [Migrating an Existing TDE Wallet to Oracle Key Vault](#)
You can use the `okvutil upload` command to start the migration of a TDE-enabled database from an existing TDE wallet to Oracle Key Vault.
- [Restoring Database Contents Previously Encrypted by TDE Using an Oracle Wallet](#)
You perform the restoration process on the endpoint where you downloaded the Oracle wallet.

About Migrating Existing TDE Wallets to Oracle Key Vault

The `sqlnet.ora` file enables the migration of existing TDE wallets to Oracle Key Vault.

For Oracle Database release 12.2.0.1 or earlier, the `sqlnet.ora` file enables the migration of existing TDE wallets to Oracle Key Vault. For Oracle Database release 18c or later, you must set the parameters `TDE_CONFIGURATION` and `WALLET_ROOT` to migrate wallets to Oracle Key Vault.

When the Transparent Data Encryption (TDE) wallets already exist, you must modify the `sqlnet.ora` file or the `TDE_CONFIGURATION` parameter to recognize Oracle Key Vault before you can migrate the existing TDE wallets to Oracle Key Vault.

Along with the current TDE master encryption key, Oracle wallets maintain historical TDE master encryption keys that are replaced by each rekey operation that rotates the TDE master encryption key. These historical TDE master encryption keys help to restore Oracle Database backups that were previously made using one of the historical TDE master encryption keys. During the TDE migration from an Oracle wallet file to Oracle Key Vault, Key Vault generates new master encryption keys. After this master encryption key generation, Oracle Key Vault maintains all new keys.

Oracle recommends that you upload the Oracle wallet to Oracle Key Vault before you perform the migration. This enables you to keep a backup of the wallet with all of the historical key information, before you begin the migration. When the migration is complete, manually delete the old wallet on the client system.

If you are operating on an Oracle Real Application Clusters (Oracle RAC) configuration, ensure that you migrate from a shared TDE wallet (in Oracle Automatic Storage Management (Oracle ASM) or Oracle Advanced Cluster File System (Oracle ACFS)) to Oracle Key Vault. Individual wallets for each Oracle RAC node are not supported.

Related Topics

- [Migrating Between a Software Password Keystore and an External Keystore](#)

- [Restoring Database Contents Previously Encrypted by TDE Using an Oracle Wallet](#)
You perform the restoration process on the endpoint where you downloaded the Oracle wallet.
- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

Migrating an Existing TDE Wallet to Oracle Key Vault

You can use the `okvutil upload` command to start the migration of a TDE-enabled database from an existing TDE wallet to Oracle Key Vault.

1. Back up the database that you want to migrate.
2. Complete the enrollment of the endpoint.
3. If you have not done so already, then use the `okvutil upload` command to upload current and historical master encryption keys from the TDE wallet into the virtual wallet in Oracle Key Vault.

```
$ OKV_HOME/bin/okvutil upload -t WALLET -l /path/to/tde-wallet -g
name_of_wallet_in_Oracle_Key_Vault -v 4
```

This step allows you to delete the wallet after a successful migration, which is often a requirement of security policies that do not allow encryption keys on the encrypting server.

Be aware that the **Extractable** attribute setting for the objects that are created in Oracle Key Vault using `okvutil upload` is inherited from the endpoint. You can download the wallet, but if the wallet's objects have the **Extractable** attribute set to `FALSE`, then those objects cannot be downloaded.

4. Configure the wallet location.
 - **For Oracle Database release 12.2.0.1 or earlier:** Edit the `METHOD` setting of the Oracle Database `sqlnet.ora` file for the migration to Oracle Key Vault:

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=OKV)
(METHOD_DATA=(DIRECTORY=wallet_location)))
```

By default, the `sqlnet.ora` file is located in the `ORACLE_HOME/network/admin` directory or in the location set by the `TNS_ADMIN` environment variable.

- **For Oracle Database release 18c or later:** Set the `TDE_CONFIGURATION` parameter.

```
ALTER SYSTEM SET TDE_CONFIGURATION = "KEystore_CONFIGURATION=OKV|FILE" SCOPE =
BOTH;
```

The `WALLET_ROOT` parameter is already set, so you do not need to change it.

5. Reconnect to the database instance (or restart an Oracle RAC database) to capture the changes in the `sqlnet.ora` file.

If you are reconnecting to the database instance, then changes to the `TDE_CONFIGURATION` parameter in Oracle Database 18c or later are applied immediately, and no restart is necessary.

6. Query the `V$ENCRYPTION_WALLET` dynamic view to ensure that the `METHOD_DATA` setting in the `sqlnet.ora` file changed.

```
SELECT CON_ID, WALLET_TYPE, WALLET_ORDER, STATUS
FROM V$ENCRYPTION_WALLET
WHERE CON_ID <> 2;
```

The output of the query should now show METHOD=OKV.

7. If the endpoint is a an Oracle Release 11g release 2 database, then close the local Oracle wallet and open the HSM wallet as follows:
8. Close the local Oracle wallet using these steps:

- a. Close the local Oracle wallet as follows:

If the auto-login wallet is open, run the following commands:

```
oracle$ cd wallet_location
oracle$ mv cwallet.sso cwallet.sso.bak
sqlplus sys as sysdba
Enter password: password
SQL> ALTER SYSTEM SET WALLET CLOSE;
```

If the password-protected wallet is open, then run the following statement in SQL*Plus:

```
ALTER SYSTEM SET WALLET CLOSE IDENTIFIED BY "wallet_password";
```

- b. In SQL*Plus, open the connection into Oracle Key Vault, where *Oracle_Key_Vault_password* is the password that was provided when the *okvclient.jar* file was installed:

```
ALTER SYSTEM SET WALLET OPEN IDENTIFIED BY "Oracle_Key_Vault_password";
```

9. Migrate from TDE wallets to Oracle Key Vault.

- **For Oracle Database 11g release 2:** If you entered a password for the wallet while installing the endpoint client software, then in SQL*Plus, run this statement:

```
ALTER SYSTEM SET ENCRYPTION KEY
IDENTIFIED BY "Oracle_Key_Vault_password"
MIGRATE USING "wallet_password";
```

If you chose the auto-login option while installing the endpoint client software, then run this statement:

```
ALTER SYSTEM SET ENCRYPTION KEY
IDENTIFIED BY "NULL"
MIGRATE USING "wallet_password";
```

- **For Oracle Database 12c or later:** As a user who has been granted the SYSKM administrative privilege:

```
ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY IDENTIFIED BY
"Oracle_Key_Vault_password" MIGRATE USING "wallet_password" WITH BACKUP;
```

Though the WITH BACKUP clause is required for the ADMINISTER KEY MANAGEMENT statement, it is ignored by TDE in Oracle Key Vault.

10. Open the wallet.

- **For Oracle Database 11g release 2:** If the endpoint requires a password to connect to Oracle Key Vault, then enter the password.

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN
IDENTIFIED BY "Oracle_Key_Vault_endpoint_password";
```

- **For Oracle Database 12c or later:**

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN
IDENTIFIED BY "Oracle_Key_Vault_endpoint_password";
```

11. After you complete the migration, if you are using an auto-login wallet, then re-enable it by renaming the *cwallet.sso.bak* file to *cwallet.sso*.

Related Topics

- [About Endpoint Enrollment and Provisioning](#)
Endpoints are Oracle Key Vault clients that use the server to store and manage security objects, share them with trusted peers, and retrieve them.
- [okvutil download Command](#)
The `okvutil download` command downloads security objects from Oracle Key Vault to the endpoint.

Restoring Database Contents Previously Encrypted by TDE Using an Oracle Wallet

You perform the restoration process on the endpoint where you downloaded the Oracle wallet.

When an Oracle database endpoint is converted from a local Oracle wallet file to using Oracle Key Vault, you may need to restore backups that were encrypted by using a key from this local wallet file.

In this case, you must download the necessary key from Oracle Key Vault to a local wallet file to be used when you decrypt the backup during the restore process. For example, suppose that the `Finance_DB` database had recently migrated to use an [online master encryption key](#) to Oracle Key Vault after you have uploaded the premigration wallet. If a system failure forces you to restore from a database backup taken before the migration to Oracle Key Vault, then you can still restore the contents of the database by using an Oracle wallet downloaded from the Oracle virtual wallet that contains the `Finance_DB` wallet data that you had uploaded earlier.

1. Download this Oracle wallet from Oracle Key Vault by using the `okvutil download` command.
2. On the endpoint where you downloaded the Oracle wallet, edit the `sqlnet.ora` file to have the following setting:

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)
(METHOD_DATA=(DIRECTORY=wallet_file_path)))
```

Put the `ENCRYPTION_WALLET_LOCATION` setting on one line.

3. Open the downloaded wallet using the password you specified.
 - For Oracle Database 11g release 2, as a user who has been granted the `ALTER SYSTEM` system privilege:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "wallet_password";
```

- For Oracle Database 12c or later, as a user who has been granted the `SYSKM` administrative privilege:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "wallet_password";
```

Opening the wallet enables the server to read the contents of the updated `sqlnet.ora` file. At this point, the endpoint server has been restored to a state where it now can run with the original version of the wallet.

Related Topics

- [okvutil download Command](#)
The `okvutil download` command downloads security objects from Oracle Key Vault to the endpoint.

Uploading and Downloading Oracle Wallets

To store and share Oracle wallets, you must upload them to Oracle Key Vault.

- [About Uploading and Downloading Oracle Wallets](#)
You use the `okvutil` utility to upload and download Oracle wallets.
- [Uploading Oracle Wallets](#)
The `okvutil upload` command uploads wallets to Oracle Key Vault.
- [Downloading Oracle Wallets](#)
The `okvutil download` command downloads an Oracle wallet from the Oracle Key Vault server to an endpoint.
- [Guidelines for Uploading and Downloading Oracle Wallets](#)
Oracle provides guidelines for uploading and downloading wallets to and from Oracle Key Vault.

About Uploading and Downloading Oracle Wallets

You use the `okvutil` utility to upload and download Oracle wallets.

After you upload a wallet to Oracle Key Vault, you can then create a new virtual wallet in Key Vault, and add security objects to it that you want to share. You must grant endpoints access to the virtual wallet before they can download it. You can use the `okvutil upload` and `okvutil download` commands to upload and download Oracle wallets between Oracle Key Vault and its endpoints. The `okvutil` utility is packaged with the endpoint software that you install at the endpoint.

The Oracle Key Vault `okvutil` software can read an Oracle wallet at the granularity level of an individual security object, and it supports security objects up to 128 KB in size. It therefore uploads the wallet contents as individual items. During download you can recreate the original wallet with the same set of security objects, or create a new wallet with different set of security objects.

You can upload and download both password-based wallets and auto-login wallets. The wallet contents can be downloaded later into a new wallet of either type. For example, an uploaded password-protected wallet can be downloaded as an auto-login wallet, or an uploaded auto-login wallet can be downloaded as a password-protected wallet.

You can use Oracle Key Vault to construct a new virtual wallet containing security objects from previously uploaded Oracle wallets. For example, given a previously uploaded Oracle wallet containing five symmetric keys and three opaque objects, you can create a new virtual wallet consisting of only three of the original five symmetric keys and one of the three original opaque objects. This virtual wallet can be downloaded like the original wallet to provide the endpoint with access to only a subset of the keys. This process does not modify the original wallet.

Be aware that the **Extractable** attribute setting for the objects that are created in Oracle Key Vault using `okvutil upload` is inherited from the endpoint. You can download the wallet, but if the wallet's objects have the **Extractable** attribute set to `FALSE`, then those objects cannot be downloaded.

Related Topics

- [Oracle Key Vault okvutil Endpoint Utility Reference](#)
The `okvutil` utility enables you to perform tasks uploading and downloading security objects.

- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

Uploading Oracle Wallets

The `okvutil upload` command uploads wallets to Oracle Key Vault.

Uploading the contents of a TDE wallet into Oracle Key Vault is a unique feature of Oracle Key Vault: If you plan to migrate the database to use Oracle Key Vault in online master encryption key mode then you must upload the wallet content before the migration step. That allows you to delete the file-based wallet after a successful migration from the database server, which is a requirement of the PCI-DSS. The upload operation uploads everything in the Oracle wallet, including security objects and their metadata so that the wallet can be reconstructed during the download process. The Oracle wallet typically contains TDE master encryption keys, historical TDE master encryption keys, SSL or TLS certificates and their metadata (stored in Oracle Key Vault as opaque objects), wallet metadata, as well as keys that you have explicitly added. Be aware that the **Extractable** attribute setting for the objects that are created in Oracle Key Vault using `okvutil upload` is inherited from the endpoint.

1. Ensure that the server containing the Oracle wallet has been enrolled and provisioned as an Oracle Key Vault endpoint.
2. Ensure that the endpoint has access to the virtual wallet that you want to use.

The endpoint must have read, modify, and manage wallet access to the virtual wallet in Oracle Key Vault.

3. Run the `okvutil upload` command to upload the wallet.

For example:

```
# okvutil upload -l /etc/oracle/wallets -t wallet -g HRWallet
Enter wallet password (<enter> for auto-login): password
Enter Oracle Key Vault endpoint password: Key_Vault_endpoint_password
Upload succeeded
```

In this example:

- `-l` specifies the directory location of the wallet that you are uploading.
- `-t` indicates the type, in this case, an Oracle wallet.
- `-g` specifies the Oracle Key Vault virtual wallet that was configured in Step 2, so that this wallet can be part of that virtual wallet.

At this point, the upload is complete. You can now share the virtual wallet with other users and endpoints.

Related Topics

- [Managing Endpoints](#)
You can enroll, reenroll, suspend, rotate, and delete endpoints.
- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.
- [okvutil upload Command](#)
The `okvutil upload` command uploads security objects to Oracle Key Vault.
- [Oracle Key Vault okvutil Endpoint Utility Reference](#)
The `okvutil` utility enables you to perform tasks uploading and downloading security objects.

Downloading Oracle Wallets

The `okvutil download` command downloads an Oracle wallet from the Oracle Key Vault server to an endpoint.

1. Ensure that security objects in the wallet have an extraction status of `TRUE`.

If the object has **Extractable** attribute setting of `FALSE`, then you cannot download it. You can check the **Extractable** attribute setting by navigating to the Object Details page. Select the **Keys & Wallets** tab, then in the left navigation bar, select **Keys & Secrets**, then for the key listed, select the **Edit** icon. The **Extractable** attribute setting is listed in the Advanced section.

2. Ensure that the endpoint has Read access on the virtual wallet that you want to download.
3. Run the `okvutil download` command to download the wallet.

For example:

```
# okvutil download -l /etc/oracle/wallets/orcl/ -t WALLET -g HRwallet
Enter new wallet password(<enter> for auto-login): Oracle_wallet_password
Confirm new wallet password: Oracle_wallet_password
Enter Oracle Key Vault endpoint password: Key_Vault_endpoint_password
```

In this example:

- `-l` is the location of the wallet to be created.
- `-t` indicates the type, in this case, an Oracle wallet.
- `-g` specifies the Oracle Key Vault virtual wallet that was configured in Step 2.

If the wallet already exists and you did not use the `-o` parameter to overwrite the existing wallet, then the following actions take place:

- The existing wallet is renamed to a backup name of the format `ewallet.p12.current_timestamp` where the *timestamp* is number of seconds since epoch.
- The newly downloaded wallet is given the name `ewallet.p12`.

4. Close and then reopen the wallet.

Closing and reopening the wallet makes the wallet content, including TDE master encryption keys, available on a database encrypted with TDE, and loads the wallet contents into the TDE database. (Auto-login wallets are automatically opened the next time that they are accessed.)

- For Oracle Database 11g release 2:

```
ALTER SYSTEM SET ENCRYPTION WALLET CLOSE IDENTIFIED BY "Oracle_wallet_password";

ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "Oracle_wallet_password";
```

- For Oracle Database 12c or later:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY
"Oracle_wallet_password";

ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
"Oracle_wallet_password";
```

5. If you are operating in a shared server configuration such as Oracle RAC, then restart the database.

Related Topics

- [okvutil download Command](#)
The `okvutil download` command downloads security objects from Oracle Key Vault to the endpoint.
- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.
- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

Guidelines for Uploading and Downloading Oracle Wallets

Oracle provides guidelines for uploading and downloading wallets to and from Oracle Key Vault.

- If there is a change to the content of the original wallet, such as a key rotation or a rekey operation, then upload the wallet again to Oracle Key Vault so that Key Vault has the latest copy of the wallet.
- Use care if you plan to use the `okvutil upload` and `okvutil download` commands, which provide an overwrite (`-o`) option. This option overwrites data in the virtual wallet that conflicts with the data to be uploaded. Before you use the `-o` option, you should create a local backup of the wallet file.
- Do not try to upload the same physical Oracle wallet to more than one virtual wallet on the Oracle Key Vault server. If you want to share an Oracle wallet with multiple endpoints, then create an endpoint group.

Related Topics

- [Managing Endpoint Groups](#)
An endpoint group is a named group of endpoints that share a common set of wallets.

Uploading and Downloading JKS and JCEKS Keystores

The `okvutil upload` and `okvutil download` commands can upload and download JKS and JCEKS keystores.

- [About Uploading and Downloading JKS and JCEKS Keystores](#)
You use the `okvutil` utility to upload and download JKS and JCEKS keystores.
- [Uploading JKS or JCEKS Keystores](#)
The `okvutil upload` command can upload a JKS or JCEKS to the Oracle Key Vault server.
- [Downloading JKS or JCEKS Keystores](#)
The `okvutil download` command can download an uploaded JKS or JCEKS keystore.
- [Guidelines for Uploading and Downloading JKS and JCEKS Keystores](#)
Oracle provides recommendations for when you upload and download JKS and JCEKS keystores.

About Uploading and Downloading JKS and JCEKS Keystores

You use the `okvutil` utility to upload and download JKS and JCEKS keystores.

You can upload JKS and JCEKS keystores to Oracle Key Vault for long-term retention, recovery, and sharing, and when you need them, download them to an endpoint.

Similar to wallets, when you upload a JKS or JCEKS keystore, Oracle Key Vault can read each item within the keystore. It uploads the keystore contents as individual items, and it supports an item up to 128 KB in size.

Be aware that the **Extractable** attribute setting for the objects that are created in Oracle Key Vault using `okvutil upload` is inherited from the endpoint. You can download the JKS and JCEKS keystore, but if this keystore's objects have the **Extractable** attribute set to `FALSE`, then those objects cannot be downloaded.

Related Topics

- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

Uploading JKS or JCEKS Keystores

The `okvutil upload` command can upload a JKS or JCEKS to the Oracle Key Vault server.

Be aware that the **Extractable** attribute setting for the objects that are created in Oracle Key Vault using `okvutil upload` is inherited from the endpoint. You can download the JKS and JCEKS keystore, but if this keystore's objects have the **Extractable** attribute set to `FALSE`, then those objects cannot be downloaded.

1. Ensure that the server containing the Java keystore has been enrolled and provisioned as an Oracle Key Vault endpoint.
2. Ensure that access control has been configured for the endpoint.

If you are uploading the keystore to a virtual wallet, then ensure that the endpoint has the read, modify, and manage wallet access to this virtual wallet.

3. Run the `okvutil upload` command to upload the keystore.

The following examples show how to upload the JKS and JCEKS keystore to a virtual wallet.

This example shows how to upload a JKS keystore:

```
# okvutil upload -l /etc/oracle/fin_jks.jks -t JKS -g FinanceGrp
Enter source Java keystore password: Java_keystore_password
Enter Oracle Key Vault endpoint password: Key_Vault_endpoint_password
Upload succeeded
```

In this example:

- `-l` is the location of the Java keystore that is being uploaded.
- `-t` is the type of JKS or JCEKS keystore. Ensure that you upload the correct type of Java keystore when you upload or later on, when you download.
- `-g` is the virtual wallet in Oracle Key Vault where the Java keystore contents will be uploaded.

This example shows how to upload a JCEKS keystore:

```
# okvutil upload -l /etc/oracle/hr_jceks.jceks -t JCEKS -g HRGrp
Enter source Java keystore password: password
Enter Oracle Key Vault endpoint password: password
Upload succeeded
```

At this point, the upload is complete. You are now ready to share or download the Java keystore as needed.

Related Topics

- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.

Downloading JKS or JCEKS Keystores

The `okvutil download` command can download an uploaded JKS or JCEKS keystore.

1. Ensure that security objects in the JKS or JCEKS keystore have an **Extractable** attribute setting of `TRUE`.

If the object has **Extractable** attribute setting of `FALSE`, then you cannot download it. You can check the **Extractable** attribute setting by navigating to the Object Details page. Select the **Keys & Wallets** tab, then in the left navigation bar, select **Keys & Secrets**, then for the key listed, select the **Edit** icon. The **Extractable** attribute setting is listed in the Advanced section.

2. Ensure that the endpoint has the read access on the virtual wallet that you want to download.
3. As an endpoint administrator, from the command line, run the `okvutil download` command to download the Java keystore.

For example:

```
# okvutil download -l /etc/oracle/new_java_files/hr_jceks.jceks -t JCEKS -g HRGrp
Enter new Java keystore password: password
Confirm new Java keystore password: password
Enter Oracle Key Vault endpoint password: Key_Vault_endpoint_password
```

In this example:

- `-l` is the directory to which you want to download the uploaded Java keystore.
- `-t` is the type of JKS or JCEKS keystore. Ensure that you download the correct type of Java keystore.
- `-g` is the name of the object to download, which in this case is the JKS keystore `HRGrp`.

Related Topics

- [okvutil download Command](#)
The `okvutil download` command downloads security objects from Oracle Key Vault to the endpoint.
- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.
- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

Guidelines for Uploading and Downloading JKS and JCEKS Keystores

Oracle provides recommendations for when you upload and download JKS and JCEKS keystores.

- If there is a change to the content of the original JKS or JCEKS keystore, then upload the keystore again to Oracle Key Vault so that Key Vault has the latest copy of the keystore.
- Use care if you plan to use the `okvutil upload` and `okvutil download` commands, which provide an overwrite (`-o`) option. This option overwrites data in the file. Before you use the `-o` option, you should create backups of the keystore files before downloading them.
- Do not try to upload the same physical JKS or JCEKS keystore to more than one virtual wallet on the Oracle Key Vault server. If you want to share a Java keystore with multiple endpoints, then create an endpoint group.

Related Topics

- [Managing Endpoint Groups](#)
An endpoint group is a named group of endpoints that share a common set of wallets.

Using a User-Defined Key as the TDE Master Encryption Key

You can import a generated key to be used as the Transparent Data Encryption (TDE) master encryption key in Oracle Key Vault.

- [About Using a User-Defined Key as the TDE Master Encryption Key](#)
Users who have the Key Administrator role can upload a user-defined key to the virtual wallet to which they have write access.
- [Step 1: Upload the User-Defined Key](#)
Use the `okvutil upload` command to upload user-defined master encryption keys to Oracle Key Vault.
- [Step 2: Activate the User-Defined Key as a TDE Master Encryption Key](#)
After you upload the user-defined key, you are ready to activate the key as a TDE master encryption key.

About Using a User-Defined Key as the TDE Master Encryption Key

Users who have the Key Administrator role can upload a user-defined key to the virtual wallet to which they have write access.

This enables it so that it can be used as the Transparent Data Encryption (TDE) master encryption key. This feature provides key administrators with more control on creation of the master encryption key used to encrypt TDE data encryption keys.

The `type` parameter of the `okvutil upload` command includes the option `TDE_KEY_BYTES`, which enables you to upload user-defined key bytes to Oracle Key Vault to be used as the TDE master encryption key. You must then activate the key as a TDE master encryption key by running the `ADMINISTER KEY MANAGEMENT` SQL statement on the database.

Be aware that the **Extractable** attribute setting for the objects that are created in Oracle Key Vault using `okvutil upload` is inherited from the endpoint. You can download the TDE master encryption key, but if this the key's objects have the **Extractable** attribute set to `FALSE`, then those objects cannot be downloaded.

Related Topics

- [Oracle Database Advanced Security Guide](#)
- [Configuring the Extractable Attribute Value of Existing Symmetric or Private Keys](#)
You can configure the extractable attribute value of existing symmetric or private keys.

Step 1: Upload the User-Defined Key

Use the `okvutil upload` command to upload user-defined master encryption keys to Oracle Key Vault.

The raw bytes data of the user-defined key is stored in a text file and uploaded to Oracle Key Vault. The raw bytes data uploaded to Oracle Key Vault forms part of the TDE Master Encryption Key and the TDE Master Encryption Key Identifier. Additional metadata is added to the raw bytes data to enable the database to identify and activate the data as the TDE Master Encryption Key and the TDE Master Encryption Key. In the text file, the raw bytes data that forms the TDE Master Key is prefixed by TDE Master Encryption Key Identifier. The raw bytes data that forms the TDE Master Key Identifier is prefixed by TDE Master Key Identifier. TDE Master Key Identifier represents the master encryption key in the database. Once the key is activated, you should see the user-defined raw bytes that form the TDE Master Key Identifier as the subset of the `KEY_ID` column of the `V$ENCRYPTION_KEYS` view. In Oracle Key Vault, the TDE Master Key and TDE Master Key Identifier values are stored as managed KMIP objects with a symmetric key as a KMIP object type.

Be aware that the **Extractable** attribute setting for the objects that are created in Oracle Key Vault using `okvutil upload` is inherited from the endpoint.

1. Create a text file containing the raw bytes data of the user-defined key.

Use the following format:

```
TDE Master Encryption Key Identifier:
contiguous_TDE_Master_Encryption_Key_Identifier_bytes_encoded_in_32_hex_characters_(16_bytes_long)
TDE Master Encryption Key:
contiguous_TDE_Master_Encryption_Key_bytes_encoded_in_64_hex_characters_(32_bytes_long)
```

For example:

```
TDE Master Encryption Key Identifier: 1F1E1D1C1B1A10191817161514131210
TDE Master Encryption Key:
97cefdb589ef06d56b8d536d3d38b7221a76dffbf3f28a60d0965b9ae1a785b
```

2. Save the file as `tde_key_bytes.txt`.
3. Use the `okvutil upload` command to upload `tde_key_bytes.txt`.

The format of the `okvutil upload` command is:

```
okvutil upload [--overwrite] --location location --type type [--group group] [--description description] [--verbose verbosity_level]
```

Example:

```
$OKV_HOME/bin/okvutil upload -l /home/oracle/tde_key_bytes.txt -t
TDE_KEY_BYTES -g "FIN_DATABASE_VIRTUAL_WALLET" -d "This key was created
for Financial database use on 1st Mar 2019"
```

In this example:

- `-l` specifies the path to the `tde_key_bytes.txt` file.
- `-t` specifies the type of the object to upload. To upload a user-defined key, specify the type as `TDE_KEY_BYTES`.
- `-g` specifies the name of an Oracle Key Vault virtual wallet to which the key is added.
- `-d` specifies a description for the key.

When `-t` is `TDE_KEY_BYTES`, the description specified for `-d` is displayed as the tag in the `V$ENCRYPTION_KEYS` dynamic view.

4. Specify the required parameters and then press **Enter**.
5. Enter the Oracle Key Vault endpoint password and press **Enter**.

The message `Upload succeeded` is displayed.

```
$OKV_HOME/bin/okvutil upload -l /home/oracle/tde_key_bytes.txt -t
TDE_KEY_BYTES -g "FIN_DATABASE_VIRTUAL_WALLET" -d "This key was created
for Financial database use on 1st Mar 2019"
Enter Oracle Key Vault endpoint password:
Upload succeeded
```

6. Carefully delete the `tde_key_bytes.txt` file.

The raw bytes data of the user-defined key is uploaded. The next step is to activate the user-defined key as a TDE master encryption key.

Related Topics

- [Step 2: Activate the User-Defined Key as a TDE Master Encryption Key](#)
After you upload the user-defined key, you are ready to activate the key as a TDE master encryption key.

Step 2: Activate the User-Defined Key as a TDE Master Encryption Key

After you upload the user-defined key, you are ready to activate the key as a TDE master encryption key.

The raw bytes data uploaded to Oracle Key Vault for the `TDE Master Key Identifier` is displayed as the `NAME` attribute of the KMIP object that is created as the corresponding TDE master encryption key in Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user with the Key Administrator role or as a user with access to the virtual wallet.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar.
3. In the Object Details page, expand **Advanced**.

The required Key ID is displayed in the Names area. The Key ID is prefixed with `ORACLE.TDE.HSM.MK.06`.

For example:

```
ORACLE.TDE.HSM.MK.061F1E1D1C1B1A10191817161514131210
```

The TDE master encryption key identifiers contain the user defined raw bytes data prefixed by additional metadata.

4. Copy and store the key ID displayed after the prefix `ORACLE.TDE.HSM.MK`

For example:

```
061F1E1D1C1B1A10191817161514131210
```

5. Connect to the Oracle database as a user who has privileges to run the `ADMINISTER KEY MANAGEMENT SQL` statement.

For example:

```
sqlplus C##crypto_admin/ as syskm
```

6. Activate the key as a TDE master encryption key using the `ADMINISTER KEY MANAGEMENT` command.

```
ADMINISTER KEY MANAGEMENT USE KEY  
'061F1E1D1C1B1A10191817161514131210' IDENTIFIED BY <OKV-pwd> | EXTERNAL  
STORE;password;
```

You can query the `TAG` column of the `V$ENCRYPTION_KEYS` view for the identifier of the newly created key.

12

Managing Oracle Key Vault Endpoints

Oracle Key Vault endpoints are computer systems like database or application servers, where keys and credentials are used to access data.

- [Overview of Managing Endpoints](#)
Learn how to manage endpoints in standalone and multi-master cluster configurations, and how a multi-master cluster works with endpoints.
- [Managing Endpoints](#)
You can enroll, reenroll, suspend, rotate, and delete endpoints.
- [Managing Endpoint Details](#)
Endpoint details refers to endpoint name, type, description, platform, and email, and adding the endpoint to a group, or upgrading the endpoint software.
- [Managing Global and Per-Endpoint Configuration Parameters and Settings](#)
Oracle Key Vault provides global and per-endpoint configuration parameters and settings that you can set in the Oracle Key Vault management console.
- [Default Wallets and Endpoints](#)
You can use a default wallet, which is a type of virtual wallet, with an endpoint.
- [Managing Endpoint Access to a Virtual Wallet](#)
You can grant an endpoint access to a virtual wallet, and revoke or modify access when it is no longer necessary.
- [Managing Endpoint Groups](#)
An endpoint group is a named group of endpoints that share a common set of wallets.

Overview of Managing Endpoints

Learn how to manage endpoints in standalone and multi-master cluster configurations, and how a multi-master cluster works with endpoints.

- [About Managing Endpoints](#)
You must register and enroll an endpoint to communicate with an Oracle Key Vault server.
- [How a Multi-Master Cluster Affects Endpoints](#)
You should be aware of how a multi-master cluster affects endpoints, both in the way an endpoint connects to it and with restrictions.

About Managing Endpoints

You must register and enroll an endpoint to communicate with an Oracle Key Vault server.

Afterward, keys in the endpoint can be uploaded to Oracle Key Vault and be shared with other endpoints and then downloaded from these endpoints so that users can access their data. Only a user with the System Administrator role or the Create Endpoint privilege can add an endpoint to Oracle Key Vault. After the endpoint is added, the endpoint administrator can enroll the endpoint by downloading and installing the endpoint software at the endpoint. The endpoint

can then use the utilities packaged with the endpoint software to upload and download security objects to and from Oracle Key Vault.

All users can create virtual wallets but only a user with the Key Administrator role can grant endpoints access to security objects contained in virtual wallets. A Key Administrator user can grant access on any wallet to an endpoint. A user with the Key Administrator role or Create Endpoint Group privilege can also create endpoint groups to enable shared access to virtual wallets. Any user (including users who created the endpoint) who has the manage wallet permission on a wallet can grant access to that wallet to an endpoint. When you grant an endpoint group access to a virtual wallet, all the member endpoints will have access to the virtual wallet. For example, you can grant all the nodes in an Oracle Real Application Clusters (Oracle RAC) database access to a virtual wallet by putting them in an endpoint group. This saves you the step of granting each node access to the virtual wallet. As an added layer of security, the Key Administrator user can enable or disable the extraction of symmetric keys from Oracle Key Vault.

If you have a large deployment, Oracle recommends that you install at least four Oracle Key Vault servers, and when you enroll the endpoints, balance them across these four servers to ensure high availability. For example, if a data center has 1000 database endpoints to register, and you have four Oracle Key Vault servers to accommodate them, then enroll 250 endpoints across each of the four servers.

If you have configured cluster subgroups, then you can set the cluster subgroup settings for the endpoints appropriately. An endpoint preferentially connects to the cluster nodes from its cluster subgroup. In a geographically distributed deployment, you may place cluster nodes deployed in a region into a cluster subgroup and assign this cluster subgroup to the endpoints from the same region. This way endpoints select the cluster nodes in the local region first when establishing new connections.

When you name an endpoint, you cannot use an Oracle Key Vault user name as the endpoint name.

Ensure that the system clocks of the endpoint host and the Oracle Key Vault server are synchronized. For Oracle Key Vault server, setting up NTP is required. Drift between endpoint time and Oracle Key Vault server time may result in issues during endpoint enrollment. When Oracle Key Vault issues the endpoint certificate, the endpoint certificate time is set on the Oracle Key Vault server. Endpoint enrollment becomes an issue if the Oracle Key Vault server time is ahead of the endpoint certificate enrollment time.

**Note:**

Ensure that the system time on the endpoint host and the Oracle Key Vault server is synchronized for a successful endpoint enrollment.

When a user with Create Endpoint or Endpoint Group privilege creates an endpoint or endpoint group, Oracle Key Vault indirectly grants the Manage Endpoint or Manage Endpoint Group privileges to the user.

The administrative roles and privileges as they pertain to endpoints are as follows:

- **Endpoint creation:** A user with either the System Administrator role or the Create Endpoint privilege can create endpoints in Oracle Key Vault. A user with the Create Endpoint privilege is automatically granted the Manage Endpoint privilege on any endpoints that they create.
- **Endpoint management:** A user with the System Administrator role can manage any endpoints. A user with the Manage Endpoint privilege can manage only his or her own

endpoints. This includes endpoints that the user was explicitly granted the Manage Endpoint privilege on, or endpoints that the user created and continues to have the Manage Endpoint privilege on. This management includes the following duties:

- Managing the endpoint metadata such as the name, type, platform, description, and email notifications
- Managing the endpoint life cycle, which consists of enrolling, suspending, re-enrolling, rotating, and deleting endpoints
- **Endpoint group creation:** A user with the Key Administrator role or Create Endpoint Group privilege can create endpoint groups in Oracle Key Vault. A user with the Create Endpoint Group privilege is automatically granted the Manage Endpoint Group privilege on any endpoint groups that they create.
- **Endpoint group management:** A user with the Key Administrator role can manage any endpoint groups. A user with the Manage Endpoint Group privilege can manage only his or her own endpoint groups. The endpoint groups that a user can manage include those that the user was explicitly granted the manage endpoint group privilege on, or those that the user created and continues to have the manage endpoint group privilege on. This management includes the following duties:
 - Managing the endpoint group life cycle, which consists of creating, modifying, and deleting endpoint groups
 - Managing the life cycle of security objects, which consists of creating, modifying and deleting security objects

Related Topics

- [Endpoint Privileges](#)
Oracle Key Vault provides privileges for creating and managing endpoints and endpoint groups.
- [Monitoring Certificates Expiry](#)
Proactively set alerts and monitor the expiry dates of the Oracle Key Vault certificates and rotate them before they expire.

How a Multi-Master Cluster Affects Endpoints

You should be aware of how a multi-master cluster affects endpoints, both in the way an endpoint connects to it and with restrictions.

In a multi-master configuration, when an endpoint attempts to make a connection to Oracle Key Vault, it performs the following actions:

- First, it obtains the list of server IPs from its configuration file (`okvclient.ora`).
- Next, it picks one at random, preferentially from those in the same cluster subgroup as the endpoint.

Be aware of the following restrictions with how endpoints work in multi-master clusters:

- An endpoint can only be enrolled from the same node where it was most recently created or reenrolled.
- You can assign a default wallet to an endpoint if one or both of them (wallet and endpoint) is in the `PENDING` state, but not if the assignment is attempted from a non-creator node. After both the endpoint and wallet are in the `ACTIVE` state, this restriction ends.

Managing Endpoints

You can enroll, reenroll, suspend, rotate, and delete endpoints.

- [Types of Endpoint Enrollment](#)
The first step in enrolling an endpoint is to add the endpoint to Oracle Key Vault.
- [Endpoint Enrollment in a Multi-Master Cluster](#)
Endpoints of a cluster are the client systems of the multi-master cluster.
- [Adding an Endpoint as an Oracle Key Vault System Administrator or Create Endpoint User](#)
A user who has been granted the System Administrator role or the Create Endpoint privilege can add an endpoint by using the **Endpoints** tab.
- [Adding Endpoints Using Self-Enrollment](#)
The self-enrollment process immediately sends the endpoint to the **Enrolled** status without the intermediate **Registered** status.
- [Deleting, Suspending, Reenrolling, or Rotating Endpoints](#)
When endpoints no longer use Oracle Key Vault to store security objects, you can delete them. You can also suspend, and later resume them when they are needed. You can also re-enroll or rotate endpoints when necessary.

Types of Endpoint Enrollment

The first step in enrolling an endpoint is to add the endpoint to Oracle Key Vault.

There are two methods for adding, also known as registering, an endpoint:

- **Initiated by an administrator**
An Oracle Key Vault user who has the System Administrator role initiates the enrollment from the Oracle Key Vault side by adding the endpoint to Oracle Key Vault. When the endpoint is added, a one-time enrollment token is generated. This token can be communicated to the endpoint administrator in two ways:
 - Directly from Oracle Key Vault by email. To use email notification you must configure SMTP in email settings.
 - Out-of-band method, such as email or telephone.

The endpoint administrator uses the enrollment token to download the endpoint software and complete the enrollment process. In a multi-master cluster, the same node that is used to add the endpoint must be used to enroll the endpoint.

After the enrollment token is used to enroll an endpoint, it cannot be used again for another enrollment. If you are reenrolling an endpoint, then the reenrollment process will generate a new one-time enrollment token for this purpose.

- **Self-enrolled**
Endpoints may enroll themselves during specific times without human administrative intervention. Endpoint self-enrollment is useful when the endpoints do not share [security objects](#), and use Oracle Key Vault primarily to store and restore their own security objects. Another use for endpoint self-enrollment is testing.

A self-enrolled endpoint is created with a generic endpoint name in this format: `ENDPT_001`. In a cluster, a self-enrolled endpoint is created with a generic endpoint name in this format: `ENDPT_XX_001`, where `XX` is a 2-digit node identifier or node number. Initially, a self-enrolled endpoint has access only to the security objects that it uploads or creates. It does not have

access to any virtual wallets. You can later grant the endpoint access to virtual wallets after verifying its identity.

Endpoint self-enrollment is disabled by default, and must be enabled by a user with the System Administrator role. Oracle recommends that you enable self-enrollment for short periods, when you expect endpoints to self enroll, and then disable it when the self-enrollment period ends.

 **Note:**

The self-enrollment process for endpoints is deprecated in Oracle Key Vault 21.5. You can use the RESTful services utility for endpoint enrollment instead.

Related Topics

- [Configuring Email Notification](#)
You can use email notifications to directly notify administrators of Key Vault status changes without logging into the Oracle Key Vault management console.

Endpoint Enrollment in a Multi-Master Cluster

Endpoints of a cluster are the client systems of the multi-master cluster.

Endpoint enrollment is divided into two steps. First you create the endpoint and then you enroll it.

The Oracle Key Vault server that becomes the controller node can have endpoints already enrolled, especially if it was upgraded from a previous release. These existing endpoints initialize, or seed, the cluster. During induction, information about the endpoints that were enrolled in the cluster is replicated to the newly added node. Oracle Key Vault also removes information about the endpoints that were previously enrolled in all candidate nodes added to the cluster.

Endpoints can only be enrolled on a read/write node.

After you enroll the endpoint, the new endpoint will have a cluster-wide presence. You can add endpoints of the Oracle Key Vault multi-master cluster to any read/write node.

 **Note:**

An endpoint must be enrolled on the same node where it was most recently added or reenrolled.

New endpoints added concurrently to the multi-master cluster on different nodes could have name conflicts. Oracle Key Vault automatically resolves the endpoint name conflicts, and then displays the conflicts in a Conflicts Resolution page. From here, a system administrator can choose to rename them.

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

Adding an Endpoint as an Oracle Key Vault System Administrator or Create Endpoint User

A user who has been granted the System Administrator role or the Create Endpoint privilege can add an endpoint by using the **Endpoints** tab.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Create Endpoint privilege.
2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.

The Endpoints page appears listing all the Oracle Key Vault endpoints.

<input type="checkbox"/>	Endpoint Name	Status	Enrollment Token	Type	Platform	Default Wallet	IP Address	Description
<input type="checkbox"/>	APPABC_DR_OGG	ENROLLED	-	Oracle (non-database)	Linux	GoldenGate_Replication_App_Wallet		Disaster Recovery replication site for ABC application
<input type="checkbox"/>	APPABC_PRIMARY_OGG	ENROLLED	-	Oracle (non-database)	Linux	GoldenGate_Replication_App_Wallet		Primary replication site for ABC application.
<input type="checkbox"/>	XYZAPP_WEBAPP_MYSQL	ENROLLED	-	MySQL Database	Linux	XYZ_MYSQL_WALLET		MySQL database managing the XYZ web application data
<input type="checkbox"/>	LONDON_ENCRYPTED_FILE_SYSTEM	REGISTERED	w85E9CXrFYV6EKwj	Oracle ACFS	Linux	ASM_Cluster_File_System_Wallet		System managing the DB executables for the London server

1 - 4 of 4

The Endpoints page displays the list of registered and enrolled endpoints with the following endpoint details: name, type, description, platform, status, enrollment token, alert, endpoint certificate expiration, common name of certificate issuer, and last active time. This listing of endpoints depends on who is logged in. The endpoint information listing depends on who is logged-in, some details such as the enrollment token, and the buttons listed above the table on **Endpoints** page may be visible depending on the user's role or privilege status. For example, a user with the System Administrator role can see buttons related to all operations such as Add, Suspend, and Resume, while one with only the Create Endpoint privilege can see only the **Add** button. The endpoint status can be either **Registered** or **Enrolled**:

- **Registered Status:** The endpoint has been added and the one-time enrollment token has been generated. This token will be displayed in the corresponding Enrollment Token column.
- **Enrolled Status:** The one-time enrollment token has been used to download the endpoint software. The Enrollment Token column displays a dash (-) to indicate that the enrollment token has been used. If you do not have the System Administrator role (that is, you have the Manage Endpoint privilege), then you can only view enrollment tokens for those endpoints that you can manage.
- **Created By:** The user who created the endpoint. If the user no longer exists, or if the endpoint was created in a version before this information was stored, then this field will show ANONYMOUS.
- **Creator Node:** The node on which the endpoint was created. This is specific to multi-master cluster environments.

- **Name Status:** The state of the endpoint. The state will be either `ACTIVE` or `PENDING`. This is specific to multi-master cluster environments.
 - **Cluster Subgroup:** The subgroup on which the endpoint was created. This is specific to multi-master cluster environments.
3. On the Endpoints page, click **Add**.

The Register Endpoint page appears. The **Make Unique** check box and **Cluster Subgroup** drop-down only appears in multi-master cluster mode.

4. In the **Endpoint Name** field, enter a name for the endpoint.

See [Naming Guidelines for Objects](#).

5. If you are using a multi-master cluster, then choose whether to select the **Make Unique** checkbox.

Make Unique helps to control naming conflicts with names across the multi-master cluster environment. Endpoints that were created before an Oracle Key Vault conversion to a cluster node are not affected by naming conflicts.

- If you select **Make Unique**, then the endpoint will be active immediately and users can use this endpoint.
- If you do not select **Make Unique**, then the endpoint will be created in the `PENDING` state. Oracle Key Vault will then begin a name resolution operation and may rename the endpoint to a name that is unique across the cluster. If there is a naming collision, then the collision will be reported on the Conflicts page on any node in the cluster. The endpoint will then be renamed to a unique name. You will need to go to a read/write node of the cluster and either accept the renamed endpoint or change the endpoint name. If you change the endpoint name, then this will restart the name resolution operation and the endpoint will return to a `PENDING` state. An endpoint in the `PENDING` state cannot be used to perform most operations.

6. From the **Type** drop-down list, select the type of endpoint.

Supported types are **Oracle Database**, **Oracle Database Cloud Service**, **Oracle (non-database)**, **Oracle ACFS**, **MySQL Database**, **SSH Server**, and **Other**. An example of **Other** is a third-party KMIP endpoint. If you are using Oracle Advanced Security Transparent Data Encryption (TDE) and want to use Oracle Key Vault to manage a TDE master encryption key or wallet, then you must set **Type** to **Oracle Database**.

7. Complete the following endpoint information:
 - **Platform:** Supported platform choices are **Linux**, **Linux for Arm (aarch64)**, **IBM: Linux on System z**, **Solaris SPARC**, **Solaris x64**, **AIX**, **HP-UX**, and **Windows**.
 - **Description:** Optionally, enter a useful identifying description such as the host name, IP address, function, or location of the endpoint.
 - **Administrator Email:** Optionally, enter the email address of the endpoint administrator to have the enrollment token and other endpoint-related alerts sent directly from Oracle Key Vault. You must have SMTP configured to use the email notification feature.
 - **Cluster Subgroup:** For a multi-master cluster environment, select a subgroup for the endpoint. If you select **No Cluster Subgroup**, then the endpoint will not be a part of any cluster subgroup. If you select the option suffixed with **(from Creator Node)**, the endpoint will be a part of the cluster subgroup to which its creator node belongs, even if the creator node's cluster subgroup changes. All other options assign an endpoint to an existing cluster subgroup, to which it will belong regardless of its creator node's cluster subgroup.
 - **SSH Server Hostname:** For type **SSH Server**, enter the hostname or the IP address of the SSH server host on which the Oracle Key Vault Secure Shell (SSH) endpoint will be deployed.
8. Click **Register**.

The Endpoints page appears listing the new endpoint with a status of **Registered**. The Enrollment Token column displays the one-time enrollment token.

<input type="checkbox"/>	Endpoint Name	Status	Enrollment Token	Type	Platform	Default Wallet	IP Address	Description
<input type="checkbox"/>	APPABC_DR_OGG	ENROLLED	-	Oracle (non-database)	Linux	GoldenGate_Replication_App_Wallet		Disaster Recovery replication site for ABC application
<input type="checkbox"/>	APPABC_PRIMARY_OGG	ENROLLED	-	Oracle (non-database)	Linux	GoldenGate_Replication_App_Wallet		Primary replication site for ABC application.
<input type="checkbox"/>	XYZAPP_WEBAPP_MYSQL	ENROLLED	-	MySQL Database	Linux	XYZ_MYSQL_WALLET		MySQL database managing the XYZ web application data
<input type="checkbox"/>	LONDON_ENCRYPTED_FILE_SYSTEM	REGISTERED	w85E9CXrFYV6EKwj	Oracle ACFS	Linux	ASM_Cluster_File_System_Wallet		System managing the DB executables for the London server

1 - 4 of 4

9. Click the endpoint name to see details for the endpoint.
The Endpoint Details page appears.

Endpoint Details

Cancel
Delete
Suspend
Save
Rotate

Endpoint Name Make Unique

Type

Description

Platform

Administrator Email

Status ENROLLED

Creation Time 07-DEC-2024 21:29:25

Certificate Expiration Time 07-DEC-2025 21:29:43

Self Enrolled NO

Cluster Subgroup

Strict IP Check Enforced Global Setting: True

Common Name of Certificate Issuer OKV_CA_2024_12_07_21:11:50 Download (PEM format) Download (wallet)

Last Active Time 07-DEC-2024 21:30:47

?

The **Send Enrollment Token** button on the Endpoint Details page *only* appears for an endpoint whose **Status** is **Registered**. The users with the System Administrator role or the Manage Endpoint privilege for the endpoint can view an **Endpoint's enrollment token**, or **Send Enrollment Token** button.

There are two ways to send the one-time enrollment token to the endpoint administrator:

- If you did configure SMTP and entered the email address, you can have Oracle Key Vault send the enrollment token directly to the endpoint administrator, shown in the next step, where you click the **Send Enrollment Token** button.
- If you did not configure SMTP or enter the email address, then you must use an out-of-band method to send the enrollment token to the endpoint administrator.

The endpoint must be enrolled and the endpoint jar file must be downloaded from the node on which the endpoint was most recently created or reenrolled.

10. Click **Send Enrollment Token**.

At this stage, the endpoint's administrator can complete the enrollment process for the endpoint. When the enrollment token is used to download and install the endpoint software on the endpoint side, the endpoint status changes from **Registered** to **Enrolled**.

Related Topics

- [Configuring Email Notification](#)
You can use email notifications to directly notify administrators of Key Vault status changes without logging into the Oracle Key Vault management console.
- [Step 1: Enroll the Endpoint and Download the Software](#)
You must have the endpoint's enrollment token before you can download the endpoint software `okvclient.jar`.

Adding Endpoints Using Self-Enrollment

The self-enrollment process immediately sends the endpoint to the **Enrolled** status without the intermediate **Registered** status.

- [About Adding Endpoints Using Self-Enrollment](#)
Oracle Key Vault associates a self-enrolled attribute with all endpoints that are enrolled through endpoint self-enrollment.
- [Adding an Endpoint Using Self-Enrollment](#)
You can configure the self-enrollment process for endpoints from the Oracle Key Vault management console.

About Adding Endpoints Using Self-Enrollment

Oracle Key Vault associates a self-enrolled attribute with all endpoints that are enrolled through endpoint self-enrollment.

Self-enrolled endpoints go directly to **Enrolled** status without the intermediate **Registered** status when a user downloads the endpoint software. You can recognize self-enrolled endpoints by their system generated names in the format `ENDPT_001`. In a multi-master cluster, system generated endpoint names are in the format `ENDPT_node_id_sequential_number`, where `node_id` is a value such as `01` or `02`. For example, `ENDPT_01_001` can be the generated name of an endpoint.

Endpoint self-enrollment is disabled by default and must be enabled by a user who has the System Administrator role.

A best practice is to enable endpoint self-enrollment for limited periods when you expect endpoints to enroll. After the expected endpoints have been enrolled, you should disable endpoint self-enrollment.



Note:

The self-enrollment process for endpoints is deprecated in Oracle Key Vault 21.5. You can use the RESTful services utility for endpoint enrollment instead.

Adding an Endpoint Using Self-Enrollment

You can configure the self-enrollment process for endpoints from the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Endpoints** tab, and then **Settings** from the left navigation bar.
The Endpoint Settings page appears.

Endpoint Settings
Save

Allow Endpoint Self Enrollment

Global Endpoint Configuration Parameters
Save Defaults
Save

Endpoint Certificate Validity (in days)

PKCS11 In-Memory Cache Timeout (in minutes)

PKCS11 Persistent Cache Timeout (in minutes)

PKCS11 Persistent Cache Refresh Window (in minutes)

PKCS11 Configuration Parameter Refresh Interval (in minutes)

Server Poll Timeout (in milliseconds)

PKCS11 Trace Directory Path

Expire PKCS11 Persistent Cache on Database Shutdown ⓘ

3. Check the box to the right of **Allow Endpoint Self-Enrollment**.
4. Click **Save**.

Related Topics

- [Step 1: Enroll the Endpoint and Download the Software](#)
You must have the endpoint's enrollment token before you can download the endpoint software `okvclient.jar`.

Deleting, Suspending, Reenrolling, or Rotating Endpoints

When endpoints no longer use Oracle Key Vault to store security objects, you can delete them. You can also suspend, and later resume them when they are needed. You can also re-enroll or rotate endpoints when necessary.

- [About Deleting Endpoints](#)
Deleting an endpoint removes it permanently from Oracle Key Vault.
- [Deleting One or More Endpoints](#)
The Endpoints page enables you to delete a group of endpoints from Oracle Key Vault at one time.
- [Deleting One Endpoint \(Alternative Method\)](#)
The Endpoint Details page provides a consolidated view for the selected endpoint including a mechanism to delete the endpoint from Oracle Key Vault.
- [Suspending an Endpoint](#)
You can suspend an endpoint temporarily for security reasons, and then resume the endpoint once the threat has passed. You can choose to suspend unused endpoints during the CA certificate rotation process to allow the CA certificate rotation process to complete.
- [Reenrolling an Endpoint](#)
When you reenroll an endpoint, the enrollment process automatically upgrades the endpoint software and also generates new endpoint certificates.

- [Rotating Endpoint Certificates](#)
Rotating an endpoint's certificate extends its certificate validity without incurring downtime for the endpoint.

About Deleting Endpoints

Deleting an endpoint removes it permanently from Oracle Key Vault.

However, [security objects](#) that were previously created or uploaded by that endpoint will remain in Oracle Key Vault. Similarly, security objects that are associated with that endpoint also remain. To permanently delete or reassign these security objects, you must be a user with the Key Administrator role or authorized to merge these objects by managing wallet privileges. The endpoint software previously downloaded at the endpoint also remains on the endpoint until the endpoint administrator removes it.

You cannot delete an endpoint that is in the `PENDING` state unless you are the user who created it. You must delete it on the node on which it was created.

Deleting One or More Endpoints

The Endpoints page enables you to delete a group of endpoints from Oracle Key Vault at one time.

You can also delete a single endpoint from this page.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Manage Endpoint privilege on that endpoint.

A user who has the Manage Endpoint privilege can only delete endpoints on which this user has been granted the Manage Endpoint privilege. To see which endpoints that a user can manage, select the **Users** tab, then select **Manage Users**. Check the User Details page for the user in question, and scroll down to the Access to Endpoints area.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.

The Endpoints page lists all the endpoints currently registered or enrolled.

3. In the Endpoints page, select the check boxes to the left of the endpoints that you want to delete.
4. Click **Delete**.
5. In the confirmation window, click **OK**.

Note:

To uninstall the endpoint software for the deleted endpoint, remove the endpoint software installation directory or the `OKV_HOME`. You should also delete the links from `$ORACLE_HOME/okv/$ORACLE_SID` and `$ORACLE_BASE/okv/$ORACLE_SID`, if they exist. If there is no other database using Oracle Key Vault on this machine, then consider removing the `liborapkcs.so` also from `/opt/oracle/extapi/64/hsm/oracle/1.0.0` directory on Linux x86-64, Linux for Arm (aarch64), IBM: Linux on System z, Solaris, AIX, and HP-UX (IA) installations and from `C:\oracle\extapi\64\hsm\oracle\1.0.0` directory on Windows installations.

Related Topics

- Performing Actions and Searches

Deleting One Endpoint (Alternative Method)

The Endpoint Details page provides a consolidated view for the selected endpoint including a mechanism to delete the endpoint from Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Manage Endpoint privilege on that endpoint.

A user who has the Manage Endpoint privilege can only delete endpoints on which this user has been granted the Manage Endpoint privilege. To see which endpoints that a user can manage, select the **Users** tab, then select **Manage Users**. Check the User Details page for the user in question, and scroll down to the Access to Endpoints area.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.

The Endpoints page appears listing all the Oracle Key Vault endpoints.

3. Click the endpoint name that you want to delete.

The Endpoint Details page appears.

4. Click **Delete**.

5. In the confirmation window, click **OK**.

Related Topics

- Performing Actions and Searches

Suspending an Endpoint

You can suspend an endpoint temporarily for security reasons, and then resume the endpoint once the threat has passed. You can choose to suspend unused endpoints during the CA certificate rotation process to allow the CA certificate rotation process to complete.

When you suspend an endpoint, its status will change from **Enrolled** to **Suspended**. You cannot suspend an endpoint that is in the `PENDING` state unless you are the user who created it.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Manage Endpoint privilege.

A user with the Manage Endpoint privilege can only suspend endpoints that they can manage.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.

The Endpoints page appears listing all the Oracle Key Vault endpoints.

3. In the Endpoints page, select the check boxes to the left of the endpoints that you want to suspend.

4. Click **Suspend**.

5. In the confirmation window, click **OK**.

When you suspend an endpoint, its **Status** on the **Endpoints** page will be **Suspended**.

6. To enable the endpoint, perform Steps 1-3.

From the Endpoint Details pane click **Enable**. The endpoint **Status** on the **Endpoints** page will now read **Enrolled**.

The following rules apply to suspending an endpoint in a multi-master cluster:

- For regular endpoints, the endpoint will continue to operate until all suspend operation requests have reached all nodes in the cluster.
- You can suspend the endpoint on any node.
- For cloud-based endpoints, the endpoint will continue to operate until the suspend operation has reached all nodes from where the reverse SSH tunnel is established.
- You can potentially suspend the endpoint on any node from the cloud-based endpoint from where the reverse SSH tunnel is established.

Related Topics

- Performing Actions and Searches

Reenrolling an Endpoint

When you reenroll an endpoint, the enrollment process automatically upgrades the endpoint software and also generates new endpoint certificates.

You must also reenroll an endpoint to accommodate changes such as pairing a primary Oracle Key Vault server with a new secondary server in a primary-standby configuration. The action of reenrolling an endpoint will immediately disallow any connections from the endpoint's old deployment. If you are reenrolling an endpoint, Oracle recommends that you immediately download `okvclient.jar` and deploy it in a directory that is separate from the existing deployment. When you deploy the software, use the `-o` option to overwrite the symbolic link pointing to the old `okvclient.ora`. You cannot reenroll an endpoint that is in the `PENDING` state unless you are the user who created the endpoint.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Manage Endpoint privilege.

A user who has the Manage Endpoint privilege can only reenroll endpoints that he or she created.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.

The Endpoints page appears listing all the Oracle Key Vault endpoints.

3. In the Endpoints page, check the boxes to the left of the endpoints that you want to reenroll.

4. Click **Reenroll**.

A new enrollment token will be generated for each reenrolled endpoint and appear in the corresponding Enrollment Token column. You can use this one-time token to reenroll the endpoint. You must download the endpoint jar file from the same node on which the endpoint was reenrolled.

Note:

After you deploy the `okvclient.jar` file, the message **The endpoint software for Oracle Key Vault installed successfully**

should appear. If, instead, the message **The endpoint software for Oracle Key Vault upgraded successfully**

appears, then the re-enrollment was performed in the old deployment directory, and as a result, the endpoint software was upgraded but not successfully re-enrolled.

You can overwrite the symbolic link reference that points to `okvclient.ora` in the new directory by using the `okvclient.jar` option `-o`.

A new enrollment token will be generated for each reenrolled endpoint and appear in the corresponding Enrollment Token column. You can use this one-time token to reenroll the endpoint. You must download the endpoint jar file from the same node on which the endpoint was reenrolled.

Related Topics

- [Step 1: Enroll the Endpoint and Download the Software](#)
You must have the endpoint's enrollment token before you can download the endpoint software `okvclient.jar`.

Rotating Endpoint Certificates

Rotating an endpoint's certificate extends its certificate validity without incurring downtime for the endpoint.

Endpoints communicate with Oracle Key Vault using TLS certificates. When an endpoint's certificate is close to expiration, you can rotate the endpoint so that it is issued with a new certificate without incurring downtime.

Note:

You can determine the remaining time to expiration of the endpoint certificate from the Oracle Key Vault Management console, by navigating to the **Endpoints** page and checking the **Endpoint Certificate Expiration** field.

See Also:

[Finding the Expiration Date of the CA Certificate](#)

You can also monitor the expiration of an endpoint certificate by configuring the Endpoint Certificate Expiration alert so as to receive a reminder when an endpoint certificate is due to expire.

See Also:

[Configuring Oracle Key Vault Alerts](#)

If an endpoint is not rotated before its certificate expires, it will experience downtime. You must re-enroll the endpoint when that happens.

You can rotate a single endpoint, or multiple endpoints at once. Rotating an endpoint generates a new certificate on Oracle Key Vault. The endpoint must then reach out to Oracle Key Vault to receive this new certificate, and subsequently acknowledge receipt of the certificate back to Oracle Key Vault to fully complete the certificate rotation.

 **Note:**

In Oracle Key Vault release 21.9, after a certificate rotation, password-protected endpoints may fail to connect to the Oracle Key Vault deployment, with `Server Connect Failed` errors. A fix for this issue is available as a patch for bug **37006283**. Apply this patch to the Oracle Key Vault deployment before rotating the endpoint certificates.

To rotate the endpoint certificates, perform the following steps.

1. Log into the Oracle Key Vault management console as a user with the System Administrator role, or the Manage Endpoint privilege on the given endpoint(s).
2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.
The Endpoints page appears listing all the Oracle Key Vault endpoints.
3. In the Endpoints page, check the boxes to the left of the endpoints that you want to rotate.
4. Click **Rotate**.

This process can take several minutes to complete, depending on the number of endpoints that are being rotated. After a few minutes, the message “One or more endpoints rotated successfully” appears. For each rotated endpoint, the Common Name of Certificate Issuer field on the Endpoints page is changed to **Updating to Current Certificate Issuer**. In the backend, Oracle Key Vault has generated a new certificate for each rotated endpoint.

An endpoint that has been rotated will receive its new certificate the next time that it reaches out to Oracle Key Vault. In a multi-master cluster, the endpoint must reach out to the Oracle Key Vault cluster node on which the endpoint certificate was rotated in order to receive the new certificate. After the endpoint has received its new certificate, it must acknowledge receipt back to Oracle Key Vault. When that happens, the endpoint is considered to have completed rotation, and its **Common Name of Certificate Issuer** is changed from **Updating to Current Certificate Issuer** to the common name of the current Oracle Key Vault CA certificate. Its Endpoint Certificate Expiration field should show the new expiration date of its certificate.

- When one or more endpoints are rotated, a banner with the message **Certificate Rotation In Progress** is displayed on the Endpoints page. In a multi-master cluster, this banner is displayed on the **Endpoints** page of all cluster nodes. It remains until all rotated endpoints successfully receive and acknowledge receipt of their new certificate, that is, complete certificate rotation. You can check how many endpoints are still being rotated through the **Common Name of Certificate Issuer** field on the **Endpoints** page. For any endpoints that have not yet completed the rotation, this field will continue to display Updating to Current Certificate Issuer.
- In a multi-master cluster environment, the endpoint will receive its new certificate only when it reaches out to the cluster node on which its certificate was rotated. Since the endpoint can communicate with any node in the endpoint node scan list, the endpoint may run many operations before it reaches the rotating node and receives its certificate. The endpoint also has to acknowledge the receipt of the new certificates by reaching out to a node in the cluster.
- The endpoint certificate rotation times increases with the number of nodes in the cluster. The endpoints prioritize the nodes in the local subgroup, hence consider rotating the endpoint certificate on a node in its cluster subgroup.

- In a multi-master cluster environment, an endpoint must reach out to the rotating node to receive its new certificate. In a cluster with many nodes, the endpoint may reach out to any node in the scan list. If this happens during certificate rotation, then the endpoint makes a best-effort attempt to also reach out to the rotating node in order to receive its new certificate in as few operations as possible. However, depending on the number of nodes in the cluster and the periodicity with which the endpoint reaches out to Oracle Key Vault, the endpoint certificate rotation can take a long time to complete. Oracle recommends that you rotate the endpoint certificate well ahead of its expiration in order to avoid re-enrolling the endpoint, which would incur downtime.
- In a multi-master cluster environment, to view the endpoint's rotating node, navigate to the **Endpoints** page, and from the **Actions** drop-down list, select **Select Columns**. Add the **Rotating Node** column to the list of columns to display, and click **Apply**. Alternatively, the rotating node is also visible on the **Endpoint Details** page for a given endpoint.

 **Note:**

The rotating node can be viewed on the **Endpoints** and **Endpoint Details** pages only while a certificate rotation is in progress.

- When an endpoint is rotated, the validity of its new certificate depends on the Endpoint Certificate Validity configuration parameter of the Oracle Key Vault deployment, and on the remaining time to expiration of the Oracle Key Vault CA certificate.
- [Guidelines for Rotating Endpoint Certificates](#)
Consider these Oracle Key Vault guidelines before you rotate an endpoint certificate.

Related Topics

- [Finding the Expiration Date of Endpoint Certificates](#)
You can find the expiration date of endpoint certificates in the Oracle Key Vault management console.
- [Global Endpoint Configuration Parameters and Settings](#)
You can set endpoint configuration parameters and settings globally for all endpoints in the Oracle Key Vault management console.

Guidelines for Rotating Endpoint Certificates

Consider these Oracle Key Vault guidelines before you rotate an endpoint certificate.

Guidelines for Endpoint Certificate Rotation

- Do not rotate an endpoint certificate while a CA certificate rotation is in progress.
- Do not rotate an endpoint certificate rotation while a server or node certificate rotation is in progress.
- Do not alter the Endpoint Certificate Validity parameter while an endpoint certificate rotation is in progress.
- Consider rotating an endpoint from a node in the cluster subgroup that the endpoint is associated with.
- Do not rotate an endpoint if its certificate has already expired. Consider re-enrolling it instead.

- Consider re-enrolling an endpoint if it does not receive its new certificate due to network or other issues.
- Ensure that the endpoint software has been upgraded to version 21.5.0.0.0 or later before rotating the endpoint.

Related Topics

- [Certificates and the Restore Operation](#)
A third-party certificate installed at the time of a backup will not be copied when you restore another server from this backup.

Managing Endpoint Details

Endpoint details refers to endpoint name, type, description, platform, and email, and adding the endpoint to a group, or upgrading the endpoint software.

- [About Endpoint Details](#)
The Endpoint Details page provides a consolidated view of the endpoint.
- [Modifying Endpoint Details](#)
You can modify the endpoint name, endpoint type, description, platform, and email.

About Endpoint Details

The Endpoint Details page provides a consolidated view of the endpoint.

To access this page, you can select the **Endpoints** tab and then click the name of an endpoint. From here you can modify endpoint details and complete endpoint management tasks. (The following screen shows a partial view.)

Endpoint Details

[Cancel](#) [Send Enrollment Token](#) [Delete](#) [Save](#) [Rotate](#)

Endpoint Name: Make Unique [?](#)

Type:

Description:

Platform:

Administrator Email:

Status: **REGISTERED**

Creation Time: 12-DEC-2024 19:44:02

Certificate Expiration Time: 12-DEC-2025 19:44:06

Self Enrolled: **No**

Cluster Subgroup:

Strict IP Check Enforced: [?](#) **Global Setting: True**

Common Name of Certificate Issuer: **CA** [?](#) [Download \(PEM format\)](#) [Download \(wallet\)](#)

Last Active Time

Modifying Endpoint Details

You can modify the endpoint name, endpoint type, description, platform, and email.

In a multi-master cluster, endpoint details can only be modified while the endpoint is in the **PENDING** state by the creator on the node on which it was created.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Manage Endpoint privilege.

A user who has the Manage Endpoint privilege can only modify endpoints that this user has created. To see which endpoints that a user can manage, select the **Users** tab, then select **Manage Users**. Check the User Details page for the user in question, and scroll down to the Access to Endpoints area.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.
The Endpoints page appears listing all the Oracle Key Vault endpoints.
3. In the Endpoints page, click the name of the endpoint.

The Endpoint Details page appears.

Endpoint Details

Cancel
Send Enrollment Token
Delete
Save
Rotate

Endpoint Name Make Unique ?

Type

Description

Platform

Administrator Email

Status REGISTERED

Creation Time 12-DEC-2024 19:44:02

Certificate Expiration Time 12-DEC-2025 19:44:06

Self Enrolled No

Cluster Subgroup

Strict IP Check Enforced ? Global Setting: True

Common Name of Certificate Issuer CA ? Download (PEM format) Download (wallet)

Last Active Time

4. In the Endpoint Details page, modify any of the following: **Endpoint Name**, **Type**, **Description**, **Platform**, **Administrator Email**, **Cluster Subgroup** (for multi-master cluster environments only), or **Strict IP Check Enforced**.

The **Strict IP Check Enforced** setting is set to Use Global Settings by default for any endpoint that was created in Oracle Key Vault. If you modify this option to **True**, then Oracle Key Vault checks if the endpoint is connecting to it using the same IP that was used when the endpoint software was first deployed. If you modify this option to **False**, then Oracle Key Vault allows the endpoint to connect to it using any IP address. The global settings can be modified from the Global Endpoint Settings page. Oracle recommends that you set the effective endpoint setting to **True** (either through Global Endpoint Settings or Local Endpoint Settings) unless otherwise required.

See [Naming Guidelines for Objects](#).

5. Click **Save**.

Managing Global and Per-Endpoint Configuration Parameters and Settings

Oracle Key Vault provides global and per-endpoint configuration parameters and settings that you can set in the Oracle Key Vault management console.

- [About Managing Global and Per-Endpoint Configuration Parameters and Settings](#)
Users who have the System Administrator role or the Key Administrator role can centrally update certain endpoint configuration parameters and settings in the Oracle Key Vault management console.
- [Global Endpoint Configuration Parameters and Settings](#)
You can set endpoint configuration parameters and settings globally for all endpoints in the Oracle Key Vault management console.
- [Per-Endpoint Configuration Parameters and Settings](#)
You can set different endpoint configuration parameters and settings for individual endpoints in the Oracle Key Vault management console.

About Managing Global and Per-Endpoint Configuration Parameters and Settings

Users who have the System Administrator role or the Key Administrator role can centrally update certain endpoint configuration parameters and settings in the Oracle Key Vault management console.

Setting endpoint configuration parameters and settings globally (for all endpoints) or on a per-endpoint basis simplifies the process of managing multiple endpoints for system and key administrators.

You can perform the following types of global endpoint and per-endpoint settings:

- **Endpoint Settings:** This includes the setting to determine if strict IP checking will be enforced. Only a user who has the System Administrator role can modify this setting.
- **Endpoint configuration parameters:** These include settings that control features such as the length of time that a certificate is valid, timeouts for various PKCS 11 settings, and the timeout in seconds for a client's attempt to connect to an Oracle Key Vault server. Only a user who has the System Administrator role or the Manage Endpoint privilege for a specific endpoint can modify these parameters. Users who have the System Administrator role can modify endpoint configuration parameters for all endpoints. Users who have the Manage Endpoint privilege can modify the configuration parameters individually for each endpoint to which they have access. To do so, this user must go to the Details page for the endpoint, scroll to the bottom, and then modify the endpoints from there.
- **Keys and secrets:** This includes setting the extractable attribute value for symmetric keys. Only a user who has the Key Administrator role can modify this setting.

When Changes in Global and Per-Endpoint Values Take Effect

The configuration parameter values that are set in the Oracle Key Vault management console are applied to endpoints dynamically. The next time that the endpoint contacts Oracle Key Vault server, the updated configuration parameters are applied to the endpoint. If there is an error, then the update is not applied.

If you use the RESTful services utility, then Oracle Key Vault does not update the endpoint configuration. In this case, use `okvutil`, C SDK, JAVA SDK, or the PKCS11 library to apply the endpoint configuration updates.

In a multi-master cluster, replication of configuration parameters and settings depends on the replication lag. It is possible that an endpoint will not be able to get an update immediately because the node to which it is connected may not yet have received the new values of the parameters or settings. The endpoint will refresh its configuration when it connects to a node that has new values or if it has not refreshed its configuration in the past hour.

Precedence and Inheritance of Global and Per-Endpoint Values

Values that are set for an individual endpoint take precedence over the same values that are set globally. Global parameters and settings take effect when endpoint-specific parameters and settings are cleared. Oracle Key Vault uses the default system parameters and settings if both the global and endpoint specific parameters are cleared or not set from Oracle Key Vault management console. For the parameters Strict IP Check Enforced, and extractable attribute settings for Symmetric Key and Private Key, the global endpoint setting cannot be cleared, only the endpoint specific settings can be cleared.

In the case of keys and secrets, suppose you create a new Symmetric Key or Private Key but do not specify an extractable attribute value at the time of the Symmetric or Private Key's creation. The key will inherit the default value that has been set for the individual endpoint in which the Symmetric or Private key was created. If the default extractable attribute value has not been set for this endpoint, then the key will inherit the global endpoint value for the extractable attribute. If this global endpoint value has not been set, then the extractable attribute value defaults to true. Suppose later on, you change the global endpoint extractable attribute value so that future endpoints will use this value. Similar to configuration parameters, the values set in the individual endpoint that already exists take precedence over the same value that is set globally.

Related Topics

- [About Managing the Extraction of Symmetric or Private Keys from Oracle Key Vault](#)
The ability to restrict symmetric or private keys (extraction) from leaving Oracle Key Vault ensures a higher level of security for these objects.

Global Endpoint Configuration Parameters and Settings

You can set endpoint configuration parameters and settings globally for all endpoints in the Oracle Key Vault management console.

- [Setting Global Endpoint Configuration Parameters](#)
You can set global endpoint configuration parameters in the Oracle Key Vault management console.
- [Configuring Global Endpoint Settings for Keys and Secrets](#)
You can set the default extractable attribute value for new symmetric keys that you create or register in the endpoint configuration.

Setting Global Endpoint Configuration Parameters

You can set global endpoint configuration parameters in the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

2. Select the **Endpoints** tab, and then **Settings** from the left navigation bar.

The Endpoint Settings page appears.

The screenshot shows the 'Endpoint Settings' page. At the top right is a 'Save' button. Below it is a yellow warning banner: 'The self-enrollment process for endpoints is deprecated in Oracle Key Vault version 21.5.0.0.0. Use the RESTful services for endpoint enrollment instead.' Below the banner are two settings: 'Allow Endpoint Self Enrollment' with an unchecked checkbox, and 'Strict IP Check Enforced' with a dropdown menu set to 'True' and a help icon. Below this is the 'Global Endpoint Configuration Parameters' section, which has 'Save Defaults' and 'Save' buttons at the top right. It contains several input fields: 'Endpoint Certificate Validity (in days)', 'PKCS11 In-Memory Cache Timeout (in minutes)', 'PKCS11 Persistent Cache Timeout (in minutes)', 'PKCS11 Persistent Cache Refresh Window (in minutes)', 'PKCS11 Configuration Parameter Refresh Interval (in minutes)', 'Server Poll Timeout (in milliseconds)', 'PKCS11 Trace Directory Path', and 'Expire PKCS11 Persistent Cache on Database Shutdown' with an unchecked checkbox and a help icon.

3. In the Endpoint Settings section, configure the following settings:
 - **Allow Endpoint Self Enrollment:** This check box allows endpoints to be self-enrolled from the software download page.
 - **Strict IP Check Enforced:** This selection enables or disables the strict IP check for connections for all endpoints to the server/node (unless they have their own per-endpoint setting configured). Selecting the value **False** allows the endpoint to connect to Oracle Key Vault using an IP address different from the one used at the time of endpoint provisioning.
4. The **Allow Endpoint Self Enrollment** check box.
5. The **Strict IP Check Enforced** drop-down with **True** and **False** options.
6. In the **Global Endpoint Configuration Parameters** section, configure the following settings:
 - **Endpoint Certificate Validity (in days):** Specify the number of days for which the current endpoint certificate is valid. Valid settings are 365 through 1095. The default is 365. When the endpoint enrolled, the endpoint certificate validity period will always be less than the CA certificate validity period.
 - **PKCS 11 In-Memory Cache Timeout (in minutes):** Specify the duration in minutes for which the master encryption key is available after it is cached in the in-memory cache.
 - **PKCS 11 Persistent Cache Timeout (in minutes):** Specify the duration in minutes for which the master encryption key is available after it is cached in the persistent master encryption key cache.
 - **PKCS 11 Persistent Cache Refresh Window (in minutes):** Specify the duration in minutes to extend the period of time for which the master encryption key is available after it is cached in the persistent master encryption key cache.

- **PKCS11 Configuration Parameter Refresh Interval (in minutes):** Specify the frequency at which a long-running process will re-read the `okvclient.ora` configuration file.
- **Server Poll Timeout (in milliseconds):** Specify a timeout in seconds for a client's attempt to connect to an Oracle Key Vault server, before trying the next server in the list. The default value is 300 (milliseconds). In Oracle Key Vault clients first establish a non-blocking TCP connection to Oracle Key Vault to quickly detect unreachable servers. After the first attempt, the client makes a second and final attempt to connect to the server but this time waits for twice as long as the duration specified by the `SERVER_POLL_TIMEOUT` parameter. This is done to overcome possible network congestion or delays.
- **PKCS 11 Trace Directory Path:** Specify a directory to save the trace files.
- **Expire PKCS11 Persistent Cache on Database Shutdown:** Enables or disables the PKCS#11 persistent cache for a given endpoint database to automatically expire upon shutdown of the endpoint database.

 **Note:**

If the **Global Endpoints Configuration Parameters** values are empty, it indicates that the manually customized values in the `okvclient.ora` file are in effect. After you set these values in the Oracle Key Vault management console, you must edit these values from the Oracle Key Vault management console only. You cannot set empty values.

7. Click **Save**.

Related Topics

- [PKCS11_CACHE_TIMEOUT Parameter](#)
The `PKCS11_CACHE_TIMEOUT` parameter sets how long a master encryption key is available in the in-memory cache.
- [PKCS11_PERSISTENT_CACHE_TIMEOUT Parameter](#)
The `PKCS11_PERSISTENT_CACHE_TIMEOUT` parameter sets how long the master encryption is available in the persistent cache.
- [PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW Parameter](#)
The `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW` parameter extends time the master encryption key is available after it is cached in the persistent master encryption key cache.
- [PKCS11_CONFIG_PARAM_REFRESH_INTERVAL Parameter](#)
The `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL` parameter describes the frequency at which a long-running process will re-read the `okvclient.ora` configuration file.
- [EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN Parameter](#)
The `EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN` parameter ensures that the PKCS#11 persistent cache for a given endpoint database automatically expires upon shutdown of the endpoint database.
- [About Configuring Certificate Validity Period for Endpoint Certificates](#)
You can set the validity period for the endpoint certificates in the Global Endpoint Configuration parameters.

Configuring Global Endpoint Settings for Keys and Secrets

You can set the default extractable attribute value for new symmetric keys that you create or register in the endpoint configuration.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Endpoints** tab, and then **Settings** from the left navigation bar.
The Endpoint Settings page appears.
3. Scroll down to the Global Endpoint Settings for Keys & Secrets section.

4. From the **Symmetric Key** menu, select one of the following choices:
 - **True** (default) allows the object value to be extracted from Oracle Key Vault.
 - **False** prevents the object value from being extracted from Oracle Key Vault.
5. From the **Private Key** menu, select one of the following choices:
 - **True** (default) allows the object value to be extracted from Oracle Key Vault.
 - **False** prevents the object value from being extracted from Oracle Key Vault.
6. Save these settings using the following choices:
 - **Save Defaults** sets the default value (of **TRUE**) which is used as the default value or the extractable attribute.
 - **Save** sets a value that is used as the default value for the extractable attribute.

Per-Endpoint Configuration Parameters and Settings

You can set different endpoint configuration parameters and settings for individual endpoints in the Oracle Key Vault management console.

- [Modifying Configuration Parameters for an Individual Endpoint](#)
A user who has the System Administrator role or the Manage Endpoint privilege can set configuration parameters for individual endpoints.
- [Configuring Endpoint Settings for Keys and Secrets for an Individual Endpoint](#)
A user who has the Key Administrator role can set values for keys and secrets in an individual endpoint.

Modifying Configuration Parameters for an Individual Endpoint

A user who has the System Administrator role or the Manage Endpoint privilege can set configuration parameters for individual endpoints.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Manage Endpoint privilege.

A user with the System Administrator role can set configuration parameters for any endpoint. A user with the Manage Endpoint privilege can only set configuration parameters for the endpoints for which the user has the Manage Endpoint privilege.

2. Select the **Endpoints** tab, and then **Endpoints** from the left navigation bar.
3. In the Endpoints page, select the endpoint that you want to modify.
4. In the Endpoint Details page, scroll down to the **Endpoint Configuration Parameters** area.
5. Modify the configuration parameters as necessary.

The configuration parameters are the same as the configuration parameters that can be modified globally.

 **Note:**

If the **Endpoints Configuration Parameters** values are empty, it indicates that the manually customized values in the `okvclient.ora` file are in effect. After you set these values in the Oracle Key Vault management console, you must edit these values from the Oracle Key Vault management console only. You cannot set empty values.

6. Click **Save**.

Configuring Endpoint Settings for Keys and Secrets for an Individual Endpoint

A user who has the Key Administrator role can set values for keys and secrets in an individual endpoint.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role.
2. Select the **Endpoints** tab, and then **Endpoints** from the left navigation bar.
3. In the Endpoints page, select the endpoint that you want to modify.
4. In the Endpoint Details page, scroll down to the Endpoint Settings for Keys & Secrets area.
5. Select one of the following settings from the **Symmetric Key** menu:
 - **True** allows the object value to be extracted from Oracle Key Vault.
 - **False** prevents the object value from being extracted from Oracle Key Vault.
 -
 - **Use Global Settings** (default) uses the global endpoint setting for the **Extractable** attribute.
6. Select one of the following settings from the **Private Key** menu:
 - **True** allows the object value to be extracted from Oracle Key Vault.
 - **False** prevents the object value from being extracted from Oracle Key Vault.
7. Click **Save**.

Related Topics

- [Configuring Global Endpoint Settings for Keys and Secrets](#)
You can set the default extractable attribute value for new symmetric keys that you create or register in the endpoint configuration.

Default Wallets and Endpoints

You can use a default wallet, which is a type of virtual wallet, with an endpoint.

- [Associating a Default Wallet with an Endpoint](#)
A default wallet is a type of virtual wallet to which security objects are uploaded when a wallet is not explicitly specified.
- [Setting the Default Wallet for an Endpoint](#)
Setting a default wallet for an endpoint automatically uploads the endpoint's security objects to the wallet if another wallet is not explicitly specified.

Associating a Default Wallet with an Endpoint

A default wallet is a type of virtual wallet to which security objects are uploaded when a wallet is not explicitly specified.

Default wallets are useful for sharing with other endpoints such as nodes in an Oracle Real Application Clusters (Oracle RAC), or primary and standby nodes in Oracle Data Guard by having all endpoints use the same default wallet.

If you want to use the default wallet, then you must set this wallet after you register the endpoint before you enroll it. If you decide to use a default wallet after enrollment, then you must remove the default wallet and subsequently reenroll the endpoint.

An enrollment status of **registered** means that the endpoint has been added to Oracle Key Vault, but the endpoint software has not yet been downloaded and installed. When the status is **registered**, then you must associate the default wallet with the endpoint.

The endpoint's enrollment status becomes **enrolled** when you download and install the endpoint software to the endpoint. If you set the default wallet after you enroll the endpoint, then you must reenroll the endpoint to ensure that all future security objects created by the endpoint are automatically associated with that wallet.

In a multi-master cluster, you can only assign the default wallet on the same node where the endpoint and wallet were created when either are still in the `PENDING` state. After both are in the `ACTIVE` state, then there are no restrictions. After the default wallet is assigned and the endpoint is enrolled, the default wallet can be accessed from any node, as long as both are in the `ACTIVE` state and the information has been replicated to that node.

Setting the Default Wallet for an Endpoint

Setting a default wallet for an endpoint automatically uploads the endpoint's security objects to the wallet if another wallet is not explicitly specified.

Oracle requires that you set the default wallet right after registering the endpoint, and before downloading the endpoint software.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role or the Manage Endpoint privilege.

If you are logging on as a user with the Manage Endpoint privilege, then you must have full wallet access (Read/Write/Manage Wallet) on the wallet that you want to set as the endpoint's default wallet.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.
The Endpoints page appears listing all the Oracle Key Vault endpoints.
3. In the Endpoints page, click the name of the endpoint that you want.
4. In the Default Wallet pane, select **Choose Wallet**.

The Add Default Wallet page appears displaying a list of available wallets.

	Wallet Name	Description	Creation Time	Creator Node
<input checked="" type="radio"/>	Cloud_App_Wallet	This wallet contains the SSH private keys, public keys of the entities being managed on Oracle public cloud	09-NOV-2020 15:56:03	node1
<input type="radio"/>	Peoplesoft_DB_Connect_Client_Wallet	This wallet contains the passwords of DB users who login to Oracle DB.	09-NOV-2020 15:55:36	node1
<input type="radio"/>	Data_Recovery_Manager_Wallet	This wallet contains the SEPS (Secure External Password Store) wallet containing various client user credentials.	09-NOV-2020 15:55:09	node1
<input type="radio"/>	Siebel_DG_Wallet	This wallet contains the TDE Master Encryption keys for the DB hosting the Siebel application data.	09-NOV-2020 15:54:49	node1
<input type="radio"/>	Solaris_OS_Wallet	This wallet contains SSH Keytabs for the different solaris OS users.	09-NOV-2020 15:54:28	node1
<input type="radio"/>	ASM_Cluster_File_System_Wallet	This wallet contains the ACFS Volume Encryption keys for the management of ACFS storage.	09-NOV-2020 15:54:09	node1
<input type="radio"/>	XYZ_MYSQL_WALLET	This wallet contains the MySQL DB Symmetric keys for the management of XYZ Application.	09-NOV-2020 15:53:37	node1
<input type="radio"/>	GoldenGate_Replication_App_Wallet	This wallet contains the GoldenGate Master keys for the replication application.	09-NOV-2020 15:52:53	node1

5. Select a wallet from the list to be the default wallet by clicking the option to the left of the wallet, and then click **Select**.

The selected wallet name appears in the **Default Wallet** pane.

- Click **Save**.

Managing Endpoint Access to a Virtual Wallet

You can grant an endpoint access to a virtual wallet, and revoke or modify access when it is no longer necessary.

- [Granting an Endpoint Access to a Virtual Wallet](#)
An endpoint must have the Read, Modify, and Manage Wallet privileges on the wallet before security objects can be uploaded or downloaded.
- [Revoking Endpoint Access to a Virtual Wallet](#)
You can revoke access to a virtual wallet for an endpoint by using the **Endpoints** tab.
- [Viewing Wallet Items Accessed by Endpoints](#)
The term wallet items refers to the security objects to which the endpoint has access.

Granting an Endpoint Access to a Virtual Wallet

An endpoint must have the Read, Modify, and Manage Wallet privileges on the wallet before security objects can be uploaded or downloaded.

You can grant an endpoint access to a virtual wallet as soon as the endpoint has been added to Oracle Key Vault, when it is still in **registered** status.

- Log in to the Oracle Key Vault management console as an administrator who has the Key Administrator role or the Manage Endpoint privilege on the endpoint.
- Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.
- On the Endpoints page, select the endpoint that must have access to the virtual wallet.

The Endpoint Details page appears. Scroll down the page to the Access to Wallets pane.

<input type="checkbox"/>	Wallet Name	Type	Access	Grant type
<input type="checkbox"/>	Cloud_App_Wallet	General	Read, Write, Manage Wallet	Direct
<input type="checkbox"/>	Cloud_App_Wallet	General	Read, Write	via Endpoint Group
<input type="checkbox"/>	oracle_ssh_wallet	SSH Server	Read	Direct

1 - 3 of 3

- In the Access to Wallets pane, which lists the wallets the endpoint already has access to, click **Add** to add another wallet to this list.

The Select Wallet page appears. A user with the Manage Endpoint privilege can only view the wallets to which this user has access.

Add Default Wallet (X)

Cancel Select

Select Wallet

	Wallet Name	Description	Creation Time	Creator Node
<input checked="" type="radio"/>	Cloud_App_Wallet	This wallet contains the SSH private keys, public keys of the entities being managed on Oracle public cloud	09-NOV-2020 15:56:03	node1
<input type="radio"/>	Peoplesoft_DB_Connect_Client_Wallet	This wallet contains the passwords of DB users who login to Oracle DB.	09-NOV-2020 15:55:36	node1
<input type="radio"/>	Data_Recovery_Manager_Wallet	This wallet contains the SEPS (Secure External Password Store) wallet containing various client user credentials.	09-NOV-2020 15:55:09	node1
<input type="radio"/>	Siebel_DG_Wallet	This wallet contains the TDE Master Encryption keys for the DB hosting the Siebel application data.	09-NOV-2020 15:54:49	node1
<input type="radio"/>	Solaris_OS_Wallet	This wallet contains SSH Keytabs for the different solaris OS users.	09-NOV-2020 15:54:28	node1
<input type="radio"/>	ASM_Cluster_File_System_Wallet	This wallet contains the ACFS Volume Encryption keys for the management of ACFS storage.	09-NOV-2020 15:54:09	node1
<input type="radio"/>	XYZ_MYSQL_WALLET	This wallet contains the MySQL DB Symmetric keys for the management of XYZ Application.	09-NOV-2020 15:53:37	node1
<input type="radio"/>	GoldenGate_Replication_App_Wallet	This wallet contains the GoldenGate Master keys for the replication application.	09-NOV-2020 15:52:53	node1

1 - 10 of 11 (X)

- Select a wallet from the available list of wallets shown on the **Add Access to Endpoint** page.
- In the Select Access Level pane, select the appropriate level of access.
- Click **Save**.

Related Topics

- [Access Control Options](#)
Access control options enable you to set the type of privileges that users have to read, write, and delete security objects.

Revoking Endpoint Access to a Virtual Wallet

You can revoke access to a virtual wallet for an endpoint by using the **Endpoints** tab.

- Log in to the Oracle Key Vault management console as an administrator who has the Key Administrator role or the Manage Endpoint privilege.

If you have the Manage Endpoint privilege on the given endpoint, then you must have the same or a higher level of access to the wallet.

- Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.
- On the Endpoints page, select the endpoint name, which will display the Endpoint Details page.

Locate the Access to Wallets pane on this page. The Access to Wallets pane shows a list of wallets that the endpoint has access to.

4. Select the wallet that you want to revoke access to.
5. Click **Remove**.
6. In the confirmation window, click **OK**.

Viewing Wallet Items Accessed by Endpoints

The term wallet items refers to the security objects to which the endpoint has access.

1. Log in to the Oracle Key Vault management console as an administrator who has the Key Administrator role or the Manage Endpoint privilege.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.

The Endpoints page appears listing all the Oracle Key Vault endpoints.

3. In the Endpoints page, click the name of the endpoint to access the Endpoint Details page, and then scroll down to the Access to Wallet Items pane.

The Access to Wallet Items pane lists the wallet items that the endpoint has access to.

Access to Wallets				
				<input type="button" value="Add"/> <input type="button" value="Remove"/>
<input type="text" value="Search"/> <input type="button" value="Go"/> <input type="button" value="Actions"/>				
<input type="checkbox"/>	Wallet Name	Type	Access	Grant type
<input type="checkbox"/>	Cloud_App_Wallet	General	Read, Write, Manage Wallet	Direct
<input type="checkbox"/>	Cloud_App_Wallet	General	Read, Write	via Endpoint Group
<input type="checkbox"/>	oracle_ssh_wallet	SSH Server	Read	Direct
				1 - 3 of 3

Managing Endpoint Groups

An endpoint group is a named group of endpoints that share a common set of wallets.

- [How a Multi-Master Cluster Affects Endpoint Groups](#)
You can create endpoint groups on any node and they will have a cluster-wide presence.
- [Creating an Endpoint Group](#)
Endpoints that must share a common set of security objects stored in wallets can be grouped into an endpoint group.
- [Modifying Endpoint Group Details](#)
You can add endpoints and access mappings to an endpoint group after creating the endpoint group.
- [Granting an Endpoint Group Access to a Virtual Wallet](#)
You can grant an endpoint group access to a virtual wallet.

- [Adding an Endpoint to an Endpoint Group](#)
You can add an endpoint to a named endpoint group.
- [Removing an Endpoint from an Endpoint Group](#)
When you remove an endpoint from an endpoint group, this removes access to wallets that are associated with that endpoint group.
- [Deleting Endpoint Groups](#)
You can delete endpoint groups if their member endpoints no longer require access to the same virtual wallets.

How a Multi-Master Cluster Affects Endpoint Groups

You can create endpoint groups on any node and they will have a cluster-wide presence.

You can add, update, or delete endpoint groups in any node, but in read/write mode only.

The Oracle Key Vault server that becomes the initial node can have endpoint groups already created. These endpoint groups are used to initialize, or seed, the cluster. During induction, the endpoint groups in the cluster are replicated to a newly added node. Endpoint groups previously created in all other nodes added to the cluster will be removed during induction.

New endpoint groups added concurrently to the multi-master cluster on different nodes may have name conflicts. Oracle Key Vault automatically resolves any endpoint group name conflicts. These conflicts are displayed in a Conflicts Resolution page and key administrators can choose to rename them.

Related Topics

- [Naming Conflicts and Resolution](#)
Oracle Key Vault can resolve naming conflicts that can arise as users create objects such as endpoints, endpoint groups, and user groups.

Creating an Endpoint Group

Endpoints that must share a common set of security objects stored in wallets can be grouped into an endpoint group.

For example, endpoints using Oracle Real Application Clusters (Oracle RAC), Oracle GoldenGate, or Oracle Active Data Guard may need to share keys for access to shared data.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role or the Manage Endpoint Group privilege.

A user who has the Manage Endpoint Group privilege will only be able to manage the endpoint groups that he or she created.

2. Select the **Endpoints** tab, then **Endpoint Groups** in the left navigation bar.

The Endpoint Groups page appears.

Endpoint Groups Create Delete ↗

Q Go Actions ▾

<input type="checkbox"/>	Endpoint Group Name	Name Status	Description	Creation Time	Created By	Creator Node	Edit
<input type="checkbox"/>	CLOUD_APP_GROUP	ACTIVE	This group contains the members managing entities running on Oracle Public cloud	09-NOV-2020 15:47:24	OKVADMIN	node1	
<input type="checkbox"/>	SIEBEL_APPLICATION_DATAGUARD_GROUP	ACTIVE	This group contains all the primary and standby databases belonging to Siebel Application	09-NOV-2020 15:47:03	OKVADMIN	node1	
<input type="checkbox"/>	HR_APPLICATION_DB_GROUP	ACTIVE	This group contains all the related HR Application Databases	09-NOV-2020 15:46:44	OKVADMIN	node1	

1 - 3 of 3

Group Members

Q Go

Endpoint Name	Description
SIEBEL_PRIMARY_DB_AUSTIN	Primary Database deployed in Austin Datacenter for the Siebel application configured in Dataguard environment

Access to Wallets

Q Go

Wallet Name	Access
Siebel_DG_Wallet	Read, Write

1 - 1 of 1

3. Click **Create**.

The Create Endpoint Group page appears.

Create Endpoint Group Cancel Save

Name * Make Unique ⓘ

Description

4. Enter the name of the new group and a brief description.

Ensure that you follow the correct naming guidelines for objects.

5. If you are using a multi-master cluster, then choose whether to select the **Make Unique** check box.

Make Unique helps to control naming conflicts with names across the multi-master cluster environment. Endpoint groups that were created before an Oracle Key Vault conversion to a cluster node are not affected by naming conflicts.

- If you select **Make Unique**, then the endpoint group will be active immediately and users can use this endpoint group. Clicking **Make Unique** also displays a list of endpoints that you can add to the endpoint group.
- If you do not select **Make Unique**, then the endpoint group will be created in the `PENDING` state. Oracle Key Vault will then begin a name resolution operation and may rename the endpoint group to a name that is unique across the cluster. If there is a naming collision, then the collision will be reported on the Conflicts page on any node

in the cluster. The endpoint group will then be renamed to a unique name. You will need to go to a read/write node of the cluster and either accept the renamed endpoint group or change the endpoint name. If you change the endpoint group name, then this will restart the name resolution operation and the endpoint group will return to a `PENDING` state. An endpoint group in the `PENDING` state cannot be used to perform most operations.

6. Click **Save** to complete creating the endpoint group.

The new endpoint group now appears in the Endpoint Groups page.

Related Topics

- [Naming Guidelines for Objects](#)
The naming guidelines affect the following Oracle Key Vault objects: users, user groups, endpoints, endpoint groups, and virtual wallets.
- [Modifying Endpoint Group Details](#)
You can add endpoints and access mappings to an endpoint group after creating the endpoint group.

Modifying Endpoint Group Details

You can add endpoints and access mappings to an endpoint group after creating the endpoint group.

An endpoint can belong to more than one endpoint group. You cannot add one endpoint group to another endpoint group.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role or the Manage Endpoint Group privilege.

A user who has the Manage Endpoint Group privilege can only modify endpoint groups that he or she created.

2. Select the **Endpoints** tab, then **Endpoint Groups** in the left navigation bar.

The Endpoint Groups page appears.

3. Click the edit pencil icon in the **Edit** column corresponding to the endpoint group.

The Endpoint Group Details page appears.

Endpoint Group Details
Cancel **Save**

Name SIEBEL_APPLICATION_DATAGUARD_GROUP Make Unique 🗨️ ⚙️

Description This group contains all the primary and standby databases belonging to Siebel Application

Creation Time 10-DEC-2020 21:31:59

Access to Wallets
Add **Remove**

🔍 Go Actions

	Wallet Name	Access	Edit
<input type="checkbox"/>	Siebel_DG_Wallet	Read	✎

1 - 1 of 1

Endpoint Group Members
Add **Remove**

🔍 Go Actions

	Endpoint Name	Type	Status	Description	Creation Time
<input type="checkbox"/>	SIEBEL_STANDBY_DB_FRANKFURT	Oracle Database	REGISTERED	Standby Database deployed in Frankfurt Datacenter for the Siebel application configured in Dataguard environment	10-DEC-2020 22:03:53
<input type="checkbox"/>	SIEBEL_STANDBY_DB_PHEONIX	Oracle Database	REGISTERED	Standby Database deployed in Pheonix Datacenter for the Siebel application configured in Dataguard environment	10-DEC-2020 22:03:19
<input type="checkbox"/>	SIEBEL_PRIMARY_DB_AUSTIN	Oracle Database	REGISTERED	Primary Database deployed in Austin Datacenter for the Siebel application configured in Dataguard environment	10-DEC-2020 22:02:07

1 - 3 of 3

4. Modify the endpoint name as necessary.
See [Naming Guidelines for Objects](#).
5. Modify the description as needed.
6. Add or remove access to wallets or endpoint group members by clicking **Add** or **Remove**.
7. Click **Save**.

Granting an Endpoint Group Access to a Virtual Wallet

You can grant an endpoint group access to a virtual wallet.

In a multi-master cluster, you cannot grant access to an endpoint group that is in the `PENDING` state to a virtual wallet.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role or the Manage Endpoint Group privilege.

A user who has the Manage Endpoint Group privilege can grant endpoint group access to wallets for only endpoint groups that the user created.

2. Select the **Endpoints** tab, then **Endpoint Groups** in the left navigation bar.
The Endpoint Groups page appears.
3. Click the pencil icon in the **Edit** column corresponding to the endpoint group.
The Endpoint Group Details page appears.
4. In the **Access to Wallets** pane, click **Add**.
5. Select a virtual wallet from the available list.
6. Select an **Access Level**:

- **Read Only:** This level grants the endpoint group read access to the virtual wallet and its items.
- **Read and Modify:** This level grants the endpoint group read and write access to the virtual wallet and its items.

7. Select the **Manage Wallet** check box if you want endpoints to:
 - Add or remove objects from the virtual wallet.
 - Grant other endpoints or endpoint groups access to the virtual wallet.
8. Click **Save**.

Related Topics

- [Manage Endpoint Group Privilege Duties and Scope](#)
The Manage Endpoint Group object privilege on an endpoint group enables a user to perform all endpoint group management operations on the endpoint group.

Adding an Endpoint to an Endpoint Group

You can add an endpoint to a named endpoint group.

In a multi-master cluster, you cannot add an endpoint that is in the `PENDING` state to an endpoint group. Also, you cannot add an endpoint to an endpoint group that is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as an administrator who has the Key Administrator role or Manage Endpoint Group privilege.

A user who has the Manage Endpoint Group privilege can only add endpoints to endpoint groups that he or she created.

2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar.

The Endpoints page appears.

3. Select the endpoint you want to add to a group.

The Endpoint Details page appears.

4. Scroll to Endpoint Group Membership and then click **Add**.

The Add Endpoint Group Membership page appears.

Add Endpoint Group Membership
Cancel Save

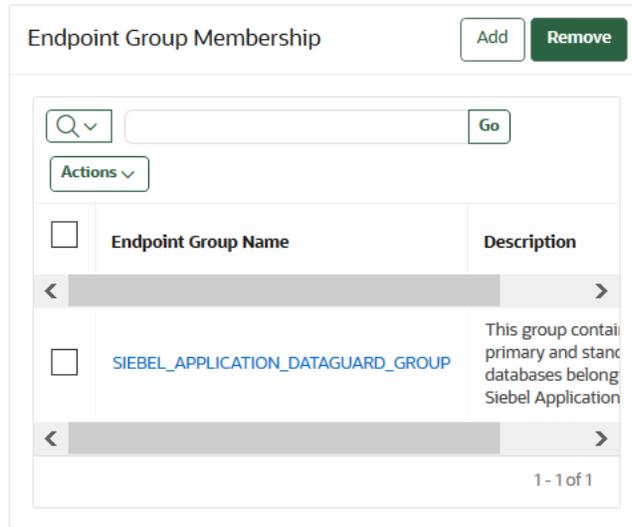
	Endpoint Group Name	Description	Creation Time
<input type="checkbox"/>	CLOUD_APP_GROUP	This group contains the members managing entities running on Oracle Public cloud	09-NOV-2020 15:47:24
<input type="checkbox"/>	SIEBEL_APPLICATION_DATAGUARD_GROUP	This group contains all the primary and standby databases belonging to Siebel Application	09-NOV-2020 15:47:03
<input type="checkbox"/>	HR_APPLICATION_DB_GROUP	This group contains all the related HR Application Databases	09-NOV-2020 15:46:44

1 - 3 of 3

A list of endpoint groups is displayed under **Endpoint Group Name**.

5. Check the boxes to the left of the endpoint groups you want to add the endpoint to.
6. Click **Save**.

The Endpoint Group Membership pane displays the checked endpoint group.



Related Topics

- [Creating an Endpoint Group](#)
Endpoints that must share a common set of security objects stored in wallets can be grouped into an endpoint group.

Removing an Endpoint from an Endpoint Group

When you remove an endpoint from an endpoint group, this removes access to wallets that are associated with that endpoint group.

The removal process completes the removal unless the endpoint has been separately granted access to the wallets, directly or through another endpoint group. In a multi-master cluster, you can remove multiple endpoints at the same time. In a multi-master cluster, you cannot remove an endpoint from an endpoint group that is in the `PENDING` state.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role or the Manage Endpoint Group privilege.
A user who has the Manage Endpoint Group privilege can only remove endpoints from endpoint groups that he or she created.
2. Select the **Endpoints** tab, then **Endpoint Groups** in the left navigation bar.
The Endpoint Groups page appears.
3. In Endpoint Groups, click the edit pencil icon next in the **Edit** column corresponding to the endpoint group.
The Endpoint Group Details page appears.
4. In the Endpoint Group Members pane, check the boxes to the left of the endpoint names to be removed.
5. Click **Remove**.
6. In the confirmation window, click **OK**.

Deleting Endpoint Groups

You can delete endpoint groups if their member endpoints no longer require access to the same virtual wallets.

This action removes the shared access of member endpoints to wallets, not the endpoints themselves. You can only delete an endpoint group that is in the `PENDING` state if it has no members or access to wallets.

1. Log in to the Oracle Key Vault management console as a user who has the Key Administrator role or the Manage Endpoint Group privilege.

A user who has the Manage Endpoint Group privilege can only remove endpoint groups that he or she created.

2. Select the **Endpoints** tab, then **Endpoint Groups** in the left navigation bar.
The Endpoint Groups page appears.
3. Check the boxes to the left of the endpoint group names that you want to delete.
4. Click **Delete**.
5. In the confirmation window, click **OK**.

13

Enrolling and Upgrading Endpoints for Oracle Key Vault

After an endpoint is registered in Oracle Key Vault, an endpoint administrator enrolls and provisions the endpoint to manage security objects in Key Vault.

- [About Endpoint Enrollment and Provisioning](#)
Endpoints are Oracle Key Vault clients that use the server to store and manage security objects, share them with trusted peers, and retrieve them.
- [Finalizing Enrollment and Provisioning](#)
To enroll and provision a registered endpoint an endpoint administrator must download and then install the `okvclient.jar` file.
- [Environment Variables and Endpoint Provisioning Guidance](#)
Environment variables such as `JAVA_HOME` and `OKV_HOME` must be correctly set so that Oracle Key Vault can access its utilities.
- [Endpoints That Do Not Use the Oracle Key Vault Client Software](#)
Third-party KMIP endpoints do not use the Oracle Key Vault software `okvutil` and `liborapkcs.so`.
- [Transparent Data Encryption Endpoint Management](#)
Oracle Key Vault can manage TDE keys by using the same PKCS#11 interface that TDE uses to communicate with an external keystore.
- [Endpoint `okvclient.ora` Configuration File](#)
Oracle Key Vault endpoint libraries and utilities use the `okvclient.ora` configuration file, which stores the configuration parameters associated with the endpoint.
- [`okvclient.ora` Parameters That Must Not Be Modified](#)
The `okvclient.ora` configuration file contains configuration parameters that you must not modify.
- [Upgrading Endpoint Software](#)
You can upgrade the endpoint software from the Oracle Key Vault management console login window.

About Endpoint Enrollment and Provisioning

Endpoints are Oracle Key Vault clients that use the server to store and manage security objects, share them with trusted peers, and retrieve them.

These clients can be systems like Oracle database servers, Oracle middleware servers, operating systems, and other information systems.

If you plan to configure the extraction of symmetric keys from Oracle Key Vault to false (to prevent them from being extracted), it is important that you first upgrade the endpoint to Oracle Key Vault release 21.4.

An Oracle Key Vault system administrator first adds (or registers) the endpoint to Key Vault, and then sends the endpoint's enrollment token (generated during registration) to the endpoint administrator. The endpoint administrator verifies the enrollment token before enrolling and

provisioning the endpoint. An enrolled endpoint can upload, download, and manage security objects using Key Vault.

Endpoint enrollment is a three-step process performed by two kinds of administrative users summarized in the following table.

Table 13-1 Summary of Endpoint Enrollment

Step#	Task	Performed by	Endpoint Status (as seen on Oracle Key Vault Management Console)
1.	<ol style="list-style-type: none"> 1. System administrator or user with Manage Endpoint privilege creates an endpoint. 2. If this is an Oracle database, a key administrator creates a virtual wallet. 3. System administrator adds or registers the endpoint to Oracle Key Vault. An enrollment token for the endpoint is generated. 4. System administrator sends the enrollment token to the endpoint administrator to complete the enrollment process. 	Users with the System Administrator role and Key Administrator role on Oracle Key Vault, or user with the Manage Endpoint privilege	Registered
2.	<ol style="list-style-type: none"> 1. Verify the enrollment token. 2. Submit enrollment token to download endpoint software <code>okvclient.jar</code> to the endpoint. 	Endpoint administrator using the Oracle Key Vault management console	Enrolled
3.	Install <code>okvclient.jar</code> on the endpoint.	Endpoint administrator on endpoint	Enrolled

Endpoint enrollment ensures that only authorized endpoints can communicate with Oracle Key Vault because the utilities needed to communicate are bundled with the `okvclient.jar` endpoint software file.

`okvclient.jar` contains the following:

- A Transport Layer Security (TLS) certificate and private key that the endpoint uses to authenticate itself to Oracle Key Vault
- A TLS certificate for Oracle Key Vault that serves as the root CA
- Endpoint libraries and utilities
- Additional information such as the Oracle Key Vault IP address that is used by `okvutil` to create the `okvclient.ora` configuration file

In an Oracle Real Application Clusters (RAC) environment, you must enroll and provision each Oracle RAC node as an endpoint. Each Oracle RAC-enabled database corresponds to one virtual wallet in Oracle Key Vault. Each Oracle RAC instance of that database corresponds to an endpoint in Oracle Key Vault. All endpoints for each database share the same wallet as their default wallet. You must download one distinct `okvclient.jar` for each instance.

Related Topics

- [Types of Endpoint Enrollment](#)
The first step in enrolling an endpoint is to add the endpoint to Oracle Key Vault.
- [Endpoint `okvclient.ora` Configuration File](#)
Oracle Key Vault endpoint libraries and utilities use the `okvclient.ora` configuration file, which stores the configuration parameters associated with the endpoint.
- [Oracle Key Vault `okvutil` Endpoint Utility Reference](#)
The `okvutil` utility enables you to perform tasks uploading and downloading security objects.

Finalizing Enrollment and Provisioning

To enroll and provision a registered endpoint an endpoint administrator must download and then install the `okvclient.jar` file.

- [Step 1: Enroll the Endpoint and Download the Software](#)
You must have the endpoint's enrollment token before you can download the endpoint software `okvclient.jar`.
- [Step 2: Prepare the Endpoint Environment](#)
You must ensure that you have the right version of the Java Development Toolkit (JDK) and that the Oracle environment variables are set.
- [Step 3: Install the Oracle Key Vault Software onto the Endpoint](#)
You can install the endpoint using downloaded `okvclient.jar` file.
- [Step 4: Perform Post-Installation Tasks](#)
The post-installation procedures include optionally configuring a TDE connection for the endpoint, checking the installation contents, and deleting the `okvclient.jar` file.

Step 1: Enroll the Endpoint and Download the Software

You must have the endpoint's enrollment token before you can download the endpoint software `okvclient.jar`.

After registering the endpoint, the Oracle Key Vault system administrator sends this endpoint's enrollment token to the endpoint administrator by email or other out-of-band method.

Note:

In Oracle Key Vault release 21.9, after a certificate rotation, password-protected endpoints may fail to connect to the Oracle Key Vault deployment, with `Server Connect Failed` errors. A fix for this issue is available as a patch for bug **37006283**. Apply this patch to the Oracle Key Vault release 21.9 deployment before provisioning endpoints.

1. Log in to the endpoint server as the endpoint administrator.
2. Connect to the Oracle Key Vault management console.

For example:

```
https://192.0.2.254
```

The login page to the Oracle Key Vault management console appears. **Do not log in.**

3. Click the **Endpoint Enrollment and Software Download** button, which is after the **Login** button.

The Enroll Endpoint & Download Software page appears.

4. If the endpoint was registered by an Oracle Key Vault system administrator, then do the following:
 - a. Enter the endpoint's enrollment token in **Enrollment Token**, and click **Submit Token**.
If the token is valid, then a valid token message appears after the **Enrollment Token** field. The **Endpoint Type**, **Endpoint Platform**, **Email** and **Description** fields are automatically populated with the values that were entered during endpoint registration.
If the token is invalid, then an invalid token message appears. Check the token and retry the download procedure.

- b. Click **Enroll** at the top right corner of the page.

5. In the directory window that appears, follow the prompt to save the `okvclient.jar` endpoint software file.

You must navigate to the directory where you want to save the file.

6. Save the file to a secure directory with appropriate permissions in place so that it cannot be read or copied by others.
7. Verify that the file has been downloaded.

If the download fails, then you must obtain a new enrollment token from the key administrator for the endpoint and repeat these steps, starting with Step 4. Note that if you did not download the file to the endpoint system, you must use an out-of-band method to copy the file to that system and install it there.

At this stage, you are ready to install the Oracle Key Vault `okvclient.jar` software file on the endpoint, starting with preparing the endpoint environment.

Related Topics

- [Step 2: Prepare the Endpoint Environment](#)
You must ensure that you have the right version of the Java Development Toolkit (JDK) and that the Oracle environment variables are set.

Step 2: Prepare the Endpoint Environment

You must ensure that you have the right version of the Java Development Toolkit (JDK) and that the Oracle environment variables are set.

1. Ensure that you have the necessary administrative privileges to install software on the endpoint.
2. Ensure that you have JDK 1.6 or later installed, and that the `PATH` environment variable includes the `java` executable (in the `JAVA_HOME/bin` directory).
Oracle Key Vault supports JDK versions, 1.6, 7, and 8. The 64-bit version of Java is required.
3. Run the shell utility `oraenv` or `source oraenv` command to set the correct environment variables on Oracle Database servers.
4. Check that the environment variables `ORACLE_BASE` and `ORACLE_HOME` are correctly set.
 - If you used `oraenv` to set these variables, then you must verify that `ORACLE_BASE` points to the root directory for Oracle Databases, and that `ORACLE_HOME` points to a sub-directory under `ORACLE_BASE` where an Oracle database is installed.
5. Shut down the database if you are installing the endpoint software for an Oracle database configured for online TDE master encryption key management.
6. As an endpoint administrator, shutdown the Oracle database server.

Step 3: Install the Oracle Key Vault Software onto the Endpoint

You can install the endpoint using downloaded `okvclient.jar` file.

Note:

To upgrade to the latest endpoint software for an enrolled endpoint, you can download the endpoint software without having to re-enroll the endpoint.

1. Ensure that you are logged in to the endpoint server as the endpoint administrator.

Note:

If you are installing the endpoint software for an Oracle database configured for online TDE master encryption key management, then shut down the database.

2. Ensure that the Oracle database is shutdown.
3. Navigate to the directory in which you saved the `okvclient.jar` file.
4. Confirm that the target directory exists, and that it is empty.
5. Run the `java` command to install the `okvclient.jar` file.

```
java -jar okvclient.jar -d /home/oracle/okv_home -v
```

In this specification:

- `-d` specifies the directory location for the endpoint software and configuration files, in this case `/home/oracle/okv_home`.
- `-v` writes the installation logs to the `/home/oracle/okv_home/log/okvutil.deploy.log` file at the server endpoint.

-o is an optional argument that enables you to overwrite the symbolic link reference to `okvclient.ora` when `okvclient.jar` is deployed in a directory other than the original directory. This argument is used only when you re-enroll an endpoint.

 **Note:**

If you are installing the `okvclient.jar` file on an Oracle Database Windows endpoint system, then make sure that the `java` command to install `okvclient.jar` file is run with administrator privileges and `$ORACLE_BASE/okv/$ORACLE_SID/` or `$ORACLE_HOME/okv/$ORACLE_SID` is a valid path on NTFS or ReFS filesystem. Failing to do so may cause installation to fail because on Windows platform administrator privileges are required to create `$ORACLE_BASE/okv/$ORACLE_SID/okvclient.ora` or `$ORACLE_HOME/okv/$ORACLE_SID` softlink during installation and softlink creation is only supported on NTFS or ReFS filesystem.

6. When you are prompted for a password, then perform either of the following two steps.

The optional password goes into two places: `okvutil` and in `ADMINISTER KEY MANAGEMENT`. With `okvutil`, only users who know that password can upload or download content to and from Oracle Key Vault. With `ADMINISTER KEY MANAGEMENT`, it becomes the password that you must use in the `IDENTIFIED BY password` clause. If you choose not to give a password, then `okvutil upload` and `download` commands will not prompt for a password, and the password for `ADMINISTER KEY MANAGEMENT` becomes `NULL`.

The choices for handling the password are as follows:

- If you want to create a password-protected wallet, at minimum enter a password between 8 and 30 characters and then press **Enter**. For better security, Oracle recommends that you include uppercase letters, lowercase characters, special characters, and numbers in the password. Use of following special characters is recommended: period (`.`), comma (`,`), underscore (`_`), plus sign (`+`), colon (`:`), and exclamation mark (`!`).

```
Enter new Key Vault endpoint password (<enter> for auto-login):
Key_Vault_endpoint_password
Confirm new endpoint password: Key_Vault_endpoint_password
```

A password-protected wallet is an Oracle wallet file that store the endpoint's credentials to access Oracle Key Vault. This password will be required whenever the endpoint connects to Oracle Key Vault.

- Alternatively, enter no password and then press **Enter**.

A successful installation of the endpoint software creates the following directories:

- `bin`: contains the `okvutil` program, the `root.sh` and `root.bat` scripts, and the binary file `okveps.x64`
- `conf`: contains the configuration file `okvclient.ora`
- `jlib`: contains the Java library files
- `lib`: contains the file `liborapkcs.so`
- `log`: contains the log files
- `ssl`: contains the TLS-related files and wallet files. The wallet files contain the endpoint credentials to connect to Oracle Key Vault.

The `ewallet.p12` file refers to a password-protected wallet. The `cwallet.sso` file refers to an auto-login wallet.

 **Note:**

Oracle recommends that you use the password-protected wallet.

Step 4: Perform Post-Installation Tasks

The post-installation procedures include optionally configuring a TDE connection for the endpoint, checking the installation contents, and deleting the `okvclient.jar` file.

1. Update the PKCS#11 library for the endpoint.

On UNIX platforms, the `liborapkcs.so` file contains the library that the Oracle database uses to communicate with Oracle Key Vault. On Windows platforms, the `liborapkcs.dll` file contains the library that the Oracle database uses to communicate with Oracle Key Vault.

If an endpoint uses online TDE master encryption key management by Oracle Key Vault, then you must install the PKCS#11 library by using `root.sh` or `root.bat` script.

 **Note:**

- You must run `root.sh` or `root.bat` script to install the Oracle Key Vault PKCS#11 library only once on a host machine that has multiple TDE-enabled Oracle databases that use Oracle Key Vault for master encryption key management.
 - Ensure that you run the `root.sh` or `root.bat` script only after the installation of the Oracle Key Vault endpoints for all of the TDE-enabled databases on the same host machine is complete.
 - Ensure that all of the TDE-enabled Oracle databases on this host are shutdown.
- **On Oracle Linux x86-64, Solaris, AIX, and HP-UX (IA) installations:** Log in as the `root` user and then run either of the following commands:

```
$ sudo bin/root.sh
```

Or:

```
$ su -  
# bin/root.sh
```

This command creates the directory tree `/opt/oracle/extapi/64/hsm/oracle/1.0.0`, changes ownership and permissions, then copies the PKCS#11 library into this directory.
 - **On Windows installations:** Run the following command:

```
bin\root.bat
```

This command copies the `liborapkcs.dll` file to the `C:\oracle\extapi\64\hsm\oracle\1.0.0` directory.

2. Use a command such as `namei` or `ls -l` to confirm that a softlink was created in `$ORACLE_BASE/okv/$ORACLE_SID/okvclient.ora` to point to the real file in the `/conf` subdirectory of the installation target directory.

If the `ORACLE_BASE` environment variable has not been set, then the softlink was created in `$ORACLE_HOME/okv/$ORACLE_SID`.

3. Start the Oracle databases if the installation of upgrade of the Oracle Key Vault endpoints for all of the TDE-enabled databases on this host machine is complete.
4. Run the `okvutil list` command to verify that the endpoint software installed correctly, and that the endpoint can connect to the Oracle Key Vault server.

```
$ ./okvutil list
```

If the endpoint is able to connect to Key Vault, then the `No objects found` message appears. If a `Server connect failed` message appears, then you must troubleshoot the installation for possible issues. Check that environment variables are correctly set. To get help on the endpoint software, run the following command:

```
java -jar okvclient.jar -h
```

Output similar to the following appears:

```
Production on Fri Apr 12 15:03:01 PDT 2019
Copyright (c) 1996, 2019 Oracle. All Rights Reserved.
Usage:
  java -jar okvclient.jar [-h | -help] [[-v | -verbose] [-d <destination directory>]
  [-o]]

Options:
  -h or -help : Display command help.
  -v or -verbose : Turn on the verbose mode. Logs will be written to files under
                  <destination directory>/log/ directory.
  -d <destination directory> : Specify the software installation directory.
  -o : Overwrite the current symbolic link to okvclient.ora.
```

Environment Variables and Endpoint Provisioning Guidance

Environment variables such as `JAVA_HOME` and `OKV_HOME` must be correctly set so that Oracle Key Vault can access its utilities.

- [How the Location of JAVA_HOME Location Is Determined](#)
The default location for the `okvclient.ora` file is the `$OKV_HOME/conf` directory.
- [Location of the okvclient.ora File and Environment Variables](#)
`$OKV_HOME` is the destination directory for the endpoint software specified with the `-d` option during installation.
- [Setting OKV_HOME for Non-Database Utilities to Communicate with Oracle Key Vault](#)
For non-database utilities, you must set the environment variable `OKV_HOME` to point to the destination directory for the endpoint software.
- [Environment Variables in sqlnet.ora File](#)
You must consider several points while using the `srvctl` utility on Oracle Database endpoints.

How the Location of JAVA_HOME Location Is Determined

The default location for the `okvclient.ora` file is the `$OKV_HOME/conf` directory.

When you provision endpoints you must know how the installation process determines the location of Java home and the `okvclient.ora` file.

The endpoint software installation process uses the following rules to determine the Java home location:

- If a user-defined `JAVA_HOME` environment variable exists, the installation process uses this value.
- If `JAVA_HOME` is not set, then the installation process looks for it in the `java.home` system property of the Java Virtual Machine (JVM).

After the `JAVA_HOME` path is determined, the installation process adds it to the `okvclient.ora` configuration file to be used by all `okvutil` commands.

You can force `okvutil` to use a different `JAVA_HOME` setting by using one of the following methods:

- Set the `JAVA_HOME` environment variable in the shell where you run `okvutil`:

```
setenv JAVA_HOME path_to_Java_home
```

Or:

```
export JAVA_HOME = path_to_Java_home
```

- Set the `JAVA_HOME` property directly in the `okvclient.ora` configuration file.

```
JAVA_HOME=path_to_Java_home
```

Restart the endpoint database after you set the `JAVA_HOME` variable in `okvclient.ora`.

You may need to periodically manually update the value of the `JAVA_HOME` environment variable setting in the `okvclient.ora` file. This can happen when a newer version of Java is installed and the previous version of Java is removed. To do this, first shut down the endpoint database, so that `okvclient.ora` is not overwritten by the database processes. Then, manually update the value of `JAVA_HOME` in `okvclient.ora`.

Location of the okvclient.ora File and Environment Variables

`$OKV_HOME` is the destination directory for the endpoint software specified with the `-d` option during installation.

The `okvclient.ora` file is a configuration file in the `$OKV_HOME/conf` directory .

In addition to the `$OKV_HOME/conf` file, the installation process creates a soft link to `okvclient.ora` for an existing database. The location of the soft link depends on the following:

- If the `$ORACLE_BASE` environment variable is set, then the installation process creates a soft link to the `okvclient.ora` configuration file (in `$OKV_HOME/conf`) in the `$ORACLE_BASE/okv/$ORACLE_SID` location.

If a soft link to `okvclient.ora` already exists in the `$ORACLE_BASE/okv/$ORACLE_SID` location, then the installation process accepts the existing soft link to `okvclient.ora` as a valid soft link.

- If the `$ORACLE_BASE/okv/$ORACLE_SID` directory is not set, then the installation process tries to create it.
- If the `$ORACLE_HOME` environment variable is set but the `$ORACLE_BASE` variable is not set, then the installation process creates a soft link for the `$ORACLE_HOME/okv/$ORACLE_SID` location to point to the configuration file in the `$OKV_HOME/conf` directory.

Setting OKV_HOME for Non-Database Utilities to Communicate with Oracle Key Vault

For non-database utilities, you must set the environment variable `OKV_HOME` to point to the destination directory for the endpoint software.

You must manually set `OKV_HOME` because the installation process does not set this variable automatically. Setting `OKV_HOME` enables utilities to communicate with Oracle Key Vault. These include utilities such as Oracle Recovery Manager (RMAN) that access Oracle Key Vault for keys.

You must set `OKV_HOME` in all environments where you will run utilities such as RMAN. For example, if you spawn a new `xterm` window, then you will need to set `OKV_HOME` in this environment before running RMAN.

Environment Variables in sqlnet.ora File

You must consider several points while using the `srvctl` utility on Oracle Database endpoints.

- If you are using the `srvctl` utility, and if you want to include environment variables in the `sqlnet.ora` configuration file, then you must set these environment variables in both the operating system and the `srvctl` environment.
- For Oracle Database endpoints, if you are using the `srvctl` utility and setting environment variables in `sqlnet.ora`, then you must set them in both the operating system and the `srvctl` environment.
- The operating system and `srvctl` utility should have `$ORACLE_SID`, `$ORACLE_HOME` and `$ORACLE_BASE` set to the same values.

Endpoints That Do Not Use the Oracle Key Vault Client Software

Third-party KMIP endpoints do not use the Oracle Key Vault software `okvutil` and `liborapkcs.so`.

In this case you must manually set the Transport Layer Security (TLS) authentication as follows:

1. Extract the `ssl` directory from the `okvclient.jar` file.

```
jar xvf okvclient.jar ssl
```
2. Use the following files to set up the TLS authentication:
 - `ssl/key.pem`: Endpoint private key
 - `ssl/cert.pem`: Endpoint certificate
 - `ssl/cert_req.pem`: Certificate request corresponding to `cert.pem`
 - `ssl/CA.pem`: Trust anchor for verifying the Oracle Key Vault server certificate

Transparent Data Encryption Endpoint Management

Oracle Key Vault can manage TDE keys by using the same PKCS#11 interface that TDE uses to communicate with an external keystore.

Therefore, you do not need to patch the database to use Oracle Key Vault for storing and retrieving TDE master encryption keys. Oracle Key Vault supplies the PKCS#11 library to communicate with Oracle Key Vault.

Oracle Key Vault improves upon TDE key management. For example, you can directly upload the keys in the wallet to Oracle Key Vault for long-term retention, to be shared with other database endpoints within the same endpoint group. Therefore, you do not need to store the wallet indefinitely after migration. Migration in this context means that the database is configured to use Oracle Key Vault for wallet backup, and that the administrator intends to migrate to an [online master encryption key](#) (formerly known as TDE direct connect).

You can continue to use the wallet, and upload wallet copies to Oracle Key Vault as part of every TDE key administration SQL operation, involving a `WITH BACKUP` SQL clause. However, be aware that TDE ignores the `WITH BACKUP` clause in an Oracle Key Vault online key deployment, even if it is required for the `ADMINISTER KEY MANAGEMENT` statement.

Oracle Database TDE endpoints for Oracle Key Vault. Endpoint enrollment and installation ensure that the PKCS#11 library is installed in the correct location for TDE to pick up and use. When the PKCS#11 library is installed, all other configurations and operations are in effect.

[Example 13-1](#) shows examples of setting an encryption key.

Example 13-1 Setting an Encryption Key

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY secret_passphrase -- For Oracle Database  
11g Release 2
```

```
ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY IDENTIFIED BY secret_passphrase WITH  
BACKUP; -- For Oracle Database 12c or later
```

Related Topics

- [Centralized Management of TDE Master Encryption Keys Using Online Master Encryption Keys](#)

You can use an online master encryption key to centralize the management of TDE master encryption keys over a direct network connection.

Endpoint `okvclient.ora` Configuration File

Oracle Key Vault endpoint libraries and utilities use the `okvclient.ora` configuration file, which stores the configuration parameters associated with the endpoint.

The `okvclient.ora` file consists of key-value pairs separated by an equal sign (=). At minimum, set the following parameters in the endpoint configuration file:

- `SERVER=node1_IP:node1_port/node1_DN,node2_IP:node2_port/node2_DN,...`

This parameter specifies the IP address and port number of the Oracle Key Vault server, separated by a colon and the server DN separated by a slash. If the port number is not specified, then it defaults to the standard KMIP port 5696.

- `STANDBY_SERVER=standby_server_IP:standby_server_port`

This is the standby server. If primary-standby is configured, then this parameter shows the standby IP address.

- `READ_SERVER=node1_IP:node1_port/node1_DN,node2_IP:node2_port/node2_DN,...`
This parameter specifies the list of read-only servers.
- `SSL_WALLET_LOC=directory`

This parameter specifies the location of the wallet containing TLS credentials for the endpoint.

- `SERVER_POLL_TIMEOUT=timeout_value`

You can use the `SERVER_POLL_TIMEOUT` parameter to specify a timeout for a client's attempt to connect to an Oracle Key Vault server before trying the next server in the list. The default value is 300 (milliseconds).

In Oracle Key Vault clients first establish a non-blocking TCP connection to Oracle Key Vault to quickly detect unreachable servers.

After the first attempt, the client makes a second and final attempt to connect to the server but this time waits for twice as long as the duration specified by the `SERVER_POLL_TIMEOUT` parameter. This is done to overcome possible network congestion or delays.

- `PKCS11_PERSISTENT_CACHE_FIRST=value` sets the persistent master encryption key cache operation mode.

The `CONF_ID` value in an `okvclient.ora` file is a unique internal value that helps an Oracle database to find its virtual wallet in Oracle Key Vault. Do not modify any settings in the `okvclient.ora` file. Instead, set endpoint configuration parameters through the Oracle Key Vault management console. Depending on your privileges, you can set these for individual endpoints or globally, for all endpoints.

Several `okvclient.ora` parameters can be modified through the Oracle Key Vault management console. You should modify the parameters only in the Oracle Key Vault management console. Depending on your privileges, you can set these parameters for individual endpoints or globally, for all endpoints.

- `PKCS11_CACHE_TIMEOUT=value` specifies the duration in minutes for which the master encryption key is available after it is cached in the in-memory cache. In the Oracle Key Vault management console, this setting is **PKCS 11 In-Memory Cache Timeout (in minutes)** in the Endpoint Settings page.
- `PKCS11_PERSISTENT_CACHE_TIMEOUT=value` specifies the duration in minutes for which the master encryption key is available after it is cached in the persistent master encryption key cache. In the Oracle Key Vault management console, this setting is **PKCS 11 Persistent Cache Timeout (in minutes)** in the Endpoint Settings page.
- `PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW=value` specifies the duration in minutes to extend the period of time for which the master encryption key is available after it is cached in the persistent master encryption key cache. In the Oracle Key Vault management console, this setting is **PKCS 11 Persistent Cache Refresh Window (in minutes)** in the Endpoint Settings page.
- `PKCS11_CONFIG_PARAM_REFRESH_INTERVAL=value` sets the frequency at which a long-running process will re-read the `okvclient.ora` configuration file. In the Oracle Key Vault management console, this setting is **PKCS11 Configuration Parameter Refresh Interval (in minutes)** in the Endpoint Settings page.
- `SERVER_POLL_TIMEOUT=value` specifies a timeout for a client's attempt to connect to an Oracle Key Vault server before trying the next server in the list. The default value is 300

(milliseconds). In the Oracle Key Vault management console, this setting is **Server Poll Timeout (in milliseconds)** in the Endpoint Settings page.

Related Topics

- [Persistent Master Encryption Key Cache Parameters](#)
Oracle Key Vault provides parameters to configure the persistent master encryption key cache.
- [Managing Endpoint Details](#)
Endpoint details refers to endpoint name, type, description, platform, and email, and adding the endpoint to a group, or upgrading the endpoint software.

okvclient.ora Parameters That Must Not Be Modified

The `okvclient.ora` configuration file contains configuration parameters that you must not modify.

These parameters are automatically populated when you add the endpoint to Oracle Key Vault. Do not modify them. They are as follows:

- `SERVER=node1_IP:node1_port/node1_DN,node2_IP:node2_port/node2_DN,...`
This parameter specifies the IP address and port number of the Oracle Key Vault server, separated by a colon and the server DN separated by a slash. If the port number is not specified, then it defaults to the standard KMIP port 5696.
- `STANDBY_SERVER=standby_server_IP:standby_server_port`
This is the standby server. If primary-standby is configured, then this parameter shows the standby IP address.
- `READ_SERVER=node1_IP:node1_port/node1_DN,node2_IP:node2_port/node2_DN,...`
This parameter specifies the list of read-only servers. Do not modify this parameter. Oracle Key Vault populates this setting when the endpoint node is added.
- `SERVER_DN=CN=server_certification,OU=product,O=company,L=city,ST=state_or_province,C=country`
- `GEN_TIMESTAMP=timestamp_information`
This parameter shows the time and date format used in the endpoint.
- `UPDATE_TIMESTAMP=updated_timestamp`
This parameter shows the date, time, and timezone for when the configuration file was last updated.
- `SW_TYPE=software_type`
This parameter shows the endpoint software type, for example, `ENROLLED_ENDPOINT_SOFTWARE`.
- `JAVA_HOME=path`
This parameter shows the directory that is defined by the `JAVA_HOME` environment variable.
- `OKV_JVM_LIB_PATH=path`
This parameter shows the directory that is defined by the `OKV_JVM_LIB_PATH` environment variable.
- `EP_TYPE=type`
This parameter indicates the type of the endpoint, such as Oracle Database.

- `OKV_HOSTNAME=host_name`
This parameter indicates the host server where Oracle Key Vault resides.
- `SSL_WALLET_LOC=directory`
This parameter specifies the location of the wallet containing TLS credentials for the endpoint.
- `_NOT_STRICT_PKCS11=value`
This parameter indicates the strict PKCS standard setting for use with Oracle ACFS.
- `PKCS11_NO_KMIP_OBJECT_ACCESS_CHECK=value`
This parameter indicates whether the endpoint will perform access checks.
- `_TRACE_DIR=.`
This parameter indicates the location where the PKCS trace files are generated. The `.` character, the default, means the current directory.
- `_TRACE_LEVEL=0`
This parameter determines the tracing level that is set for the PKCS traces. `0`, the default, disables tracing. Enter a value up to `16` to enable full tracing.
- `NUM_AFFINITY_RW_NODES=`
This parameter defines the number of read/write nodes that are in the cluster and have the same subgroup as the endpoint.
- `NUM_AFFINITY_RO_NODES=`
This parameter defines the number of read-only nodes that are in the cluster and have the same subgroup as the endpoint.
- `CONF_ID` is a unique internal value that helps an Oracle database to find its virtual wallet in Oracle Key Vault. Do not modify this value.

Upgrading Endpoint Software

You can upgrade the endpoint software from the Oracle Key Vault management console login window.

- [Step 1: Prepare the Endpoint Environment](#)
Ensure that you have the correct privileges and that the endpoint has the correct configuration, such as Oracle environment variables.
- [Step 2: Download the Oracle Key Vault Software onto the Endpoint](#)
You download the `okvclient.jar` file to local computer.
- [Step 3: Install the Oracle Key Vault Software onto the Endpoint](#)
You must be the endpoint administrator to install the Oracle Key Vault software onto the endpoint.
- [Step 4: Perform Post-Installation Tasks](#)
After you complete the installation, you can update the library used by TDE and then verify that the endpoint software was installed correctly.
- [Upgrading Endpoint Software on an Enrolled Endpoint](#)
You should upgrade the endpoint software on an enrolled endpoint any time you upgraded to a new release of Oracle Key Vault.

Step 1: Prepare the Endpoint Environment

Ensure that you have the correct privileges and that the endpoint has the correct configuration, such as Oracle environment variables.

These steps assume that the endpoint has already been enrolled in a previous release of Oracle Key Vault.

Note:

In Oracle Key Vault release 21.9, after a certificate rotation, password-protected endpoints may fail to connect to the Oracle Key Vault deployment, with `Server Connect Failed` errors. A fix for this issue is available as a patch for bug **37006283**. Apply this patch to the Oracle Key Vault deployment after upgrading all systems in the deployment, and before upgrading endpoint software.

1. Ensure that you have the necessary administrative privileges to install software on the endpoint.
2. Ensure that you have JDK 1.6 or later installed, and that the `PATH` environment variable includes the `java` executable (in the `JAVA_HOME/bin` directory).

Oracle Key Vault supports JDK versions, 1.6, 7, and 8. The 64-bit version of Java is required.
3. Run the shell utility `oraenv` or `source oraenv` command to set the correct environment variables on Oracle Database servers.
4. Check that the environment variables `ORACLE_BASE` and `ORACLE_HOME` are correctly set.
 - If you used `oraenv` to set these variables, then you must verify that `ORACLE_BASE` points to the root directory for Oracle Databases, and that `ORACLE_HOME` points to a sub-directory under `ORACLE_BASE` where an Oracle database is installed.
5. Shut down the database if you are installing the endpoint software for an Oracle database configured for online TDE master encryption key management.
6. As an endpoint administrator, shutdown the Oracle database server.

Step 2: Download the Oracle Key Vault Software onto the Endpoint

You download the `okvclient.jar` file to local computer.

You can download the endpoint software without having to reenroll the endpoint.

1. Log in to the endpoint server as the endpoint administrator.
2. Connect to the Oracle Key Vault management console.

For example:
`https://192.0.2.254`

The login page to the Oracle Key Vault management console appears. **Do not log in.**

3. In the lower-right corner of the login page under **Login**, click **Endpoint Enrollment and Software Download**.

The **Enroll Endpoint & Download Software** page appears.

- At the top of the page, click the **Download Endpoint Software Only** tab.

- In the Download Endpoint Software Only page, select the endpoint platform from the **Platform** drop down menu and click **Download**.
- Save the file `okvclient.jar` to a desired location.

Related Topics

- [Environment Variables and Endpoint Provisioning Guidance](#)
Environment variables such as `JAVA_HOME` and `OKV_HOME` must be correctly set so that Oracle Key Vault can access its utilities.
- [Centralized Management of TDE Master Encryption Keys Using Online Master Encryption Keys](#)
You can use an online master encryption key to centralize the management of TDE master encryption keys over a direct network connection.

Step 3: Install the Oracle Key Vault Software onto the Endpoint

You must be the endpoint administrator to install the Oracle Key Vault software onto the endpoint.

- Ensure that you are logged in to the endpoint server as the endpoint administrator.

Note:

If you are installing the endpoint software for an Oracle database configured for online TDE master encryption key management, then shut down the database.

- Ensure that the Oracle database is shutdown.
- Navigate to the directory in which you saved the `okvclient.jar` file.
- Confirm that the target directory exists, and that it is empty.
- Run the `java` command to install the `okvclient.jar` file.

```
java -jar okvclient.jar -d /home/oracle/okv_home -v
```

In this specification:

- `-d` specifies the directory location for the endpoint software and configuration files, in this case `/home/oracle/okv_home`.
- `-v` writes the installation logs to the `/home/oracle/okv_home/log/okvutil.deploy.log` file at the server endpoint.

`-o` is an optional argument that enables you to overwrite the symbolic link reference to `okvclient.ora` when `okvclient.jar` is deployed in a directory other than the original directory. This argument is used only when you re-enroll an endpoint.

 **Note:**

If you are installing the `okvclient.jar` file on an Oracle Database Windows endpoint system, then make sure that the `java` command to install `okvclient.jar` file is run with administrator privileges and `$ORACLE_BASE/okv/$ORACLE_SID/` or `$ORACLE_HOME/okv/$ORACLE_SID` is a valid path on NTFS or ReFS filesystem. Failing to do so may cause installation to fail because on Windows platform administrator privileges are required to create `$ORACLE_BASE/okv/$ORACLE_SID/okvclient.ora` or `$ORACLE_HOME/okv/$ORACLE_SID` softlink during installation and softlink creation is only supported on NTFS or ReFS filesystem.

6. When you are prompted for a password, then perform either of the following two steps.

The optional password goes into two places: `okvutil` and in `ADMINISTER KEY MANAGEMENT`. With `okvutil`, only users who know that password can upload or download content to and from Oracle Key Vault. With `ADMINISTER KEY MANAGEMENT`, it becomes the password that you must use in the `IDENTIFIED BY password` clause. If you choose not to give a password, then `okvutil` upload and download commands will not prompt for a password, and the password for `ADMINISTER KEY MANAGEMENT` becomes `NULL`.

The choices for handling the password are as follows:

- If you want to create a password-protected wallet, at minimum enter a password between 8 and 30 characters and then press **Enter**. For better security, Oracle recommends that you include uppercase letters, lowercase characters, special characters, and numbers in the password. Use of following special characters is recommended: period (`.`), comma (`,`), underscore (`_`), plus sign (`+`), colon (`:`), and exclamation mark (`!`).

```
Enter new Key Vault endpoint password (<enter> for auto-login):
Key_Vault_endpoint_password
Confirm new endpoint password: Key_Vault_endpoint_password
```

A password-protected wallet is an Oracle wallet file that store the endpoint's credentials to access Oracle Key Vault. This password will be required whenever the endpoint connects to Oracle Key Vault.

- Alternatively, enter no password and then press **Enter**.

A successful installation of the endpoint software creates the following directories:

- `bin`: contains the `okvutil` program, the `root.sh` and `root.bat` scripts, and the binary file `okveps.x64`
- `conf`: contains the configuration file `okvclient.ora`
- `jlib`: contains the Java library files
- `lib`: contains the file `liborapkcs.so`
- `log`: contains the log files
- `ssl`: contains the TLS-related files and wallet files. The wallet files contain the endpoint credentials to connect to Oracle Key Vault.

The `ewallet.p12` file refers to a password-protected wallet. The `cwallet.sso` file refers to an auto-login wallet.

 **Note:**

Oracle recommends that you use the password-protected wallet.

Related Topics

- [Environment Variables and Endpoint Provisioning Guidance](#)
Environment variables such as `JAVA_HOME` and `OKV_HOME` must be correctly set so that Oracle Key Vault can access its utilities.
- [Centralized Management of TDE Master Encryption Keys Using Online Master Encryption Keys](#)
You can use an online master encryption key to centralize the management of TDE master encryption keys over a direct network connection.

Step 4: Perform Post-Installation Tasks

After you complete the installation, you can update the library used by TDE and then verify that the endpoint software was installed correctly.

1. Update the PKCS#11 library for the endpoint.

On UNIX platforms, the `liborapkcs.so` file contains the library that the Oracle database uses to communicate with Oracle Key Vault. On Windows platforms, the `liborapkcs.dll` file contains the library that the Oracle database uses to communicate with Oracle Key Vault.

If an endpoint uses online TDE master encryption key management by Oracle Key Vault, then you must install the PKCS#11 library by using `root.sh` or `root.bat` script.

 **Note:**

- You must run `root.sh` or `root.bat` script to install the Oracle Key Vault PKCS#11 library only once on a host machine that has multiple TDE-enabled Oracle databases that use Oracle Key Vault for master encryption key management.
 - Ensure that you run the `root.sh` or `root.bat` script only after the installation of the Oracle Key Vault endpoints for all of the TDE-enabled databases on the same host machine is complete.
 - Ensure that all of the TDE-enabled Oracle databases on this host are shutdown.
- **On Oracle Linux x86-64, Solaris, AIX, and HP-UX (IA) installations:** Log in as the `root` user and then run either of the following commands:

```
$ sudo bin/root.sh
```

Or:

```
$ su -  
# bin/root.sh
```

This command creates the directory tree `/opt/oracle/extapi/64/hsm/oracle/1.0.0`, changes ownership and permissions, then copies the PKCS#11 library into this directory.

- **On Windows installations:** Run the following command:

```
bin\root.bat
```

This command copies the `liborapkcs.dll` file to the `C:\oracle\extapi\64\hsm\oracle\1.0.0` directory.

2. Use a command such as `namei` or `ls -l` to confirm that a softlink was created in `$ORACLE_BASE/okv/$ORACLE_SID/okvclient.ora` to point to the real file in the `/conf` subdirectory of the installation target directory.

If the `ORACLE_BASE` environment variable has not been set, then the softlink was created in `$ORACLE_HOME/okv/$ORACLE_SID`.

3. Start the Oracle databases if the installation of upgrade of the Oracle Key Vault endpoints for all of the TDE-enabled databases on this host machine is complete.
4. Run the `okvutil list` command to verify that the endpoint software installed correctly, and that the endpoint can connect to the Oracle Key Vault server.

```
$ ./okvutil list
```

If the endpoint is able to connect to Key Vault, then the `No objects found` message appears. If a `Server connect failed` message appears, then you must troubleshoot the installation for possible issues. Check that environment variables are correctly set. To get help on the endpoint software, run the following command:

```
java -jar okvclient.jar -h
```

Output similar to the following appears:

```
Production on Fri Apr 12 15:03:01 PDT 2019
Copyright (c) 1996, 2019 Oracle. All Rights Reserved.
Usage:
  java -jar okvclient.jar [-h | -help] [[-v | -verbose] [-d <destination directory>]
  [-o]]

Options:
  -h or -help : Display command help.
  -v or -verbose : Turn on the verbose mode. Logs will be written to files under
                  <destination directory>/log/ directory.
  -d <destination directory> : Specify the software installation directory.
  -o : Overwrite the current symbolic link to okvclient.ora.
```

Upgrading Endpoint Software on an Enrolled Endpoint

You should upgrade the endpoint software on an enrolled endpoint any time you upgraded to a new release of Oracle Key Vault.

This ensures that you have the latest software on both the Oracle Key Vault server and the endpoint. Oracle highly recommends this for optimum performance. Oracle Key Vault servers can work with endpoint software from the previous major release, but may not work properly with endpoint software that is older. To upgrade the software on an already enrolled endpoint you can download and install the software `okvclient.jar` on the endpoint. You do not need to reenroll the endpoint.

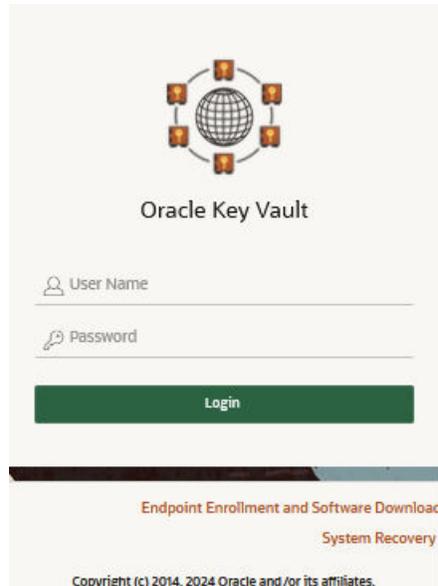
1. Log in to the endpoint server as the endpoint administrator.

2. Connect to the Oracle Key Vault management console.

For example:

`https://192.0.2.254`

The login page to the Oracle Key Vault management console appears. **Do not log in.**



3. In the lower-right corner of the login screen, under **Login**, click **Endpoint Enrollment and Software Download**.

4. In the Enroll Endpoint & Download Software page, click **Download Endpoint Software Only**.

The Download Endpoint Software Only page appears.

5. Select the **Platform** from the drop-down list and then click **Download**.

A directory window appears, and prompts you to save the endpoint software file `okvclient.jar`. Navigate to the folder where you want to save the file.

6. Save the file to an appropriate directory.
7. Verify that the file is downloaded.

After you complete these steps, you can install the Oracle Key Vault software on the endpoint, using the same steps that can be used for an unenrolled endpoint. Oracle recommends that you extract the jar file in the existing endpoint directory because the upgrade endpoint software will not work otherwise. For example:

```
java -jar /path/to/okvclient.jar -d /path/to/existing/ep/files -v
```

8. When the endpoint has been successfully upgraded, the following message appears:

The endpoint software for Oracle Key Vault upgraded successfully

Related Topics

- [Step 3: Install the Oracle Key Vault Software onto the Endpoint](#)
You must be the endpoint administrator to install the Oracle Key Vault software onto the endpoint.

Managing Keys for Oracle Products

You can use Oracle Key Vault with other Oracle features and products, such as Oracle GoldenGate or Oracle Data Guard.

- [Using a TDE-Configured Oracle Database in an Oracle RAC Environment](#)
Each Oracle Real Application Clusters (Oracle RAC) database has its own Oracle virtual wallet in Oracle Key Vault.
- [Using a TDE-Configured Oracle Database in an Oracle GoldenGate Environment](#)
Oracle Key Vault supports the use of Oracle wallets with Oracle GoldenGate shared secrets.
- [Using a TDE-Configured Oracle Database in an Oracle Data Guard Environment](#)
You can perform the activities such as uploading Oracle wallets or using online master encryption keys in an Oracle Data Guard environment.
- [Uploading Keystores from Automatic Storage Management to Oracle Key Vault](#)
You can copy a keystore from Automatic Storage Management (ASM) to Oracle Key Vault and vice versa in a two-step process.
- [MySQL Integration with Oracle Key Vault](#)
You can manage TDE encryption keys in MySQL with Oracle Key Vault.
- [Other Oracle Database Features That Oracle Key Vault Supports](#)
You can deploy Transparent Data Encryption (TDE) in multiple topologies with other database features that move data or use clustered deployments.

Using a TDE-Configured Oracle Database in an Oracle RAC Environment

Each Oracle Real Application Clusters (Oracle RAC) database has its own Oracle virtual wallet in Oracle Key Vault.

In an Oracle Real Application Clusters (Oracle RAC) environment, each Oracle RAC instance has its own endpoint in Oracle Key Vault; these endpoints share the same virtual wallet in Oracle Key Vault as their default wallet.

You can enable the cluster to share the virtual wallet by using either of the following approaches:

- If the Oracle RAC database is using TDE with individual wallets, then confirm that these wallets have the identical content. Run the `mkstore -wrl /directory/to/TDE-wallet -list` command to compare the content of each wallet. If they all contain the same keys, then upload the content of one of them into the shared virtual wallet in Oracle Key Vault.
- If the Oracle RAC database is using TDE with a shared wallet (which is the recommended deployment), then upload that wallet to Oracle Key Vault.
- Establish an auto-open connection with Oracle Key Vault.
- Migrate the Oracle RAC database to Oracle Key Vault.

As with single-instance database environments, after you download a password-protected wallet, you must manually open it. If you have one wallet on the primary node and then download the wallet to the other nodes, then you must explicitly open the wallets on each of these nodes.

Each Oracle RAC node is a different endpoint of the database and has its own individual persistent cache. For Oracle RAC databases, you should initiate a query from each Oracle RAC node to cache the latest master encryption key in the Oracle RAC node for uninterrupted operations

Related Topics

- [Migrating Between a Software Password Keystore and an External Keystore](#)
- [About the Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache ensures the availability of TDE master encryption keys.
- [Uploading Oracle Wallets](#)
The `okvutil upload` command uploads wallets to Oracle Key Vault.

Using a TDE-Configured Oracle Database in an Oracle GoldenGate Environment

Oracle Key Vault supports the use of Oracle wallets with Oracle GoldenGate shared secrets.

You can upload or migrate Oracle wallets that contain Oracle GoldenGate shared secrets and TDE master encryption keys to the Oracle Key Vault server.

- [Oracle Wallets in an Oracle GoldenGate Environment](#)
An Oracle GoldenGate shared secret can be in the same Oracle wallet where master encryption keys are stored.
- [Configuring Online Master Encryption Keys in an Oracle GoldenGate Deployment](#)
There are two configuration steps to using the online master encryption key in an Oracle GoldenGate deployment.
- [Migration of TDE Wallets in Oracle GoldenGate to Oracle Key Vault](#)
Oracle wallets can contain both a TDE master encryption key and an Oracle GoldenGate shared secret.

Oracle Wallets in an Oracle GoldenGate Environment

An Oracle GoldenGate shared secret can be in the same Oracle wallet where master encryption keys are stored.

In an environment where Oracle Key Vault is not used and an Oracle TDE-enabled database is configured with an Oracle wallet with Oracle GoldenGate, this database (called the source database) stores an Oracle GoldenGate shared secret in the same Oracle wallet where master encryption keys are stored.

This means that when you configure the source database as an Oracle Key Vault endpoint, the Oracle GoldenGate shared secret is stored in Oracle Key Vault in the same virtual wallet where the master encryption keys are stored for the TDE-enabled source database.

When you migrate an Oracle wallet that contains an Oracle GoldenGate shared secret and TDE master encryption keys to Oracle Key Vault using the `okvutil` command-line utility, the

default wallet for the TDE-enabled source database now stores the entire Oracle wallet migrated with shared secret and master encryption keys.

In addition, if the configured target database is an Oracle database, then you must ensure that this target database is TDE-enabled so that all the TDE commands can be replicated. The two Oracle TDE-enabled databases, source and target, do not need to have the same master encryption key in the Oracle wallet. If you configure this target database as a new Oracle Key Vault endpoint, then you can upload and download wallets to and from Oracle Key Vault as you normally would with any independent Oracle Key Vault endpoint. No additional configuration is necessary.

Configuring Online Master Encryption Keys in an Oracle GoldenGate Deployment

There are two configuration steps to using the online master encryption key in an Oracle GoldenGate deployment.

1. Configure a connection between the source database in the GoldenGate deployment and Oracle Key Vault.
2. Configure the storage of Oracle GoldenGate secrets in the Oracle wallet on the source database.

At this stage, the configuration is complete. If you have configured the `sqlnet.ora` file correctly and completed the other configuration steps required for TDE on the source database, then when you set the encryption key (using either `ALTER SYSTEM SET ENCRYPTION KEY` or `ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY`), a TDE master encryption key is created in Oracle Key Vault. You can encrypt tables or create encrypted tablespaces in the database. The encrypted data created in the source database continues to be replicated on the target database after you perform this procedure. The other Oracle GoldenGate shared secrets are stored in Oracle Key Vault.

See Also:

- [Step 1: Configure the Oracle Key Vault Server Environment](#) for instructions to connect a source database in GoldenGate to Oracle Key Vault.
- *Oracle Database Advanced Security Guide* for more information on configuring the storage of Oracle GoldenGate secrets in the source database.

Migration of TDE Wallets in Oracle GoldenGate to Oracle Key Vault

Oracle wallets can contain both a TDE master encryption key and an Oracle GoldenGate shared secret.

In an Oracle GoldenGate environment with a TDE-configured database, an Oracle wallet contains both the TDE master encryption keys and the Oracle GoldenGate shared secret.

You can also configure target Oracle TDE-enabled databases that are used in this Oracle GoldenGate environment to use Oracle Key Vault or continue to use an Oracle wallet. You should treat these databases as you would any standalone TDE database endpoint.

After you complete this migration, the configuration is complete. If you have configured the `sqlnet.ora` file correctly and completed the other configuration required for TDE, then when

you set the encryption key (using either `ALTER SYSTEM SET ENCRYPTION KEY` or `ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY`), a TDE master encryption key is created in Oracle Key Vault. You can continue to create and use encrypted tables or tablespaces in the database. The encrypted data created in the source database continues to be replicated on the target database after this procedure is performed.

Related Topics

- [Migration of TDE Wallets in Oracle GoldenGate to Oracle Key Vault](#)
Oracle wallets can contain both a TDE master encryption key and an Oracle GoldenGate shared secret.

Using a TDE-Configured Oracle Database in an Oracle Data Guard Environment

You can perform the activities such as uploading Oracle wallets or using online master encryption keys in an Oracle Data Guard environment.

- [About Uploading Oracle Wallets in an Oracle Data Guard Environment](#)
The upload operation enables both a primary and standby to benefit from the use of Oracle wallets.
- [Uploading Oracle Wallets in an Oracle Data Guard Environment](#)
You can upload an Oracle wallet to an Oracle Data Guard environment.
- [Performing an Online Master Encryption Key Connection in an Oracle Data Guard Environment](#)
The procedure for performing an online master encryption key in an Oracle Data Guard environment is the same as in a standard Oracle Database environment.
- [Migrating Oracle Wallets in an Oracle Data Guard Environment](#)
You can migrate an Oracle wallet in an Oracle Data Guard environment by using `okvutil` and `SQL*Plus`.
- [Reverse Migrating Oracle Wallets in an Oracle Data Guard Environment](#)
You can use `okvutil` and `SQL*Plus` to reverse migrate an Oracle wallet in an Oracle Data Guard environment.
- [Migrating an Oracle TDE Wallet to Oracle Key Vault for a Logical Standby Database](#)
You can migrate a TDE wallet to Oracle Key Vault to a logical standby database using Oracle Database release 12c or 18c.
- [Checking the Oracle TDE Wallet Migration for a Logical Standby Database](#)
You use `SQL*Plus` to check the migration.

About Uploading Oracle Wallets in an Oracle Data Guard Environment

The upload operation enables both a primary and standby to benefit from the use of Oracle wallets.

In an Oracle Data Guard environment with a TDE-enabled primary and standby databases using an Oracle wallet, you must physically copy the Oracle wallet file from the primary database to the standby and restart the managed recovery process after the initial TDE configuration or later, when you rekey the master encryption key on the primary database.

Whereas, when using Oracle Key Vault with a TDE-enabled Oracle Data Guard database, you must register the primary and standby databases in Oracle Key Vault as endpoints. You must

ensure that the endpoints for the primary and all standby databases share the same virtual wallet.

This way, the primary and standby databases can benefit from centralized key management without the need of a manual copy of the wallet file from the primary database to the standby database.

In an Oracle Data Guard environment, for a persistent cache, a rekey operation on the primary database will cache the master encryption key in its own persistent cache. When the new redo logs from the primary are applied on the standby, only then will the standby fetch the new key from the Oracle Key Vault and cache it in the persistent cache of the standby. There is a time lag between the caching of the key in primary and the caching of the key in standby. Oracle recommends that you synchronize the primary and standby as soon as possible after the rekey operation. In addition, you should confirm the content of the persistent cache on the primary and standby databases with the following command:

```
$ okvutil list -t okv_persistent_cache -l /<WALLET_ROOT>/okv/conf/
```

Related Topics

- [About the Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache ensures the availability of TDE master encryption keys.

Uploading Oracle Wallets in an Oracle Data Guard Environment

You can upload an Oracle wallet to an Oracle Data Guard environment.

1. Register one endpoint each for the primary and standby databases.
2. Download the `okvclient.jar` file for each endpoint on the respective databases.
3. Ensure that both the primary and standby database endpoints use the same default virtual wallet.

Related Topics

- [Managing Endpoints](#)
You can enroll, reenroll, suspend, rotate, and delete endpoints.

Performing an Online Master Encryption Key Connection in an Oracle Data Guard Environment

The procedure for performing an online master encryption key in an Oracle Data Guard environment is the same as in a standard Oracle Database environment.

Related Topics

- [Centralized Management of TDE Master Encryption Keys Using Online Master Encryption Keys](#)
You can use an online master encryption key to centralize the management of TDE master encryption keys over a direct network connection.

Migrating Oracle Wallets in an Oracle Data Guard Environment

You can migrate an Oracle wallet in an Oracle Data Guard environment by using `okvutil` and `SQL*Plus`.

1. Use the `okvutil upload` command to upload the contents of the local Oracle wallet that is on the primary database to Oracle Key Vault.
2. Perform the steps to migrate the wallet, as described in [Migrating an Existing TDE Wallet to Oracle Key Vault](#).
3. Close the existing Oracle wallet on the standby database.
 - For Oracle Database 11g release 2, as a user who has been granted the `ALTER SYSTEM` system privilege:


```
ALTER SYSTEM SET ENCRYPTION WALLET CLOSE
IDENTIFIED BY "Key_Vault_endpoint_password";
```
 - For Oracle Database 12c or later, as a user who has been granted the `SYSKM` administrative privilege:


```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE
IDENTIFIED BY "Key_Vault_endpoint_password";
```
4. Restart the standby database.
5. Open the Oracle wallet.
 - For Oracle Database 11g release 2, as a user who has been granted the `ALTER SYSTEM` system privilege:


```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN
IDENTIFIED BY "Key_Vault_endpoint_password";
```
 - For Oracle Database 12c or 18c, as a user who has been granted the `SYSKM` administrative privilege:


```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN
IDENTIFIED BY "Key_Vault_endpoint_password";
```
6. Start the apply process on the standby database, as described in *Oracle Data Guard Concepts and Administration*.

Related Topics

- [okvutil upload Command](#)
The `okvutil upload` command uploads security objects to Oracle Key Vault.
- [Migrating Existing TDE Wallets to Oracle Key Vault](#)
A migrated TDE wallet can be used to restore database contents that were previously encrypted by TDE.
- *Oracle Data Guard Concepts and Administration*

Reverse Migrating Oracle Wallets in an Oracle Data Guard Environment

You can use `okvutil` and `SQL*Plus` to reverse migrate an Oracle wallet in an Oracle Data Guard environment.

1. Use the `okvutil download` command to download the Oracle wallet keys onto the primary database from Oracle Key Vault. Download these keys to a local keystore.
2. Perform a reverse migration, as described in *Oracle Database Advanced Security Guide*.
3. Close the existing Oracle wallet on the standby database.
 - For Oracle Database 11g release 2:


```
ALTER SYSTEM SET ENCRYPTION WALLET CLOSE
IDENTIFIED BY "Key_Vault_endpoint_password";
```

- For Oracle Database 12.1.0.2 and later:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE
IDENTIFIED BY "Key_Vault_endpoint_password";
```
- 4. Copy the Oracle wallet from the primary database to the standby database, as described in *Oracle Database Advanced Security Guide*.
- 5. Open the Oracle wallet on the standby database.
 - For Oracle Database 11g release 2:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN
IDENTIFIED BY "Key_Vault_endpoint_password";
```
 - For Oracle Database 12.1.0.2 and later:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN
IDENTIFIED BY "Key_Vault_endpoint_password";
```
- 6. Start the apply process on the standby database, as described in *Oracle Data Guard Concepts and Administration*.

If the endpoint password and the local TDE wallet password are different, then use the auto-login HSM feature.

Related Topics

- [Migrating Between a Software Password Keystore and an External Keystore](#)
- [okvutil upload Command](#)
The `okvutil upload` command uploads security objects to Oracle Key Vault.
- *Oracle Data Guard Concepts and Administration*

Migrating an Oracle TDE Wallet to Oracle Key Vault for a Logical Standby Database

You can migrate a TDE wallet to Oracle Key Vault to a logical standby database using Oracle Database release 12c or 18c.

1. Register the primary and standby endpoints to have the same default virtual wallet.
2. If necessary, download and install the `okvclient.jar` file to each endpoint.
3. Perform the migration on the primary database.
4. Complete the SQL apply process on the logical standby and then restart the standby database, as described in *Oracle Data Guard Concepts and Administration*.
5. To check that the status of that migration was successful, query the `V$ENCRYPTION_WALLET` dynamic view.

Related Topics

- [Migrating an Existing TDE Wallet to Oracle Key Vault](#)
You can use the `okvutil upload` command to start the migration of a TDE-enabled database from an existing TDE wallet to Oracle Key Vault.
- *Oracle Data Guard Concepts and Administration*

Checking the Oracle TDE Wallet Migration for a Logical Standby Database

You use SQL*Plus to check the migration.

In an Oracle Database Release 12c environment, after you have migrated an Oracle TDE wallet in a logical standby configuration, you can check the configuration.

1. In the standby database instance, log in to SQL*Plus.

For example:

```
sqlplus / as sysdba
```

2. Query the `WRL_TYPE` and `WALLET_ORDER` columns of the `V$ENCRYPTION_WALLET` dynamic view.

The `V$ENCRYPTION_WALLET` view tracks the primary keystore. If you have only a single wallet configured, then the `WALLET_ORDER` column is set to `SINGLE`. In a two-wallet or mixed configuration, the column is set to `PRIMARY` or `SECONDARY`, depending on where the active master encryption key is located.

For example, in the following, only a single wallet is configured:

```
SELECT WRL_TYPE, WALLET_ORDER FROM V$ENCRYPTION_WALLET;
```

```
WRL_TYPE          WALLET_ORDER
-----
FILE              SINGLE
```

In this query in a logical standby configuration, the active master encryption key has been migrated to an Oracle Key Vault virtual wallet:

```
SELECT WRL_TYPE, WALLET_ORDER FROM V$ENCRYPTION_WALLET;
```

```
WRL_TYPE          WALLET_ORDER
-----
FILE              SECONDARY
OKV               PRIMARY
```

This query should show the `OKV` as the `PRIMARY` wallet in both the primary and standby database for the logical configuration.

Uploading Keystores from Automatic Storage Management to Oracle Key Vault

You can copy a keystore from Automatic Storage Management (ASM) to Oracle Key Vault and vice versa in a two-step process.

- [About Uploading Keystores from Automatic Storage Management to Oracle Key Vault](#)
 Uploading a keystore from Oracle Automatic Storage Management (ASM) to Oracle Key Vault is a two-step process.
- [Uploading a Keystore from Automatic Storage Management to Oracle Key Vault](#)
 You can use the `ADMINISTER KEY MANAGEMENT` statement to move a software keystore out of Automatic Storage Management (ASM).
- [Copying a Keystore from Oracle Key Vault to Automatic Storage Management](#)
 You use both `okvutil download` and SQL*Plus to complete the copy process.

About Uploading Keystores from Automatic Storage Management to Oracle Key Vault

Uploading a keystore from Oracle Automatic Storage Management (ASM) to Oracle Key Vault is a two-step process.

1. Copy the keystore from ASM to the file system.
2. Upload the keystore from the file system to Oracle Key Vault.

Copying a keystore from ASM to the file system or vice versa requires the keystore merge operation that merges one software keystore to an existing key store. Therefore, in order to copy a keystore from a source path to a target path, a keystore must exist at the target path.

Uploading a Keystore from Automatic Storage Management to Oracle Key Vault

You can use the `ADMINISTER KEY MANAGEMENT` statement to move a software keystore out of Automatic Storage Management (ASM).

1. Initialize a target keystore on the file system with the following SQL statement:

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE targetKeystorePath
IDENTIFIED BY targetKeystorePassword;
```

In this specification:

- *targetKeystorePath* is the directory path to the target keystore on the file system.
- *targetKeystorePassword* is a password that you create for the keystore.

For example:

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/etc/ORACLE/KEYSTORE/DB1/'
IDENTIFIED BY "destination_password";
```

In this specification, */etc/ORACLE/KEYSTORE/DB1/* is the path to the target keystore in the file system and *destination_password* is the keystore password.

You now can copy the keystore from ASM to the target keystore.

2. Copy the keystore from ASM to the target keystore that you just created.

This step requires that you merge the keystore from ASM to the file system as follows:

```
ADMINISTER KEY MANAGEMENT MERGE KEYSTORE 'srcKeystorePath'
[IDENTIFIED BY srcKeystorePassword]
INTO EXISTING KEYSTORE 'targetKeystorePath'
IDENTIFIED BY targetKeystorePassword
WITH BACKUP [USING backupIdentifier];
```

In this specification:

- *srcKeystorePath* is the directory path to the source keystore.
- *srcKeystorePassword* is the source keystore password.
- *targetKeystorePath* is the path to the target keystore.
- *targetKeystorePassword* is the target keystore password.
- *backupIdentifier* is the backup identifier to be added to the backup file name.

For example:

```
ADMINISTER KEY MANAGEMENT MERGE KEYSTORE '+DATA/<DB_UNIQUE_NAME>/tde' IDENTIFIED BY
"srcPassword" INTO EXISTING KEYSTORE '/etc/ORACLE/KEYSTORE/DB1/' IDENTIFIED BY
"destination_password" WITH BACKUP;
```

The keystore is copied to the file system and can now be uploaded to Oracle Key Vault.

3. Upload keystore from file system to Oracle Key Vault by using the `okvutil upload` command.

```
$ okvutil upload -l location -t type
```

In this specification:

- `location` is the path to the target keystore in the file system
- `type` is `wallet`

For example:

```
$ okvutil upload -l /etc/ORACLE/KEYSTORE/DB1 -t wallet
```

Copying a Keystore from Oracle Key Vault to Automatic Storage Management

You use both `okvutil download` and `SQL*Plus` to complete the copy process.

To copy a keystore from Oracle Key Vault to Automatic Storage Management (ASM), use the reverse procedure from copying the keystore from ASM to Oracle Key Vault.

1. Initialize a target keystore on the file system, *if the keystore does not exist*.

If the keystore does exist on the file system, then bypass this step.

2. Copy the keystore from Oracle Key Vault to the target keystore on the file system using the `okvutil download` command.

```
$ okvutil download -l location -t type
```

In this specification:

- `location` is the path to the target keystore in the file system
- `type` is `wallet`

For example:

```
$ okvutil download -l /etc/ORACLE/KEYSTORE/DB1 -t wallet
```

3. Initialize a keystore on the ASM instance by using the following SQL statement:

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE asmKeystorePath
IDENTIFIED BY asmKeystorePassword;
```

In this specification:

- `asmKeystorePath` is the directory path for the keystore on the ASM file system.
 - `asmKeystorePassword` is a password that you create for the keystore.
4. Copy the keystore to the initialized ASM keystore that you just created.

This step merges the keystore from the file system to ASM as follows:

```
ADMINISTER KEY MANAGEMENT MERGE KEYSTORE srcKeystorePath  
IDENTIFIED BY srcKeystorePassword  
INTO EXISTING KEYSTORE asmKeystorePath  
IDENTIFIED BY asmKeystorePassword  
WITH BACKUP USING backupIdentifier;
```

In this specification:

- *srcKeystorePath* is the directory path to the source keystore.
- *srcKeystorePassword* is the source keystore password.
- *asmKeystorePath* is the path to the ASM keystore.
- *asmKeystorePassword* is the ASM keystore password.
- *backupIdentifier* is the backup identifier to be added to the backup file name.

MySQL Integration with Oracle Key Vault

You can manage TDE encryption keys in MySQL with Oracle Key Vault.

Oracle Key Vault supports integration with MySQL from Release 12.2 or later.



Note:

MySQL Windows databases are not supported.

Oracle Key Vault can manage MySQL TDE encryption keys.

Other Oracle Database Features That Oracle Key Vault Supports

You can deploy Transparent Data Encryption (TDE) in multiple topologies with other database features that move data or use clustered deployments.

Data movement and replication are major challenges for Oracle Advanced Security TDE because it must keep the master encryption key synchronized at both source and target. To help with these challenges, Oracle Key Vault supports common Oracle Database features.

To move data, Oracle Key Vault supports the following:

- Oracle Recovery Manager (RMAN) backup and recovery operations
- Oracle Data Pump
- Transportable tablespaces (Oracle Database 12c or later)
- Pluggable database plug/unplug
- Pluggable database remote clones

For clustered deployments, Oracle Key Vault supports the following:

- Oracle Data Guard
- Oracle Real Application Clusters (Oracle RAC)
- Oracle GoldenGate

 **Note:**

You can rotate the TDE master encryption key of the database only from the database using the Administer Key Management Set Encryption Key commands. You cannot initiate the TDE master encryption key rotation centrally from the Oracle Key Vault management console.

Related Topics

- [Managing Cloned PDBs with Encrypted Data in United Mode](#)

SSH Keys Management Concepts

Secure Shell (SSH) is the protocol used for remote administration and operations of hosts.

- [SSH Protocol](#)
Secure Shell (SSH) is a protocol for logging in to remote hosts securely. In most organizations, users routinely use the SSH software to log in to remote hosts for administrative purposes, running commands, tunneling or copying files.
- [SSH Public Key Authentication](#)
Client authentication can be done by one of the user authentication methods like password authentication, host-based authentication, or the public key authentication. Public key authentication makes use of the public-private key pair to establish the identity of the user connecting to the Secure Shell (SSH) server.
- [OpenSSH Implementation of the SSH Protocol](#)
OpenSSH is an implementation of the Secure Shell (SSH) protocol that is available with Oracle Linux and in general with Linux and Unix-based operating systems.
- [Challenges with SSH Public Key Authentication](#)
Following sections describe the challenges faced with SSH Public Key Authentication.
- [Controlling Access to SSH Server Centrally with Oracle Key Vault](#)
Oracle Key Vault can centrally manage the access to Secure Shell (SSH) servers. Centralized access control improves security, enables quicker responses to threats, reduces human error, and simplifies authorization management at scale.
- [Managing SSH User Keys with Oracle Key Vault](#)
Oracle Key Vault can centrally manage the Secure Shell (SSH) key pairs of the SSH client users.
- [Oracle Key Vault and SSH Integration](#)
While Oracle Key Vault can manage the Secure Shell (SSH) keys and can also control the access to SSH servers independently, ideally both the SSH keys and access control of SSH servers should be managed in Oracle Key Vault centrally.
- [Supported Platforms for SSH Server and Client Endpoints](#)
Oracle supports 64-bit platforms for SSH server and client endpoints.

SSH Protocol

Secure Shell (SSH) is a protocol for logging in to remote hosts securely. In most organizations, users routinely use the SSH software to log in to remote hosts for administrative purposes, running commands, tunneling or copying files.

SSH has replaced the less secure means of remote host access like telnet and less secure means of remote copy like `rcp`.

SSH protocol has a client-server architecture and consists of three components: the transport layer protocol, the user authentication protocol, and the connection protocol. SSH client users can be users or other processes on the client hosts running the SSH client software. SSH server is a host running the SSH service. The SSH client user initiates the connection with the SSH server. The protocol authenticates the server, establishes an encrypted session and also establishes the authenticity of the client user.

There are many implementations of the SSH protocol. OpenSSH is the open source implementation commonly found on GNU/Linux and UNIX-like operating systems.

Figure 15-1 SSH Protocol



SSH Public Key Authentication

Client authentication can be done by one of the user authentication methods like password authentication, host-based authentication, or the public key authentication. Public key authentication makes use of the public-private key pair to establish the identity of the user connecting to the Secure Shell (SSH) server.

After the connection has been setup and the encryption and integrity algorithms have been negotiated, the client is authenticated. Public key authentication requires the client user to be configured with an RSA or DSA private and public key pairs. The private key is often called *identity keys*. The private key is only available to the client user. The public key is shared with the SSH server and is also called the *authorized key*. The public and private key pair used for the SSH protocol are also called *SSH user keys* or *SSH keys*.

The possession of the private key serves as the basis for authentication. For public key authentication, the SSH client software generates a signature using the private key and the server validates the signature using the public key that is associated with the client. The session is created if the signature is valid but the SSH user may be subject to other authentications.

Most common terms that are used with SSH public key authentication include:

SSH Server

The host running the SSH service. A client user can establish a connection to the SSH server and operate as a user of the SSH server host.

SSH Client

The host from which a client user tries to establish the connection with the SSH server.

SSH Client User or SSH User

The principal (user or host) trying to establish a connection with the SSH server.

SSH User Key Pair or SSH Key Pair

Public and private keys of the SSH user used to setup the SSH connection. Only RSA algorithm with key sizes of 2048, 3072, and 4096 bits is supported with Oracle Key Vault 21.7.

SSH User Keys or SSH Keys

Public or private keys of the SSH client user used to setup the SSH connection.

OpenSSH Implementation of the SSH Protocol

OpenSSH is an implementation of the Secure Shell (SSH) protocol that is available with Oracle Linux and in general with Linux and Unix-based operating systems.

Secure Shell Daemon application (`sshd`) is the OpenSSH service running on SSH servers listening for incoming SSH connection requests. `sshd` typically runs as the `root` user. `ssh` is the OpenSSH client software used to initiate SSH connections. Both the client and server implementations have their own configuration files.

OpenSSH does support public key authentication. It requires the creation of an SSH user key pair. The private key is on the SSH client host and should be secured. The public key is typically added to the `authorized_keys` file of a host user on the SSH server. The client connects as that host user on the SSH server. Connection as any other host user will be denied unless the client's public key also exists in the `authorized_keys` file of that host user. The authorized keys file is typically stored as `~/.ssh/authorized_keys` file. The public key is in the SSH format in the `authorized_keys` file.

Figure 15-2 Open SSH Setup



SSH user keys of the SSH client *John* are created by *John* or by an administrator. The keys are stored on the SSH client host. *John's* public key is added to the `authorized_keys` file of the *oracle* user on SSH server *phoenix*. *John* can now initiate connections as *oracle* to SSH server *phoenix*. However, connections as other users like `opc` on *phoenix* is denied since *John's* public key does not exist in the `authorized_keys` file of `opc`.

Once a client logs in as *oracle* to *phoenix*, the client may be able to change the `authorized_keys` file thereby granting access to other clients, or revoking access from existing clients.

Challenges with SSH Public Key Authentication

Following sections describe the challenges faced with SSH Public Key Authentication.

- **Limited Decentralized Access Control**
The typical public key authentication setup involves the Secure Shell (SSH) server administrator or the SSH server host user adding the public key to the `authorized_keys` file of the SSH server host user.

- **Client Key Management**
Since Secure Shell (SSH) is the tool of choice for accessing remote hosts for operation and administrative purposes, hundreds of users in the organization use SSH.
- **Governance**
With SSH keys scattered all over the organization, enforcing uniform enterprise-wide key-management policies and ensuring that key-management best practices are followed becomes difficult.
- **No Reporting**
With the distributed access to Secure Shell (SSH) servers, there are no records of who (SSH client user) has access to what (SSH server). There is no enterprise-wide view of the SSH configuration.

Limited Decentralized Access Control

The typical public key authentication setup involves the Secure Shell (SSH) server administrator or the SSH server host user adding the public key to the *authorized_keys* file of the SSH server host user.

The Secure Shell (SSH) server host user typically has access to its own *authorized_keys* file. Authorizing a user to login to an SSH server as the SSH server host user is controlled:

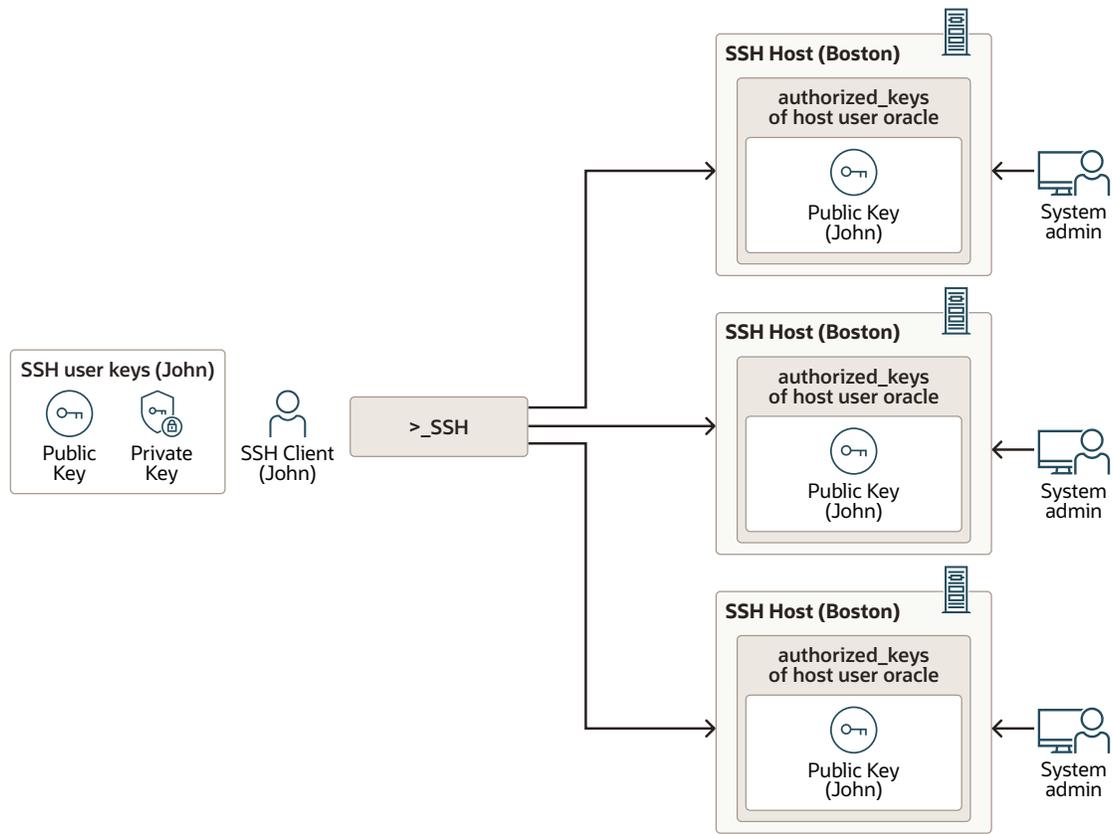
1. on the SSH server itself
2. by the SSH server host users or SSH server administrators.

Since the authorization control lies with multiple actors on the SSH servers, there is *no consistent and uniform access control*.

This problem is further exacerbated because potentially more than one SSH client user can become the SSH server host user. After an SSH client user successfully connects to the SSH server as a SSH server host user, this user can then add public keys of other SSH client users to the *authorized_keys* file. Such an uncontrolled access grant by the client user operating as SSH server host user essentially bypasses the administrator access authorization which had been intended to grant authorization to only a specific SSH client user.

Additionally, since the authorization control is done on each SSH server, SSH administrators of all these SSH servers are involved which leads to human errors. Three different administrators may have to add the keys to the three different SSH servers to grant a single user access to all three of them. This is not ideal as it introduces a time skew between the time when authorization is done to the time it takes effect on the SSH servers. Furthermore, the decentralized approach is time consuming, requires coordination, and is still error prone.

Figure 15-3 localized-SSH-server-access



Finally, the decentralized nature of the access control architecture has its own share of security risks. The public keys of users that no longer need access to SSH servers must be removed from those SSH servers in a timely manner. However, with decentralized access control the revoke process may take time. Another example is where the SSH server host user grants access to the SSH server out of band. There will be no record of this authorization and no awareness about it, thus increasing the security risks.

In an enterprise setting, multiple administrators grant or revoke access to hundreds of users on thousands of hosts within their domain. At this scale, the decentralized nature of the access control architecture quickly becomes overwhelming, making the system generally more prone to errors and thus more exposed to security risks.

Client Key Management

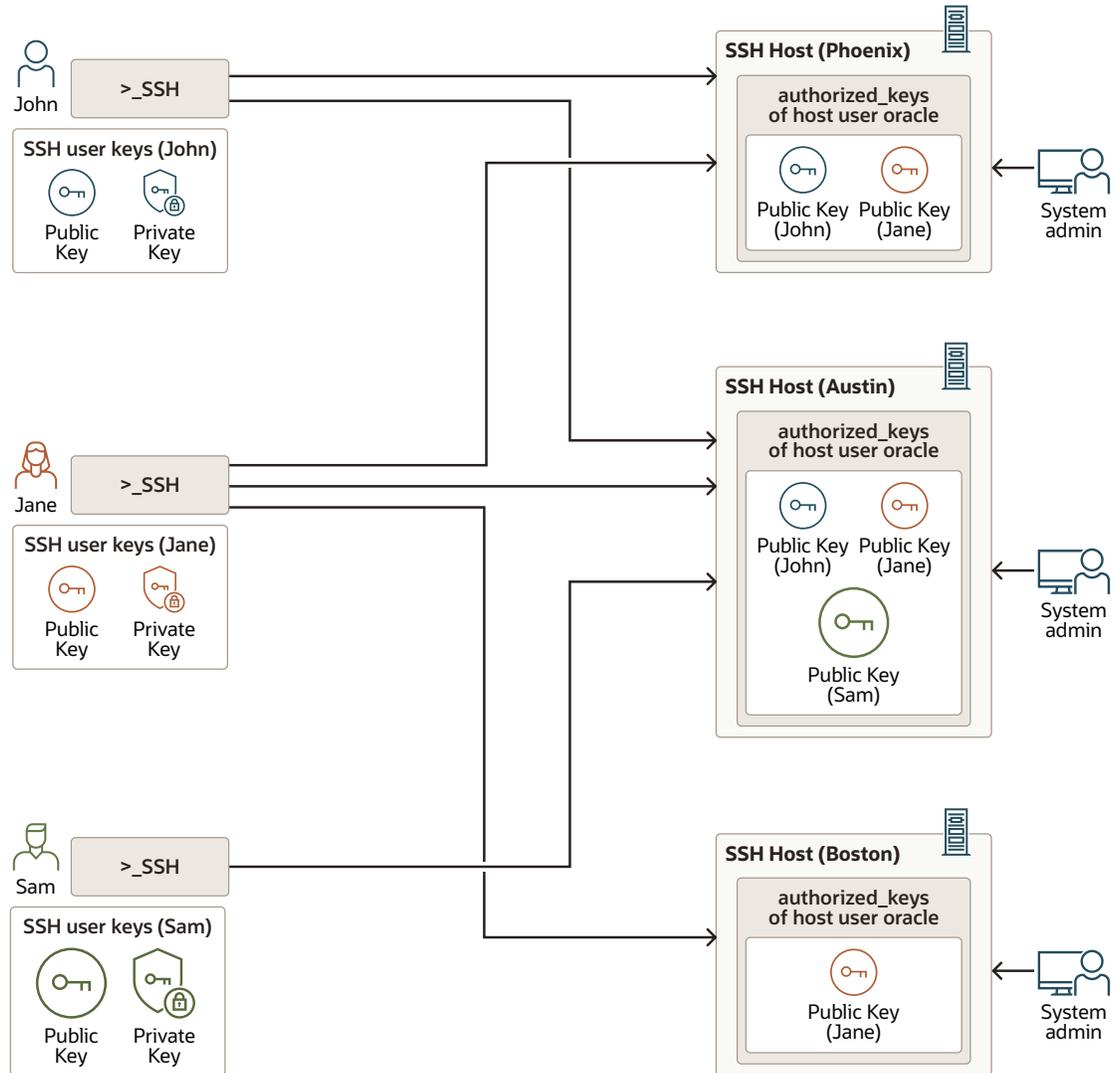
Since Secure Shell (SSH) is the tool of choice for accessing remote hosts for operation and administrative purposes, hundreds of users in the organization use SSH.

Moreover, for better security, public key authentication is often the preferred method of user authentication. Secure Shell (SSH) is also the tool of choice for server-to-server communication and operations. Hundreds of SSH client users, users, and servers alike, generate their own SSH key pairs. Every SSH client users has to manage the SSH key pairs. Consequently there is so much *key sprawl* that there are bound to be security weaknesses for any organization-wide deployment.

Every SSH client users has to secure their private keys against compromise. Every SSH client users has to do life cycle management of their SSH keys. SSH key pairs should be rotated at

periodic intervals. This is made more complicated because the SSH keys do not come with an expiry date. Every SSH client users needs to ensure that the old key pairs are removed from the SSH servers and SSH client hosts. Enforcing proper key life cycle management is a daunting task when the SSH keys are spread across the enterprise.

Figure 15-4 Limited-SSH-User-Key-Management



Governance

With SSH keys scattered all over the organization, enforcing uniform enterprise-wide key-management policies and ensuring that key-management best practices are followed becomes difficult.

Secure Shell (SSH) client users can choose to create keys of different sizes. They can also choose to use different SSH keys to access different SSH servers which in some cases is desirable but not always. It is very difficult to enforce policies granting temporary access to an SSH server or removing access from SSH server machines temporarily, say, if an employee is on vacation. Without centralized control, relying on every SSH client user to enforce governance is error prone and not likely to be feasible at scale.

No Reporting

With the distributed access to Secure Shell (SSH) servers, there are no records of who (SSH client user) has access to what (SSH server). There is no enterprise-wide view of the SSH configuration.

Records are essential when revoking or granting access to SSH client users and auditing the authorizations within the enterprise.

When a distributed approach to SSH key management is used, there is no centralized inventory of the SSH keys and it is difficult to identify whether SSH client users are using redundant keys, multiple keys and if the old keys have been purged. Without the complete visibility of their SSH key inventory, organizations cannot identify and remove weak links, and enforce uniform policies across enterprise.

Controlling Access to SSH Server Centrally with Oracle Key Vault

Oracle Key Vault can centrally manage the access to Secure Shell (SSH) servers. Centralized access control improves security, enables quicker responses to threats, reduces human error, and simplifies authorization management at scale.

The configuration file of the OpenSSH daemon `sshd_config` includes the `AuthorizedKeysCommand` keyword. This keyword specifies a program that can be used to lookup the public key of the SSH client users. The configuration can further supply this program with the server host user to log in as the SSH server host (or the SSH server host user) and the public key of the SSH client user. This program should emit the authorized public keys in SSH format to standard output. The `AuthorizedKeysCommand` keyword applies to public key authentication only. Oracle Key Vault ships with a script (`okv_ssh_ep_lookup_authorized_keys.sh`) located at `$OKV_HOME/bin`. The script is used to look up the public key in Oracle Key Vault.

SSH Server Host User

This is the Operating System user on the SSH server that the SSH client user uses to log in as the SSH server host.

SSH Server Endpoint

With release 21.7, Oracle Key Vault has a new endpoint type, the *SSH Server*. The *SSH Server* endpoint type must be deployed on the *SSH Server*. *SSH Server* endpoint software includes the script to lookup the authorized public keys in Oracle Key Vault server. The `AuthorizedKeysCommand` keyword of the SSH daemon should be configured to invoke this `okv_ssh_ep_lookup_authorized_keys.sh` script. SSH Server endpoints have the SSH server hostname associated with them. See, [Supported Platforms for SSH Server and Client Endpoints](#) for platforms on which you can deploy SSH server endpoint.

SSH Server Wallet

Starting with Oracle Key Vault release 21.7, you can create wallets of different types:

- *SSH Server* wallet
 - This wallet has the SSH server host user associated with it.
 - It only contains public keys that are authorized to access SSH server as a SSH server host user
- *General* wallet

- This type covers all existing wallets
- It can store TDE keys, secrets, opaque objects, etc

The wallet to be used with SSH Server endpoint should be of type *SSH Server*. SSH server wallets can contain only public keys and only SSH Server endpoints can be granted access to SSH Server wallets. SSH Server wallets have the SSH server host user associated with them.

SSH Server Mapping File

SSH Server endpoints must set up the SSH server mapping file. This is the configuration file for the `okv_ssh_ep_lookup_authorized_keys.sh` script. This file contains the mapping of the SSH server host user to its corresponding SSH Server wallet. For each SSH server host user there must be a corresponding SSH wallet set up in Oracle Key Vault. The mapping helps identify whether a given public key in the SSH wallet can be used to authorize the SSH client user to operate as the SSH server host user on SSH server.

```
# Configuration file for Oracle Key Vault SSH server endpoints
# [ <SSH server host user> ]
# ssh_server_wallet= <SSH wallet>

[ opc ]
ssh_server_wallet=opc_ssh_wallet

[ oracle ]
ssh_server_wallet=oracle_ssh_wallet
```

SSH Server Endpoint lookup Script

OpenSSH daemon should be run as the *root* user and SSH endpoint should be deployed as the *root* user. OpenSSH daemon must be configured to use the `AuthorizedKeysCommand` keyword to run Oracle Key Vault's `okv_ssh_ep_lookup_authorized_keys.sh` script. SSH endpoint uses the `okv_ssh_ep_lookup_authorized_keys.sh` script to setup a secure connection to Oracle Key Vault and validate that the endpoint has access to the public key. For this purpose, OpenSSH daemon is configured to supply the script with the SSH server host user and the public key of the SSH client user. The script uses the SSH server host user to lookup the corresponding SSH wallet mapping stored in SSH server mapping file. Once the SSH wallet has been identified from the mapping file, the script locates it in Oracle Key Vault, opens it, and looks up the public key of the SSH client user. If the public key is accessible to the SSH endpoint, then the SSH client user is authorized to proceed further in the SSH authentication. If the public key presented by the SSH client user is not available in mapped SSH wallet, the SSH client software retries the operation using other public keys, if any, of the SSH user. If no matching public key is found in the SSH wallet, then the connection is terminated.

- [Deployment](#)
Secure Shell (SSH) wallets are the means to manage the authorization of an SSH server host user on SSH server centrally.

Deployment

Secure Shell (SSH) wallets are the means to manage the authorization of an SSH server host user on SSH server centrally.

As long as the public key of SSH client user is in the SSH server wallet of SSH server host user, the client is authorized to access the SSH server as that SSH server host user. An administrator can add or remove the SSH client user's public key to/from the SSH wallet to grant or revoke the SSH client user's access to an SSH server. SSH endpoints identify the

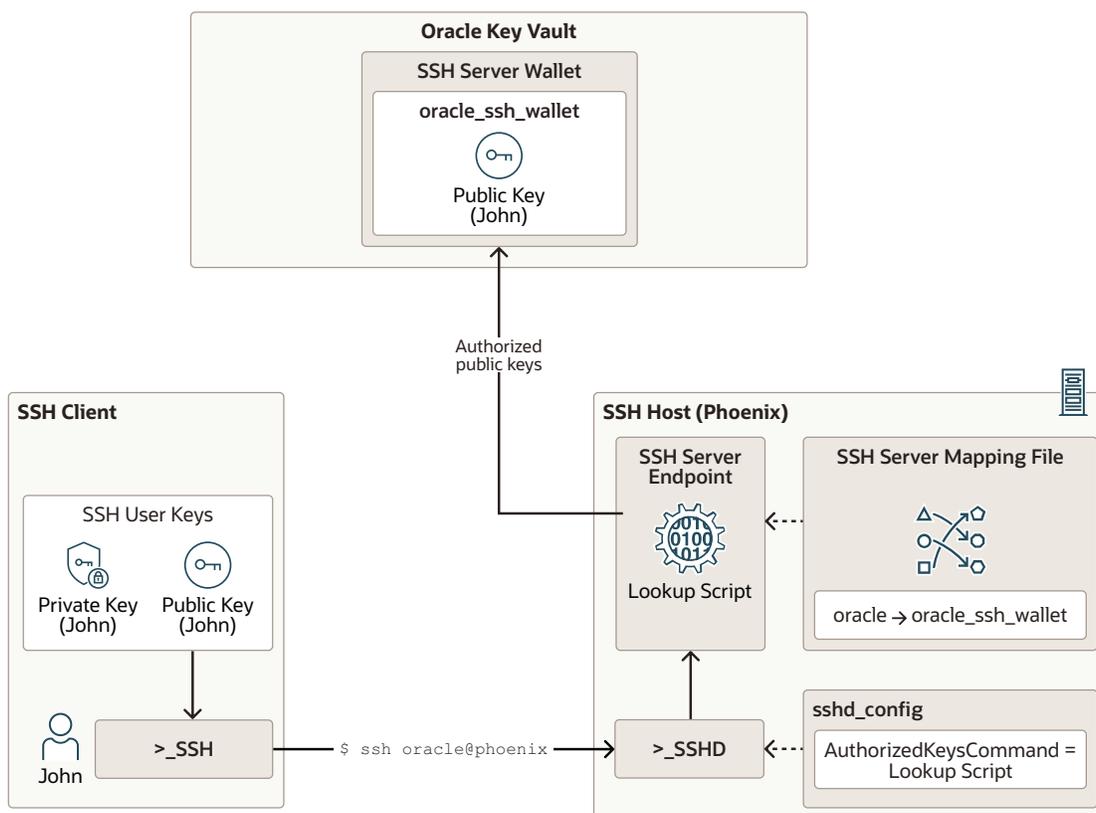
SSH server to which the SSH client user is granted access. Since the SSH client users can request access to more than one SSH server host user, an SSH endpoint may require access to more than one SSH wallet.

Figure displays a typical setup where SSH server access is centrally managed in Oracle Key Vault.

The SSH client user, *John* is trying to login to SSH server, *phoenix*, as SSH server host user *oracle*. *phoenix* is setup with an SSH endpoint and the OpenSSH daemon is setup to use the lookup script (*okv_ssh_ep_lookup_authorized_keys*) to check the public keys in Oracle Key Vault. For the OS user *oracle* on *phoenix*, the SSH server mapping file maps *phoenix* OS user *oracle* to *oracle_ssh_wallet* in Oracle Key Vault. The SSH wallet (*oracle_ssh_wallet*) is created in Oracle Key Vault and the SSH endpoint on *phoenix* is granted read access to *oracle_ssh_wallet*.

As *John* tries to log in to *phoenix* as user *oracle*, the SSH daemon hands off the public key received from *John* along with the user *oracle*, that *John* is trying to log in as to the lookup script. The lookup script checks the mapping of *oracle* to an SSH wallet, *oracle_ssh_wallet* in this case. The lookup script next checks if *John's* public key exists in *oracle_ssh_wallet*. If not, the clients retries with the next public key, if any. If none of the public keys exist in the *oracle_ssh_wallet*, the client connection is denied, otherwise the SSH user authentication continues further. Whether *John* can eventually login to *phoenix* as *oracle* depends on whether *John* has the corresponding private key and how the SSH user authentication is setup on *phoenix*.

Figure 15-5 Centralized-SSH-Server-Access



SSH wallets, SSH endpoints and the SSH server host user to SSH wallet mapping must be setup to manage the authorization of an SSH server as SSH server host user centrally. The

SSH daemon should be deployed as the root user. SSH endpoints are deployed as root user. *AuthorizedKeysCommand* keyword-argument in `sshd_config` file must be updated to point to the SSH server endpoint look up script. For each SSH server host user that the SSH client users log in as, there must be a mapping entry in the SSH server mapping file. Multiple SSH servers can share the same SSH server wallet to manage the authorization of the corresponding SSH host user on these servers. This implies that the same set of SSH client users are authorized to log in as that host user on any of these SSH servers.

SSH endpoint working in tandem with the OpenSSH daemon verifies if the incoming connection is authorized. The authorization depends on whether the public key presented by the SSH client exists in the SSH wallet. The public key of one or more SSH client users can be added or removed from the SSH wallets of host users of multiple SSH servers centrally in Oracle Key Vault, enabling centralized access of SSH servers.

- [Limiting Access Control to Root User](#)
Even though the authorization of an SSH server is managed centrally, the OpenSSH daemon configuration could still allow SSH client users to login into the SSH server.

Limiting Access Control to Root User

Even though the authorization of an SSH server is managed centrally, the OpenSSH daemon configuration could still allow SSH client users to login into the SSH server.

Even though the SSH client user's initial login into an SSH server is controlled centrally in Oracle Key Vault, once the the client user is logged into the SSH server as a host user, this user can then add the public keys of other SSH client users into the authorized keys file of the host user, thus bypassing the central access control in Oracle Key Vault. For this reason, the SSH daemon should be configured to disallow the support of the *authorized_keys* file for regular SSH server host users, thereby enforcing central access management. If SSH is the primary means of SSH server host administration, it is recommended to have *authorized_keys* file as a fall back option for the *root* user so that the system can be recovered in case of emergencies.

Managing SSH User Keys with Oracle Key Vault

Oracle Key Vault can centrally manage the Secure Shell (SSH) key pairs of the SSH client users.

Centralized key management not only brings structure and organization to SSH keys in one repository but also improves security by enforcing key governance, key life cycle management and key operations like rotation and revocation.

- [Deployment](#)
Oracle Key Vault integrates with OpenSSH to support the SSH authentication while using the SSH user keys from Oracle Key Vault. Oracle Key Vault provides the PKCS#11 library that the OpenSSH client can leverage to perform SSH authentication steps using the SSH keys stored in Oracle Key Vault.

Deployment

Oracle Key Vault integrates with OpenSSH to support the SSH authentication while using the SSH user keys from Oracle Key Vault. Oracle Key Vault provides the PKCS#11 library that the OpenSSH client can leverage to perform SSH authentication steps using the SSH keys stored in Oracle Key Vault.

Any Oracle Key Vault endpoint can upload the SSH key pairs to Oracle Key Vault. For better management and access control, the SSH keys pairs should be uploaded to wallets. Starting

with Oracle Key Vault 21.7, you can also create the SSH keys in Oracle Key Vault. All Oracle Key Vault endpoint types include Oracle Key Vault's PKCS#11 library. As such any Oracle Key Vault endpoint can function as an SSH client user. See, *Creating Keys in Managing Oracle Key Vault Virtual Wallets and Security Objects*.

The SSH client software that Oracle Key Vault can work with are limited to OpenSSH. Deploy the Oracle Key Vault endpoint on the SSH client. The endpoints PKCS#11 library is loaded by the OpenSSH client program. The PKCS#11 library communicates with Oracle Key Vault and run the SSH operations requested by the SSH client program. The operations are performed using the SSH keys stored in Oracle Key Vault.

With SSH keys in Oracle Key Vault, the SSH client user sets up the connection using the following command:

For UNIX Platforms:

```
ssh -I $OKV_HOME/lib/liborapkcs.so <ssh_server_user>@<ssh_server_host>
```

For Windows Platforms:

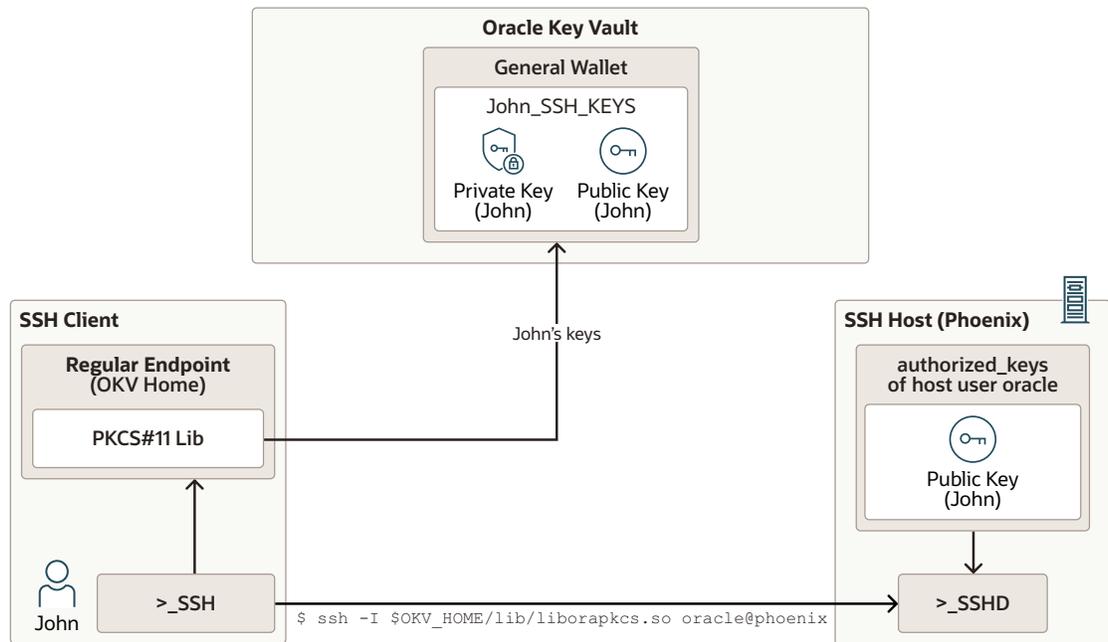
```
ssh -I $OKV_HOME/lib/libsshpkcs.dll <ssh_server_user>@<ssh_server_host>
```

Oracle Key Vault's PKCS#11 library is loaded by the OpenSSH client software using the `-I` option. Other relevant OpenSSH client software can always be used. Users may also want to make use of the `ssh-agent` to cache the connection pin. The connection pin is **NULL** if the endpoint is deployed with auto-login setup (Oracle Linux 8), else the connection pin is the endpoint password used when deploying endpoint (Oracle Linux 9).

It is recommended to keep the **extractable** attribute of the SSH private key set to `FALSE` so that the SSH private key never leaves the boundary of the Oracle Key Vault cluster. An endpoint is deployed on the SSH client host. The SSH client endpoint should have the read and modify privileges on this wallet. The manage wallet privilege is required if the SSH client user needs to register SSH keys. The SSH endpoint also includes the PKCS#11 library which is used with the OpenSSH client software when the SSH client user needs to setup a connection to the Oracle Key Vault server.

The *Centralized-SSH-Key-Management* figure displays the setup for managing SSH keys in Oracle Key Vault. Users should create a general wallet per SSH client user to store and manage the SSH keys.

Figure 15-6 Centralized-SSH-User-Key-Management



With the SSH keys registered or created in Oracle Key Vault, the key management of these keys becomes simpler. Deactivation date can be associated with these keys and the customer is notified of when the keys are expiring and need to be rotated. SSH keys could be rotated periodically. SSH keys could be revoked quickly in case they are compromised. Moreover, when the SSH keys of SSH client users are registered in Oracle Key Vault, it is easy to identify and weed out redundant and unused keys, thus streamlining the organization and management of SSH user keys. Finally, administrators can subject the SSH keys to key life cycle management.

Managing the SSH keys in Oracle Key Vault simplifies their organization, administration and enforces key management best practices.

Oracle Key Vault and SSH Integration

While Oracle Key Vault can manage the Secure Shell (SSH) keys and can also control the access to SSH servers independently, ideally both the SSH keys and access control of SSH servers should be managed in Oracle Key Vault centrally.

- [About Oracle Key Vault and SSH integration](#)
Oracle Key Vault centrally manages both the Secure Shell (SSH) private keys and SSH public keys. Managing the public keys enables access control of the SSH servers in the enterprise and managing the private keys enforces key governance, key management and organization for SSH keys across the enterprise.
- [SSH Admin Managing the SSH User Keys and Access to SSH Servers](#)
In this model, the Secure Shell (SSH) admin is responsible for the SSH key and access to SSH server management for the entire enterprise. SSH client users in a manner subscribe to the keys managed by the SSH admin.

- [SSH Client Users Manage SSH Keys and SSH Admin Manage Access to SSH Servers](#)
In this model, the Secure Shell (SSH) admin is responsible only for controlling the access to SSH servers for the entire enterprise. The SSH client users own and manage the SSH user keys themselves. The SSH client users own and manage the SSH user keys.

About Oracle Key Vault and SSH integration

Oracle Key Vault centrally manages both the Secure Shell (SSH) private keys and SSH public keys. Managing the public keys enables access control of the SSH servers in the enterprise and managing the private keys enforces key governance, key management and organization for SSH keys across the enterprise.

You can perform both, SSH key setup and access control of SSH server setup as described in SSH Public Key Authentication and Controlling Access to SSH Server Centrally with Oracle Key Vault.

The *Centralized-SSH-Key-Management* figure illustrates the SSH key and access control management, *John* initiates the connection to phoenix as SSH server host user Oracle.

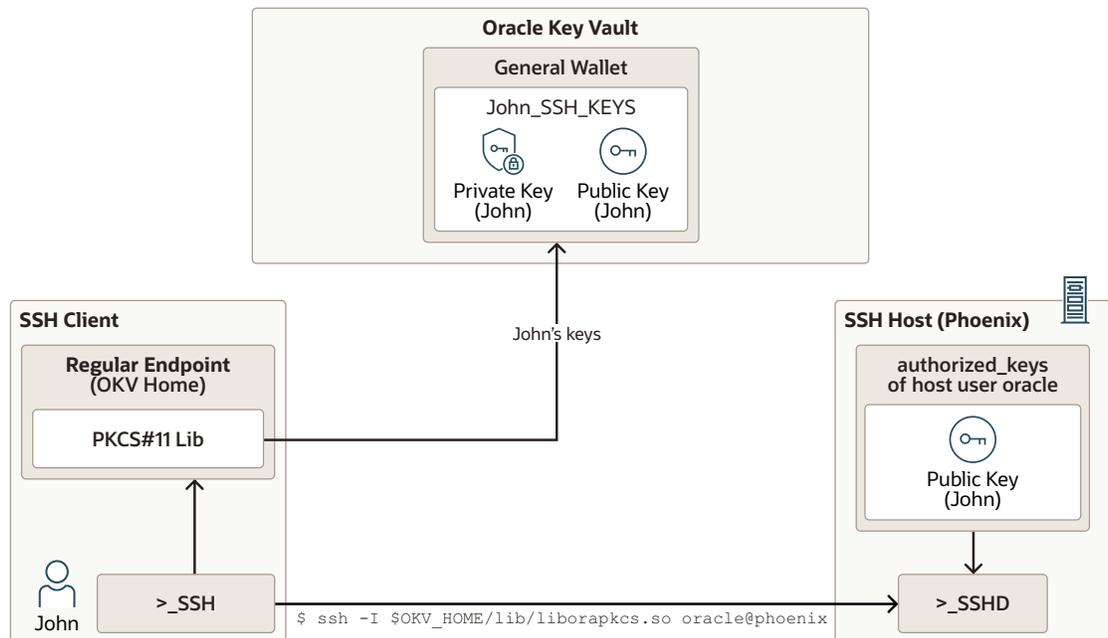
John supplies the SSH client endpoint's PKCS#11 library. The PKCS#11 library fetches all the public keys that can be used to setup the connection. The keys are tried one by one until the connection is established. On the SSH server, Oracle Key Vault's endpoint lookup script checks for the corresponding public key in SSH Server wallet on the SSH server for authorization. If authorized, the SSH authentication process continues using the matched/authorized public key otherwise the connection attempt is blocked.



Note:

There is no footprint of the private and public keys on the SSH client or SSH server.

Figure 15-7 Centralized-SSH-Key-Management



An added advantage for SSH key and access control management is that the rotation can be done centrally. Revoke not only prevents the use of the compromised key on SSH clients but you can use it to block access to the SSH server as well.

You can use one or both of these measures to suspend SSH client user access of SSH servers. This helps when the SSH client user is not accessing the system for short durations like vacation, and you want to prevent unintentional, accidental, or malicious access of the SSH servers using the SSH client user's keys.

SSH Admin

SSH admin is an Oracle Key Vault user that owns the SSH endpoints and SSH wallets and may or may not be managing the SSH user keys. Typically, the SSH admin would also control the access to one or more SSH server or the *SSH domain* of the SSH admin.

There are two models of managing SSH user keys and access control to SSH servers. One where the SSH admin manages both. The other where the SSH client users manage their own keys and the SSH admin control access to SSH servers.

SSH Admin Managing the SSH User Keys and Access to SSH Servers

In this model, the Secure Shell (SSH) admin is responsible for the SSH key and access to SSH server management for the entire enterprise. SSH client users in a manner subscribe to the keys managed by the SSH admin.

The SSH admin has permissions to read, modify and manage the SSH server wallets and the general wallets that store the SSH keys of the SSH client users. SSH admin would typically create these wallets and grant necessary privileges to the SSH client users and SSH server endpoints.

SSH admin would also create and own the SSH server endpoints. And SSH admin would also be responsible for deploying the SSH server endpoint software on the SSH servers and modifying the OpenSSH daemon configuration.

SSH User Key Management

SSH admin creates the SSH key pair or registers an SSH key pair in Oracle Key Vault. The keys are created with certain key sizes depending on the organizational policies. The SSH admin can choose to associate a deactivation date with the SSH key pair. A notification is sent to the administrators when the key is expiring so that they can rotate the keys.

The SSH admin creates a general wallet for the SSH client user and grants the read access of the wallet to the endpoint used by the SSH client user. The SSH admin adds the newly created or registered SSH key pair into the SSH client user's general wallet. No other user or endpoint need to have access to this SSH key pair, specifically the SSH private key.

Access Control

SSH admin adds the SSH client user's public key to the SSH wallets of the SSH servers to grant SSH client user access to those SSH servers. SSH admin can remove the SSH client user's public key from the SSH wallet to revoke access of the SSH client user on SSH servers.

SSH admin can also remove the SSH client user's access from the SSH private key to prevent the SSH client user from accessing any SSH servers at all in the SSH domain. For instance, when an employee quits the organization and the access needs to be revoked from all SSH endpoints immediately.

SSH Client Users Manage SSH Keys and SSH Admin Manage Access to SSH Servers

In this model, the Secure Shell (SSH) admin is responsible only for controlling the access to SSH servers for the entire enterprise. The SSH client users own and manage the SSH user keys themselves. The SSH client users own and manage the SSH user keys.

The SSH client user creates two wallets: A general wallet holding the SSH public and private key pair of the SSH client user and an SSH Server wallet that just hold the SSH public key. Only the SSH client user has access to the first wallet. SSH admin has read access to the second wallet, the SSH client user's public key wallet.

The SSH client user creates or registers the SSH key pair in Oracle Key Vault and stores the key pair in the general wallet. Only the SSH public keys are stored in the SSH client user's public key wallet. The SSH client user owns and manages the SSH client user endpoint also. The SSH key management in this model is the responsibility of the SSH client user.

The SSH admin creates and owns the SSH server endpoints and the SSH server wallets. The SSH admins are responsible for deploying the SSH server endpoint software on the SSH servers and modifying the OpenSSH daemon configuration.

SSH User Key Management

The SSH client user creates the SSH key pair or registers an SSH key pair in Oracle Key Vault. You should create the keys with certain key sizes depending on the organizational policies. The SSH client users can chose to associate a deactivation date with the SSH key pair. A notification is sent to the SSH user and to the Oracle Key Vault key administrators before the key expires. The SSH key admins can also inform the SSH user that the keys are expiring in case the SSH user has not set up email alerts. The SSH client user rotates the SSH keys. The new SSH public key should be authorized by SSH admin in the same manner as the previous SSH public key.

Only the SSH client user has access to the SSH public and private keys. The SSH admins cannot deny the SSH client user access to the SSH keys.

Access Control

The SSH admin adds the SSH client user's public key from the SSH client user's public key wallet to the SSH wallets of the SSH servers to grant the SSH client user access to those SSH servers. The SSH admin can remove the SSH client user's public key from the SSH wallet to revoke access of the SSH client user on SSH servers.

Supported Platforms for SSH Server and Client Endpoints

Oracle supports 64-bit platforms for SSH server and client endpoints.

Oracle Key Vault SSH server endpoint, used to centrally manage access to SSH servers, can be deployed on the following platforms:

- Oracle Linux (7, 8)
- Linux for Arm (aarch64)
- IBM: Linux on System z
- Oracle Solaris SPARC 11.4
- Oracle Solaris X64 11.4

- HP-UX (IA) 11.31
- AIX 7.3



Note:

SSH server endpoints are not supported on Windows. The SSH server must use OpenSSH 8.1p1 or later.

Oracle Key Vault endpoints used to manage SSH user keys are supported on the following platforms:

- Oracle Linux (7, 8)
- Linux for Arm (aarch64)
- IBM: Linux on System z
- Oracle Solaris SPARC 11.4
- Oracle Solaris X64 11.4
- HP-UX (IA) 11.31
- AIX 7.3
- Windows



Note:

OpenSSH deployed on these systems must be at least version 7.2p1 for endpoints that do not require a password and at version 8.1p1 for endpoints that do require a password.

16

Management of SSH Keys - Setup and Configuration

You can use Oracle Key Vault to centrally manage Secure Shell (SSH) private and public keys and control access to SSH servers.

- [Setup SSH Admin](#)
An Secure Shell (SSH) administrator is an Oracle Key Vault user who should be able to manage access to SSH servers for SSH users centrally in Oracle Key Vault.
- [Controlling Access to SSH Server Centrally with Oracle Key Vault](#)
You can centrally manage which SSH users have access to which SSH servers.
- [Managing SSH User Keys with Oracle Key Vault](#)
You can manage the Secure Shell (SSH) keys of the SSH users centrally in Oracle Key Vault and enforce key management and governance.
- [Oracle Key Vault and SSH Integration](#)
Users can manage both the SSH user's keys and the access to SSH servers with Oracle Key Vault. This setup is ideal since it not only provides the ability to enumerate the SSH user's keys and the SSH server access using those keys, but it also allows for transparent rotation of the SSH user's keys. SSH user's private key can also be marked not extractable which means the private key would never leave the boundary of the Oracle Key Vault deployment.
- [Migrating Existing SSH Deployments to Oracle Key Vault](#)
Administrators can migrate their SSH deployments for SSH user key management or managing access control of SSH servers or both.
- [Guidelines for OpenSSH SSHD Configuration](#)
When you use Oracle Key Vault to securely manage SSH user keys or control access to SSH servers, the time to establish an SSH connection depends on multiple factors.
- [Reports](#)
Oracle Key Vault offers several reports that you can use to check Secure Shell (SSH) user keys, SSH server access control configuration and also to track the SSH user activity.

Setup SSH Admin

An Secure Shell (SSH) administrator is an Oracle Key Vault user who should be able to manage access to SSH servers for SSH users centrally in Oracle Key Vault.

An SSH admin manages the SSH Server wallets and SSH Server endpoints needed for the controlling access to SSH servers from Oracle Key Vault. An SSH admin may or may not manage the SSH user keys.

To setup an SSH administrator, you must:

1. Create an Oracle Key Vault user.
2. Grant the Create Endpoint privilege to the user. This user can then manage any endpoints that it creates.

3. Grant the Manage Endpoint privilege to the user. This may be needed for those endpoint that are created by another user.

This section details the steps required to set up an SSH administrator.

Create a Regular Oracle Key Vault User

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar. The Manage Users page appears with a list of existing users.
3. In the Manage Users page, click **Create**. The Create User page appears.
4. Provide the details for the required fields on the Create User page. For more information see, [Creating an Oracle Key Vault User Account](#).
5. Click **Save**.
6. In a multi-master cluster, the user is initially created in the **PENDING** state. After cluster naming conflict resolution for the new user is completed, the user is transitioned to the **ACTIVE** state.

Grant the Create Endpoint Privilege

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Users who have the Create Endpoint privilege cannot grant it to other users.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar. The Manage Users page appears displaying the list of users.
3. Select the user to whom you want to grant the **Create Endpoint** privilege.
4. Under User Details, select the **Create Endpoint** check box.
5. Click **Save**.

 **Note:**

When a local Oracle Key Vault user with the Create Endpoint privilege creates an endpoint, Oracle Key Vault grants the Manage Endpoint privilege on that endpoint to the local user.

Grant the Manage Endpoint Privilege

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Users who have the Manage Endpoint privilege cannot grant it to other users.
2. Select the **Users** tab, and then **Manage Users** in the left navigation bar. The Manage Users page appears displaying the list of users.
3. Select the user to whom you want to grant the Manage Endpoint privilege.
4. In the **Access on Endpoints** area, click Add.
5. In the **Add Endpoint Access to User** page, under Select Endpoint, select the endpoint to which you want to grant the user the Manage Endpoint privilege.
6. Click **Save**.

Repeat the preceding steps to grant the Manage Endpoint privilege on all other endpoints that the SSH admin needs to manage.

Controlling Access to SSH Server Centrally with Oracle Key Vault

You can centrally manage which SSH users have access to which SSH servers.

- [About Controlling Access to SSH Server Centrally with Oracle Key Vault](#)
As an Secure Shell (SSH) administrator, you manage one or more SSH servers within an organization. With Oracle Key Vault, you can centrally manage the SSH users access for SSH Servers.
- [Setup in Oracle Key Vault Server](#)
As the SSH admin user, perform the steps given in this section for configuring the SSH Server endpoint and SSH Server wallets in Oracle Key Vault.
- [Setup on SSH Server](#)
Perform the steps given in this section to deploy and configure the SSH Server endpoint and modify the OpenSSH configuration on the SSH server.
- [Manage SSH Server Access from Oracle Key Vault](#)
After completing the configuration steps in the Oracle Key Vault and the SSH Server, you can now control which users can connect to the SSH server using SSH public key authentication.

About Controlling Access to SSH Server Centrally with Oracle Key Vault

As an Secure Shell (SSH) administrator, you manage one or more SSH servers within an organization. With Oracle Key Vault, you can centrally manage the SSH users access for SSH Servers.

To prepare an SSH server for the centralized access management in Oracle Key Vault, you must identify the SSH server host users you plan to allow the SSH users to log in as. It is recommended that you modify the OpenSSH daemon configuration to disallow the use of `authorized_keys` file to prevent access grants outside of Oracle Key Vault.

To centrally manage the SSH access of an SSH server, you need to:

1. Register an SSH Server endpoint.
2. Setup an SSH Server wallet for each SSH server host user that you plan to allow to log in as to the SSH server.
3. Grant the SSH Server endpoint read-only access on the SSH server wallets.
4. Provision the SSH Server endpoint on the SSH server.
5. Setup the OpenSSH SSH daemon to use the Oracle Key Vault SSH Server endpoint software.
6. Add the public keys of the SSH client users to the SSH Server wallets.

Note:

To authorize an SSH client user to connect to the SSH server as an SSH server host user, you add the user's public key in the SSH Server wallet corresponding to the SSH server host user.

Setup in Oracle Key Vault Server

As the SSH admin user, perform the steps given in this section for configuring the SSH Server endpoint and SSH Server wallets in Oracle Key Vault.

Register an SSH Server endpoint

You need to setup an SSH Server endpoint on the SSH server host. The OpenSSH daemon configuration is modified to use the SSH Server endpoint script to validate whether the key offered by the SSH client is among the authorized keys for the host user.

1. Log in to the Oracle Key Vault management console as an SSH admin user. The SSH admin user must have the privilege to create an endpoint.
2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar. The **Endpoints** page appears listing all the Oracle Key Vault endpoints.
3. On the Endpoints page, click **Add**.
4. The **Register Endpoint** page appears.
5. From the **Type** drop-down list, select the type of endpoint as **SSH Server**.
6. Selecting the endpoint type as SSH Server displays another field **SSH Server Hostname**. You add the hostname or IP address of the SSH server where this endpoint will be deployed.
7. Provide the details of the required field in the Register Endpoint page. See [Adding an Endpoint as an Oracle Key Vault System Administrator or Create Endpoint User](#). The following are the supported platforms on which SSH Server endpoints can be deployed:
 - Oracle Linux (7, 8)
 - Linux for Arm (aarch64)
 - IBM: Linux on System z
 - Oracle Solaris SPARC 11.4
 - Oracle Solaris X64 11.4
 - HP-UX (IA) 11.31
 - AIX 7.3



Note:

SSH Server endpoints are not supported on Windows.

8. Click **Register** to create the new endpoint.

Once the endpoint is registered successfully, **Endpoints** page appears. Copy and save the **Enrollment Token** value for the new endpoint. You will need it to download the endpoint software and then enroll the endpoint on the SSH server.

The screenshot shows the 'Register Endpoint' form with the following details:

- Endpoint Name ***: PHOENIX_SSH_SERVER
- Type ***: SSH Server
- SSH Server Hostname ***: phoenix.devcenter.com
- Platform ***: Linux
- Description**: SSH server endpoint on phoenix.
- Administrator Email**: (empty)
- Cluster Subgroup ***: No Cluster Subgroup

Create SSH Server wallets

You need to setup an SSH Server wallet for each SSH server host user that you plan to allow SSH client users to log in as to the SSH server.

1. Log in to the Oracle Key Vault management console as an SSH admin user.
2. Select the **Keys & Wallets** tab, and then select **Wallets** from the left navigation bar.
3. In the Wallets area, click **Create**. The **Create Wallet** page appears.
4. For the **Wallet Type**, select **SSH Server**.
5. Selecting the **Wallet Type** as **SSH Server** displays another field **SSH Server Host User**. Enter the name of the SSH server host user for whom this wallet will control SSH access. SSH client users whose public keys are added to this wallet can access the SSH server as this host user.
6. Provide the details of the required fields in the **Create Wallet** page. See, [Creating a Virtual Wallet](#).
7. Click **Save** to create the new wallet.

The screenshot shows the 'Create Wallet' form with the following details:

- Name ***: oracle_ssh_wallet
- Description**: SSH server wallet for host user 'oracle' on SSH server 'phoenix', 'austin'.
- Wallet Type**: SSH Server (selected)
- SSH Server Host User ***: oracle

 **Note:**

You can use an SSH Server wallet to manage the authorization for an SSH server host user on the multiple SSH servers if the same set of the SSH client users need to log in as that host user on all of these SSH servers. Such shared use of SSH Server wallets may further simplify the SSH access management at scale.

Grant SSH Server endpoint Read Only access on the SSH Server wallets

Grant the SSH Server endpoint Read Only access to each SSH Server wallets. This access would allow the SSH Server endpoint to validate if the public key offered by the ssh client is present in the SSH Server wallet of the host user that the client user is connecting as.

1. Log in to the Oracle Key Vault management console as an SSH admin user.
2. Select the **Endpoints** tab, and then select **Endpoints** from the left navigation bar.
3. Endpoints page is displayed. Find the SSH Server endpoint, and then click the endpoint name. **Endpoint Details** page is displayed.
4. In the **Access to Wallets** area, click **Add**. **Add Access To Endpoint** page is displayed.
5. In the **Select Wallet** area, select the SSH Server wallets on which you want to grant access to.
6. In the **Select Access Level** area, select **Read Only**.
7. Click **Save**.

Related Topics

- [Adding an Endpoint as an Oracle Key Vault System Administrator or Create Endpoint User](#)
A user who has been granted the System Administrator role or the Create Endpoint privilege can add an endpoint by using the **Endpoints** tab.
- [Creating a Virtual Wallet](#)
You can create a virtual wallet and add security objects to it at the same time.
- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.

Setup on SSH Server

Perform the steps given in this section to deploy and configure the SSH Server endpoint and modify the OpenSSH configuration on the SSH server.

Because the OpenSSH daemon is typically run as the root user, ensure that the SSH Server endpoint is deployed as the root user on the SSH server.

If the SSH admin user does not have the *root* access on the SSH server, then the SSH admin must coordinate the execution of following configuration steps with an administrator who has the root access of the SSH server. This document assumes that the SSH admin has the root access on the SSH server.

Provision the SSH Server endpoint on the SSH server using the following steps:

Enroll the Endpoint and Download the Software

You can enroll the endpoint using the enrollment token and then download the endpoint software.

You can download the endpoint software from the SSH server where you want to deploy it, or you can download the software from another host and then copy over the downloaded endpoint software (`okvclient.jar`) to the SSH server. Ensure that you protect the integrity of the downloaded software by applying stricter file permissions.

After registering the endpoint, the Oracle Key Vault system administrator sends this endpoint's enrollment token to the endpoint administrator by email or other out-of-band method.

1. Log in to the SSH server as the root user.
2. Connect to the Oracle Key Vault management console as an SSH admin user
For example: `https://192.0.2.254`.
The login page to the Oracle Key Vault management console appears. **Do not log in.**
3. Click the **Endpoint Enrollment and Software Download** immediately after the **Login** button. The **Enroll Endpoint & Download Software** page appears.
4. In the **Enrollment Token** field, enter the SSH Server endpoint's enrollment token, and click **Submit Token**. You saved the enrollment token during the endpoint registration step.
 - a. If the token is valid, then a valid token message appears after the **Enrollment Token** field. The Endpoint Type, Endpoint Platform, Email and Description fields are automatically populated with the values that were entered during endpoint registration. If the token is invalid, then an invalid token message appears. Check the token and retry the download procedure.
 - b. Click **Enroll** at the top right corner of the page.
5. In the directory window that appears, follow the prompt to save the `okvclient.jar` endpoint software file.
You must navigate to the directory where you want to save the file.
6. Save the file to a secure directory with appropriate permissions in place so that it cannot be read or copied by others.
7. Verify that the file has been downloaded.

 **Note:**

If you did not download the file to the endpoint system, you must use an out-of-band method to copy the file to that system and install it there. At this stage, you are ready to install the Oracle Key Vault `okvclient.jar` software file on the SSH server.

Install the SSH Server endpoint on the SSH Server

You install the SSH Server endpoint on the SSH server using the downloaded endpoint software file `okvclient.jar`.

You install the endpoint as the root user on the SSH server because the OpenSSH daemon is typical run as the root user.

Before you install the SSH Server endpoint, ensure the following:

- That you have JDK 1.5 or later installed, and that the PATH environment variable includes the java executable (in the `JAVA_HOME/bin` directory). Oracle Key Vault supports JDK versions 1.5, 1.6, 7, and 8. The 64-bit version of Java is required.
- That the version of OpenSSH used on the system is version 8 or later (with support for the `AuthorizedKeysCommand` keyword).

To install the SSH Server endpoint onto the SSH server host:

1. Ensure that you are logged in to the SSH server host as the root user.
2. Navigate to a new SSH Server endpoint directory on the server where you want to deploy the endpoint software.
3. Move the `okvclient.jar` file from its saved location to this new SSH endpoint directory.
4. After confirming this directory is empty other than the `okvclient.jar` file, run the `java` command as the root user to install the `okvclient.jar` file. For example,

```
java -jar okvclient.jar -d /<path_to_SSH_EP_directory>/<SSH_EP_directory> -v
```

In this specification:

- `-d` specifies the directory location for the endpoint software and configuration files, in this case `/<path_to_SSH_EP_directory>/<SSH_EP_directory>`.
 - `-v` writes the installation logs to the `/<path_to_SSH_EP_directory>/<SSH_EP_directory>/log/okvutil.deploy.log` file at the server endpoint.
5. When you are prompted for the password, enter:
 - `<enter>` for the auto-login endpoint or
 - a valid password for the password-protected endpoint.
 - If you want to create a password-protected endpoint, at minimum enter a password between 8 and 30 characters and then press Enter. For better security, Oracle recommends that you include uppercase letters, lowercase characters, special characters, and numbers in the password. Use of following special characters is recommended: period (`.`), comma (`,`), underscore (`_`), plus sign (`+`), colon (`:`), and exclamation mark (`!`).

```
Enter new Key Vault endpoint password (<enter> for auto-login):  
Key_Vault_endpoint_password  
Confirm new endpoint password: Key_Vault_endpoint_password
```

 **Note:**

A password-protected wallet is an Oracle wallet file that stores the endpoint's credentials to access Oracle Key Vault. This password will be required whenever the endpoint connects to Oracle Key Vault.

Additional configuration for Password-protected endpoint

If you installed the SSH Server endpoint as the auto-login endpoint, skip these steps.

For the password-protected endpoint, additional configuration is required to automate the supplying of the endpoint password to the SSH Server endpoint's authorized keys lookup script. The OpenSSH daemon executes this script when authenticating the SSH client connections and this script must be able to connect to the Oracle Key Vault without asking for the endpoint password.

To automate the supplying of the endpoint password, complete the following configuration:

- Create a script or an executable which when executed returns the endpoint password in cleartext on the standard output.

- Change directory to the endpoint's directory.
- Open the script `bin/okv_ssh_ep_lookup_authorized_keys` for editing.
- `SSH_SERVER_ENDPOINT_GET_PASSWORD_SCRIPT="<full-path-of-the-password-script>"`

Configure SSH Server Host User to SSH Server Wallet mapping

You must setup the SSH server mapping file which contains the SSH Server host user to SSH Server wallet mapping. This is the configuration file for the `okv_ssh_ep_lookup_authorized_keys` script. For each SSH server host user there must be a corresponding SSH Server wallet setup in Oracle Key Vault. When an SSH client connection is received, the `okv_ssh_ep_lookup_authorized_keys` script first looks up the SSH Server wallet corresponding to the host user of the incoming SSH connection. The `okv_ssh_ep_lookup_authorized_keys` then verifies whether the public key offered by the client is present in mapped SSH Server wallet.

Update the SSH Server endpoint configuration file

```
/<path_to_SSH_EP_directory>/<SSH_EP_directory>/conf/okvsshendpoint.conf
```

to map the SSH server host users with their corresponding SSH Server wallets.

For example, following configuration file shows the mapping for the host OS users `oracle`, `opc`, and `root` on the SSH Server *phoenix* to their mapped SSH Server wallets `oracle_ssh_wallet`, `opc_ssh_wallet`, and `root_ssh_wallet_for_phoenix` respectively.

```
# Configuration file for Oracle Key Vault SSH server endpoints
[ oracle ]
ssh_server_wallet=oracle_ssh_wallet

[ opc ]
ssh_server_wallet=opc_ssh_wallet

[ root ]
ssh_server_wallet=root_ssh_wallet_for_phoenix
```

Configure OpenSSH daemon to integrate with Oracle Key Vault

Configure the OpenSSH SSH daemon to use the Oracle Key Vault SSH endpoint software. On the SSH Server update the `AuthorizedKeysCommand` keyword of the OpenSSH daemon configuration file `sshd_config` and set it to invoke the SSH Server endpoint lookup script `okv_ssh_ep_lookup_authorized_keys`.

1. On the SSH server host, stop the SSH daemon using commands specific to the platform on which it runs. For example, on Oracle Linux 7, stop the SSH daemon as follows:

```
/usr/bin/systemctl stop sshd
```

2. Edit the SSH server host's `sshd_config` file to use SSH Server endpoint lookup script by updating the following lines:

```
#AuthorizedKeysCommand none

#AuthorizedKeysCommandUser nobody

AuthorizedKeysCommand
    /<path_to_SSH_EP_directory>/<SSH_EP_directory>/bin/
okv_ssh_ep_lookup_authorized_keys
    get_authorized_keys_for_user %u %f %k

AuthorizedKeysCommandUser root
```

Ensure that you specify the arguments to the `okv_ssh_ep_lookup_authorized_keys` script exactly as shown in the preceding steps, including their relative order.

3. Restart the SSH daemon on the SSH server host using commands specific to the platform on which the SSH server host runs. For example, on Oracle Linux 7, restart the SSH daemon using the `systemctl` command.

```
/usr/bin/systemctl restart sshd
```

 **Note:**

You should keep a root user session open on the SSH server until you have validated that the ssh users can connect to the SSH server.

Manage SSH Server Access from Oracle Key Vault

After completing the configuration steps in the Oracle Key Vault and the SSH Server, you can now control which users can connect to the SSH server using SSH public key authentication.

To authorize a ssh client user to connect as an SSH server host user on an SSH server, you add the ssh client user's public key to the SSH Server wallet that is configured to hold the authorized public keys for the the host user on the SSH server.

If the SSH user's public key is not yet available in Oracle Key Vault, then first upload the SSH public key using the Oracle Key Vault RESTful services utility. See Oracle Key Vault RESTful Services Administrator's Guide for the details on configuring and using the RESTful services utility.

If the public key that you are uploading is in the SSH-format, as is the case with the public keys in the host user's `authorized_keys` file, then first convert the public key to the PKCS8 format using the `ssh-keygen` utility as shown in the following example:

```
$ ssh-keygen -e -m PKCS8 -f john-pubkey-ssh.pub > john-pubkey-pkcs8.pub
```

Verify that the public key is in the PKCS8 format.

```
$ cat opc-pubkey-pkcs8.pub
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArbHYwXcOapqD6Xv4B8VX
8Ce8Ii1ZBU12iUtYcl1K/179IxD+ViD6B+yBYO+Yp0J5aXvFSwMyztEfQdSn7GmG
ASUVSXGmqQKr0skWxkVsIyxzGC7zxBXcISkmb3kkZfMCwLsj+hzpEQfKTFItMGaZ
eTfC0CxtSprP3HQopn796KMsxvsuubTGayy05pZQXCScryTIR1Mcwa/quKdFy1Vj
t3VI/nzLAdMTGYN9MAOnkt1hmD2jE+dBBQ/excoHl+WoZLjek0wNcyifFRxeOM64
d7LDv90G5pjQctQk+73gkqPhBewFlMh2Ql068i2Jg0ygbceK6qoUqxRnFsUkBDXa
/QIDAQAB
-----END PUBLIC KEY-----
```

You upload the SSH public key in the PKCS8 format to the Oracle Key Vault using the RESTful services utility:

```
okv managed-object public-key register --output_format text --object john-
pubkey-pkcs8.pub --algorithm RSA --length 2048 -mask "ENCRYPT"
--ssh-user "John" --name "SSH public key of John" --activation-date now --
deactivation-date "2025-10-15 00:00:00"
```

The preceding command returns the unique identifier of the new SSH public key in Oracle Key Vault.

Next, you add the SSH public key to the SSH Server wallet using the following steps:

1. Log in to the Oracle Key Vault management console as an SSH admin user.
2. Select the **Keys & Wallets** tab, then **Keys & Secrets** in the left navigation bar. The **Keys & Secrets** page appears.
3. In the search bar, enter the unique identifier of the SSH public key created in the preceding step. Click **Go**.
4. For the key found, click **Edit**. The Object Details page appears.
5. In the Wallet Membership area, click **Add Wallet Membership**.
6. In the Add Wallet Membership dialog box, select the SSH Server wallet where you want to add this SSH public key and click **Add**.
You have now authorized the SSH user to connect to the SSH server as the SSH server host user associated with the SSH server wallet.

Implementation considerations:

- If the user's SSH public key is already present in the Oracle Key Vault, then do not upload the key again. As an SSH admin, you can add the same public key to different SSH Server wallets to grant the user access to different SSH servers and as different host users.
- If the SSH keys are created and managed by SSH users themselves, then the SSH users must add their public key to a separate wallet and grant the access of that wallet to the SSH admin. This way, the SSH admin can access the user's public key from that wallet and then add it to the required SSH Server wallets.
- If the SSH keys are created and managed by an SSH admin, then the SSH admin already has access to the SSH user's public key. The SSH admin can add the user's public key to the required SSH Server wallets without needing another intermediate wallet to hold the user's public key.

Related Topics

- [okvutil upload Command](#)
The `okvutil upload` command uploads security objects to Oracle Key Vault.
- [Managing SSH User Keys with Oracle Key Vault](#)
You can manage the Secure Shell (SSH) keys of the SSH users centrally in Oracle Key Vault and enforce key management and governance.

Managing SSH User Keys with Oracle Key Vault

You can manage the Secure Shell (SSH) keys of the SSH users centrally in Oracle Key Vault and enforce key management and governance.

- [About Managing the SSH User Keys with Oracle Key Vault](#)
You can manage the SSH keys of the SSH users centrally in Oracle Key Vault and enforce key management and governance.
- [Oracle Key Vault Server Setup](#)
Register an endpoint for SSH user.
- [SSH Client Host Setup](#)
You install the SSH user's endpoint on the SSH client host from where the SSH user would connect to SSH servers.
- [Authorize SSH User's Public Key on SSH Server](#)
Before an SSH user can connect to an SSH server using SSH public key authentication, user's public key must be authorized on the SSH servers that user needs to connect to.
- [Connect to SSH Servers using Oracle Key Vault PKCS#11 library](#)
After completing the configuration steps in the Oracle Key Vault and the SSH client host, the SSH user can now initiate the SSH connections using OpenSSH client and Oracle Key Vault PKCS#11 library.

About Managing the SSH User Keys with Oracle Key Vault

You can manage the SSH keys of the SSH users centrally in Oracle Key Vault and enforce key management and governance.

The SSH public key authentication is often the preferred method of user authentication within an enterprise. Hundreds of SSH client, users and servers alike, generate and manage their own SSH key pairs. Not all users are familiar with the best practices around key management. These poorly managed SSH keys are distributed across enterprise leading to a "key sprawl" that presents as a security weakness for any organization-wide deployment.

The Oracle Key Vault PKCS#11 library supports SSH public key authentication using a SSH key pair that is stored in Oracle Key Vault. You can centrally manage the SSH keys of the enterprise users in Oracle Key Vault to simplify key life-cycle management, enable key governance and uniformly enforce policies. The SSH users connect to the SSH servers using their keys directly from Oracle Key Vault.

There are two models to manage the SSH users keys.

- The SSH admin is responsible for the SSH user key management.
In this deployment model, a central team of administrators manages the complete life cycle of the SSH users keys. The SSH users are only given the authorization to the use their SSH keys, but they do not have the ownership rights on their keys. In this model, the SSH users cannot share their private keys with other users if the SSH admin set the private keys as non-extractable.

In this model, the SSH users need not be an Oracle Key Vault user.

- The SSH users are responsible for the management of their own keys. In this deployment model, the SSH users create and manage their own SSH keys. They retain the full ownership of their SSH user keys and are responsible for managing the complete life cycle.

In this model, each SSH user must have their own user account in Oracle Key Vault.

Depending upon the model you choose to manage the SSH keys - by the SSH admins or the SSH users, the following steps described may vary slightly. This document assumes that the SSH user key management is done by the SSH administrators.

To centrally manage the SSH user keys in Oracle Key Vault, as an SSH admin you would need to:

1. Register an endpoint for the SSH user and deploy it on the SSH client host.
2. Create a general wallet with exclusive access to the endpoint created previously.
3. Create or Register the SSH key pair and it to the general wallet created previously.
 - You can use the UI or RESTful utility to create an SSH key pair.
 - Or you can register an existing SSH key pair using the RESTful utility.
4. Download the public key and share it with the administrator of the SSH server.

SSH server admin adds the keys to relevant `authorized_keys` files.

Oracle Key Vault Server Setup

Register an endpoint for SSH user.

Register an endpoint for SSH user

You register an endpoint to use with the OpenSSH client. You can also use an existing endpoint, such as an Oracle Database endpoint, with the OpenSSH client.

1. Log in to the Oracle Key Vault management console as an SSH admin user or a user with the System administrator role or the privilege to create an endpoint.
2. Select the **Endpoints** tab, then **Endpoints** in the left navigation bar. The **Endpoints** page appears listing all the Oracle Key Vault endpoints.
3. On the Endpoints page, click **Add**.
4. The **Register Endpoint** page appears.
5. From the **Type** drop-down list, select the endpoint type. You can select any endpoint type, as appropriate, except for the **SSH Server** as they are meant to be deployed on the SSH Servers.
6. Provide the details of the required field in the Register Endpoint page. See, [Adding an Endpoint as an Oracle Key Vault System Administrator or Create Endpoint User](#).
7. Click **Register**. Once the endpoint is registered successfully, **Endpoints** page appears. Copy and save the **Enrollment Token** value for the new endpoint. You will need it to download the endpoint software and then enroll the endpoint on the SSH client host.

Create a General Wallet for SSH user keys

You create a general wallet to store the SSH keys for a SSH user. You will store the SSH public and private keys of an SSH user in this wallet.

1. Log in to the Oracle Key Vault management console as an SSH admin user.
2. Select the **Keys & Wallets** tab, and then select **Wallets** from the left navigation bar.
3. In the Wallets area, click **Create**. The Create Wallet page appears.
4. For the **Wallet Type**, select **General**.
5. Provide the details of the required fields in the Create Wallet page. See, [Creating a Virtual Wallet](#) .
6. Click **Save** to create the new wallet.

Create or Register an SSH key pair

You create or register an SSH key pair and store them in the general wallet created previously.

You can create an SSH key pair using the Oracle Key Vault management console or RESTful services utility. When creating the key from the Oracle key Vault management console, you can specify the general wallet at the time of key creation itself.

You can register the existing SSH keys in the Oracle Key Vault using the RESTful services utility only. After registering, you move them to the general wallet.

Use the following steps to create an SSH key pair using the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as an SSH admin.
2. Select the **Keys & Wallets** tab, and then select **Keys & Secrets** from the left navigation bar.
3. In the Keys & Secrets page, click **Create**.
4. In the page that appears, select **SSH Key Pair** from the **Application Keys** section. The **Create SSH Key Pair** page appears.

Create SSH Key Pair Cancel Create

Create a new SSH key pair for an SSH user and add it to the wallet of the SSH user endpoint.
You can authorize the SSH user to access an SSH server by adding the SSH user's public key to the SSH server wallet.

SSH User * ⓘ

Cryptographic Algorithm * ▼

Cryptographic Length * ▼

Private Key Extractable * ▼

Date of Activation ⓘ

Date of Deactivation ⓘ

Wallet Membership ⓘ Select Wallet

5. Complete the information on SSH Key Pair page:
 - **SSH User:** Enter the name of SSH client user who will be using these SSH keys. This user need not be an Oracle Key Vault user.
 - **Cryptographic Algorithm:** Supported algorithm is **RSA**.

- **Cryptographic Length:** Key length. Keys of three lengths are supported: 2048, 3072, 4096.
- **Private Key Extractable:** Should the private key of the SSH key pair be extractable from the Oracle Key Vault server. The default and recommended value is False.
- **Date of Activation:** The time from when the SSH key pair should become active. The default value is the current system time.
- **Date of Deactivation:** The time from when the SSH key pair should become deactivated i.e. expired. You should rotate the SSH keys before they expire. The default value is 2 years from the current system time.
- **Wallet Membership:** The wallet where you want to store the new SSH keys. You must have the Manage Wallet permission on the wallet. This should be set to the general wallet created in the preceding steps.
- Click **Create** to create the new SSH key pair.

Grant the SSH user's endpoint read only access on the general wallet for the SSH user

You grant the SSH user's endpoint read only access to the general wallet for the SSH user. This allows the endpoint to use the SSH keys from the general wallet when user attempts an SSH connection using SSH public key authentication.

1. Log in to the Oracle Key Vault management console as an SSH admin user.
2. Select the **Keys and Wallets** tab, and then select **Wallets** from the left navigation bar.
3. **Wallets** page is displayed. Find the SSH user's general wallet and click on the pencil icon. The **Wallet Overview** page appears.
4. In the Wallet Access Settings area, click Add.
In the Select Endpoint/User Group area, select type Endpoints from the drop-down list.
5. Select the checkbox for the SSH user's endpoint that you want to grant access to.
6. In the **Select Access Level** area, select **Read Only**.
7. Click **Save**

Note:

If you are using the deployment model where SSH users create and manage their own SSH keys, many of these steps described previously are done by SSH users themselves. An SSH user must have an Oracle Key Vault user account in order to manage their own SSH keys in Oracle Key Vault.

In this model, previous steps would change as the following:

1. Register an endpoint for SSH user: No change. You continue to run this step as an SSH admin.
2. Create a general wallet for SSH user keys: You now run this step as an SSH user.
3. Create or register an SSH key pair: You now run this step as an SSH user.
4. Grant the SSH user's endpoint read only access on the general wallet for the SSH user: You now run this step as an SSH user.

SSH Client Host Setup

You install the SSH user's endpoint on the SSH client host from where the SSH user would connect to SSH servers.

To perform the SSH public key authentication using the SSH keys from the Oracle Key Vault, you need an Oracle Key Vault endpoint on the SSH client host. The SSH user uses the OpenSSH client with the endpoint's Oracle Key Vault PKCS#11 library to establish SSH connections.

You may choose to install the endpoint in a network folder such as under the user's home directory. This would allow the user to make SSH connections using this endpoint from multiple hosts.

The SSH user's endpoint provides access to the private keys of the SSH user. It is recommended that you install the endpoint as password-protected. The permissions on the endpoint must be restricted to the SSH user. Generally, SSH users would install their endpoints themselves.

In certain deployments, such as when SSH users are applications or machines and not the human users, you may use existing endpoints. For example, you may use an existing Oracle Database endpoint to make SSH connections as the SSH client user 'oracle'. In such setup, you would grant this endpoint the Read Only access to the wallet that holds the keys for the application or machine user.

Use these steps to provision the SSH user's endpoint on the SSH client host.

Enroll the Endpoint and Download the Software

You enroll the endpoint using the enrollment token and then download the endpoint software.

You can download the endpoint software from the SSH client host where you want to deploy it, or you can download the software from another host and then copy over the downloaded endpoint software (`okvclient.jar`) to the SSH server. Ensure that you protect the integrity of the downloaded software by applying stricter file permissions.

1. Log in to the SSH client host as the SSH user
2. Connect to the Oracle Key Vault management console as an SSH admin user or a user with the System administrator role or the privilege to create an endpoint.
For example: <https://192.0.2.254>.
The login page to the Oracle Key Vault management console appears. Do not log in.
3. Click the **Endpoint Enrollment and Software Download** immediately after the **Login** button. The Enroll Endpoint & Download Software page appears.
4. In the Enrollment Token field, enter the SSH Server endpoint's enrollment token, and click **Submit Token**. You saved the enrollment token during the endpoint registration step.
 - a. If the token is valid, then a valid token message appears after the Enrollment Token field. The Endpoint Type, Endpoint Platform, Email, and Description fields are automatically populated with the values that were entered during endpoint registration. If the token is invalid, then an invalid token message appears. Check the token and retry the download procedure.
 - b. Click **Enroll** at the top right corner of the page.
 - c. Click the **Endpoint Enrollment and Software Download** immediately after the **Login** button. The Enroll Endpoint & Download Software page appears.

- d. In the Enrollment Token field, enter the SSH Server endpoint's enrollment token, and click **Submit Token**. You saved the enrollment token during the endpoint registration step.
 - i. If the token is valid, then a valid token message appears after the Enrollment Token field. The Endpoint Type, Endpoint Platform, Email, and Description fields are automatically populated with the values that were entered during endpoint registration.
If the token is invalid, then an invalid token message appears. Check the token and retry the download procedure.
 - ii. Click **Enroll** at the top right corner of the page.
- e. In the directory window that appears, follow the prompt to save the `okvclient.jar` endpoint software file.
You must navigate to the directory where you want to save the file.
- f. Save the file to a secure directory with appropriate permissions in place so that it cannot be read or copied by others.
- g. Verify that the file has been downloaded.

 **Note:**

If you did not download the file to the SSH client host, you must use an out-of-band method to copy the file to that system and install it there.

At this stage, you are ready to install the Oracle Key Vault `okvclient.jar` software file on the SSH client host.

Install the SSH user's endpoint on the SSH client host

As the SSH user, you install the SSH user's endpoint on the SSH client host using the downloaded endpoint software file `okvclient.jar`.

Before you install the SSH user's endpoint, ensure that you have JDK 1.5 or later installed, and that the `PATH` environment variable includes the `java` executable (in the `JAVA_HOME/bin` directory). Oracle Key Vault supports JDK versions 1.5, 1.6, 7, and 8. The 64-bit version of Java is required.

To install the SSH user's endpoint onto the SSH client host:

1. Ensure that you are logged in to the SSH user.
2. Navigate to a new directory on the SSH client host where you want to deploy the endpoint software.
3. Move the `okvclient.jar` file from its saved location to this new endpoint directory.
4. After confirming this directory is empty other than the `okvclient.jar` file, run the `java` command as the root user to install the `okvclient.jar` file.
For example,

```
java -jar okvclient.jar -d /<path_to_EP_directory>/<EP_directory> -v
```

In this specification:

- `-d` specifies the directory location for the endpoint software and configuration files, in this case `/<path_to_EP_directory>/<EP_directory>`.

- -v writes the installation logs to the `/<path_to_EP_directory>/<EP_directory>/log/okvutil.deploy.log` file at the server endpoint.
5. When you are prompted for the password, enter:
- `<enter>` for the auto-login endpoint or
 - a valid password for the password-protected endpoint
If you want to create a password-protected endpoint, at minimum enter a password between 8 and 30 characters and then press Enter. For better security, Oracle recommends that you include uppercase letters, lowercase characters, special characters, and numbers in the password. Use of following special characters is recommended: period (`.`), comma (`,`), underscore (`_`), plus sign (`+`), colon (`:`), and exclamation mark (`!`).

```
Enter new Key Vault endpoint password (<enter> for auto-login):
Key_Vault_endpoint_password
Confirm new endpoint password: Key_Vault_endpoint_password
```

A password-protected wallet is an Oracle wallet file that store the endpoint's credentials to access Oracle Key Vault. This password will be required whenever the endpoint connects to Oracle Key Vault. It is recommended that you install the endpoint as password-protected as the endpoint has the access to the SSH user's private keys.

Authorize SSH User's Public Key on SSH Server

Before an SSH user can connect to an SSH server using SSH public key authentication, user's public key must be authorized on the SSH servers that user needs to connect to.

If you are using the Oracle Key Vault for the SSH user key management only, then the user's public key needs to added to the `authorized_keys` files on different SSH servers individually. You need to coordinate this step with the administrator of respective SSH servers. As an SSH admin, you download the public key, convert it to the SSH format and share it with the SSH server administrator. The SSH administrator adds the public key to the `authorized_keys` file of the relevant host user on the SSH server.

Use the following steps to download the SSH public key in SSH format:

1. As an SSH admin, ensure that you have an endpoint setup to use with the RESTful services utility. See, [Running Oracle Key Vault RESTful Services Utility Commands](#).
2. Get the Unique Identifier of the SSH public key using these steps:
 - a. Log in to the Oracle Key Vault management console as an SSH admin.
 - b. Select the **Reports** tab, and then select **Reports** from the left navigation bar.
 - c. Expand on the SSH Reports and click on the **SSH Key Metadata Report**.
 - d. Search and identify the SSH public key of the new SSH user. Click **Key Details**. The Object Details page appears.
 - e. Copy the value of the Unique Identifier.
3. Use the RESTful services utility command `'okv managed-object public-key get'` with the Unique ID of the public key to download the SSH public key.

```
./bin/okv managed-object public-key get --output_format TEXT --uuid
FAB07067-03F5-4F01-BF3F-67CEDCDE0307
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAgarY8fRQlhpbtawGyVSC
```

```
LsPrQN4RLWcnzzEo+QIcqRc4/rpVbTs7aEx63MYtheGDQ5V2NPPBYwrZOrb09ETw
QvkXqZBD+xvAVRc8t8c6wTCvGvMgAkoDzBA7nD7n10WBd4PVnrkDbAdELQx4DLNW
lke/XhBeQOOCKT2yA31CVDDnMKZE1wcKy3G6J3jd02e3jV5Daj5r8h7RN0f6a9HJO
8Xmaa61Vj9duTrlSKvLmG1gAwZ2Jzkat6+PgTsKE+3n8Evy+E9Y0tvyieXJG/c0V
GZ3Uwx8HUKqkw5h6nQkCVEjNWabFHDVkgkn1aq9axg1oYJMhPTJszGSmqbfNlo6N
ZQIDAQAB
-----END PUBLIC KEY-----
```

Redirect the standard output of the previous command to a file, say `public_key.pub`. The downloaded public key is in the PKCS#8 format.

4. Use the `sshkey-gen` utility to convert the downloaded public key from the PKCS#8 format to the SSH format that the `authorized_keys` file accepts.

```
ssh-keygen -i -f public_key.pub -m PKCS8
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACpqtjx9FCWGlulrAbJVIIuw+tA3hEtZyfPMSj5AhyFzj
+ulVtOztoTHrcxi2F4YNDlXY088FjCtk6tvT0RPBC+RepkEP7G8BVFzy3xZRBMK8a8yACSGPMED
ucPueXRYF3g9WeuQNSb0QtDHgMs1aWR79eEF5A4IpPbIDfUJUM0cwpkSXBwrLcboneN07Z7eNXk
NqPmvyHtE3R/pr0ck7xeZprVWP125OvVIq8uYbWADbnYnORq3r4+B0woT7efwS/L4T1g62/KJ5
ckb9zRUZndTDHwdSSqTDmHqdCQJUSMlZpsUcNWQaSfVqr1rGDWhgkwc9MmzMKapt82Wjoll
```

Redirect the standard output of the earlier command to a file. This file now contains the SSH user's public key in the SSH format.

5. Share the user's public key with the administrators of SSH servers. These admins will add the public key to the appropriate `authorized_keys` file.

Related Topics

-

Connect to SSH Servers using Oracle Key Vault PKCS#11 library

After completing the configuration steps in the Oracle Key Vault and the SSH client host, the SSH user can now initiate the SSH connections using OpenSSH client and Oracle Key Vault PKCS#11 library.

Ensure that OpenSSH client version is 8.1 or higher.

To initiate SSH connections from the SSH client host to the SSH server host:

1. As an SSH user, login to the SSH client host.
2. Set `OKV_HOME` environment variable to the SSH user endpoint's installation directory.
3. Connect as follows:

For UNIX Platforms:

```
ssh -I $OKV_HOME/lib/liborapkcs.so <ssh_server_user>@<ssh_server_host>
```

Example:

```
ssh -I $OKV_HOME/lib/liborapkcs.so oracle@phoenix
```

For Windows Platforms:

```
ssh -I $OKV_HOME/lib/libsshpkcs.dll <ssh_server_user>@<ssh_server_host>
```

Example:

```
ssh -I $OKV_HOME/lib/libsshpkcs.dll oracle@phoenix
```

4. The preceding command asks for the endpoint's password with the prompt:

```
Enter pin for 'label':
```

For the auto-login endpoint, enter NULL. Otherwise, enter the endpoint's password and press enter.

5. SSH session is now established.

 **Note:**

Use ssh-agent to avoid supplying the PKCS#11 library path and the pin repeatedly. After ssh agent is setup and the pin for the endpoint's PKCS#11 library has been added, you can then execute the ssh command as usual. For example:

```
ssh oracle@phoenix
```

Oracle Key Vault and SSH Integration

Users can manage both the SSH user's keys and the access to SSH servers with Oracle Key Vault. This setup is ideal since it not only provides the ability to enumerate the SSH user's keys and the SSH server access using those keys, but it also allows for transparent rotation of the SSH user's keys. SSH user's private key can also be marked not extractable which means the private key would never leave the boundary of the Oracle Key Vault deployment.

- [SSH Administrators Managing both the SSH User's Keys and the SSH Server Access](#)
As an SSH administrator, you can setup the SSH Server wallet, the SSH Server endpoint, and the OpenSSH configuration on the SSH server as described in the section *Controlling access to SSH server centrally with Oracle Key Vault*.
- [SSH Client Managing SSH User Keys and SSH Admin Managing the SSH Server Access](#)
As an SSH administrator, you can setup the SSH server wallet, the SSH server endpoint, and the OpenSSH configuration on the SSH server as described in the section *Controlling access to SSH server centrally with Oracle Key Vault*.

SSH Administrators Managing both the SSH User's Keys and the SSH Server Access

As an SSH administrator, you can setup the SSH Server wallet, the SSH Server endpoint, and the OpenSSH configuration on the SSH server as described in the section *Controlling access to SSH server centrally with Oracle Key Vault*.

The SSH administrator can setup the SSH user's keys as and SSH client endpoint. See [Managing SSH User Keys with Oracle Key Vault](#).

Since the SSH administrator has full control over the SSH user's keys, the SSH administrator can rotate them per policy and also add the SSH user's public keys to the SSH server wallets to grant or revoke access to SSH servers.

Related Topics

- [Controlling Access to SSH Server Centrally with Oracle Key Vault](#)
You can centrally manage which SSH users have access to which SSH servers.

SSH Client Managing SSH User Keys and SSH Admin Managing the SSH Server Access

As an SSH administrator, you can setup the SSH server wallet, the SSH server endpoint, and the OpenSSH configuration on the SSH server as described in the section *Controlling access to SSH server centrally with Oracle Key Vault*.

The SSH users can setup their SSH user's keys and SSH client endpoint. See [Managing SSH User Keys with Oracle Key Vault](#).

The SSH administrator can add the public keys from the SSH user's public keys wallet to the SSH server wallets to grant or revoke access to SSH servers. The SSH administrator cannot rotate the user's keys on their own and would need to notify the SSH user to perform key rotation. But, the SSH administrator will be able to revoke access from the SSH server hosts if the keys are not rotated on time.

Related Topics

- [Controlling Access to SSH Server Centrally with Oracle Key Vault](#)
You can centrally manage which SSH users have access to which SSH servers.

Migrating Existing SSH Deployments to Oracle Key Vault

Administrators can migrate their SSH deployments for SSH user key management or managing access control of SSH servers or both.

To migrate the access control of SSH server into Oracle Key Vault:

1. Identify the SSH users on each SSH server host.
2. For each SSH user identified on the SSH server host, create an SSH server wallet.
3. For each SSH server create an SSH server endpoint.
4. Grant the SSH server endpoint **read only** rights to SSH server wallet.
5. Deploy the SSH endpoint on the SSH server host.
6. Upload the public keys (from the SSH server host user's *authorized_keys* file) as SSH keys into their corresponding SSH server wallet on the Oracle Key Vault server.
7. Setup the SSH server mapping file.
8. Setup the OpenSSH daemon to look at the SSH server endpoint lookup script inside the deployed SSH endpoint environment.
9. Optionally limit the *authorized_keys* file use on the SSH servers.

To migrate SSH user key management:

1. Identify the SSH users you want to migrate to Oracle Key Vault.
2. Create a regular endpoint and general wallet for SSH users.
3. Deploy the regular endpoint on SSH client host.
4. Use the REST APIs to register the public-private keys of all SSH users as SSH keys into their corresponding general wallets on the Oracle Key Vault server.
5. Use the PKCS#11 library when initiating an ssh connection.
For UNIX Platforms:

```
ssh -I $OKV_HOME/lib/liborapkcs.so <ssh_server_user>@<ssh_server_host>
```

Example:

```
ssh -I $OKV_HOME/lib/liborapkcs.so oracle@phoenix
```

For Windows Platforms:

```
ssh -I $OKV_HOME/lib/libsshpkcs.dll <ssh_server_user>@<ssh_server_host>
```

Example:

```
ssh -I $OKV_HOME/lib/libsshpkcs.dll oracle@phoenix
```

If you are migrating both the access control of SSH servers and the SSH user key management to Oracle Key Vault, then do it in two phases. First migrate the access control of SSH server into Oracle Key Vault. Once every SSH server that needs migration is centrally managed in Oracle Key Vault then migrate the SSH user key management into Oracle Key Vault.

Guidelines for OpenSSH SSHD Configuration

When you use Oracle Key Vault to securely manage SSH user keys or control access to SSH servers, the time to establish an SSH connection depends on multiple factors.

Some of the factors that determine the performance characteristics of establishing an SSH connection are:

- SSH user keys are managed locally or in an Oracle Key Vault.
- The number of SSH user keys in the user's general wallet.
- SSH server access is controlled through the `authorized_keys` file or from the Oracle Key Vault.
- Number of concurrent SSH connection requests.

Additionally, one or both the SSH clients or SSH servers may connect to Oracle Key Vault, adding a network delay during the SSH authentication process.

Review the following OpenSSH SSHD configuration parameters and modify them as necessary:

LoginGraceTime

- The server disconnects after `LoginGraceTime` time if the user has not successfully logged in.

- If the SSH user's endpoint has access to multiple SSH keys, then `openssh` makes an attempt to authenticate each key, one at a time. You should set the `LoginGraceTime` to a value that is sufficient for the authentication attempts to be made using all available keys for the user.
- Oracle recommends that you maintain a small number of keys for each user. Move the old, unused keys to other wallets.

MaxAuthTries

- Specifies the maximum number of authentication attempts permitted per connection.
- If the SSH user's endpoint has access to multiple SSH keys, then `openssh` makes an attempt to authenticate using each key, one at a time. Each offered key counts as one attempt. An SSH connection fails if authentication remains unsuccessful after `MaxAuthTries` authentication attempts. You should configure `MaxAuthTries` parameter to a value larger than the number of SSH keys accessible to SSH user's endpoint.
- Oracle recommends that you maintain a small number of keys for each user. Move the old, unused keys to other wallets.

MaxStartUps

- Specifies the maximum number of concurrent, unauthenticated connections to the SSHD. Additional connections will be dropped until authentication succeeds or the `LoginGraceTime` expires for a connection.
- You should set `MaxStartUps` considering the expected number of concurrent connections in your environment.

Reports

Oracle Key Vault offers several reports that you can use to check Secure Shell (SSH) user keys, SSH server access control configuration and also to track the SSH user activity.

- [Viewing SSH Reports](#)
All users can view the SSH reports.
- [SSH Key Details Reports](#)
The SSH Key details reports group includes the SSH Key Metadata Report and SSH Key Usage Report.
- [SSH Server Access Management Reports](#)
Oracle Key Vault offers SSH Server Access Management reports to view the authorized SSH user who can access the reports.
- [SSH User Private Key Management Reports](#)
Oracle Key Vault offers SSH User Private Key Management reports to view the SSH user who can access the private keys.

Viewing SSH Reports

All users can view the SSH reports.

Oracle Key Vault offers several reports that you can use to centrally manage the configuration of SSH user keys, their authorization to access SSH servers, and monitor the SSH server host access. These include:

SSH Key Details Reports

The SSH Key details reports group includes the SSH Key Metadata Report and SSH Key Usage Report.

Oracle Key Vault Secure Shell (SSH) Details reports consists of the following reports:

- SSH Key Metadata Report
- SSH Key Usage Report

SSH Key Metadata Report

This report shows the metadata for SSH public and private keys including owner, key length, expiration, and wallet membership.

You use this report to gain complete visibility of the SSH keys inventory and ensure that SSH key management best practices and enterprise specific policies are uniformly applied.

You can use this report to:

- identify weaker keys - keys with key size smaller than what your enterprise security policy may require
- identify keys that are expiring soon and rotate them before they expire to avoid any disruption
- identify keys that are likely unused if they are not a member of any wallets

SSH User	Key Fingerprint	Key Type	Algorithm	Key Length	Key Expiration	Extractable	State	Wallet Membership	Key Details
John	SHA256:z+JQz0VnTQyg7Good5lcDVMSrNM+48l9Ww6RqIhIps	Public Key	RSA	2048	05-OCT-2025 18:29:22 (in 729 days)	-	Active	oracle_ssh_wallet, John_SSH_Keys_Wallet	Details
John	SHA256:z+JQz0VnTQyg7Good5lcDVMSrNM+48l9Ww6RqIhIps	Private Key	RSA	2048	05-OCT-2025 18:29:22 (in 729 days)	False	Active	John_SSH_Keys_Wallet	Details
Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwM5pTXtehsiuSlmuHRhkJU	Public Key	RSA	2048	05-OCT-2025 18:29:28 (in 729 days)	-	Active	opc_ssh_wallet, Jane_SSH_Keys_Wallet	Details
Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwM5pTXtehsiuSlmuHRhkJU	Private Key	RSA	2048	05-OCT-2025 18:29:28 (in 729 days)	False	Active	Jane_SSH_Keys_Wallet	Details
Sam	SHA256:HZrlsgGhKr6d45WT/b2Xb5CHZByVmO+Dz19J9glj5c	Public Key	RSA	4096	05-OCT-2025 18:29:35 (in 729 days)	-	Active	root_ssh_wallet_for_austin, Sam_sysadmin_SSH_Keys_Wallet	Details
					05-OCT-2025 18:29:35				

SSH Key Usage Report

This report shows all operations on currently managed SSH keys by endpoints and the users.

You use this report to:

- monitor usage of the SSH keys to identify any suspicious activity or any tampering to the key authorization or key attributes.
- identify usage pattern and unused keys. For example:
 - a user may have multiple SSH keys but some of the keys may not be in use anymore.

- a user may have both public and private keys in Oracle Key Vault but only private key access is shown, indicating use of local `authorized_keys` files on SSH servers.

SSH Key Usage Report Close

Q Go Actions ▾

SSH User	SSH User Key Fingerprint	Key Type	Operation	User	Endpoint	Endpoint Type	Time	Result
Jill	SHA256:TqernzdN4VmZR1wfNpnXjYWolmoWUgBfll6yXNrtCuc	Private Key	Create Key Pair	SSH_ADMIN	-	-	07-OCT-2023 15:18:03	✓
John	SHA256:z+JQz0VnTQygl7Good5icDVMSrNM+48i9Ww6RqIhIps	Private Key	Sign	-	JOHN_EP	Oracle Database	07-OCT-2023 15:12:55	✓
John	SHA256:z+JQz0VnTQygl7Good5icDVMSrNM+48i9Ww6RqIhIps	Private Key	Get Attribute(s)	-	JOHN_EP	Oracle Database	07-OCT-2023 15:12:55	✓
John	SHA256:z+JQz0VnTQygl7Good5icDVMSrNM+48i9Ww6RqIhIps	Public Key	Get	-	PHOENIX_SSH_SERVER	SSH Server	07-OCT-2023 15:12:51	✓
John	SHA256:z+JQz0VnTQygl7Good5icDVMSrNM+48i9Ww6RqIhIps	Public Key	Locate	-	PHOENIX_SSH_SERVER	SSH Server	07-OCT-2023 15:12:50	✓
John	SHA256:z+JQz0VnTQygl7Good5icDVMSrNM+48i9Ww6RqIhIps	Public Key	Get Attribute(s)	-	JOHN_EP	Oracle Database	07-OCT-2023 15:12:43	✓
John	SHA256:z+JQz0VnTQygl7Good5icDVMSrNM+48i9Ww6RqIhIps	Public Key	Get	-	JOHN_EP	Oracle Database	07-OCT-2023 15:12:43	✓
John	SHA256:z+JQz0VnTQygl7Good5icDVMSrNM+48i9Ww6RqIhIps	Public Key	Get Attribute(s)	-	JOHN_EP	Oracle Database	07-OCT-2023 15:12:43	✓

1 - 116 of 116

SSH Server Access Management Reports

Oracle Key Vault offers SSH Server Access Management reports to view the authorized SSH user who can access the reports.

Oracle Key Vault SSH Server Access Management reports consists of the following reports:

- **SSH Server Authorization Report**
- **SSH Server Access Report**
- **SSH Server Wallet Report**

SSH Server Authorization Report

This report shows SSH servers and the SSH users who are authorized to access them. It is used to review who has access to SSH servers.

You use this report to:

- review the authorization granted to SSH users to access the SSH Servers through their public keys.
This will help you ensure that access is granted to only those who need it and identify any unintended or unexpected authorization.
- identify left-over access grants, such as from when an employee leaves the organization or changes his/her role.

SSH Server Authorization Report						Close
SSH Server ↑	SSH Server Host User	SSH Server Wallet	SSH User	SSH User Public Key Fingerprint	SSH User Public Key Details	
austin.oracle.com	opc	opc_ssh_wallet	Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwM5pTXtehsiu5ImuHRhkU		
austin.oracle.com	oracle	oracle_ssh_wallet	John	SHA256:z+JQz0VnTQyg7Good5lcDVMsRNM+48i9Ww6RqlhIps		
austin.oracle.com	root	root_ssh_wallet_for_austin	Sam	SHA256:HzrlsgGHKr6D45WT/b2Xb5CHHZByVmO+Dz19J9glj5c		
phoenix.oracle.com	opc	opc_ssh_wallet	Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwM5pTXtehsiu5ImuHRhkU		
phoenix.oracle.com	oracle	oracle_ssh_wallet	John	SHA256:z+JQz0VnTQyg7Good5lcDVMsRNM+48i9Ww6RqlhIps		
phoenix.oracle.com	root	root_ssh_wallet_for_phoenix	Sally	SHA256:Ke4M7IYO4W4XRwpf5/fh3NLs0MgQpiqirT27h4Dgfw		

1 - 6 of 6

SSH Server Access Report

This report shows attempts to access SSH servers using SSH users' public keys. It is used to assess who is accessing the SSH servers.

As part of SSH user authentication, fingerprint of the public key offered by the client, the SSH host user and the wallet that the SSH server endpoint looks up to determine access are recorded. The information shown in the SSH User column is derived from the SSH public key metadata if it exists in system. The SSH User column will show the value 'Unknown' if the SSH public key that is used to access an SSH server is managed outside of Oracle Key Vault, that is SSH user information associated with the SSH public key is unavailable within Oracle Key Vault.

A user trying to establish an SSH connection may have more than one public key. This report shows attempts made using each public key to authenticate the user. If an SSH user has multiple public keys and some of these public keys are not authorized to access the SSH server, then you may see entries with failed access attempts before an entry for the successful access attempt for that user.

You use this report to:

- monitor the SSH server access to ensure that SSH servers are accessed by only authorized users. This information is useful for the forensic analysis of a security incident.
- review the failed attempts to access an SSH server to determine if these attempts were legitimate or otherwise.

SSH Server Access Report Close

Q Go Actions

SSH Server	SSH Server Host User	SSH User	SSH User Public Key Fingerprint	Access Time	Result
phoenix.oracle.com	oracle	John	SHA256:z+JQz0VnTQygI7Good5lcDVMsRNM+48i9Ww6RqIhIps	07-OCT-2023 15:23:27	✓
phoenix.oracle.com	root	Sally	SHA256:Ke4M7IYO4W4XRwpf5/fh3NLs0MgQpiqairT27h4Dgfw	07-OCT-2023 15:23:25	✓
phoenix.oracle.com	root	Malfoy	SHA256:a+f5d9UuF7JDYfiyRr1ejlH2WkvQsp19Bh0Um0Yu44	07-OCT-2023 15:23:24	Failed
austin.oracle.com	opc	Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwMSPtXtehsiuSlmuHRhkJU	07-OCT-2023 15:23:24	✓
austin.oracle.com	root	Sam	SHA256:HZrlsgGHK6d4SWT/bzXb5CHHZByVmO+Dz19J9glj5c	07-OCT-2023 15:23:23	✓
austin.oracle.com	root	Malfoy	SHA256:a+f5d9UuF7JDYfiyRr1ejlH2WkvQsp19Bh0Um0Yu44	07-OCT-2023 15:23:23	Failed
phoenix.oracle.com	oracle	John	SHA256:z+JQz0VnTQygI7Good5lcDVMsRNM+48i9Ww6RqIhIps	07-OCT-2023 15:12:50	✓
austin.oracle.com	opc	Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwMSPtXtehsiuSlmuHRhkJU	07-OCT-2023 15:12:20	✓
phoenix.oracle.com	oracle	John	SHA256:z+JQz0VnTQygI7Good5lcDVMsRNM+48i9Ww6RqIhIps	07-OCT-2023 15:04:41	✓
phoenix.oracle.com	oracle	John	SHA256:z+JQz0VnTQygI7Good5lcDVMsRNM+48i9Ww6RqIhIps	07-OCT-2023 13:40:36	✓
phoenix.oracle.com	root	Sally	SHA256:Ke4M7IYO4W4XRwpf5/fh3NLs0MgQpiqairT27h4Dgfw	07-OCT-2023 13:40:35	✓

1 - 20 of 20

SSH Server Wallet Report

This report shows SSH server wallets that authorize the access of SSH servers as SSH server host users. It is used when provisioning new SSH users and/or new SSH servers.

You use this report to:

- determine the SSH server wallets where SSH public key of a newly provisioned user needs to be added.
- determine the SSH server wallets that a newly provisioned server should use if the new server follows the SSH access configuration of other servers in the enterprise.
- get the enterprise-wide view of the SSH server wallets and the SSH servers for which they authorize access. You can use an SSH server wallet to authorize SSH access on multiple servers.

SSH Server Wallet Report Close

Q Go Actions

SSH Server	SSH Server Host User	SSH Server Wallet	Wallet Details
austin.oracle.com	opc	opc_ssh_wallet	
austin.oracle.com	oracle	oracle_ssh_wallet	
austin.oracle.com	root	root_ssh_wallet_for_austin	
phoenix.oracle.com	opc	opc_ssh_wallet	
phoenix.oracle.com	oracle	oracle_ssh_wallet	
phoenix.oracle.com	root	root_ssh_wallet_for_phoenix	

1 - 6 of 6

SSH User Private Key Management Reports

Oracle Key Vault offers SSH User Private Key Management reports to view the SSH user who can access the private keys.

Oracle Key Vault SSH User Private Key Management reports consists of the following reports:

- **SSH Private Key Authorization Report**
- **SSH Private Key Usage Report**

SSH Private Key Authorization Report

This report shows the endpoints and users who are authorized to use SSH private keys and also how their access is granted. It is used to review who has access to SSH private keys.

You use this report to:

- review and audit endpoints and users who are granted access of the SSH private keys.
- identify and limit use of shared private keys. An SSH private key is meant for a specific user or an endpoint. Any temporary sharing of SSH private keys must be removed as soon as the business need is completed.

SSH User	SSH User Key Fingerprint	User	Endpoint	Access Type	Access Through Wallet	SSH User Private Key Details
Sally	SHA256:Ke4M7IYO4W4XRwpf5/fh3NLs0MgQpiqirT27h4Dgfw	-	SALLY_SYSADMIN_EP	Direct	Sally_sysadmin_SSH_Keys_Wallet	Details
Sally	SHA256:Ke4M7IYO4W4XRwpf5/fh3NLs0MgQpiqirT27h4Dgfw	-	SSHADMIN	Creator		Details
John	SHA256:z+JQz0VnTQygl7Good5lcDVMsRNM+48i9Ww6Rqlhps	-	JOHN_EP	Direct	John_SSH_Keys_Wallet	Details
John	SHA256:z+JQz0VnTQygl7Good5lcDVMsRNM+48i9Ww6Rqlhps	-	SSHADMIN	Creator		Details
Jane	SHA256:R4+IsT6af1+mC3+IqrhcVwMSPtXtehsiuSlmuHRhkU	-	JANE_EP	Direct	Jane_SSH_Keys_Wallet	Details
Jane	SHA256:R4+IsT6af1+mC3+IqrhcVwMSPtXtehsiuSlmuHRhkU	-	SSHADMIN	Creator		Details
Malfoy	SHA256:a+f5d9UuF7JDYfiyRr1ej1lH2WkvQsp19Bh0Um0Yu44	-	MALFOY_EP	Direct	Malfoy_Wallet	Details
Malfoy	SHA256:a+f5d9UuF7JDYfiyRr1ej1lH2WkvQsp19Bh0Um0Yu44	-	SSHADMIN	Creator		Details
Jill	SHA256:TqernzdN4VmZR1wfnPnXjYVolmoWugBfl6yXNr1Cuc	SSH_ADMIN	-	Creator		Details
Sam	SHA256:HZrlsgGhKr6D45WT/b2Xb5CHHZByVmO+Dz19J9glj5c	-	SAM_SYSADMIN_EP	Direct	Sam_sysadmin_SSH_Keys_Wallet	Details
Sam	SHA256:HZrlsgGhKr6D45WT/b2Xb5CHHZByVmO+Dz19J9glj5c	-	SSHADMIN	Creator		Details

1 - 11 of 11

SSH Private Key Usage Report

This report shows the use of SSH user private keys by endpoints and users. It is used to assess changes to the SSH users private keys and review their access.

This report includes the activity information from the use of the SSH private keys to access SSH servers as well as all operations that are performed on them.

You use this report to:

- review and audit the use of SSH private keys. You can review access patterns of the private key and identify unusual activity like use of a user's private key when user is away on a vacation or fetch of a user's private key by an unexpected endpoint.

- identify any unusual changes that are made to SSH private keys. For example, the extractable attribute of an SSH private key is modified so that the key can leave the Oracle Key Vault boundary. Such changes could be a cause of concern and must be reviewed. You should immediately rotate SSH keys whose integrity may be suspect.

SSH Private Key Usage Report Close

Q Go Actions

SSH User	SSH User Key Fingerprint	Operation	User	Endpoint	Time	Result
Jill	SHA256:TqernzdN4VmZR1wfNpnXjYWolmoWUgBfll6yXNr1Cuc	Create Key Pair	SSH_ADMIN	-	07-OCT-2023 15:18:03	✓
John	SHA256:z+JQz0VnTQygl7Good5lcDVMsRNM+48i9Ww6RqlhIps	Sign	-	JOHN_EP	07-OCT-2023 15:12:55	✓
John	SHA256:z+JQz0VnTQygl7Good5lcDVMsRNM+48i9Ww6RqlhIps	Get Attribute(s)	-	JOHN_EP	07-OCT-2023 15:12:55	✓
Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwMSPtXtehsiuSImuHRhkU	Sign	-	JANE_EP	07-OCT-2023 15:12:30	✓
Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwMSPtXtehsiuSImuHRhkU	Get Attribute(s)	-	JANE_EP	07-OCT-2023 15:12:30	✓
Jane	SHA256:R4+IsT6af1+mC3+IQrhcxVwMSPtXtehsiuSImuHRhkU	Get Attribute(s)	-	JANE_EP	07-OCT-2023 15:10:40	✓
John	SHA256:z+JQz0VnTQygl7Good5lcDVMsRNM+48i9Ww6RqlhIps	Sign	-	JOHN_EP	07-OCT-2023 15:04:50	✓
John	SHA256:z+JQz0VnTQygl7Good5lcDVMsRNM+48i9Ww6RqlhIps	Get Attribute(s)	-	JOHN_EP	07-OCT-2023 15:04:50	✓

1 - 22 of 22

17

Managing Online and Offline Secrets

You can store and manage credential files in Oracle Key Vault as opaque objects that an endpoint can retrieve when needed.

- [Uploading and Downloading Credential Files](#)
The `okvutil upload` and `okvutil download` commands can upload and download credential files.
- [Managing Secrets and Credentials for SSH](#)
You can perform public key authentication with private keys that are protected in Oracle Key Vault.
- [Integrating Oracle Key Vault with SSH Public Key Authentication](#)
You can use Oracle Key Vault to store your public and private keys for SSH.
- [Centrally Managing Passwords in Oracle Key Vault](#)
You can centrally manage passwords in Oracle Key Vault by using external keystores or adding them to shared virtual wallets as secrets.

Uploading and Downloading Credential Files

The `okvutil upload` and `okvutil download` commands can upload and download credential files.

- [About Uploading and Downloading Credential Files](#)
You use the `okvutil` utility to upload and download credential files.
- [Uploading a Credential File](#)
The `okvutil upload` command can upload credential files.
- [Downloading a Credential File](#)
The `okvutil download` command can download credential files.
- [Guidelines for Uploading and Downloading Credential Files](#)
Oracle provides recommendations for when you upload and download credential files.

About Uploading and Downloading Credential Files

You use the `okvutil` utility to upload and download credential files.

Credential files are uploaded and stored as opaque objects in Oracle Key Vault, which means that Oracle Key Vault does not parse the contents of the file like an Oracle wallet or Java keystore. The upload process does not alter the credential file.

Examples of opaque objects are as follows:

- Files that contain X.509 certificates
- Kerberos keytabs
- Files containing Passwords
- Public and (non-extractable) private SSH keys

Uploading these credential files provides a central, secure location for long-term retention. After you have uploaded a credential file, you can download it in the same server location or share it with other trusted servers. Oracle Key Vault supports credential files up to 128 KB in size.

You can place the credential file anywhere in your server infrastructure (which includes database servers and application servers) that is accessible by an Oracle Key Vault endpoint.

Uploading a Credential File

The `okvutil upload` command can upload credential files.

1. Ensure that the server that contains the credential file has been enrolled and provisioned as an Oracle Key Vault endpoint.
2. Ensure that access control has been configured for the endpoint.

If you are uploading the credential file to a virtual wallet, then ensure that the endpoint has read, modify, and manage wallet access to the wallet.

3. Run the `okvutil upload` command.

For example:

```
# okvutil upload -l "/etc/oracle/app/creds/hr.keytab" -t kerberos -g HRWallet -d  
"Kerberos keytab file for HR group, 06_11_14"  
Enter Oracle Key Vault endpoint password: Key_Vault_endpoint_password
```

In this example:

- `-l` is the directory path to the `hr.keytab` credential file, which is being uploaded. Enclose the directory location in double quotation marks.
- `-t` specifies the type of credential file, which in this example is a Kerberos keytab file. In addition to `KERBEROS`, other types that you can specify are as follows:
 - `SSH` for an SSH key file
 - `OTHER` for other files that store secrets, such as uploaded or downloaded files
- `-g` adds the credential file to the virtual wallet `HRWallet`, which must already exist. This parameter enables you to upload the credential to a wallet that is specifically for the HR application users' needs, rather than to the default virtual wallet. In this example, `HRWallet` is the Oracle Key Vault virtual wallet to which access control was configured in Step 2.
- `-d` is an optional description. As a best practice, include a brief description of what the credential file is used for and the date you performed the upload. This information helps for future reference and tracking of the credential file. You can modify this description later on in the Oracle Key Vault management console if necessary.

Related Topics

- [okvutil upload Command](#)
The `okvutil upload` command uploads security objects to Oracle Key Vault.
- [Managing Endpoints](#)
You can enroll, reenroll, suspend, rotate, and delete endpoints.
- [Granting Access to Users, User Groups, Endpoints, and Endpoint Groups](#)
You can grant the **Read Only**, **Read and Modify**, and **Manage Wallet** access levels to users, user groups, endpoints, and endpoint groups.

Downloading a Credential File

The `okvutil download` command can download credential files.

1. Find the unique ID of the credential file that you must download, by using one of the following methods:
 - **Oracle Key Vault management console:** Log in as user who has the necessary access to the virtual wallet. In the Oracle Key Vault management console, from the **Keys & Wallets** tab, select **Keys, Secrets & Objects** to find the uploaded files. Note the unique ID of the uploaded file that you want to download. Credential files are listed as opaque objects.
 - **okvutil list command:** Run the `okvutil list` command from an endpoint that has access to the credential file or a virtual wallet that contains the credential file. Locate the unique ID of the credential file that you must download based on the description that you provided when you uploaded the file.
2. From the command line, run the `okvutil download` command to download the credential file.

For example:

```
# okvutil download -l "/etc/oracle/app/newcreds/hr.keytab" -t kerberos -i  
6ba7b810-9dad-11d1-80b4-00c04fd430c8  
Enter Oracle Key Vault endpoint password: Key_Vault_endpoint_password
```

In this example:

- `-l` is the directory to which you want to download the uploaded credential.
- `-t` specifies the type of credential file, which in this example is a Kerberos keytab file. In addition to `KERBEROS`, other types that you can specify are as follows:
 - `SSH` for an SSH key file
 - `OTHER` for other files that store secrets, such as uploaded or downloaded files
- `-i` is the unique ID of the credential file.

Related Topics

- [okvutil upload Command](#)
The `okvutil upload` command uploads security objects to Oracle Key Vault.

Guidelines for Uploading and Downloading Credential Files

Oracle provides recommendations for when you upload and download credential files.

- After you complete the upload, upload the credential file again the next time it is changed. Otherwise, the uploaded (and subsequent downloaded version) file will be outdated. Periodically, you should compare the last modification date of the credential file with the timestamp of the uploaded version.
- Use care if you use the `okvutil upload` and `okvutil download` commands, which provide an overwrite (`-o`) option. This option overwrites the uploaded credential file. You may want to create backups of the credential files before beginning the upload and download processes.
- You can share one credential file among multiple server endpoints. Add the credential file as an opaque object to a virtual wallet using the `-g` option of the `okvutil upload`

command. Grant access of that virtual wallet to all the server endpoints. Optionally, define an endpoint group and then make all the server endpoints members of that endpoint group. Grant this endpoint group access to that virtual wallet. Afterward, all the members of the group will have access to that wallet.

Related Topics

- [About Configuring Alerts](#)

System administrators can configure alerts from the Oracle Key Vault dashboard, but all users can see alerts for the security objects to which they have access.

Managing Secrets and Credentials for SSH

You can perform public key authentication with private keys that are protected in Oracle Key Vault.

In many IT departments, public key authentication is the default approach to securely log into remote servers without having an administrator having to remember the password, for both on-premise or cloud-based hosts. If you are using Oracle Cloud Infrastructure (Oracle OCI), then the password of the user `opc` is unknown, so public key authentication is the only way to log into an OCI compute instance. This method is convenient, but not without risk. If the private key is lost, then remotely logging in is no longer possible. In addition, the process of provisioning a new public key is time consuming. Another problem is that because the private key uniquely identifies its owner, if it is stolen, copied, or compromised, then an intruder can easily cause a great deal of trouble, with the evidence pointing at the owner, not the intruder.

As a solution to these problems, consider uploading the SSH private keys into Oracle Key Vault by using the `okvutil upload` command. The benefits are as follows:

- Because the SSH keys are not on your host computer, they cannot be copied or stolen, and furthermore, they are safe from disk or file corruptions.
- The SSH keys are included in Oracle Key Vault backup operations, so losing them is impossible.
- Oracle Key Vault provides continuous and fault-tolerant availability by allowing up to 16 Oracle Key Vault servers to connect to one multi-master cluster. Read/write pairs guarantee that highly sensitive information (such as encryption keys or passwords) are replicated to at least one or more Oracle Key Vault servers.

Related Topics

- [okvutil upload Command](#)

The `okvutil upload` command uploads security objects to Oracle Key Vault.

Integrating Oracle Key Vault with SSH Public Key Authentication

You can use Oracle Key Vault to store your public and private keys for SSH.

The Key Administrator or users with appropriate access can centrally manage the public or private key pairs for SSH public key authentication in Oracle Key Vault. When required, the key pairs can be rotated or revoked. As the keys are centrally managed, an authorized user can login to a remote machine without the private key existing on the local disk.

- [Step 1: Creating and Uploading the Key Pair](#)
Use an Oracle Key Vault endpoint to create and upload the key pair.
- [Step 2: Using an Endpoint to SSH a Remote Host](#)
You can use an endpoint to SSH to a remote host.

- **Step 3: Incorporating an ssh-agent**
You can incorporate an ssh-agent to prevent specifying the PKCS#11 library.

Step 1: Creating and Uploading the Key Pair

Use an Oracle Key Vault endpoint to create and upload the key pair.

OpenSSH must be at least at version 7.2p1 for endpoints that do not require a password and at version 8.1p1 for endpoints that do require a password.

1. Create a key pair, both in PKCS#8 format. It must be an RSA key pair, with a key size of 1024, 2048, or 4096.

Note:

Do not enter a passphrase when prompted. Oracle Key Vault does not currently support using encrypted private keys for SSH.

This generates the following files:

- `key.pem`: The private key in PKCS#8 format, should start with `-----BEGIN PRIVATE KEY-----`.
- `key.pem.pub`: The public key in RFC4716 format.
- `public.pem`: The public key in PKCS#8 format, should start with `-----BEGIN PUBLIC KEY-----`.

```
ssh-keygen -q -t rsa -m PKCS8 -f key.pem -b <key size>
ssh-keygen -e -f key.pem.pub -mPKCS8 > public.pem
```

2. Add the contents of `key.pem.pub` to the remote host's `authorized_keys` file.

Note:

While creating the `authorized_keys` file or its parent directory, if either does not exist, ensure they are created with the correct permissions.

3. To test public key authentication at this stage (without using the Oracle Key Vault PKCS#11 library), use the following command: `ssh -i ./key.pem <username>@<remote host>`.
4. Upload the private key to Oracle Key Vault using the RESTful services utility. `./bin/okv managed-object private-key register --object ./key.pem --algorithm RSA --length <key size> --activation-date now`
Make sure the `<key size>` should be same as chosen previously.
5. Upload the public key to Oracle Key Vault using the RESTful services utility. `./bin/okv managed-object public-key register --object ./public.pem --algorithm RSA --length <key size> --private-key-uuid <private key UUID> --activation-date now`
Make sure that the `private key UUID` is the same as the UUID returned by the previous command.

6. Delete the public and private keys from the local machine.

You can retrieve the private and public key ;later in the PKCS#8 format using the RESTful services utility,

```
rm key.pem key.pem.pub public.pem
```

Step 2: Using an Endpoint to SSH a Remote Host

You can use an endpoint to SSH to a remote host.

The endpoints that SSH to the remote host must have access to both the public and private keys. Endpoints can be given access using the following:

- being the endpoint that created the key pair
- being directly given access to the wallet in which the keys are located
- being added to an endpoint group that has been given access to the wallet in which the keys are located

After completing the configuration steps in the Oracle Key Vault and the SSH client host, the SSH user can now initiate the SSH connections using OpenSSH client and Oracle Key Vault PKCS#11 library.

Ensure that OpenSSH client version is 8.1 or higher.

To initiate SSH connections from the SSH client host to the SSH server host:

1. As an SSH user, log in to the SSH client host.
2. Set the `OKV_HOME` environment variable to the SSH user endpoint's installation directory.
3. Use the PKCS#11 library from `$OKV_HOME/lib` to connect to the SSH server host:

For UNIX Platforms:

```
ssh -I $OKV_HOME/lib/liborapkcs.so <ssh_server_user>@<ssh_server_host>
```

For Windows Platforms:

Use the PKCS#11 library `$OKV_HOME/lib/libsshpkcs.dll` instead:

```
ssh -I $OKV_HOME/lib/libsshpkcs.dll <ssh_server_user>@<ssh_server_host>
```

4. The preceding command asks for the endpoint's password with the prompt:

```
Enter pin for 'label':
```

For the auto-login endpoint, enter `NULL`. Otherwise, enter the endpoint's password and press **Enter**.

5. The SSH session is now established.

 **Note:**

Use ssh-agent to avoid supplying the PKCS#11 library path and the pin repeatedly. After the ssh agent is setup and the pin for the endpoint's PKCS#11 library has been added, you can then run the ssh command as usual. For example:

```
ssh oracle@phoenix
```

 **Note:**

Oracle recommends that you use a password protected endpoint. When you use an auto-login endpoint, the protection of the SSH keys of the user is dependent on the file permissions only.

Step 3: Incorporating an ssh-agent

You can incorporate an ssh-agent to prevent specifying the PKCS#11 library.

You can use ssh-agent to prevent specifying the PKCS#11 library location and endpoint password each time you want to SSH into a remote host.

This requires having setup the key pair as described in, *Step 2: Using an Endpoint to SSH a Remote Host*.

1. Identify the directory in which the

```
liborapkcs.so
```

is located.

2. Setup ssh-agent, whitelisting the directory from the previous step:

```
eval `ssh-agent -P "<full path to directory containing liborapkcs.so>/*"`
```

3. Add PKCS#11 library provided keys to the agent via ssh-add.
4. Enter the endpoint password, when prompted for the PIN.
5. If the endpoint does not have a password, enter "NULL".

```
ssh-add -s <full path to liborapkcs.so>
```

6. Test the setup using ssh command without any option.

```
ssh <username>@<remote host>
```

Centrally Managing Passwords in Oracle Key Vault

You can centrally manage passwords in Oracle Key Vault by using external keystores or adding them to shared virtual wallets as secrets.

- [About Centrally Managing Passwords in Oracle Key Vault](#)
You can store passwords centrally in Oracle Key Vault and retrieve these passwords securely (for example, when logging into an Oracle database).
- [Creating and Sharing Centrally Managed Passwords](#)
To create and share centrally managed passwords (external keystore passwords) for large database deployments, you first must use an Oracle Key Vault client and RESTful services utility commands.
- [Example: Script for Storing and Retrieving Oracle Database User Passwords in and from Oracle Key Vault](#)
You can store passwords in Oracle Key Vault and retrieve them when needed.
- [Sharing Secrets with Other Databases](#)
Sharing information in Oracle Key Vault requires a virtual wallet in Oracle Key Vault, and multiple databases connecting into their own endpoints that have access to the shared wallet.
- [Changing Passwords for a Large Database Deployment](#)
For better security, on a regular basis, you should change passwords that are used by both humans and automated processes.

About Centrally Managing Passwords in Oracle Key Vault

You can store passwords centrally in Oracle Key Vault and retrieve these passwords securely (for example, when logging into an Oracle database).

Storing passwords centrally in Oracle Key Vault has the following benefits:

- It eliminates the need for clear text passwords in your maintenance scripts.
- It eliminates the need for the secure external password store.
- It simplifies password changes.
- It does not require changes to the `sqlnet.ora` file.

Database administrators often use automated scripts to perform regular maintenance such as Oracle Recovery Manager (Oracle RMAN) backups, loading data into Oracle Data Warehouse, or refreshing materialized views. These scripts log in to the Oracle Database with the passwords of highly privileged users, which unfortunately is a poor security practice in that it entails hard-coded clear-text passwords in the script. And of course if the password changes, then all the scripts must be changed as well. One way to remove the need for clear-text passwords is to put the passwords into an auto-open wallet (called the secure external password store) and instruct the client to retrieve the password from there. This practice works well for a few databases, but if you have hundreds of databases, it is difficult to manage. Furthermore, password changes require you to update every secure external password store. By storing passwords in a central location, you eliminate this problem.

Creating and Sharing Centrally Managed Passwords

To create and share centrally managed passwords (external keystore passwords) for large database deployments, you first must use an Oracle Key Vault client and RESTful services utility commands.

1. Ensure that you have installed the Oracle Key Vault RESTful services utility command-line interface.

If you do not have the RESTful services utility command-line interface installed, then you can download it from the Oracle Key Vault management console as follows:

```
curl -O -k https://Oracle_Key_Vault_IP_address:5695/okvrestclipackage.zip
```

2. Install an Oracle Key Vault client that does not use a password.

If you already have an Oracle Key Vault client installed (for example, for TDE key management), then do not use this endpoint. Instead, install an additional endpoint for secrets management. You must use an Oracle Key Vault client that is not password protected.

To install the Oracle Key Vault client without a password, run the following RESTful services utility commands:

```
okv admin endpoint create --endpoint endpoint_name  
--description "Secrets management for endpoint_name" --type ORACLE_NON_DB  
--platform platform_OS
```

```
okv admin endpoint provision --endpoint endpoint_name  
--location installation_directory --auto-login TRUE
```

For this use case, executing the `root` script is not needed.

3. Upload the passwords for your database accounts to Oracle Key Vault.
 - a. Create text files that contain the passwords for each account. For example, suppose you have two maintenance scripts that log into the database. The first script uses an RMAN account `nightly_backup`, which logs into the PDB as Oracle Recovery Manager (Oracle RMAN) with the `SYSBACKUP` administrative privilege. The second script uses the `refresh_dwh` account, which refreshes a data warehouse by connecting to a PDB.

For illustration purposes, the following commands show the example passwords for each account. Oracle Database account passwords can have a maximum of 30 characters:

```
$ more backup_pwd password
```

```
$ more refresh_pwd password
```

- b. Register user names and passwords in Oracle Key Vault.

First, the password for `nightly_backup` is registered with Oracle Key Vault, followed by the password for `refresh_dwh` account. Oracle Key Vault returns the universally unique ID (UUID) of each of these managed objects. First, the registration:

- Register the password for `nightly_backup`.

```
$ PWD_RMAN=$(okv managed-object secret register --output_format
"text" --object "./backup_pwd"
--custom-attribute '[{"name": "x-NAME", "value": "NIGHTLY_BACKUP"},
{"name": "x-CONNECT-STRING", "value": "'${TWO_TASK}'"}]'
--activation-date "now")
echo $PWD_RMAN password
```

- Register the password for `refresh_dwh`.

```
PWD_REFRESH=$(okv managed-object secret register --output_format
"text" --object "./refresh_pwd"
--custom-attribute '[{"name": "x-NAME", "value" : "REFRESH_DWH"},
{"name": "x-CONNECT-STRING", "value": "'${TWO_TASK}'"}]'
--activation-date "now")
echo $PWD_REFRESH password
```

4. On Linux, securely delete the file that contain passwords.

```
shred -xzu ./backup_pwd ./refresh_pwd
```

After you complete this configuration, you can create a script that retrieves the password for these users from Oracle Key Vault and inserts the password into the SQL*Plus commands.

Related Topics

- [Example: Script for Storing and Retrieving Oracle Database User Passwords in and from Oracle Key Vault](#)
You can store passwords in Oracle Key Vault and retrieve them when needed.

Example: Script for Storing and Retrieving Oracle Database User Passwords in and from Oracle Key Vault

You can store passwords in Oracle Key Vault and retrieve them when needed.

The benefit of this script is that no matter how many administrative or maintenance scripts log into that database, the script stays the same. For each new user account, you only upload the password and attributes file for the user, and then activate the secret.

For example, a script called `log-me-in.sh` can be as follows:

```
#!/bin/bash
set +x
# *****
# ** Read user name and connect string from *****
# ** command line into variables *****
# *****

export user="${1}"
export TWO_TASK="${2}"
export PRIV="${3}"

KMIP_ID=$(okv managed-object object locate --output_format text --custom-
attribute '[{"name": "x-NAME", "value" : "'${user}'",
```

```
 {"name": "x-CONNECT-STRING", "value": "'${TWO_TASK}''"}]')
pwd="'$(okv managed-object secret get --output_format text --uuid $
{KMIP_ID})'"

if [ "${PRIV}" == 'AS SYSBACKUP' ]; then
  rman target ''''${user}/${pwd}@${TWO_TASK} ${PRIV}''''
else
  sqlplus ${user}/${pwd}
fi
```

**Note:**

`TWO_TASK` is an Oracle environment variable that holds the connect string

You can make this file immutable by applying extended attributes. For example, in Linux as the root user, use the following `chattr` command:

```
# chattr +i log-me-in.sh
```

To run this script, use the following syntax:

```
$ ./log-me-in.sh user_name hostname:port/service 'AS privilege'
```

For example to log in the `nightly_backup` user:

```
$ ./log-me-in.sh nightly_backup sales19c.us.example.com:1521/
finpdb.us.example.com 'AS SYSBACKUP'
```

Output similar to the following appears:

```
Recovery Manager: Release 19.0.0.0.0 - Production on Fri Jun 26 12:22:04 2020
Version 19.7.0.0.0
```

```
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: sales19c:finpdb (DBID=3200396863)
```

```
RMAN>
```

The following example command logs in as the `refresh_dwh` user:

```
$ ./log-me-in.sh refresh_dwh sales19c.us.example.com:1521/
finpdb.us.example.com
```

```
SQL> SHOW USER
REFRESH_DWH
```

If the owner of the target schema allows the `refresh_dwh` account user to log in, then the schema owner must grant the `CONNECT THROUGH` privilege to `refresh_dwh`, as follows:

1. In SQL*Plus, run the `ALTER USER` statement to create the proxy. For example:

```
ALTER USER HR GRANT CONNECT THROUGH refresh_dwh;
```

2. In the script, change the last line (`sqlplus "${user}"/"${pwd}"`) to include the proxy. For example, to include HR:

```
sqlplus "${user}"[HR]/"${pwd}"
```

Related Topics

- [Creating and Sharing Centrally Managed Passwords](#)
To create and share centrally managed passwords (external keystore passwords) for large database deployments, you first must use an Oracle Key Vault client and RESTful services utility commands.

Sharing Secrets with Other Databases

Sharing information in Oracle Key Vault requires a virtual wallet in Oracle Key Vault, and multiple databases connecting into their own endpoints that have access to the shared wallet.

1. Log in to the server from which you want to share the secret (the source server).
2. Create a virtual wallet and make it the default wallet of the endpoint.

For example, to make a default wallet for an endpoint called `secrets_db_name`:

```
okv manage-access wallet create --wallet shared_secrets
okv manage-access wallet set-default --wallet shared_secrets --endpoint
secrets_db_name
```

3. Add the secret (password) to the shared virtual wallet.

```
okv managed-object wallet add-member --uuid 994CB3E5-C5B3-4F75-BFED-
CB41AE15D0B1 --wallet shared_secrets
```

994CB3E5-C5B3-4F75-BFED-CB41AE15D0B1 is the UUID for the secret. In the event that you have forgotten the UUID, you can retrieve it. For example, to retrieve the UUID for `nightly_backup`:

```
$ okv managed-object object locate --name nightly_backup
```

4. Log in to the server that will use the shared secret (the destination server).
5. Create an endpoint, and then make the `shared_secrets` wallet the default wallet of that new endpoint:

```
okv admin endpoint create --endpoint secrets_db_name --description
"Endpoint for Secrets Management in db_name" --type ORACLE_NON_DB --
platform LINUX64
okv manage-access wallet set-default --wallet shared_secrets --endpoint
secrets_db_name
okv admin endpoint provision --endpoint secrets_db_name --location /
directory/for/new/endpoint/ --auto-login TRUE
```

6. Securely copy the script that you created (for example, `log-me-in.sh`) from its source server to the destination server that will use the shared secret.

```
$ scp log-me-in.sh destination_server:/remote_path
```

7. In the destination server, use the following syntax to run the script:

```
$ ./log-me-in.sh user_name hostname:port/service 'AS privilege'
```

For example:

```
$ ./log-me-in.sh nightly_backup 192.0.2.1:1521/hr_pdb 'AS SYSBACKUP'
```

Output similar to the following should appear:

```
Recovery Manager: Release 19.0.0.0.0 - Production on Fri Jun 26 13:10:49  
2020  
Version 19.7.0.0.0
```

```
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights  
reserved.
```

```
connected to target database: hr_db:hr_pdb (DBID=1932795327)
```

```
RMAN>
```

Related Topics

- [Example: Script for Storing and Retrieving Oracle Database User Passwords in and from Oracle Key Vault](#)
You can store passwords in Oracle Key Vault and retrieve them when needed.

Changing Passwords for a Large Database Deployment

For better security, on a regular basis, you should change passwords that are used by both humans and automated processes.

To change the passwords that are stored in Oracle Key Vault, you revoke and destroy the old password, and then upload and activate the new password. Finally, you must change the password in SQL*Plus.

1. Log in to a server that has access to the stored password.
2. Retrieve the UUID of this password.

```
$ okv managed-object object locate --name nightly_backup
```

UUID output similar to the following appears:

```
994CB3E5-C5B3-4F75-BFED-CB41AE15D0B1
```

3. Use this UUID to revoke this password.

```
$ okv managed-object object revoke --code CESSATION_OF_OPERATION --uuid
994CB3E5-C5B3-4F75-BFED-CB41AE15D0B1
```

You must provide a code for revoking the password. Possible values are as follows:

- UNSPECIFIED
- KEY_COMPROMISE
- CA_COMPROMISE
- AFFILIATION_CHANGED
- SUPERSEDED
- CESSATION_OF_OPERATION
- PRIVILEGE_WITHDRAWN

4. Destroy this password.

```
$ okv managed-object object destroy --uuid 994CB3E5-C5B3-4F75-BFED-
CB41AE15D0B1
```

5. In the Oracle Key Vault management console, delete the secret.

6. Create a text file to contain the new password, similar to the original password text file that was used to configure the original external keystore password.

```
$ more backup_pwd
cn0KpOnY9vNec2sLVHFnJwR6
```

7. Upload the new password and attributes, and then activate the new secret.

a. Generate JSON output.

```
okv managed-object secret register --generate-json-input >
reg_secret.json
```

b. Check the generated output.

```
$ more reg_secret.json

{
  "service" : {
    "category" : "managed-object",
    "resource" : "secret",
    "action" : "register",
    "options" : {
      "object" : "./backup_pwd",
      "type" : "PASSWORD",
      "mask" : "ENCRYPT",
      "attributes" : {
        "name" : "NIGHTLY_BACKUP"
      }
    }
  }
}
```

```
}  
}
```

- c. Perform the registration.

```
okv managed-object secret register --from-json ./reg_secret.json
```

- d. Activate the secret.

```
$ okv managed-object object activate --uuid 0739649D-3058-4F34-  
BF84-50B2BD652C2D
```

8. On Linux, securely delete the file that contain passwords.

```
shred -xz backup_pwd  
rm backup_pwd
```

9. In each Oracle database that shares the secret, log in to SQL*Plus and use the `password` command to change the password.

For example:

```
SYS> password nightly_backup  
Changing password for nightly_backup  
New password: cn0KpOnY9vNec2sLVHFnJwR6  
Retype new password: cn0KpOnY9vNec2sLVHFnJwR6  
Password changed
```

Related Topics

- [Destroying a Key or Security Object](#)
When a key is no longer used or compromised in some way, then you can destroy it.

Oracle Key Vault General System Administration

General system administration refers to system management tasks for the Oracle Key Vault server, such as configuring network details and services.

- [Overview of Oracle Key Vault General System Administration](#)
System administrators can perform most general administration tasks in the Oracle Key Vault management console, including finding the current status of the overall system.
- [Configuring Oracle Key Vault in a Non-Multi-Master Cluster Environment](#)
On the system Settings page, you can configure settings for either a standalone environment or a primary-standby environment.
- [Configuring Oracle Key Vault in a Multi-Master Cluster Environment](#)
When you configure Oracle Key Vault in a multi-master cluster environment, you can configure either individual nodes or the entire multi-master cluster environment.
- [Managing System Recovery](#)
System recovery includes tasks such as recovering lost administrative passwords.
- [Support for a Primary-Standby Environment](#)
To ensure that Oracle Key Vault can always access security objects, you can deploy Oracle Key Vault in a primary-standby (highly available) configuration.
- [Commercial National Security Algorithm Suite Support](#)
You can use scripts to perform Commercial National Security Algorithm (CNSA) operations for Oracle Key Vault HSM backup and upgrade operations.
- [Minimizing Downtime](#)
Business-critical operations require data to be accessible and recoverable with minimum downtime.

Overview of Oracle Key Vault General System Administration

System administrators can perform most general administration tasks in the Oracle Key Vault management console, including finding the current status of the overall system.

- [About Oracle Key Vault General System Administration](#)
System administrators configure the Oracle Key Vault system settings.
- [Viewing the Oracle Key Vault Dashboard](#)
The dashboard presents the current state of the Oracle Key Vault at a high level and is visible to all users.
- [Using the Status Panes in the Dashboard](#)
The status panes on the dashboard provide a high-level overview of the current state of Oracle Key Vault, including outstanding alerts, aggregated summary of managed contents, and also the state and status of various objects, entities and services.

About Oracle Key Vault General System Administration

System administrators configure the Oracle Key Vault system settings.

The Oracle Key Vault system settings include administration, local and remote monitoring, email notification, backup and recovery operations, and auditing. You must have the appropriate role for performing these tasks. Users who have the System Administrator role can perform most of the administrative tasks, and users with the Audit Manager role can configure audit settings, export audit records, and integrate Oracle Key Vault with Oracle Audit Vault. In most cases, you will perform these tasks in the Oracle Key Vault management console.

To quickly find information about the current state of the Oracle Key Vault system at a high level, you can view the Oracle Key Vault dashboard.

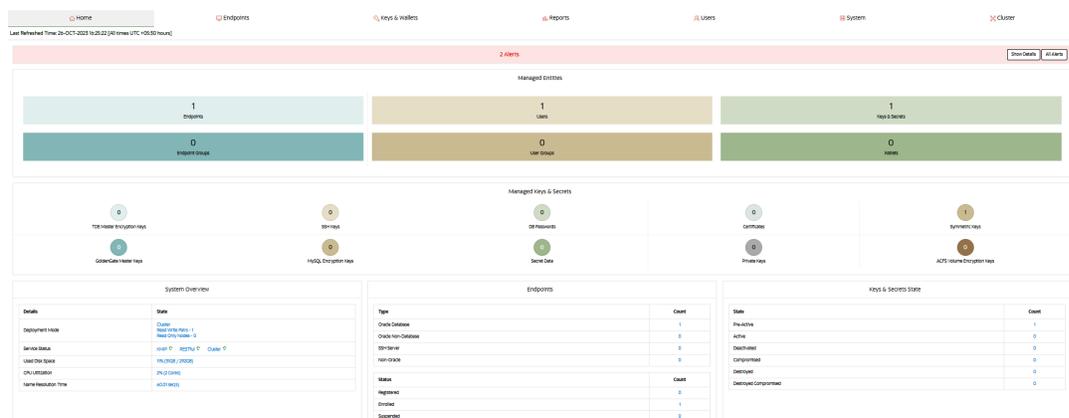
Related Topics

- [Managing Oracle Key Vault Users](#)
Oracle Key Vault users administer the system, enroll endpoints, manage users and endpoints, control access to security objects, and grant other users administrative roles.
- [Managing Oracle Key Vault Endpoints](#)
Oracle Key Vault endpoints are computer systems like database or application servers, where keys and credentials are used to access data.
- [Managing Oracle Key Vault Virtual Wallets and Security Objects](#)

Viewing the Oracle Key Vault Dashboard

The dashboard presents the current state of the Oracle Key Vault at a high level and is visible to all users.

- Click the **Home** tab to display the dashboard. By default, this page appears when you log in to the Oracle Key Vault management console.



Using the Status Panes in the Dashboard

The status panes on the dashboard provide a high-level overview of the current state of Oracle Key Vault, including outstanding alerts, aggregated summary of managed contents, and also the state and status of various objects, entities and services.

Log in to the Oracle Key Vault management console.

The dashboard appears in the **Home** tab.

The dashboard is organized into different panes. These dashboard panes display aggregated information about alerts, managed entities, security objects, and overall system overview of Oracle Key Vault.

The dashboard consists of:

1. Tabs at the top of the page for you to perform various administrative tasks. For example, to create a new endpoint, click the **Endpoints** tab.
2. The **Alerts** pane allows you to access existing alerts. For more information, click **Show Details** or **All Alerts**.

To take corrective action on an alert:

- a. Click **Show Details** to list a summary of the alerts.
- b. Click the link that corresponds to the alert to display the appropriate page.
- c. Take corrective action for the alert as necessary.

To configure alerts that you want to see on the dashboard:

- a. Click the **Reports** tab, and then click **Alerts** in the left side bar to display the **Alerts** page.
 - b. In the top right of the page, click **Configure** to display the Configure Alerts page.
 - c. Select the **Alert Type**, check **Enabled**, set the **Limit**, and then click **Save**.
3. The **Managed Entities** pane displays the aggregated information about these categories: Endpoints, Endpoint Groups, Users, User Groups, Keys & Secrets, and Wallets. For each category, the number of items that you configure for that respective category are shown.

For example, if the system has 21 endpoints, then 21 is displayed before the **Endpoints** label. To find and modify the details of these endpoints, click the **Endpoints** label.
 4. The **Managed Keys & Secrets** pane displays the aggregated information about different types of supported security objects: TDE Master Encryption Keys, MySQL Master Encryption Keys, DB Passwords, Secret Data, Certificates, Private Keys, Symmetric Keys, Opaque Objects, GoldenGate Master Keys, and ACFS Volume Encryption Keys. Similar to the Managed Entities contents, to find and modify the details about a security object, click the corresponding label.
 5. The **System Overview** pane provides information about the installation: Deployment Mode, Service Status, Used Disk Space, and CPU Utilization. For a multi-master cluster, the **System Overview** pane also displays the **Name Resolution Time** information.
 6. The **Endpoints** pane provides a count of endpoints broken down by their **Type** (Oracle Database, Oracle Non-Database, or Non-Oracle) and **Status** (Registered, Enrolled, or Suspended).
 7. The **Keys, Secrets State** pane provides a count of objects broken down by their object state: Pre-Active, Active, Deactivated, Compromised, Destroyed, and Destroyed Compromised.

In the home page, the item type and item state are displayed as of the last time refreshed. The number and count of entities and objects displayed may vary for different users depending upon their authorization.

Related Topics

- [Searching for Security Object Items](#)
You can search for individual security objects if you have privileges to view these objects.

Configuring Oracle Key Vault in a Non-Multi-Master Cluster Environment

On the system Settings page, you can configure settings for either a standalone environment or a primary-standby environment.

- [Configuring the Network Details](#)
Learn how to configure the network details from the Oracle Key Vault management console.
- [Configuring Network Access](#)
In a non-multi-master cluster, you can configure the network services from the Oracle Key Vault management console.
- [Configuring DNS](#)
You can configure up to three domain name service (DNS) servers with IP addresses that Oracle Key Vault will use to resolve host names.
- [Configuring the System Time](#)
Oracle strongly recommends that you synchronize Oracle Key Vault with an NTP time source.
- [Configuring FIPS Mode](#)
You can either enable or disable FIPS mode for Oracle Key Vault.
- [Configuring Syslog](#)
You can enable syslog for specific destinations and transmit the records either using Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).
- [Changing the Network Interface Mode](#)
You can switch between dual NIC mode and classic mode for the network interface.
- [Configuring RESTful Services Utility](#)
RESTful services utility allows you to automate the management of endpoints, wallets, security objects, deployment operations, and backup operations.
- [Checking the Oracle Audit Vault Integration Status](#)
Oracle Key Vault can send audit records to Oracle Audit Vault for centralized audit reporting and alerting.
- [Configuring the Oracle Key Vault Management Console Web Session Timeout](#)
You can configure a timeout value in minutes for the Oracle Key Vault management console Web session.
- [Restarting or Powering Off Oracle Key Vault](#)
You can manually restart or power off Oracle Key Vault as required for maintenance or for patch and upgrade procedures.

Configuring the Network Details

Learn how to configure the network details from the Oracle Key Vault management console.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In Network Details area, click **Network Info**.

The information displayed in the Network Details area depends on whether you are using Classic mode or Dual NIC mode.

 **Note:**

If you are logged in as a user with the monitoring privilege, then you can only view these screens in read-only mode with only the **Cancel** button visible.

Network Details ⊗

Host Name *

IP Address *

Network Mask *

Gateway

MAC Address **00:00:17:00:AF:2D**

Network Details ⊗

Host Name *

IP Address *

Network Mask *

Gateway

MAC Address **08:00:27:07:02:20**

Network Mode **DUAL-NIC**

Bonding Mode **Balance-RR**

Slave Interface	enp0s3	Link Status	↑	Link Speed	1000Mb/s
Slave Interface	enp0s8	Link Status	↑	Link Speed	1000Mb/s

 **Note:**

The Dual NIC screen displays the bonding mode and the current status of each interface.

4. Update the values for the following fields:
 - **Host Name:** Enter the name of the server.
 - **IP Address:** Enter the IP address of the server.
 - **Network Mask:** Enter the network mask of the server.
 - **Gateway:** Enter the network gateway of the server.

The fields in this pane are automatically populated with the IP address and host name of your Oracle Key Vault server. But if you want, then you can update the **Host Name**, **IP Address**, **Network Mask**, and the **Gateway** for the Oracle Key Vault installation.

You cannot change the **MAC Address** settings. If you are using Dual NIC mode, then the same applies to **Network Mode** settings, that is, you cannot change the `Bonding Mode`, `Active Interface`, and `Backup Interface` settings, but these values are useful if you want to check their status.

5. Click **Save**.

If you have a high availability configuration, then you must unpair the primary and standby Oracle Key Vault servers before changing the IP address. After you have changed the IP address of the primary or standby Oracle Key Vault server, pair the two servers again. After you complete the pairing process, reenroll the Oracle Key Vault endpoints to ensure that they are updated with the new IP addresses for both the primary and the standby Oracle Key Vault servers.

Configuring Network Access

In a non-multi-master cluster, you can configure the network services from the Oracle Key Vault management console.

You can enable services for **Web Access** and **SSH Access** (Secure Shell Access) for all, none, or a subset of clients, determined by their IP addresses.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Details area, select **Web Access**.

 **Note:**

If you are logged in as a user with the monitoring privilege, then you can only view the screen in read-only mode with only the **Cancel** button visible.

Web Access
ⓧ

Cancel Save

Web Access allowed from *

IP address(es) All

<IP Address1> <IP Address2> ...

4. Select one of the following options:
 - **All** to select all IP addresses
 - **IP Address(es)** to select a set of IP addresses that you specify in the next field, separating each IP address by a space. The **IP address(es)** web access option enables you to restrict access to the Oracle Key Vault management console to a limited set of IP addresses that you specify to meet your organizational needs.
5. Click **Save**.
6. To set the SSH access, in the Settings page under Network Details, select **SSH Access**.

Enabling **SSH Access** gives you access to Oracle Key Vault from the command line. This helps you diagnose problems not immediately apparent from the management console. You must log in as the user `support`, with the support password that you created during installation. SSH access is used only under the direction of Oracle Support, or when you upgrade.

Enabling or disabling SSH access will enable or disable the **inbound** SSH connection to the Oracle Key Vault server. Enabling or disabling SSH access in this manner has no bearing on the SSH Tunnel settings or any other outbound SSH connections that the Oracle Key Vault server itself establishes. SSH connections can still be established by the Oracle Key Vault to other servers as in the case of SSH Tunnel settings.

 **Note:**

Oracle recommends that you always disable SSH access, except when you are applying an Oracle Key Vault Release Update (RU), or when directed by Oracle Support.

The default port used for SSH access is 22. Starting with Oracle Key Vault release 21.10.0.0, you can configure this port to a value of your choice.

To change the port, select **Click here to configure** and change the **SSHD Port** port number in the Service Ports window.

7. In the SSH Access window, use the following settings:
 - **Disabled** to disable all IP addresses SSH access.
 - **IP Address(es)** to select a set of IP addresses that you specify in the next field, separating each IP address by a space.
 - **All** to enable SSH access for all IP addresses.
8. Click **Save**.
9. SSH Port shows the currently-configured port for SSH access. To change this port to a different value, select **Click here to configure**. On the Service Ports page, update the value for the following field:

SSHD Port: Enter the port to use for SSH access. Click **Change Port**. This takes a few minutes to complete.

If the port for SSH access is no longer the default of 22, you can SSH into the Oracle Key Vault system using a command similar to the following:

```
ssh support@<Oracle_Key_Vault_IP_Address> -p
<configured_port_for_SSH_access>
```

Configuring DNS

You can configure up to three domain name service (DNS) servers with IP addresses that Oracle Key Vault will use to resolve host names.

This configuration is useful if you only know the host name and not the IP address of a server that Oracle Key Vault needs to access. For example, while configuring the SMTP server for email notifications, you can optionally enter the host name instead of the IP address, after you set up DNS.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Services area, click **DNS**.

Server	IP address
DNS Server 1	192.0.2.70
DNS Server 2	192.0.2.1
DNS Server 3	192.0.2.2

4. Enter up to three IP addresses for DNS servers.
You must at minimum configure the DNS setting for Server 1. While only the first value is required, two entries are recommended for fault tolerance.
5. Click **Save**.

Configuring the System Time

Oracle strongly recommends that you synchronize Oracle Key Vault with an NTP time source.

Synchronizing Oracle Key Vault with a time source is important for reliable time stamps in audit records, and the activation, deactivation, protectStop, and processStart dates for keys and secrets.

You can configure Oracle Key Vault to synchronize its system time with the NTP servers. Oracle Key Vault provides fields to enter information for up to three NTP servers. If an NTP server is not available, then set the current time manually. Use the calendar icon to set the date and time so that these values are stored in the correct format.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Services area, click **NTP** to display the System Time window.

The following screen shows a partial view of the System Time window. For NTP servers, only use IP addresses and not host names. As a best practice, configure all three servers.

The screenshot shows the 'System Time' configuration window. At the top, there is a title bar with 'System Time' and a close button. Below the title bar, there is a 'Save' button. The main content area has two radio buttons: 'Set Manually' (unselected) and 'Use Network Time Protocol' (selected). Below this, there are three sections for NTP servers:

- NTP Server 1:** Address: 0.north-america.pool.ntp.org; Server Time (in UTC): 10-JAN-2024 13:10:04 (0.000138 seconds difference); Test Server button.
- NTP Server 2:** Address: 1.north-america.pool.ntp.org; Server Time (in UTC): (blank); Test Server button.
- NTP Server 3:** Address: 2.north-america.pool.ntp.org; Server Time (in UTC): (blank); Test Server button.

4. Choose **Use Network Time Protocol**.
5. Enter values for the following fields:
 - **Server 1:** Enter the IP address of an NTP server. You must supply an IP address for Server 1. To test the NTP server, click **Test Server**. To immediately synchronize the system time with this server, click **Apply Server**.
 - **Server 2:** Enter the IP address of a second NTP server. This value is optional. To test the NTP server, click **Test Server**. To immediately synchronize the system time with this server, click **Apply Server**.
 - **Server 3:** Enter the IP address of a third NTP server. This value is optional. To test the NTP server, click **Test Server**. To immediately synchronize the system time with this server, click **Apply Server**.

6. Click **Save**.

 **Note:**

To perform a synchronization of the Oracle Key Vault server with the NTP server, click the **Apply Server** button on the **System Time** page.

Related Topics

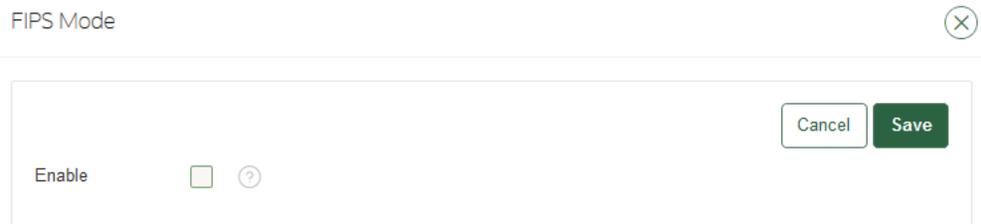
- [Configuring DNS](#)
You can configure up to three domain name service (DNS) servers with IP addresses that Oracle Key Vault will use to resolve host names.

Configuring FIPS Mode

You can either enable or disable FIPS mode for Oracle Key Vault.

In a primary-standby environment, ensure that both servers are consistent in their FIPS mode setting: either both are enabled, or both are disabled.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click FIPS.



FIPS Mode

Enable ?

Cancel Save

4. Do one of the following:
 - To enable FIPS mode, select the **Enable** check box.
 - To disable FIPS mode, clear the **Enable** check box.

Enabling or disabling FIPS mode will take a few minutes and will also restart the system automatically.

5. Click **Save**.

After you click **Save**, a confirmation dialog box will appear.

6. In the confirmation dialog box, click **OK** to apply the changes and restart the Oracle Key Vault system.

If you click **OK**, be aware that the operation cannot be canceled. The restart operation takes place immediately.

Configuring Syslog

You can enable syslog for specific destinations and transmit the records either using Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).

All system related alerts are sent to syslog.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **Syslog**.

The screenshot shows the Syslog configuration page. At the top, there is a 'Syslog' header with a close button (X). Below this is a 'Save' button. The main content area has two radio buttons: 'TCP' (which is selected) and 'UDP'. Below the radio buttons is a text input field labeled 'Syslog Destinations' containing the placeholder text '<IP Address1>:<port1> <IP Address2>:<port2> ...'.

4. Select one of the following protocols:
 - **TCP**: Enables syslog using the TCP protocol.
 - **UDP**: Enables syslog using the UDP protocol.
5. Enter the syslog destination IP addresses and port numbers in the **Syslog Destinations** field, in the format *IP_address:port*.
You can enter multiple destinations, separated by a space.
6. Click **Save**.

Note:

You can use syslog forwarding to send syslog messages (including audit records, if audit records are sent to syslog) to SIEM (Security Information and Event Management) solutions, such as Splunk.

Changing the Network Interface Mode

You can switch between dual NIC mode and classic mode for the network interface.

You can both switch the network mode and update the IP information of a standalone Oracle Key Vault server. In a primary-standby configuration, you cannot change the network mode and update the IP information. You need to first configure the network interface mode when you install the Oracle Key Vault appliance software.

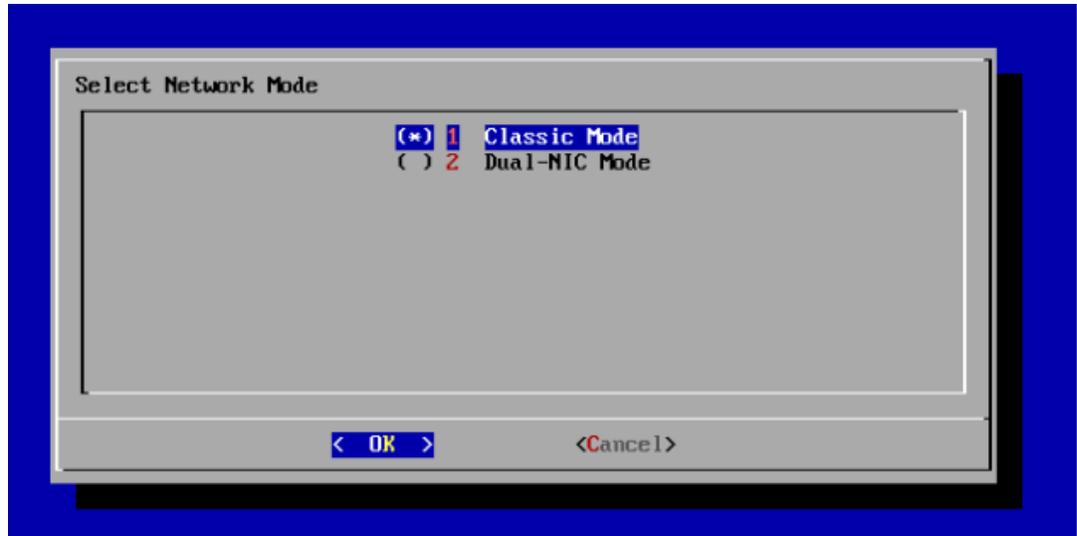
1. Log in to the Oracle Key Vault server as the `support` user.
2. Switch to the root user.

```
su - root
```

3. At the command line, run the following script:

```
/usr/local/okv/bin/okv_configure_network
```

4. In the Select Network Mode window, select the network interface that you want to use, and then select **OK**.



5. For the network mode, if you want Classic mode, then follow these steps:

Classic mode, the only mode available before 21.x releases of Oracle Key Vault, allows one network interface to be used. If you later decide to switch to dual NIC mode, then you can do so, but only if you are using a standalone configuration. You cannot change the mode if you are using a multi-master cluster or primary-standby configuration. Choose this option if the server has only one network card.

- a. Select **1** to choose Classic mode and then select **OK**.
 - b. In the Select default network interface screen, select from the available options, and then select **OK**.
 - c. In the Network settings screen, enter the **IP address**, **Network mask**, and **Gateway** settings for the default network interface. The network administrator for your site can provide this information.
 - d. Select **OK**.
6. If you want the dual NIC network mode, then follow these steps:

Dual NIC mode enables you to configure Oracle Key Vault to use two network interfaces, or ethernet ports. It is useful as a guard against physical or software failures and adds redundancy to the network layer. Select the dual NIC mode if there is a greater need for operational continuity and to avoid eviction from the cluster due to prolonged unavailability of the network. Dual NIC mode helps to prevent situations where a node may lose connectivity and risk missing changes that have been made to data in the cluster.

- a. Select **2** to select Dual-NIC mode and then select **OK**.

- b. In the Select Bond Mode screen, select from the bond mode choices for the two network interfaces that you plan to use, and then select **OK**.
 - **Round Robin** configures the network interfaces such that network packets are transmitted and received sequentially from the first available interface through the last. This bonding mode is the default. This mode provides fault tolerance and load balancing and requires the links to be connected to a network switch with EtherChannel support.
 - **Active-Backup** configures the network interfaces as active and backup. Only one interface in the bond is active. A different interface becomes active if, and only if, the active interface fails. The network communication happens over the active interface. This mode provides fault tolerance and does not require any switch support.
 - **802.3ad** creates aggregation groups that share the same speed and duplex settings. Network packets are transmitted and received on all interfaces. This mode provides fault tolerance and load balancing and requires a switch that supports IEEE 802.3ad dynamic link aggregation.
- c. In the Select two network interfaces screen, select the two network interfaces that you want, and then select **OK**.
- d. In the Network settings screen, enter the **IP address**, **Network mask**, **Gateway**, and **Hostname** settings for the default network interface. The network administrator for your site can provide this information. For the host name, use only lowercase characters. The host name can be the fully qualified host name or the short host name.
- e. Select **OK**.

You do not need to restart Oracle Key Vault after changing the network mode.

Related Topics

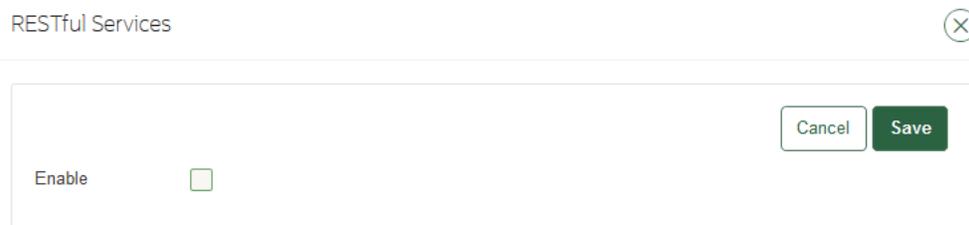
-

Configuring RESTful Services Utility

RESTful services utility allows you to automate the management of endpoints, wallets, security objects, deployment operations, and backup operations.

RESTful services utility also supports regular key management activities.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **RESTful Services**.



RESTful Services

Enable

Cancel Save

4. Select the **Enable** checkbox in the RESTful Services section.
To disable RESTful services, clear the **Enable** checkbox.

5. Click **Save**.

Oracle Key Vault provides the management of endpoint operations, virtual wallet operations, downloading and provisioning operations as RESTful services.

Click **Download** to download the Oracle Key Vault RESTful Service Utility, `okvrestclipackage.zip`, to use these services. Click **Download Classic Utility** to download the classic utility, `okvrestservices.jar`.

Checking the Oracle Audit Vault Integration Status

Oracle Key Vault can send audit records to Oracle Audit Vault for centralized audit reporting and alerting.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role or the Audit Manager role.
2. Depending on the roles that you have, navigate as follows:
 - **If you have the System Administrator role but not the Audit Manager role, or if you have both the System Administrator role and the Audit Manager roles:** Select the **System** tab, and then **Settings** from the left navigation bar. In the Monitoring and Alerts area, click **Audit Vault**.
 - **If you only have the Audit Manager role:** Select the **System** tab, and then in the left navigation bar, select **Audit Vault Integration**.
3. Select the **Monitoring** tab.

The Monitoring pane indicates if the monitoring is active. If Oracle Key Vault is not integrated with Oracle Audit Vault, then only the Deployment pane appears. If you want to have Audit Vault integrated, then log on as a user who has the Audit Manager role to perform the integration.

Related Topics

- [Configuring Oracle Key Vault with Oracle Audit Vault](#)
A user who has the Audit Manager role can configure Oracle Key Vault to send audit records to Oracle Audit Vault for centralized audit reporting and alerting.

Configuring the Oracle Key Vault Management Console Web Session Timeout

You can configure a timeout value in minutes for the Oracle Key Vault management console Web session.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Console Timeout**.
4. Enter the value in minutes for the timeout.

The default value is 10. The range you can enter is 1 through 100.

5. Click **Save**.

After you click **Save**, for the currently active user Web session as well as for other active sessions, this setting takes effect when the session is extended, the user refreshes the page, or the user navigates to another page. The user session remains active as long as

the user clicks a button, moves the mouse or presses a key, or is performing other activities. If the user session is idle for more than the management console timeout duration, then the user is logged out and the login screen appears.

Just before the Web session ends, starting earlier if the timeout value is larger, the user will be notified and is given the option to extend the session for the same length of time that was set for timeout value. For example, if the timeout was set to 20 minutes, then the user can extend the session for another 20 minutes.

Restarting or Powering Off Oracle Key Vault

You can manually restart or power off Oracle Key Vault as required for maintenance or for patch and upgrade procedures.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Status** from the left navigation bar.
3. Go to the top of the Status page.



4. Do one of the following:
 - To restart, click **Reboot**.
 - To power off, click **Power Off**.

Configuring Oracle Key Vault in a Multi-Master Cluster Environment

When you configure Oracle Key Vault in a multi-master cluster environment, you can configure either individual nodes or the entire multi-master cluster environment.

- [About Configuring Oracle Key Vault in a Multi-Master Cluster Environment](#)
You have the option of configuring settings for individual nodes or the entire multi-master cluster.
- [Configuring System Settings for Individual Multi-Master Cluster Nodes](#)
You set or change the settings for an individual multi-master cluster node from the Oracle Key Vault management console for that node.
- [Configuring System Settings for an Entire Oracle Key Vault Multi-Master Cluster](#)
You can set or change the settings for the entire multi-master cluster from the Oracle Key Vault management console of any node.

About Configuring Oracle Key Vault in a Multi-Master Cluster Environment

You have the option of configuring settings for individual nodes or the entire multi-master cluster.

Some settings are same for the entire multi-master cluster and they apply to all the cluster nodes. For such settings, you cannot configure different values for different nodes. Examples of such settings include Console Timeout and Maximum Disable Node Duration.

You can configure some settings only at the individual cluster node level. You must configure such settings on each cluster node individually. Examples of such settings include Network Info, and SSH Access.

You can configure some settings at both the individual cluster node level and the entire cluster level. If you configure the settings at both levels, values set at the cluster node level are effective. Examples of such settings include DNS, NTP, and SNMP

When you set a value for the entire cluster, it may take several minutes for changes to propagate to other nodes.

When you start the configuration from the **Settings** page, you can select the following **View Settings** menu options to filter the settings based on whether they can be set at the node level only, cluster level only, or both:

- **Node only** Shows settings that can only be configured at the individual node level. You configure such settings on each node individually. Examples of such settings include Network Info, and SSH Access.
- **Cluster only** Shows settings that are cluster-wide and updating them will change the settings for all cluster nodes. Examples of such settings include Alerts, Console Timeout, and Maximum Disable Node Duration.
- **Both** Shows settings that can be set at both the node level and the cluster level. If you configure the settings at both levels, values set at the node level are effective for that node. Examples of such settings include DNS, NTP, and SNMP.

You can navigate these settings between node and cluster settings using the right arrow button in the respective setting page. For example, if you select DNS, then you can configure DNS settings for either current node, or for the entire cluster, or both.

- **All** shows all the available settings without any filter.

Configuring System Settings for Individual Multi-Master Cluster Nodes

You set or change the settings for an individual multi-master cluster node from the Oracle Key Vault management console for that node.

These include settings that can be set at the:

- individual node level only
- individual node level as well as the entire-cluster level.

Values set for the individual node override the values set at the cluster level. You can clear any individual node level setting to revert to the cluster level setting.

Examples of these settings are network details, network access, system time, FIPS mode, syslog, and Oracle Audit Vault integration.

- [Configuring the Network Details for the Node](#)
In a multi-master cluster, you can change the host name for a node.
- [Configuring Network Access for the Node](#)
In a multi-master cluster, you can configure network access for a node.
- [Configuring DNS for the Node](#)
When you configure the DNS for a multi-master cluster node, you should enter more than one DNS IP address.

- [Configuring the System Time for the Node](#)
In a multi-master cluster, you must synchronize Oracle Key Vault with an NTP time source. All nodes of the cluster should operate on the same system time (or coordinated system time) for the inter node replication to work correctly.
- [Configuring the FIPS Mode for the Node](#)
All multi-master cluster nodes must use the same FIPS mode setting or you will receive an alert.
- [Configuring Syslog for the Node](#)
In a multi-master cluster node, you can enable syslog for specific destinations and transmit the records either using Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).
- [Changing the Network Interface Mode for the Node](#)
You can switch between dual NIC mode and classic mode for the network interface for a node in a multi-master cluster environment.
- [Configuring Auditing for the Node](#)
You can enable or disable audit settings for a node.
- [Configuring SNMP Settings for the Node](#)
You can enable or disable SNMP access for a multi-master cluster node.
- [Checking the Oracle Audit Vault Integration for the Node](#)
Oracle Key Vault can send audit records from a node to Oracle Audit Vault for centralized audit reporting and alerting.
- [Restarting or Powering Off Oracle Key Vault from a Node](#)
You can manually restart or power off an Oracle Key Vault node as required for maintenance or for patch and upgrade procedures.

Configuring the Network Details for the Node

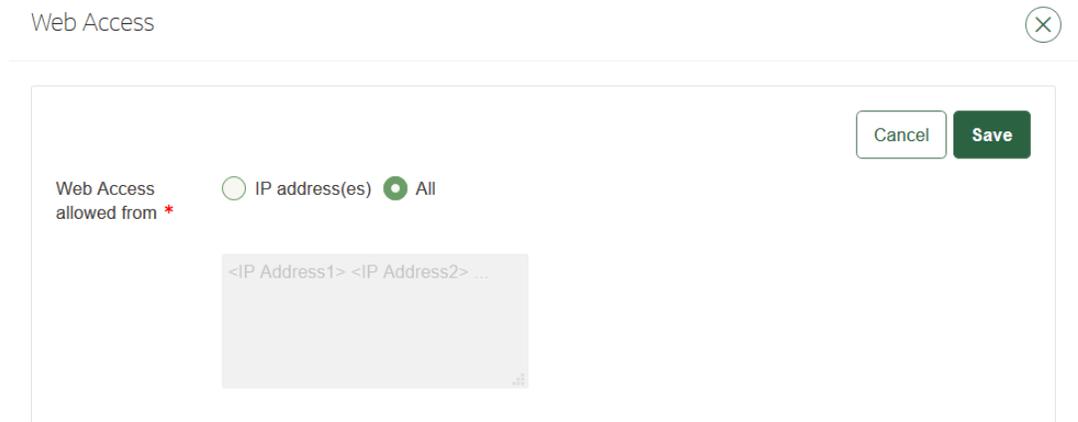
In a multi-master cluster, you can change the host name for a node.

1. Log into any Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Details area, select **Network Info**.
4. In the **Host Name** field, enter the name of the host name for the node.
You cannot modify the **IP Address**, **Network Mask**, **Gateway**, and **MAC Address** fields, which are automatically populated.
5. Click **Save**.

Configuring Network Access for the Node

In a multi-master cluster, you can configure network access for a node.

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Details area, select **Web Access**.



4. Select one of the following options:
 - **All** to select all IP addresses
 - **IP Address(es)** to select a set of IP addresses that you specify in the next field, separating each IP address by a space. The **IP address(es)** web access option enables you to restrict access to the Oracle Key Vault management console to a limited set of IP addresses that you specify to meet your organizational needs.
5. Click **Save**.
6. To set the SSH access, in the Settings page under Network Details, select **SSH Access**.
 Enabling **SSH Access** gives you access to Oracle Key Vault from the command line. This helps you diagnose problems not immediately apparent from the management console. You must log in as the user `support`, with the support password that you created during installation. SSH access is used only under the direction of Oracle Support, or when you upgrade.
 As a best practice, enable SSH access for short durations, solely for diagnostic, troubleshooting, or upgrade purposes, and then disable it as soon as you are done.
 Enabling or disabling SSH access will enable or disable the **inbound** SSH connection to the Oracle Key Vault server. Enabling or disabling SSH access in this manner has no bearing on the SSH Tunnel settings or any other outbound SSH connections that the Oracle Key Vault server itself establishes. SSH connections can still be established by the Oracle Key Vault to other servers as in the case of SSH Tunnel settings.
7. In the SSH Access window, enter the following settings:
 - **Disabled** to disable all IP addresses SSH access
 - **IP Address(es)** to select a set of IP addresses that you specify in the next field, separating each IP address by a space.
 - **All** to enable SSH access from all IP addresses.
8. Click **Save**.

Configuring DNS for the Node

When you configure the DNS for a multi-master cluster node, you should enter more than one DNS IP address.

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.

3. In the Network Services area, click **DNS**.
4. In the DNS window, ensure that you are on Node Details - Effective on this Node page.
If you are on the Cluster Details page, then click the arrow on the right to toggle back to Node Details - Effective on this Node.
5. In the Node Details - Effective on this Node window, enter up to three DNS server IP addresses.
You must at minimum configure the DNS setting for Server 1. While only the first value is required, two entries are recommended for fault tolerance.
6. Click **Save**.

Configuring the System Time for the Node

In a multi-master cluster, you must synchronize Oracle Key Vault with an NTP time source. All nodes of the cluster should operate on the same system time (or coordinated system time) for the inter node replication to work correctly.

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Services area, click **NTP**.
In cluster mode, you must use the Network Time Protocol (NTP), so you cannot change the selection from **Use Network Time Protocol** to **Set Manually**.
4. In the System Time window, ensure that you are on Node Details - Effective on this Node page.
If you are on the Cluster Details page, then click the arrow on the right to toggle back to Node Details - Effective on this Node.
5. In the Node Details - Effective on this Node page, enter values for the following fields:
 - **Server 1:** Enter the IP address of the first NTP server. To test the NTP server, click **Test Server**. To persist the changes to the NTP configuration, click **Apply Server**.
 - **Server 2:** Enter the IP address of the second NTP server. To test the NTP server, click **Test Server**. To persist the changes to the NTP configuration, click **Apply Server**.
 - **Server 3:** Enter the IP address of the third NTP server. To test the NTP server, click **Test Server**. To persist the changes to the NTP configuration, click **Apply Server**.
6. Click **Save** (or **Save to Cluster**).

Configuring the FIPS Mode for the Node

All multi-master cluster nodes must use the same FIPS mode setting or you will receive an alert.

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **FIPS**.
4. In the FIPS Mode window, do one of the following:
 - To enable FIPS mode, select the **Enable** check box.

- To disable FIPS mode, clear the **Enable** check box.

Enabling or disabling FIPS mode will take a few minutes and will also restart the system automatically.

5. Click **Save**.

After you click **Save**, Oracle Key Vault will restart automatically.

Configuring Syslog for the Node

In a multi-master cluster node, you can enable syslog for specific destinations and transmit the records either using Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **Syslog**.
4. In the Syslog window, ensure that you are on Node Details - Effective on this Node page. If you are on the Cluster Details page, then click the arrow on the right to toggle back to Node Details - Effective on this Node.
5. In the Node Details - Effective on this Node page, select one of the following protocols:
 - **TCP**: Enables syslog using the TCP protocol.
 - **UDP**: Enables syslog using the UDP protocol.
6. Enter the syslog destination IP addresses and port numbers in the **Syslog Destinations** field, in the format *IP_address:port*.

You can enter multiple destinations, separated by a space.

7. Click **Save**.

Note:

You can use syslog forwarding to send syslog messages (including audit records, if audit records are sent to syslog) to SIEM (Security Information and Event Management) solutions, such as Splunk.

Related Topics

- [Managing System Auditing](#)
Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.

Changing the Network Interface Mode for the Node

You can switch between dual NIC mode and classic mode for the network interface for a node in a multi-master cluster environment.

If you are using a primary-standby configuration, then you cannot change this mode. You first configured the network interface mode when you installed the Oracle Key Vault appliance software. Note that the cluster node must be disabled before you can change its network mode.

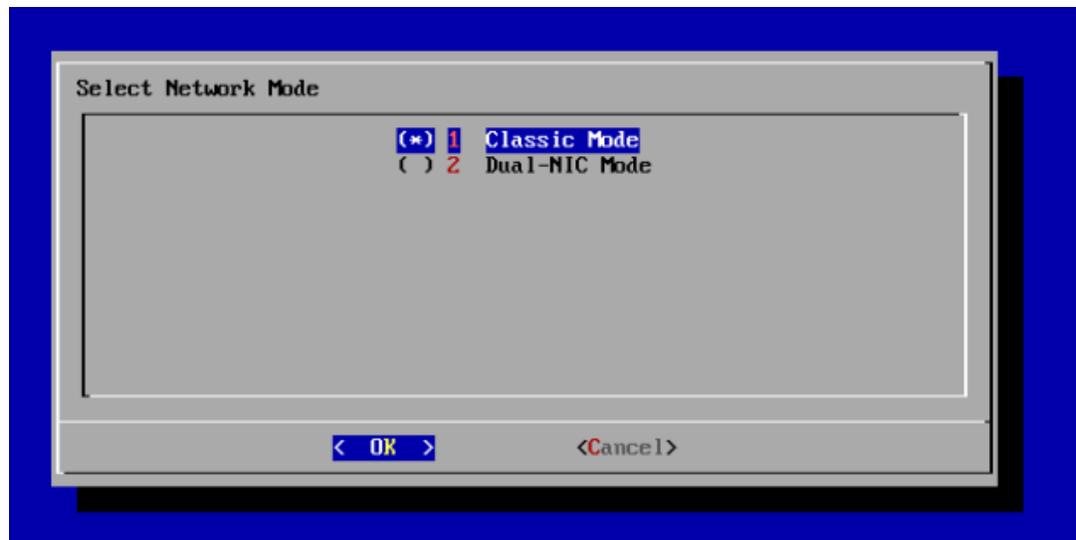
1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Disable the cluster node.
3. Log in to the Oracle Key Vault server through SSH as `support` user.
4. Switch user `su` to `root`.

```
ssh support@okv_server_IP_address  
su - root
```

5. At the command line, run the following script:

```
/usr/local/okv/bin/okv_configure_network
```

6. In the Select Network Mode window, select the network interface that you want to use, and then select **OK**.



7. For the network mode, if you want Classic mode, then follow these steps:

Classic mode, the only mode available before 21.x releases of Oracle Key Vault, allows one network interface to be used. If you later decide to switch to dual NIC mode, then you can do so, but only if you are using a standalone configuration. You cannot change the mode if you are using a multi-master cluster or primary-standby configuration. Choose this option if the server has only one network card.

- a. Select **1** to choose Classic mode and then select **OK**.
 - b. In the Select default network interface screen, select from the available options, and then select **OK**.
 - c. In the Network settings screen, enter the **IP address**, **Network mask**, and **Gateway** settings for the default network interface. The network administrator for your site can provide this information.
 - d. Select **OK**.
8. If you want the dual NIC network mode, then follow these steps:

Dual NIC mode enables you to configure Oracle Key Vault to use two network interfaces, or ethernet ports. It is useful as a guard against physical or software failures and adds redundancy to the network layer. Select the dual NIC mode if there is a greater need for

operational continuity and to avoid eviction from the cluster due to prolonged unavailability of the network. Dual NIC mode helps to prevent situations where a node may lose connectivity and risk missing changes that have been made to data in the cluster.

- a. Select **2** to select Dual-NIC mode and then select **OK**.
 - b. In the Select Bond Mode screen, select from the bond mode choices for the two network interfaces that you plan to use, and then select **OK**.
 - **Round Robin** configures the network interfaces such that network packets are transmitted and received sequentially from the first available interface through the last. This bonding mode is the default. This mode provides fault tolerance and load balancing and requires the links to be connected to a network switch with EtherChannel support.
 - **Active-Backup** configures the network interfaces as active and backup. Only one interface in the bond is active. A different interface becomes active if, and only if, the active interface fails. The network communication happens over the active interface. This mode provides fault tolerance and does not require any switch support.
 - **802.3ad** creates aggregation groups that share the same speed and duplex settings. Network packets are transmitted and received on all interfaces. This mode provides fault tolerance and load balancing and requires a switch that supports IEEE 802.3ad dynamic link aggregation.
 - c. In the Select two network interfaces screen, select the two network interfaces that you want, and then select **OK**.
 - d. In the Network settings screen, enter the **IP address**, **Network mask**, **Gateway**, and **Hostname** settings for the default network interface. The network administrator for your site can provide this information. For the host name, use only lowercase characters. The host name can be the fully qualified host name or the short host name.
 - e. Select **OK**.
9. Re-enable the disabled cluster node.

You do not need to restart Oracle Key Vault after changing the network mode.

Related Topics

- [Disabling a Cluster Node](#)
You can temporarily disable a cluster node, which is required for upgrades and maintenance.
- [Enabling a Disabled Cluster Node](#)
You can enable any cluster node that was previously disabled. You must perform this operation from the disabled node.
- *Oracle Key Vault Installation and Upgrade Guide*

Configuring Auditing for the Node

You can enable or disable audit settings for a node.

1. Log into the Oracle Key Vault management console for the node as a user who has the Audit Manager role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **Audit**.

You can enable or disable the following auditing categories:

- **Auto Purge Audit Records:** This setting will turn the auto purge of the auditing records on or off.

 **Note:**

The cluster level setting does not apply for **Auto Purge Audit Records**.

- **Enable Auditing:** This setting will turn the auditing on or off. Turning off this setting will not generate audit records.
 - **Send Audit Records to Syslog:** This setting writes the audit records to syslog. To enable this setting, you must first configure the syslog destination.
4. In the Audit Settings window for each of these categories, ensure that you are on Node Details - Effective on this Node page.
If you are on the Cluster Details page, then click the arrow on the right to toggle back to Node Details - Effective on this Node.
 5. In the Node Details - Effective on this Node page, select **Yes** or **No** for each category.
 6. Click **Save**.

Related Topics

- [Managing System Auditing](#)
Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.

Configuring SNMP Settings for the Node

You can enable or disable SNMP access for a multi-master cluster node.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **SNMP**.
4. In the SNMP window, ensure that you are on Node Details - Effective on this Node page.
If you are on the Cluster Details page, then click the arrow on the right to toggle back to Node Details - Effective on this Node.
5. In the Node Details - Effective on this Node page, select who has SNMP access to the multi-master cluster by choosing one of the options:
 - **All:** Allows SNMP access from all IP addresses.
 - **Disabled:** Allows no SNMP access.
 - **IP address(es):** Allows SNMP access from the list of IP addresses supplied in the address box. Enter a space-separated list of IP addresses.
6. Enter values for the following fields:
 - **Username:** Enter the SNMP user name.
 - **Password:** Enter the SNMP password.
 - **Reenter Password:** Enter the SNMP password again.
7. Click **Save**.

Alternatively, you can select **Delete** to remove the SNMP setting.

Checking the Oracle Audit Vault Integration for the Node

Oracle Key Vault can send audit records from a node to Oracle Audit Vault for centralized audit reporting and alerting.

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role or the Audit Manager.
2. Depending on the roles that you have, navigate as follows:
 - **If you have the System Administrator role but not the Audit Manager role, or if you have both the System Administrator role and the Audit Manager roles:** Select the **System** tab, and then **Settings** from the left navigation bar. In the Monitoring and Alerts area, click **Audit Vault**.
 - **If you only have the Audit Manager role:** Select the **System** tab, and then in the left navigation bar, select **Audit Vault Integration**.
3. Select the **Monitoring** tab.

The Monitoring pane indicates if the monitoring is active. If Oracle Key Vault is not integrated with Oracle Audit Vault, then only the Deployment pane appears. If you want to have Audit Vault integrated, then log on as a user who has the Audit Manager role to perform the integration.

Related Topics

- [Configuring Oracle Key Vault with Oracle Audit Vault](#)
A user who has the Audit Manager role can configure Oracle Key Vault to send audit records to Oracle Audit Vault for centralized audit reporting and alerting.

Restarting or Powering Off Oracle Key Vault from a Node

You can manually restart or power off an Oracle Key Vault node as required for maintenance or for patch and upgrade procedures.

When you restart or power-off Oracle Key Vault nodes, only the current node is restarted or powered-off. The other nodes in the cluster are unable to send information to and from the nodes that are powered off. When the nodes are restarted, there will be a period needed for the restarted nodes to catch up on activities that took place while they were down.

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Status** from the left navigation bar.
3. At the top of the Status page, do one of the following to restart or power off the node:
 - To restart, click **Reboot**.
 - To power off, click **Power Off**.

Configuring System Settings for an Entire Oracle Key Vault Multi-Master Cluster

You can set or change the settings for the entire multi-master cluster from the Oracle Key Vault management console of any node.

These include settings that can be set at the:

- Entire cluster level only.
 - Individual node-level and the entire cluster level.
- Values set at the cluster level do not override the values set the node level. To use cluster level setting, you need to clear any individual node level settings.

Example of these settings are console timeout, SNMP, system time, DNS, and auditing.

- [Configuring the System Time for the Cluster](#)
In a multi-master cluster, you can synchronize Oracle Key Vault with an NTP time source.
- [Configuring DNS for the Cluster](#)
When you configure the DNS for a cluster, you can enter up to three DNS server IP addresses.
- [Configuring the Maximum Disable Node Duration for the Cluster](#)
You can set the maximum disable node duration time for the cluster in hours.
- [Configuring Syslog for the Cluster](#)
In a multi-master cluster environment, you can enable syslog for specific destinations and transmit the records either using Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).
- [Configuring RESTful Services for the Cluster](#)
You can enable or disable RESTful Services for the cluster.
- [Configuring Auditing for the Cluster](#)
You can enable or disable audit settings for the cluster.
- [Configuring SNMP Settings for the Cluster](#)
You can enable or disable SNMP access for a multi-master cluster.
- [Configuring the Oracle Key Vault Management Console Web Session Timeout for the Cluster](#)
You can configure a timeout value in minutes for the Oracle Key Vault management console for all nodes in a multi-master cluster.
- [Configuring the Service Ports](#)
You can configure the service ports and change the default port based on your requirement.

Configuring the System Time for the Cluster

In a multi-master cluster, you can synchronize Oracle Key Vault with an NTP time source.

All nodes of the cluster should operate on the same system time (or coordinated system time) for the inter-node replication to work correctly.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Services area, click **NTP**.
4. In the System Time page, click the arrow to the left to toggle to the Cluster Details page.
5. In the Cluster Details page, enter values for the following fields:
 - **Server 1:** Enter the IP address of the first NTP server. To test the NTP server, click **Test Server**. To immediately synchronize the system time with this server, click **Apply Server**.

- **Server 2:** Enter the IP address of the second NTP server. To test the NTP server, click **Test Server**. To persist the changes to the NTP configuration, click **Apply Server**.
 - **Server 3:** Enter the IP address of the third NTP server. To test the NTP server, click **Test Server**. To persist the changes to the NTP configuration, click **Apply Server**.
6. Click **Save**.

Configuring DNS for the Cluster

When you configure the DNS for a cluster, you can enter up to three DNS server IP addresses.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Network Services area, click **DNS**.
In the DNS window, ensure that you are on **Cluster Details** page.
4. If you are on the Node Details - Effective on the Node page, click the arrow on the right to toggle back to Cluster Details page.
5. In the Cluster Details page, enter up to three DNS Server IP addresses.
You must at minimum configure the DNS setting for Server 1. While only the first value is required, two entries are recommended for fault tolerance.
6. Click **Save**.

Configuring the Maximum Disable Node Duration for the Cluster

You can set the maximum disable node duration time for the cluster in hours.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Maximum Disable Node Duration**.
4. In the Maximum Disable Node Duration window, enter a value, in hours, for the duration that a node can be disabled before it is evicted from the cluster.
Note that as this value is increased, the average amount of disk space consumed by cluster-related data also increases.
5. Click **Save**.

Note:

If you are logged in as a user with the monitoring privilege, you can only view the page in read-only mode. Only the **Cancel** button is visible.

Configuring Syslog for the Cluster

In a multi-master cluster environment, you can enable syslog for specific destinations and transmit the records either using Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitor and Alerts area, click **Syslog**.
4. In the Syslog page, click the arrow to the left to toggle to the Cluster Details page.
5. In the Cluster Details page, select one of the following protocols:
 - **TCP**: Enables syslog using the TCP protocol.
 - **UDP**: Enables syslog using the UDP protocol.
6. Enter the syslog destination IP addresses and port numbers in the **Syslog Destinations** field, in the format *IP_address:port*.
You can enter multiple destinations, separated by a space.
7. Click **Save**.

 **Note:**

You can use syslog forwarding to send syslog messages (including audit records, if audit records are sent to syslog) to SIEM (Security Information and Event Management) solutions, such as Splunk.

Related Topics

- [Managing System Auditing](#)
Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.

Configuring RESTful Services for the Cluster

You can enable or disable RESTful Services for the cluster.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **RESTful Services**.
4. In the RESTful Services window, select the **Enable** check box.
5. Click **Save**.
6. Click **Download** to download the Oracle Key Vault RESTful Service Utility, `okvrestclipackage.zip`, to use these services.
7. Click **Download Classic Utility** to download the classic utility, `okvrestservices.jar`.

Configuring Auditing for the Cluster

You can enable or disable audit settings for the cluster.

1. Log into any Oracle Key Vault management console as a user who has the Audit Manager role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **Auditing**.

You can enable or disable the following auditing categories:

- **Enable Auditing:** This setting will turn the auditing on or off. Turning off this setting will not generate audit records.
 - **Replicate Audit Records:** This setting applies only to a cluster configuration. It indicates if the audit records are replicated across the cluster nodes.
 - **Send Audit Records to Syslog:** This setting writes the audit records to syslog. To enable this setting, you must first configure the syslog destination.
 - **Auto Purge Audit Records:** This setting writes the auto purge audit records. The cluster level setting does not apply to Auto Purge Audit Records.
4. In the Audit Settings page, for each of these categories, click the arrow to the left to toggle to the Cluster Details page.
 5. In the Cluster Details page, select **Yes** or **No** for each category.
 6. Click **Save**.

Related Topics

- [Managing System Auditing](#)
Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.

Configuring SNMP Settings for the Cluster

You can enable or disable SNMP access for a multi-master cluster.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **SNMP**.
4. Select who has SNMP access to the multi-master cluster by choosing one of the options:
 - **All:** Allows SNMP access from all IP addresses.
 - **Disabled:** Allows no SNMP access.
 - **IP address(es):** Allows SNMP access from the list of IP addresses supplied in the address box. Enter a space-separated list of IP addresses.
5. Enter values for the following fields:
 - **Username:** Enter the SNMP user name.
 - **Password:** Enter the SNMP password.
 - **Reenter Password:** Enter the SNMP password again.
6. Click **Save**.

Alternatively, you can select **Delete** to remove the SNMP setting.

Configuring the Oracle Key Vault Management Console Web Session Timeout for the Cluster

You can configure a timeout value in minutes for the Oracle Key Vault management console for all nodes in a multi-master cluster.

1. Log into any Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Console Timeout**.
4. In the Management Console Timeout window, enter the value in minutes for the timeout. The default value is 10. The range you can enter is 1 through 100.

5. Click **Save**.

After you click **Save**, the setting is applied to all nodes in the cluster. For the currently active user Web session as well as for other active sessions, this setting takes effect when the session is extended, the user refreshes the page, or the user navigates to another page. The user session remains active as long as the user clicks a button, moves the mouse or presses a key, or is performing other activities. If the user session is idle for more than the management console timeout duration, then the user is logged out and the login screen appears.

Just before the Web session ends, the user will be notified, starting earlier if the timeout value is larger, and is given the option to extend the session for the same length of time that was set for timeout value. For example, if the timeout was set to 20 minutes, then the user can extend the session for another 20 minutes.

Configuring the Service Ports

You can configure the service ports and change the default port based on your requirement.

- [Guidelines for Service Ports Configuration](#)
Consider these Oracle Key Vault guidelines for service ports configuration.
- [Configuring Cluster Services Ports](#)
You can configure the Service Ports for cluster services from the Oracle Key Vault management console.
- [Configuring Node Service Ports](#)
You can change the default service port number to your specifications using the Service Ports feature in Oracle Key Vault.

Guidelines for Service Ports Configuration

Consider these Oracle Key Vault guidelines for service ports configuration.

- The cluster service port changes are applied to nodes in pairs.
- Only a System Administrator can change the ports.
- You cannot perform a CA certificate rotation when the cluster service port configuration is in process as it affects the replication process between the nodes.
- Similarly, you cannot reconfigure the ports during the CA certificate rotation process.
- You cannot add a new node during the cluster service port change process.
- You cannot change the port if a new node is being added.
- You can delete or force delete any node during the port change process.
- You cannot change a port for a node that is disabled.
- After you restore a backup, the Oracle Key Vault ports are restored to the original ports.

Configuring Cluster Services Ports

You can configure the Service Ports for cluster services from the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab and then **Settings** from the left navigation bar.
3. In the Network Details area, select **Service Ports**. The Service Ports window appears.

4. For changing the cluster services ports for the cluster nodes, enter the port number in the **Read-Write Peer Listener Port** field.
5. Click **Change Port**. The system starts preparing all the nodes for the newly configured port.

Note:

You can stop the node change process using the **Abort** button that appears once the node change process starts.

6. The Cluster Service Ports window pane **Status** column displays the following status during the node change process.
 - **Ready to enable port *port number*.**
 - **Enabling port *port number***
 - **Enabled port *port number***
 - **IN USE**
7. After the node change completion, the changed port number becomes the default port number for the new nodes being added.

Configuring Node Service Ports

You can change the default service port number to your specifications using the Service Ports feature in Oracle Key Vault.

To configure the service ports for the node service in Oracle Key Vault:

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab and then **Settings** from the left navigation bar.
3. In the Network Details area, select **Service Ports**. The Service Ports window appears.

4. Enter the port number in the **SSH Port** text box.
5. Click **Change Port**.

 **Note:**

The port number changes to the new port number and henceforth this new changed port number is considered as default service port.

Managing System Recovery

System recovery includes tasks such as recovering lost administrative passwords.

- [About Managing System Recovery](#)
To perform system recovery, you use the recovery passphrase.
- [Recovering Credentials for Administrators](#)
You can recover the system by adding credentials for administrative users.
- [Changing the Recovery Passphrase in a Non-Multi-Master Cluster Environment](#)
Periodically changing the recovery passphrase is a good security practice.
- [Changing the Recovery Passphrase in a Multi-Master Cluster](#)
Changing the recovery passphrase in a multi-master cluster is a two-step process.

About Managing System Recovery

To perform system recovery, you use the recovery passphrase.

In an emergency when no administrative users are available, or you must change the password of administrative users, you can recover the system with the current recovery passphrase of Oracle Key Vault. In addition, you can change the recovery passphrase to keep up with security best practices.

Recovering Credentials for Administrators

You can recover the system by adding credentials for administrative users.

1. From a web browser using HTTPS, enter the IP address of the Oracle Key Vault server.
2. In the Oracle Key Vault login page, **do not log in**.
3. Click the **System Recovery** link at the lower right corner of the login page.

A new login page appears with a single field: **Recovery Passphrase**.

4. In the **Recovery Passphrase** field, enter the recovery passphrase and then click **Login**.
5. In the page that appears, select the **Administrator Recovery** tab to display the Administrator Management page.
6. In the **User Name** field, enter the name of the Oracle Key Vault administrative user and then click **Search**.

The Administrator Recovery page expands to show the **Allow Forward Grant** check box for each role that is currently granted to the user. For example:

Administrator Recovery Administrator Management Recovery Passphrase User Password Recovery

Administrator Management Modify

User Name

<input checked="" type="checkbox"/>	Audit Manager	<input checked="" type="checkbox"/>	Allow Forward Grant
<input checked="" type="checkbox"/>	Key Administrator	<input checked="" type="checkbox"/>	Allow Forward Grant
<input checked="" type="checkbox"/>	System Administrator	<input checked="" type="checkbox"/>	Allow Forward Grant

7. Select or deselect each role to be granted to the administrative user. For each selected role, if you want to allow the user to grant the role to other users, then select the **Allow Forward Grant** check box.
8. Click **Modify**.

Changing the Recovery Passphrase in a Non-Multi-Master Cluster Environment

Periodically changing the recovery passphrase is a good security practice.

A user with the System Administrator role should perform a new backup whenever the recovery passphrase changes, so that there is always a backup protected with the current recovery passphrase. This ensures that you will have at least one backup with the latest data.

1. Perform a server backup.
2. From a web browser, enter the IP address of your Oracle Key Vault server.
3. In the Oracle Key Vault login page, **do not log in**.
4. Click the **System Recovery** link at the lower right corner of the login page.
A new login page appears with a single field: **Recovery Passphrase**.
5. In the **Recovery Passphrase** field, enter the recovery passphrase and then click **Login**.
6. In the page that appears, select the **Administrator Recovery** tab.
7. Select the **Recovery Passphrase** tab.

8. In the Recovery Passphrase page, enter and re-enter a new password for the recovery passphrase.
9. Click **Submit**.
10. Perform a server backup.

Changing the Recovery Passphrase in a Multi-Master Cluster

Changing the recovery passphrase in a multi-master cluster is a two-step process.

- [About Changing the Recovery Passphrase for a Multi-Master Cluster](#)
To change the recovery passphrase for a multi-master cluster, you must first initiate the recovery passphrase change from one of the cluster node. Other cluster nodes are notified of the impending passphrase change.
- [Step 1: Initiate the Recovery Passphrase Change Across the Nodes](#)
You must initiate the change for the recovery passphrase so that all nodes in the multi-master cluster will be notified of the impending change.
- [Step 2: Change the Recovery Passphrase](#)
After the multi-master cluster nodes have been notified of the impending recovery passphrase change, you can change the recovery passphrase.

About Changing the Recovery Passphrase for a Multi-Master Cluster

To change the recovery passphrase for a multi-master cluster, you must first initiate the recovery passphrase change from one of the cluster node. Other cluster nodes are notified of the impending passphrase change.

When cluster nodes are ready to accept the new recovery password, set the recovery phrase on each node in the following order:

1. Set the new recovery passphrase on all nodes other than the one which initiated the recovery passphrase change.
2. After the new recovery passphrase change on other nodes has been set, set the new recovery passphrase for the node that initiated the recovery passphrase change.
3. Set the same passphrase on every node.

Step 1: Initiate the Recovery Passphrase Change Across the Nodes

You must initiate the change for the recovery passphrase so that all nodes in the multi-master cluster will be notified of the impending change.

A user with the System Administrator role should perform a new backup whenever the recovery passphrase changes. This is so that there is always a backup protected with the current recovery passphrase. This ensures that you will have at least one backup with the latest data.

1. Perform a server backup.
2. Ensure that all nodes are in the `ACTIVE` state and replication has been verified between all nodes. Ensure that there are no cluster operations going on (such as adding a node).
3. From a web browser, enter the IP address of a multi-master cluster node that is not in read-only restricted mode.
4. In the Oracle Key Vault login page, **do not log in**.
5. Click the **System Recovery** link at the lower right corner of the login page.

A new login page appears with a single field: **Recovery Passphrase**.

6. In the **Recovery Passphrase** field, enter the recovery passphrase and then click **Login**.
7. Click the **Recovery Passphrase** tab.
8. Click the **Initiate Change** button.
9. Log out.
10. Wait 3 to 4 minutes before continuing.

During this time, all nodes will be notified that a passphrase change will be performed. To cancel a passphrase change, click the **Reset** button.

All nodes will determine if more than one passphrase change has been initiated. If more than one passphrase change has been initiated, conflict resolution will be performed.

After you cancel a passphrase change by using the **Reset** button, Oracle recommends that you remedy the issue and again initiate a passphrase change, making sure to then change the passphrase on every node in the cluster.

Related Topics

- [Step 2: Change the Recovery Passphrase](#)
After the multi-master cluster nodes have been notified of the impending recovery passphrase change, you can change the recovery passphrase.

Step 2: Change the Recovery Passphrase

After the multi-master cluster nodes have been notified of the impending recovery passphrase change, you can change the recovery passphrase.

Follow the mentioned guidelines during the recovery passphrase change across the cluster.

- The new recovery passphrase is set on each node.
- The new recovery passphrase is set on all the nodes first other than the initiator node, that is, the node that has initiated the recovery passphrase change.
- Set the new recovery passphrase on the initiator node in the end.
- All the nodes are using the same Recovery passphrase.

Before changing the **Recovery Passphrase**, ensure that the node is ready to accept the new passphrase. This is indicated by the presence of the fields to accept the new password on the **Recovery Passphrase** tab of the **System Recovery** page.

1. From a Web browser, enter the IP address of a multi-master cluster node in the Oracle Key Vault installation.

You can find a list of available nodes in the Oracle Key Vault management console by selecting the Clusters tab and then checking the Cluster Details section.

2. In the Oracle Key Vault login page, **do not log in**.
3. Click the **System Recovery** link at the lower right corner of the login page.

A new login page appears with a single field: **Recovery Passphrase**.

4. In the **Recovery Passphrase** field, enter the recovery passphrase and then click **Login**.
5. Click the **Recovery Passphrase** tab.
6. Enter the new recovery passphrase in the two fields.
7. Click **Submit**.

- Repeat these steps for each node in the cluster. Ensure that you perform these steps on the initiator node last.

 **Note:**

HSM reverse migrate cannot run when the recovery passphrase is being changed.

 **Caution:**

It is your responsibility to keep the recovery passphrase the same on all nodes in the cluster. If you set the recovery passphrase differently on cluster nodes it will negatively impact cluster functionality, such as adding nodes and HSM-enabling nodes. In addition to the addition of nodes and nodes being HSM-enabled, certificate rotation in a multi-master cluster depends on all nodes having the same recovery passphrase.

Related Topics

- [Step 1: Initiate the Recovery Passphrase Change Across the Nodes](#)
You must initiate the change for the recovery passphrase so that all nodes in the multi-master cluster will be notified of the impending change.

Support for a Primary-Standby Environment

To ensure that Oracle Key Vault can always access security objects, you can deploy Oracle Key Vault in a primary-standby (highly available) configuration.

This configuration also supports disaster recovery scenarios.

You can deploy two Oracle Key Vault servers in a primary-standby configuration. The primary server services the requests that come from endpoints. If the primary server fails, then the standby server takes over after a configurable preset delay. This configurable delay ensures that the standby server does not take over prematurely in case of short communication gaps.

The primary-standby configuration was previously known as the high availability configuration. The primary-standby configuration and the multi-master cluster configuration are mutually exclusive.

Oracle Key Vault supports primary-standby read-only restricted mode. When the primary server is affected by server, hardware, or network failures, primary-standby read-only restricted mode ensures that an Oracle Key Vault server is available to service endpoints, thus ensuring operational continuity. However, key and sensitive operations, such as generation of keys are disabled, while operations such as generation of audit logs are unaffected.

When an unplanned shutdown makes the standby server unreachable, the primary server is still available to the endpoints in read-only mode.

Related Topics

- [About the Oracle Key Vault Primary-Standby Configuration](#)
You configure a primary-standby environment by providing the primary and standby servers with each other's IP address and certificate, and then pairing them.

Commercial National Security Algorithm Suite Support

You can use scripts to perform Commercial National Security Algorithm (CNSA) operations for Oracle Key Vault HSM backup and upgrade operations.

- [About Commercial National Security Algorithm Suite Support](#)
You can configure Oracle Key Vault for compliance with the Commercial National Security Algorithm (CNSA) Suite.
- [Running the Commercial National Security Algorithm Scripts](#)
You configure Oracle Key Vault to use the Commercial National Security Algorithm (CNSA) suite by running CNSA scripts.
- [Performing Backup Restore Operations with CNSA](#)
After you restore a backup of the Oracle Key Vault that was configured to use the enhanced Commercial National Security Algorithm (CNSA) Suite, use `/usr/local/okv/bin/okv_cnsa` to reconfigure CNSA compliance.
- [Upgrading a Standalone Oracle Key Vault Server with CNSA](#)
You can upgrade a standalone Oracle Key Vault while using Commercial National Security Algorithm (CNSA) compliance by upgrading and then executing the `okv_cnsa` script.
- [Upgrading Primary-Standby Oracle Key Vault Servers to Use CNSA](#)
You can upgrade Oracle Key Vault primary-standby servers while using Commercial National Security Algorithm (CNSA) compliance by upgrading and then executing the `okv_cnsa` script.

About Commercial National Security Algorithm Suite Support

You can configure Oracle Key Vault for compliance with the Commercial National Security Algorithm (CNSA) Suite.

This compliance applies to TLS connections to and from the Oracle Key Vault appliance.

The CNSA suite is a list of strong encryption algorithms and key lengths, that offer greater security and relevance into the future.

Oracle Key Vault release 12.2 BP3 or later do not provide complete compliance across every component in the system. You will be able to switch to the CNSA algorithms, where available by means of the following scripts that are packaged with the Oracle Key Vault ISO:

- `/usr/local/okv/bin/okv_cnsa` makes configuration file changes to update as many components as possible to use the enhanced algorithms.
- `/usr/local/okv/bin/okv_cnsa_cert` regenerates CNSA compliant public key pairs and certificates.

 **Note:**

The `/usr/local/okv/bin/okv_cnsa` and `/usr/local/okv/bin/okv_cnsa_cert` scripts are both disruptive because they replace the old key pairs with new ones. This has consequences for the following operations:

- **Endpoint Enrollment:** Enroll endpoints after running this script when possible. If you had endpoints enrolled before running the CNSA script, you must reenroll them so that fresh CNSA compliant keys are generated using CNSA algorithms.
- **Primary-Standby:** Run the CNSA scripts on both Oracle Key Vault instances before pairing them in a primary-standby configuration when possible. If you had primary-standby before you run the CNSA scripts, then you must re-configure primary-standby as follows: unpair the primary and standby servers, reinstall the standby server, run the CNSA scripts individually on each server, and then pair them again.

 **Limitations:**

- CNSA compliance is not supported for all components in the Oracle Key Vault infrastructure (for example, SSH or Transparent Data Encryption (TDE)).
- The Firefox browser is not supported for use with the Oracle Key Vault management console when CNSA is enabled. This is because the Firefox browser does not support CNSA-approved cipher suites.

Running the Commercial National Security Algorithm Scripts

You configure Oracle Key Vault to use the Commercial National Security Algorithm (CNSA) suite by running CNSA scripts.

1. Back up Oracle Key Vault.
2. If necessary, enable SSH access.

Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Select the **System** tab, then **Settings**. In the Network Details area, click **SSH Access**. Select **IP address(es)** and then enter only the IP addresses that you need, or select **All**. Click **Save**.

3. SSH into the Oracle Key Vault server as the `support` user, entering the `support` user password that was created during post-installation, when prompted.

```
$ ssh support@okv_instance
```

4. Change to the `root` user:

```
$ su root
```

5. Run the scripts as follows:

```
root# /usr/local/okv/bin/okv_cnsa
root# /usr/local/okv/bin/okv_cnsa_cert
```

6. Disable SSH access and then restart the Oracle Key Vault server.

Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Select the **System** tab, then **Settings**. In the Network Details area, click **SSH Access**. Select **Disabled**. Click **Save**. Restart the Oracle Key Vault server by selecting the **System** tab, then **Status**.

The scripts update the `/usr/local/okv/etc/okv_security.conf` with the following line:

```
USE_ENHANCED_ALGORITHMS_ONLY="1"
```

Related Topics

- [Backup and Restore Operations](#)
Backups provide the ability to restore Oracle Key Vault to a previous state in the case of a failure.

Performing Backup Restore Operations with CNSA

After you restore a backup of the Oracle Key Vault that was configured to use the enhanced Commercial National Security Algorithm (CNSA) Suite, use `/usr/local/okv/bin/okv_cnsa` to reconfigure CNSA compliance.

1. Perform the backup restore operations.
2. Wait until the restore operation is complete and the system has restarted.

Do not proceed without completing this step.

3. SSH into the Oracle Key Vault server as the `support` user:

```
$ ssh support@okv_instance
```

4. Switch to the `root` user:

```
$ su root
```

5. Run the following CNSA script :

```
root# /usr/local/okv/bin/okv_cnsa
```

Related Topics

- [Backup and Restore Operations](#)
Backups provide the ability to restore Oracle Key Vault to a previous state in the case of a failure.

Upgrading a Standalone Oracle Key Vault Server with CNSA

You can upgrade a standalone Oracle Key Vault while using Commercial National Security Algorithm (CNSA) compliance by upgrading and then executing the `okv_cnsa` script.

1. Ensure that you have backed up the server you are upgrading so your data is safe and recoverable.

Do not proceed without completing this step.

2. Log into the Oracle Key Vault management console as a user who has the System Administrator role.

3. If necessary, enable SSH access.

Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Select the **System** tab, then **Settings**. In the Network Details area, click **SSH Access**. Select **IP address(es)** and then enter only the IP addresses that you need, or select **All**. Click **Save**.

4. Ensure you have enough space in the destination directory for the upgrade ISO files.
5. Log in to the Oracle Key Vault server through SSH as user support, then switch user `su` to `root`.
6. Copy the upgrade ISO file to the destination directory using Secure Copy Protocol or other secure transmission method.

```
scp remote_host:remote_path/okv-upgrade-disc-  
new_software_release_number.iso /var/lib/oracle/destination_directory_for_iso_file
```

In this specification:

- `remote_host` is the IP address of the computer containing the ISO upgrade file.
- `remote_path` is the location of the ISO upgrade file.

7. Make the upgrade accessible by using the `mount` command:

```
root# /bin/mount -o loop,ro /var/lib/oracle/okv-upgrade-disc-  
new_software_release_number.iso /images
```

8. Clear the cache using the `clean all` command:

```
root# yum -c /images/upgrade.repo clean all
```

9. Run the following upgrade ruby script:

```
root# /usr/bin/ruby/images/upgrade.rb --confirm
```

If the system is successfully upgraded, then the command will display the following message:

```
Remove media and reboot now to fully apply changes
```

If you see an error message, then check the log file `/var/log/messages` for additional information.

10. Run the first CNSA script, which is available from the Oracle Key Vault ISO files location:

```
root# /usr/local/okv/bin/okv_cnsa
```

11. Restart the Oracle Key Vault server:

```
root# /sbin/reboot
```

On the first restart of the computer after the upgrade, the system will apply the necessary changes. This can take a few hours. Do not shut down the system during this time.

The upgrade is completed when the screen with heading: Oracle Key Vault Server `release_number` installation has completed. The `release_number` value should reflect the upgraded release.

12. Confirm that Oracle Key Vault has been upgraded to the correct version.
 - a. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
 - b. Select the **System** tab, and then select **Status**.
 - c. Verify that the version displayed is that of the new software release.

The release number is also at the bottom of each page, to the right of the copyright information.

13. Disable SSH access.

Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Select the **System** tab, then **Settings**. In the Network Details area, click **SSH Access**. Select **Disabled**. Click **Save**.

Related Topics

- <https://cryptome.org/2016/01/CNSA-Suite-and-Quantum-Computing-FAQ.pdf>

Upgrading Primary-Standby Oracle Key Vault Servers to Use CNSA

You can upgrade Oracle Key Vault primary-standby servers while using Commercial National Security Algorithm (CNSA) compliance by upgrading and then executing the `okv_cnsa` script.

You must perform the upgrade standby and primary servers in one session with as little time between the standby and primary upgrade as possible. The upgrade time is approximate and a function of the volume of data stored and managed by Oracle Key Vault. For large volumes of data, the upgrade time may be longer than several hours.

1. Prepare for the upgrade.
 - While the upgrade is in progress, do not change any settings or perform any other operations that are not part of the following upgrade instructions:
 - Upgrade the Oracle Key Vault server during a planned maintenance window because the upgrade process requires the endpoints to be shut down during the upgrade, if no persistent cache has been configured. With persistent cache enabled, endpoints will continue to be operational during the upgrade process.
 - Ensure that both the primary and standby systems have 8 GB memory.

2. Ensure that you have backed up the server you are upgrading so your data is safe and recoverable.

Ensure that in the time between the backup and shutting down the Oracle Key Vault servers for upgrade, that no databases perform a set or rekey operation (for example, using the `ADMINISTER KEY MANAGEMENT` statement), since these new keys will not be included in the backup.

Do not proceed without completing this step.

3. First, upgrade the standby server while the primary server is running.

Follow Steps 2 through Step 11 of the standalone server upgrade process for CNSA.
4. Ensure that the upgraded standby Oracle Key Vault server is restarted and running.
5. Upgrade the primary Oracle Key Vault server following Steps 1 through 11 of the standalone server upgrade.

After both the standby and primary Oracle Key Vault servers are upgraded, the two servers will automatically synchronize.

6. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
7. Select the **System** tab, and then **Status**.
8. Verify that the **Version** field displays the new software version release number.

Related Topics

- [Upgrading a Standalone Oracle Key Vault Server with CNSA](#)

You can upgrade a standalone Oracle Key Vault while using Commercial National Security Algorithm (CNSA) compliance by upgrading and then executing the `okv_cnsa` script.

Minimizing Downtime

Business-critical operations require data to be accessible and recoverable with minimum downtime.

You can configure Oracle Key Vault to ensure minimum downtime in the following ways:

- **Configuring a multi-master cluster:** You can configure a multi-master cluster by adding redundancy in the form of additional nodes. The client can access any available node. In the event of a failure of any node, a client will automatically connect to another node in the endpoint node scan list. This reduces and potentially eliminates downtime.
- **Configuring a primary-standby environment:** A primary-standby environment is configured by adding redundancy in the form of a standby server. The standby server takes over from the primary server in the event of a failure, thus eliminating single points of failure, and minimizing downtime.
- **Enabling read-only restricted mode:** Primary-standby read-only restricted mode ensures endpoint operational continuity when primary or standby Oracle Key Vault servers are affected by server, hardware, or network failures. When an unplanned shutdown causes the standby server to become unreachable, the primary server is still available to the endpoints.

If primary-standby read-only restricted mode is disabled, then the primary server will become unavailable and stop accepting requests in the event of a standby failure. Endpoints connected to Oracle Key Vault are unable to retrieve keys until connectivity is restored between primary and standby servers.

To ensure endpoint operational continuity in the event of a primary or standby server failure, enable read-only restricted mode.

- **Enabling persistent master encryption key cache:** The persistent master encryption key cache ensures that the endpoints can access keys in the event of a primary or standby server failure. While the surviving server is taking over from the failed peer, the endpoints can retrieve keys from the persistent cache and continue operations normally.
- **Apply the TDE heartbeat database patch on endpoints:** Apply the database patch for Bug 22734547 to tune the Oracle Key Vault heartbeat.

Oracle strongly recommends that you back up Oracle Key Vault data regularly on a schedule. This practice ensures that backups are current and hold the most recent data. You can use this backup to restore a new or existing Oracle Key Vault server and enable it to be fully operational with minimum downtime and data loss.

If the Oracle Key Vault installation uses an online master encryption key (formerly known as TDE direct connect), then during an upgrade, ensure that you upgrade database endpoints in parallel to reduce total downtime.

Related Topics

- [Using the Persistent Master Encryption Key Cache](#)
The persistent master encryption key cache feature enables databases to be operational when the Oracle Key Vault server is unavailable.

Managing Service Certificates

This chapter explains about Oracle Key Vault-generated certificates. You will also learn how to manage self-signed and third-party certificates.

- [Overview of Oracle Key Vault Certificates](#)
Oracle Key Vault uses certificates for various purposes including endpoint authentication, server authentication, and securing the communication channels using the TLS protocol.
- [Certificates Validity Period](#)
You can set the validity periods for Oracle Key Vault certificates to meet the security, compliance, and operational requirements.
- [Monitoring Certificates Expiry](#)
Proactively set alerts and monitor the expiry dates of the Oracle Key Vault certificates and rotate them before they expire.
- [Managing CA Certificate Rotation](#)
You can use the Oracle Key Vault management console to rotate the CA certificate before the certificate gets expired. The new CA certificate can be a self-signed Root CA certificate or an intermediate CA certificate.
- [Managing Server Certificates and Node Certificates Rotation](#)
Use the Oracle Key Vault management console to rotate server or node certificates.
- [Managing the Oracle Key Vault CA Certificate After Expiry](#)
You cannot start a CA certificate rotation when the Oracle Key Vault CA certificate has already expired.
- [Configuring Oracle Key Vault with an Alternate Hostname](#)
Learn how to configure Oracle Key Vault with an alternate hostname.
- [Managing Platform Certificates](#)
Rotate platform certificates before they expire using the Oracle Key Vault management console.

Overview of Oracle Key Vault Certificates

Oracle Key Vault uses certificates for various purposes including endpoint authentication, server authentication, and securing the communication channels using the TLS protocol.

The TLS protocol protects communication between the Oracle Key Vault server or node, and the endpoints. The TLS protocol also protects the back channel communication between the Oracle Key Vault nodes in the cluster deployment or Oracle Key Vault servers in the primary-standby deployment. The TLS certificates used by endpoints and the Oracle Key Vault servers or cluster-nodes are issued by the Oracle Key Vault itself using its CA certificate. The Oracle Key Vault's CA certificate may be a self-signed Root CA or an intermediate CA.

Oracle Key Vault generates the TLS certificates with the exception of the intermediate CA certificate.

CA Certificate

The CA certificate is a self-signed Root CA or an intermediate CA certificate that the Oracle Key Vault uses to issue endpoint certificates, as well as server or node certificates. The self-

signed Root CA certificate is generated at the time of Oracle Key Vault installation. Customers can choose to replace it with an intermediate CA certificate that is signed by the organization's own internal CA or a third-party CA post-installation or post-upgrade. The CA certificate is the same for all nodes in a multi-master cluster deployment and for both the primary and standby servers of a primary-standby deployment. The CA certificate is different from the console certificates.

If you do not rotate the CA certificate before it expires, none of the endpoints can communicate with the Oracle Key Vault server or any node of the Oracle Key Vault cluster and all the endpoints will face a downtime. In the cluster deployment none of the Oracle Key Vault nodes will be able to communicate with each other and in case of primary-standby, the communication between primary and standby servers will breakdown.

 **Note:**

The CA Certificate must be rotated before it expires to prevent outage to endpoints. Start the CA certificate rotation several weeks in advance of CA certificate expiry to prevent outage to the Oracle Key Vault deployment and endpoints.

Rotating CA certificates also rotates the server or node certificates and endpoint certificates.

Server and Node Certificate

Server or Node Certificate is the TLS certificate of the Oracle Key Vault server or a cluster node. While in a standalone or primary-standby deployment, Oracle Key Vault uses server certificates to communicate with its endpoints. In a multi-master deployment of Oracle Key Vault, each cluster node has its own node certificate. Oracle Key Vault cluster nodes use node certificates to communicate with each other and with the endpoints.

These certificates are referred to as server certificates for standalone and primary-standby systems and as node certificates in multi-master cluster configurations. The Oracle Key Vault CA certificate is used to issue these certificates.

Rotate the server or node certificate before they expire as described in section [Managing Server Certificates and Node Certificates Rotation](#). The CA and endpoint certificates are not rotated when server or node certificates are rotated.

 **Note:**

Rotating the node certificate in a multi-master cluster deployment is a per-node operation.

If the server certificate is not rotated in a standalone deployment before it expires, none of the endpoints can communicate to the Oracle Key Vault server and all the endpoints will face a downtime. If you do not rotate the server certificate in a primary-standby deployment before it expires, then none of the endpoints can communicate to the primary server and all the endpoints face a downtime.

If you do not rotate the node certificate in cluster deployment before it expires, the endpoints use the other nodes for the endpoint operations like fetching a key. However, the inter-node communication will be impacted and operations like creating a new endpoint or creating a new wallet will be impacted.

**Note:**

If all of the node certificates in the cluster deployment have expired, endpoints cannot communicate with any node in the multi-master cluster.

Endpoint Certificate

Each endpoint is issued a unique endpoint TLS certificate that is used to authenticate the endpoint with the Oracle Key Vault. The Oracle Key Vault Certificate Authority (CA) certificate is used to issue the endpoint certificates. Rotate an endpoint's certificate before it expires as described in section [Rotating Endpoint Certificates](#) . The CA or server or node certificates are not rotated when endpoint certificates are rotated.

If an endpoint is not rotated before its certificate expires, the endpoint experiences downtime and is required to be re-enrolled.

Platform Certificates

Oracle Key Vault platform certificates are used when adding a new node to an Oracle Key Vault multi-master cluster. Platform certificates are also used when shipping redo between read/write nodes in the cluster. The platform certificates are different from the Oracle Key Vault service certificates.

Related Topics

- [Rotating CA Certificate](#)
Use the Oracle Key Vault management console to rotate CA certificate and enable either a self-signed root CA certificate or an intermediate CA certificate.
- [Deleting, Suspending, Reenrolling, or Rotating Endpoints](#)
When endpoints no longer use Oracle Key Vault to store security objects, you can delete them. You can also suspend, and later resume them when they are needed. You can also re-enroll or rotate endpoints when necessary.
- [Finding the Expiration Date of the CA Certificate](#)
You can find how much time the Oracle Key Vault CA certificate has before it expires by navigating to the Service Certificates page.
- [Finding the Expiration Date of Server Certificates and Node Certificates](#)
You can find the expiration date of server certificates and node certificates in the Oracle Key Vault management console.
- [Finding the Expiration Date of Endpoint Certificates](#)
You can find the expiration date of endpoint certificates in the Oracle Key Vault management console.

Certificates Validity Period

You can set the validity periods for Oracle Key Vault certificates to meet the security, compliance, and operational requirements.

- [About Certificates Validity Period](#)
Compliance and best security practices have different requirements for certificate validity depending upon the purpose and use of the certificate.
- [Setting Validity Period of Self-Signed Root CA Certificate](#)
You can configure the validity period for the self-signed Root certificate authority (CA) certificate from the Oracle Key Vault management console.

- [Configuring Certificate Validity Period for Server and Node Certificates](#)
You can configure the validity period for server or node certificates in the Oracle Key Vault management console.
- [About Configuring Certificate Validity Period for Endpoint Certificates](#)
You can set the validity period for the endpoint certificates in the Global Endpoint Configuration parameters.

About Certificates Validity Period

Compliance and best security practices have different requirements for certificate validity depending upon the purpose and use of the certificate.

For simplicity, up until Oracle Key Vault release version 21.3 all the three certificates, the Oracle Key Vault CA, server or node, and endpoint certificate, including the self-signed Root CA certificate, are rotated together. However, the server or node, endpoint, and CA certificates can have different validity periods. Generally, the validity period requirements for the endpoint and server or node certificates are different than that of the CA certificate. You can configure the validity periods of the self-signed Root CA, the server or node certificates and the endpoint certificates independently with different values. You can rotate the server or node certificates independent of the CA certificate rotation. Starting in Oracle Key Vault release 21.5, you can rotate the endpoint certificate independently of the CA and server or node certificates.

The default and the range of the validity periods of the TLS certificates in Oracle Key Vault are described in the table.

Table 19-1 Certificates Validity Period

Certificate	Default Validity (out of the box)	Minimum Validity	Maximum Validity
Self-Signed Root CA	1095 days or 3 years	365 days or 1 year	3650 days or 10 years
Intermediate CA	Defined by signing CA	Defined by signing CA	Defined by signing CA
Server/Node Certificate	365 days or 1 year	365 days or 1 year	1095 days or 3 years
Endpoint Certificate	365 days or 1 year	365 days or 1 year	1095 days or 3 years

The certificate validity period automatically determines the certificate expiry. Rotate the certificates before they expire.

You can set different validity periods for each type of certificate to meet your requirements.

Setting the validity period of certificates does not affect the validity period of existing certificates. The configured validity periods take effect when a new certificate is generated either during the certificate rotation or when you set up a new endpoint or cluster node.



Note:

If you are logged in as a user with the monitoring privilege, you cannot view the **Manage CA Certificate** and **Manage Server/Node Certificate** options.

The CA signing authority sets the validity period of the intermediate CA certificate.

 **Note:**

For simplicity, until Oracle Key Vault release 21.4, all three types of certificates - self-signed Root CA certificate, server or node certificate, and the endpoint certificate had the same certificate validity period. The server or node certificates could not be rotated independently of the CA certificate rotation, nor could they be configured with different certificate validity periods. Until Oracle Key Vault release 21.5, the endpoint certificates could not be rotated independently of the CA certificate rotation.

Related Topics

- [Managing CA Certificate Rotation](#)
You can use the Oracle Key Vault management console to rotate the CA certificate before the certificate gets expired. The new CA certificate can be a self-signed Root CA certificate or an intermediate CA certificate.
- [Rotating CA Certificate](#)
Use the Oracle Key Vault management console to rotate CA certificate and enable either a self-signed root CA certificate or an intermediate CA certificate.
- [Finding the Expiration Date of Server Certificates and Node Certificates](#)
You can find the expiration date of server certificates and node certificates in the Oracle Key Vault management console.
- [Finding the Expiration Date of Endpoint Certificates](#)
You can find the expiration date of endpoint certificates in the Oracle Key Vault management console.
- [Finding the Expiration Date of the CA Certificate](#)
You can find how much time the Oracle Key Vault CA certificate has before it expires by navigating to the Service Certificates page.

Setting Validity Period of Self-Signed Root CA Certificate

You can configure the validity period for the self-signed Root certificate authority (CA) certificate from the Oracle Key Vault management console.

The CA certificate validity period governs the end date of the CA certificate. The end date of the CA certificate acts as an upper bound on the validity period of the server or node, and the endpoint certificates, when they are issued.

Setting the validity of self-signed Root CA certificate does not enable it, that is, switch it into use. You have to rotate the CA certificate to generate and enable a new self-signed Root CA with the set validity period as described in [Rotating CA Certificate](#).

To set the validity period for the self-signed Root CA:

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

In a primary-standby environment, log in to the primary Oracle Key Vault server. In a multi-master cluster environment, log in to the node selected for CA certificate rotation in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.

5. In the **CA Certificate Details** page, select the **Self-Signed Root CA** option. The Self-Signed Root CA option is selected by default.
6. Set the validity value in the **Self-Signed Root CA Certificate Validity (in days)** field. The default is 1095 days (3 years). You can set a maximum validity period of 3650 days (10 years) and a minimum validity period of 365 days (1 year).
7. Click **Save**.



The screenshot shows a dialog box titled "CA Certificate Details". At the top right, there are "Cancel" and "Rotate CA Certificate" buttons. Below the title, there are two radio buttons: "Self-Signed Root CA" (which is selected) and "Intermediate CA". Underneath, there is a section titled "Self-Signed Root CA Settings" with a "Save" button on the right. Inside this section, there is a field labeled "Self-Signed Root CA Certificate Validity (in days)" with a value of "1095" and a spinner icon.

Configuring Certificate Validity Period for Server and Node Certificates

You can configure the validity period for server or node certificates in the Oracle Key Vault management console.

The certificate validity period takes effect the next time you rotate the server or node certificates. It will also be taken into account when you generate the server or node certificates as part of a CA certificate rotation, or when you add a new node to the cluster, to the node certificates for that new node. Irrespective of the value that the server or node certificate validity is set to, when the certificates are eventually generated, Oracle Key Vault ensures that their expiry date is less than that of the CA certificate.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

In a multi-master cluster environment, you can log in to any node in the cluster.

2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, select **Service Certificates**.
4. Depending on your environment, perform the following:
 - In a standalone or primary-standby environment: In the Current Server Certificate area, select **Manage Server Certificate**.
 - In a multi-master cluster environment: In the Current Node Certificate area, select **Manage Node Certificate**.
5. In the **Server Certificate Validity (in days)** or **Node Certificate Validity (in days)** field, enter a value between 365 days (the minimum and the default) and 1095 days for this setting.
6. Click **Save**.

About Configuring Certificate Validity Period for Endpoint Certificates

You can set the validity period for the endpoint certificates in the Global Endpoint Configuration parameters.

The default value is 365 days (1 year). You can set a maximum validity period of 1095 days (3 years) and a minimum validity period of 365 days (1 year).

The certificate validity period takes effect the next time that an existing endpoint is rotated or re-enrolled, or when a new endpoint is added. Irrespective of the value that the endpoint certificate validity is set to, when the endpoint is eventually rotated, Oracle Key Vault ensures that the endpoint certificate expiry date is less than that of the CA certificate.

Related Topics

- [Setting Global Endpoint Configuration Parameters](#)
You can set global endpoint configuration parameters in the Oracle Key Vault management console.
- [Reenrolling an Endpoint](#)
When you reenroll an endpoint, the enrollment process automatically upgrades the endpoint software and also generates new endpoint certificates.
- [Rotating Endpoint Certificates](#)
Rotating an endpoint's certificate extends its certificate validity without incurring downtime for the endpoint.

Monitoring Certificates Expiry

Proactively set alerts and monitor the expiry dates of the Oracle Key Vault certificates and rotate them before they expire.

- [Monitoring Certificates Expiry Using Certificate Expiration Alerts](#)
Set expiration alerts as reminders to rotate the certificates before their expiration date.
- [Finding the Expiration Date of Endpoint Certificates](#)
You can find the expiration date of endpoint certificates in the Oracle Key Vault management console.
- [CA Certificate Expiration Date on Status Page](#)
You can check the CA Certificate Expiration Date, which is the expiration date of the Oracle Key Vault CA certificate, from the Status page.
- [Server and Node Certificate Expiration on Status Page](#)
You can check the Server Certificate Expiration Date (in a standalone or primary-standby environment) or the Node Certificate Expiration Date (in a multi-master cluster environment) from the Status page.
- [Finding the Expiration Date of the CA Certificate](#)
You can find how much time the Oracle Key Vault CA certificate has before it expires by navigating to the Service Certificates page.
- [Finding the Expiration Date of Server Certificates and Node Certificates](#)
You can find the expiration date of server certificates and node certificates in the Oracle Key Vault management console.

Monitoring Certificates Expiry Using Certificate Expiration Alerts

Set expiration alerts as reminders to rotate the certificates before their expiration date.

Expiration of a certificate, especially the CA certificate, breaks the endpoint and Oracle Key Vault communication, and impacts the operations of one or more endpoints to the extent of stopping of endpoint operations completely. In addition, upgrades and communication between the Oracle Key Vault multi-master cluster nodes may also fail. Ensure that you rotate certificates much before their expiration date.

To avoid this scenario, Oracle recommends that you configure alerts as a reminder to rotate the certificates before they expire. There are separate alerts for endpoint certificate expiration, server or node certificate expiration, and CA certificate expiration.

If using an intermediate CA certificate, monitor the certificate expiry of the CA certificate trust chain independently. The intermediate CA certificate must be rotated before any of the certificates in its certificate trust chain expires. This prevents an outage to endpoints. Start the CA certificate rotation several weeks in advance of CA certificate expiry to prevent outage to the Oracle Key Vault deployment and endpoints .

 **Note:**

- If you are using an intermediate CA certificate, you must monitor the certificate expiry in the CA trust chain independently. The expiration of any certificate in the CA trust chain causes an outage to Oracle Key Vault.
- You must promptly address the certification expiration alerts by rotating the certificates indicated by the alert. Depending upon your deployment, the rotation of the Oracle Key Vault CA certificate, in particular, may take a very long time (in the order of several days). Begin the CA certificate rotation process well before CA certificate expiry to avoid outages.

Related Topics

- [Configuring Oracle Key Vault Alerts](#)
You can select the type of alerts that you want to see in the Oracle Key Vault dashboard.
- [Managing Server Certificates and Node Certificates Rotation](#)
Use the Oracle Key Vault management console to rotate server or node certificates.
- [Steps for Managing CA Certificate Rotation](#)
A user with the System Administrator role can perform CA certificate rotation when the CA is expiring. The user can set up a new self-signed Root CA or an intermediate certificate and put the new certificate into use. The server or node certificates, and the endpoint certificates are also rotated as part of this process.

Finding the Expiration Date of Endpoint Certificates

You can find the expiration date of endpoint certificates in the Oracle Key Vault management console.

To find the expiration date of the endpoint certificates, navigate to the Endpoints page and check the Endpoint Certification Expiration field.

1. Log in to the Oracle Key Vault management console.
In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **Endpoints** tab.
3. In the **Endpoints** table, check **Endpoint Certification Expiration**.

Related Topics

- [Certificates Validity Period](#)
You can set the validity periods for Oracle Key Vault certificates to meet the security, compliance, and operational requirements.

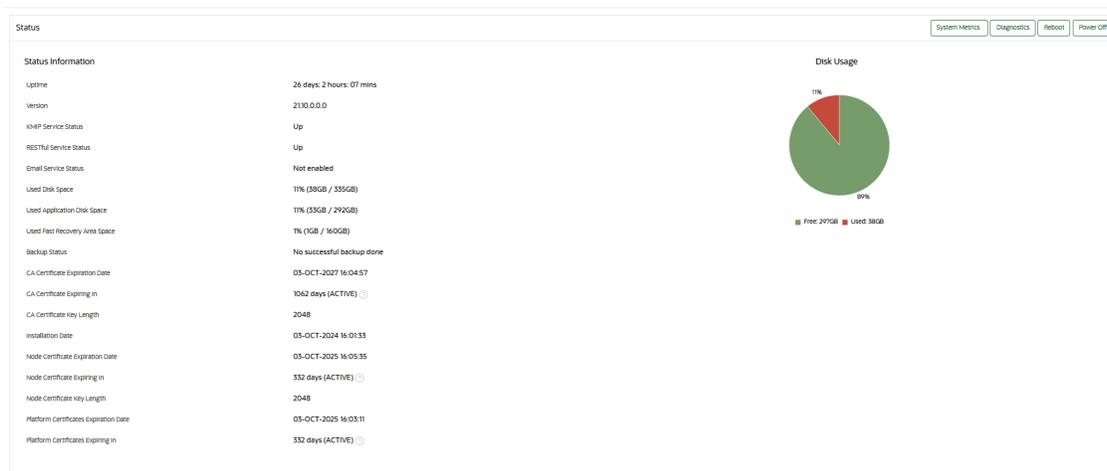
CA Certificate Expiration Date on Status Page

You can check the CA Certificate Expiration Date, which is the expiration date of the Oracle Key Vault CA certificate, from the Status page.

The **CA Certificate Expiration Date** field in the **System Status** page reflects the expiration date of the CA certificate. The **CA Certificate Expiring In** on the **System Status** page shows how many days are left to expire for the CA certificate.

To navigate to the **System Status** page, select the **System** tab.

1. Log in to the Oracle Key Vault management console as a System Administrator. In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **System** tab and then **Status** from the left navigation side bar.
3. Check the **CA Certificate Expiration Date** field.
4. Check the **CA Certificate Expiring in** field.



Oracle Key Vault raises an alert for the CA certificate expiration when the **CA Certificate Expiration Date** falls within the alert threshold period. You can also monitor the **CA Certificate Expiration Date** over SNMP.

Related Topics

- [Configuring Oracle Key Vault Alerts](#)
You can select the type of alerts that you want to see in the Oracle Key Vault dashboard.

Server and Node Certificate Expiration on Status Page

You can check the Server Certificate Expiration Date (in a standalone or primary-standby environment) or the Node Certificate Expiration Date (in a multi-master cluster environment) from the Status page.

The **Server Certificate Expiration Date** field on the **System Status** reflects the expiration date of the server certificate. The **Server Certificate Expiring In** on the **Status** page shows how many days are left for the server certificate to expire.

In a multi-master cluster environment, these fields are called **Node Certificate Expiration Date** and **Node Certificate Expiring In**. The **Node Certificate Expiration Date** field reflects

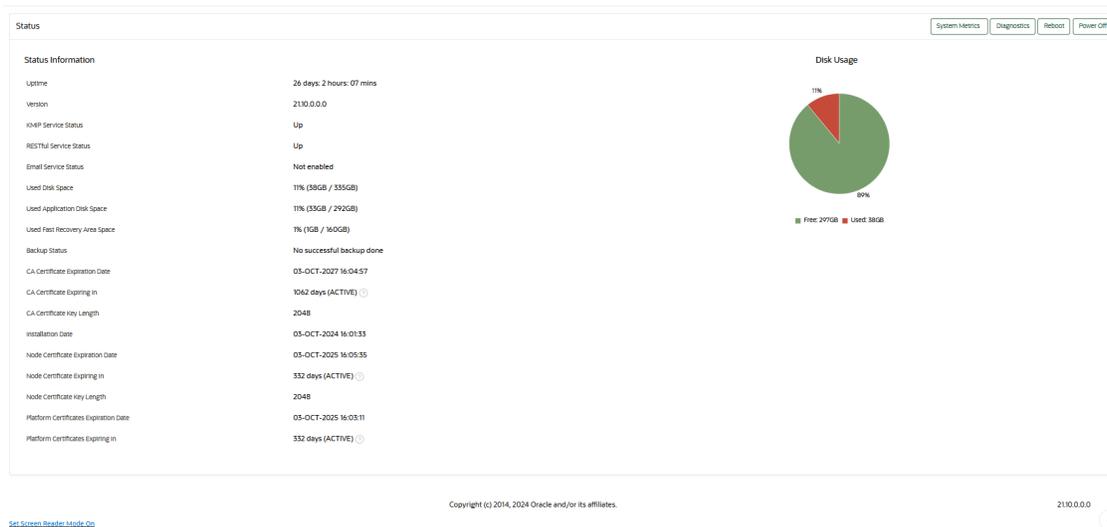
the expiration date of the node certificate, while **Node Certificate Expiring In** shows how many days are left for the node certificate to expire.

Note:

In a multi-master cluster environment, log into the node whose node certificate expiration date you wish to check. Different nodes can have different node certificate expiration dates.

To navigate to the **Status** page, select the **System** tab.

1. Log in to the Oracle Key Vault management console as a System Administrator.
2. Select the **System** tab and then **Status** from the left navigation side bar.
3. Check the **Server Certificate Expiration Date** field. In a multi-master cluster environment, this field is **Node Certificate Expiration Date**.
4. Check the **Server Certificate Expiring in** field. In a multi-master cluster environment, this field is **Node Certificate Expiring In**.



Oracle Key Vault raises an alert for the Server or Node certificate expiration, when the server or node certificate expiration date falls within the configured alert threshold period. You can also monitor the **Server Certificate Expiration Date** over SNMP.

Finding the Expiration Date of the CA Certificate

You can find how much time the Oracle Key Vault CA certificate has before it expires by navigating to the Service Certificates page.

1. Log in to the Oracle Key Vault management console as the System Administrator.
In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, select **Service Certificates**.
4. In the **Current CA Certificate** area, check the **End Date** setting to know when the CA certificate is expiring.

The **Expiring In** setting shows the number of days left for the CA certificate to expire.

 **Note:**

You can also check the CA certificate expiration date on the Oracle Key Vault System Status page.

Service Certificates

Current CA Certificate Manage CA Certificate

Common Name	CA
Start Date	01-FEB-2022 13:56:47
End Date	31-JAN-2025 13:56:47
Expiring In	1092 days
Certificate Issuer	CA

Current Node Certificate Manage Node Certificate

Start Date	01-FEB-2022 13:57:06
End Date	01-FEB-2023 13:57:06
Expiring In	362 days
Certificate Issuer	CA

Related Topics

- [Configuring Oracle Key Vault Alerts](#)
You can select the type of alerts that you want to see in the Oracle Key Vault dashboard.

Finding the Expiration Date of Server Certificates and Node Certificates

You can find the expiration date of server certificates and node certificates in the Oracle Key Vault management console.

Perform the following steps to review the end dates and time to expire of all the node certificates in the cluster:

1. Log in to Oracle Key Vault management console as a user who has the System Administrator role.
In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the Certificates area, select **Service Certificates**.
4. In a standalone or primary-standby environment:
 - Under **Current Server Certificate**, check the **End Date** setting to determine when the server certificate is expiring. The **Expiring In** setting also shows the number of days left for the server certificate expiry.
5. In a multi-master cluster environment:
 - In the Current Node Certificate area, select **Manage Node Certificate**. Under **Current Node Certificate**, check the **End Date** setting. The **Expiring In** setting also shows the number of days left for the node certificate expiry.
 - You can view the end dates and time to expire of all the node certificates in the cluster in the **Cluster Node Certificate Details** area.

Current Node Certificate

Start Date	21-MAR-2022 11:52:35
End Date	21-MAR-2023 11:52:35
Expiring In	364 days
Certificate Issuer	CA

Cluster Node Certificates Details

Node ID	Node Name	IP Address	Certificate Start Date	Certificate End Date	Certificate Status	Expiring In (days)
1	Node1_Austin	192.0.2.1	21-MAR-2022 14:09:09	03-AUG-2023 14:09:09	ACTIVE	499
2	Node2_Boston	192.0.2.2	21-MAR-2022 11:52:35	21-MAR-2023 11:52:35	ACTIVE	364

1 - 2

If a server or node certificate is expiring soon, then Oracle recommends that you rotate the certificate at the earliest.



Note:

You can also check the Server Certificates and Node Certificates expiration date on the Oracle Key Vault System Status page.

Related Topics

- [Certificates Validity Period](#)
You can set the validity periods for Oracle Key Vault certificates to meet the security, compliance, and operational requirements.
- [Managing Server Certificates and Node Certificates Rotation](#)
Use the Oracle Key Vault management console to rotate server or node certificates.

Managing CA Certificate Rotation

You can use the Oracle Key Vault management console to rotate the CA certificate before the certificate gets expired. The new CA certificate can be a self-signed Root CA certificate or an intermediate CA certificate.

- [Steps for Managing CA Certificate Rotation](#)
A user with the System Administrator role can perform CA certificate rotation when the CA is expiring. The user can set up a new self-signed Root CA or an intermediate certificate and put the new certificate into use. The server or node certificates, and the endpoint certificates are also rotated as part of this process.
- [Checking for Self-Signed Root CA or Intermediate CA Certificate](#)
Oracle Key Vault uses either a self-signed root CA certificate or an intermediate CA certificate.
- [Setting the Key Length of the CA Certificate](#)
You can select between the 2048-bits or 4096-bits key length values for the certificate authority (CA) certificate key length.
- [Setting the Validity of Self-Signed Root CA Certificate](#)
You can set the number of days for the validity of a self-signed Root certificate authority (CA) certificate.
- [Setting Up the Intermediate CA Certificate](#)
Use the Oracle Key Vault management console to generate the certificate signing request for the intermediate CA certificate, and upload the intermediate CA certificate signed by a trusted third party.
- [Rotating CA Certificate](#)
Use the Oracle Key Vault management console to rotate CA certificate and enable either a self-signed root CA certificate or an intermediate CA certificate.
- [Setting the Endpoint Certificate Rotation Batch Size](#)
The endpoint certificate rotation batch size value represents the number of endpoints that can be in the `ROTATED` state on a given Oracle Key Vault server or node during the CA certification rotation process.
- [Setting the Endpoint Certificate Rotation Sequence](#)
In a multi-master cluster environment, when you rotate certificate authority (CA) certificate, you broadly set the order in which endpoints can be rotated by ordering cluster subgroups.
- [Checking Overall Certificate Rotation Status](#)
Use the Oracle Key Vault management console to check the overall status of a certificate rotation.
- [Checking Certificate Rotation Status for Endpoints](#)
Use the Oracle Key Vault management console to check the status of a certificate rotation for endpoints.
- [Post-CA Certificate Rotation Tasks](#)
After you complete the CA certificate rotation, perform the post-rotation tasks.
- [Factors Affecting CA Certificate Rotation Process](#)
Consider these factors that affect the certificate authority (CA) certificate rotation process in cluster environments.
- [Guidelines for Managing CA Certificate Rotations](#)
Consider these Oracle Key Vault guidelines for managing certificate authority (CA) certificate.

Steps for Managing CA Certificate Rotation

A user with the System Administrator role can perform CA certificate rotation when the CA is expiring. The user can set up a new self-signed Root CA or an intermediate certificate and put the new certificate into use. The server or node certificates, and the endpoint certificates are also rotated as part of this process.

A user with System Administrator role can perform CA certificate rotation. The CA certificate rotation process involves the following steps:

1. Set the validity of the self-signed Root CA certificate or set up an intermediate CA certificate.
2. Choose the endpoint certificate rotation controls.
3. Start CA certificate rotation.
 - In the case of a self-signed CA certificate rotation, Oracle Key Vault generates and puts into use a new self-signed CA certificate.
 - In the case of an intermediate CA certificate rotation, the intermediate CA certificate uploaded in an earlier step is put into use.
4. Monitor the progress of automatic endpoint updates, as each endpoint is issued with new certificates by the new CA certificate.
5. After all endpoints have been successfully rotated, Oracle Key Vault issues server or node certificate with the new CA certificate to complete the CA certificate rotation.
6. Perform post-CA certificate rotation tasks.

 **Note:**

Starting with Oracle Key Vault release 21.5, CA certificate rotation is now a single-step process, that is, **Start CA Certificate Rotation**. Previously, in Oracle Key Vault release 21.4 and earlier, CA certificate rotation was a two-step process, that is, **Start CA Certificate Rotation**, and **Activate CA Certificate**.

CA certificate rotation process is the same for standalone, primary-standby, and cluster environments. In the multi-master environment, Oracle recommends that you select one of the cluster nodes to drive the CA certificate rotation. Oracle Key Vault automatically synchronizes the certificates in both systems in a primary-standby configuration, and in all nodes in a multi-master cluster configuration. You do not have to perform any extra configuration.

Checking for Self-Signed Root CA or Intermediate CA Certificate

Oracle Key Vault uses either a self-signed root CA certificate or an intermediate CA certificate.

To check if the current Oracle Key Vault CA certificate is a self-signed root CA or an intermediate CA, check the **Common Name** and **Certificate Issuer** fields in the **Service Certificates** page. If they are similar, for example, both are `CA`, or start with `OKV_CA_`, then the current CA certificate is a self-signed root CA. Otherwise, the current CA certificate is an intermediate CA. Additionally, in the intermediate CA certificate, the **Certificate Issuer** field displays the common name of the trusted third party.

Check the Common Name and Certificate Issuer of the current CA certificate in the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, select **Service Certificates**.
4. In the **Current CA Certificate** area, check and compare the **Common Name and Certificate Issuer** fields. Help text is available for the **Common Name** field that indicates whether the CA certificate is a self-signed CA or an intermediate CA.

Related Topics

- [Finding the Expiration Date of the CA Certificate](#)
You can find how much time the Oracle Key Vault CA certificate has before it expires by navigating to the Service Certificates page.

Setting the Key Length of the CA Certificate

You can select between the 2048-bits or 4096-bits key length values for the certificate authority (CA) certificate key length.

The CA certificate key length gets applied to the server certificates, node certificates, and endpoint certificates.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
In a primary-standby environment, log in to the primary Oracle Key Vault server. In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.
5. In the **CA Certificate Details** page, select the **Self-Signed Root CA** option or **Intermediate CA** option.
6. Choose the **key length** from the drop down menu. The default key length value is **2048**.
7. Click **Save**.

Setting the Validity of Self-Signed Root CA Certificate

You can set the number of days for the validity of a self-signed Root certificate authority (CA) certificate.

The CA certificate validity period acts as an upper limit on the validity period of the server certificates, node certificates and endpoint certificates.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
In a primary-standby environment, log in to the primary Oracle Key Vault server. In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.

5. In the **CA Certificate Details** page, select the **Self-Signed Root CA** option (this should be selected by default).
6. Set the validity value in the **Self-Signed Root CA Certificate Validity (in days)** field. The default is 1095 days (3 years). You can set a maximum of 3650 days (10 years).
7. Click **Save**.
Go to section [Rotating CA Certificate](#) to generate and enable the self-signed root CA certificate.

Related Topics

- [Setting Validity Period of Self-Signed Root CA Certificate](#)
You can configure the validity period for the self-signed Root certificate authority (CA) certificate from the Oracle Key Vault management console.

Setting Up the Intermediate CA Certificate

Use the Oracle Key Vault management console to generate the certificate signing request for the intermediate CA certificate, and upload the intermediate CA certificate signed by a trusted third party.

Uploading the intermediate CA certificate does not enable it (that is, uploading the intermediate CA certificate does not put it into use). Perform the following steps to rotate the CA certificate and enable the uploaded intermediate CA certificate:

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
In a primary-standby environment, log in to the primary Oracle Key Vault server. In a multi-master cluster environment, log in to the node selected for CA certificate rotation in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.
5. Under **CA Certificate Details**, select the **Intermediate CA** option.
6. Enter information about your organization in the following fields:
 - Organization Name
 - Country/Region
 - Organization Unit
 - City
 - State/Province

7. Select **Generate Certificate Request**.
8. In the dialog box that lets you know that the generation will take a few minutes, click **OK**.
9. After Oracle Key Vault generates the certificate request, in the **CA Certificate Details** area, select **Download Certificate Request** to download the certificate request file.

The certificate signing request file is named as follows:

```
OKV_Intermediate_CA_Certificate.csr
```

The **Intermediate CA Certificate Signing Request Details** area shows the details of certificate signing request.

At this stage, the current CA certificate is still enabled in Oracle Key Vault. The **Current Certificate** area displays the details of the currently active CA. If you want to cancel the setup of the intermediate CA certificate, then click **Abort**.

Common Name	Organization	Country/Region	Organizational Unit	Locality	State/Province
OKV_CA_20220321103725691543	KeyVault	US	Security	RedwoodCity	CA

10. Have a trusted third party issue the intermediate CA certificate using the downloaded certificate signing request.
11. Upload the intermediate CA certificate. In the **CA Certificate Details** area, select **Choose File** for **Intermediate CA Certificate** to find and select the intermediate CA certificate file, and then click **Upload**. In a multi-master cluster environment, you must upload the intermediate CA certificate on the same node where the certificate signing request was downloaded.

12. Upload the chain of trust for the intermediate CA certificate. In the **CA Certificate Details** area, select **Choose File** for **Certificate Chain of Trust** to find and select the chain of trust file, and then click **Upload**.

The chain of trust file is a PEM bundle that consists of the CA certificate used by the external signing authority to sign the intermediate certificate signing request, `OKV_Intermediate_CA_Certificate.csr` file, as well as all of the certificates in that CA's trust chain, in reverse order.

For example, suppose that `OKV_Intermediate_CA_certificate.csr` has been signed by the external signing authority, and that the generated certificate is called `OKV_Intermediate_CA_certificate.crt`. Also suppose that the external signing authority used its CA certificate, `CACertA`, to generate `OKV_Intermediate_CA_certificate.crt` from `OKV_Intermediate_CA_certificate.csr`. `CACertA` was, in turn, issued by `CACertB`. `CACertB` was issued by `CACertC`. The certificate trust chain file that you must upload must consist of `CACertA`, `CACertB`, `CACertC`, in that order, in the PEM bundle format. It should NOT contain `OKV_Intermediate_CA_certificate.crt`. For example, assuming that `CACertA`, `CACertB`, and `CACertC` are all certificates in PEM format, where each certificate file is of the form:

```
-----BEGIN CERTIFICATE-----
<cert contents>
-----END CERTIFICATE-----
```

The certificate chain of trust would look like this:

```
-----BEGIN CERTIFICATE-----
<CACertA contents>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<CACertB contents>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<CACertC contents>
-----END CERTIFICATE-----
```

In a multi-master cluster environment, you must upload the certificate chain of trust on the same node where you uploaded the intermediate CA certificate.

As part of the upload, Oracle Key Vault performs the following validations:

- a. The uploaded intermediate CA is verified using the uploaded certificate chain of trust.
- b. The certificate chain of trust has a depth of less than or equal to 8.

After the uploads are successful, the **Rotate CA Certificate** button is displayed.

Go to section [Rotating CA Certificate](#) to enable the uploaded intermediate CA certificate.

 **Note:**

- If you choose to set up an intermediate CA certificate, it is recommended that the certificate signature algorithm is a valid SHA-2 algorithm, such as sha256.
- If you choose to set up an intermediate CA certificate, ensure that the certificate can be used as a CA for both TLS clients and servers. This can be verified by checking the certificate's properties.

Related Topics

- [Rotating CA Certificate](#)
Use the Oracle Key Vault management console to rotate CA certificate and enable either a self-signed root CA certificate or an intermediate CA certificate.
- [Configuring Oracle Key Vault Alerts](#)
You can select the type of alerts that you want to see in the Oracle Key Vault dashboard.

Rotating CA Certificate

Use the Oracle Key Vault management console to rotate CA certificate and enable either a self-signed root CA certificate or an intermediate CA certificate.

Back up Oracle Key Vault before you start the certificate rotation process.

CA certificate rotation issues new certificates for the Oracle Key Vault servers, nodes, and endpoints.

 **Note:**

In Oracle Key Vault release 21.9, after a certificate rotation, password-protected endpoints may fail to connect to the Oracle Key Vault deployment, with `Server Connect Failed` errors. A fix for this issue is available as a patch for bug **37006283**. Apply this patch to the Oracle Key Vault release 21.9 deployment before initiating a CA certificate rotation.

Perform these steps to complete the CA certificate rotation process throughout the Oracle Key Vault environment.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

In a primary-standby environment, log in to the primary Oracle Key Vault server.

In a multi-master cluster environment, log in to the node selected for CA certificate rotation in the cluster. If you want to enable an intermediate CA certificate, then ensure that you initiate the CA certificate rotation from the same node where the intermediate certificate was uploaded.

2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.
5. If you want to enable a self-signed root CA certificate, then in **CA Certificate Details Area**, select the **Self-Signed Root CA** option.

If necessary, set the Self-Signed Root CA Certificate Validity value as described in [Setting the Validity of Self-Signed Root CA Certificate](#)

6. If you want to enable an intermediate CA certificate, then upload the intermediate CA certificate and the certificate chain of trust successfully. The Rotate CA Certificate button will now be visible.

If you do not see the Rotate CA Certification button, then set up the intermediate CA certificate as described in section [Setting Up the Intermediate CA Certificate](#).

7. Click **Rotate CA certificate**.
8. In the **Manage CA Certificate** page, you may set endpoint certificate rotation controls - batch size and in multi-master cluster deployments, sequence.

In a multi-master cluster environment, if necessary, choose the sequence in which the endpoint certificates should be rotated as described in section [Setting the Endpoint Certificate Rotation Sequence](#)

Manage CA Certificate
Cancel

You can rotate the Oracle Key Vault certificates (CA, server and endpoint certificates) by initiating CA certificate rotation and then activating the new CA certificate. Click on the 'Start CA Certificate Rotation' button to initiate the CA certificate rotation process.

Additionally, you can also set endpoint certificate rotation controls like choosing the order in which the endpoints will be selected for certificate rotation.

Current CA Certificate
Start CA Certificate Rotation

Common Name	CA
Start Date	17-MAR-2022 20:08:59
End Date	16-MAR-2025 20:08:59
Expiring In	1090 days
Certificate Issuer	CA

Endpoint Certificate Rotation Controls

Endpoint Certificate Rotation Batch Size

Save

Endpoint Certificate Rotation Sequence

Endpoints enrolled with the node in cluster subgroup that is higher in the cluster subgroup order will be processed first.

Cluster Subgroup Order
Delete
Select Cluster Subgroup

BostonSubgroup
AustinSubgroup

Current Server Certificate

Start Date	21-MAR-2022 14:09:09
End Date	03-AUG-2023 14:09:09
Expiring In	498 days
Certificate Issuer	CA

9. In the **Manage CA Certificate** page, in the **Current CA Certificate** area, select **Start CA Certificate Rotation**.

10. In the confirmation dialog box, click **OK**.

If you enable a self-signed root CA certificate, a new self-signed root CA certificate is created. In a multi-master cluster environment, Oracle Key Vault distributes and installs the newly created self-signed root CA certificate or uploaded intermediate CA certificate to all nodes of the cluster. In a primary-standby environment Oracle Key Vault distributes and installs these certificates to the standby. In case of a standalone environment, Oracle Key Vault simply installs the certificate that you enable.

At this stage, the endpoints continue to use the certificates issued using the previous CA certificate. The **Old CA Certificate** area displays the details of the currently active CA. The **New CA Certificate** area displays the certificate you have rotated along with its common name. Additionally, once a CA certificate rotation has begun, a banner is displayed, indicating that a CA certificate rotation is in progress and the number of days left to expiration of the old CA. It continues to be displayed until CA certificate rotation is complete. The banner is periodically updated with the number of days left to expiration of the old CA, and continues to be displayed until CA certificate rotation is complete.

If you want to cancel the rotation process, click **Abort CA Certificate Rotation**.

The **Abort CA Certificate Rotation** operation is available for initial few minutes. After this, when the **Abort CA Certificate Rotation** button is no longer visible, Oracle Key Vault automatically begins the process of enabling the new Oracle Key Vault CA certificate.

 **Note:**

In Oracle Key Vault 21.4 and earlier, CA certificate rotation involved two steps: **Start CA Certificate Rotation**, followed by **Activate CA Certificate**. Starting with Oracle Key Vault 21.5, it is now a single-step process initiated by **Start CA Certificate Rotation**. Please exercise caution and initiate CA certificate rotation only when fully ready to do so.

Manage CA Certificate
Cancel **Abort CA Certificate Rotation**

CA certificate rotation has been initiated in the cluster.

The CA certificate bundle that was generated is being copied from the initiating node and installed on other nodes of the cluster. This can take several minutes to complete.

When each node has received and installed the new CA certificate, the 'Abort CA Certificate Rotation' button will disappear on that node. After this button has disappeared across ALL nodes in the cluster, Oracle Key Vault will automatically begin the process of putting this new CA certificate into use across the multi-master cluster and publishing the new CA certificate to the endpoints.

Click on the 'Abort CA Certificate Rotation' button to skip unpacking the newly generated CA certificate and retain the CA certificate presently in use. Please note that this operation is available for only a few minutes, after which the new CA certificate will automatically be deployed across the cluster.

New CA Certificate	
Status	CA certificate rotation initiated in cluster, replicating new CA certificate to all nodes in the cluster
Common Name	OKV_CA_2022_09_15_10:53:10
Start Date	15-SEP-2022 10:53:20
End Date	14-SEP-2025 10:53:20

In a multi-master cluster environment, note the following:

- After the start of the certification rotation process, the details of the new certificate that was generated is displayed on the node on which you started the CA rotation. If you refresh the **Manage CA Certificate** page on all of the other nodes, this page displays a message that the new certificate is propagated to that node.
- To access this page, select the **System** tab, select **Settings** in the left navigation bar, select **Service Certificates**, and then select **Manage CA Certificate** in the **Certificates** area.
- The certificate is now distributed to all the nodes. The propagation process takes a few minutes to complete.
- You can abort the certificate rotation before the point where:
 - All nodes in the cluster have received the new CA certificates.
 - Each node has notified the other nodes that it has received the certificate.

Periodically refresh the **Manage CA Certificate** page, in case there are any changes to the rotation status. For example, refresh the page to determine if the **Abort CA Certificate Rotation** button is no longer displayed.

Oracle Key Vault automatically initiates the activation when all nodes receive the new CA certificate and displays the message,

Automatic certificate update of the endpoints is in progress.

After you click **Start CA Certificate Rotation**, you have only a few minutes to cancel the CA certificate rotation process. When the Abort CA Certificate Rotation button is no longer visible, the certificate rotation cannot be aborted and will proceed. It is therefore recommended that you click **Start CA Certificate Rotation** only when required.

Manage CA Certificate
Cancel **Abort CA Certificate Rotation**

CA certificate rotation has been initiated in the cluster.

The CA certificate bundle that was generated is being copied from the initiating node and installed on other nodes of the cluster. This can take several minutes to complete.

When each node has received and installed the new CA certificate, the 'Abort CA Certificate Rotation' button will disappear on that node. After this button has disappeared across ALL nodes in the cluster, Oracle Key Vault will automatically begin the process of putting this new CA certificate into use across the multi-master cluster and publishing the new CA certificate to the endpoints.

Click on the 'Abort CA Certificate Rotation' button to skip unpacking the newly generated CA certificate and retain the CA certificate presently in use. Please note that this operation is available for only a few minutes, after which the new CA certificate will automatically be deployed across the cluster.

New CA Certificate

Status	CA certificate rotation initiated in cluster, replicating new CA certificate to all nodes in the cluster
Common Name	OKV_CA_2022_09_15_10:53:10
Start Date	15-SEP-2022 10:53:20
End Date	14-SEP-2025 10:53:20

The new CA certificate takes a few minutes to propagate to all the nodes and the Manage CA Certificate page on other nodes may show no change in status. Refresh the **Manage CA Certificate** page on the other nodes till the following message is displayed:

Automatic certificate update of the endpoints is in progress.

The new CA certificate is now activated and the Oracle Key Vault servers or nodes begin issuing new endpoint certificates signed by the new CA certificate. The endpoints can now connect to the Oracle Key Vault server or nodes using the endpoint certificate issued by either the new or the old Oracle Key Vault CA. In the background, Oracle Key Vault starts issuing certificates for its endpoints, a few endpoints at a time.

When a new certificate is generated on Oracle Key Vault for an endpoint, it is not delivered to the endpoint right away. The endpoint receives the new certificate the next time it reaches out to Oracle Key Vault, and in particular, to the server or node that has generated the certificate. After the endpoint has received the new certificate, the endpoint must connect to Oracle Key Vault a second time to let the server know that the endpoint has successfully received (and is using) the new certificate. When Oracle Key Vault receives this acknowledgment from the endpoint, Oracle Key Vault updates the **Common Name of Certificate Issuer** field for that endpoint on the Endpoints page to the common name of the new Oracle Key Vault CA certificate.

 **Note:**

Periodically check the status of replication across the cluster by viewing either the Cluster Monitoring page or the Cluster Management page. To access either of these pages, click the **Cluster** tab, and then select either **Management** or **Monitoring** in the left navigation bar.

- To check if the credentials for an endpoint are updated, click the **Check Endpoint Progress** button.

Click the **Check Endpoint Progress** button to display the Endpoints page.

For more information, see, [Checking Certificate Rotation Status for Endpoints](#)

Manage CA Certificate

Cancel
Check Endpoint Progress

Automatic certificate update of endpoints is in progress. There is no downtime for endpoints.

New CA Certificate

Status	Updating endpoints to use the new certificate
Common Name	OKV_CA_2022_01_20_14:20:26
Start Date	20-JAN-2022 16:30:21
End Date	19-JAN-2027 16:30:21

Old CA Certificate

Common Name	CA
Start Date	20-JAN-2022 11:11:42
End Date	19-JAN-2025 11:11:42

- Complete the CA certificate rotation.

After Oracle Key Vault issues certificates to all the endpoints using the new CA certificate, the Oracle Key Vault server rotates the server certificates for standalone and primary-standby environments and the node certificates for the cluster environment.

CA certificate rotation process is complete when the Manage CA Certificate page does not list the certificates but only lists the new CA certificate. In a multi-master cluster environment, to check if rotation is complete, go to each node and check the Manage CA Certificate page for that node. The CA certificate rotation process is complete when the Start CA Certificate Rotation button is available on the Manage CA Certificate page, along with the Current CA Certificate and Current Server Certificate.

The CA certificate rotation process is complete when clicking the **Manage CA certificate** button on the **Service Certificates** page takes you to the **CA Certificate Details** page and you can make a choice between the Self-Signed Root CA and Intermediate CA. In a multi-master cluster environment CA certificate rotation process is complete when certificate rotation is complete on every node of the cluster.

You can initiate another certificate rotation only after all the servers or nodes have completed their certification rotation process. After you complete the rotation, configure an alert for when the new certificate should be rotated next.

 **Note:**

- The CA Certificate rotation process can take several days to complete. Oracle recommends that you start the process ahead of the CA certificate expiration to avoid Oracle Key Vault and endpoint downtime.
- The CA certificate rotation should be completed before expiration of the old CA to avoid disruption to endpoints. However, if you are unable to complete the CA certificate rotation before expiration of the old CA certificate, Oracle Key Vault forces the CA certificate rotation process to completion by rotating the server or node certificates, to avoid downtime for the Oracle Key Vault deployment. If this happens, some endpoints will need to be re-enrolled, as explained here:
 - All endpoints in the deployment that were successfully rotated will continue to operate with no downtime. However, any endpoints that had not yet been rotated, or were still in the process of being updated when the old CA expired, will experience an outage and will need to be re-enrolled.
 - You can determine which endpoints need to be re-enrolled by logging in to the Oracle Key Vault Management console as a user with the System Administrator role, then navigating to Endpoints, and checking the **Common Name of Certificate Issuer** field. Any endpoints whose certificate issuer is not the new CA, or in the **Updating to Current Certificate Issuer** status, will need to be re-enrolled.

Related Topics

- [Backup and Restore Operations](#)
Backups provide the ability to restore Oracle Key Vault to a previous state in the case of a failure.
- [Configuring Oracle Key Vault Alerts](#)
You can select the type of alerts that you want to see in the Oracle Key Vault dashboard.

Setting the Endpoint Certificate Rotation Batch Size

The endpoint certificate rotation batch size value represents the number of endpoints that can be in the `ROTATED` state on a given Oracle Key Vault server or node during the CA certification rotation process.

During the CA certificate rotation process, an endpoint is considered to be in a `ROTATED` state when Oracle Key Vault server or node has issued the endpoint certificate using the new CA certificate but the new endpoint certificate is either not yet received or acknowledged by the endpoint.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

In a primary-standby environment, log in to the primary Oracle Key Vault server.
In a multi-master cluster environment, log in to the node selected for initiating the CA certificate rotation in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.

5. In the **CA Certificate Details** page, select the **Self-Signed Root CA** option or the **Intermediate CA** option (this should be selected by default). Click **Rotate CA Certificate**. For the **Intermediate CA** option, **Rotate CA Certificate** button is displayed only after the intermediate CA and its trust chain has been uploaded.
6. Scroll to the **Endpoint Certificate Rotation Controls** area.
7. Enter a value in the **Endpoint Certificate Rotation Batch Size** field.
Enter a value from 5 through 50. The default is 15.
8. Click **Save**.

Setting the Endpoint Certificate Rotation Sequence

In a multi-master cluster environment, when you rotate certificate authority (CA) certificate, you broadly set the order in which endpoints can be rotated by ordering cluster subgroups.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

In a primary-standby environment, log in to the primary Oracle Key Vault server.
In a multi-master cluster environment, log in to the node selected for initiating the CA certificate rotation in the cluster.

2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.
5. In the **CA Certificate Details** page, select the **Self-Signed Root CA** option or the intermediate CA option (this should be selected by default). Then click **Rotate CA Certificate**. For the **Intermediate CA** option, **Rotate CA Certificate** button is displayed only after the intermediate CA and its trust chain has been uploaded.
6. Scroll to the **Endpoint Certificate Rotation Sequence** area.
7. Click **Select Cluster Subgroup**.
8. In the **Select Cluster Subgroup Order** dialog box, move the cluster subgroups that contain the endpoints to rotate to the right, and then use the arrow keys to set their order.

For example, if this is your priority list:

- a. ClusterSubgroupA (EP1, EP4)
- b. ClusterSubgroupB (EP2, EP3, EP5)
- c. ClusterSubgroupC (EP6, EP7)

Endpoints EP1 and EP4, which belong to ClusterSubgroupA, will be rotated first. After EP1 and EP4 receive and acknowledge their updated endpoint certificates, the rotation process will move to the next set of endpoints, ClusterSubgroupB (EP2, EP3, EP5).

You can check if an endpoint has received and acknowledged its new certifications by navigating to the Endpoints page. The endpoint's Certificate Issuer field will change from **Updating to Current Certificate Issuer** to **DN_of_new_OKV_CA**.

 **Note:**

If you specify the cluster subgroup priority order, then the number of endpoints that are processed at a time may be less than the **Endpoint Certificate Rotation Batch Size** parameter. For instance, if a given cluster subgroup has far fewer endpoints associated with it than the Endpoint Certificate Rotation Batch Size parameter, then only endpoints from the chosen cluster subgroup will be processed. Oracle Key Vault server or node does not begin processing of endpoints from other cluster subgroups with the lower priority order until certificate rotation is complete for all of the endpoints from the current cluster subgroup.

9. Click **Apply.**

Cluster subgroups are usually used to group endpoints in a region or data center. Since the re-issue of endpoint certificates during CA certificate rotation could be a time consuming process, it is convenient to process endpoints per cluster subgroup for operations simplicity.

Checking Overall Certificate Rotation Status

Use the Oracle Key Vault management console to check the overall status of a certificate rotation.

After all the endpoints have been updated to using the new certificate, the Oracle Key Vault server begins the process of fully rotating its own server certificates in the background.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** in the left navigation side bar.
3. In the **Certificates** area, select **Service Certificate**.
By default, **Service Certificate** is selected.
4. Check the **Manage CA Certificate** page.
5. Check the certificate rotation status.

After clicking the **Manage CA Certificate** and if you are directed to the **CA Certificate Details** page, you can make a choice between the **Self-Signed Root CA** and **Intermediate CA**, the certificate rotation is complete. Otherwise it is still in progress.

The End Date field in the **Service Certificates** page should reflect the expiration time of the new CA certificate.

In a multi-master cluster environment CA certificate rotation process is complete when certificate rotation is complete on every node of the cluster.

When a CA certificate rotation is in progress, OKV management console displays a banner on the Home page, as well as **Manage Service Certificates**, to that effect. In a multi-master cluster environment, the presence of the banner on a given node indicates that CA certificate rotation is still in progress on that node.

You can initiate another certificate rotation only after all the nodes have completed their certification rotation process.

Checking Certificate Rotation Status for Endpoints

Use the Oracle Key Vault management console to check the status of a certificate rotation for endpoints.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **Endpoints** tab.
3. Select **Endpoints**.

In the Endpoints page, **Common Name of Certificate Issuer** field tracks the progress of how many endpoints have been issued certificates using the new CA certificate.

The **Common Name of Certificate Issuer** field shows if the endpoint certificate is issued by the Old CA, the new CA, or if the endpoint is in the process of updating its endpoint certificate.

For an endpoint, whose endpoint certificate:

- Has been issued using the new CA certificate, the Common Name of Certificate Issuer field shows the common name of the new CA.
- Is in the process of being issued using the new CA certificate, the Common Name of Certificate Issuer field shows *Updating to Current Certificate Issuer*.
- Has not been issued using the new CA certificate, Common Name of Certificate Issuer field shows the common name of the old CA.

Note:

If there are errors with the certificate rotation of an endpoint, then Oracle recommends that you re-enroll the endpoint.

Post-CA Certificate Rotation Tasks

After you complete the CA certificate rotation, perform the post-rotation tasks.

- If you had previously downloaded the Oracle Key Vault RESTful services software utility (`okvrestclipackage.zip`), then download it again to continue to use the RESTful services utility.
Ensure that you have fully rotated the certificate, across all the nodes in a multi-master cluster environment and in the servers of a primary-standby environment, before you download `okvrestclipackage.zip`.
To do this, select the **Endpoint Enrollment and Software Download** link on the Oracle Key Vault management console login page. Select the **Download RESTful Service Utility** tab, and then click **Download** to download the `okvrestclipackage.zip` file to a secure location.
- **Update the backup destinations**
After the CA certificate rotation, each server or node will have been issued a new certificate. The public key of the Oracle Key Vault node or server will also have changed. You need to copy the public key that appears in the **Public Key field** on the **Backup Destination Details** page and then paste it in the appropriate configuration file, such as `authorized_keys`, on the backup destination server.

To do so, navigate to the System tab, then Settings in the left navigation side bar. In the System Configuration area, select Backup and Restore. Click the Manage Backup Destination to view all backup destinations. Click the **Create** button. The **Public Key** field will have the new public key.

- **Back up all Oracle Key Vault nodes and servers.**

It is important to perform this backup operation after the certificate rotation is complete. Later, if you have to restore a backup, the backup to restore must have been initiated after the CA certificate rotation. Restoring the backup before the CA certificate rotation can make the Oracle Key Vault server available but then the endpoints will not be able to connect to the restored Oracle Key Vault server. The CA certificate of the restored system may have expired and the endpoints would be using the endpoint certificates issued by the new CA not present in the backup done before CA certificate rotation.

Factors Affecting CA Certificate Rotation Process

Consider these factors that affect the certificate authority (CA) certificate rotation process in cluster environments.

The duration of CA certificate rotation is determined by how quickly the CA, node, and endpoint certificates are rotated. The endpoint certificate rotation takes the most time.

During the CA certificate rotation process, Oracle Key Vault rotates certificates for endpoints in batches on each node of the cluster, with an upper limit on the number of endpoints that are allowed to be in the ROTATED state at any one time. The number of endpoints that can be in a ROTATED state at any given time on an Oracle Key Vault node is defined by the endpoint certificate rotation batch size. The endpoint must receive its new certificate from the rotating node and acknowledge the receipt of the certificate back to the rotating node. An endpoint must have created at least one object for it to receive the certificate.

 **Note:**

Generally, the node that rotates an endpoint's certificate is one of those in the endpoint's affiliated cluster subgroup.

The following factors affect the endpoint certificate rotation process:

- In order to receive the new certificates, the endpoint must reach out to the rotating node on which its certificates have been generated. Since the endpoint can communicate with any node in the **endpoint node scan list**, the endpoint may run many operations before it reaches the rotating node and receives its certificate. The endpoint also has to acknowledge the receipt of the new certificates by reaching out to a node in the cluster.
- The endpoint certificate rotation times increases with the number of nodes in the cluster. The endpoints prioritize the nodes in the local subgroup, hence consider setting a different subgroup for each node during the CA certificate rotation.
- In a multi-master cluster environment, an endpoint must connect to the rotating node to receive its new certificate. In a cluster with multiple nodes, the endpoint may reach out to any node in the scan list. If this happens during the CA certificate rotation, then the endpoint makes an attempt to also reach out to the rotating node to receive its new certificate in as few operations as possible. However, depending on the number of nodes in the cluster and the periodicity with which the endpoint reaches out to Oracle Key Vault, the endpoint certificate rotation can take a long time to complete. In large deployments with many endpoints, given that each endpoint may take several days to complete rotation, the overall CA certificate rotation process may take many weeks to complete. Hence, to avoid

downtime, Oracle recommends that you complete the CA certificate rotation well ahead of the CA expiration.

- In a multi-master cluster environment, you can view the rotating node of the endpoint in the Oracle Key Vault management console. To do this, navigate to the **Endpoints** page, click **Actions**, then **Select Columns**. Add the **Rotating Node** column to the list of columns to display, and click **Apply**. Alternatively, the rotating node is also visible on the Endpoint Details page for a specific endpoint.

 **Note:**

You can view a rotating node only when a certificate rotation is in-progress.

- The endpoint certificate rotation batch size applies to each node of the cluster. So, if the endpoints are created on each node evenly, each node will rotate the number of endpoints equal to the batch size simultaneously. However, if all the endpoints are created on a single node, then the certificate rotation burden for all the endpoints will fall on that one node instead of being distributed across other nodes.
- For faster endpoint certification rotation and general load balancing in the cluster, consider distributing the endpoint creation among all nodes of the cluster.
- If the endpoints were created before an upgrade from Oracle Key Vault release 12.2, then the endpoints may all be associated with one single node. This can make the rotation process slower than if the endpoints had been created on different cluster nodes.
- An endpoint can only successfully receive an update if it has at least one object uploaded to the Oracle Key Vault server. You can check if the endpoint has objects by executing the `okvutil list` command.
- For any endpoint stalling the endpoint certificate rotation, consider endpoint re-enroll or running the `okvutil list` command. You can also suspend or delete the endpoint.

Related Topics

- [Setting the Endpoint Certificate Rotation Batch Size](#)
The endpoint certificate rotation batch size value represents the number of endpoints that can be in the `ROTATED` state on a given Oracle Key Vault server or node during the CA certification rotation process.
- [Cluster Subgroups](#)
A cluster subgroup is a group of one or more nodes of the cluster.
- [Deleting, Suspending, Reenrolling, or Rotating Endpoints](#)
When endpoints no longer use Oracle Key Vault to store security objects, you can delete them. You can also suspend, and later resume them when they are needed. You can also re-enroll or rotate endpoints when necessary.

Guidelines for Managing CA Certificate Rotations

Consider these Oracle Key Vault guidelines for managing certificate authority (CA) certificate.

Guidelines for Endpoint Software Versions

- For self-signed root CA certificate rotation, ensure that all the endpoints software are at version 18.2.0.0.0 or later.
- For intermediate CA certificate rotation, ensure that all the endpoints software are at version 21.4.0.0.0 or later.

- Upgrade the endpoint software to the same version as Oracle Key Vault before initiating a CA certificate rotation to ensure that the latest fixes to certificate rotation are also available on the endpoint software.

Recommendations for CA Certificate Rotation

- In a multi-master cluster environment, Oracle recommends that you initiate the rotation from one node only. Use this node to complete the CA certificate rotation process. In case a node is made unavailable during certificate rotation, pick another node and use that node to complete the rest of the CA certificate rotation process. Do not switch nodes while performing certificate rotation.
- Before performing a CA certificate rotation, back up the Oracle Key Vault system.
- If a given endpoint does not receive its re-issued endpoint certificate due to network or other issues, Oracle recommends that you re-enroll the endpoint. If it is unused and no longer needed, you can also choose to suspend or delete it.
- If an endpoint uses the persistent master encryption key cache, it is recommended that the **PKCS11 Persistent Cache Refresh Window** parameter should be set to a large value before initiating a CA certificate rotation process. Optionally, you can also reduce the **PKCS11 Persistent Cache Timeout** parameter to a lesser value for the duration of the CA certificate rotation only, so that the endpoint connects to Oracle Key Vault more frequently, therefore completing its certificate rotation faster.

Checks Before Initiating CA Certificate Rotation

- Before beginning certificate rotation, ensure that the recovery pass phrase is the same across all multi-master cluster nodes.
- You cannot perform a CA certificate rotation when a backup operation or a restore operation is in progress.
- Depending on the deployment, the CA certificate rotation process can take several days to complete, begin the CA certificate rotation well in advance of the CA certificate expiry.
- Before beginning the CA certificate rotation, identify all unused endpoints and either delete or suspend them. Suspended endpoints will be skipped during a CA certificate rotation and will not be issued with a new certificate issued by the new CA. If you do not delete or suspend such endpoints, the CA certificate rotation will stall and you will need to re-enroll those endpoints to allow the rotation to complete.
- You can identify unused endpoints from the Oracle Key Vault management console by navigating to Endpoints, then clicking on the Endpoints tab in the left navigation bar. This brings up the Endpoints page, listing all endpoints in the deployment. Check the Last Active Time column to determine when a given endpoint last reached out to Oracle Key Vault. You can delete or suspend all endpoints whose Last Active Time column shows that they have been inactive.
- Ensure node addition is not in progress. Do not initiate a CA certificate rotation while a node addition is in progress.
- Ensure any node operation is not in progress. Do not try node operations (such as adding or disabling nodes) when a CA certificate rotation is in process.
- In the multi-master cluster environment, ensure all the nodes are active. Do not initiate CA certificate rotation till all nodes in the cluster are active. You can check if a node is active by checking the **Cluster Monitoring** page. Click the **Cluster** tab, and then select **Monitoring** from the left navigation bar.
- In a primary-standby environment, ensure the primary server is active. Do not perform CA certificate rotation if the primary server is in read-only restricted mode. Only initiate a CA

certificate rotation when both servers in the configuration are active and synchronized with each other.

- Ensure endpoint certificate rotation is not in progress. Do not initiate a CA certificate rotation while an endpoint certificate rotation is being performed.

Expired CA Certificate

- Do not upgrade Oracle Key Vault if the CA certificate has already expired. The upgrade will fail.
- In Oracle Key Vault release 21.5 and later, you cannot start a CA certificate rotation if the CA has already expired. You must generate a new CA certificate manually and re-enroll all endpoints instead. See section 17.6, *Managing the Oracle Key Vault CA Certificate after it has expired*, for details on how to do so. In Oracle Key Vault 21.4 and earlier, contact Oracle Support.

Certificate Rotation for Non-Oracle Database

- For an endpoint that does not automatically reach out to the Oracle Key Vault server (for example, an ACFS endpoint), it is recommended that you use the `okvutil list` command to force the endpoint to get the new endpoint certificate.
- You may need to run the `okvutil list` more than once to ensure that the command reaches the cluster node that regenerated the endpoint's certificate. Also, ensure that the endpoint has access to at least one security object.

Related Topics

- [Certificates and the Restore Operation](#)
A third-party certificate installed at the time of a backup will not be copied when you restore another server from this backup.

Managing Server Certificates and Node Certificates Rotation

Use the Oracle Key Vault management console to rotate server or node certificates.

- [About Server Certificates and Node Certificates Rotation](#)
Oracle Key Vault uses server certificates to communicate with its endpoints. Oracle Key Vault cluster nodes use node certificates to communicate with each other and with the endpoints.
- [Configuring Certificate Validity Period for Server and Node Certificates](#)
You can configure the validity period for server or node certificates in the Oracle Key Vault management console.
- [Rotating Server Certificates and Node Certificates](#)
You can rotate server certificates and node certificates in the Oracle Key Vault management console.
- [Guidelines for Rotating Server Certificates and Node Certificates](#)
Review these guidelines before you perform a rotation of server certificates or node certificates.

About Server Certificates and Node Certificates Rotation

Oracle Key Vault uses server certificates to communicate with its endpoints. Oracle Key Vault cluster nodes use node certificates to communicate with each other and with the endpoints.

These certificates are referred to as server certificates for standalone and primary-standby configurations and as node certificates in multi-master cluster configurations. The Oracle Key Vault certificate authority (CA) certificate issues these certificates.

You can rotate just these certificates, independently of the CA certificate rotation process. Doing so has no impact on the certificate expiry dates of the Oracle Key Vault CA or on any endpoints.

It is useful to rotate just the server and node certificates in situations where the Oracle Key Vault CA is still valid for much longer, but the server node certification will expire soon. This can happen because the CA validity is usually longer than the server or node certification validity.

The server or node certificate rotation process is described as follows:

- Set the validity of the server or node certificate
- Rotate server or node certificate

Configuring Certificate Validity Period for Server and Node Certificates

You can configure the validity period for server or node certificates in the Oracle Key Vault management console.

The certificate validity period takes effect the next time you rotate the server or node certificates. It will also be taken into account when you generate the server or node certificates as part of a CA certificate rotation, or when you add a new node to the cluster, to the node certificates for that new node. Irrespective of the value that the server or node certificate validity is set to, when the certificates are eventually generated, Oracle Key Vault ensures that their expiry date is less than that of the CA certificate.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

In a multi-master cluster environment, you can log in to any node in the cluster.

2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, select **Service Certificates**.
4. Depending on your environment, perform the following:
 - In a standalone or primary-standby environment: In the Current Server Certificate area, select **Manage Server Certificate**.
 - In a multi-master cluster environment: In the Current Node Certificate area, select **Manage Node Certificate**.
5. In the **Server Certificate Validity (in days)** or **Node Certificate Validity (in days)** field, enter a value between 365 days (the minimum and the default) and 1095 days for this setting.
6. Click **Save**.

Rotating Server Certificates and Node Certificates

You can rotate server certificates and node certificates in the Oracle Key Vault management console.

Before you perform the rotation, ensure that you read the guidelines for rotating server certificates and node certificates.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
In a multi-master cluster environment, you can log in to any node in the cluster.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, select **Service Certificates**.
4. Depending on your environment, perform the following:
 - In a standalone or primary-standby environment: In the Current Server Certificate area, select **Manage Server Certificate**.
 - In a multi-master cluster environment: In the Current Node Certificate area, select **Manage Node Certificate**.
5. If required, in the **Server Certificate Validity (in days)** field (for standalone or primary-standby environments) or **Node Certificate Validity (in days)** field (for multi-master cluster environments), enter a value between 365 days (the minimum and the default) and 1095 days for this setting.

Wait several minutes to make sure that this setting takes effect, particularly in a multi-master cluster environment. When the change is visible across all cluster nodes (navigate to the same page on each node to verify), you are ready to initiate a server or node certificate rotation.

6. Depending on your environment, do the following:
 - **In a standalone or primary-standby environment:** Select **Generate Server Certificate**.
 - **In a multi-master cluster environment:** Select **Generate Node Certificate**.
7. In the confirmation window, click **OK**.

This process can take several minutes to complete. It may also result in a momentary disruption of endpoint servicing.

If the process successfully completes, then the Current Server Certificate (for standalone or primary-standby environments) and the Current Node Certificate (for multi-master cluster environments) sections display new values for the End Date and Expiring in settings. In a multi-master cluster environment, you can view the expiry dates of all the node certificates in the cluster in the Cluster Node Certificate Details area.

Guidelines for Rotating Server Certificates and Node Certificates

Review these guidelines before you perform a rotation of server certificates or node certificates.

- Do not perform a certificate authority (CA) certificate rotation while a server or node certificate rotation is in progress.
- Do not perform a server or node certificate rotation while a CA certificate rotation is in progress.
- Do not perform a server or node certificate rotation while a Endpoint certificate rotation is in progress.
- Do not perform a node certificate rotation on one node while another is in progress on a different node.
- Do not alter the CA certificate validity period while a CA certificate rotation is in progress.
- Do not attempt to rotate the server certificates if the CA certificate is already expired.

- Do not alter the **Server Certificate Validity (in days)** field (for standalone or primary-standby environments) or **Node Certificate Validity (in days)** field while either a CA certificate rotation or a server or node certificate rotation is in progress.
- After a server or node certificate rotation, each server or node is issued a new certificate. The public key of the Oracle Key Vault server or node is also changed. You must copy the public key that appears in the **Public Key** field in the **Backup Destination Details** page and paste it in the appropriate configuration file, such as `authorized_keys`, in the backup destination server. See, [Post-CA Certificate Rotation Tasks](#) for the steps to get the new Public Key.

Managing the Oracle Key Vault CA Certificate After Expiry

You cannot start a CA certificate rotation when the Oracle Key Vault CA certificate has already expired.

When the endpoints cannot communicate with Oracle Key Vault it results in an outage. As a result, in a multi-master cluster environment, Oracle Key Vault nodes cannot communicate with each other. In such a scenario, you must regenerate a new CA certificate manually using the following steps.

In a multi-master cluster environment, this new CA certificate must be distributed from the generating node to all other nodes. After the CA certificate has been distributed across the cluster, you must rotate the node certificates on each cluster node in turn.

Finally, you must re-enroll all endpoints because this involves endpoint outage until they have been re-enrolled. Oracle recommends that you configure alerts for CA certificate expiration and complete CA certificate rotation, see [Managing CA Certificate Rotation](#) before the CA certificate expires, in preference to the following steps. This ensures that the CA certificate rotation is completed with minimal disruption to endpoints and in a multi-master cluster environment, to the Oracle Key Vault cluster nodes.

Back up Oracle Key Vault before commencing with these steps.

To manually issue a new CA certificate after the current CA certificate has expired, perform the following steps.

1. Log into the Oracle Key Vault management console as a user with the System Administrator role.
 - In a primary-standby environment: Log into the Oracle Key Vault primary server.
 - In a multi-master cluster environment: Select a node to generate a new CA certificate and log into that node. If you want to enable an intermediate CA certificate, then log into the node where the intermediate CA certificate was previously uploaded.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the **Service Certificates** page, select **Manage CA Certificate**.

The current CA certificate details including its start and end date displays, as well as a message to the effect that the CA certificate has expired.

5. Click on **Check Expired CA Certificate Rotation Status**.

6. In the confirmation dialog box, click **OK**. In the backend, this performs a series of checks on the Oracle Key Vault system.
7. In a multi-master cluster environment, you **must** perform steps 1 - 6 on each node of the cluster.
8. On successful validation, the **Generate New CA Certificate** button becomes active. In a multi-master cluster environment, an additional button, **Upload CA Certificate Bundle**, is shown.

Note:

These buttons are shown on each node of a multi-master cluster only if the checks are completed successfully on that node. If the checks are unsuccessful on any cluster node or if the Generate New CA Certificate and Upload CA Certificate Bundle buttons do not show on every node of the cluster, then do not proceed to the next step. Contact Oracle Support.

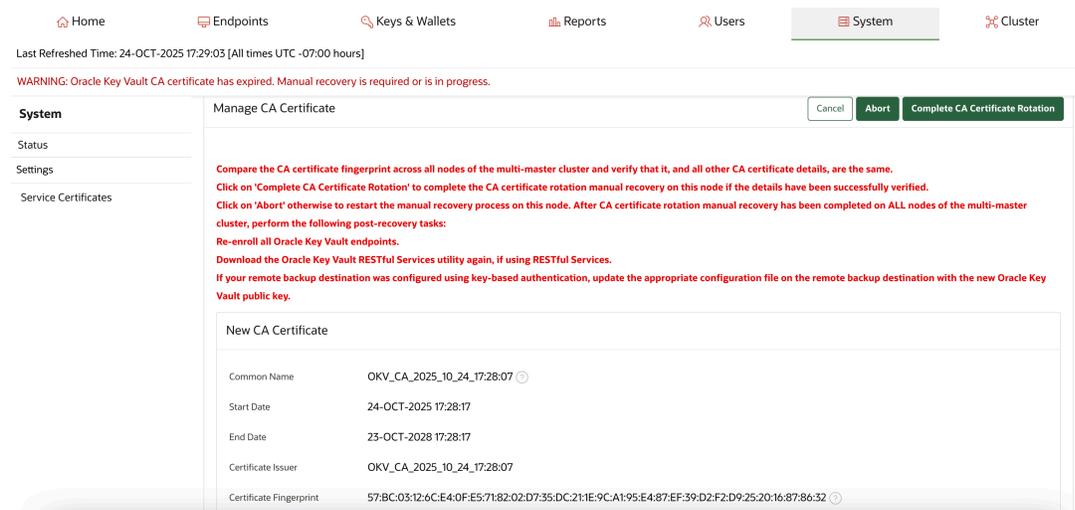
9. On the Oracle Key Vault system selected for generation of the new CA certificate in step 1, click **Generate New CA Certificate**.
In a primary-standby environment, this is the Oracle Key Vault primary server.

In a multi-master cluster environment, this is the cluster node that was chosen in step 1 to generate the new CA. If you want to enable an intermediate CA certificate, then this is the node that the intermediate CA was uploaded on before the current CA expired.

10. Click **OK** in the confirmation box. In the backend, this generates a new CA certificate.

In a multi-master cluster environment, this also creates a certificate bundle that is made available to download. The bundle must be stored in a safe place because it must be distributed to all cluster nodes.

11. Refresh the page on the Oracle Key Vault management console on which the new CA certificate was generated. The **Complete CA Certificate Rotation** button is now active. Additionally, other details of the new CA certificate, such as the certificate common name and certificate fingerprint are displayed.



In a multi-master cluster environment, proceed to step 12. In standalone and primary-standby environments, proceed to step 15.

12. In a multi-master cluster environment, log into a node of the cluster other than the one on which the CA certificate was generated. Click on **Upload CA Certificate Bundle** and upload the bundle that was downloaded in step 10. When this successfully completes, details of the new CA are displayed, including the common name of the CA and its certificate fingerprint. Compare these details with those displayed on the generating node (see step 11).
13. If any of the details of the new CA certificate uploaded in step 12 do not match with the details displayed on the generating node in step 11, then click **Abort** and repeat step 12 again. Proceed to step 14 only after the CA certificate details on this node match with that of the generating node.
14. Perform step 12 (and if necessary, step 13) on every node of the cluster other than the generating node. Proceed to step 15 only after the CA certificate bundle has been uploaded to all nodes of the cluster, and the certificate details have been verified to be an exact match across all nodes.
15. Click **Complete CA Certificate Rotation**. In the confirmation dialog box, click **OK**. This process may take several minutes to complete.

In the backend, this generates new server or node certificates, issued by the new CA that was generated in step 9. In a primary-standby environment, this operation must be performed on the Oracle Key Vault primary server. In a multi-master cluster environment, this operation must be performed on **every node of the cluster**, one after the other.

16. In case of an error, click **Abort**. In a multi-master cluster environment, you can upload the certificate bundle and try performing the process again.
17. You can also select **Complete CA Certificate Rotation** again.

 **Note:**

In a multi-master cluster environment proceed to next steps only after successful certificate validation on all nodes.

18. In a multi-master cluster environment, check the **Cluster Monitoring** pages, verify that communication is restored between all cluster nodes. Also consider testing replication between nodes by creating a wallet on each node of the cluster and verifying that it transitions from the **PENDING** to **ACTIVE** state.
19. Re-enroll all the Oracle Key Vault endpoints.
20. Complete the recommended post-recovery tasks - download the Oracle Key Vault RESTful services utility again and update the Oracle Key Vault remote backup destination configuration file with the new public key, if required.

Configuring Oracle Key Vault with an Alternate Hostname

Learn how to configure Oracle Key Vault with an alternate hostname.

- [About Configuring Oracle Key Vault with an Alternate Hostname](#)
You can configure Oracle Key Vault with an alternate hostname, that is, a fully-qualified domain name (FQDN) or a secondary IP address.
- [Configuring Oracle Key Vault Alternate Hostname on the Management Console](#)
You can configure alternate hostnames on the Oracle Key Vault management console. The alternate hostname must be a valid IP address or a fully-qualified domain name (FQDN).
- [Choosing the Alternate Hostname to Use in Endpoint Configuration](#)
After successfully configuring Oracle Key Vault with one or more alternate hostnames, you can choose one of these alternate hostnames as the identity that endpoints will use when connecting to the Oracle Key Vault server/node.
- [Guidelines for Configuring Alternate Hostnames](#)
Review these guidelines before configuring an alternate hostname for endpoints to communicate with Oracle Key Vault.

About Configuring Oracle Key Vault with an Alternate Hostname

You can configure Oracle Key Vault with an alternate hostname, that is, a fully-qualified domain name (FQDN) or a secondary IP address.

An Oracle Key Vault system is configured with an IP address when it is first installed (modifiable until it is converted to a multi-master cluster node). Its endpoints communicate with it using this IP address, which they read from their configuration files `okvclient.ora` or `okvrestcli.ini`. The Oracle Key Vault IP address therefore serves as the primary identity of the server/node for communication with its endpoints. In the case of systems deployed on Oracle Cloud Infrastructure (OCI) compute instances, which may have two IPs (a public IP and a private IP), endpoint communication is using the private IP by default.

You can configure Oracle Key Vault to allow its endpoints to communicate with it using a FQDN or an alternate IP address (hereinafter referred to as *alternate hostname*). The configuration is

a two-step process. First, provide the alternate hostname as input and regenerate Oracle Key Vault server/node certificates; next, choose the hostname that you wish endpoints to use when communicating with the server/node.

 **Note:**

- This feature is available only in standalone and multi-master cluster deployments. It is not available in primary-standby deployments (deprecated in Oracle Key Vault release 21.5).
- The networking changes to set up an alternate IP address or fully-qualified domain name are out of scope of this document. The steps refers only to changes that must be made on the Oracle Key Vault server/node to allow endpoint communication by using the new alternate hostname.

Configuring Oracle Key Vault Alternate Hostname on the Management Console

You can configure alternate hostnames on the Oracle Key Vault management console. The alternate hostname must be a valid IP address or a fully-qualified domain name (FQDN).

Up to two alternate hostnames can be configured for a given Oracle Key Vault server/node. You can choose one of these alternate hostnames (or the Oracle Key Vault server/node IP address) as the hostname for endpoints to use when communicating with the system.

Configuring the alternate hostname requires rotating the Oracle Key Vault server/node certificates. Ensure that you follow the guidelines for server/node certificate rotation when doing so.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role. In a multi-master cluster environment, you must log into the management console of the node whose alternate hostname you are configuring.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, select **Service Certificates**.
4. In a standalone environment, select **Manage Server Certificate**. In a multi-master cluster environment, select **Manage Node Certificate**. Alternate hostnames cannot be configured in primary-standby deployments.
5. Depending on the environment, do the following:
 - a. In a standalone environment, on the **Server Certificates Details** page, scroll down to the **Current Server Alternate Hostname** area. Enter a valid IP address or fully-qualified domain name (FQDN) in the **Alternate Hostname1** field. If desired, enter a different IP address or FQDN in the **Alternate Hostname2** field. Select **Generate Server Certificate**.
 - b. In a multi-master cluster environment, on the **Node Certificates Details** page, scroll down to the **Current Node Alternate Hostname** area. Enter a valid IP address or fully-qualified domain name (FQDN) in the **Alternate Hostname1** field. If desired, enter a different IP address or FQDN in the **Alternate Hostname2** field. Select **Generate Node Certificate**.
6. In the confirmation window, select **OK**.

This process can take several minutes to complete.

7. After the server/node certificates have been successfully generated, the alternate hostname(s) can be viewed in the **Current Server Certificate** (in standalone environments) or **Current Node Certificate** (in multi-master clusters) section. In a multi-master cluster environment, the alternate hostname(s) used by each multi-master cluster node can be viewed in the **Cluster Node Certificates Details** section of the **Node Certificates Details** page.

Choosing the Alternate Hostname to Use in Endpoint Configuration

After successfully configuring Oracle Key Vault with one or more alternate hostnames, you can choose one of these alternate hostnames as the identity that endpoints will use when connecting to the Oracle Key Vault server/node.

1. Log into the Oracle Key Vault management console as a user who has the System Administrator role. In a multi-master cluster environment, you must log into the management console of the node whose alternate hostname you are configuring.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates area**, select **Service Certificates**.
4. In a standalone environment, select **Manage Server Certificate**. In a multi-master cluster environment, select **Manage Node Certificate**. Alternate hostnames cannot be configured in primary-standby deployments.
5. Depending on the environment, do the following:
 - a. In a standalone environment, scroll down to the **Current Server Alternate Hostname** section.
 - b. In a multi-master cluster environment, scroll down to the **Current Node Alternate Hostname** section.
6. In the **Hostname to use in Endpoint Configuration** drop-down menu, choose the desired hostname for endpoints to use, and select **Save**.
7. In the confirmation window, select **OK**. In a multi-master cluster environment, wait for a few minutes for this change to propagate to all nodes of the cluster.

New endpoints that are registered with Oracle Key Vault now use this desired hostname to communicate with the Oracle Key Vault server/node. Existing Oracle Key Vault endpoints will be notified of the alternate hostname when they next reach out to Oracle Key Vault, and subsequently use the alternate hostname for communication.

Guidelines for Configuring Alternate Hostnames

Review these guidelines before configuring an alternate hostname for endpoints to communicate with Oracle Key Vault.

- Configure the alternate hostname during initial setup of the Oracle Key Vault deployment, before registering endpoints.
- In a multi-master cluster environment, each node must be given its own (unique) alternate hostname. Configure the alternate hostname one node at a time, after all nodes have been added to the multi-master cluster.
- Alternate hostnames cannot be configured or used by endpoints in primary-standby deployments.
- Configuring an alternate hostname requires rotating the Oracle Key Vault server/node certificates. Follow the guidelines for rotating server/node certificates during the configuration.

- Up to two alternate hostnames can be configured for a given Oracle Key Vault server/node. However, only one of these values may be chosen as the hostname for endpoints to use when reaching out to the server/node.
- Before updating the **Hostname to use in Endpoint Configuration**, verify network connectivity using the desired alternate hostname.

Related Topics

- [Endpoint okvclient.ora Configuration File](#)
Oracle Key Vault endpoint libraries and utilities use the `okvclient.ora` configuration file, which stores the configuration parameters associated with the endpoint.
- [About Launching the Oracle Key Vault Compute Instance](#)
The launch process requires some minor preparation work on your system.
- `okvrestcli.ini` Configuration Parameters
- [Guidelines for Rotating Server Certificates and Node Certificates](#)
Review these guidelines before you perform a rotation of server certificates or node certificates.

Managing Platform Certificates

Rotate platform certificates before they expire using the Oracle Key Vault management console.

- [About Platform Certificates](#)
Oracle Key Vault platform certificates are used when adding a new node to an Oracle Key Vault multi-master cluster. Platform certificates are also used when shipping redo between read/write nodes in the cluster.
- [Guidelines for Rotating Platform Certificates](#)
Consider these guidelines when rotating platform certificates.
- [Rotating Platform Certificates](#)
Learn how to rotate platform certificates in a multi-master cluster environment.

About Platform Certificates

Oracle Key Vault platform certificates are used when adding a new node to an Oracle Key Vault multi-master cluster. Platform certificates are also used when shipping redo between read/write nodes in the cluster.

Platform certificates are managed using a different process than the Oracle Key Vault service certificates. You must rotate the Oracle Key Vault platform certificates before they expire.

Rotating the Oracle Key Vault platform certificates does not rotate the Oracle Key Vault service certificates and does not impact endpoint communication with Oracle Key Vault. Similarly, rotating Oracle Key Vault service certificates does not rotate the platform certificates.

Note:

You *cannot* initiate a platform certificate rotation in **deprecated primary-standby deployments**. For more information, see [Rotating Platform Certificates on a Standalone Oracle Key Vault Server](#).

Guidelines for Rotating Platform Certificates

Consider these guidelines when rotating platform certificates.

- You must be a user who has the System Administrator role to rotate the platform certificates.
 - Platform certification operation is possible only when all nodes of the cluster are of version Oracle Key Vault 21.10 or later.
 - You cannot rotate the platform certificates when the cluster is being upgraded.
 - You can initiate the platform certificates rotation on any cluster node. Other nodes will be automatically notified. You do not have to initiate on other cluster nodes and you should avoid this.
 - The platform certificate rotation process is not allowed when any of the following cluster operations are in progress:
 1. CA certificate rotation
 2. Cluster services port change
 3. Recovery Passphrase change
 4. Adding a node
 5. Disabling a node
 6. Converting a server node to a cluster node
- Conversely, you cannot initiate these operations when platform certificate rotation is in-progress.
- Deleting or force-deleting a node is always possible, so that there are no restrictions when performing either a delete or a force-delete when a platform certificate rotation is in-progress.

Rotating Platform Certificates

Learn how to rotate platform certificates in a multi-master cluster environment.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the **Certificates** area, click **Service Certificates**.
4. In the Service Certificates page, select **Manage Platform Certificates**. The **Platform Certificates Details** page appears.

Platform Certificates Details
Cancel Start Platform Certificates Rotation

You can rotate the Oracle Key Vault platform certificates by initiating a platform certificate rotation. Oracle Key Vault will generate new platform certificates that are then put into use on all systems in the deployment. Click on the 'Start Platform Certificates Rotation' button to initiate the process.

Current Platform Certificates

End Date 21-FEB-2027 17:44:40 ⓘ

Expiring In 803 days (ACTIVE)

Node ID	Node Name	IP Address	Status
1	n1	10.32.19.198	IN USE
2	n2	10.32.19.215	IN USE
3	n3	10.32.19.207	IN USE

1 - 3

The **Current Platform Certificates** window pane displays the following:

- **End Date:** The platform certificate expiry date with time. The expiration date displayed is the minimum of the expiration dates of the platform certificates used for Oracle Key Vault operations.
- **Expiring In:** The number of days the platform certificates will be active.
- **Node ID:** The ID of the node.
- **Node Name:** The name of the node.
- **IP Address:** The IP address of the node.
- **Status:** The current status of the node.

 **Note:**

After you start the platform certificates rotation process, the status of the node changes according to the stage of the platform certificate rotation.

5. Click **Start Platform Certificates Rotation** button. A confirmation dialogue box for starting the platform certificate rotation is displayed.
6. Select **Yes**. The platform certificate rotation process starts in the background. The Oracle Key Vault management console displays the platform certificate rotation notification with a message as shown in the following image:

Platform Certificates Details
Cancel **Abort**

Platform certificates rotation has been initiated in the cluster. The platform certificate bundle that was generated is being copied from the initiating node and installed on other nodes of the cluster. This can take several minutes to complete. When each node has received and installed the new certificates, the 'Abort' button will disappear on that node. After this button has disappeared across ALL nodes in the cluster, Oracle Key Vault will automatically begin the process of putting the new certificates into use across the multi-master cluster. Click on the 'Abort' button to skip unpacking the newly generated platform certificates and retain those presently in use. Note that this operation is available for only a few minutes, after which the new platform certificates will automatically be deployed across the cluster.

Current Platform Certificates

End Date 21-FEB-2027 17:44:40 ⓘ

Expiring In 803 days (ACTIVE)

Node ID	Node Name	IP Address	Status
1	n1	10.32.19.198	INITIATED
2	n2	10.32.19.215	INITIATED
3	n3	10.32.19.207	INITIATED

1 - 3

- The **Abort** button appears on the Platform Certificate Details page.

Note:

You can select the **Abort** button to prevent the platform certificates rotation process from proceeding. The **Abort** button disappears after a few minutes and then you can no longer abort the process. You might have to refresh the web console to check if you are past the stage where abort is possible.

- After the **Abort** button disappears from the **Platform Certificate Details** page, the platform certificate rotation process is initiated with the following message:

```
Platform certificates rotation has been initiated.
In a few minutes, Oracle key Vault will automatically start to put these
new certificates into use.
```

After the platform certification rotation process is complete, the **End Date** field is updated with the new expiry date.

Managing Console Certificates

You can use the Oracle Key Vault management console to manage console certificates.

- [About Managing Console Certificates](#)
Oracle Key Vault enables you to install a certificate signed by a Certificate Authority (CA) for more secure connections.
- [Step 1: Download the Certificate Request](#)
When you request the console certificate, you can suppress warning messages.
- [Step 2: Have the Certificate Signed](#)
After you download the Oracle Key Vault `certificate.csr` file, you can have it signed.
- [Step 3: Upload the Signed Certificate to Oracle Key Vault](#)
In addition to uploading the signed certificate, you can optionally choose to deactivate and re-activate the certificate.
- [Console Certificates in Special Use Case Scenarios](#)
Depending on the situation, you must perform additional steps when you use console certificates.

About Managing Console Certificates

Oracle Key Vault enables you to install a certificate signed by a Certificate Authority (CA) for more secure connections.

You can upload a certificate that was signed by a third-party CA to Oracle Key Vault to prove its identity, encrypt the communication channel, and protect the data that is exchanged throughout the Oracle Key Vault system.

To install a console certificate, you must generate a certificate request, get it signed by a CA, and then upload the signed certificate back to Oracle Key Vault.



Note:

If you are logged in as a user with the monitoring privilege, you cannot view the console certificate page.

Step 1: Download the Certificate Request

When you request the console certificate, you can suppress warning messages.

These warning messages appear when the browser detects a mismatch between the attributes of the server certificate and the attributes of the login session to the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.

3. In the Certificates area, click **Console Certificate**.
4. In the Console Certificate page, click **Generate Certificate Request**.

Generate Certificate Request



Generate Certificate Request Cancel Submit and Download

Common Name **okvatlanta** [Change](#)

Common Name is an FQDN (fully qualified domain name) which matches the hostname in the URL used to access this Oracle Key Vault Server.
Either change the host name to FQDN by clicking the Change link above, or add an FQDN as the Subject Alternative Name.

Subject Alternative Name ?

Suppress warnings for IP based URL access

Organization Name *

Country / Region *

Organizational Unit

City

State/Province

Email

5. If you need to change the host name of the Oracle Key Vault server, which appears next to **Common Name**, then click **Change**.
The Network Details window appears, where you can change the **Host Name** setting. Click **Save** afterward.

Note:

If you do not want to change the hostname of the Oracle Key Vault server but still want to use fully qualified domain name (FQDN), you can add the FQDN to the SAN field. Additionally, you can also support two FQDNs for the Oracle Key Vault server one by changing the hostname and the other by adding to the SAN field.

6. Check the box to the left of text **Suppress warnings for IP based URL access** if you want to suppress browser warnings for server IP address changes.
7. Enter the required fields marked with an asterisk, **Organization Name** and **Country / Region**.
You must enter values for these fields in order to proceed without errors. You may enter values in the rest of the optional fields as needed.
8. Click **Submit and Download** to the top right.

A directory window appears, where you can save the `certificate.csr` file. Select a directory and save the file to a secure location.

 **Note:**

In the event that multiple Certificate Signing Requests are generated concurrently, you can verify which is the most recently generated Certificate Signing Request by reloading the Generate Certificate Request page and verifying that the information in the table matches with the correct `certificate.csr` file. If a table isn't populated on the reload of the page, this means that there is a corrupted `certificate.csr` file. To resolve this, generate a new `certificate.csr` file by following the steps outlined earlier [Step 1: Download the Certificate Request](#).

Step 2: Have the Certificate Signed

After you download the Oracle Key Vault `certificate.csr` file, you can have it signed.

- Use any out-of-band method to have the `certificate.csr` file signed by a CA of your choice.

Afterward, you can upload the signed certificate back to Oracle Key Vault using the management console.

Step 3: Upload the Signed Certificate to Oracle Key Vault

In addition to uploading the signed certificate, you can optionally choose to deactivate and re-activate the certificate.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Certificates area, click **Console Certificate**.
4. Click **Upload Certificate** at the top right to display the **Upload Certificate** page.
5. Select **Choose File** to display a directory window on your local system.
6. Navigate to the directory where you stored the signed certificate and select it. When you are done, you will see the file name to the right of text **Choose File**.

After you select the certificate, you will see the file name to the right of **Choose File**.

7. Click **Upload**.

If the certificate is installed with no errors, then you will see its details appear in a new **Uploaded Certificate Details** panel after the **Console Certificate**.

At this stage, if you need to, you can deactivate the certificate by clicking **Deactivate** on the top right of the **Uploaded Certificate Details** section. When you deactivate the certificate, the **Deactivate** button is replaced by an **Apply Certificate** button. You can click this button to re-activate the certificate.

 **Note:**

After having deactivated the certificate, you will be able to reactivate it only until the point that a new certificate request is generated. You must upload a new signed certificate (generated from the new certificate request) rather than reactivating the old certificate in this scenario.

Console Certificates in Special Use Case Scenarios

Depending on the situation, you must perform additional steps when you use console certificates.

- **Primary-standby environments:** If you want to use a console certificate in a primary-standby configuration, then you must install it on the primary and standby servers first, and then pair them.
- **Restored data from a backup:** If you install a console certificate, perform a backup, and then restore another Oracle Key Vault appliance from that backup, you must re-install the console certificate on the new server before you can use it. The restore process does not copy the console certificate.

Backup and Restore Operations

Backups provide the ability to restore Oracle Key Vault to a previous state in the case of a failure.

- [About Backing Up and Restoring Data in Oracle Key Vault](#)
You can use Oracle Key Vault to back up and restore Oracle Key Vault data.
- [Oracle Key Vault Backup Destinations](#)
A backup destination is the location where Oracle Key Vault data will be copied to and stored.
- [Scheduled Backups and States](#)
Oracle Key Vault provides scheduled backup types depending on the backup destination, and different states that indicate the progress of the backup activity.
- [Scheduling and Managing Oracle Key Vault Backups](#)
You can schedule Oracle Key Vault backups to specific backup destinations and times.
- [Restoring Oracle Key Vault Data](#)
Oracle Key Vault data from a remote backup destination can be restored onto another Oracle Key Vault server.
- [Scheduling the Purging of Old Oracle Key Vault Backups](#)
To better manage disk space used by Oracle Key Vault backups on remote destinations, you can schedule the periodic purging of old backups from them.
- [Manually Deleting a Local Oracle Key Vault Backup](#)
You can manually delete a local backup by using the Oracle Key Vault management console.
- [Configuring Oracle ZFS Storage Appliance to Store Oracle Key Vault Backups](#)
Oracle ZFS Storage Appliance is an enterprise storage system that is designed for the storage of data from Oracle products such as Oracle Key Vault.
- [Backup and Restore Best Practices](#)
Oracle provides best practices to keep backups current so that you can recover from catastrophic failures with minimum downtime and data loss.

About Backing Up and Restoring Data in Oracle Key Vault

You can use Oracle Key Vault to back up and restore Oracle Key Vault data.

You should back up data periodically to reduce downtime and recover from unexpected data losses and system failures. You can restore a new or existing Oracle Key Vault server from a backup. When old backups are no longer needed, you can schedule their periodic deletion.

You can perform backup and restore operations from the Oracle Key Vault management console or by using the Oracle Key Vault RESTful services commands. You must be a user who has the System Administrator role to back up and restore Oracle Key Vault data. You can schedule backups at periodic intervals to run automatically at designated times. You also can run these operations on-demand to save a current snapshot of the system.

Oracle strongly recommends that you back up Oracle Key Vault data regularly on a schedule. This practice ensures that backups are current and hold the most recent data. You can use this backup to restore a new or existing Oracle Key Vault server and be fully operational with minimum data loss.

Oracle Key Vault encrypts all backed up data. When you use a remote destination, this data is copied using the secure copy protocol (SCP) or the secure file transfer protocol (SFTP). You must therefore ensure that either SCP or SFTP is supported at the remote backup destination.

In an Oracle Key Vault multi-master cluster environment, the replication intrinsically creates copies of data on other nodes in the cluster. However, you can still perform backups and backup-related operations on all individual Oracle Key Vault cluster nodes. Be aware that backups can still only be restored to standalone Oracle Key Vault servers. Therefore, backups in a cluster are taken for disaster recovery in case of a complete cluster failure and should all be on remote destinations.

 **Note:**

Oracle Key Vault does not support backups taken as snapshot. Restoring from such snapshot backups is not supported.

Oracle Key Vault Backup Destinations

A backup destination is the location where Oracle Key Vault data will be copied to and stored.

- [About the Oracle Key Vault Backup Destination](#)
The backup destination enables the backup data to be available on Oracle Key Vault itself or on another server.
- [Creating a Remote Backup Destination](#)
You can use the Oracle Key Vault management console to create a remote backup destination.
- [Changing Settings on a Remote Backup Destination](#)
After you have created the backup destination, you can change the SCP or SFTP port number and details of the user account.
- [Deleting a Remote Backup Destination](#)
You can delete a remote backup destination (but not the local destination) to stop future backups to that destination server.

About the Oracle Key Vault Backup Destination

The backup destination enables the backup data to be available on Oracle Key Vault itself or on another server.

This ensures that you have all the relevant data to recover in case of a catastrophic failure with the Oracle Key Vault server or hardware.

The backup destination is usually another server or computer system that you have access to. You can add, delete, and modify a backup destination.

The backup operation copies Oracle Key Vault data to a backup destination of your choice. The backup destination stores the data until it is needed.

Oracle Key Vault provides two types of backup destinations: local and remote. The local backup destination resides on the Oracle Key Vault server itself, the remote one resides externally in a different server or computer system. You can create more than one backup destination for greater availability.

Local and remote backup destinations have the following characteristics:

- **Local backup destination:** The local backup destination, `LOCAL`, is present by default and cannot be removed. Backups to the local backup destination are local backups.

Backups to `LOCAL` are useful to save a current state of Oracle Key Vault. Because these backups are stored on disk, they could be lost in the case of hardware or other catastrophic failure. They will also not be available after a failover or switchover in a primary-standby configuration. You cannot restore the backups to a primary-standby without first unpairing the primary from the standby, nor can you restore the backups to a cluster configuration. Therefore, you should back up the data to a remote destination when using these configurations.

A `LOCAL` destination can store only the last full backup and the cumulative incremental backups after that full backup. After a new full backup of the periodic backup to `LOCAL` completes, Oracle Key Vault deletes the previous periodic full or cumulative incremental backup. In addition, you can also delete a backup manually.

- **Remote backup destinations:** Remote backup destinations reside on external servers and can be dispersed geographically for disaster recovery purposes. Backups to remote backup destinations are remote backups. To ensure that the remote backup destination does not accumulate too many backups and hence use up too much disk space, you can schedule a periodic automatic purging of these old backups.

Each backup destination on the external server is associated with a backup catalog file called `okvbackup.mgr` that Oracle Key Vault maintains at the backup destination. The `okvbackup.mgr` file catalogs the backups performed and is used to restore data.

 **Note:**

You cannot use another Oracle Key Vault server as a remote backup destination.

 **Caution:**

- Oracle Key Vault may not be able to find the backups if you delete or modify the backup catalog file. Therefore, do not delete or modify this file.
- Do not configure the same remote backup destination directory for different Oracle Key Vault servers as backup destinations, because backups that happen concurrently from different Oracle Key Vault servers will overwrite each other's catalog file, with the result that Oracle Key Vault may not be able to locate the backups correctly.
- After you restore a backup that contains a remote backup destination, do not continue to use that remote backup destination. Delete any backup jobs that are configured to send backups to that destination. Continuing to use this backup destination could corrupt the backup catalog file. Oracle Key Vault may not be able to locate backups correctly.
- Configure each node in a multi-master cluster to send their backups to a different backup destination.

Related Topics

- [Types of Oracle Key Vault Backups](#)
Oracle Key Vault provides two types of backup jobs that can be scheduled: one-time backups and periodic backups.

Creating a Remote Backup Destination

You can use the Oracle Key Vault management console to create a remote backup destination.

To create a remote backup destination, you must provide a user account, a unique existing directory location on an external server, and an authentication method (password or key-based). Oracle Key Vault needs this information to make a secure connection with the remote server.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, select **Manage Backup Destinations**.
5. Click **Create**.

The Create Backup Destination window appears.

Create Backup Destination

Cancel
Save

Destination Name *

Transfer Method scp sftp

Policy None ▾

Hostname *

Port *

Destination Path *

User Name *

Authentication Method Key-based Authentication Password Authentication

Public Key

ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAQDAQ11D6r0Rm/EG3k1
 /nbc0Dv9C20iU9V0A4LEFD1'sr1kh
 /cDs-rTXn4penDEH9irFVy6Nu5Ga65cIh3qz4FzH6AgAp0
 /cBZeYUJFCFw2f886wNB*rXDXuyPoe+j90b3Suo3RmNSjPrbT
 /UIZcTGi d11qypzvczr02wLqDL89sZJTwT1wUs+sNCBnnHET599

6. Enter the following information for the backup location:

- **Destination Name:** Enter a descriptive name to identify the backup destination.
- **Transfer Method:** Choose between `scp` and `sftp` to copy files to the remote destination.
- **Policy:** Select the backup policy from the list.
- **Hostname:** Enter the host name or IP address of the remote server for the backup destination. If you enter the host name, then ensure that DNS is configured to translate the host name to its corresponding IP address. Do not include spaces, single quotation marks, or double quotation marks in a host name that is in a remote backup destination.
- **Port:** Enter the SCP or SFTP port number on the external server. The default is 22.
- **Destination Path:** Enter the path to an existing directory on the external server where the backup file will be copied. You cannot modify this directory location after the backup destination is created. This path must not be the destination for backups from another Oracle Key Vault server. Do not include spaces, single quotation marks, or double quotation marks destination path that is in a remote backup destination.
- **Username** Enter the user name of the user account that can be used to establish an SCP or SFTP connection to the remote server. Ensure that this user has write permissions on the directory that is specified in **Destination Path**. Do not include spaces, single quotation marks, or double quotation marks in a user name that is in a remote backup destination.
- **Authentication Method:** Choose one of the following:
 - **Key-based Authentication:** Copy the public key that appears and paste it in the appropriate configuration file, such as `authorized_keys`, on the destination server. Be aware that certain events may trigger a change of the public key, which means that Oracle Key Vault cannot use the backup destination until the new public key is re-copied from Oracle Key Vault to the appropriate configuration file. These events

include but are not limited to certificate rotation, changing the IP address, and conversion to a cluster node.

- **Password Authentication:** The password of the user account entered in the **Username** field.

 **Note:**

When using password authentication with `scp` transfer method, the home directory for the user account on the remote server must be present and has the correct permissions set. The remote backup destination creation will fail if the home directory is missing or does not have the correct permissions set.

7. Click **Save**.

Oracle Key Vault validates the input that you supplied to create the backup destination by creating empty test files under both `/tmp` and the directory that you supplied in the **Destination Path** field. If the validation fails, then the backup destination is not created. If this happens, then check values for the user account on the remote server (user name and password or key) and ensure that the directory has write permissions for the user. Finally, ensure that the remote server is active.

Changing Settings on a Remote Backup Destination

After you have created the backup destination, you can change the SCP or SFTP port number and details of the user account.

You cannot change any other setting.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. Select **Manage Backup Destinations**.

The Manage Backup Destinations page appears displaying `LOCAL` and remote backup destinations.

5. Click the **Edit** icon for the backup destination that you want to modify.

The Edit Backup Destination page appears.

6. Modify the following information:
 - **Policy:** Select a backup destination policy from the list.
 - **Port:** Change the SCP or SFTP port number on the external server.
 - **Destination Public Key:** This field shows the public key information that Oracle Key Vault currently has stored in its `known_hosts` file for the remote server. If the remote server's public key changes, then Oracle Key Vault cannot copy backups to or from the remote server. In order to store the new public key in the `known_hosts` file, click the **Reset Dest Public Key** button, which will retrieve and save the new public key from the remote server. After you click **Reset Dest Public Key**, verify that the correct public key was saved.

- **Username:** Enter the user name of the user account that can be used to establish an SCP or SFTP connection to the remote server. Ensure that the new user has write permissions on the directory that is specified in **Destination Path**, because this path cannot be changed.
- **Authentication Method:** Choose one of the following:
 - **Password Authentication:** The password of the user account entered in the **Username** field.
 - **Key-based Authentication:** If the Oracle Key Vault public key has changed, re-copy the public key that appears in the **Public Key** field and then paste it in the appropriate configuration file, such as `authorized_keys`, on the destination server. Be aware that certain events may trigger a change of the public key, which means that Oracle Key Vault cannot use the backup destination until the new public key is re-copied from Oracle Key Vault to the appropriate configuration file. These events include but are not limited to certificate rotation, changing the IP address, and conversion to a cluster node.

7. Click **Save**.

Oracle Key Vault validates the input that you supplied to update the backup destination. If the validation fails, then the backup destination is not updated. If this happens, then check values for the user account on the remote server (user name and password) and ensure that the directory has write permissions for the user. Finally, ensure that the remote server is active.

Related Topics

- [Scheduling the Purging of Old Oracle Key Vault Backups](#)
To better manage disk space used by Oracle Key Vault backups on remote destinations, you can schedule the periodic purging of old backups from them.

Deleting a Remote Backup Destination

You can delete a remote backup destination (but not the local destination) to stop future backups to that destination server.

Deleting a remote backup destination from Oracle Key Vault does not remove the backups on the destination.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. Select **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, check the boxes for the backup destinations that you want to delete.
6. Click **Delete**.

Scheduled Backups and States

Oracle Key Vault provides scheduled backup types depending on the backup destination, and different states that indicate the progress of the backup activity.

- [About Schedule Backup Types and States](#)
You can schedule backups in Oracle Key Vault for specific times and backup destinations.

- [Types of Oracle Key Vault Backups](#)
Oracle Key Vault provides two types of backup jobs that can be scheduled: one-time backups and periodic backups.
- [Scheduled Backup States in Oracle Key Vault](#)
Scheduled backups have four states, which indicate whether the backup is scheduled, in progress, completed, or paused.

About Schedule Backup Types and States

You can schedule backups in Oracle Key Vault for specific times and backup destinations.

The backup process starts at the scheduled time and generates a system backup, which is a file that is stored on the backup destination. There is one backup file for each completed backup.

No backup can start if another backup is in progress. You can change the schedule of backups as needs change. You can continue working with Oracle Key Vault while the backup is in progress.

A system restart will terminate any ongoing backup. If you must restart the system during the time a backup is scheduled to occur, then you can pause the backup and resume the backup after the system restarts.

Types of Oracle Key Vault Backups

Oracle Key Vault provides two types of backup jobs that can be scheduled: one-time backups and periodic backups.

- **One-time backup:** A one-time backup makes a full backup of the Oracle Key Vault system. You can schedule multiple one-time backup jobs, each with its own start time.

You should make one-time local backups before making significant configuration changes to Oracle Key Vault, in case you need to recover from configuration failures.

`LOCAL` destinations can only store the last one-time backup. When a one-time backup to `LOCAL` completes, the previous backup is deleted.

- **Periodic backup:** After you schedule a periodic backup, Oracle Key Vault takes a full backup at the designated start time and then puts the backup schedule in the `ACTIVE` state. After the backup period passes, another backup starts. If it has been less than 7 days since the last full backup, then the next backup will be a cumulative incremental backup, which holds changes since the last full backup. If it has been more than 7 days since the last full backup, then the next backup will be a full backup.

For example, if the backup period is one day, then every seventh one is a full backup. If the backup period is 8 days, then all backups are full backups. If the backup period is 12 hours, then there are 13 cumulative backups before a full backup.

You should schedule periodic backups with a period of 1 day or less to minimize data loss.

A `LOCAL` destination can store only the last full backup and the cumulative incremental backups after that full backup. After the periodic backup schedule takes a new full backup to `LOCAL`, previous periodic full and cumulative backups in `LOCAL` are deleted.

Cumulative incremental backups are faster than full backups. Only one periodic backup can be scheduled at any time.

Related Topics

- [Scheduled Backup States in Oracle Key Vault](#)
Scheduled backups have four states, which indicate whether the backup is scheduled, in progress, completed, or paused.

Scheduled Backup States in Oracle Key Vault

Scheduled backups have four states, which indicate whether the backup is scheduled, in progress, completed, or paused.

- **ACTIVE:** The backup is scheduled and will start at the next start time. (The start time is indicated in the Start Time column on the Scheduled Backups page.)
- **PAUSED:** All future backups are on hold and will not start even if the start time has passed. They will start when they are explicitly resumed. You can change the state from active to paused and back. Put a scheduled backup in the paused state for these situations:
 - When communication between Oracle Key Vault and the remote destination is broken
 - If the remote destination is unavailable
 - If you want to defer the backup
- **ONGOING:** The backup is in progress.
- **DONE:** The backup is complete.

Scheduling and Managing Oracle Key Vault Backups

You can schedule Oracle Key Vault backups to specific backup destinations and times.

You must create the backup destinations that you will use beforehand, and you can modify or delete backup schedules.

- [Scheduling a Backup for Oracle Key Vault](#)
You can schedule a one-time or a periodic backup to a local or remote backup destination.
- [Changing a Backup Schedule for Oracle Key Vault](#)
You cannot change the schedule of a backup in progress.
- [Deleting a Backup Schedule from Oracle Key Vault](#)
You can delete a backup schedule from the Oracle Key Vault management console.
- [How Primary-Standby Affects Oracle Key Vault Backups](#)
In a primary-standby deployment, you must perform backups on the primary server.
- [How Using a Cluster Affects Oracle Key Vault Backups](#)
In a multi-master cluster environment, be aware of how the backup process works in individual nodes and the entire cluster.
- [Protecting the Backup Using the Recovery Passphrase](#)
Oracle Key Vault uses the recovery passphrase to control who can restore system backups.

Scheduling a Backup for Oracle Key Vault

You can schedule a one-time or a periodic backup to a local or remote backup destination.

You can start a one-time backup to start immediately without setting a time. However, do not schedule backup operations if a certificate rotation or cluster operation is in progress.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the Backup page, click **Backup**.

The Backup page appears. The following image shows how it appears when **PERIODIC** is selected.

5. In the **Name** field, enter a name for the backup.
6. If you want to perform a periodic backup, then do the following:
 - a. In the **Start Time** field, use the Calendar icon to specify the start time for the backup. If you want the first backup to perform immediately after you click **Schedule**, then select **Now**.
 - b. For **Type**, select **PERIODIC**.
The additional fields **Days**, **Hours**, and **Mins** appear.
 - c. In the **Days**, **Hours**, and **Mins** fields, specify the interval at which periodic backups will occur.
 - d. For **Destination**, select the backup destination.
 - e. Click **Schedule** to add the scheduled backup to the Scheduled Backup(s) page.

 **Note:**

A periodic backup is skipped if another backup is ongoing at the same time. The periodic backup will start at next scheduled interval after the ongoing backup completes.

7. If you want to perform a one-time backup, then do the following:
 - a. In the **Start Time** field, use the Calendar icon to specify the start time for the backup. If you want the backup to perform immediately after you click **Schedule**, then select **Now**.
 - b. For **Type**, select **ONE-TIME**.
 - c. For **Destination**, select the destination backup from the list.
 - d. Click **Schedule** to add the scheduled backup to the Scheduled Backup(s) page.

Changing a Backup Schedule for Oracle Key Vault

You cannot change the schedule of a backup in progress.

To change the backup schedule the state must be active or paused.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. Click the name of the scheduled backup in the Scheduled Backup(s) page.
5. Enter the **Start Time** or manually enter or click the calendar icon to choose the **Start Time** of the backup schedule.

You can only change the scheduled start time if it has not already passed. This means that the state cannot be ongoing or done. For a periodic backup you can change the start time if the scheduled start time has not passed.

6. If you are changing a periodic backup schedule, then in the **Days**, **Hours**, and **Mins** fields, specify the interval at which periodic backups will occur.
7. Select **Save**.

Deleting a Backup Schedule from Oracle Key Vault

You can delete a backup schedule from the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. Check the boxes of scheduled backups listed in the Scheduled Backup(s) page.
5. Click **Delete** to delete the selected backup schedules.

How Primary-Standby Affects Oracle Key Vault Backups

In a primary-standby deployment, you must perform backups on the primary server.

Because the standby synchronizes its state with the primary, you do not need to back up the standby.

Be aware of the following behavior for failover or switchover operations in a primary-standby deployment:

- Any backups in progress will terminate if there is a failover or a primary-standby switchover. Backups to `LOCAL` are private to the Oracle Key Vault server and therefore the local backup on the primary server is not available after a failover or switchover.
- Backups scheduled with password authentication start as usual after the failover or switchover.
- Because backups can only be restored to standalone servers, you must unpair primary-standby deployments before you can perform a backup restore operation on the former primary.

How Using a Cluster Affects Oracle Key Vault Backups

In a multi-master cluster environment, be aware of how the backup process works in individual nodes and the entire cluster.

- In an Oracle Key Vault multi-master cluster environment, the replication intrinsically creates copies of data on other nodes in the cluster. However, you can still perform backups and backup-related operations on all individual Oracle Key Vault cluster nodes.
- Backups in a cluster are taken for disaster recovery in case of a complete cluster failure and should all be done on remote destinations.
- Backups can still only be restored to standalone Oracle Key Vault servers. Because a cluster node cannot be switched back to a standalone server, only remote backups should be taken.

Protecting the Backup Using the Recovery Passphrase

Oracle Key Vault uses the recovery passphrase to control who can restore system backups.

To restore a backup, use the Oracle Key Vault recovery passphrase from the time when the backup was initiated. This is necessary even if the recovery passphrase was changed after the backup completed. Oracle recommends that you make a new backup every time the recovery passphrase is changed to ensure that there is always a copy of the backup that is protected by the most recent recovery passphrase.

Related Topics

- [Emergency System Recovery Process](#)
During installation, you will be required to create a special recovery passphrase that Oracle Key Vault uses to recover from emergency situations.

Restoring Oracle Key Vault Data

Oracle Key Vault data from a remote backup destination can be restored onto another Oracle Key Vault server.

This restore operation minimizes downtime and data loss.

- [About the Oracle Key Vault Restore Process](#)
The restore process replaces the database with the backup data.
- [Procedure for Restoring Oracle Key Vault Data](#)
You can restore Oracle Key Vault data to a standalone server using the Oracle Key Vault management console.
- [Multi-Master Cluster and the Restore Operation](#)
In a multi-master cluster deployment, you must consider several factors before you restore data to Oracle Key Vault.
- [Primary-Standby and the Restore Operation](#)
In a primary-standby deployment, you must consider several factors before you restore data to Oracle Key Vault.
- [Certificates and the Restore Operation](#)
A third-party certificate installed at the time of a backup will not be copied when you restore another server from this backup.

- [Changes Resulting from a System State Restore](#)
Restoring an Oracle Key Vault server brings the system state back to the time when the backup last performed.

About the Oracle Key Vault Restore Process

The restore process replaces the database with the backup data.

You must restore Oracle Key Vault data to a server only after ensuring that all scheduled backups on the server are completed.

Restoring data to an Oracle Key Vault server replaces the data in the server with that of the backup. The data restored is only as current as the backup. The restore operation replaces the Oracle Key Vault server with the backup. This means that some data can be lost. You might need to restore the endpoint database. Any data that is not in the backup that is getting restored will be lost. Backups can only be restored to the same version of Oracle Key Vault at which the backup was taken.

The maximum life of a backup is 1 year.

You must have the recovery passphrase that was in effect at the time of the backup in order to restore data from a backup. If you have not changed the recovery passphrase since installing Oracle Key Vault, then you must use the recovery passphrase that you created during the post-installation process.

Restoring data in Oracle Key Vault entails the following general steps:

1. Setting up the backup environment, which includes, after installing Oracle Key Vault, configuring backup destinations.
2. Performing the restore operation by determining the backup to use from a local or remote backup destination, and then providing the recovery passphrase to begin the restore process.

Related Topics

-

Procedure for Restoring Oracle Key Vault Data

You can restore Oracle Key Vault data to a standalone server using the Oracle Key Vault management console.

Before you restore, ensure that you have the correct recovery passphrase. You will need to enter this passphrase during the restore process. In addition, do not perform a restore operation while a certificate rotation process is in progress.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Restore**.

The Restore page appears.

Restore
Cancel
Restore

Source: REMOTE_BACKUP_DESTINATION ▼

Backup List

Go Actions ▼

Restore	Type	File	Backup Time	Version
<input type="radio"/>	Periodic	okvbackup_periodic_incr_20240314230811	14-MAR-2024 16:08:11	218.0.0.0
<input type="radio"/>	One-Time	okvbackup_onetime_onetime_20240314223010	14-MAR-2024 15:30:10	218.0.0.0
<input type="radio"/>	Periodic	okvbackup_periodic_full_20240314220917	14-MAR-2024 15:09:17	218.0.0.0

1 - 3 of 3

Last Restore Details

Restore Time	Destination Name	Run Error	Status
14-MAR-2024 16:31:27	REMOTE_BACKUP_DESTINATION	-	DONE

row(s) 1 - 1 of 1

5. Select **Source** from the drop-down list.
Values are either **LOCAL** or the names of configured remote destinations.
6. Select **Restore** next to the backup you want to restore from.
7. Click **Restore** to initiate the restore or recovery process.
You are prompted for the recovery passphrase.
8. Enter the recovery passphrase and then click **Restore** to begin.
The system will restore from the backup and then restart.
9. After the restore is complete and the system has restarted, delete any paused periodic backup jobs and then re-create them, using a new backup destination.
Oracle recommends that you delete such jobs in order to avoid corrupting the backup catalog file.
10. If your site uses the Commercial National Security Algorithm (CNSA) suite, then re-install these algorithms on the Oracle Key Vault server after the restore operation is complete.
11. If you had integrated Oracle Audit Vault with Oracle Key Vault before the restoration process, then do the following:
 - a. Log in to the Audit Vault Server console as an administrator and then delete the target and agent for Oracle Key Vault.
 - b. Log in to the Oracle Key Vault management console and then re-integrate Oracle Audit Vault with Oracle Key Vault.

Related Topics

- [Performing Backup Restore Operations with CNSA](#)
After you restore a backup of the Oracle Key Vault that was configured to use the enhanced Commercial National Security Algorithm (CNSA) Suite, use `/usr/local/okv/bin/okv_cnsa` to reconfigure CNSA compliance.
- [Integrating Oracle Audit Vault with Oracle Key Vault](#)
You can perform this integration from the Oracle Key Vault management console.

Multi-Master Cluster and the Restore Operation

In a multi-master cluster deployment, you must consider several factors before you restore data to Oracle Key Vault.

- You must restore only if all nodes in the cluster are lost.
- You must restore the backup to a standalone Oracle Key Vault server that has the same IP address as the node from which the backup was taken. Not doing so may affect the ability of endpoints to connect to the restored backup.
- After the restore operation, you must now use the restored server as the first node of a new cluster.

Primary-Standby and the Restore Operation

In a primary-standby deployment, you must consider several factors before you restore data to Oracle Key Vault.

- You must perform the restore operation only if both the primary and standby data are lost.
- You must restore the backup on a standalone Oracle Key Vault server only, even if the backup was taken from the primary.
- If you restore a backup taken from the primary node, you must use a freshly installed Oracle Key Vault server as the new standby.
- If the standby server has taken over as primary, and the former primary is lost, then there is no need to restore data from a backup to a new standby server. Just add a new standby server to the primary-standby deployment, which will automatically synchronize with the new primary.

Related Topics

- [Performing Backup Restore Operations with CNSA](#)
After you restore a backup of the Oracle Key Vault that was configured to use the enhanced Commercial National Security Algorithm (CNSA) Suite, use `/usr/local/okv/bin/okv_cnsa` to reconfigure CNSA compliance.

Certificates and the Restore Operation

A third-party certificate installed at the time of a backup will not be copied when you restore another server from this backup.

A third-party signed console certificate in use at the time of a backup is not part of the Oracle Key Vault backup. When you restore an Oracle Key Vault server from the backup, Oracle Key Vault does not restore the third-party signed console. You must reinstall the third-party console certificate on the newly restored Oracle Key Vault server.

The Oracle Key Vault service certificates (CA or server certificate) are included in the Oracle Key Vault backup. When you restore the backup, the newly restored Oracle Key Vault server will include the service certificates from the backup. However, these certificates are from the time when the backup was taken. The restored CA or the server certificate could have expired later.

You can also perform the CA certificate rotation after the backup is taken and hence the CA certificate in the backup may be from before the CA certificate rotation was done. Because the CA certificate rotation was already done, all the endpoints have been issued new endpoint certificates with the new CA certificate created after the backup was taken. After the restore,

the old CA certificate is in use and the endpoints with certificates issued using the new CA certificate will not be able to connect. When you restore a backup from before the CA certificate rotation, you must treat the old CA certificate as expired.

If the CA or the server certificate is nearing its expiration or you rotated the certificate after the backup that you restored, Oracle recommends that you rotate the CA and the server certificate right after the restore but before you proceed to set up a primary-standby or cluster deployment using the restored server.

If the CA or the server certificate is already expired, please contact Oracle Support.

As a best practice, backup the Oracle Key Vault server before and after performing a CA certificate rotation.

Related Topics

- [Managing Service Certificates](#)
This chapter explains about Oracle Key Vault-generated certificates. You will also learn how to manage self-signed and third-party certificates.
- [Finding the Expiration Date of the CA Certificate](#)
You can find how much time the Oracle Key Vault CA certificate has before it expires by navigating to the Service Certificates page.
- [Finding the Expiration Date of Server Certificates and Node Certificates](#)
You can find the expiration date of server certificates and node certificates in the Oracle Key Vault management console.
- [Managing CA Certificate Rotation](#)
You can use the Oracle Key Vault management console to rotate the CA certificate before the certificate gets expired. The new CA certificate can be a self-signed Root CA certificate or an intermediate CA certificate.
- [Deleting, Suspending, Reenrolling, or Rotating Endpoints](#)
When endpoints no longer use Oracle Key Vault to store security objects, you can delete them. You can also suspend, and later resume them when they are needed. You can also re-enroll or rotate endpoints when necessary.

Changes Resulting from a System State Restore

Restoring an Oracle Key Vault server brings the system state back to the time when the backup last performed.

Therefore, any changes that were made after the backup was made do not exist on the restored system. For example, if a user's password was changed after the backup operation, the new password will not be available in the restored system. The restored system will have the password that was in effect when the backup was made. As another example, the user account profile parameters values are restored to the parameter values that existed at the time the backup was taken.



Note:

Restoring also changes the recovery passphrase to the one that was in effect during the backup.

You should change the user passwords, enroll the endpoints created after backup, and make other similar changes, if required. You should confirm that everything is configured correctly after restoring.

If you are not certain that you restored the correct backup, then you can restore a different one, provided that Oracle Key Vault continues to remain a standalone server. To restore another backup, first configure the remote destination of this backup on the restored Oracle Key Vault itself, and then start the restore process. You do not need to reinstall the Oracle Key Vault appliance.

When the Oracle Key Vault server has been restored and is functional, you can continue to back up Oracle Key Vault data to new remote destinations.

Oracle recommends that you change user passwords after a restore operation and backup the Oracle Key Vault.

Scheduling the Purging of Old Oracle Key Vault Backups

To better manage disk space used by Oracle Key Vault backups on remote destinations, you can schedule the periodic purging of old backups from them.

- [About Scheduling the Purging of Old Oracle Key Vault Backups](#)
You can automatically purge old Oracle Key Vault backups from remote destinations.
- [Creating a Backup Destination Policy](#)
The Oracle Key Vault management console enables you to manage periodic purging of Oracle Key Vault backups from remote destinations.
- [Adding a Backup Destination Policy to a Remote Backup Destination](#)
After you have created a backup destination policy, you can associate it with one or more remote backup destinations.
- [Changing a Backup Destination Policy](#)
You can modify backup destination policies by using the Oracle Key Vault management console.
- [Suspending a Backup Destination Policy](#)
You can suspend a backup destination policy for a remote backup destination by using the Oracle Key Vault management console.
- [Resuming a Suspended Backup Destination Policy](#)
You can resume a suspended backup destination policy for a remote backup destination by using the Oracle Key Vault management console.
- [Deleting a Backup Destination Policy](#)
You can delete a backup destination policy by using the Oracle Key Vault management console.
- [Finding Information about Backup Destination Policies](#)
You can find information about backup destination policies on the Manage Backup Destinations page.

About Scheduling the Purging of Old Oracle Key Vault Backups

You can automatically purge old Oracle Key Vault backups from remote destinations.

Performing a regularly scheduled removal of old Oracle Key Vault backups helps to ensure that your backup destinations have sufficient room available to store new backups. You can perform this task from either the Oracle Key Vault management console or by using the Oracle Key Vault RESTful services commands. To purge backups from a remote destination

automatically at periodic intervals, you can create a backup destination policy and assign it to the remote destination. The backup destination policy defines the rules for selecting backups that should be purged. You can associate a backup destination policy with more than one remote destinations. You can suspend or resume a backup destination policy for the remote destinations individually. You must be a user who has the System Administrator role to create and assign a policy to a backup destination.

Creating a Backup Destination Policy

The Oracle Key Vault management console enables you to manage periodic purging of Oracle Key Vault backups from remote destinations.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, click **Manage Policies**.
6. In the Manage Backup Destination Policies page, click **Create**.

The Create Backup Destination Policy page appears.

Create Backup Destination Policy

Cancel Save

Name *

Recent Backups to Preserve * ?

Purge Backup After (in days) * ?

7. Enter the following values:
 - **Name:** Enter a name for the backup destination policy.
 - **Recent Backups to Preserve:** Enter the number of most recent backups that must always be preserved. Valid values are from 1 through 999. For example, configuring a value of 10 will ensure that Oracle Key Vault does not purge the most recent 10 backups regardless of their age. These backups will remain available for use.
 - **Purge Backup After (in days):** Enter the number of days after which a backup is to be purged. A backup that is eligible for purging will continue to remain available if it is among the number of most recent backups that are specified in the **Recent Backups to Reserve** field. Valid values are 1 through 999. For example, configuring a value of 30 will purge any backups that are older than 30 days unless they are among the required number of most recent backups to preserve.
8. Click **Save**.

After you create the policy, you can modify a remote backup destination to use this policy.

Related Topics

- [Adding a Backup Destination Policy to a Remote Backup Destination](#)
After you have created a backup destination policy, you can associate it with one or more remote backup destinations.

Adding a Backup Destination Policy to a Remote Backup Destination

After you have created a backup destination policy, you can associate it with one or more remote backup destinations.

After you have added a backup destination policy to a remote backup destination, every time a backup job runs on the destination, Oracle Key Vault removes the backups according to the policy.

 **Note:**

If a remote backup destination uses SCP as transfer method then the files associated with removed backups on the destination are replaced with zero bytes sized files. It is safe to delete these zero bytes sized files.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.
5. Click the **Edit** icon for the backup destination that you want to associate with a backup destination policy.
The Edit Backup Destination page appears.
6. From the **Policy** menu, select a backup destination policy.
7. Click **Save**.

Changing a Backup Destination Policy

You can modify backup destination policies by using the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, click **Manage Policies**.
Manage Backup Destination Policies lists existing policies.
6. Click the **Edit** icon for the backup destination policy that you want to change.
7. In the Edit Backup Destination Policy page, modify the following fields:
 - **Recent Backups to Preserve:** Enter the number of most recent backups that must always be preserved. Valid values are from 1 through 999. For example, configuring a value of 10 will ensure that Oracle Key Vault does not purge the most recent 10 backups regardless of their age. These backups will remain available for use.

- **Purge Backup After (in days):** Enter the number of days after which a backup is to be purged. A backup that is eligible for purging will continue to remain available if it is among the number of most recent backups that are specified in the **Recent Backups to Reserve** field. Valid values are 1 through 999. For example, configuring a value of 30 will purge any backups that are older than 30 days unless they are among the required number of most recent backups to preserve.

8. Click **Save**.

Suspending a Backup Destination Policy

You can suspend a backup destination policy for a remote backup destination by using the Oracle Key Vault management console.

Oracle Key Vault does not purge backups from a remote destination if the backup destination policy that is associated with the destination is suspended. You must suspend a backup destination policy for each associated remote destination individually.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, click the name of the destination you want to suspend a policy for.

The Backup Destination Policy area displays the associated policy.

6. In the Backup Destination Policy, click **Suspend**.

On success, State changes to Suspended.

Resuming a Suspended Backup Destination Policy

You can resume a suspended backup destination policy for a remote backup destination by using the Oracle Key Vault management console.

Oracle Key Vault resumes the purging of backups from a remote destination if the backup destination policy that is associated with the destination is resumed. You must resume a backup destination policy for each associated remote destination individually.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, click the name of the destination you want to suspend the policy for.

The Backup Destination Policy area displays the associated policy.

6. In the Backup Destination Policy area, click **Resume**.

On success, State changes to Active.

Deleting a Backup Destination Policy

You can delete a backup destination policy by using the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, click **Manage Policies**.
6. In the Manage Backup Destination Policies page, select the checkbox for the backup destination policy that you want to delete, and then click **Delete**.
7. In the confirmation dialog box, click **OK**.

Finding Information about Backup Destination Policies

You can find information about backup destination policies on the Manage Backup Destinations page.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.

The Manage Backup Destinations page lists currently configured backup destinations:

- Available backups on the destination
- The policy that is associated with the backup destination

5. In the Manage Backup Destinations page, click **Manage Policies**.

The Manage Backup Destination Policies page lists currently configured backup destination policies. Clicking on a policy name displays backups that have been purged by the policy. A status value of Purged indicates that a backup was successfully removed by the policy. A value of Unknown indicates there was a problem while deleting the backup. Examples of problems can be a backup not being available on the remote destination, or Oracle Key Vault not having permission to delete this remote backup.

- Current backup destination policies
- Purged backups

Manually Deleting a Local Oracle Key Vault Backup

You can manually delete a local backup by using the Oracle Key Vault management console.



Note:

If you delete a full periodic backup, then all of the incremental backups are also deleted.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the System Configuration area, click **Backup and Restore**.
4. In the System Backup page, click **Manage Backup Destinations**.
5. In the Backup Destinations area, click the **Edit** icon for Local destination.
The Edit Backup Destination page for Local destination lists backups that are available on the destination.
6. In the Available Backups table, select the check box for the backups that you want to delete, and then click **Delete**.
7. In the confirmation dialog box, click **OK**.

Configuring Oracle ZFS Storage Appliance to Store Oracle Key Vault Backups

Oracle ZFS Storage Appliance is an enterprise storage system that is designed for the storage of data from Oracle products such as Oracle Key Vault.

- [Step 1: Create a Storage Project in Oracle ZFS Storage Appliance](#)
The Oracle ZFS Storage Appliance can create immutable snapshots of Oracle Key Vault backups.
- [Step 2: Copy the Oracle Key Vault Public Key to the Oracle ZFS Storage Appliance](#)
The Oracle ZFS Storage Appliance storage project requires the backup key that is generated when you create an Oracle Key Vault backup.
- [Step 3: Complete Creating the Oracle ZFS Storage Appliance Project](#)
With the Oracle Key Vault backup key, you can complete the configuration for the Oracle ZFS Storage Appliance project.
- [Step 4: Configure Oracle Key Vault to Connect to the Oracle ZFS Storage Appliance Project](#)
The Oracle ZFS Storage Appliance storage project requires the backup key that is generated when you create an Oracle Key Vault backup.

Step 1: Create a Storage Project in Oracle ZFS Storage Appliance

The Oracle ZFS Storage Appliance can create immutable snapshots of Oracle Key Vault backups.

The Oracle ZFS Storage Appliance retention period feature works well with the Oracle Key Vault backup policy feature to provide a secure and space-efficient backup solution for Oracle Key Vault.

The steps to configure Oracle ZFS Storage Appliance as a backup destination for Oracle Key Vault involve creating a user in Oracle ZFS Storage Appliance that will allow Oracle Key Vault to log in (only using Secure File Transfer Protocol (SFTP), and optionally authenticated using public key authentication). Next, you must create a project, a file system, a schedule to create immutable snapshots, and define a retention period. Finally, in Oracle Key Vault, you must define Oracle ZFS Storage Appliance as the backup destination, and then create a backup schedule.

1. Log in to Oracle ZFS Storage Appliance as a user has privileges for creating backup projects.
2. Select the **Configuration** tab, then in the Configuration page, expand **Users** by clicking its plus (+) sign.
3. In the User Properties page, create a user who will be in charge of the Oracle Key Vault backup project. On the User Properties page, do the following:
 - a. From the **Type** menu, select **Local**.
 - b. In the **Username** and **Full Name** fields, enter the name of the user.
 - c. In the **Password** and **Confirm** fields, enter the user password.
 - d. Click **ADD**.
4. Configure SFTP.
 - a. In the Configuration page, expand **Services**.
 - b. Under Data Services, select **SFTP** to enable it. This page will expand so to the SFTP Properties page so that you can add details for the SFTP configuration
 - c. Under General Settings, confirm that the **Port (for incoming connections)** field is set to 218.
5. Do not exit this page; you will need to return to it in a later step to complete the backup destination creation.

Step 2: Copy the Oracle Key Vault Public Key to the Oracle ZFS Storage Appliance

The Oracle ZFS Storage Appliance storage project requires the backup key that is generated when you create an Oracle Key Vault backup.

1. Log in to Oracle Key Vault as a user who has the System Administrator role.
2. Select the **System** tab, and in the left navigation bar, select **Settings**.
3. Under System Configuration select **Backup and Restore**.
4. Select **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, click **Create**.

6. On the Create Backup Destination page, scroll to the **Public Key** field.
7. Copy the contents of the field, except for the line `ssh-rsa`.
8. Do not exit Oracle Key Vault; you need to return to it in a later step.

Step 3: Complete Creating the Oracle ZFS Storage Appliance Project

With the Oracle Key Vault backup key, you can complete the configuration for the Oracle ZFS Storage Appliance project.

1. In Oracle ZFS Storage Appliance, return to the Services page for SFTP.
2. Expand the **Keys** area, which appears after **Security Settings**.
3. In the **New Key** dialog box, do the following:
 - a. For **Cipher**, ensure that **RSA** is selected.
 - b. In the **Key** field, paste the public key from Oracle Key Vault.
 - c. Click **ADD**.
4. Click the **Shares** tab, and then click **Projects**.
5. Next to Projects, clicking the plus (+) sign.
6. In the Create Project dialog box, in the **Name** field enter a name for the project. Click **APPLY**.
7. Move the mouse over the project name and then click the pencil icon that appears on the right
The new project appears as a tab next to Projects.
8. Click the **General** tab.
9. Scroll to the bottom of the project page to the Default Settings area.
10. In the **User** field, enter the name of the user that you created when you began the configuration.
11. Configure the protocols for the Oracle Key Vault backup connection.
 - a. Click the **Protocols** tab.
 - b. Under SFTP, from the **Share mode** menu, select **Read/write**.
 - c. Click **APPLY**.
12. Configure the retention for the Oracle Key Vault configuration.
 - a. Click the **Snapshots** tab.
 - b. Under Properties, in the **Scheduled Snapshot Label** field, enter a name for the periodic backups on Oracle ZFS Storage Appliance.
 - c. Select the **Enable retention policy for Scheduled Snapshots** option to enable the retention.
 - d. Click **APPLY**.
13. Configure the frequency for the backup schedule.
 - a. In the Snapshots area, select the plus sign (+) next to **Schedules**.
 - b. Under **FREQUENCY**, set the frequency for the backup, such as **every day**.
 - c. After **scheduled time**, set the time of day for the backup to occur.

- d. Under **KEEP AT MOST**, select the number of backups in total that you want to keep.
 - e. Under **RETENTION**, select **Locked** to ensure that the backups will be immutable.
 - f. Click **APPLY**.
14. Create the file system for the Oracle Key Vault backup.
 - a. Select the **Shares** tab.
 - b. Next to **Filesystems**, click the plus sign (+).
 - c. In the Create Filesystem dialog box, do the following:
 - From the **Project** menu, select the name of the Oracle Key Vault project.
 - In the **Name** field, enter a name for the file system (for example, the same name that is used for the Oracle Key Vault project).
 - In the **User** field, enter the name of the user who is responsible for the Oracle Key Vault project .
 - Click **APPLY**.
 15. Optionally, exit Oracle ZFS Storage Appliance.

Step 4: Configure Oracle Key Vault to Connect to the Oracle ZFS Storage Appliance Project

The Oracle ZFS Storage Appliance storage project requires the backup key that is generated when you create an Oracle Key Vault backup.

1. Log in to Oracle Key Vault as a user who has the System Administrator role.
2. Select the **System** tab, and in the left navigation bar, select **Settings**.
3. Under System Configuration select **Backup and Restore**.
4. Select **Manage Backup Destinations**.
5. In the Manage Backup Destinations page, click **Create**.
6. On the Create Backup Destination page, do the following:
 - a. In the **Destination Name** field, enter a name for the backup destination.
 - b. For **Transfer Method**, select **sftp**.
 - c. For **Hostname**, enter the host information for the server where Oracle ZFS Storage Appliance resides, either the IP address or the name of the server.
 - d. For **Port**, ensure that it matches the port number that was specified in the Oracle ZFS Storage Appliance project that you created.
 - e. For **Destination Path**, enter `/export/` followed by the file system name that you gave in Oracle ZFS Storage Appliance when you configured the file system.
 - f. In the **User Name** field, enter the name of the user who was configured to manage the Oracle Key Vault backup project in Oracle ZFS Storage Appliance
 - g. Click **Save** to create the backup destination.
7. In the left navigation bar, select **System Backup** and then **Backup**.
8. In the Backup page, do the following:
 - a. In the **Name** field, enter a name for the backup.
 - b. For **Start Time**, use the calendar icon to specify a time for the backup to begin.

- c. For **Destination**, select **ZFS** from the menu.
- d. For **Type**, select **Periodic**, and then enter the days, hours, and minutes for the backup.
- e. Click **Schedule**.

The scheduled backup appears in the Scheduled Backups area of the System Backup page.

Backup and Restore Best Practices

Oracle provides best practices to keep backups current so that you can recover from catastrophic failures with minimum downtime and data loss.

- Ensure that the recovery passphrase at the time of backup is accessible because you will need it to restore data from a backup.
- Back up data any time you change the recovery passphrase.
- Ensure that you create at least one remote backup destination in a primary-standby deployment. Because the local backup resides on the Oracle Key Vault server itself, it will be lost in a failover or switchover situation.
- Do not edit or delete the backup catalog file that is associated with a remote backup destination, even if you stop using the backup destination. If you ever need to restore from a backup on this server, you will need the backup catalog file.
- If you use the same remote server for multiple backup destinations, then ensure that the directories are unique so that you have distinct backup catalog files associated with each backup destination. If you fail to do this, then the backup catalog file will be overwritten during subsequent backups and become unusable.
- When you restore a backup, do so to a standalone Oracle Key Vault server that has the same IP address as the Oracle Key Vault server on which the backup was taken. Failing to do so may result in endpoints not being able to connect to the restored backup.
- Before you restore data, ensure that all scheduled backups are complete.
- To create remote backup destinations successfully:
 - Ensure that the servers used as remote backup destinations are enabled and active.
 - Ensure that there is connectivity between Oracle Key Vault and remote server that you plan to use as a backup destination.
 - Ensure that the remote server designated as a backup destination supports the secure copy protocol (SCP) or the SSH file transfer protocol (SFTP).
 - Validate the user account credentials on the remote server before you create the backup destination on Oracle Key Vault.
 - Ensure that the destination directory has write permissions.
 - Create more than one remote backup destination on multiple servers for redundancy.
 - Ensure that the destination directories are unique if you are using the same remote server for multiple backup destinations. You must do this to prevent later backups from overwriting previous ones.
- Perform a one-time backup once every seven days.
- Schedule a periodic backup with a period of one day. This ensures that you have a full backup once in seven days.
- Perform a local one-time backup before system changes. You can use this backup as a restore point.

- Backup before and after upgrading Oracle Key Vault server software.
- Backup before and after performing critical operations such as rotating certificates.
- Change the backup destination after each upgrade. If at all possible do not reuse the backup destination.

Monitoring and Auditing Oracle Key Vault

Oracle Key Vault administrators can monitor and audit the Oracle Key Vault system, configure alerts and use reports.

- [Managing System Monitoring](#)
System monitoring refers to tasks such as configuring SNMP connections, email notifications, the syslog destination, and system diagnostics.
- [Configuring Oracle Key Vault Alerts](#)
You can select the type of alerts that you want to see in the Oracle Key Vault dashboard.
- [Managing System Auditing](#)
Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.
- [Using Oracle Key Vault Reports](#)
Oracle Key Vault collects statistical information on a range of activities that impact Key Vault operations.

Managing System Monitoring

System monitoring refers to tasks such as configuring SNMP connections, email notifications, the syslog destination, and system diagnostics.

- [Configuring Remote Monitoring to Use SNMP](#)
With Simple Network Management Protocol (SNMP) enabled, system administrators can remotely monitor the Oracle Key Vault appliance and its services.
- [Configuring Email Notification](#)
You can use email notifications to directly notify administrators of Key Vault status changes without logging into the Oracle Key Vault management console.
- [Configuring the Syslog Destination for Individual Multi-Master Cluster Nodes](#)
On each node, you can forward syslog entries to a remote service such as Splunk or SIEM.
- [Capturing System Diagnostics](#)
To troubleshoot problems that may arise, you can generate a diagnostics package.
- [Monitoring System Metrics](#)
You can use the System Metrics Monitoring feature to view and collect data for key system resource usage including CPU and Memory Usage in Oracle Key Vault.

Configuring Remote Monitoring to Use SNMP

With Simple Network Management Protocol (SNMP) enabled, system administrators can remotely monitor the Oracle Key Vault appliance and its services.

The collected data can be further processed and presented for the needs of the enterprise.

- [About Using SNMP for Oracle Key Vault](#)
You can use the Simple Network Management Protocol (SNMP) to monitor devices on a network for resource usage.
- [Granting SNMP Access to Users](#)
You can grant any user, including users who are not Oracle Key Vault administrators, access to SNMP data.
- [Changing the SNMP User Name and Password](#)
You can change the SNMP user name and password for a node at any time.
- [Changing SNMP Settings on the Standby Server](#)
You change the SNMP settings from the command line on the standby server.
- [Remotely Monitoring Oracle Key Vault Using SNMP](#)
SNMP enables you to monitor the vital components of Oracle Key Vault remotely without having to install new software in Oracle Key Vault.
- [SNMP Management Information Base Variables for Oracle Key Vault](#)
Oracle Key Vault provides a set of SNMP Management Information Base (MIB) variables that you can track.
- [Example: Simplified Remote Monitoring of Oracle Key Vault Using SNMP](#)
In Linux, you can simplify the SNMP commands you manually enter to find Oracle Key Vault information, yet still have useful and detailed output.
- [Configuring SNMP to Work with Old SNMP Clients](#)
Oracle Key Vault uses SHA256 for authentication protocol and AES256 for privacy protocol.

About Using SNMP for Oracle Key Vault

You can use the Simple Network Management Protocol (SNMP) to monitor devices on a network for resource usage.

Monitoring Oracle Key Vault is an important aspect how critical Oracle Key Vault's availability is when hundreds or thousands of Oracle and MySQL databases store their TDE master encryption keys in an Oracle Key Vault multi-master cluster. The types of resource usage that you should monitor include memory, CPU utilization, and processes. Even though Oracle Key Vault provides continuous key availability by allowing up to 16 (geographically distributed) instances to be connected to a single cluster, the health of each individual node contributes to the performance and availability of the entire cluster.

You can use Simple Network Management Protocol (SNMP) third-party tool to monitor remote systems that access Oracle Key Vault. The benefits of using SNMP to monitor Oracle Key Vault are as follows:

- There is no need to allow SSH access to Oracle Key Vault. (SSH access should only be enabled for the window of time in which it is being used.)
- You do not need to install additional tools to perform an SNMP monitoring operation.

Oracle Key Vault uses SNMP version 3 for user authentication and data encryption features. Unlike SNMP versions 1 and 2 that communicate in readable, insecure plaintext, SNMP 3 authenticates users and encrypts data on the communication channel between the monitoring server and the target. The information from Oracle Key Vault is unreadable to an intruder, even if the communication channel is intercepted.

In addition, with SNMP enabled on Oracle Key Vault, you can determine whether the key management server (KMIP daemon) is running. To track this information, you must use a third-party SNMP client to poll the Oracle Key Vault instance, because Oracle Key Vault does not provide SNMP client software.

Oracle Key Vault audits the creation and modification of SNMP credentials.

You must be a user with the System Administrator role to configure the SNMP account with a user name and password. These SNMP credentials are needed to access SNMP data.

In a multi-master cluster, the SNMP account with a user name and password can be set for all nodes of the cluster at once. It can also be set for each individual node.

 **Note:**

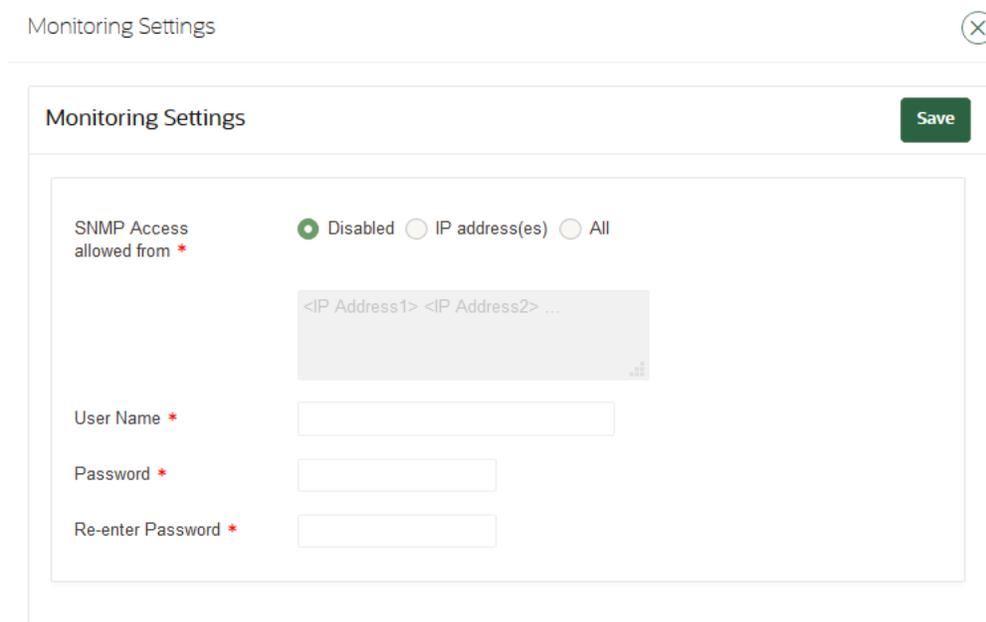
You must ensure that the SNMP username and password is *not* the same username and password as any of the Oracle Key Vault administrative user accounts with the System Administrator, Key Administrator, or Audit Manager role.

Granting SNMP Access to Users

You can grant any user, including users who are not Oracle Key Vault administrators, access to SNMP data.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **System** tab, and then select **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **SNMP**.

The Monitoring page appears.



4. In the Monitoring Settings page, enter the following information:
 - **SNMP Access allowed from:** Select **All** to enable a client at any IP address to poll Oracle Key Vault for information, **Disabled** to prevent any client, regardless of the client IP address, to poll Oracle Key Vault for information, or **IP Address(es)** if you want to restrict polling to clients with specific IP addresses. If you select **IP Address(es)**, then enter the IP addresses of the users you want to grant access to in

the IP Address field. Separate multiple IP addresses by a space. You cannot enter a range of IP addresses. You must list each IP address individually.

- **User Name:** Enter a name to associate with the SNMP configuration that will perform the monitoring.
- **Password and Re-enter Password:** Enter a secure password for this user that is at least 8 or more characters and contains at least one of each of the following: an uppercase letter, a lowercase letter, a number, and a special character from the set: period (.), comma (,), underscore (_), colon (:), forward slash (/). The SNMP password must **not** be the same as the password used to login into the Oracle Key Vault management console in any of the administrative roles.

5. Click **Save**.

Changing the SNMP User Name and Password

You can change the SNMP user name and password for a node at any time.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then select **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **SNMP**.
4. In the **User Name**, **Password**, and **Re-enter Password** fields, enter the user name and password information.
5. Click **Save**.

Changing SNMP Settings on the Standby Server

You change the SNMP settings from the command line on the standby server.

To add SNMP support in a primary-standby environment, you should configure SNMP on both the primary and standby servers before pairing them. This is because the standby server is no longer accessible from the Oracle Key Vault management console because all requests are forwarded to the primary server. However, you can change SNMP settings on the standby server in a primary-standby environment.

1. Log in to the standby server as the `support` user.
2. Switch to the root user.

```
su -
```

3. Go to the Oracle Key Vault bin directory.

```
cd /usr/local/okv/bin/
```

4. Run the `stdby_snmp_enable` script.

```
./stdby_snmp_enable parameter "options"
```

In this specification:

- `parameter` can be the following:
 - `-a`, which sets the SNMP access. It accepts the following `options`:
 - * `all` grants SNMP access.
 - * `disabled` disables SNMP access.

- * *IP_addresses* specifies one or more IP addresses to be granted SNMP access. Separate each IP address with a space.
- *-u* sets the user's SNMP name. This is the user name that was configured as the *snmpuser* when SNMP was enabled.
- *-p* sets the user's SNMP password. This password was created when for the *snmpuser* when SNMP was enabled.
- *options* is only used with the *-a* parameter.

The following examples show how to change SNMP settings on a standby server:

To grant SNMP access to all IP addresses and assign a user name *snmpuser* and password *password*:

```
./stdby_snmp_enable -a "all" -u "snmpuser" -p "password"
```

To disable SNMP access from all IP addresses:

```
./stdby_snmp_enable -a "disabled"
```

To grant SNMP access from certain IP addresses and assign user name *snmpuser* and the password *password*:

```
./stdby_snmp_enable -a "192.0.2.1 192.0.2.2 192.0.2.3" -u "snmpuser" -p "password"
```

 **Note:**

If SNMP is configured on a standby server then the configuration is removed during upgrade. You need to configure SNMP again after an upgrade. You can use the steps provided earlier in this section to configure SNMP after an upgrade.

Remotely Monitoring Oracle Key Vault Using SNMP

SNMP enables you to monitor the vital components of Oracle Key Vault remotely without having to install new software in Oracle Key Vault.

Though there are third-party tools that graphically display the information that SNMP extracts from Oracle Key Vault, the examples shown here are given with *snmpwalk* and *snmpget* from the command line on a remote computer that has a network connection into the SNMP account in Oracle Key Vault.

1. Log in to the remote host that will monitor Oracle Key Vault.
2. Confirm that the `UCD-SNMP-MIB` is installed on the remote host from which Oracle Key Vault is monitored.
3. Query the object ID for an Oracle Key Vault-supported SNMP Management Information Base (MIB) variable.

For example, suppose you wanted to track the number of processes running for the SNMP host. You can use a third-party SNMP client utility to query the status of the KMIP MIB whose object ID is `1.3.6.1.4.1.2021.2`, as follows:

```
third_party_snmp_client_command -v 3 OKV_IP_address -u SNMP_user -a SHA -A
SNMP_password -x AES -X SNMP_password -l authPriv iso.3.6.1.4.1.2021.2.1.2
```

The output is similar to the following:

```
iso.3.6.1.4.1.2021.2.1.2.1 = STRING: "mwecsvc"           <== Event collector
iso.3.6.1.4.1.2021.2.1.2.2 = STRING: "httpd"         <== httpd
iso.3.6.1.4.1.2021.2.1.2.3 = STRING: "kmipd"         <== KMIP daemon
iso.3.6.1.4.1.2021.2.1.2.4 = STRING: "ora_pmon_dbfwdb" <== embedded DB
iso.3.6.1.4.1.2021.2.1.2.5 = STRING: "ServiceManager" <== Golden Gate Service
Manager (Monitors other processes and reports status)
iso.3.6.1.4.1.2021.2.1.2.6 = STRING: "adminsrvr"     <== Golden Gate Admin
Server (Communicates with the DB to perform certain maintenance/admin tasks)
iso.3.6.1.4.1.2021.2.1.2.7 = STRING: "distsrvr"     <== Golden Gate
Distribution Server (Sends the OGG changes to other nodes)
iso.3.6.1.4.1.2021.2.1.2.8 = STRING: "recvsrvr"     <== Golden Gate Receiver
Server
```

Related Topics

- [SNMP Management Information Base Variables for Oracle Key Vault](#)
Oracle Key Vault provides a set of SNMP Management Information Base (MIB) variables that you can track.

SNMP Management Information Base Variables for Oracle Key Vault

Oracle Key Vault provides a set of SNMP Management Information Base (MIB) variables that you can track.

The following table lists the MIB variables that are supported.

Table 22-1 MIBs That SNMP Tracks for Oracle Key Vault

MIB Variable	Object ID	Description
sysDescr	1.3.6.1.2.1.1.1	Represents the product name and version identification of Oracle Key Vault.
sysUpTime	1.3.6.1.2.1.1.3	Represents the time (in hundredths of a second) since the network management portion of the system was last re-initialized.
sysName	1.3.6.1.2.1.1.5	Represents an administratively-assigned name. By convention, this is the node's fully-qualified domain name.
ifAdminStatus.x	1.3.6.1.2.1.2.2.1.7	Tracks if the Oracle Key Vault network interface (x) are running, not running, or being tested. Values are as follows: <ul style="list-style-type: none"> • 1: Instance is running • 2: Instance is down • 3: Instance is being tested
hrSystemUptime	1.3.6.1.2.1.25.1.1	Tracks the amount of time that an Oracle Key Vault instance has been running

Table 22-1 (Cont.) MIBs That SNMP Tracks for Oracle Key Vault

MIB Variable	Object ID	Description
laLoad	1.3.6.1.4.1.2021.10.1.3	Tracks the 1, 5 and 15 minute load averages (over 3 rows).
ssCpuRawUser	1.3.6.1.4.1.2021.11.50	For CPU monitoring; tracks the number of ticks (typically 1/100s) spent processing user-level code
ssCpuRawNice	1.3.6.1.4.1.2021.11.51	For CPU monitoring; tracks the number of ticks (typically 1/100s) spent processing reduced-priority code
ssCpuRawSystem	1.3.6.1.4.1.2021.11.52	For CPU monitoring; tracks the number of ticks (typically 1/100s) spent processing system-level code
ssCpuRawIdle	1.3.6.1.4.1.2021.11.53	For CPU monitoring; tracks the number of ticks (typically 1/100s) spent idle
ssCpuRawWait	1.3.6.1.4.1.2021.11.54	For CPU monitoring; tracks the number of ticks (typically 1/100s) spent waiting for input-output (IO)
ssCpuRawKernel	1.3.6.1.4.1.2021.11.55	For CPU monitoring; tracks the number of ticks (typically 1/100s) spent processing kernel-level code
ssCpuRawInterrupt	1.3.6.1.4.1.2021.11.56	For CPU monitoring; tracks the number of ticks (typically 1/100s) spent processing hardware interrupts
UCD-SNMP-MIB.prTable	1.3.6.1.4.1.2021.2	Tracks the number of processes running under a certain name. Names we monitor are <code>httpd</code> (the http server), <code>kmipd</code> (the kmip daemon), and <code>ora_pmon_dbfwdb</code> (an indicator if the DB is down)
memTotalSwap	1.3.6.1.4.1.2021.4.3	Tracks the total amount of swap space configured.
memAvailSwap	1.3.6.1.4.1.2021.4.4	Tracks the amount of swap space currently unused or available.
memTotalReal	1.3.6.1.4.1.2021.4.5	Tracks the total amount of RAM being used
memAvailReal	1.3.6.1.4.1.2021.4.6	Tracks the available RAM

Table 22-1 (Cont.) MIBs That SNMP Tracks for Oracle Key Vault

MIB Variable	Object ID	Description
nsExtendOutputFull	1.3.6.1.4.1.8072.1.3.2.3.1.2	<p>The nsExtendOutputFull MIB base variable shows the following information:</p> <ul style="list-style-type: none"> For Fast Recovery Area Space utilization; displays the size, used, and free space. CA and the server certificate expiration date and time, and server certificate SAN value. Also shows the platform certificate expiration date and time. <p>For the certificate expiration, the time zone that is shown for the date and time is in UTC. The output may be inconsistent if Oracle Key Vault is in the middle of a certification rotation.</p> <ul style="list-style-type: none"> Status of Oracle Audit Vault agent Status of Apache Tomcat Web server <p>Tomcat status can be:</p> <ul style="list-style-type: none"> Up Down Failed <p>The Tomcat status shows as Failed if the Tomcat is unresponsive due to Out of Memory (OOM) error.</p> <ul style="list-style-type: none"> CPU usage percentage

 **See Also:**

For more information refer to the Net-SNMP documentation at <http://www.net-snmp.org>

Example: Simplified Remote Monitoring of Oracle Key Vault Using SNMP

In Linux, you can simplify the SNMP commands you manually enter to find Oracle Key Vault information, yet still have useful and detailed output.

The configuration in this section assumes that you have granted SNMP access to a trusted user. It also assumes that you have installed the SNMP Management Information Base (MIB) variables on the remote host that will monitor Oracle Key Vault.

For example, a lengthy version of the `snmpwalk` command for an SNMP user named `snmp_admin` is as follows:

```
snmpwalk -v3 OKV_IP_address -n "" -l authPriv -u snmp_admin -a SHA-256
-A snmp_user_password -x AES-256 -X snmp_user_password
```

This command lists the vital services that are running on Oracle Key Vault. However, you can modify the command (and other SNMP commands) to be not only shorter, but to show additional information, such as whether the services are running or not running.

To simplify this type of command, you can edit the `/etc/snmp/snmp.conf` configuration file so that the SNMP commands you enter will automatically include commonly used settings, such as the default user or the default security level. The example in this topic omits password parameters so that users can enter the password at the command line interactively.

1. Log in to the remote host that will monitor Oracle Key Vault.
2. Edit the `/etc/snmp/snmp.conf`, which appears as follows:

```
# As the snmp packages come without MIB files due to license reasons,
# loading MIBs is disabled by default. If you added the MIBs you
# can reenale loading them by commenting out the following line.
mibs :
```

3. Comment out the `# mibs :` line and then add the following lines, as follows:

```
# loading MIBs is disabled by default. If you added the MIBs you
# can reenale loading them by commenting out the following line.
# mibs :
defSecurityName snmp_admin
defSecurityLevel authPriv defAuthType SHA-256
defPrivType AES-256
```

In this example:

- `defSecurityName`: Enter the name of the user to whom you granted SNMP access. This example uses `snmp_admin`.
- `defSecurityLevel`: Enter the default security level to use. This example uses `authPriv`, which enables communication with authentication and privacy.
- `defAuthType`: Enter the default authentication protocol. This example uses `SHA-256`.
- `defPrivType`: Enter the default privacy protocol. This example uses `AES-256`.

4. Restart `snmpd` to load the configuration file.

For example, for Linux 7 and later:

```
systemctl restart snmpd
```

For Linux 6:

```
service snmpd restart
```

5. To run the simplified version of the `snmpwalk` command that was shown earlier, enter the following command:

```
snmpwalk okv_ip_address prNames -A snmp_user_pwd -X snmp_user_pwd
```

In this command, `prNames` refers to "process names", which displays the names of processes instead of numbers. For example:

```
$ snmpwalk 192.0.2.254 prNames -A snmp_user_pwd -X snmp_user_pwd
UCD-SNMP-MIB::prNames.1 = STRING: httpd
UCD-SNMP-MIB::prNames.2 = STRING: kmipd
```

```
UCD-SNMP-MIB::prNames.3 = STRING: kmipusd
UCD-SNMP-MIB::prNames.4 = STRING: ora_pmon_dbfwdb
UCD-SNMP-MIB::prNames.5 = STRING: ServiceManager
UCD-SNMP-MIB::prNames.6 = STRING: adminsrvr
UCD-SNMP-MIB::prNames.7 = STRING: distsrvr
UCD-SNMP-MIB::prNames.8 = STRING: recvsrvr
UCD-SNMP-MIB::prNames.9 = STRING: av_agent_monitor
```

An example of running the `snmptable` command now becomes the following.

```
snmptable okv_ip_address prTable -A snmp_user_pwd -X snmp_user_pwd
```

Output similar to the following appears.

```
$ snmptable 192.168.1.181 -A Manager_1 -X Manager_1 prTable
SNMP table: UCD-SNMP-MIB::prTable

prIndex      prNames prMin prMax prCount prErrorFlag
prErrMessage prErrFix prErrFixCmd
1            httpd   1    20    8
noError
2            kmipd   1    2    2
noError
3            kmipusd 1    2    2
noError
4 ora_pmon_dbfwdb 1    1    1
noError
5 ServiceManager 1    1    1
noError
6 adminsrvr     1    1    1
noError
7 distsrvr     1    1    1
noError
8 recvsrvr     1    1    1
noError
9 av_agent_monitor 1    1    0    error No av_agent_monitor process
running noError
```

The next example shows how you would now run the `snmpdf` command:

```
snmpdf okv_ip_address -A snmp_user_pwd -X snmp_user_pwd
```

Output similar to the following appears.

Description	Size (kB)	Used	Available	Used%
/	20027260	7247856	12779404	36%
/usr/local/dbfw/tmp	6932408	15764	6916644	0%
/var/log	5932616	19932	5912684	0%
/tmp	1999184	3072	1996112	0%
/var/lib/oracle	143592160	35023900	108568260	24%

Configuring SNMP to Work with Old SNMP Clients

Oracle Key Vault uses SHA256 for authentication protocol and AES256 for privacy protocol.

To configure Oracle Key Vault to work with old SNMP clients that do not support SHA256 for authentication protocol and AES256 for privacy protocol, perform the following steps:

1. Log in to the Oracle Key Vault server as a support user through SSH.

```
ssh support@okv_instance_ip_address
```

2. Switch to the `root` user.

```
su - root
```

3. Edit the `/usr/local/dbfw/templates/template-snmpd.conf` file and modify the line with `createUser` directive to use SHA as the authentication protocol and AES as the privacy protocol.

- **Before modification**

```
createUser <%= @username %> SHA-256 "<%= @dapassword %>" AES256 "<%= @depassword %>"
```

- **After modification**

```
createUser <%= @username %> SHA "<%= @dapassword %>" AES "<%= @depassword %>"
```

4. If you have not configured SNMP, then configure SNMP by following the steps in [Granting SNMP Access to Users](#), else to reapply the SNMP configuration, follow the steps in [Changing the SNMP User Name and Password](#).

Configuring Email Notification

You can use email notifications to directly notify administrators of Key Vault status changes without logging into the Oracle Key Vault management console.

- [About Email Notification](#)
Email notifications alert users of status changes and are used to complete the processes of endpoint enrollment and user password reset operations.
- [Configuring Email Settings](#)
You can configure the Simple Mail Transfer Protocol (SMTP) server properties to receive email notifications from Oracle Key Vault.
- [Testing the Email Configuration](#)
Oracle Key Vault management console enables you to send test emails to test the email configuration.
- [Disabling Email Notifications for a User](#)
You can use the Oracle Key Vault management console to enable or disable email notifications.

About Email Notification

Email notifications alert users of status changes and are used to complete the processes of endpoint enrollment and user password reset operations.

To enable email notification you must set your email preferences in Oracle Key Vault. You can choose the events that you want updates to. The events include Oracle Key Vault system status like disk utilization, backup, and primary-standby, or user and endpoint status like expiration of user passwords, endpoint certificates, and keys, or cluster status like the heartbeat lag, naming conflicts, cluster-wide HSM status, and others.

In cluster deployments, you must configure and validate email settings on all nodes of the cluster. Email settings of a cluster node are local to that node.

Configuring Email Settings is driven by the SMTP provider. Once you confirm that the SMTP server is reachable from the Oracle Key Vault server. You need to follow the required setting from the SMTP provider.

You can modify the SMTP server configuration at any time. If a custom SMTP certificate was used initially, and you later decide to use the default, you can modify the trust store setting to default, instead of custom.

For example:

- The enrollment token generated during endpoint enrollment can be mailed directly to the endpoint administrator from Oracle Key Vault.
- An Oracle Key Vault system administrator can send the random temporary password directly to the user when the user password is reset.

To enable email notifications successfully, there must be a connection between Oracle Key Vault and the SMTP server.

You can disable email notifications at any time.

 **Note:**

If you are using Oracle Key Vault in an Oracle Cloud Infrastructure (OCI) environment, then see My Oracle Support note [2501601.1](#) for information about how to configure Postfix to use email delivery on the Oracle Linux 6 and 7 platforms. After you complete the configuration, ensure that you populate the `From Address` field with the approved sender from OCI.

Configuring Email Settings

You can configure the Simple Mail Transfer Protocol (SMTP) server properties to receive email notifications from Oracle Key Vault.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **System** tab, and then select **Settings** from the left navigation bar.
3. In the Network Services area, click **Email**.

The Email Settings page appears.

Email Settings



SMTP Server Address * ?

SMTP Port *

Name * ?

From Address *

Require Secure Connection

Require Credentials

Custom SMTP Server Certificate

Send Test Email

Email Address

4. In the Email Settings page, enter the following values:

- **SMTP Server Address:** Enter a valid SMTP server address or host name for the user account. This setting should match the SMTP server setting of the user's email account. Ensure that the SMTP server or hostname is reachable from Oracle Key Vault. If you enter the SMTP hostname, you must configure DNS from the **System Settings** menu, so the host name can be resolved.
- **SMTP Port:** Enter the SMTP port number of the outgoing SMTP server, usually 465. This port number can be another number, if expressly configured that way in your organization.
- **Name:** Enter an alias for the SMTP user that will appear in the From field of the email.
- **From Address:** Enter the email address that you want to provide as a sender.
- If the SMTP server requires a secure connection, select **Require Secure Connection**. If you are using anonymous relay on Microsoft Exchange Server, or an external SMTP server such as Gmail or Office 365, do not select **Require Secure Connection**. Ensure that your firewall rules allow forwarding of SMTP requests to an external SMTP server.

If **Require Secure Connection** is selected, the **Authentication Protocol** field is displayed with two options, **SSL** and **TLS**. Select the authentication protocol for the email server, either **SSL** or **TLS**. The default is **TLS**.
- If you have an SMTP user account, then check the box **Require Credentials**. When checked, the input fields **User Name**, **Password**, and **Re-enter Password** appear:
 - Enter the username of the SMTP user account.
 - Enter the password for the SMTP user account.

- Reenter the password for the SMTP user account.

Caution:

Oracle strongly recommends that you have a secure connection to the SMTP server, because auto-generated tokens are sent over email for operations such as the creation of administrative users and Oracle Key Vault system alerts.

Do not check **Require Credentials** for non-secure connections.

- If **Custom SMTP Server Certificate** is checked, then the field **Upload Certificate File** appears with the **Choose File** button to its right. Select this option if you want to upload a custom SMTP server's certificate to establish a TLS session between SMTP and Oracle Key Vault. This is how you can add a custom truststore in cases where the default Java truststore does not contain a necessary certificate. After Upload Certificate File, click **Browse** to upload a custom certificate file.
5. Click **Configure**.

On successful configuration, a `SMTP successfully configured` message is displayed.

If the configuration fails, then check that the SMTP server settings of the user email account are correct. Error messages highlight the field where the error has occurred to help isolate the problem.

Testing the Email Configuration

Oracle Key Vault management console enables you to send test emails to test the email configuration.

You can test the email configuration of the SMTP user account any time *after* you save the configuration. If you change an existing SMTP configuration, then you must save the configuration before you can test it.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select the **System** tab, and then select **Settings** from the left navigation bar.
3. In the Network Services area, click **Email**.
4. In the Email Settings window, configure the user's SMTP settings.
5. Save the configuration by clicking **Configure**.

You must save the configuration before you can test it.

6. In the **Send Test Email** section, enter the user email address in the **Email Address** field. Then click **Test**.

An email is sent to the user with `Oracle Key Vault: Test Message` in the subject line.

Depending on the Oracle Key Vault server timestamp, the email notification may not show up as the latest email.

The email notification may also not show up in your inbox, in which case you must check the spam folder.

If the email notification is not received, click the **Reports** tab and select **System Reports** from the left sidebar. On the **System Reports** page, click **Notification Report**. Check the list to determine the issue encountered while sending the email notification.

Related Topics

- [Configuring Email Settings](#)
You can configure the Simple Mail Transfer Protocol (SMTP) server properties to receive email notifications from Oracle Key Vault.

Disabling Email Notifications for a User

You can use the Oracle Key Vault management console to enable or disable email notifications.

An Oracle Key Vault user may elect not to receive email alerts. Only a user with the System Administrator role, or a user managing his own account can disable email notifications.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select the **Users** tab.
The Manage Users page appears.
3. Click user name of the user.
The User Details page appears.
4. Check the box to the left of text **Do not receive email alerts**.
5. Click **Save**.

Configuring the Syslog Destination for Individual Multi-Master Cluster Nodes

On each node, you can forward syslog entries to a remote service such as Splunk or SIEM.

- [Setting the Syslog Destination Setting for the Node](#)
You can set the syslog destination to use either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).
- [Clearing the Syslog Destination Setting for the Node](#)
You can clear the syslog destination setting for the node and then reset the node to the cluster setting.

Setting the Syslog Destination Setting for the Node

You can set the syslog destination to use either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **Syslog**.
4. In the **Syslog** window, select one of the following options:
 - **TCP**: Enables syslog using the TCP protocol.
 - **UDP**: Enables syslog using the UDP protocol.
5. Enter the syslog destination IP addresses and port numbers in the **Syslog Destinations** field, in the format *IP_address:port*.

You can enter multiple destinations, each separated by a space.

6. Click **Save**.

Clearing the Syslog Destination Setting for the Node

You can clear the syslog destination setting for the node and then reset the node to the cluster setting.

1. Log into the Oracle Key Vault management console for the node as a user who has the System Administrator role.
2. Select the **System** tab, and then **Settings** from the left navigation bar.
3. In the Monitoring and Alerts area, click **Syslog**.
4. In the Syslog window, click **Delete**.

Capturing System Diagnostics

To troubleshoot problems that may arise, you can generate a diagnostics package.

- [About Capturing System Diagnostics](#)
The Oracle Key Vault diagnostics file provides advanced debugging and troubleshooting information for problems that you may encounter while using Oracle Key Vault.
- [Configuring the Oracle Key Vault Application Tracing Level](#)
The System Administrator can configure the tracing level for the unified application tracing from the configure diagnostics page.
- [Downloading the Diagnostics Package](#)
The system administrator can download the diagnostic files using the Oracle Key Vault management console.
- [Unpacking the Diagnostics Package](#)
When the generated diagnostics bundle file size is smaller than the partition size, the diagnostics package is available in a singular .zip file. Otherwise, the diagnostics bundle is split into and is available in parts with the extension .zip-partXX.
- [Deleting Trace Files](#)
You can delete the old trace files after they are downloaded to free up the disk space.

About Capturing System Diagnostics

The Oracle Key Vault diagnostics file provides advanced debugging and troubleshooting information for problems that you may encounter while using Oracle Key Vault.

You can download diagnostics file and provide it to Oracle support for further analysis and debugging.

By default, diagnostics reporting is enabled on Oracle Key Vault. With the simplified diagnostics collection, system administrators are able to select which diagnostics components are to be packaged for the downloadable diagnostics bundle. Be aware that the first time you run the diagnostic utility or after the Oracle Key Vault system's internal database has been restarted, it can take longer to produce the bundle compared to subsequent runs because it must gather all the diagnostic information of the system.

If you plan to perform an upgrade of the Oracle Key Vault server, then disable the diagnostics packaging utility by ensuring that there are no files available to download. This can be confirmed by checking if the **Diagnostics** page has a section called **Diagnostics Package Files**. If it does, click **Clear** to disable the utility.

During upgrades, the current trace level for each component will reset to the *Mandatory* trace level.

Configuring the Oracle Key Vault Application Tracing Level

The System Administrator can configure the tracing level for the unified application tracing from the configure diagnostics page.

The steps describe how the system administrator can configure the tracing level from the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select **System**, and then **Settings**.
3. Click **Diagnostics** in the **Monitoring and Alerts** area.

The **Diagnostics Configuration** page appears.

4. Select the tracing option for **Application Trace Level** from the **Configurable Components** area. By default, the **Database Trace Level** and **System Trace Level** are always set to **MANDATORY**.

- **Trace Levels**

- The **Application Trace Level** enables the `LOG` option of the earlier release's diagnostics utility as well as adjusts the tracing level of the Oracle Key Vault application.

Note:

Setting the application trace level does not affect the trace level of the KMIP server. Contact Oracle support for instructions on how to enable tracing for the KMIP server.

- The **Database Trace Level** enables the `DATABASE` option of the earlier release's diagnostics utility.
- The **System Trace Level** enables the `SYSTEM`, `SOS_REPORT`, and `PLATFORM_COMMANDS` options of the earlier release's diagnostics utility.
- The **MANDATORY** level collects traces that are considered critical system conditions.
- The **ERROR** level collects traces in instances of errors and exceptions.
- The **WARNING** level collects traces in instances of warning conditions.
- The **INFO** level collects traces for general operational information.
- The **DEBUG** level collects all available traces.

 **Note:**

Adjusting the trace level only sets the **Configurable Components** for the specific node or server. To update the trace level on other cluster nodes or the standby server of a Primary-Standby deployment, repeat these steps on the other nodes or servers.

5. Click **Save**.

The **Oracle Key Vault tracing selections updated successfully** message ensures the settings are saved successfully.

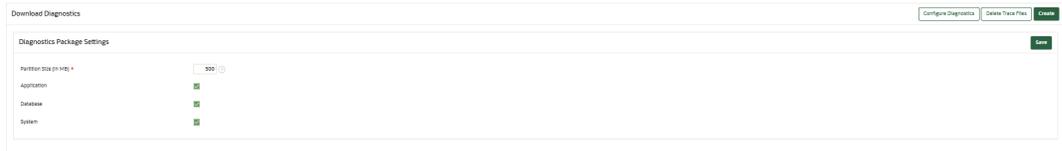
Downloading the Diagnostics Package

The system administrator can download the diagnostic files using the Oracle Key Vault management console.

The steps explain how you can customize the diagnostic package contents by only selecting the components of interest.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select **System**, and then **Diagnostics**.

The **Download Diagnostics** page appears.



3. Select the component for downloading the related diagnostics.

 **Note:**

The KMIP server tracing data is also included when you select the component **Application**.

4. You can also adjust the **Partition Size** based on your requirement. The default diagnostics package file partition size is 500 MB.
5. Click **Save** to save the download settings for future use.
6. Click **Create** to generate the bundle.
7. The **Diagnostics Package Files** pane appears with the list of files ready to be downloaded.

Diagnostics Package Files Clear Download

Diagnostics files will be downloaded in parts. ?

Q Go Actions ▾

Select File	File Path	Status
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part02	READY
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part03	READY
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part04	READY
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part05	READY
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part06	READY
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part07	READY
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part00	READY
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip-part01	READY

8. Select the files to download and click **Download**.

Note:

Ensure all parts are downloaded before recombining them for the full diagnostics bundle. To combine the file parts, see [Unpacking the Diagnostics Package](#).

Unpacking the Diagnostics Package

When the generated diagnostics bundle file size is smaller than the partition size, the diagnostics package is available in a singular `.zip` file. Otherwise, the diagnostics bundle is split into and is available in parts with the extension `.zip-partXX`.

1. Ensure you have followed steps from section Downloading the Diagnostics Package, see [Downloading the Diagnostics Package](#)
2. On the **Diagnostics Package** page, select the tracing file(s) to download and click **Download** from the **Diagnostics Package Files** pane.

Diagnostics Package Files Clear Download

Q Go Actions ▾

Select File	File Path	Status
<input type="checkbox"/>	/var/lib/oracle/okv_application_traces/okv_traces.zip	READY

1 - 1

3. Download and save the `.zip` file that contains the diagnostic reports to a secure location.

Note:

As each node is traced independently, in a cluster deployment mode, trace level is required to be set manually for each node.

If the diagnostics package file is with `zip-partXX` extension, recombine the files using the following steps:

- a. After the user has downloaded all the parts, run the following commands in a shell where the files have been downloaded to combine the parts into one zip file.

For UNIX Platforms:

```
$ cat okv_traces.zip-part* > okv_traces.zip
```

For Windows Platforms:

```
$ type okv_traces.zip-part* > okv_traces.zip
```

- b. unzip the file using the following command:

For UNIX Platforms:

```
$ unzip okv_traces.zip -d path_to_unpack_traces_to
```

For Windows Platforms:

```
$ Expand-Archive -LiteralPath okv_traces.zip -DestinationPath  
path_to_unpack_traces_to
```

- c. The application traces are available at `<download_location>/var/lib/oracle/okv_application_traces/` path while the database and system diagnostics are located at the same directory locations as in earlier Oracle Key Vault releases.

The application traces will have the following format:

```
VERSION | TIMESTAMP (YYYY-MM-DD HH:MM:SS.FF3 format) | HOSTNAME | TYPE  
| USER-CONTEXT IDENTIFIER | EXECUTION IDENTIFIER | TRACE SEVERITY LEVEL  
| COMPONENT | FILE NAME[LINE NUMBER] | MESSAGE
```

```
v1 | 2023-01-19 18:49:45.810 | okv02001703c6fe | PLSQL |  
DD9E6226-742F-43BB-A725 -0CF6C8A7EBC0 | SID:1334,SPID:3460,CPID:1234 |  
2 | GEN_SERVER | LDAP[3143] | Unable to reach the hostname when testing  
LDAP configuration
```

- **VERSION:** Indicates the version of the trace line format.
- **HOSTNAME:** Provides the host information (server or cluster) where the trace level are getting recorded.
- **TYPE:** Indicates the file type that executed the line of code for the trace statement.
- **USER-CONTEXT IDENTIFIER:** Indicates the user content identifier to identify the user who initiated the action. This can refer to an endpoint as well as a user.
- **EXECUTION IDENTIFIER:** Identifies the execution context of a particular sequence of trace statement will include different types of IDs.
- **TRACE SEVERITY LEVEL :** Identifies the trace level a specific message was printed. Possible trace levels are: 1, 2, 4, 8, 16.
- **COMPONENT:** Indicates the component been traced. Currently, the component available is GEN_SERVER.
- **FILE NAME[LINE NUMBER]:** Help in identifying the file name and line number for each trace statement.
- **MESSAGE:** Indicates the specific trace message that contains the information about the trace level.

- d. Decompress the application trace files using the `$ gzip -d okv_trace.trc.<date>.gz` command.
4. After downloading all parts of the diagnostics package, clear disk space in `/usr/local/dbfw/tmp` on the Oracle Key Vault system by performing the following steps:
Log in to the Oracle Key Vault management console as a user with the System Administrator role, and select **System, Diagnostics**. Scroll down to Diagnostics Package Files and click **Clear**.

 **Note:**

The **Diagnostics Package Files** pane and the **Clear** button are available in the Diagnostics page only if you had generated a diagnostics bundle using the steps in [Downloading the Diagnostics Package](#).

Related Topics

- [Configuring the Oracle Key Vault Application Tracing Level](#)
The System Administrator can configure the tracing level for the unified application tracing from the configure diagnostics page.

Deleting Trace Files

You can delete the old trace files after they are downloaded to free up the disk space.

1. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
2. Select **System**, and then **Diagnostics**. The **Download Diagnostics** page displays.
3. Click **Delete Trace Files**.

Review and confirm the prompt to proceed with the deletion.

 **Note:**

- If you have previously generated a diagnostics package, then, to clean up the diagnostics and trace files, select **System, Diagnostics** and click **Clear** in the **Diagnostics Package Files** section.
- The **Delete Trace Files** also deletes the older KMIP server trace data. The trace files that are currently in use are not deleted regardless of their size.

Monitoring System Metrics

You can use the System Metrics Monitoring feature to view and collect data for key system resource usage including CPU and Memory Usage in Oracle Key Vault.

- [About Capturing System Metrics](#)
The System Metrics Monitoring feature provide resource monitoring capabilities using the Oracle Key Vault management console.

- [Viewing System Metrics](#)
You can use the Oracle Key Vault management console to view and download the system monitoring data.

About Capturing System Metrics

The System Metrics Monitoring feature provide resource monitoring capabilities using the Oracle Key Vault management console.

Oracle Key Vault periodically collects CPU and Memory Usage, Disk I/O, Network and Application Metrics. You can view and collect system metrics data using the Oracle Key Vault management console. The Oracle Key Vault System Metrics Monitoring eliminates the need to first login to Oracle Key Vault server and then monitor the system manually.

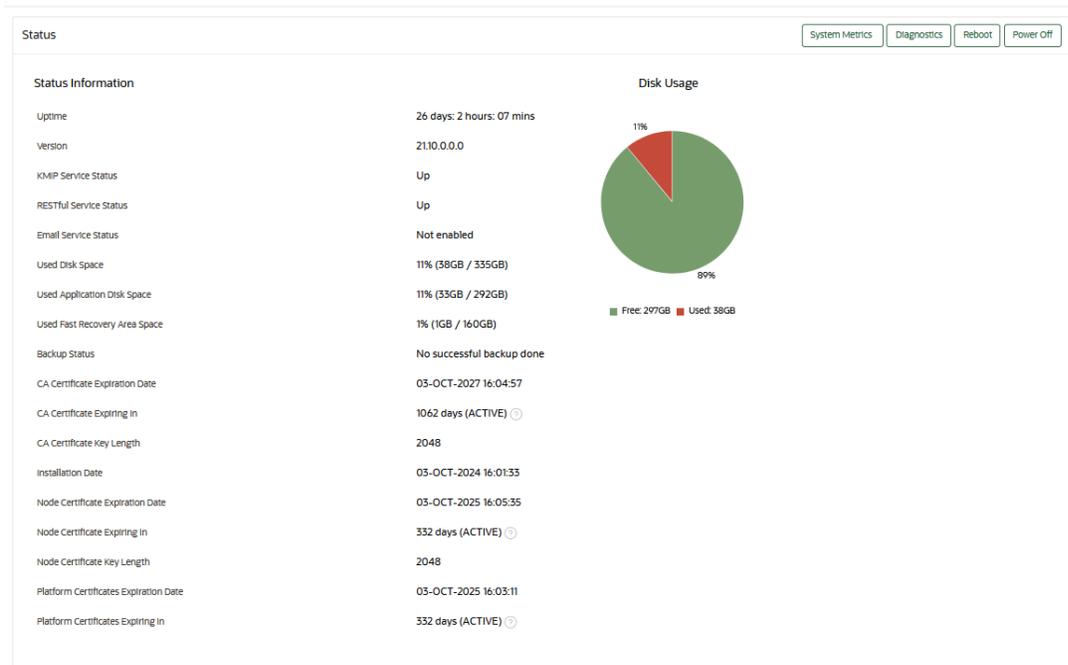
Viewing System Metrics

You can use the Oracle Key Vault management console to view and download the system monitoring data.

The instructions also explain how you can customize the output to collect required data.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Select **System**.

The Status page appears.



3. Click **System Metrics**.

System Metrics page shows the status of metrics collection service. **Up** means the service is running and collecting the data and **Down** means there's an issue and Oracle Key Vault is not able to collect the data.

System Metrics page appears with collapsed regions for each metrics category.



Expand the System Metrics region to display the graph for:

- a. CPU and Memory usage
- b. Disk I/O Metrics
- c. Network Metrics
- d. Application Metrics

Table 22-2 System Metrics

System Metrics	Description
CPU and Memory usage percentage	<p>Hovering mouse over a point in CPU Usage graph shows following:</p> <ul style="list-style-type: none"> • Time at which the data point collected • CPU usage percentage at the data point • Number of CPU cores at the data point • CPU load averages for last one, five and fifteen minutes at the data point <p>Hovering mouse over a point in Memory Usage graph shows following:</p> <ul style="list-style-type: none"> • Time at which the data point collected • Memory usage percentage at the data point • Total memory in GB at the data point • Free memory in GB at the data point • High Swap Daemon Activity <p>Note: When an Oracle Key Vault instance runs low on memory, the kernel memory swap daemon activates to swap the memory pages out, freeing up the memory. If the swap daemon exhibits high activity (indicated by significant CPU usage and ranking among the top three CPU-consuming processes) Oracle Key Vault considers this a high swap daemon activity which is an indication of high-memory pressure.</p>

Table 22-2 (Cont.) System Metrics

System Metrics	Description
Disk I/O Metrics	<p>Hovering mouse over a point in Disk Reads graph shows following:</p> <ul style="list-style-type: none"> • Time at which the data point collected • Total number of disk reads <p>Hovering mouse over a point in Disk Writes graph shows following:</p> <ul style="list-style-type: none"> • Time at which the disk write data collected • Total number of disk writes
Network Metrics	<p>Hovering mouse over a point in Data Received graph shows following:</p> <ul style="list-style-type: none"> • Time at which the data point collected • Collected data in bytes • Average rate at which data received <p>Hovering mouse over a point in Data Sent graph shows following:</p> <ul style="list-style-type: none"> • Time at which the data point collected • Collected data in bytes • Average rate at which data sent
Incoming TCP Connections	<p>Hovering mouse over a point in Incoming TCP Connections graph shows following:</p> <ul style="list-style-type: none"> • Time at which the data point collected • Number of incoming TCP connections
Application Metrics	<p>Hovering mouse over a point in KMIP Connections graph shows following:</p> <ul style="list-style-type: none"> • Time at which the data point collected • Total number of KMIP connections • Number of KMIP connections on non-RESTful interface • Number of KMIP connections on RESTful interface

4. Select the different options from the **Period** drop-down to view percentage usage in **Last 1 Hour, Last 24 Hour, Last Week, Last Month, or Date Range(Period)**.

 **Note:**

You can use the **Date Range (Period)** option to view and collect the usage data by specifying the **From** and **To** dates.

5. Select the different options from the **Interval** drop-down to aggregate the displayed data. You can select the **Auto** option for optimized performance.
6. Select an aggregate function from the **Statistic** drop-down. You can select from **Mean, Min, Max** or **Count** value.

 **Note:**

The **Count** statistic is not applicable to all the metrics.

7. Click **Refresh** to refresh and display the data according to the specified fields.

8. Click **Download** to save the data in a .csv file.

 **Note:**

- Clicking on **Download** button save the raw data for the period specified. Also, the downloaded data do not have interval and statistic filters applied on it.
- You might not be able to view the data or make any changes to the System metrics when the **Metrics Service** is **Down**.

Configuring Oracle Key Vault Alerts

You can select the type of alerts that you want to see in the Oracle Key Vault dashboard.

- [About Configuring Alerts](#)
System administrators can configure alerts from the Oracle Key Vault dashboard, but all users can see alerts for the security objects to which they have access.
- [Configuring Alerts](#)
You can configure alerts in the Reports page of the Oracle Key Vault management console.
- [Viewing Open Alerts](#)
Users can view alerts depending on their privileges.

About Configuring Alerts

System administrators can configure alerts from the Oracle Key Vault dashboard, but all users can see alerts for the security objects to which they have access.

Email notifications must be enabled for users to receive alerts.

The Oracle Key Vault dashboard is the first page you see on logging into the management console. You can navigate to this page by clicking the **Home** tab. All users can see the alerts on security objects they have access to, but only users with the System Administrator role can configure alerts.

Oracle Key Vault offers several types of alerts that you can configure with appropriate thresholds according to your requirements. The alert types that appear are based on the type of environment that you are using: standalone, primary-standby, or multi-master cluster. You can also configure alerts for an HSM-enabled Oracle Key Vault server.

Oracle Key Vault alerts are categorized to one of the severity levels: CRITICAL, HIGH, MEDIUM, and LOW. You should resolve the higher severity alerts first.

An alert configuration consists of whether an alert is enabled or disabled and the threshold limit. In a multi-master cluster, by default, same alert configuration is applied to all of the nodes in a cluster. For following alerts, alert configuration can be on either Cluster or Node scope.

- Fast Recovery Area Space Utilization
- High CPU Usage
- Failed System Backup

- High Memory Usage
- Disk Utilization
- System Backup

You can configure the following alerts, which are listed in ascending order:

Table 22-3 Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
Certificate Object Expiration	HIGH	Standalone, primary-standby, and multi-master cluster environments	Cluster-wide	<p>Raised when a certificate object's deactivation date is within the threshold value (default 7 days).</p> <p>This alert is raised only if the certificate object is in the <code>PRE-ACTIVE</code> or <code>ACTIVE</code> state.</p> <p>When an expiration alert is within its threshold, an email notification is sent once every 24 hours until the certificate object expires.</p>	Deleted if the certificate object is no longer expiring within the threshold value as a result of changes to either object's deactivation date or the configured threshold value. This alert is also deleted when a certificate object is revoked or destroyed.
Cluster FIPS Not Consistent	MEDIUM	Multi-master cluster only	Cluster-wide	Raised when at least one, but not all, <code>ACTIVE</code> nodes in the cluster are in FIPS mode	Deleted when all cluster nodes are in FIPS mode or all nodes are not in FIPS mode

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
Cluster Heartbeat Lag	HIGH	Multi-master cluster only	Node specific	Raised when a node has not received a heartbeat from another ACTIVE node in the cluster for over the threshold value (default 5 minutes)	Deleted when a node has once again received a heartbeat from the other node in the configured threshold period, as long as the node had received a heartbeat from the other node within the last Maximum Disable Node Duration period of time. This alert is also deleted when a node involved has been deleted from the cluster.
Cluster HSM Not Consistent	MEDIUM	Multi-master cluster only	Cluster-wide	Raised when at least one, but not all, ACTIVE nodes in the cluster are HSM-enabled	Deleted when all nodes are HSM-enabled or all nodes are not HSM-enabled

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
Cluster Naming Conflict	LOW	Multi-master cluster only	Cluster-wide	Raised when a naming conflict has been automatically resolved by Oracle Key Vault	Deleted when the object is deleted or renamed, or has had the new name explicitly accepted
Cluster Redo Shipping Status	HIGH	Multi-master cluster only	Node specific	Raised when a read/write node is unable to ship redo to its read/write peer, and as a result, is in read-only restricted mode. In addition to redo-shipping inactive status information, the alert indicates that the node in the cluster is operating in read-only mode.	Deleted when a read/write node is once again able to ship redo to its read/write peer, or when the node is deleted. The email notification states that the redo-shipping status is back up and the node in the cluster is no longer operating in read-only mode.
Cluster Replication Lag	HIGH	Multi-master cluster only	Node specific	Raised when incoming replication lag is greater than the threshold value (default 60 seconds)	Deleted when replication lag falls below the threshold value, or when any node in the replication link is deleted.

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
Disk Utilization	MEDIUM	Standalone, primary-standby, and multi-master cluster environments	Node specific	Raised when the free disk space percentage of the <code>/var/lib/oracle</code> partition is lower than the threshold value (default 25 percent)	Deleted when free disk space is once again higher than the threshold
Endpoint Certificate Expiration	HIGH	Standalone, primary-standby, and multi-master cluster environments	Cluster-wide	Raised when an endpoint's certificate is expiring within the threshold value (default 30 days). When an expiration alert is within its threshold, an email notification is sent once every 24 hours until the endpoint's certificate expires.	Deleted when the endpoint's certificate is no longer expiring within the threshold value or when the endpoint is deleted
Failed System Backup	MEDIUM	Standalone, primary-standby, and multi-master cluster environments	Node specific	Raised when the last backup did not complete successfully	Deleted when the most recent backup completed successfully
Failed OKV Services	CRITICAL	Standalone, primary-standby, and multi-master cluster environments	Node Specific	Raised when DB, KMIP, REST, Email, Cluster, or Audit Vault service stops because of a failure.	Deleted when all of the failed services start running successfully.

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
Fast Recovery Area Space Utilization	HIGH	Standalone, primary-standby, and multi-master cluster environments	Node specific	<p>Raised when Fast Recovery Area Space utilization of Oracle Key Vault's embedded database exceeds the configured threshold value (default 70 percent).</p> <p>To remedy this problem, try the following:</p> <p>Reduce the Maximum Disable Node Duration setting of the cluster node. Minimize the duration when peer node is not available. Consider deleting the node from the cluster and adding it back later.</p>	Deleted when Fast Recovery Area Space utilization of Oracle Key Vault's embedded database is once again within the configured threshold
High CPU Usage	HIGH	Standalone, primary-standby, and multi-master cluster environments	Node Specific	<p>Raised when average memory usage is greater than the threshold in last 5 minutes. Default value of threshold is 99%.</p> <p>Setting a threshold above 90% takes memory swapping into consideration along with memory usage for raising an alert.</p>	Deleted when 24 hours are passed since the first alert is raised and the CPU utilization is less than the threshold in last 5 minutes.

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
High Memory Usage	HIGH	Standalone, primary-standby, and multi-master cluster environments	Node Specific	Raised when average memory usage is greater than the threshold in last 5 minutes. Default value of threshold is 99%.	Deleted when 24 hours are passed since the first alert is raised and the memory usage is less than the threshold in the last 5 minutes.
Invalid HSM Configuration	CRITICAL	Standalone, primary-standby, and multi-master cluster environments	Node specific	Raised when there is an error in the HSM configuration (checked by default every 5 minutes)	Deleted when there is no longer an error in the HSM configuration
Key Rotations	HIGH	Standalone, primary-standby, and multi-master cluster environments	Cluster-wide	Raised when a key's deactivation date is within the threshold value (default 7 days) This alert is raised only if the key object is in the PRE-ACTIVE or ACTIVE state. When an expiration alert is within its threshold, an email notification is sent once every 24 hours until the certificate object expires.	Deleted if the key object is no longer expiring within the threshold value as a result of changes to either object's deactivation date or the configured threshold value. This alert is also deleted when a key object is revoked or destroyed.

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
OKV CA Certificate Expiration	CRITICAL	Standalone, primary-standby, and multi-master cluster environments	Node specific	Raised when the Oracle Key Vault CA certificate is expiring within the threshold value (default 90 days). When an expiration alert is within its threshold, an email notification is sent once every 24 hours until the CA certificate expires. Be aware that if the CA certificate expires, then endpoints will no longer be able to communicate with Oracle Key Vault. If an intermediate CA certificate is being used, the expiry date is determined from the earliest expiring certificate in the trust chain and not the expiry date of the leaf certificate.	Deleted when the CA certificate is no longer expiring within the threshold value

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
OKV Server/Node Certificate Expiration	CRITICAL	Standalone, primary-standby, and multi-master cluster environments	Node specific	<ul style="list-style-type: none"> • Raised in a standalone or primary-standby deployment when the Oracle Key Vault server certificate is expiring within the threshold value (default 90 days) • Raised in a multi-master cluster environment when a node's node certificate is expiring within the threshold value (default 90 days). • Be aware that if the server certificate expires in a standalone or primary-standby deployment, then endpoints will no longer be able to communicate with the Oracle Key Vault server. This will result in downtime for all endpoints. • If a node certificate expires in a multi-master cluster environment, endpoints will be able to use 	Deleted when the server or node certificate is no longer expiring within the threshold value.

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
				other nodes for endpoint operations (like fetching a key). However, the node will no longer be able to communicate with other multi-master cluster nodes, and operations like creating a new wallet will be impacted	

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
OKV Platform Certificates Expiration	CRITICAL	Standalone, primary-standby, and multi-master cluster environments	Node specific	<p>Raised when the Oracle Key Vault platform certificates are expiring within the threshold value (default 90 days). When an expiration alert is within its threshold, an email notification is sent once every 24 hours. The platform certificates are used when a new node is added to the cluster. The node whose platform certificates have expired cannot add any new node. Platform certificates are also used when shipping redo between read/write nodes of a cluster. The expired platform certificates may cause the redo shipping to fail resulting in read/write nodes of a cluster to go in read-only restricted mode.</p> <p>Note: Although expired platform certificates may impact communication between a node and its read/write peer, they do not impact endpoint communications with Oracle Key Vault. You cannot upgrade an Oracle</p>	Deleted when the platform certificates are no longer expiring within the threshold value.

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
				Key Vault node with the expired platform certificates.	
OKV Server Certificate Expiration		Standalone, primary-standby, and multi-master cluster environments	Node specific	<p>Raised when the Oracle Key Vault server certificate is expiring within the threshold value (default 90 days). When an expiration alert is within its threshold, an email notification is sent once every 24 hours until the certificate expires. Be aware that if the server certificate expires, then endpoints will no longer be able to communicate with the Oracle Key Vault server. This will result in downtime.</p>	Deleted when the server certificate is no longer expiring within the threshold value
Primary-standby Data Guard Broker Status	HIGH	Primary-standby only	-	Raised when the Oracle Data Guard Broker status is not ENABLED	Deleted when the broker status is once again ENABLED or when Oracle Key Vault is no longer in primary-standby mode

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
Primary - Standby Data Guard Fast-Start Failover Status	MEDIUM	Primary-standby only	-	Raised when the fast-start failover status is not SYNCHRONIZED	Deleted when the fast-start failover status is once again SYNCHRONIZED or when Oracle Key Vault is no longer in a primary-standby configuration
Primary - Standby Destination Failure	HIGH	Primary-standby only	-	Raised when the switchover status is FAILED DESTINATION	Deleted when the switchover status is no longer FAILED DESTINATION or when Oracle Key Vault is no longer in a primary-standby configuration
Primary - Standby Restricted Mode	HIGH	Primary-standby only	-	Raised when in primary-standby environment and the primary is running in read-only restricted mode	Deleted when the primary is no longer in read-only restricted mode or when Oracle Key Vault is no longer in a primary-standby configuration

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
Primary - Standby Role Change	LOW	Primary-standby only	-	Raised when there is a role change	Deleted when Oracle Key Vault is no longer in a primary-standby configuration
Secret Object Expiration	HIGH	Standalone, primary-standby, and multi-master cluster environments	Cluster-wide	Raised when a secret object's deactivation date is within the threshold value (default 7 days). This alert is raised only if the object is in the PRE-ACTIVE or ACTIVE state. When an expiration alert is within its threshold, an email notification is sent once every 24 hours until the secret object expires.	Deleted if the secret object is no longer expiring within the threshold value as a result of changes to either object's deactivation date or the configured threshold value. This alert is also deleted when a secret object is revoked or destroyed.
SSH Tunnel Failure	HIGH	Standalone, primary-standby, and multi-master cluster environments	Node specific	Raised when an SSH tunnel is not available	Deleted when the SSH tunnel is once again available or when the SSH tunnel is deleted

Table 22-3 (Cont.) Available Alerts

Alert Type	Severity	Environment	Multi-master Cluster Applicability	Purpose of Alert	When Alert Is Deleted
System Backup	MEDIUM	Standalone, primary-standby, and multi-master cluster environments	Node specific	Raised when the last successful backup is over the threshold value (default 14 days)	Deleted when the last successful backup was within the threshold value
System User Password Expiration	MEDIUM	Standalone, primary-standby, and multi-master cluster environments	Node specific	Raised when either the <code>root</code> or <code>support</code> (or both) users' passwords are expired or will expire within the threshold value (default 14 days).	Deleted when a system user's password no longer expires within the threshold value.
User Password Expiration	MEDIUM	Standalone, primary-standby, and multi-master cluster environments	Cluster-wide	Raised when a user's password expires within the threshold value (default 14 days). When an expiration alert is within its threshold, an email notification is sent once every 24 hours until the password expires. If the user password expires, user cannot login and Administrative tasks cannot be performed.	Deleted when a user's password no longer expires within the threshold value, or when the user is deleted

Related Topics

- [Configuring the Maximum Disable Node Duration for the Cluster](#)
You can set the maximum disable node duration time for the cluster in hours.
- [Managing Service Certificates](#)
This chapter explains about Oracle Key Vault-generated certificates You will also learn how to manage self-signed and third-party certificates.
- [Oracle Key Vault Root of Trust HSM Configuration Guide](#)

Configuring Alerts

You can configure alerts in the Reports page of the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Access the Alerts page by using one of the following methods:
 - Select the **System** tab, then **Settings**, and in the Monitoring and Alerts area, click **Alerts**.
 - Select the **Reports** tab, and then select **Alert** from the left navigation bar. In the Alerts page, click **Configure**.
 - On the Home page, expand **Alerts** at the top of the page, and then click **All Alerts**. Then click **Configure**.

The **Configure Alerts** page appears, listing various alert types and for some, configurable threshold limit and unit (such as the days until expiration for key rotations alert type). If you are using a multi-master cluster, then the Configure Alerts page will provide cluster-specific alerts, such as the cluster heartbeat lag, redo shipping status, or whether naming conflicts resolution is enabled. The following image shows how the Configure Alerts page appears in a multi-master cluster environment.

 **Note:**

Alerts are assigned severity based on their impact on the Oracle Key Vault and registered endpoints.

Configure Alerts						Alerts	Save
Alert Type	Enabled	Scope	Limit	Description	Severity		
Failed OKV Services	<input checked="" type="checkbox"/>	Cluster		-	CRITICAL		
Invalid HSM Configuration	<input checked="" type="checkbox"/>	Cluster	5	Minutes between HSM configuration verifications	CRITICAL		
OKV CA Certificate Expiration	<input checked="" type="checkbox"/>	Cluster	90	Days until expiration	CRITICAL		
OKV Platform Certificates Expiration	<input checked="" type="checkbox"/>	Cluster	90	Days until expiration	CRITICAL		
OKV Server/Node Certificate Expiration	<input checked="" type="checkbox"/>	Cluster	60	Days until expiration	CRITICAL		
Cluster Heartbeat Lag	<input checked="" type="checkbox"/>	Cluster	5	Minutes since the last heartbeat	HIGH		
Cluster Redo Shipping Status	<input checked="" type="checkbox"/>	Cluster		-	HIGH		
Fast Recovery Area Space Utilization	<input checked="" type="checkbox"/>	Cluster v	70	Percent space used	HIGH		
High CPU Usage	<input checked="" type="checkbox"/>	Cluster v	70	Percent CPU usage	HIGH		
SSH Tunnel Failure	<input checked="" type="checkbox"/>	Cluster		-	HIGH		
Certificate Object Expiration	<input checked="" type="checkbox"/>	Cluster	7	Days until expiration	MEDIUM		
Cluster Replication Lag	<input checked="" type="checkbox"/>	Cluster	60	Seconds of replication lag	MEDIUM		
Endpoint Certificate Expiration	<input checked="" type="checkbox"/>	Cluster	60	Days until expiration	MEDIUM		
Failed System Backup	<input checked="" type="checkbox"/>	Cluster v		-	MEDIUM		
High Memory Usage	<input checked="" type="checkbox"/>	Cluster v	99	Percent memory usage	MEDIUM		
Key Rotations	<input checked="" type="checkbox"/>	Cluster	7	Days until expiration	MEDIUM		
SSH Key Expiration	<input checked="" type="checkbox"/>	Cluster	7	Days until expiration	MEDIUM		
Secret Object Expiration	<input checked="" type="checkbox"/>	Cluster	7	Days until expiration	MEDIUM		
System User Password Expiration	<input checked="" type="checkbox"/>	Cluster v	14	Days until expiration	MEDIUM		
User Password Expiration	<input checked="" type="checkbox"/>	Cluster	14	Days until expiration	MEDIUM		
Cluster FIPS Not Consistent	<input checked="" type="checkbox"/>	Cluster		-	LOW		
Cluster HSM Not Consistent	<input checked="" type="checkbox"/>	Cluster		-	LOW		
Cluster Naming Conflict	<input checked="" type="checkbox"/>	Cluster		-	LOW		
Disk Utilization	<input checked="" type="checkbox"/>	Cluster v	25	Percent free space	LOW		
System Backup	<input checked="" type="checkbox"/>	Cluster v	14	Days since last successful backup	LOW		

3. Check the boxes in the **Enabled** column to the right of the alert types to enable the alert. Then set the threshold value in the box under **Limit**. This value determines when the alert will be sent. You can uncheck the boxes by alerts that you do not want to appear in the dashboard.
4. You can apply node or cluster specific configuration for the following alerts:
 - Fast Recovery Area Space Utilization
 - High CPU Usage
 - Failed System Backup
 - High Memory Usage
 - Disk Utilization
 - System Backup
 - System User Password Expiration

 **Note:**

Node specific configuration overrides the configuration set at the cluster level.

5. Click **Save**.

- [Alert Configuration at Cluster vs Node Scope](#)
You must consider the following guidelines before configuring the Alerts.

Related Topics

- [Viewing the Oracle Key Vault Dashboard](#)
The dashboard presents the current state of the Oracle Key Vault at a high level and is visible to all users.
- [Managing Oracle Key Vault Platform Certificates](#)
This chapter describes how to manage Oracle Key Vault Platform certificates by monitoring and rotating platform certificates before they expire.

Alert Configuration at Cluster vs Node Scope

You must consider the following guidelines before configuring the Alerts.

For the alerts which support a configurable alert scope, following is the behavior when either the Cluster or the Node scope is selected.

When the Cluster scope is selected:

- On changing the alert configuration, the changes are applied to all nodes which have the scope set to Cluster.
- When an alert is disabled, all previously generated alerts, for selected alert, are deleted from all nodes which have the scope set to Cluster.

When the Node scope is selected:

- On changing the alert configuration, the changes are applied to the current node only.
- When an alert is generated on a node where the Node scope is set, the alert message has a post fix "**node configuration in effect**".
- When an alert is disabled, the previously generated alert, for selected alert, is deleted from the current node.

For the alerts which support a configurable alert scope, following is the behavior when the scope is changed from Cluster to Node or vice versa.

- When the selected scope for an alert is changed from Cluster to Node or vice versa, then the previously generated alert, for selected alert, is deleted from the current node.
- When the selected scope for an alert is changed from Node to Cluster, then the node specific alert configuration is deleted.
- When the selected scope for an alert is changed from Cluster to Node, the Enabled column and Limit column, for selected alert, on Configure Alerts page get populated with corresponding values from cluster alert configuration.

Viewing Open Alerts

Users can view alerts depending on their privileges.

Users with the System Administrator role can view all alerts. Users without system administrator privileges can only view alerts related to objects they can access.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
2. Access the Alerts page by using one of the following methods:
 - Select the **System** tab, then **Settings**, and in the Monitoring and Alerts area, click **Alerts**.
 - Select the **Reports** tab, and then select **Alerts** from the left navigation bar.
 - On the Home page, expand **Alerts** at the top of the page, and then click **All Alerts**. The Home page is also a convenient way to go immediately to how to solve a problem that an alert raises. Under **Alerts**, click **Show Details**, and then in the listing of alert types, click the appropriate link. For example, alerts describing upcoming key expiration will take you to the Alerts configuration page, where only alerts for key rotations are displayed. From there, you can examine the details of keys that are expiring.

The Alerts page appears, displaying all the unresolved alerts. Alerts are listed in the order of their severity and severity based color code scheme. The alerts with higher severity should be resolved first.

When you resolve the issue stated in the alert message, the alerts are automatically removed. To delete an alert message, select it and then click **Delete**. If the issue that caused the alert still exists, then the alert will be regenerated and appear again in this list.

Alert Type	Object	Time	Message	Severity
OKV Server/Node Certificate Expiration		04-30-2022 00:25:07	The Oracle Key Vault Server certificate is expiring within 545 days! This could cause downtime for Oracle Key Vault and all its endpoints!	CRITICAL
User Password Expiration		04-30-2022 15:40:00	User password expiration: 05-30-22 09:59:45 +00:00	MEDIUM
Key Rotations		18-JUN-2022 14:25:00	Key expiration 19-JUN-22 08:55:00 +00:00	MEDIUM
Memory Usage		04-30-2022 18:35:07	Memory usage exceeds 95% (currently 970%)	MEDIUM
System Backup		04-30-2022 01:28:00	No successful back-up done	LOW

Oracle Key Vault sends all system alerts to the syslog. The following is an example of a system alert in syslog:

```
Mar 29 18:36:29 okv080027361e7e logger[13171]: No successful backup done for 4 day(s)
```

The following table lists the conditions that trigger alerts, and the accompanying system alert message:

Condition	System Alert Message
Certificate Object Expiration	Certificate object <i>unique_ID</i> expiration: <i>date</i>
Cluster FIPS Not Consistent	At least one, but not all, active OKV nodes are in FIPS Mode.
Cluster Heartbeat Lag	Replication lag from node <i>node_name</i> to node <i>current_node_name</i> exceeds <i>threshold_value</i> seconds

Condition	System Alert Message
Cluster HSM Not Consistent	At least one, but not all, active OKV nodes are HSM-enabled.
Cluster Naming Conflict	<p>Naming conflict for <i>object_type</i>: <i>object_name</i></p> <p>The label <i>object_type</i> can be endpoint, endpoint group, user, or user group, with <i>object_name</i> being the corresponding object.</p>
Cluster Redo Shipping Status	<p>Any of the following messages:</p> <ul style="list-style-type: none"> • No heartbeats received from node <i>source_node</i> to node <i>current_node</i>. • Last heartbeat from node <i>source_node</i> to node <i>current_node</i> was more than <i>threshold</i> minutes ago. • Last heartbeat from node <i>source_node</i> to node <i>current_node</i> was more than one hour ago. • Last heartbeat from node <i>source_node</i> to node <i>current_node</i> was more than six hours ago. • Last heartbeat from node <i>source_node</i> to node <i>current_node</i> was more than <i>Maximum_Disable_Node_Duration</i> hours (50% of the Maximum Disable Node Duration) ago. Take immediate action to restore communication from node <i>source_node</i> to avoid issues in the cluster. • Last heartbeat from node <i>source_node</i> to node <i>current_node</i> was more than <i>Maximum_Disable_Node_Duration</i> hours (75% of the Maximum Disable Node Duration) ago. Take immediate action to restore communication from node <i>source_node</i> to avoid issues in the cluster. • Last heartbeat from node <i>source_node</i> to node <i>current_node</i> was more than <i>Maximum_Disable_Node_Duration</i> hours (Maximum Disable Node Duration) ago. Node <i>current_node</i> may not have received all records from node <i>source_node</i> even if replication is restored.
Cluster Replication Lag	Replication lag from node <i>node_name</i> to node <i>node_name</i> is greater than <i>threshold</i> seconds. Current lag is <i>current</i> seconds.

Condition	System Alert Message
Disk utilization	<p>When Cluster scope is set:</p> <ul style="list-style-type: none"> Free disk space is below <i>threshold_value</i> (currently <i>current_value</i>) Free disk space is below <i>threshold_value</i> (currently <i>current_value</i>) - node configuration in effect
Endpoint certificate expiration	<p>Endpoint <i>endpoint_name</i> certificate expiration <i>date</i></p>
Failed OKV Services	<p><i>service_name</i> service is failed or <i>service_name1</i>, <i>service_name2</i> services are failed.</p>
Failed system backup	<p>When Cluster scope is set:</p> <ul style="list-style-type: none"> Most recent backup failed! <p>When Node scope is set:</p> <ul style="list-style-type: none"> Most recent backup failed! - node configuration in effect
Fast Recovery Area Space Utilization	<p>When Cluster scope is set:</p> <ul style="list-style-type: none"> Fast Recovery Area space usage exceeds <i>threshold_value%</i> (currently <i>current_value%</i>) <p>When Node scope is set:</p> <ul style="list-style-type: none"> Fast Recovery Area space usage exceeds <i>threshold_value%</i> (currently <i>current_value%</i>) - node configuration in effect
High CPU usage	<p>When Cluster scope is set:</p> <ul style="list-style-type: none"> CPU usage exceeds <i>threshold_value</i> (currently <i>current_value</i>) <p>When Node scope is set:</p> <ul style="list-style-type: none"> CPU usage exceeds <i>threshold_value</i> (currently <i>current_value</i>) - node configuration in effect
High Memory Usage	<p>When Cluster scope is set:</p> <ul style="list-style-type: none"> Memory usage exceeds <i>threshold_value</i> (currently <i>current_value</i>) <p>When Node scope is set:</p> <ul style="list-style-type: none"> Memory usage exceeds <i>threshold_value</i> (currently <i>current_value</i>) - node configuration in effect
Invalid HSM Configuration	<p>HSM configuration error. Please refer to the HSM Alert section in the Oracle Key Vault HSM Integration Guide</p>
Key rotations	<p>Key <i>unique_ID</i> expiration: <i>date</i></p>
Primary-standby destination failure	<p>One or more standby servers are in an error state. HA destination failure.</p>
Primary-standby Oracle Data Guard Broker status	<p>Data Guard Broker is disabled</p>

Condition	System Alert Message
Primary-standby Oracle Data Guard fast-start failover status	HA FSFO is not synchronized. FSFO status is <i>HA_status</i>
Primary-standby restricted mode	HA running in read-only restricted mode
Primary-standby role change	HA role changed. Primary IP Address: <i>IP_address</i>
Secret Object Expiration	Secret object <i>unique_ID</i> expiration : <i>date</i>
SSH tunnel failure	SSH tunnel (IP <i>IP_address</i>) is not available
System backup	<p>When Cluster scope is set:</p> <ul style="list-style-type: none"> No successful backup for number day(s) <p>When Node scope is set:</p> <ul style="list-style-type: none"> No successful backup for number day(s) - node configuration in effect
User password expiration	User <i>user_name</i> password expiration: <i>date</i>

Managing System Auditing

Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.

- [About Auditing in Oracle Key Vault](#)
Oracle Key Vault records and time-stamps all endpoint and user activity.
- [Oracle Key Vault Audit Trail](#)
The Oracle Key Vault audit trail captures information about activities that are performed in Oracle Key Vault, such as the name of an action and who performed it.
- [Oracle Key Vault Audit Configuration](#)
Auditing in Oracle Key Vault is enabled by default on all the operations. Each operation type in Oracle Key Vault is uniquely identified by the audit event ID. Oracle Key Vault provides options to configure auditing for specific operations based on audit event categories and audit event IDs.
- [Viewing Audit Records](#)
To view audit records, access the Oracle Key Vault management console Audit Trail page.
- [Exporting and Deleting Audit Records Manually](#)
Oracle Key Vault audit records are stored in a `.csv` file.
- [Deleting Audit Records Automatically](#)
You can configure Oracle Key Vault to automatically delete or purge the audit records that are older than the specified retention period.
- [Oracle Key Vault Audit Event IDs](#)
Oracle Key Vault Audit Event ID identifies the audit operation type.
- [Configuring Oracle Key Vault with Oracle Audit Vault](#)
A user who has the Audit Manager role can configure Oracle Key Vault to send audit records to Oracle Audit Vault for centralized audit reporting and alerting.

About Auditing in Oracle Key Vault

Oracle Key Vault records and time-stamps all endpoint and user activity.

The audit records include endpoint groups and user groups, from endpoint enrollment and user password reset, to the management of keys and wallets, and changes to system settings and SNMP credentials. The audit trail captures details on who initiated which action, with what keys and tokens, and the result of the action. In addition, it records the success or failure of each action.

Auditing in Oracle Key Vault is enabled by default.

A user with the Audit Manager role can select events to be audited in the system and also manage the audit trail. Other users can see only those audit records that relate to the security objects they have created or have been granted access to.

The audit manager can export audit records to view system activity offline. After exporting the records, the audit manager can delete them from the system to free up resources.

Related Topics

- [Audit Manager Role Duties](#)

The Oracle Key Vault Audit Manager is responsible for the following audit-related tasks.

Oracle Key Vault Audit Trail

The Oracle Key Vault audit trail captures information about activities that are performed in Oracle Key Vault, such as the name of an action and who performed it.

The following table lists the contents of the Oracle Key Vault audit trail.

Table 22-4 Oracle Key Vault Audit Trail

Column Name	Description
Event ID	ID to identify each audit operation type uniquely.
Client IP	The IP address of the client host or the IP address of the proxy server between the client and the Oracle Key Vault server that is making its IP address available to Oracle Key Vault
Node ID	ID of the Oracle Key Vault cluster node on which the operation was performed
Node IP Address	IP address of the Oracle Key Vault cluster node on which the operation was performed
Node Name	Name of the Oracle Key Vault cluster node on which the operation was performed
Object	Captures the name of object on which the operation is performed
Object Type	Type of object on which the operation is performed (for example, User, Endpoint)
Operation	Name of the operation performed
Result	Result of the operation indicating whether it was successful or failure

Table 22-4 (Cont.) Oracle Key Vault Audit Trail

Column Name	Description
Subject	Captures the name of entity that performed the operation
Subject Type	Type of entity, User or Endpoint
Time	Timestamp of the operation

- [Enabling Auditing and Configuring Syslog to Store Audit Records](#)
You can enable or disable auditing and then configure the Oracle Key Vault syslog to store audit records if the System Administrator has enabled this functionality.

Enabling Auditing and Configuring Syslog to Store Audit Records

You can enable or disable auditing and then configure the Oracle Key Vault syslog to store audit records if the System Administrator has enabled this functionality.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Access the Audit Settings page by using one of the following methods:
 - Select the **Systems** tab, then **Settings** in the left navigation bar. In the Monitoring and Alerts area, click **Audit**.
 - Select the **Reports** tab, then **Audit Trail** in the left navigation bar. In the Audit Trail page, click **Audit Settings**.

The Audit Settings page appears. The categories that you can configure are as follows:

- Auto Purge Audit Records
 - Enable Auditing
 - Send Audit Records to Syslog
3. Next, do the following:
 - a. In a multi-master cluster environment, click the arrow on the right that appears next to the **Save** button to toggle between the current node and the entire cluster. Also, the **Auto Purge Audit Records** setting can only be configured at the node level only.
 - b. Select either the **Yes** or **No** option for the auditing.
 - c. Click **Save**.
 4. If syslog is configured, then perform additional steps as needed.

If syslog is not configured, then the `Syslog forwarding to remote machines not enabled` error message appears. If this error appears, then enable syslog.

- a. Select the **System** tab, and then select **Settings**.
- b. In the Monitoring and Alerts area, select **Syslog**.
- c. In a multi-master cluster environment, toggle between Node Details - Effective on this Node and Cluster Details by clicking the arrow under the **Save** button.
 - a. Select the protocol to use to transfer syslog files: **TCP** or **UDP**.
 - b. Enter the IP address of the remote system where the syslog files will be stored.
 - c. Click **Save**.

 **Note:**

To avoid any accidental deletion of Audit records it is recommended to use TCP protocol for transferring syslog files.

Oracle Key Vault Audit Configuration

Auditing in Oracle Key Vault is enabled by default on all the operations. Each operation type in Oracle Key Vault is uniquely identified by the audit event ID. Oracle Key Vault provides options to configure auditing for specific operations based on audit event categories and audit event IDs.

- [About Oracle Key Vault Audit Configuration](#)
Oracle Key Vault audits all system operations by default and also offers various levels of customization in the event auditing.
- [Configuring All Events Auditing](#)
By default, Oracle Key Vault audits all the operations in the system.
- [Configuring Custom Events Auditing](#)
Oracle Key Vault provides different ways to configure custom events auditing. You can configure custom events auditing through the audit event categories and the audit event IDs.
- [Configuring Auditing on Event Categories](#)
Oracle Key Vault groups related audit events into Audit Event Categories. You can configure auditing based on the Audit Event Categories, thereby enabling auditing of all the event IDs belonging to those categories.
- [Including Specific Audit Event IDs for Auditing](#)
Oracle Key Vault provides an option to include specific audit event ID from the unselected event categories using the Include Events functionality. Including an event from the unselected category helps in auditing the specific event even if the category is not selected. This helps in selectively auditing critical events in the system.
- [Excluding Specific Audit Event IDs from Auditing](#)
Oracle Key Vault provides an option to exclude the specific audit event ID from the selected category using the Exclude Events functionality. Excluding an event from the selected event category disables auditing of the particular event in the system.

About Oracle Key Vault Audit Configuration

Oracle Key Vault audits all system operations by default and also offers various levels of customization in the event auditing.

You can configure custom event auditing to selectively audit critical events at an audit event category level or at an audit event ID level. This flexibility allows you to fine-tune audit configuration and perform targeted auditing of the system. You can also fine-tune the custom event auditing configuration as per the requirements before making them effective in the system.

Oracle Key Vault groups related audit events together into audit event categories. When you configure auditing at the audit event category level, then all the events that belong to the audit event category are audited in the system without the need to configure each individual event. You can enable the auditing of broader areas of interest more easily by enabling the auditing at the category level.

For more granular control over events to audit, you can set the audit configuration using individual event IDs. You can include or exclude specific events from auditing using their audit event IDs.

You can use a combination of audit event categories and individual event IDs to perform targeted auditing of the system and avoid generating audit records for non-critical events in the system.

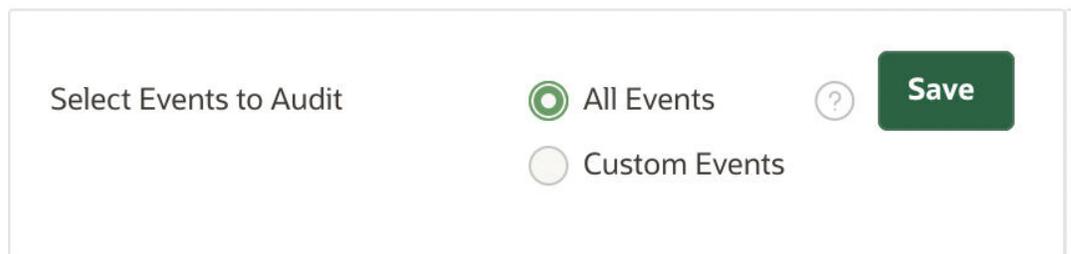
Configuring All Events Auditing

By default, Oracle Key Vault audits all the operations in the system.

You can configure auditing on the Audit Settings page of the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Access the Audit Settings page by using one of the following methods:
 - From the **Systems** tab:
 - a. Select the **Systems** tab, and then **Settings** in the left navigation bar.
 - b. From the **Monitoring and Alerts** area, click **Audit**.
 - From the **Reports** tab:
 - a. Select the **Reports** tab, and then select **Audit Trail** in the left navigation bar.
 - b. In the **Audit Trail** page, click **Audit Settings**.

The **Audit Settings** page is displayed. The **Advanced Audit Configuration** section shows the **Select Events to Audit** pane.



Select Events to Audit

All Events Custom Events

Save

3. Select **All Events** to audit all the operations in Oracle Key Vault.

 **Note:**

The **All Events** option is the default selection for auditing all the operations in Oracle Key Vault.

4. Click **Save**.

Configuring Custom Events Auditing

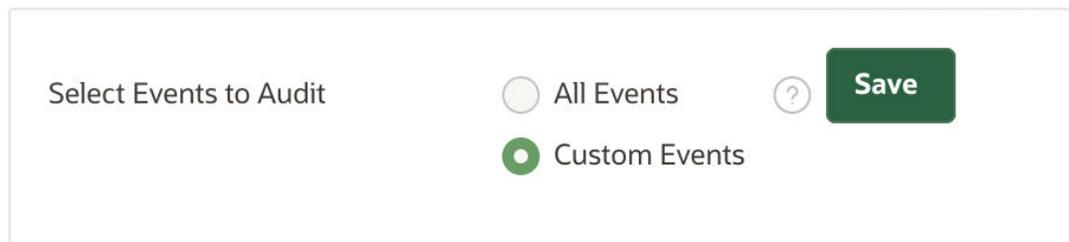
Oracle Key Vault provides different ways to configure custom events auditing. You can configure custom events auditing through the audit event categories and the audit event IDs.

Perform the following steps to configure custom events auditing:

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Access the Audit Settings page by using one of the following methods:
 - From the **Systems** tab:
 - a. Select the **Systems** tab and then **Settings** in the left navigation bar.
 - b. From the **Monitoring and Alerts** area, click **Audit**.
 - From the **Reports** tab:
 - a. Select the **Reports** tab and then select **Audit Trail** in the left navigation bar.
 - b. In the **Audit Trail** page, click **Audit Settings**.

The Audit Settings page is displayed. The **Advanced Audit Configuration** section shows the **Select Events to Audit** pane.

3. Select the **Custom Events** option. The **Custom Events** area to configure auditing based on event categories or event IDs is displayed.



Select Events to Audit

All Events Custom Events

Save

4. Click **Save**. The **Custom Events** Auditing is now enabled in the system and the **Custom Events** audit configuration is applied. The following screen displays the **Custom Events** page with the event categories.

Configuring Auditing on Event Categories

Oracle Key Vault groups related audit events into Audit Event Categories. You can configure auditing based on the Audit Event Categories, thereby enabling auditing of all the event IDs belonging to those categories.

You can use Audit Categories to audit group of Audit Event IDs in the system.

Table 22-5 Oracle Key Vault Audit Event Categories

Audit Event Categories	Description
Authentication	Includes audit events related to user authentication, failed login attempts.
User Management	Includes audit events related to user account management, user profile management, and system recovery.
Endpoint Management	Includes audit events related to endpoint management and associated configurations.
Privilege Management	Includes audit events related to management of user groups, endpoint groups, wallets, and associated privileges.
Deployment Operations	Includes audit events related to cluster management, replication services and primary-standby configuration.
Keys & Secret Operations	Includes audit events related to keys and secrets management using KMIP.
System Management	Includes audit events related to alerts, FIPS mode, system configuration, such as network settings, SSH, and so on.
Audit Management	Includes audit events related to audit configuration, replication, and audit trail management.

Table 22-5 (Cont.) Oracle Key Vault Audit Event Categories

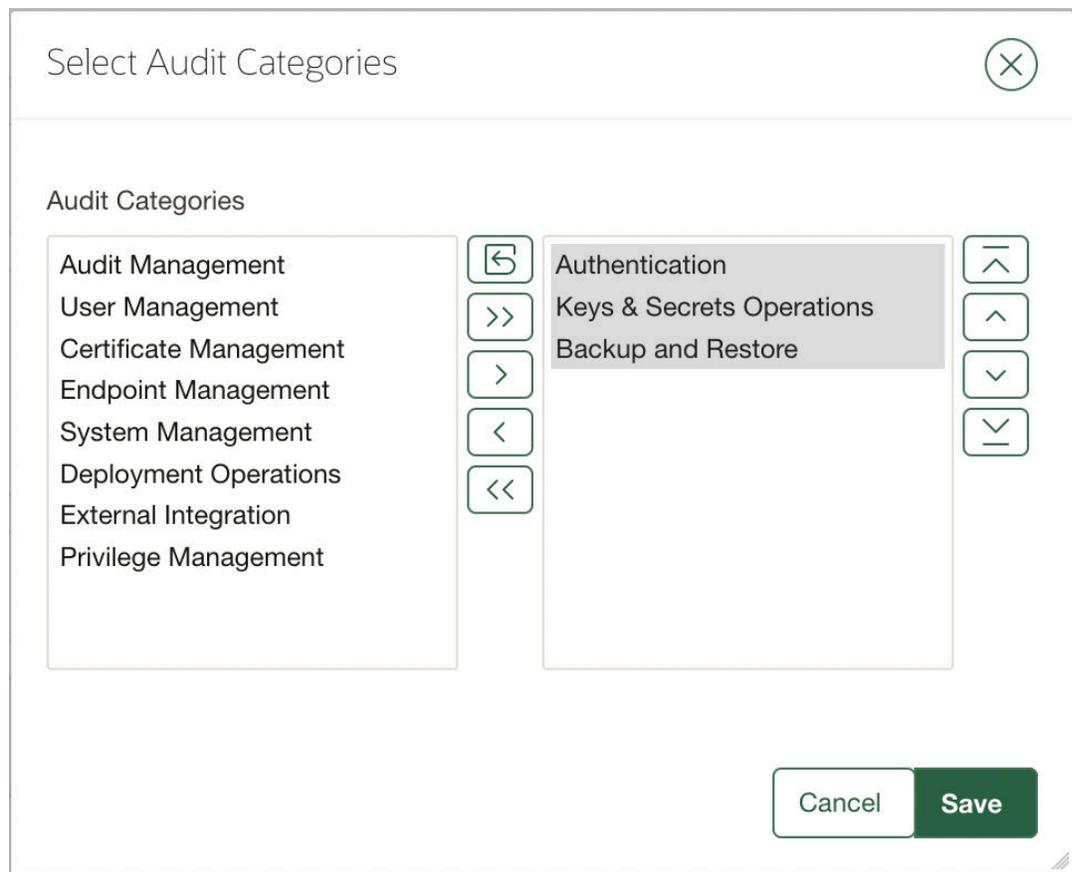
Audit Event Categories	Description
Backup and Restore	Includes audit events related to backup, restore operations, and backup destinations.
Certificate Management	Includes audit events related to certificate generation, certificate rotation and so on.
External Integration	Includes audit events related to LDAP configuration, single sign-on, Audit Vault integration, HSM configuration, and so on.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Access the Audit Settings page by using one of the following methods:
 - From the **Systems** tab:
 - a. Select the **Systems** tab and then **Settings** in the left navigation bar.
 - b. From the **Monitoring and Alerts** area, click **Audit**.
 - From the **Reports** tab:
 - a. Select the **Reports** tab and then select **Audit Trail** in the left navigation bar.
 - b. In the **Audit Trail** page, click **Audit Settings**.

The Audit Settings page is displayed. The **Advanced Audit Configuration** section shows the **Select Events to Audit** pane.
3. Select the **Custom Events** option. The **Custom Events** area to configure auditing based on event categories or event IDs is displayed.
4. In the **Custom Events** section, click **Select Event Categories**. **Select Audit Categories** dialog box is displayed.
5. From the **Audit Categories** box, click the greater than > icon to include the audit category one-by-one. Alternatively, you can click the >> icon to include all the audit categories at once.

 **Note:**

To enable the audit category move the selected audit category to the right. To disable move the selected audit category to the left box.



6. Click **Save**.

Note:

- Enabling auditing on an event category automatically deletes the audit event IDs in that category from the **Events Included from Unselected categories** list. Also, disabling auditing on an event category automatically deletes the audit event IDs in that category from the **Events Excluded from Selected Categories** list.

Including Specific Audit Event IDs for Auditing

Oracle Key Vault provides an option to include specific audit event ID from the unselected event categories using the Include Events functionality. Including an event from the unselected category helps in auditing the specific event even if the category is not selected. This helps in selectively auditing critical events in the system.

Perform the following steps to Include Events from the Unselected Event Categories:

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Access the Audit Settings page by using one of the following methods:
 - From the **Systems** tab:

- a. Select the **Systems** tab and then **Settings** in the left navigation bar.
- b. From the **Monitoring and Alerts** area, click **Audit**.
- From the **Reports** tab:
 - a. Select the **Reports** tab and then select **Audit Trail** in the left navigation bar.
 - b. In the **Audit Trail** page, click **Audit Settings**.
The Audit Settings page is displayed. The **Advanced Audit Configuration** section shows the **Select Events to Audit** pane.
3. In the **Events Included from Unselected Categories** section, click **Include Events**.
The **Include Events** dialog box is displayed.
4. From the **Include Events** dialog box, select a category from **Unselected Event Categories**. This displays the categories on which auditing is disabled.
5. From the **Include Events** box, click the greater than > icon to include the audit event ID one-by-one. Alternatively, you can click the >> icon to include all the audit event IDs at once.

Include Events

6. Click **Save**.

Excluding Specific Audit Event IDs from Auditing

Oracle Key Vault provides an option to exclude the specific audit event ID from the selected category using the Exclude Events functionality. Excluding an event from the selected event category disables auditing of the particular event in the system.

Excluding the events from the selected category avoids audit record generation for non-critical events in the system, thereby ensuring effective usage of system resources.

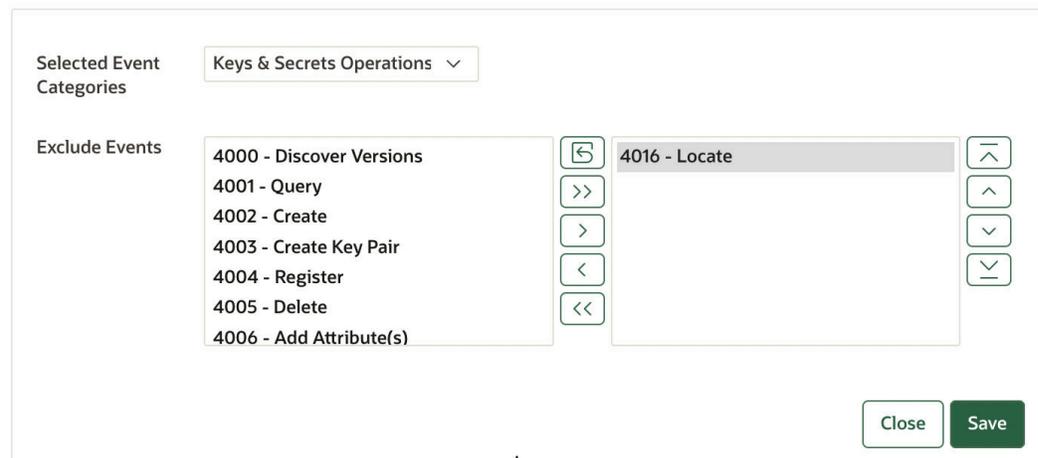
Perform the following steps to exclude specific event IDs from the selected categories.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Access the Audit Settings page by using one of the following methods:

- From the **Systems** tab:
 - a. Select the **Systems** tab and then **Settings** in the left navigation bar.
 - b. From the **Monitoring and Alerts** area, click **Audit**.
- From the **Reports** tab:
 - a. Select the **Reports** tab and then select **Audit Trail** in the left navigation bar.
 - b. In the **Audit Trail** page, click **Audit Settings**.

The Audit Settings page is displayed. The **Advanced Audit Configuration** section shows the **Select Events to Audit** pane.
- 3. In the **Events Included from Unselected Categories** section, click **Exclude Events**.
Exclude Events dialog box is displayed.
- 4. From the **Exclude Events** dialog box, select category from **Selected Event Categories**. This displays the categories where auditing is enabled.
- 5. From the **Exclude Events** box, click the greater than > icon to exclude the audit event ID one-by-one. Alternatively, you can click the >> icon to exclude all the audit event IDs at once.

Exclude Events



6. Click **Save**.
The final page after making the customization(s) look similar to the screen displayed.

Select Events to Audit

Select Events to Audit

All Events ? Save

Custom Events

▼ Custom Events

Excluding certain events from auditing may affect Oracle Key Vault reports that are based on audit trail.

Custom Events configuration applies only when you selected 'Custom Events' option for the 'Select Events to Audit'.

Selected Event Categories Select Event Categories

Authentication

Keys & Secrets Operations

Backup and Restore

Events Included from Unselected Categories Include Events

1300 - Add Endpoint

1301 - Delete Endpoint

Events Excluded from Selected Categories Exclude Events

4016 - Locate

Viewing Audit Records

To view audit records, access the Oracle Key Vault management console Audit Trail page.

The reports page shows the Audit Trail page by default. The Audit Trail page lists all system activity with details on who performed an operation, when the operation was performed, what object was used to perform the operation, and the result.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Click the **Reports** tab, and then **Audit Trail** from the left navigation bar.

The Audit Trail page appears. Optionally, filter records by selecting the table column heads, and from the drop-down list, select the type of sort order that you want. The private key is captured as the object in the audit records for the create key pair operation.

Each Event ID indicates the specific event uniquely associated with an audited operation. For example, operations associated with login Attempted are grouped under the Event ID 1000. For example, to search the audit records for successful logins, filter the audit records with Event ID of 1001.

Note:

For the audit records that are generated prior to Oracle Key Vault 21.6, the **Event ID** column does not have any value.

Exporting and Deleting Audit Records Manually

Oracle Key Vault audit records are stored in a `.csv` file.

A user with the Audit Manager role can export the audit trail to a `.csv` file that can be downloaded to the user's local system. The `.csv` file contains the same details found in the audit trail on the Reports page. The timestamp in the `.csv` file reflects the time zone of the particular Oracle Key Vault server whose records were exported. After you export the records, you can delete them from the Oracle Key Vault server to free up space.

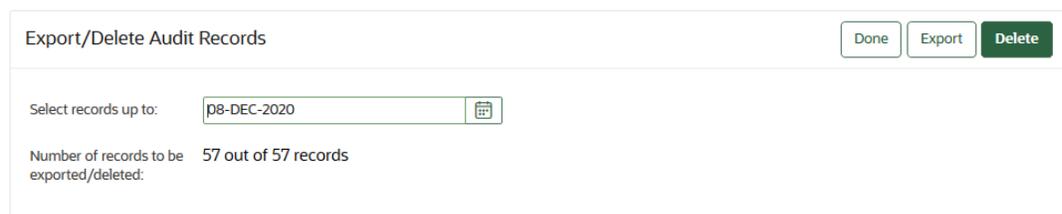
1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.

2. Click the **Reports** tab, then **Audit Trail** in the left navigation bar.

The Audit Trail appears.

3. Click **Export/Delete Audit Records** on the top right.

The **Export/Delete Audit Records** page appears.



4. Select the date by clicking the calendar icon.

Based on the date that you select, the number of records appears after the **Number of records to be exported/deleted** label.

5. Click **Export** to download the audit records in `.csv` file format to a local folder.

After you export the records, you can delete them from Oracle Key Vault to free up resources.

6. Click **Delete** to remove the audit records.
7. Click **OK** to delete or **Cancel** to stop.

Deleting Audit Records Automatically

You can configure Oracle Key Vault to automatically delete or purge the audit records that are older than the specified retention period.

The **Audit Settings** page shows the **Auto Purge Audit Records** pane. The user with the Audit Manager role can purge the Audit Records.

1. Log in to the Oracle Key Vault management console as a user who has the audit manager role.
2. Access the **Audit Settings** page by using one of the following methods:
 - Select the **Systems** tab, then **Settings** in the left navigation bar. In the Monitoring and Alerts area, click **Audit**.
 - Select the **Reports** tab, then **Audit Trail** in the left navigation bar. In the Audit Trail page, click **Audit Settings**.

The **Audit Settings** page appears. The categories that you can configure are as follows:

- Auto Purge Audit Records
 - Enable Auditing
 - Send Audit Records to Syslog
 - Replicate Audit Records, available in cluster environment.
3. Select **Yes**, in the **Auto Purge Audit Records** pane.

The **Retention Period in Days** field appears.

4. Enter the number of days to retain the audit records. Oracle Key Vault will now periodically purge audit records that are older than the specified number of days.
5. Click **Save**.

Note:

When Oracle Key Vault is integrated with Oracle Audit Vault, audit records are purged only after they are collected by the Audit Vault.

Related Topics

- [Exporting and Deleting Audit Records Manually](#)
Oracle Key Vault audit records are stored in a `.csv` file.

Oracle Key Vault Audit Event IDs

Oracle Key Vault Audit Event ID identifies the audit operation type.

The Event ID in Oracle Key Vault audit records represents a stable identity that is uniquely associated with an audit operation type.

Table 22-6 Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
1000	Login Attempted	Authentication
1001	Login Attempted and Succeeded	Authentication
1002	Logout	Authentication
1003	Logout on Console Timeout	Authentication
1100	System Recovery: Initiate Recovery Passphrase Change	User Management
1101	System Recovery: Reset Recovery Passphrase Change	User Management
1102	System Recovery: Change Recovery Passphrase	User Management
1103	System Recovery: Reset Administrative Accounts	User Management
1104	System Recovery: Manage Administrator	User Management

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
1105	System Recovery: Modify User Account Profile Parameters	User Management
1106	System Recovery: Reset User Account Profile Parameters	User Management
1107	System Recovery: Modify Enforce Separation of Administrator Roles Parameter	User Management
1200	Add User	User Management
1201	Delete User	User Management
1202	Modify User Attributes	User Management
1203	Reset User Password	User Management
1204	Change User Password	User Management
1205	Password Email Sent	User Management
1300	Add Endpoint	Endpoint Management
1301	Delete Endpoint	Endpoint Management
1302	Modify Endpoint Attributes	Endpoint Management
1303	Enroll Endpoint	Endpoint Management
1304	Enrollment Token Email Sent	Endpoint Management
1305	Get Enrollment Token	Endpoint Management
1306	Reenroll Endpoint	Endpoint Management
1307	Suspend Endpoint	Endpoint Management
1308	Resume Endpoint	Endpoint Management
1309	Rotate Endpoint Certificate	Endpoint Management
1310	Endpoint Certificate Rotation Initiated	Endpoint Management
1311	Endpoint Certificate Rotation Completed	Endpoint Management
1312	Modify Endpoint Configuration Parameters	Endpoint Management
1313	Reset Endpoint Configuration Parameters	Endpoint Management
1314	Clear Endpoint Configuration Parameters	Endpoint Management
1315	Modify Endpoint Settings for Keys & Secrets	Endpoint Management & Secrets
1316	Reset Endpoint Settings for Keys & Secrets	Endpoint Management & Secrets
1317	Clear Endpoint Settings for Keys & Secrets	Endpoint Management & Secrets
1400	Modify Endpoint Self Enrollment Setting	Endpoint Management
1401	Modify Global Endpoint Configuration Parameters	Endpoint Management
1402	Reset Global Endpoint Configuration Parameters	Endpoint Management
1403	Modify Global Endpoint Settings for Keys & Secrets	Endpoint Management

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
1404	Reset Global Endpoint Settings for Keys & Secrets	Endpoint Management
2000	Add User Group	Privilege Management
2001	Delete User Group	Privilege Management
2002	Modify User Group Attributes	Privilege Management
2003	Add User Group Member(s)	Privilege Management
2004	Drop User Group Member(s)	Privilege Management
2005	Add User Group Membership	Privilege Management
2006	Drop User Group Membership	Privilege Management
2100	Add Endpoint Group	Privilege Management
2101	Delete Endpoint Group	Privilege Management
2102	Modify Endpoint Group Attributes	Privilege Management
2103	Add Endpoint Group Member(s)	Privilege Management
2104	Drop Endpoint Group Member(s)	Privilege Management
2105	Add Endpoint Group Membership	Privilege Management
2106	Drop Endpoint Group Membership	Privilege Management
2200	Create Wallet	Privilege Management
2201	Delete Wallet	Privilege Management
2202	Modify Wallet	Privilege Management
2203	Add Wallet Membership	Privilege Management
2204	Drop Wallet Membership	Privilege Management
2205	Add Wallet Member(s)	Privilege Management
2206	Remove Wallet Member(s)	Privilege Management
2207	Add Access Mapping	Privilege Management
2208	Delete Access Mapping	Privilege Management
2209	Modify Access Mapping	Privilege Management
2210	Assign Default Wallet	Privilege Management
2211	Grant Endpoint Management Privilege	Privilege Management
2212	Revoke Endpoint Management Privilege	Privilege Management
2213	Grant Endpoint Group Management Privilege	Privilege Management
2214	Revoke Endpoint Group Management Privilege	Privilege Management
3000	Convert Server to First Cluster Node	Deployment Operations
3001	Convert Server to Candidate Node	Deployment Operations
3002	Add Candidate Node	Deployment Operations
3003	Abort Conversion of Candidate Node	Deployment Operations
3004	Abort Candidate Node	Deployment Operations
3005	Add Node to Cluster	Deployment Operations

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
3006	Abort Addition of Node to Cluster	Deployment Operations
3007	Finish Pairing on Candidate Node	Deployment Operations
3008	Initiate Delete Node	Deployment Operations
3009	Initiate Force Delete Node	Deployment Operations
3010	Cleanup Deleted Node	Deployment Operations
3011	Finish Delete Node	Deployment Operations
3012	Initiate Enable Node	Deployment Operations
3013	Finish Enable Node	Deployment Operations
3014	Initiate Disable Node	Deployment Operations
3015	Finish Disable Node	Deployment Operations
3016	Cancel Disable Node	Deployment Operations
3017	Edit Cluster Subgroup	Deployment Operations
3018	Start Replication for Node	Deployment Operations
3019	Stop Replication for Node	Deployment Operations
3020	Start Cluster Services	Deployment Operations
3021	Stop Cluster Services	Deployment Operations
3022	User Name Conflict Resolution	Deployment Operations
3023	User Group Name Conflict Resolution	Deployment Operations
3024	Endpoint Name Conflict Resolution	Deployment Operations
3025	Endpoint Group Name Conflict Resolution	Deployment Operations
3026	Wallet Name Conflict Resolution	Deployment Operations
3027	KMIP Attribute Name Conflict Resolution	Deployment Operations
3900	Initiate Primary-Standby Pairing	Deployment Operations
3901	Initiate Primary-Standby Unpairing	Deployment Operations
3902	Primary-Standby Role Switch	Deployment Operations
4000	Discover Versions	Keys & Secrets Operations
4001	Query	Keys & Secrets Operations
4002	Create	Keys & Secrets Operations
4003	Create Key Pair	Keys & Secrets Operations
4004	Register	Keys & Secrets Operations
4005	Delete	Keys & Secrets Operations
4006	Add Attribute(s)	Keys & Secrets Operations
4007	Delete Attribute(s)	Keys & Secrets Operations
4008	Modify Attribute(s)	Keys & Secrets Operations
4009	Get Attribute(s)	Keys & Secrets Operations
4010	Get Attribute List	Keys & Secrets Operations
4011	Check Object Status	Keys & Secrets Operations
4012	Activate	Keys & Secrets Operations

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
4013	Revoke	Keys & Secrets Operations
4014	Destroy	Keys & Secrets Operations
4015	Check	Keys & Secrets Operations
4016	Locate	Keys & Secrets Operations
4017	Get	Keys & Secrets Operations
4018	Rekey	Keys & Secrets Operations
4019	Store Endpoint Metadata	Keys & Secrets Operations
4020	Encrypt	Keys & Secrets Operations
4021	Decrypt	Keys & Secrets Operations
4022	Sign	Keys & Secrets Operations
4023	Signature Verify	Keys & Secrets Operations
4024	Get Endpoint Information	Keys & Secrets Operations
5000	Power Off	System Management
5001	Reboot System	System Management
5002	Rebooting the Server Operating with FIPS Mode Enabled	System Management
5003	Rebooting the Server Operating with FIPS Mode Disabled	System Management
5100	Modify Trace Levels	System Management
5101	Download Diagnostics	System Management
5200	Configure Alerts	System Management
5201	Delete Alert(s)	System Management
5202	Critical Alert(s) Ignored	System Management
5203	Critical Alert(s) Viewed	System Management
5300	Modify Monitoring Settings	System Management
5301	Modify Monitoring Settings for Cluster	System Management
5302	Clear Monitoring Settings for Node	System Management
5303	Modify SNMP Credentials	System Management
5400	Enable Auditing	Audit Management
5401	Disable Auditing	Audit Management
5402	Enable Auditing in Cluster	Audit Management
5403	Disable Auditing in Cluster	Audit Management
5404	Modify Audit Settings	Audit Management
5405	Modify Audit Settings for Cluster	Audit Management
5406	Clear Audit Settings for Node	Audit Management
5407	Enable Audit Replication	Audit Management
5408	Disable Audit Replication	Audit Management
5409	Enable Audit Replication in Cluster	Audit Management
5410	Disable Audit Replication in Cluster	Audit Management
5411	Clear Audit Replication Settings for Node	Audit Management

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
5412	Start Sending Audit Records to SYSLOG	Audit Management
5413	Stop Sending Audit Records to SYSLOG	Audit Management
5414	Start Sending Audit Records to SYSLOG in Cluster	Audit Management
5415	Stop Sending Audit Records to SYSLOG in Cluster	Audit Management
5416	Clear Send Audit Records to SYSLOG Settings for Node	Audit Management
5417	Enable Audit Records Auto Purge	Audit Management
5418	Disable Audit Records Auto Purge	Audit Management
5419	Clear Audit Records Auto Purge Settings for Node	Audit Management
5420	Delete Audit Records	Audit Management
5421	Modify Audit Categories	Audit Management
5422	Modify Audit Include Events	Audit Management
5423	Modify Audit Exclude Events	Audit Management
5600	Modify System Settings	System Management
5601	Modify Network Details	System Management
5602	Modify DNS Settings	System Management
5603	Modify DNS Settings for Cluster	System Management
5604	Clear DNS Settings for Node	System Management
5605	Modify System Time Settings	System Management
5606	Modify NTP Settings	System Management
5607	Modify NTP Settings for Cluster	System Management
5608	Clear NTP Settings for Node	System Management
5609	Modify Web Access	System Management
5610	Modify SSH Access	System Management
5611	Modify Management Console Timeout Settings	System Management
5612	Enable FIPS Mode	System Management
5613	Disable FIPS Mode	System Management
5614	Enable RESTful Services	System Management
5615	Disable RESTful Services	System Management
5616	Modify Syslog Settings	System Management
5617	Modify Syslog Settings for Cluster	System Management
5618	Clear Syslog Settings for Node	System Management
5619	Configure SMTP	System Management
5620	Unconfigure SMTP	System Management
5621	Modify SMTP Settings	System Management
5622	Modify Maximum Disable Node Duration Settings for Cluster	System Management

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
5623	Add SSH Tunnel	System Management
5624	Delete SSH Tunnel	System Management
5625	Enable SSH Tunnel	System Management
5626	Disable SSH Tunnel	System Management
5627	Tunnel Cleanup on Failover	System Management
5628	Delete Cloud IP Reservation	System Management
6000	Restore	Backup and Restore
6001	Create Backup	Backup and Restore
6002	Delete Backup	Backup and Restore
6003	Edit Backup	Backup and Restore
6004	Backup Initiated	Backup and Restore
6005	Backup Completed	Backup and Restore
6006	Pause Backup	Backup and Restore
6007	Resume Backup	Backup and Restore
6008	Purge Backup	Backup and Restore
6009	Create Backup Destination	Backup and Restore
6010	Delete Backup Destination	Backup and Restore
6011	Edit Backup Destination	Backup and Restore
6012	Reset Backup Host Public Key	Backup and Restore
6013	Create Backup Destination Policy	Backup and Restore
6014	Delete Backup Destination Policy	Backup and Restore
6015	Edit Backup Destination Policy	Backup and Restore
6016	Suspend Backup Destination Policy	Backup and Restore
6017	Resume Backup Destination Policy	Backup and Restore
6100	Generate New Server Certificate	Certificate Management
6101	Generate New Node Certificate	Certificate Management
6102	Modify Server Certificate Details	Certificate Management
6103	Modify Server Certificate Alternate Name and IP Address Details	Certificate Management
6104	Modify CA Certificate Details	Certificate Management
6105	Generate and Download Intermediate CA Certificate Request	Certificate Management
6106	Abort Generation of Intermediate CA Certificate	Certificate Management
6107	Download Intermediate CA Certificate Request	Certificate Management
6108	Upload Intermediate CA Certificate	Certificate Management
6109	Upload Intermediate CA Trust Chain	Certificate Management
6110	Generate New CA Certificate	Certificate Management

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
6111	Unpack New CA Certificate Bundle	Certificate Management
6112	Abort CA Certificate Rotation	Certificate Management
6113	Activate New CA Certificate	Certificate Management
6114	Update Endpoint Certificate Rotation Window Size	Certificate Management
6115	Update Cluster Subgroup Order	Certificate Management
6116	Remove Cluster Subgroup Order	Certificate Management
6117	Update Endpoint Group Order	Certificate Management
6118	Check Expired CA Certificate Rotation State	Certificate Management
6119	Generate New CA Certificate for Manual Recovery	Certificate Management
6120	Upload CA Certificate Bundle	Certificate Management
6121	CA Certificate Rotation Manual Recovery Completed	Certificate Management
6122	CA Certificate Rotation Manual Recovery Verified	Certificate Management
6123	Abort CA Certificate Rotation Manual Recovery	Certificate Management
6180	Generate Console Certificate	Certificate Management
6181	Upload Console Certificate	Certificate Management
6182	Remove Console Certificate	Certificate Management
6183	Restore Console Certificate	Certificate Management
8000	Add LDAP Configuration	External Integration
8001	Delete LDAP Configuration	External Integration
8002	Force Delete LDAP Configuration	External Integration
8003	Test LDAP Configuration	External Integration
8004	Edit LDAP Configuration	External Integration
8005	Enable LDAP Configuration	External Integration
8006	Disable LDAP Configuration	External Integration
8007	Add LDAP Server	External Integration
8008	Delete LDAP Server(s)	External Integration
8009	Test LDAP Server	External Integration
8010	Test LDAP Connections	External Integration
8011	Add LDAP Group Access Mapping	External Integration
8012	Delete LDAP Group Access Mapping	External Integration
8013	Edit LDAP Group Access Mapping	External Integration
8014	Remove User Group from LDAP Group Access Mapping	External Integration
8015	Validate LDAP Group(s) for LDAP Configuration	External Integration

Table 22-6 (Cont.) Oracle Key Vault Audit Event IDs

Event ID	Operation	Category
8016	Validate LDAP User(s) for LDAP configuration	External Integration
8018	Delete LDAP User	External Integration
8019	Delete Stale LDAP User	External Integration
8020	Modify LDAP User Attributes	External Integration
8021	Download LDAP Log	External Integration
8100	Add Single Sign-On Configuration	External Integration
8101	Delete Single Sign-On Configuration	External Integration
8104	Edit Single Sign-On Configuration	External Integration
8105	Enable Single Sign-On Configuration	External Integration
8106	Disable Single Sign-On Configuration	External Integration
8200	Enable Audit Vault Integration	External Integration
8201	Disable Audit Vault Integration	External Integration
8202	Suspend Audit Vault Audit Record Collection	External Integration
8203	Resume Audit Vault Audit Record Collection	External Integration
8300	HSM Initialize	External Integration
8301	HSM Set Credential	External Integration
8302	HSM Create Bundle	External Integration
8303	HSM Apply Bundle	External Integration
8304	HSM Reverse Migrate	External Integration

Configuring Oracle Key Vault with Oracle Audit Vault

A user who has the Audit Manager role can configure Oracle Key Vault to send audit records to Oracle Audit Vault for centralized audit reporting and alerting.

- [Integrating Oracle Audit Vault with Oracle Key Vault](#)
You can perform this integration from the Oracle Key Vault management console.
- [Viewing Oracle Key Vault Audit Data Collected by Oracle Audit Vault](#)
You can use the Oracle Audit Vault Server console to view data that is collected by Oracle Key Vault and Oracle Audit Vault.
- [Suspending an Oracle Audit Vault Monitoring Operation](#)
You can suspend an Oracle Audit Vault monitoring operation from the Oracle Key Vault management console.
- [Resuming an Oracle Audit Vault Monitoring Operation](#)
You can resume a suspended Oracle Audit Vault monitoring operation from the Oracle Key Vault management console.

- [Deleting an Oracle Audit Vault Integration](#)
You can delete an Oracle Audit Vault integration by using the Oracle Key Vault management console.
- [Guidance for Integrating Oracle Audit Vault in a Multi-Master Cluster or Primary-Standby Environment](#)
You must follow special guidelines to integrate Oracle Audit Vault with Oracle Key Vault in a multi-master cluster or primary-standby environment.

Integrating Oracle Audit Vault with Oracle Key Vault

You can perform this integration from the Oracle Key Vault management console.

- [Step 1: Check the Environment](#)
Before you begin the integration, you should ensure that the required components are all in place.
- [Step 2: Configure Oracle Key Vault as a Registered Host and a Secured Target with Oracle Audit Vault](#)
A user who has the Audit Manager role must configure the Oracle Key Vault server as a secured target on the Oracle Audit Vault server.
- [Aborting an Oracle Audit Vault Integration](#)
You can abort an Oracle Audit Vault integration by using the Oracle Key Vault management console.

Step 1: Check the Environment

Before you begin the integration, you should ensure that the required components are all in place.

1. Ensure that Oracle Audit Vault is properly installed and configured.
This activity requires administrative access to the Oracle Audit Vault server in order to register Oracle Key Vault as a secured target.
2. Ensure that you have the credentials of the Oracle Audit Vault administrator in order to register Oracle Key Vault as a secured target in Oracle Audit Vault server. This user does not need to be a super administrator.
3. Enable `SSH` access to Oracle Audit Vault.
Log in to the Oracle Audit Vault Server console as a user who has the Super Administrator role. Select the **Settings** tab, then **System**. In the Configuration area, click **System Settings** and then **Web/SSH/SNMP**. Turn on **SSH Access**, select **IP addresses** and then enter only the IP addresses that you need, or select **All**. Click **Save**.
4. Verify if the user `support` can SSH in to the Oracle Audit Vault server. For more information, see [Step 2: Configure Oracle Key Vault as a Registered Host and a Secured Target with Oracle Audit Vault](#).

Step 2: Configure Oracle Key Vault as a Registered Host and a Secured Target with Oracle Audit Vault

A user who has the Audit Manager role must configure the Oracle Key Vault server as a secured target on the Oracle Audit Vault server.

In a multi-master cluster environment, perform these steps on each node. Each node will send the audit records that were generated from that node to the Oracle Audit Vault server.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Select the **System** tab, and then **Audit Vault Integration** from the left navigation bar.
If you also have the System Administrator role, then the navigation will be slightly different. Click **System**, then in the left navigation pane, select **Settings**. In the Monitoring and Alerts area, click **Audit Vault**.
3. In the Deployment pane, enter the following settings:
 - **Hostname:** Enter the host name or IP address of the Oracle Audit Vault server.
 - **Public Host Key:** Enter the public host key of the Oracle Audit Vault server by following the guidelines mentioned in the help text.
 - **Support User Password:** Enter the support user password of the Oracle Audit Vault server.
 - **Administrator Name:** Enter the user name of the Oracle Audit Vault server user who has the Administrator role.
 - **Administrator Password:** Enter the password of the Oracle Audit Vault server user who has the Administrator role.
 - **Recovery Passphrase:** Enter the recovery passphrase of the Oracle Key Vault server.

The screenshot shows the 'Deployment' configuration page in the Oracle Audit Vault management console. The page is divided into two main sections: 'Audit Vault Details' and 'Key Vault Details'. In the 'Audit Vault Details' section, there are five input fields: 'Hostname' (192.0.2.74), 'Public Host Key' (ssh_host_ed25519_key.pub), 'Support User Password' (masked with dots), 'Administrator Name' (ADMINUSER2), and 'Administrator Password' (masked with dots). A 'Clear' button is located next to the Public Host Key field. Below the 'Key Vault Details' section, there is a 'Recovery Passphrase' field (masked with dots) and a green 'Deploy' button.

4. Click **Deploy**.

The integration may take about 10 minutes to complete. Do not attempt to re-initiate the Audit Vault integration during this interval. The Oracle Key Vault server may become unavailable for some time until the integration completes.

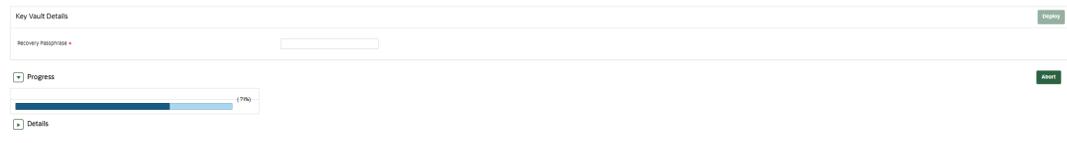
After the integration completes, a **Monitoring** tab appears and will show the Audit Vault agent status.

Aborting an Oracle Audit Vault Integration

You can abort an Oracle Audit Vault integration by using the Oracle Key Vault management console.

The Audit Vault integration with Oracle Key Vault may take about 10 minutes to complete. If the integration gets stuck Oracle Key Vault displays a **Server Error: 500**.

1. In the **Recovery Passphrase** field, under **Key Vault Details** pane, provide the recovery passphrase.
2. Under the **Progress** window pane, on the Audit Vault Details page, click **Abort** to abort the integration.



3. Perform the Integration again and start providing the information, see [Step 2: Configure Oracle Key Vault as a Registered Host and a Secured Target with Oracle Audit Vault](#).

The integration continues. Do not attempt to re-initiate the Audit Vault integration during this interval. The Oracle Key Vault server may become unavailable for some time until the integration completes.

After the integration completes, a **Monitoring** tab appears and will show the Audit Vault agent status.

Viewing Oracle Key Vault Audit Data Collected by Oracle Audit Vault

You can use the Oracle Audit Vault Server console to view data that is collected by Oracle Key Vault and Oracle Audit Vault.

1. Log in to the Oracle Audit Vault Server console as an auditor.
2. Select the **Reports** tab.
3. Select **Activity Reports**.
4. Select **All Activity**.
5. Filter the target to get all records belonging to `okv_db_Oracle_Key_Vault_IP_address`.

Assuming that you filtered for the target `okv_db_192.0.2.78`, the report could be similar to the following:

	Target	User	Client Host	Client Program	Event	Object	Event Status	Event Time ↓
	okv db 192.0.2.78	CLIENT1			GET ATTRIBUTES	1FC88514-A83C-4F87-BF61-82B0A4BC2246	SUCCESS	9/30/2021 7:17:37 AM
	okv db 192.0.2.78	CLIENT1			GET ATTRIBUTE LIST	1FC88514-A83C-4F87-BF61-82B0A4BC2246	SUCCESS	9/30/2021 7:17:37 AM
	okv db 192.0.2.78	CLIENT1			GET ATTRIBUTE LIST	1FC88514-A83C-4F87-BF61-82B0A4BC2246	SUCCESS	9/30/2021 7:17:37 AM

Suspending an Oracle Audit Vault Monitoring Operation

You can suspend an Oracle Audit Vault monitoring operation from the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Select the **System** tab, and then **Audit Vault Integration** in the left navigation bar.
If you also have the System Administrator role, then the navigation will be slightly different. Click **System**, then in the left navigation pane, select **Settings**. In the Monitoring and Alerts area, click **Audit Vault**.
3. Select the **Monitoring** tab.
4. In the Audit Vault pane, click **Suspend**.

Resuming an Oracle Audit Vault Monitoring Operation

You can resume a suspended Oracle Audit Vault monitoring operation from the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Select the **System** tab, and then **Audit Vault Integration** in the left navigation bar.
If you also have the System Administrator role, then the navigation will be slightly different. Click **System**, then in the left navigation pane, select **Settings**. In the Monitoring and Alerts area, click **Audit Vault**.
3. Select the **Monitoring** tab.
4. In the Audit Vault pane, click **Resume**.

Deleting an Oracle Audit Vault Integration

You can delete an Oracle Audit Vault integration by using the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a user who has the Audit Manager role.
2. Select the **System** tab, and then **Audit Vault Integration** in the left navigation bar.
If you also have the System Administrator role, then the navigation will be slightly different. Click **System**, then in the left navigation pane, select **Settings**. In the Monitoring and Alerts area, click **Audit Vault**.
3. Under Deployment, in the Audit Vault Details page, enter the following settings:
 - **Public Host Key:** Enter the public host key of the Oracle Audit Vault server by following the guidelines mentioned in the help text.
 - **Support User Password:** Enter the support user password of the Oracle Audit Vault server.
 - **Administrator Name:** Enter the user name of the Oracle Audit Vault server user who has the Administrator role.
 - **Administrator Password:** Enter the password of the Oracle Audit Vault server user who has the Administrator role.

Deployment
Delete

Audit Vault Details

Hostname *	192.0.2.74
Public Host Key *	<div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> <input style="width: 90%;" type="text" value="ssh_host_ed25519_key.pub"/> 🗑️ Clear </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 2px; font-family: monospace; font-size: 0.8em;"> ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKzLyLIUZ5F5Z RX2xqjdmTxPzrxikI79Z3ITcSbeDDcY </div>
Support User Password *	<input style="width: 90%;" type="password" value="....."/>
Administrator Name *	<input style="width: 90%;" type="text" value="ADMINUSERZ"/>
Administrator Password *	<input style="width: 90%;" type="password" value="....."/>

4. In the Deployment pane, click **Delete**.
5. Click **OK** to confirm.

Audit records that are already collected in Oracle Audit Vault are not affected when Oracle Key Vault integration with Oracle Audit Vault is deleted.

6. Because you no longer need to copy files from one server to another, disable SSH access to Oracle Audit Vault.

Log in to the Oracle Audit Vault Server console as a user who has the Super Administrator role. Select the **Settings** tab, then **System**. In the Configuration area, click **System Settings** and then **Web/SSH/SNMP**. Turn off **SSH Access**. Click **Save**.

Guidance for Integrating Oracle Audit Vault in a Multi-Master Cluster or Primary-Standby Environment

You must follow special guidelines to integrate Oracle Audit Vault with Oracle Key Vault in a multi-master cluster or primary-standby environment.

Multi-Master Cluster Environments

- If Oracle Key Vault is configured to use multi-master clusters, then you must perform the Oracle Audit Vault integration individually on each node. Each node will send audit records that are generated only on that node to Oracle Audit Vault irrespective of whether audit record replication is enabled.

Primary-Standby Environments

- Perform the integration only on the primary server, not the standby server.
- If you must perform a switchover operation, then note the following:
 - You must switch back to the primary server if you want to suspend, resume, or delete the integration. You do not need to perform additional steps.
 - To integrate the new primary server with Oracle Audit Vault, optionally, ensure that you use the same Oracle Audit Vault host name and administrator credentials that were used in the old primary server.
 - If you perform an unpair operation after performing a switchover operation, then you must perform a new Oracle Audit Vault integration with the new primary server.

- If you delete the new integration, then the old integration becomes non-functional. You must then delete this old integration by switching back to the old primary server.
- If a failover operation occurs and the original primary server is no longer available, then you must perform a new Oracle Audit Vault integration with the new primary server.
- However, if the original primary server is not lost and it is possible to bring back the original primary server as the new standby server, then you do not need to perform additional steps.

Using Oracle Key Vault Reports

Oracle Key Vault collects statistical information on a range of activities that impact Key Vault operations.

- [About Oracle Key Vault Reports](#)
The reports cover system activity, certificate expiration, keys, passwords, entitlement status, extraction status, and metadata.
- [Viewing Key Management Reports for Oracle Endpoints](#)
All users can view the key management reports for Oracle endpoints.
- [Viewing Keys and Wallets Reports](#)
The keys and wallets reports require different privileges for viewing, depending on the report.
- [Viewing Secrets Management Reports](#)
All users can view the secrets management reports.
- [Viewing SSH Reports](#)
All users can view the SSH reports.
- [Viewing Endpoint Reports](#)
You must have the System Administrator role or the Audit Manager role to view the five categories of endpoint reports.
- [Viewing User Reports](#)
You must have the System Administrator role, the Key Management role, or the Audit Manager role to view the four categories of user reports.
- [Viewing System Reports](#)
You must have the System Administrator role or the Audit Manager role to view the system reports.

About Oracle Key Vault Reports

The reports cover system activity, certificate expiration, keys, passwords, entitlement status, extraction status, and metadata.

Oracle Key Vault provides seven types of reports for endpoints, users, keys and wallets, SSH keys configuration and usage, and system. In a multi-master cluster, some reports contain additional information, such as the node ID, node name, and IP address.

The seven report types are as follows:

- Key management reports for Oracle endpoints, which includes information about TDE master encryption keys, GoldenGate master keys, and ACFS volume encryption key details
- Keys and wallets reports list the access privileges granted to all keys and wallets, and the details of TDE master encryption keys managed by Oracle Key Vault

- Secrets management reports for database passwords, secret data, and opaque objects
- SSH reports for SSH user key management, SSH server access management and detailed information on SSH user keys inventory, authorization, and usage.
- Endpoint reports contain details of all endpoint and endpoint group activity, certificate and password expiration, and access privileges
- User reports contain details of all user and user group activity, their certificate and password expiration, and access privileges
- System reports contain a history of system backups taken and scheduled, details of remote restoration points, and RESTful command-line interface usage

A user who has the Audit Manager role can view all reports, including reports that are accessible from the Audit Trail pages in the Oracle Key Vault management console. A user with the Key Administrator role can view user reports and keys and wallets reports. Users with the System Administrator role can view endpoint, user, and system reports.

Reports may include additional columns that are hidden by default. To include such columns in the displayed report, Click **Actions**, then **Select Columns**. The **Select Columns** dialog box appears. Select the columns that you want to include in the report from the list shown under **Do Not Display** section and move them to the list shown under **Display In Report** section by clicking **Move** button (shown as >). Likewise, you can also remove columns from the report.

To view the reports:

1. Log in to the Oracle Key Vault management console.
2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand the report type to see the corresponding reports.

Related Topics

- [Managing System Auditing](#)
Auditing entails tasks such as capturing audit records in a syslog file or downloading the audit records to a local file.

Viewing Key Management Reports for Oracle Endpoints

All users can view the key management reports for Oracle endpoints.

1. Log in to the Oracle Key Vault management console.
2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand **Key Management Reports for Oracle Endpoints Reports**.

The **Key Management Reports for Oracle Endpoints** page appears displaying the five types of reports.

Expand All Collapse All

▼ Key Management Reports for Oracle Endpoints

Name	Description
TDE Master Encryption Key Details Report	Creation, ownership, state and database-specific details of TDE master encryption keys
TDE Master Encryption Key Endpoint Usage Report	TDE master encryption key use by the endpoints
DB Generated TDE Master Encryption Key Attribute Report	Database-generated TDE master encryption key creation and activation attributes
GoldenGate Master Key Details Report	Creation, ownership, state and GoldenGate-specific details of GoldenGate master keys
ACFS Volume Encryption Key Details Report	Creation, ownership, state and ACFS-specific details of ACFS volume encryption keys

▶ Keys and Wallets Reports

▶ Secrets Management Reports

▶ Endpoint Reports

▶ User Reports

▶ System Reports

4. Click the report name to see the corresponding user report.

Viewing Keys and Wallets Reports

The keys and wallets reports require different privileges for viewing, depending on the report.

1. Log in to the Oracle Key Vault management console as a user who has the appropriate privileges.

Only a user who has the Key Administrator role or Audit Manager role can view the Wallet Entitlement Report. All users can view the remaining reports.

2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand **Keys and Wallets Reports**.

The **Keys and Wallets Reports** page appears displaying the reports.

Expand All Collapse All

▶ Key Management Reports for Oracle Endpoints

▼ Keys and Wallets Reports

Name	Description
Wallet Entitlement Report	Access privileges of all users, user groups, endpoints and endpoint groups on wallets and wallet items
Keys & Secrets Endpoint Usage Report	Details of key, secret data, opaque object, certificate and template use by the endpoints
Symmetric Key Details Report	Ownership, creation, state and wallet details of symmetric keys that are not TDE, GoldenGate or ACFS keys
Private Key Details Report	Ownership, creation, state and wallet details of all private keys
Public Key Details Report	Ownership, creation, state and wallet details of all public keys
Certificate Details Report	Ownership, creation, state and wallet details of all certificate objects
Certificate Request Details Report	Ownership, creation and wallet details of all certificate requests

4. Click the report name to see the corresponding report.

Viewing Secrets Management Reports

All users can view the secrets management reports.

1. Log in to the Oracle Key Vault management console.
2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand **Secrets Management Reports**.

The **Secrets Management Reports** page appears displaying the reports.

- Key Management Reports for Oracle Endpoints
- Keys and Wallets Reports
- Secrets Management Reports

Name	Description
DB Password Details Report	Ownership, creation, state and wallet details of all objects uploaded from secure external password store (SEPS) wallets
Secret Data Details Report	Ownership, creation, state and wallet details of all secret data but DB password objects
Opaque Object Details Report	Ownership, creation and wallet details of all opaque objects
- Endpoint Reports
- User Reports
- System Reports

4. Click the report name to see the corresponding report.

Viewing SSH Reports

All users can view the SSH reports.

1. Log in to the Oracle Key Vault management console.
2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand **SSH Reports**.

The **SSH Reports** page appears displaying the reports.

Expand All Collapse All

Key Management Reports for Oracle Endpoints

Keys and Wallets Reports

Secrets Management Reports

SSH Reports

SSH Key Details

Name	Description
SSH Key Metadata Report	Metadata for SSH public and private keys including owner, key length, expiration, and wallet membership
SSH Key Usage Report	All operations on SSH keys by endpoints and users

SSH Server Access Management

Name	Description
SSH Server Authorization Report	Users authorized to connect to SSH servers
SSH Server Access Report	Users attempting to connect to SSH servers
SSH Server Wallet Report	SSH server wallets determine the users authorized to connect to SSH servers

SSH User Private Key Management

Name	Description
SSH Private Key Authorization Report	Users and endpoints authorized to use SSH Private Keys
SSH Private Key Usage Report	Users and endpoints using SSH Private Keys

Endpoint Reports

User Reports

System Reports

4. Click the report name to see the corresponding report.

Viewing Endpoint Reports

You must have the System Administrator role or the Audit Manager role to view the five categories of endpoint reports.

Oracle Key Vault offers five endpoint reports: Endpoint Activity, Endpoint Entitlement, Endpoint Certificate Expiry, Endpoint Metadata, and Sign & Signature Verify Activity.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Audit Manager role.
2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand **Endpoint Reports**.

▶ SSH Reports

▼ Endpoint Reports

Name	Description
Activity Report	Statistics of all endpoint and endpoint group activity
Entitlement Report	Access privileges of all endpoints and endpoint groups on wallets and wallet items
Certificate Expiration Report	Certificate expiration state of endpoints
Metadata Report	Endpoint information and deployment configuration details
Sign & Signature Verify Activity Report	Endpoint Sign & Signature Verify activity details

▶ User Reports

4. Select the name of the report that you want to view.

For example, the Activity Report for endpoints appears similar to the following:

All Endpoint Activity Close

Go Actions

Endpoint Name	Operation	Object	Object Type	Time
HCM_DB_RAC_INST1	Get Attributes	889FF729-10AE-4FAD-BF93-96D28F90BA2C	KEY	07-DEC-2017 17:54:0
HCM_DB_RAC_INST1	Get	889FF729-10AE-4FAD-BF93-96D28F90BA2C	KEY	07-DEC-2017 17:53:51
HCM_DB_RAC_INST1	Get Attributes	889FF729-10AE-4FAD-BF93-96D28F90BA2C	KEY	06-DEC-2017 17:53:51
HCM_DB_RAC_INST1	Get	889FF729-10AE-4FAD-BF93-96D28F90BA2C	KEY	06-DEC-2017 17:53:51
HCM_DB_RAC_INST1	Get Attributes	889FF729-10AE-4FAD-BF93-96D28F90BA2C	KEY	05-DEC-2017 17:53:51
HCM_DB_RAC_INST1	Get	889FF729-10AE-4FAD-BF93-96D28F90BA2C	KEY	05-DEC-2017 17:53:4
HCM_DB_RAC_INST1	Get Attributes	889FF729-10AE-4FAD-BF93-96D28F90BA2C	KEY	04-DEC-2017 17:53:51

1 - 200 of 624

Specific Endpoint Activity

Select Endpoint: ACME_UNITED_MODE

Operation	Count	Percentage
Locate	4	(67%)
Version	1	(17%)
Store Endpoint Metadata	1	(17%)

row(s) 1 - 3 of 3

Specific Endpoint Group Activity

Select Endpoint Group: CLOUD_APP_GROUP

Operation	Count	Percentage
Register	2	(67%)
Store Endpoint Metadata	1	(33%)

row(s) 1 - 2 of 2

Viewing User Reports

You must have the System Administrator role, the Key Management role, or the Audit Manager role to view the four categories of user reports.

Oracle Key Vault offers four user reports: User Activity, User Entitlement, User Expiry, and User Failed Login.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role, the Key Management role, or the Audit Manager role.
2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand **User Reports** to see user-specific reports.

The **User Reports** page appears.

- Key Management Reports for Oracle Endpoints
- Keys and Wallets Reports
- Secrets Management Reports
- Endpoint Reports
- User Reports

Name	Description
Activity Report	Statistics of all user and user group activity
Entitlement Report	Access privileges of all users and user groups on wallets and wallet items
Account Report	Password expiration and account status of the regular Oracle Key Vault users
Failed Login Report	Summary of all failed login attempts to Oracle Key Vault
- System Reports

4. Click the report name to see the corresponding user report.

Viewing System Reports

You must have the System Administrator role or the Audit Manager role to view the system reports.

Oracle Key Vault offers three system reports: Backup History, Notification, and RESTful Services Usage.

1. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the Audit Manager role.
2. Click the **Reports** tab and then **Reports** from left navigation bar.
3. Expand **System Reports**.

The **System Reports** page appears displaying the system reports available.

[Expand All](#) [Collapse All](#)

Key Management Reports for Oracle Endpoints

Keys and Wallets Reports

Secrets Management Reports

Endpoint Reports

User Reports

System Reports

Name	Description
Cluster Report	Cluster node and service operation details
Conflict Resolution Report	Resolved name conflicts for wallets, endpoints, endpoint groups, users and user groups
Backup History Report	History of the processed system backups
Notification Report	Alert messages and notifications sent over email
RESTful Services Usage Report	Summary of the operations performed using RESTful Service

4. Click the report type to see the corresponding system report.

 **Note:**

For a multi-master cluster configuration, additional reports are available to monitor naming conflicts in the cluster. To view the **Conflict Resolution** reports, click the **Cluster** tab, and then **Conflict Resolution** from the left navigation bar.

Managing an Oracle Key Vault Primary-Standby Configuration

You can deploy Oracle Key Vault in a primary-standby server configuration.

- [Overview of the Oracle Key Vault Primary-Standby Configuration](#)
The Oracle Key Vault primary-standby configuration provides benefits based on the type of deployment your site needs.
- [Configuring the Primary-Standby Environment](#)
To configure a primary-standby environment, you must have the System Administrator role with access to the two servers (one primary and one standby).
- [Switching the Primary and Standby Servers](#)
You can switch the roles of the primary and standby server for situations such as maintenance periods.
- [Restoring Primary-Standby After a Failover](#)
A failover takes place if the primary server fails.
- [Disabling or Unpairing the Primary-Standby Configuration](#)
You can disable the primary-standby configuration by unpairing the primary and standby servers.
- [Read-Only Restricted Mode in a Primary-Standby Configuration](#)
The read-only restricted mode is the default mode in a primary-standby configuration.
- [Best Practices for Using Oracle Key Vault in a Primary-Standby Configuration](#)
Oracle provides guidelines for ensuring operational continuity and minimal downtime of Oracle Key Vault.

Overview of the Oracle Key Vault Primary-Standby Configuration

The Oracle Key Vault primary-standby configuration provides benefits based on the type of deployment your site needs.

- [About the Oracle Key Vault Primary-Standby Configuration](#)
You configure a primary-standby environment by providing the primary and standby servers with each other's IP address and certificate, and then pairing them.
- [Benefits of an Oracle Key Vault Primary-Standby Configuration](#)
The benefits of an Oracle Key Vault primary-standby configuration include high availability necessary for business-critical operations.
- [Difference Between Primary-Standby Configuration and Multi-Master Cluster](#)
In both primary-standby and multi-master cluster configurations, one server will always operate in read/write mode.
- [Primary Server Role in a Primary-Standby Configuration](#)
A primary-standby deployment consists of two Oracle Key Vault servers operating in a primary-standby configuration.

- [Standby Server Role in a Primary-Standby Configuration](#)
In a primary-standby environment, one server runs in the standby server role.

About the Oracle Key Vault Primary-Standby Configuration

You configure a primary-standby environment by providing the primary and standby servers with each other's IP address and certificate, and then pairing them.

While pairing the primary and standby servers, you can select one as the primary server, and the other as the standby. A failover timeout that you set determines when the standby starts to take over as the primary server.

Note:

Oracle recommends that you keep the primary and standby systems as identical as possible, because their roles can be reversed in maintenance periods and failure situations. These include the following:

- Oracle Key Vault software versions
- Disk size
- RAM size
- System clocks on both systems must be synchronized

If your deployment requires a primary-standby configuration, then Oracle recommends that you configure it *before* adding endpoints to Oracle Key Vault. This enables the endpoints to know about both the primary and standby servers. An endpoint that is added before the standby server configuration will not know about the standby server, unless you reenroll the endpoint. If you configure the primary-standby environment after adding endpoints, then you must reenroll the endpoints to ensure the endpoints recognize both servers that were previously enrolled with the primary and standby servers in standalone mode.

WARNING:

Configure primary-standby deployments before adding endpoints to ensure that the endpoints know about both nodes.

If you want to add Simple Network Management Protocol (SNMP) support in a primary-standby environment, then ideally, configure SNMP on both the primary and the standby servers before pairing them. This is because the standby server is no longer accessible from the Oracle Key Vault management console, because all requests are forwarded to the primary server. However you can also add SNMP support to the standby after pairing the servers by accessing the standby using Secure Shell (SSH).

If you want to use a third-party certificate in a primary-standby configuration, then you must install it on the primary and standby servers first, and then pair them.

If you want to enable FIPS mode in a primary-standby environment, then you must ensure that both the primary and standby servers use the same FIPS mode: either both are enabled, or both are disabled for FIPS mode. This is because the standby server is no longer accessible from the Oracle Key Vault management console, because all requests are forwarded to the primary server.

With persistent cache enabled, both the primary and the standby servers will cache the master encryption keys from Oracle Key Vault independently. Ensure that TDE operations have run on the primary and standby servers after these servers have started to verify the persistent cache. The persistent cache feature also enables endpoints to be operational during primary-standby operations, such as configuration, switchovers, and failovers.

If enabled, read-only restricted mode ensures endpoint operational continuity (such as enabling the endpoints to fetch keys) if either the standby or primary server is not available. For example, if the standby shuts down, then the primary will go into read-only restricted mode and enable the endpoints to fetch keys and continue operations.

A primary-standby configuration is characterized by continuous synchronization between the primary server and the standby server. When synchronization is lost between the primary and standby servers, it is possible to encounter a split-brain scenario where two primary servers might be active simultaneously. In such a scenario, both servers record new data that diverges from the last synchronized state. When connectivity is restored between the primary and standby servers, it may not be possible to reconcile the changes on the two servers and data loss may occur.

You can enable or disable restricted mode when configuring the primary-standby environment by selecting the **Allow Read-Only Restricted Mode** option to **Yes** or **No** on the Configure Primary-Standby page.

When read-only restricted mode is enabled, the primary server enters read-only restricted mode if the standby server is unavailable. In read-only restricted mode, the primary server allows keys to be retrieved, but does not allow keys to be modified or new keys to be added. This ensures that endpoints still have access to their keys, and key data or metadata is not lost due to a split-brain scenario. However, the primary server still writes audit records, which may be lost if a split-brain scenario occurs with the standby server.

When read-only restricted mode is disabled, the primary server becomes unavailable and stops accepting new requests if the standby server is unavailable. Endpoints connected to Oracle Key Vault will be unable to retrieve keys from the server until connectivity is restored between primary and standby servers. You can use the persistent master encryption key cache feature to avoid endpoint downtime. With this feature, data integrity is ensured by allowing endpoints to communicate with one primary server at any given time. This avoids split-brain situations, and the risk of data loss associated with such situations.

Related Topics

- [Changing SNMP Settings on the Standby Server](#)
You change the SNMP settings from the command line on the standby server.

Benefits of an Oracle Key Vault Primary-Standby Configuration

The benefits of an Oracle Key Vault primary-standby configuration include high availability necessary for business-critical operations.

Users performing business-critical operations must have data to be accessible and recoverable with minimum downtime. These requirements are met in a primary-standby configuration.

You achieve high availability by adding redundancy in the form of a standby server that can take over the functions of the primary server in case of failure. The standby server helps you eliminate single points of failure and reduce server downtime. This is a significant reason to deploy Oracle Key Vault in a primary-standby configuration. In a classic primary-standby configuration, the emphasis is on key preservation. In a multi-master cluster, emphasis is on both key preservation and availability of the keys.

You can create a cluster of Oracle Key Vault server nodes for greater availability and redundancy. A primary-standby configuration is limited to two servers, whereas a multi-master cluster can have up to 16 geographically distributed nodes. The primary-standby configuration and the multi-master configuration are mutually exclusive.

Related Topics

- [Oracle Key Vault Multi-Master Cluster Overview](#)
The multi-master cluster nodes provide high availability, disaster recovery, load distribution, and geographic distribution to an Oracle Key Vault environment.

Difference Between Primary-Standby Configuration and Multi-Master Cluster

In both primary-standby and multi-master cluster configurations, one server will always operate in read/write mode.

In a primary-standby configuration, when both servers are available, one of the servers operates in read/write mode in the primary server role, and the other operates in the standby server role. The endpoints only connect to the server running in the primary server role. The roles can be switched manually to support maintenance operations, or automatically due to a server or connectivity failure. If either the primary or standby server becomes unavailable, then the remaining server operates in a read-only restricted mode, limiting normal updates while allowing audits and other internal updates.

In a multi-master cluster, the endpoints can connect to any Oracle Key Vault server. Some servers are configured as bi-directional read/write pairs in which information updated in either node must be successfully replicated to the other node immediately. If one of the nodes in a read/write pair becomes unavailable, the surviving node operates in read-only restricted mode until the other node is restored and synchronization resumes. A fully functional multi-master cluster must have at least one read/write pair.

When a successful update occurs in a read/write pair, the update is propagated to all other nodes in the cluster.

A primary-standby configuration and a multi-master cluster configuration are mutually exclusive and incompatible configurations. The specific configuration of an Oracle Key Vault deployment has no ramification on the endpoint side configuration.

Primary Server Role in a Primary-Standby Configuration

A primary-standby deployment consists of two Oracle Key Vault servers operating in a primary-standby configuration.

By default, endpoints only connect to the primary server until it becomes unavailable. At any time, only one server operates in the primary server role and that server actively accepts client connections. The other server operates in the standby server role, which receives updates from the primary server. On failure of the server running in the primary server role, the standby assumes the primary role. There may be restrictions in operations if the primary-standby pair is not fully available and operational.

Standby Server Role in a Primary-Standby Configuration

In a primary-standby environment, one server runs in the standby server role.

This standby server does not accept client connections while in that role. The server receives updates only from the paired server running in primary server role. If the primary server is no

longer available, including being available to the administrator, then the server running in the standby role switches to assume the primary server role. There may be restrictions in operations if the primary-standby pair is not fully available and operational.

Configuring the Primary-Standby Environment

To configure a primary-standby environment, you must have the System Administrator role with access to the two servers (one primary and one standby).

- **Step 1: Configure the Primary Server**
To configure the primary server, you must enable it to connect to the standby server. Ensure that the system time of primary and standby servers are in sync. Oracle recommends that you set up Network Time Protocol (NTP) before configuring the primary and standby servers.
- **Step 2: Configure the Standby Server**
To configure the standby server, you must enable it to connect to the primary server.
- **Step 3: Complete the Configuration on the Primary Server**
After you configure the primary and standby servers, you can enable the primary-standby on the designated primary server.

Step 1: Configure the Primary Server

To configure the primary server, you must enable it to connect to the standby server. Ensure that the system time of primary and standby servers are in sync. Oracle recommends that you set up Network Time Protocol (NTP) before configuring the primary and standby servers.

If you plan to configure an Hardware Security Module (HSM) (such as Thales, Entrust, Utimaco, or other HSM with Oracle Key Vault), then you must first enable this HSM in Oracle Key Vault before configuring the primary server.

1. Open a web browser and enter the IP address of the designated primary server.
The Oracle Key Vault Management Console login screen is displayed.
2. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
3. Check if the server has FIPS mode enabled, and if necessary, enable or disable it.

You must ensure that both the primary and standby servers use the same FIPS mode setting: either both are enabled, or both are disabled, for FIPS mode. Changing the FIPS mode setting requires a restart of Oracle Key Vault.

- a. Select the **System** tab, and then select **Settings** in the left navigation bar.
- b. In the System Configuration section, click **FIPS**.

Note:

If you are logged in as a user with the Monitor privilege, then you can only view the pages in read-only mode.

- a. In FIPS Mode dialog box, either select or deselect the **Enable** check box, depending on whether both servers will use FIPS mode.
- b. Click **Save**.

Oracle Key Vault will restart.

4. If you changed the FIPS mode, then log back into the designated primary server as a user who has the System Administrator role.
5. Select the **System** tab, and then **Settings** in the left navigation bar.
6. In the System Configuration area, click **Primary-Standby**.

The Primary-Standby page appears. The following are the fields on this page:

- **Current status:** Displays the IP address and status of the current server.
- **Fast Start Failover Threshold (in secs):** Displays the duration (in seconds) that will elapse before the server takes over from a failed peer server. The default is 60 seconds.
To avoid fail over during brief or intermittent failures, increase the duration.
- **Configure this server as:** Displays whether the server is configured as a **Primary server** or **Standby server**.
- **Allow Read-Only Restricted Mode:** Displays the status of read-only restricted mode. The default is **Yes**.

When enabled, read-only restricted mode ensures operational continuity of the endpoints if the primary or standby Oracle Key Vault server is affected by server, hardware, or network failures

- **FIPS Mode:** Displays the current FIPS mode status of the server.
 - **Current Server Certificate:** Displays the server certificate.
7. Copy the following information, and then store it in a text file named `primary.txt`.
You will need this information when you configure standby server.
 - From the **Current status** field, copy the IP address and paste it in `primary.txt`.
 - From the **Current Server Certificate** field, copy the server certificate and paste it on a new line in `primary.txt` after the IP address.

Save `primary.txt`.

Next, you are ready to configure the standby server.

Related Topics

- *Oracle Key Vault Root of Trust HSM Configuration Guide*

Step 2: Configure the Standby Server

To configure the standby server, you must enable it to connect to the primary server.

If you plan to configure an HSM (such as SafeNet or Utimaco) with Oracle Key Vault, then you must first enable this HSM in Oracle Key Vault before configuring the standby server.

1. Open a web browser and enter the IP address of the designated standby server.
The Oracle Key Vault Management Console login screen is displayed.
2. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
3. Check if the server has FIPS mode enabled, and if necessary, enable or disable it.

You must ensure that both the primary and standby servers use the same FIPS mode setting: either both are enabled, or both are disabled, for FIPS mode. Changing the FIPS mode setting requires a restart of Oracle Key Vault.

- a. Select the **System** tab, and then **Settings** in the left navigation bar.
- b. In the System Configuration area, click **Primary-Standby**.
- c. Either select or clear the **Enable** check box, depending on whether both servers will use FIPS mode.
- d. Click **Save**.

Oracle Key Vault will restart.

4. If you changed the FIPS mode, then log back into the designated standby server as a user who has the System Administrator role.
5. Select the **System** tab, and then **Settings** in the left navigation bar.
6. In the System Configuration area, click **Primary-Standby**.
7. Copy the following information, and store it in a text file named `standby.txt`.

You will need this information when you configure the primary server.

- From the **Current status** field, copy the IP address and paste it in `standby.txt`.
- From the **Current Server Certificate** field, copy the server certificate and paste it on a new line in `standby.txt` after the IP address.

Save `standby.txt`.

8. In the **Configure this server as** field, select **Standby server**.

The **Primary server IP address** and **Primary server certificate** fields are displayed.

Ensure that **Yes** is selected in the **Allow Read-Only Restricted Mode** field.

Do not disable read only restricted mode unless necessary. If the primary-standby configuration is configured with read only restricted mode disabled, then you must enable it by reinstalling and configuring Oracle Key Vault again.

9. Copy the following information from `primary.txt`, and paste it in the **Configure Primary-Standby** page of the standby server:
 - Copy the IP address and paste it in the **Primary server IP address** field.
 - Copy the server certificate and paste it in the **Primary server certificate** field.

10. Click **Save**.

The **Settings Saved** page is displayed.

The **Reset** button enables you to delete the primary-standby configuration, if necessary.

11. Do not exit the management console.

At this stage, the primary-standby configuration is complete on the designated standby server. The next step is to enable primary-standby on the designated primary server.

Related Topics

- [Oracle Key Vault Root of Trust HSM Configuration Guide](#)Configuring HSM

Step 3: Complete the Configuration on the Primary Server

After you configure the primary and standby servers, you can enable the primary-standby on the designated primary server.

1. Ensure that you are logged in to the standby server as a user with the System Administrator Role and that the Oracle Key Vault management console Configure Primary-Standby page is displayed.

2. On the **Settings Saved** page, click the IP address of the primary server displayed at the top of the page.

The **Oracle Key Vault Management Console login screen** of the primary server is displayed.

3. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.

4. Select the **System** tab, and then **Settings** in the left navigation bar.

5. In the System Configuration area, click **Primary-Standby**.

6. In the **Configure this server as** field, select **Primary server**.

The **Standby server IP address** and **Standby server certificate** fields are displayed.

Ensure that **Yes** is selected in the **Allow Read-Only Restricted Mode** field.

Do not disable read-only restricted mode unless necessary. If the primary-standby configuration is configured with read-only restricted mode disabled, then you must enable it by reinstalling and configuring Oracle Key Vault again.

7. Copy the following information from `standby.txt`, and paste it in the Configure Primary-Standby page of the primary server:

- Copy the IP address and paste it in the **Standby server IP address** field.
- Copy the server certificate and paste it in the **Standby server certificate** field.

8. Click **Initiate Pairing**.

9. In the confirmation message that is displayed, click **OK**. The **Operation in Progress** page is displayed.

 **Caution:**

Allow at least 10 minutes to elapse before performing the next step.

10. After at least 10 minutes have elapsed, click **Refresh**.

If the pairing of primary and standby servers is successful, then the current session is terminated. The **Oracle Key Vault Management Console login screen** of the primary server is displayed. The primary-standby configuration is now complete.

11. Check that the configuration was successful.

- a. Log in as the System Administrator.
- b. Select the **System** tab, and then **Settings** in the left navigation bar.
- c. In the System Configuration area, click **Primary-Standby**.

The Primary-Standby Status page appears.

- d. Ensure that the **Status** label is set to **Primary-Standby mode is enabled**.
- e. Ensure that the **Switchover Status** is correct. In this example, the status is correctly set to **TO STANDBY**.

At this stage, the primary-standby configuration should be complete and ready to use. Note the following:

- You cannot log in to the standby server using a web browser because all configuration is propagated from the primary.
- With the persistent cache enabled, endpoints will continue to operate while the primary-standby configuration is enabled. The **IP Address**, **Network Mask**, and **Gateway** fields in the Network Info page (found from selecting the **System** tab, and then **Settings** in the left navigation bar) will no longer be modifiable.
- To manage the primary-standby deployment, log in to the primary server using a web browser.

 **Caution:**

Ensure that you leave read-only restricted mode enabled while configuring primary-standby. Enabling it later requires a reinstall of the Oracle Key Vault server software on the standby server.

After configuring the primary-standby environment, do not change the system time on the primary server. The changed system time causes the standby server to go down, thus disrupting the functioning of the primary-standby configuration.

Switching the Primary and Standby Servers

You can switch the roles of the primary and standby server for situations such as maintenance periods.

During such maintenance periods, you might want to shut down a server to upgrade software or install patches. If you have persistent cache enabled and the persistent cache timeout is sufficiently tuned, then the endpoints will continue to be operational during the switchover, minimizing endpoint downtime.

1. Log in to the Oracle Key Vault management console of the primary node as a user with the System Administrator role.
2. Before switching the primary and standby servers, ensure that there are no primary-standby related alerts on the **Alerts** page.

To access the **Alerts** page, click the **Reports** tab, and then click **Alerts** in the left pane. Ensure that all primary-standby related alerts on the **Alerts** page are addressed before switching the primary and standby servers.

3. Select the **System** tab, and then **Settings** in the left navigation bar.
4. In the System Configuration area, click **Primary-Standby**.

The **Primary-Standby Status** page appears.

5. Click **Switch Roles** on the top right.

The **Switch Roles** button allows you to switch the roles of the primary server and the standby server. The primary server then assumes the role of the standby server, while the standby server assumes the role of the new primary server.

Click **OK** in the confirmation message.

An operation-initiated message is followed by the **Operation in Progress** page indicating that the switchover operation will take 10 minutes to complete successfully.

 **Caution:**

You must wait for a minimum period of 10 minutes for the switchover operation to complete successfully. If you refresh the UI before the switchover operation is complete, an error message is displayed. The error message is displayed until the switchover is completed successfully.

6. Ensure that at least 10 minutes have elapsed, and only then, click **Refresh**.

This logs you out of the current session and then opens a login page to the switched primary server. Otherwise, try accessing the new primary server's URL directly.

Both the primary and standby servers are restarted. However, you will only be able to log in to the new primary node's web console. The primary server is the active server, and all requests to the standby will be forwarded to the primary.

7. Log in to the primary server to see the IP address of the switched standby node.
8. Select the **System** tab, then **Settings** in the left navigation bar, and then click Primary-Standby in the System Configuration area.

The Primary-Standby Status page appears. The **Standby server IP address** field displays the IP address.

Restoring Primary-Standby After a Failover

A failover takes place if the primary server fails.

If the primary server is not available, then the standby server takes over the primary role. If the standby server does not hear from the primary server for a time exceeding the **Fast Start Failover Threshold** value, then it will assume that the primary is shut down and start the failover process. You can configure the value in the **Fast Start Failover Threshold** field from the Oracle Key Vault management console from the default of 60 seconds. If the failed server (the old primary) becomes available again, then in most cases it will automatically become the new standby server. If the primary server fails permanently, then the standby server will take over as the primary. In this case, you must restore the primary-standby configuration.

1. Reinstall the Oracle Key Vault image on the failed server.

Ensure that you use the original IP address for the failed server.

2. Log on to the newly installed server and follow the steps to configure the primary-standby environment.

You can designate the new server as the standby server (because the cluster has a functional primary) and then pair it with the functioning primary.

3. If you want to restore the original configuration and set the new server as the primary, then click the **Switch Roles** option after you successfully pair the two nodes and enable primary-standby.

The **Switch Roles** button enables you to switch the roles of the primary server and the standby server. The primary server then assumes the role of the standby server, while the standby server assumes the role of the new primary server.

 **Note:**

When read-only restricted mode is disabled, the primary server's failover status goes into suspended state causing the standby server to wait indefinitely for the primary server to come back up. This is expected behavior to avoid a split-brain scenario where two primary servers are simultaneously active.

When read-only restricted mode is enabled, a primary or standby server failure causes the operational peer to enter read-only restricted mode, thus ensuring endpoint operational continuity.

Related Topics

•

Disabling or Unpairing the Primary-Standby Configuration

You can disable the primary-standby configuration by unpairing the primary and standby servers.

After the two servers are unpaired, the primary and standby servers will operate in standalone mode. To prevent endpoints from connecting to the old standby (now standalone) Oracle Key Vault server, you must take the old standby off the network. See *Oracle Key Vault Release Notes* for guidance about setting the permissions of the `/var/lib/oracle/diag/rdbms/dbfwdb/dbfwdb/metadata_pv` directory beforehand. Check the *Release Notes* for additional issues related to unpair operations.

1. Log in to the primary server's management console as a user with System Administrator privileges.
2. Select the **System** tab, and then **Settings** in the left navigation bar.
3. In the System Configuration area, click **Primary-Standby**.

The Primary-Standby Status page appears with **Unpair** and **Switch Roles** on the top right. The **Unpair** and **Switch Roles** options do the following:

- The **Unpair** button allows you to disconnect the primary server from the standby server, if required.
- The **Switch Roles** button allows you to switch the roles of the primary server and the standby server, if required. The primary server then assumes the role of the standby server, while the standby server assumes the role of the new primary server.

4. Click **Unpair**.

A brief message with a green check is displayed indicating that the operation has been successfully initiated.

The Operation in Progress page is displayed, indicating a wait time of at least 10 minutes for the un-pairing to complete.

Wait 10 minutes.

5. After 10 minutes, click the **Refresh** button to be logged out of the current session.
6. Log back in to the management console of the primary server.
7. Select the **System** tab, then **Settings** in the left navigation bar, and then click **Primary-Standby** in the System Configuration area.

The Configure Primary-Standby page appears. The **Current status** field shows the server in standalone mode.

Caution:

If you want to use the old standby (now standalone) Oracle Key Vault server as a standby in a new primary-standby deployment, or as part of a multi-master cluster, then you must re-install the Oracle Key Vault software

Related Topics

- [Oracle Key Vault Release Notes](#)

Read-Only Restricted Mode in a Primary-Standby Configuration

The read-only restricted mode is the default mode in a primary-standby configuration.

- [About Read-Only Restricted Mode in a Primary-Standby Configuration](#)
Primary-standby read-only restricted mode ensures endpoint operational continuity.
- [Primary-Standby with Read-Only Restricted Mode](#)
Read-only restricted mode is the default primary-standby mode in Oracle Key Vault.
- [Primary-Standby without Read-Only Restricted Mode](#)
When a primary-standby environment is configured without read-only restricted mode, the impact on endpoint operations differs.
- [States of Read-Only Restricted Mode](#)
A server using read-only restricted mode is affected by the failure in a primary server, a standby server, and the network.
- [Enabling Read-Only Restricted Mode](#)
Read-only restricted mode is enabled by default when primary-standby is configured.
- [Disabling Read-Only Restricted Mode](#)
Read-only restricted mode is enabled by default when primary-standby is configured.
- [Recovering from Read-Only Restricted Mode](#)
To recover an instance from read-only restricted mode after a network failure or standby server failure, manual intervention may be required.
- [Read-Only Restricted Mode Notifications](#)
When the primary or standby server enters read-only restricted mode, Oracle Key Vault generates an alert.

About Read-Only Restricted Mode in a Primary-Standby Configuration

Primary-standby read-only restricted mode ensures endpoint operational continuity.

This endpoint operational continuity is essential when the primary or standby Oracle Key Vault servers are affected by server, hardware, or network failures.

When an unplanned shutdown makes the primary or standby server offline, the endpoints can still connect to the surviving peer server to perform critical operations. Primary-standby read-only restricted mode ensures that operations that replicate data are blocked. Operations that replicate data are allowed when both primary and standby servers are back online, thus ensuring that no critical data is lost.

In a primary-standby Oracle Key Vault configuration, the single point of failure is eliminated when you replicate the primary server's data to the standby server. Read-only restricted mode enables the generation of non-critical data such as audit records. However, generation of critical data such as keys is disabled. When the primary server is down, operations that generate new critical data on the standby are disabled. The reverse is also true. When the standby server is down, operations that attempt to modify or create any data on the primary server are disabled.

In a primary-standby deployment without read-only restricted mode, most endpoint operations are blocked because endpoint operations generate audit records, which is data that needs replication, thus disrupting operational continuity.

The following are the benefits of using read-only restricted mode:

- Enables endpoint operational continuity when the primary or standby server is offline
- Ensures symmetrical behavior when the primary or standby server is offline

Primary-Standby with Read-Only Restricted Mode

Read-only restricted mode is the default primary-standby mode in Oracle Key Vault.

Note:

You can disable read-only restricted mode during the primary-standby configuration. Oracle recommends that you configure primary-standby with read-only restricted mode enabled, which is the default mode. While configuring primary-standby, ensure that **Yes** is selected in the **Allow Read-Only Restricted Mode** field on the Configure Primary-Standby page.

Read-only restricted mode ensures endpoint operational continuity as well as symmetrical behavior when the primary or standby server is offline. Symmetrical behavior ensures that the online server seamlessly takes over from its failed peer, and continues to service the endpoints without any disruption.

In read-only restricted mode, the surviving Oracle Key Vault server operates with limited functionality. Endpoint operations that add or modify critical data on the Oracle Key Vault server are blocked. However, endpoint operations that involve fetching of data are allowed. This ensures endpoint operational continuity and data integrity. For more information about blocked and allowed operations, see [About the States of Read-Only Restricted Mode](#).

For more information about read-only restricted mode, see [States of Read-Only Restricted Mode](#).

Note:

Read-only restricted mode has no impact on a standalone server.

Primary-Standby without Read-Only Restricted Mode

When a primary-standby environment is configured without read-only restricted mode, the impact on endpoint operations differs.

This impact depends on the type of failure encountered: primary failure, standby failure, or a network failure that prevents communication between the primary and standby servers. The following are the possible scenarios:

- **Primary server failure:** The standby server will fail over and take over from the affected primary server. This allows the Oracle Key Vault service to remain operational. Data modifications are stored on the primary server until they can be replicated to the standby server. This ensures endpoint operational continuity when the primary server goes offline due to an unplanned shutdown.
- **Standby server failure:** The primary server is unavailable to the endpoints, because it is not possible to distinguish a standby server failure from a network failure that prevents communication between the primary and standby servers.
- **Power loss or network connectivity failure:** The primary and standby servers are unable to communicate. The standby server will fail over and take over from the primary server. To avoid a split-brain scenario, only one of the servers is allowed to service the endpoints.

 **Note:**

A split-brain scenario in Oracle Key Vault occurs when the primary server fails, causing the standby server to fail over and take over from the primary server. This causes a situation where the primary and standby servers are available to service the endpoints, and create new data. A split-brain scenario causes data on the primary and standby servers to go out of sync. This can lead to data loss and corruption, as well as loss of operational continuity. To avoid a split-brain scenario, only one of the servers is allowed to service the endpoints after a fail over occurs.

In primary-standby without read-only restricted mode, one of the following situations is triggered when a failure occurs:

- Endpoints suffer a temporary operational disruption to avoid a split-brain scenario.
- The standby server accepts new requests and generates new data without attempting to synchronize the data with the failed primary server. Replication of data is temporarily disabled until the primary server is online, thus ensuring operational continuity.

States of Read-Only Restricted Mode

A server using read-only restricted mode is affected by the failure in a primary server, a standby server, and the network.

- [About the States of Read-Only Restricted Mode](#)
Read-only restricted mode puts the Oracle Key Vault instance into the read-only restricted mode state.
- [Read-Only Restricted State Functionality During a Primary Server Failure](#)
You can set a failover threshold value to determine when a standby server takes over for a failed primary server.
- [Read-Only Restricted Mode Functionality During a Standby Server Failure](#)
If a standby fails, the primary server waits for the duration in the **Fast Start Failover Threshold** field on the **Configure Primary-Standby** page.

- [Read-Only Restricted State Functionality During a Network Failure](#)
When a network failure affects communication between primary and standby servers, communication between certain endpoints and the primary server may also be affected.

About the States of Read-Only Restricted Mode

Read-only restricted mode puts the Oracle Key Vault instance into the read-only restricted mode state.

However, read-only restricted mode does not put the embedded Oracle Key Vault database into the read-only restricted mode state. In read-only restricted mode, the following behavior occurs when a primary or a standby server is unavailable:

- When the primary server is down, data cannot be replicated and so the standby server will fail over and disable all operations that generate new data. However, the standby can fetch existing data.
- When the standby server is down, data cannot be replicated and so the primary server disables all operations that generate new data. However, the primary can fetch existing data.

Read-only restricted mode introduces the following deviations from normal functionality:

- All operations that generate new data are blocked. Operations that fetch existing data are allowed. Audit records for endpoint operations are generated as in normal operation. Internal system operations of the Oracle Key Vault database are not impacted. Functionality such as alerts continue to work normally.
- Endpoints are allowed to fetch keys from the Oracle Key Vault server. Endpoints cannot create new keys or modify existing keys.
- Administrators can log in to the Oracle Key Vault management console. Creation of an endpoint or a wallet, deletion of keys, and operations that modify or delete data are blocked.
- Unpairing of primary and standby Oracle Key Vault servers running in read-only restricted mode are allowed.
- Backup operations are blocked to avoid data mismatches between backups.

Table 23-1 Allowed and Blocked Operations in Read-Only Restricted Mode

Operation	Allowed or Blocked
Log in to Oracle Key Vault	Allowed
Endpoint operations such as fetching keys from the cache	Allowed
Endpoint operations that add, modify, or delete data such as rotation of keys on the database	Blocked
System operations such as enabling SSH access	Allowed
System operations that write data such as setting up a REST server and creating virtual wallets	Blocked
Oracle Key Vault management console access	Allowed
All Administrator and endpoint operations that add new data or modify existing data	Blocked
Backup operations	Blocked

In read-only restricted mode, if you attempt to run the operations that generate new data or modify existing data on the Oracle Key Vault server, the `Key Vault Server in read-only restricted Mode` error is displayed.

If you attempt to upload a wallet to the Java keystore, then you are prompted for the source Java keystore password. After entering the password, the `Key Vault Server in read-only restricted Mode` error is displayed.

Read-Only Restricted State Functionality During a Primary Server Failure

You can set a failover threshold value to determine when a standby server takes over for a failed primary server.

In the event of a primary server failure, the standby server waits for the duration specified in the **Fast Start Failover Threshold (in secs)** field on the Configure Primary-Standby page. If the primary server is not reachable after the specified duration has elapsed, the standby server enters read-only restricted mode. In read-only restricted mode, only operations that fetch data are allowed. Endpoint operations that add new data or modify existing data on the Oracle Key Vault server are blocked.

Related Topics

- [Configuring the Primary-Standby Environment](#)
To configure a primary-standby environment, you must have the System Administrator role with access to the two servers (one primary and one standby).

Read-Only Restricted Mode Functionality During a Standby Server Failure

If a standby fails, the primary server waits for the duration in the **Fast Start Failover Threshold** field on the **Configure Primary-Standby** page.

If the standby server is not reachable after the specified duration has elapsed, the primary server enters read-only restricted mode. In read-only restricted mode, only operations that fetch data are allowed. Endpoint operations that add new data or modify existing data on the Oracle Key Vault server are blocked.

The primary server continues to provide limited service to the endpoints.

Related Topics

- [Configuring the Primary-Standby Environment](#)
To configure a primary-standby environment, you must have the System Administrator role with access to the two servers (one primary and one standby).

Read-Only Restricted State Functionality During a Network Failure

When a network failure affects communication between primary and standby servers, communication between certain endpoints and the primary server may also be affected.

The primary server waits for the duration specified in the **Fast Start Failover Threshold** field on the Configure Primary-Standby page. If the standby server is not reachable after the specified duration has elapsed, the primary server enters read-only restricted mode.

The standby server will also wait for the same duration. If the primary server is not reachable after the specified duration has elapsed, the standby server enters read-only restricted mode. The standby server takes over as the new primary server, and provides service to endpoints that cannot communicate with the affected primary server.

Related Topics

- [Configuring the Primary-Standby Environment](#)
To configure a primary-standby environment, you must have the System Administrator role with access to the two servers (one primary and one standby).

Enabling Read-Only Restricted Mode

Read-only restricted mode is enabled by default when primary-standby is configured.

Oracle recommends that you configure the primary-standby servers with read-only restricted mode enabled.

1. Unpair the primary server from the standby server, and then reinstall Oracle Key Vault on the standby server.
2. Perform post-installation tasks on the standby server.
3. Log in to the standby server Oracle Key Vault management console as a user who has the System Administrator role.
4. Select the **System** tab, and then **Settings** in the left navigation bar.
5. In the System Configuration area, click **Primary-Standby**.

In the Primary-Standby page, ensure that you select **Yes** in the **Allow Read-Only Restricted Mode** field.

6. Log in to the primary server Oracle Key Vault management console as a user who has the System Administrator role.
7. On the Primary-Standby page, ensure that you select **Yes** in the **Allow Read-Only Restricted Mode** field.
8. Click **Initiate Pairing**.

Read-only restricted mode takes effect if connectivity is lost between the primary and standby servers. Read-only restricted mode has no effect on a standalone server.

Related Topics

- Oracle Key Vault Installation and Upgrade Guide
- [Configuring the Primary-Standby Environment](#)
To configure a primary-standby environment, you must have the System Administrator role with access to the two servers (one primary and one standby).

Disabling Read-Only Restricted Mode

Read-only restricted mode is enabled by default when primary-standby is configured.

Oracle recommends that you configure primary-standby with read-only restricted mode enabled. Follow these steps if an existing primary-standby deployment with read-only restricted mode that is enabled must be converted to a deployment that has read-only restricted mode disabled.

1. Unpair the primary server from the standby server, and reinstall Oracle Key Vault on the standby server.
2. Perform postinstallation tasks on the standby server.
3. Log in to the standby server Oracle Key Vault management console as a user who has the System Administrator role.
4. Select the **System** tab, and then **Settings** in the left navigation bar.
5. In the System Configuration area, click **Primary-Standby**.

On the Primary-Standby page, ensure that you select **No** in the **Allow Read-Only Restricted Mode** field.

6. Log in to the primary server Oracle Key Vault management console as a user who has the System Administrator role.
7. On the Primary-Standby page, ensure that you select **No** in the **Allow Read-Only Restricted Mode** field.
8. Click **Initiate Pairing**.

After read-only restricted mode is disabled, it does not take effect if connectivity is lost between the primary and standby servers. Read-only restricted mode has no effect on a standalone server.

Related Topics

- Oracle Key Vault Installation and Upgrade Guide
- [Configuring the Primary-Standby Environment](#)
To configure a primary-standby environment, you must have the System Administrator role with access to the two servers (one primary and one standby).

Recovering from Read-Only Restricted Mode

To recover an instance from read-only restricted mode after a network failure or standby server failure, manual intervention may be required.

You must unpair and reset the surviving instance, reinstate a new Oracle Key Vault server, and pair it as the new standby to the surviving server. The following are the possible scenarios:

- **Primary server failure:** Depending on the operational state of the primary server at the time of failure, it could be restarted and some functionality may be available. However, due to possible corruption of the embedded Oracle Key Vault database, recovery may not be possible. You would then need to reinstate the Oracle Key Vault instance because of the partial failure. If the failed server is unable to again pair with the peer server within 20 minutes, then you must re instantiate the server.

Even though the endpoint processes communicating with the Oracle Key Vault servers retain the IP address of the last known reachable server, they must determine the IP address of the new Oracle Key Vault server when spawned. The endpoint processes attempt to communicate with the Oracle Key Vault server configured as the primary server in the configuration scripts, and then wait for a response before trying to reach the server configured as the standby server in the configuration scripts. To minimize downtime, Oracle recommends that you initiate a switchover after reinstating the failed primary server.

- **Standby server failure:** The primary server runs in the read-only restricted mode if there is a standby server failure. Reinstall the standby server if it does not automatically pair with the primary server.
- **Power loss or network connectivity failure:** When a network failure occurs, the primary and standby servers are unable to communicate, and both servers enter read-only restricted mode. The standby also attempts to failover to the primary server. Once communication is reestablished between the primary and standby servers, the old primary server is automatically converted to the new standby. The data from the new primary server overwrites the old primary server's data, resulting in the loss of audit records from the old primary server. Oracle recommends that you enable syslog auditing to preserve the audit records that were overwritten on the old primary. Similar to recovering from primary server failure, Oracle recommends that you perform a switchover after recovery. You should not enroll any new endpoints before the switchover.

Related Topics

- [Restoring Primary-Standby After a Failover](#)
A failover takes place if the primary server fails.

Read-Only Restricted Mode Notifications

When the primary or standby server enters read-only restricted mode, Oracle Key Vault generates an alert.

You can view these alerts on the **Alerts** page. If email notifications are configured, then an email notification is sent.

Related Topics

- [Viewing Open Alerts](#)
Users can view alerts depending on their privileges.
- [Configuring Email Notification](#)
You can use email notifications to directly notify administrators of Key Vault status changes without logging into the Oracle Key Vault management console.

Best Practices for Using Oracle Key Vault in a Primary-Standby Configuration

Oracle provides guidelines for ensuring operational continuity and minimal downtime of Oracle Key Vault.

- Configure your Transparent Data Encryption (TDE)-enabled databases to have an auto-login connection to Oracle Key Vault. *Oracle Database Advanced Security Guide* describes how to configure auto-login keystores.
- Apply the database patch for Bug 22734547 to tune the Oracle Key Vault heartbeat.
- Ensure that read-only restricted mode is enabled in primary-standby Oracle Key Vault deployments.
- Set the duration in the **Fast Start Failover Threshold** field on the Configure Primary-Standby page to a value that avoids unnecessary failover due to transient network interruptions.
- Configure syslog auditing to capture audit records in read-only restricted mode.
- Switch over to the original primary server in case the primary server is reinstated.
- Before attempting any unpair operations, check *Oracle Key Vault Release Notes* for known issues.
- Before attempting any switchover or unpair operations, check *Oracle Key Vault Release Notes* for any known issues.

Related Topics

- [Oracle Database Advanced Security Guide](#)
- [Oracle Key Vault Release Notes](#)

Oracle Key Vault Integration Accelerator

Oracle Key Vault provides accelerators to quickly spin up integration of use-cases with Key Vault. The accelerators can be used as provided or extended to suit application needs.

You can download an integration accelerator from the Oracle Key Vault management console. Oracle Key Vault provides an integration accelerator for:

- [Key Management for Oracle DBMS_CRYPT0 Integration Accelerator](#)
Starting with release 21.10, Oracle Key Vault provides the Key Management for Oracle DBMS_CRYPT0 integration accelerator to manage encryption keys used with DBMS_CRYPT0 operations in Oracle Key Vault.

Key Management for Oracle DBMS_CRYPT0 Integration Accelerator

Starting with release 21.10, Oracle Key Vault provides the Key Management for Oracle DBMS_CRYPT0 integration accelerator to manage encryption keys used with DBMS_CRYPT0 operations in Oracle Key Vault.

Oracle DBMS_CRYPT0 package provides a PL/SQL interface to perform encrypt and decrypt operations. PL/SQL applications need to securely manage encryption keys used with DBMS_CRYPT0 operations. Using this integration accelerator, you can manage the symmetric keys used with Oracle DBMS_CRYPT0 operations in Oracle Key Vault. This integration accelerator package provides a PL/SQL wrapper built over an Oracle Key Vault JAVA SDK program.

- [Downloading Key Management for Oracle DBMS_CRYPT0 Integration Accelerator](#)
You can download the Key Management in Oracle DBMS_CRYPT0 integration accelerator from the Oracle Key Vault management console.
- [Using Key Management for Oracle DBMS_CRYPT0 Integration Accelerator](#)
Learn how to use Key Management for DBMS_CRYPT0 integration accelerator.

Downloading Key Management for Oracle DBMS_CRYPT0 Integration Accelerator

You can download the Key Management in Oracle DBMS_CRYPT0 integration accelerator from the Oracle Key Vault management console.

1. Connect to the Oracle Key Vault management console.

For example:

```
https://192.0.2.254
```

The login page to the Oracle Key Vault management console appears. **Do not log in.**

2. In the lower-right corner of the login page under **Login**, click **Endpoint Enrollment and Software Download**.

The **Enroll Endpoint & Download Software** page appears.

3. At the top of the page, click the **Download Integration Accelerator** tab.
4. In the **Download Integration Accelerator** page, select **Key Management for DBMS_CRYPTO** package from the Integration Accelerator drop-down menu.
5. Click **Download**.
6. Save the `okv_key_management_dbms_crypto.zip` file to a desired location.

Using Key Management for Oracle DBMS_CRYPTO Integration Accelerator

Learn how to use Key Management for DBMS_CRYPTO integration accelerator.

- [Prerequisites for Using Key Management for Oracle DBMS_CRYPTO Integration Accelerator](#)
Before you start working with the Key Management for Oracle DBMS_CRYPTO Integration Accelerator, ensure that you meet these prerequisites.
- [Preparing the Environment for Using Key Management for Oracle DBMS_CRYPTO Integration Accelerator](#)
Ensure that you prepare the Oracle Database environment with the necessary permissions to run the Key Management for Oracle DBMS_CRYPTO Integration Accelerator.
- [Using Key Management for Oracle DBMS_CRYPTO Integration Accelerator in a PL/SQL Application](#)
You can now use the PL/SQL package `okv_key_mgmt` supplied by the integration accelerator in your PL/SQL applications.
- [Extending the Functionality of Key Management for Oracle DBMS_CRYPTO Integration Accelerator](#)
You can extend functionality of the Key Management for Oracle DBMS_CRYPTO Integration Accelerator to suite your application needs.

Prerequisites for Using Key Management for Oracle DBMS_CRYPTO Integration Accelerator

Before you start working with the Key Management for Oracle DBMS_CRYPTO Integration Accelerator, ensure that you meet these prerequisites.

- Enroll Oracle Database as an endpoint to Oracle Key Vault 21.10 or later, on which you intend to run the sample Oracle DBMS_CRYPTO application.

Note:

For more information see, [About Endpoint Enrollment and Provisioning](#)

- Ensure that you have Java 8 or later.
- Ensure that Oracle Database is patched with the latest Oracle JVM.

- Set the `OKV_HOME` environment variable appropriately.

 **Note:**

The `$OKV_HOME` is the destination directory for the endpoint software that you specified with the `-d` option during the endpoint installation.

- Check if the `okvclient.ora` file is available and created as part of the endpoint enrollment. The `okvclient.ora` file is usually located at `$OKV_HOME/conf/okvclient.ora`.
- Ensure that `okv_jsdk.zip` is unzipped under `$OKV_HOME`.

In Bash UNIX shell:

```
$ cd $OKV_HOME
$ unzip -o okv_jsdk.zip
```

 **Note:**

For information about downloading the Oracle Key Vault Java SDK, see Oracle Key Vault Developer's Guide

- Extract the contents of `okv_key_management_dbms_crypto.zip` to a separate folder. The `okv_key_management_dbms_crypto.zip` file contains the following:
 - `setup_java_permissions.sql`: This SQL script sets up the required Java security permissions and database privileges to access the resources stored within the database.
 - `KeyManager.java`: This program creates and fetches a symmetric key from the Oracle Key Vault server using APIs from Oracle Key Vault Java SDK.
 - `okv_key_mgmt.pkb`: A PL/SQL wrapper that invokes the Oracle Key Vault Java SDK API (through `KeyManager.java`) to perform create and fetch operations of symmetric key in Oracle Key Vault.
 - `dbms_crypto_using_okv_keymgmt.sql`: Oracle DBMS_CRYPTO use case for demonstrating key management in Oracle Key Vault using Java SDK.
 - `README.txt`: Instructions to set up and run the sample Oracle DBMS_CRYPTO application.

Related Topics

- [About Endpoint Enrollment and Provisioning](#)
Endpoints are Oracle Key Vault clients that use the server to store and manage security objects, share them with trusted peers, and retrieve them.
- Oracle Key Vault Developer's Guide

Preparing the Environment for Using Key Management for Oracle DBMS_CRYPTO Integration Accelerator

Ensure that you prepare the Oracle Database environment with the necessary permissions to run the Key Management for Oracle DBMS_CRYPTO Integration Accelerator.

1. Ensure that the `JAVA_HOME` environment variable points to the Java Development Toolkit (JDK) location.
2. Set the `CLASSPATH` to contain the Oracle Key Vault Java SDK jar `$OKV_HOME/jsdk/lib/okvjsdk.jar`.

Set the `CLASSPATH` in Bash UNIX:

```
$ export CLASSPATH=$CLASSPATH:$OKV_HOME/jsdk/lib/okvjsdk.jar
```

3. Ensure that the current directory `'.'` is also part of `CLASSPATH`. If not, then add the current directory at the beginning of `CLASSPATH`.

In Bash UNIX shell, set as follows:

```
$ export CLASSPATH=.$CLASSPATH
```

4. Compile the `KeyManager.java` file.

Set the current directory to the directory where you extracted the `okv_key_management_dbms_crypto.zip` file.

Compile the `KeyManager.java` in Bash UNIX:

```
$ cd folder_path
$ javac KeyManager.java
```

5. Log in to Oracle Database as the `sys` user and set up the database user.

Determine the default table space `default_tablespace` of the database user `db_username` using the following query:

```
SQL> select default_tablespace from dba_users where username = db_username;
```

 **Note:**

Ensure `db_username` is in upper case.

Grant the following permissions to `db_username`:

```
SQL> grant execute on sys.dbms_crypto to db_username;
SQL> grant create session to db_username;
SQL> grant create procedure, create table to db_username;
SQL> alter user db_username quota 100M on default_tablespace;
SQL> commit;
SQL> exit;
```

6. Ensure that you add the `$ORACLE_HOME/bin` directory to the `PATH` environment variable.
7. Load the Java class file `KeyManager.class` and the dependent JAR file `okvjsdk.jar` to Oracle Database as the database user `db_username`, database user password `db_pswd`, and database name `db_name`.

```
$ loadjava -user db_username/db_pswd@db_name -verbose $OKV_HOME/jsdk/lib/okvjsdk.jar
```

```
$ cd folder_path
$ loadjava -user db_username/db_pswd@db_name -verbose KeyManager.class
```

8. Log in to the database as the `sys` user and run the following SQL file by passing the database user `db_username` and actual value of `OKV_HOME` as arguments.

```
SQL> @folder_path/setup_java_permissions.sql db_username OKV_HOME;
```

9. Log in to the database as the `sys` user and grant permissions required to allow the database user `db_username` to establish a secure socket connection with Oracle Key Vault:

```
SQL> exec dbms_java.grant_permission( db_username,
'SYS:java.net.SocketPermission', 'Oracle Key Vault IP Address:Port#',
'connect,resolve' );
SQL> commit;
SQL> exit;
```

Note:

In a multi-master cluster deployment, grant the permission for each cluster node using the IP address and port-number. Ensure that the input `db_username` is in upper case for these commands.

The IP address and port-number information is available in the `$OKV_HOME/conf/okvclient.ora` file.

For example, the endpoint configuration file `$OKV_HOME/conf/okvclient.ora` for a 3-node Oracle Key Vault cluster with IP addresses 100.100.100.101, 100.100.100.102 and 100.100.100.103 for the respective nodes is similar to the following:

```
SERVER=100.100.100.101:5696/
CN=server_cert,OU=Key_Vault,O=Oracle,L=Redwood_City,ST=California,C=us,100.
100.100.102:5696/
CN=node_2_cert,OU=Key_Vault,O=Oracle,L=Redwood_City,ST=California,C=us
READ_SERVER=100.100.100.103:5696/
CN=node_3_cert,OU=Key_Vault,O=Oracle,L=Redwood_City,ST=California,C=us
```

Extract the IP address and port details from the previous configuration file and run SQL commands as the `sys` user for the previous example cluster as follows:

```
SQL> exec dbms_java.grant_permission( db_username,
'SYS:java.net.SocketPermission', '100.100.100.101:5696',
'connect,resolve' );
SQL> exec dbms_java.grant_permission( db_username,
'SYS:java.net.SocketPermission', '100.100.100.102:5696',
'connect,resolve' );
SQL> exec dbms_java.grant_permission( db_username,
'SYS:java.net.SocketPermission', '100.100.100.103:5696',
'connect,resolve' );
SQL> commit;
SQL> exit;
```

Ensure that you change these permissions whenever there are changes to the Oracle Key Vault cluster node configuration, such as adding or deleting cluster nodes.

For example, when a new cluster node 100.100.100.104 is added:

```
SQL> exec
dbms_java.grant_permission( db_username, 'SYS:java.net.SocketPermission',
'100.100.100.104:5696', 'connect,resolve' );
SQL> commit;
SQL> exit;
```

When the cluster node 100.100.100.103 is removed:

```
SQL> exec dbms_java.revoke_permission( db_username,
'SYS:java.net.SocketPermission', '100.100.100.103:5696',
'connect,resolve' );
SQL> commit;
SQL> exit;
```

10. Log in to the database with your database user name and password and grant the following permissions to the database user *db_username* to enable the database user to load the endpoint wallet under `$OKV_HOME/ssl` file when establishing a secure connection with Oracle Key Vault.

```
SQL> exec dbms_java.grant_policy_permission( db_username, db_username,
'oracle.security.pki.OracleWalletPermission', 'load' );
SQL> exec dbms_java.grant_permission( db_username,
'db_username:oracle.security.pki.OracleWalletPermission', 'load', '' );
SQL> commit;
SQL> exit;
```

 **Note:**

Ensure that the input *db_username* is in upper case for these commands.

11. Connect to the database with your database user name and password and load the package `okv_key_mgmt.pkb`.

```
SQL> @folder_path/okv_key_mgmt.pkb;
SQL> exit;
```

Using Key Management for Oracle DBMS_CRYPTO Integration Accelerator in a PL/SQL Application

You can now use the PL/SQL package `okv_key_mgmt` supplied by the integration accelerator in your PL/SQL applications.

The `okv_key_mgmt` package provides the following functions to integrate key management with Oracle Key Vault.

- `create_key`: Creates a symmetric key in Oracle Key Vault. This is a wrapper function calling `KeyManager.java:createKey()` which makes use of the Oracle Key Vault Java SDK APIs to create a symmetric key in Oracle Key Vault.

- `fetch_key`: Fetches a symmetric key from Oracle Key Vault. This is a wrapper function calling `KeyManager.java:fetchKey()` which makes use of the Oracle Key Vault Java SDK APIs to fetch a symmetric key in Oracle Key Vault.

You can modify PL/SQL applications that perform symmetric key based crypto operations using DBMS_CRYPTO to fetch and use the keys from Oracle Key Vault using `okv_key_mgmt` package. This way, you can simplify the key management for your PL/SQL applications.

Review the supplied sample application `dbms_crypto_using_okv_keymgmt.sql` included with the integration accelerator to learn how to use key management functions of the `okv_key_mgmt` package.

You can run the sample application `dbms_crypto_using_okv_keymgmt.sql` included with the Key Management for DBMS_CRYPTO integration accelerator package.

1. Log in to database with your user name and password to the configured database.
2. Run the `dbms_crypto_using_okv_keymgmt.sql` file using the following command:

```
SQL> @folder_path/dbms_crypto_using_okv_keymgmt.sql;
```

Note:

- This SQL script works only for auto-login wallet of the installed Oracle Key Vault endpoint software. If your endpoint is password-protected, then enter the password of your endpoint in 'endpt_pswd' variable present in this SQL file.
- If you want to give a name for the key created in Oracle Key Vault, then enter the key name value in 'key_name' variable present in this SQL file.

The Key Management for DBMS_CRYPTO integration accelerator provides basic functions to demonstrate the management of keys for PL/SQL application in Oracle Key Vault. You can modify the supplied Java SDK program `KeyManager.java` and PL/SQL wrapper package `okv_key_mgmt.pkb` to extend its functionality to cover more advanced usage.

Extending the Functionality of Key Management for Oracle DBMS_CRYPTO Integration Accelerator

You can extend functionality of the Key Management for Oracle DBMS_CRYPTO Integration Accelerator to suite your application needs.

The available PL/SQL wrapper package `okv_key_mgmt.pkb` provides only the basic functions `create_key()` and `fetch_key()` to create and fetch a key respectively. The function `fetch_key()` requires the unique identifier of the key (key ID) as input.

In many cases, you may find it easier to associate human-friendly names to the keys. That way, you must first retrieve the ID of the key by locating the key using its name and then pass the retrieved key ID to `fetch_key()` to obtain the key to use with DBMS_CRYPTO operations. The available `create_key()` function already accepts a name to associate with the newly created key. However, the function to locate a key by its name is not available.

To extend the Key Management for Oracle DBMS_CRYPTO integrator accelerator to support a function to locate a key by its name, you must modify the available Java SDK program

KeyManager.java and PL/SQL wrapper package okv_key_mgmt.pkb using the following procedure:

1. Add the function locateKey() for locating the key using Oracle Key Vault Java SDK APIs in KeyManager.java:

```
public static String locateKey(String epPswd, String nameAttrValue)
                                throws OKVException {

    /* Create a connection to Oracle Key Vault */
    char[] endpointPswd = null;

    if ((epPswd != null) && !(epPswd.trim().isEmpty())) {
        endpointPswd = epPswd.toCharArray();
    }
    OKVService service = setupOKVConnection(endpointPswd);

    /* Locate the key */
    System.out.println("\nLocate a key on Oracle Key Vault");

    /* Create the attribute list for locate operation */
    OKVTTLV attrList = OKVTTLV.okvEnvGetOpRequestObj();
    OKVTTLV nameAttr = attrList.okvAddAttributeObject(OKVTag.NAME, 0);
    nameAttr.okvAttrAddName(nameAttrValue,

OKVTagEnum.NAME_TYPE_UNINTERPRETED_TEXT_STRING);
    OKVTTLV stateAttr = attrList.okvAddAttributeObject(OKVTag.STATE,
0);
    stateAttr.okvAttrAddState(OKVTagEnum.STATE_ACTIVE);

    OKVUidListResponse result = service.okvLocate(null, 0, null,
                                                attrList);

    /* Verify if the key is located */
    int count = result.getUidCnt();
    String uniqueId = null;
    if (count == 1) {
        uniqueId = result.getUids().get(0);
        System.out.println("\tSuccessfully located the key. " +
            "Unique ID of the key: " + uniqueId);
    }
    else {
        System.out.println("\tUnable to locate the key, " +
            "number of unique ID's got: " + count);
    }

    /* Release the connection from Oracle Key Vault */
    releaseOKVConnection(service);

    return uniqueId;
}
```

2. Add the wrapper function locate_key() in okv_key_mgmt.pkb:

a. Declaration:

```
function locate_key(ep_pswd varchar2, key_name varchar2) return
varchar2;
```

b. Definition:

```
function locate_key(ep_pswd varchar2, key_name varchar2) return
varchar2 as language java name 'KeyManager.locateKey
(java.lang.String,java.lang.String) return java.lang.String';
```

- 3. Add the call to `okv_key_mgmt.locate_key()` before calling `okv_key_mgmt.fetch_key()` in `dbms_crypto_using_okv_keymgmt.sql` similar to the following:**

```
key_id := okv_key_mgmt.locate_key(endpt_pswd, key_name);
key_value := okv_key_mgmt.fetch_key(endpt_pswd, trim(key_id));
```

 **Note:**

Ensure that 'key_name' is not an empty string or null in the `dbms_crypto_using_okv_keymgmt.sql` file.

- 4. Recompile modified `KeyManager.java`:**

```
$ cd folder_path
$ javac KeyManager.java
```

- 5. Reload `KeyManager.class` to Oracle Database to include the new changes:**

```
$ cd folder_path
$ loadjava -user db_username/db_pswd@db_name -verbose KeyManager.class
```

- 6. Connect to the database with your user name and password and reload `okv_key_mgmt.pkb` to include the new `locate_key()` functionality:**

```
SQL> @folder_path/okv_key_mgmt.pkb;
SQL> exit;
```

- 7. To verify the newly included `Locate` operation, connect to the database with your user name and password and run the `dbms_crypto_using_okv_keymgmt.sql` file using the following command:**

```
SQL> @folder_path/dbms_crypto_using_okv_keymgmt.sql;
```

A

Oracle Key Vault Multi-Master Cluster Operations

There are restrictions and conditions for Oracle Key Vault multi-master cluster operations on cluster nodes.

Table A-1 Oracle Key Vault Multi-Master Cluster Operations on Cluster Nodes

Management Console Tab and Operation	Read-Only Node	Read/Write Node in Read-Only Restricted Mode	Read/Write Node in normal (read/write) Mode
Home tab	No restrictions	No restrictions	No restrictions
Endpoints tab	Updated only via replication from a read/write node	Updated only via replication from a read/write node	Directly updated using client tools on this node
<ul style="list-style-type: none"> • Add, Delete • Suspend, Resume • Reenroll 			Also updated by replication from other read/write nodes
Endpoint Groups	No restrictions	No restrictions	No restrictions
<ul style="list-style-type: none"> • Create Group • Delete Group 			
Update Endpoint Settings			
Keys & Wallets tab	Updated only through replication from a read/write node	Updated only via replication from a read/write node	Updated using client tools on this node.
Wallets <ul style="list-style-type: none"> • Create, Delete, Edit Keys, Secrets & Objects <ul style="list-style-type: none"> • Delete • Edit <ul style="list-style-type: none"> – Update – Revoke, Destroy – Change Wallet Membership 			Also updated by replication from other read/write nodes.
Reports tab	No restrictions	No restrictions	No restrictions
Audit			
<ul style="list-style-type: none"> • Generate audit report • Export audit records • Delete audit records 			
Reports			
<ul style="list-style-type: none"> • Generate any report 			
Alerts			
<ul style="list-style-type: none"> • View alerts • Configure alerts 			

Table A-1 (Cont.) Oracle Key Vault Multi-Master Cluster Operations on Cluster Nodes

Management Console Tab and Operation	Read-Only Node	Read/Write Node in Read-Only Restricted Mode	Read/Write Node in normal (read/write) Mode
Users tab Users <ul style="list-style-type: none"> • Create, Delete • Check Conflict Status Manage Access (User Groups) <ul style="list-style-type: none"> • Update • Add, Remove Wallet Access • Add, Remove Members Change Password	Updated only with replication from a read/write node	Updated only with replication from a read/write node	Updated using client tools on this node Also updated by replication from other read/write nodes There are additional considerations and restrictions based on the status of the user name and user group name.
System tab System Settings <ul style="list-style-type: none"> • Reboot, Poweroff • Edit Network Details • Edit Network Services • Edit System Time • Edit DNS • Enable FIPS Mode • Configure Syslog • Enable Audit Vault Integration 	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node. The DNS settings and System Time are not set for the cluster here.
System tab Cluster System Settings <ul style="list-style-type: none"> • Edit System Time • Edit DNS • Edit Max Disable Node Duration • Enable RESTful Services • Configure Syslog 	Updated only with replication from a read/write node	Updated only with replication from a read/write node	Updated using client tools on this node Also updated by replication from other read/write nodes
System tab Audit Settings, Scope 'Node' <ul style="list-style-type: none"> • Enable Auditing • Replicate Audit Records • Send Audit to Syslog 	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.

Table A-1 (Cont.) Oracle Key Vault Multi-Master Cluster Operations on Cluster Nodes

Management Console Tab and Operation	Read-Only Node	Read/Write Node in Read-Only Restricted Mode	Read/Write Node in normal (read/write) Mode
System tab Audit Settings, Scope 'Cluster' <ul style="list-style-type: none"> • Enable Auditing • Replicate Audit Records • Send Audit to Syslog 	Updated only with replication from a read/write node	Updated only with replication from a read/write node	Updated using client tools on this node. Also updated by replication from other read/write nodes
System tab Email Settings <ul style="list-style-type: none"> • Edit 	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.
System tab Monitoring Settings, Scope 'Node' <ul style="list-style-type: none"> • Enable Monitoring • Limit Access • Edit 	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.
System tab Monitoring Settings, Scope 'Cluster' <ul style="list-style-type: none"> • Enable Monitoring • Limit Access • Edit 	Updated only with replication from a read/write node	Updated only with replication from a read/write node	Updated using client tools on this node. Also updated by replication from other read/write nodes
System tab System Backup <ul style="list-style-type: none"> • Configure • Perform Backup • Perform Restore 	Node is used to update these settings. The updates are local to the node. A backup can only be restored to a standalone Oracle Key Vault server. Restoring a backup implies that the entire cluster has failed and needs to be rebuilt.	Node is used to update these settings. The updates are local to the node. A backup can only be restored to a standalone Oracle Key Vault server. Restoring a backup implies that the entire cluster has failed and needs to be rebuilt.	Node is used to update these settings. The updates are local to the node. A backup can only be restored to a standalone Oracle Key Vault server. Restoring a backup implies that the entire cluster has failed and needs to be rebuilt.
System tab Console Certificate <ul style="list-style-type: none"> • Generate, Upload 	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.
System tab SSH Tunnel Settings <ul style="list-style-type: none"> • Add, Delete • Edit 	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.
System tab HSM <ul style="list-style-type: none"> • All operations 	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.	Node is used to update these settings. The updates are local to the node.

Table A-1 (Cont.) Oracle Key Vault Multi-Master Cluster Operations on Cluster Nodes

Management Console Tab and Operation	Read-Only Node	Read/Write Node in Read-Only Restricted Mode	Read/Write Node in normal (read/write) Mode
Cluster tab Management section <ul style="list-style-type: none"> Add Node Delete Node Force Delete Node Disable Node Enable Node 	A node in the ACTIVE state may be used to add, delete, force delete, or disable a node. When adding a node, selecting Add Node as a Read-Write Peer creates a read/write pair. Only a disabled node may enable itself. Delete and force delete have special considerations as noted.	A node in the ACTIVE state may be used to add, delete, force delete, or disable a node. When adding a node, this node cannot be added as a read/write peer to the new node, as it is already in a read/write pair. Only a disabled node may enable itself. Delete and force delete have special considerations as noted.	A node in the ACTIVE state may be used to add, delete, force delete, or disable a node. When adding a node, this node cannot be added as a read/write peer to the new node, as it is already in a read/write pair. Only a disabled node may enable itself. Delete and force delete have special considerations as noted.
Cluster tab Monitoring <ul style="list-style-type: none"> View information Enable, Disable link state 	Node can access and update these settings. The updates are local to the node.	Node can access and update these settings. The updates are local to the node.	Node can access and update these settings. The updates are local to the node.
Cluster tab Conflict Resolution <ul style="list-style-type: none"> Edit Accept 	Node can access but not resolve conflicts. Updates are received only from active read/write nodes in the cluster through replication.	Node can access but not resolve conflicts. Updates are received only from active read/write nodes in the cluster through replication.	Node can access and resolve conflicts. Updates are propagated to all other nodes in the cluster.
Join read/write pair	Only through induction from a read-only node. Requires Add Node as Read-Write Peer set to Yes .	Not applicable. Since this node is already a member of a read/write pair, when replication is once again available from this node to its read/write peer, it will return to its read/write state.	Not applicable

B

Oracle Key Vault `okvutil` Endpoint Utility Reference

The `okvutil` utility enables you to perform tasks uploading and downloading security objects.

- [About the `okvutil` Utility](#)
The `okvutil` utility is a command-line utility that you can use to manage security objects.
- [`okvutil` Command Syntax](#)
The `okvutil` utility syntax provides short and long options for specifying commands.
- [`okvutil changepwd` Command](#)
The `okvutil changepwd` command changes the password associated with the credentials used to connect to Oracle Key Vault.
- [`okvutil diagnostics` Command](#)
The `okvutil diagnostics` command collects diagnostic and environmental information on an endpoint to troubleshoot deployment issues.
- [`okvutil download` Command](#)
The `okvutil download` command downloads security objects from Oracle Key Vault to the endpoint.
- [`okvutil list` Command](#)
The `okvutil list` command lists the available security objects that are uploaded.
- [`okvutil upload` Command](#)
The `okvutil upload` command uploads security objects to Oracle Key Vault.
- [`okvutil sign` Command](#)
The `okvutil sign` command generates a digital signature for a message or message digest by using the private key stored on the Oracle Key Vault server.
- [`okvutil sign-verify` Command](#)
The `okvutil sign-verify` command verifies digital signature of a message or message digest using the public key or certificate stored on Oracle Key Vault Server.
- [`okvutil show` Command](#)
The `okvutil show` command displays the endpoint information.
- [`okvutil` Common Errors](#)
The `okvutil` common errors describes the error conditions and the reason for their occurrence.

About the `okvutil` Utility

The `okvutil` utility is a command-line utility that you can use to manage security objects.

The `okvutil` command-line utility enables you to locate, upload, and download security objects to and from Oracle Key Vault. You can also use `okvutil` to change the wallet password and collect system diagnostics.

The `okvutil` utility uses the Transport Layer Security (TLS) credentials provisioned for the endpoint to authenticate to Oracle Key Vault.

okvutil Command Syntax

The `okvutil` utility syntax provides short and long options for specifying commands.

Syntax

```
okvutil command arguments [-v verbosity_level]
```

Parameters

Table B-1 okvutil Command Syntax

Parameter	Description
command	Refers to any of the following commands: <code>upload</code> , <code>list</code> , <code>download</code> , <code>changepwd</code> , <code>diagnostics</code>
arguments	Refers to the arguments that you pass for the accompanying command.
<code>-v</code> , <code>--verbose</code>	Refers to verbosity level. Possible values are 0, 1, 2, 3, and 4. Verbosity level 0 disables the verbose output. The level of detail that is printed on the standard output during command execution increases with higher verbosity levels. Verbosity level 4 provides the highest level of detail.
<code>-h</code> , <code>--help</code>	Use option to get help with any <code>okvutil</code> command. For example: <code>okvutil <i>command</i> --help</code>

Short and Long Forms of Specifying Options

You can specify the options in either a short form or a long form.



Note:

Endpoint platforms AIX and HP-UX (IA) support only short form options currently

- **Short form:** Only use one hyphen and the single-letter option name. For example:

```
-l /home/username  
-t wallet
```

- **Long form:** Provide two hyphens and the full option name. For example:

```
--location /home/username  
--type wallet
```

The examples in this guide use the short form.

How Password Prompts for okvutil Work

The `okvutil` commands prompt for passwords in the following situations:

- If you created a password-protected wallet during endpoint installation to access Oracle Key Vault.

- If you specify an Oracle wallet file or Java keystore file using the `-l` option, `okvutil` prompts you to provide the password for the wallet or keystore that `okvutil` is trying to upload to Oracle Key Vault.

okvutil changepwd Command

The `okvutil changepwd` command changes the password associated with the credentials used to connect to Oracle Key Vault.

Use this command if you used a password-protected wallet to store the Oracle Key Vault endpoint user credentials. The new password does not need to be the same password for the JCKS or wallet file when it was uploaded.

Syntax

Short format:

```
okvutil changepwd -l location [-t type] [-v verbosity_level ]
```

Long format:

```
okvutil changepwd
--location location
[--type type]
[--verbose verbosity_level]
```

Parameters

Table B-2 okvutil changepwd Command Options

Parameter	Description
<code>-l, --location</code>	Specifies the directory location of the wallet whose password you want to change.
<code>-t, --type</code>	Specifies the data type. Enter <code>WALLET</code> . If you omit the <code>-t, --type</code> option, then the default type is <code>WALLET</code> .
<code>-v, --verbose</code>	Refers to the verbosity level from 0 (none), 1 (debug), 2 (detailed debug).

Example: Changing an Oracle Key Vault Endpoint Password

The following example shows how to use the `okvutil changepwd` command change the endpoint password. When you are prompted to create the new password, enter a password that is between 8 and 30 characters.

```
$ okvutil changepwd -l ./home/oracle/okvutil/ssl -t WALLET
Enter wallet password: current_endpoint_password
Enter new wallet password: new_endpoint_password
Confirm new wallet password: new_endpoint_password
```

okvutil diagnostics Command

The `okvutil diagnostics` command collects diagnostic and environmental information on an endpoint to troubleshoot deployment issues.

The information is placed in a `diagnostics.zip` file, which can be given to Oracle support for further analysis and debugging.

The information gathered includes information on the following:

- The shell environment variables: OKV_HOME, ORACLE_HOME, ORACLE_BASE, ORACLE_SID, PATH, CLASSPATH
- Configuration and IP address of the Oracle Key Vault server from okvclient.ora
- Directory listing of OKV_HOME and its sub-directories
- Oracle Key Vault log files from the endpoint
- Listing of symbolic links created by the Oracle Key Vault endpoint installer
- Network settings and ping results

The `okvutil diagnostics` command does not collect sensitive information such as user credentials or security objects.

Syntax

Short format:

```
okvutil diagnostics [-v verbosity_level]
```

Long format:

```
okvutil diagnostics [--verbose verbosity_level]
```

Parameters

Table B-3 okvutil diagnostics Command Options

Parameter	Description
<code>-v, --verbose</code>	Refers to the verbosity level from 0 (none), 1 (debug), 2 (detailed debug)

Example: Collecting System Diagnostics

The following example shows how to run the `okvutil diagnostics` command. After you run the command, when the `Diagnostics complete` message appears, then the `diagnostics.zip` file will be available in the current directory.

```
$ okvutil diagnostics
Diagnostics collection complete.
ls
diagnostics.zip
```

okvutil download Command

The `okvutil download` command downloads security objects from Oracle Key Vault to the endpoint.

These security objects include Oracle wallets including auto-login wallets, Java keystores, credential files, SSH public keys, and other types of key storage files.

You can only download the contents of a virtual wallet into a keystore (a container such as an Oracle wallet or a JCEKS keystore that can hold multiple security objects), and not into a credential file.

Some keystores only support the storage of certain types of security objects. An error occurs if you upload a DSA key from a Java keystore or later try to download it to a different type of keystore like an Oracle wallet.

Endpoints of type SSH server can download public keys from wallets of type SSH server by providing the fingerprint of the SSH public key.

Syntax

Short format:

```
okvutil download [-l location] -t type [-g group | -i object_id | -g <group> -F  
<fingerprint> -H <SSH-server-host-user>] [-o] [-v verbosity_level]
```

Long format:

```
okvutil download [--location location] --type type [--group group | --item object_id | --  
group <group> --fingerprint <fingerprint> --ssh-server-host-user <SSH-server-host-user>]  
[--overwrite] [--verbose verbosity_level]
```

Parameters

Table B-4 okvutil download Command Options

Parameter	Description
-l, --location	Specifies the file location to store the items that you want to download. Ensure that you have permission to create wallets in this location. Ensure that the file you download is no more than 120 KB. This setting is mandatory if you are downloading the SSH public key.

Table B-4 (Cont.) okvutil download Command Options

Parameter	Description
-t, --type	<p>Specifies the data type of the object being downloaded from Oracle Key Vault. It must be a value from the following list:</p> <ul style="list-style-type: none"> • WALLET for an Oracle wallet • JKS for a Java keystore • JCEKS for a Java Cryptography Extension keystore (JCEKS) • SSH for an SSH key file, to be downloaded as an opaque object. • KERBEROS for a Kerberos keytab, to be downloaded as an opaque object. • SSH_PUBLIC_KEY for SSH public keys. • OTHER for opaque objects, which are other files that store secrets. <p>The WALLET, JKS, and JCEKS types contain multiple objects. Oracle Key Vault downloads each of these objects individually. The SSH, KERBEROS, and OTHER types, being opaque objects, are downloaded as single files.</p> <p>The object of type SSH_PUBLIC_KEY being an SSH public key is downloaded to a single file or is displayed on the screen if no file is specified with the -l or --location option.</p> <p>This setting is not case-sensitive. This setting is mandatory.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>When running download command to download a wallet and store it as an auto-login Oracle wallet, ensure that ORACLE_HOME environment variable is set. If the ORACLE_HOME environment variable is not set, then the Oracle Key Vault endpoint utility is unable to find the orapki utility and show an error Missing Auto-Login Utility.</p> </div>
-g, --group	<p>Is the name of a virtual wallet from which you download an item for the WALLET, JKS, and JCEKS types. The virtual wallet must already exist, and the user must have authorization to access it. The okvutil utility downloads the entire virtual wallet specified by the -g option, and stores it in a new wallet. There must be no existing wallet at the specified location. The okvutil utility will create one. okvutil prompts you to create and enter a password for the new wallet. Record this password for the future. Remember that the group name is case-sensitive.</p> <p>If the type is WALLET, JKS, or JCEKS, then you can either include or omit the group setting. If the type is SSH, KERBEROS, or OTHER, then you must include the object_id option, but not include the group setting.</p> <p>In a multi-master cluster, only the default wallet assigned to the endpoint can be specified when the name status is PENDING.</p> <p>If the type is SSH_PUBLIC_KEY, then you need to specify the Wallet of type SSH server holding the SSH public key you are downloading.</p>
-i, --item	<p>Refers to the unique ID of the object that you want to download, such as secrets (for example, -i oracle.security.client.password1 for the first secure external password store (SEPS) entry inside a wallet).</p>
-F, --fingerprint	<p>Refers to the fingerprint of the SSH public key you want to download from the wallet of type SSH server. This option is mandatory if type is SSH_PUBLIC_KEY.</p>

Table B-4 (Cont.) okvutil download Command Options

Parameter	Description
-H, --ssh-server-host-user	Refers to the host user on the SSH server who is requesting the SSH public key. This option is mandatory if type is SSH_PUBLIC_KEY.
-o, --overwrite	Downloads data into an existing WALLET, JKS, or JCEKS file specified by -l, which must exist. If a conflict arises between the data to download and the data that already exists in the container, then the new data overwrites the old data. The -o, --overwrite option does not apply to the other types (SSH, KERBEROS, and OTHER). Use care if you plan to specify this option. If you omit the o or overwrite option when you download wallets that already exist in the current directory, then the original wallet file is renamed to either ewallet.pl2.timestamp.bak or owallet.sso.timestamp.bak before the new wallet file is downloaded. For files that are not wallets (such as Java keystore files), an error appears, and you will need to rename the file or move it to a new location before performing the download.
-v, --verbose	Refers to the verbosity level from 0 (none), 1 (debug), 2 (detailed debug).

Example: Downloading a Virtual Wallet to a Java Keystore

The following example shows how to use the `okvutil download` command to download the Oracle Key Vault virtual wallet `FinanceWallet` to a Java keystore. This is useful if you are sharing the same Java key store across multiple application servers and want to use the same wallet.

```
$ okvutil download -l ./fin/okv/work -t JCEKS -g FinanceWallet
```

The command will prompt for a new password for the Java Keystore as follows:

```
Enter new Java keystore password:
Confirm new Java keystore password:
Download succeeded
```

Related Topics

- [okvutil list Command](#)
The `okvutil list` command lists the available security objects that are uploaded.
- [Downloading Oracle Wallets](#)
The `okvutil download` command downloads an Oracle wallet from the Oracle Key Vault server to an endpoint.

okvutil list Command

The `okvutil list` command lists the available security objects that are uploaded.

When used without options or with the `-g group` option, it displays the unique ID, object type, and a descriptor for each item it lists from Oracle Key Vault.

Syntax

Short format:

```
okvutil list [-l location -t type | -g group] [-v verbosity_level] [-a]
```

Long format:

```
okvutil list [--location location --type type | --group group]
[--verbose verbosity_level][ --additional-attributes]
```

Parameters

Table B-5 okvutil list Command Options

Parameter	Description
-l, --location	Specifies the location of an Oracle wallet file or a Java keystore. For an Oracle wallet, the location is the directory that contains the .p12 or .sso files. For all other types, the location is the path name of the file itself. If you omit the -l, --location option, then the default location is Oracle Key Vault. In this case, the okvutil list command lists all the available keys in the server. If you use this setting, then you must also include the -t, --type setting, described next.
-t, --type	Specifies one of the following types: <ul style="list-style-type: none"> • WALLETT for an Oracle wallet • JKS for the Java keystore • JCEKS for the Java Cryptography Extension keystore (JCEKS) • OKV_PERSISTENT_CACHE for the persistent cache of Oracle Key Vault (Note that this setting becomes unavailable if EXPIRE_PKCS11_PERSISTENT_CACHE_ON_DATABASE_SHUTDOWN has been set for a given endpoint database, or when this endpoint has been set to use non-extractable TDE keys). <p>The WALLETT, JKS, and JCEKS types are containers for security objects which Oracle Key Vault lists individually. The SSH, KERBEROS, and OTHER are opaque objects, and are listed as single files.</p> <p>This setting is not case-sensitive.</p>
-g, --group	Lists the content from a single virtual wallet. This option only applies when you omit the -l, --location option to list the objects stored in Oracle Key Vault. Only the default wallet assigned to the endpoint can be specified when the name status is PENDING.
-v, --verbose	Refers to the verbosity level from 0 (none), 1 (debug), 2 (detailed debug).
-a, --additional-attributes	Shows the Extractable setting of a symmetric or private key.

Example: Listing Security Objects for the Current Endpoint

The following example shows how to use the okvutil list command to list all the authorized security objects for the current endpoint. In the last three lines, the DB Connect Password entries refer to the password that was used to log in to the instance (for example, the password for user psmith on the database instance inst01).

```
$ okvutil list
Enter Oracle Key Vault endpoint password: password

Unique ID                               Type           Identifier
F63E3F4A-C8FB-5560-E043-7A6BF00AA4A6   Symmetric Key  TDE Master Key:
062C4F5BAC53E84F2DBF95B96CE577B525
F63E3F4A-C8FC-5560-E043-7A6BF00AA4A6   Symmetric Key  TDE Master Key:
069A5253CF9A384F61BFDD9CC07D8A6B07
F63E3F4A-C8FD-5560-E043-7A6BF00AA4A6   Opaque Object  -
F63E3F4A-C8FE-5560-E043-7A6BF00AA4A6   Symmetric Key  TDE Master Key:
06A66967E70DB24FE6BFD75447F518525E
F63E3F4A-C8FF-5560-E043-7A6BF00AA4A6   Symmetric Key  TDE Master Key:
0636D18F2E3FF64F7ABF80900843F37456
```

```

F63E3F4A-C900-5560-E043-7A6BF00AA4A6 Opaque Object -
F63E3F4A-C901-5560-E043-7A6BF00AA4A6 Symmetric Key TDE Master Key:
0611E6ABD666954F2FBF8359DE172BA787
F63E3F4A-C902-5560-E043-7A6BF00AA4A6 Symmetric Key TDE Master Key:
0657F27D64D1C04FAEBFE00B5105B3CBAD
F63E3F4A-C91B-5560-E043-7A6BF00AA4A6 Opaque Object Certificate Request
F63E3F4A-C91C-5560-E043-7A6BF00AA4A6 Certificate X509 DN:OU=Class 1 Public
Primary Certification Authority,O=VeriSign\, Inc.,C=US
F63E3F4A-C903-5560-E043-7A6BF00AA4A6 Secret Data DB Connect Password:
psmith@inst01
F63E3F4A-C904-5560-E043-7A6BF00AA4A6 Secret Data DB Connect Password:
jdaley@inst02
F63E3F4A-C905-5560-E043-7A6BF00AA4A6 Secret Data DB Connect Password:
tjones@inst03

```

Example: Listing the Contents of an Oracle Wallet File

This example shows the contents of an Oracle wallet file.

```

$ okvutil list -t WALLET -l /home/oracle/wallets
Enter target wallet password: Oracle_wallet_password

Dumping secret store of wallet:
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY
ORACLE.SECURITY.DB.ENCRYPTION.Aa4JEUaCeE8qv0Dsmmwe5S4AAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.ID.ENCRYPTION.
ORACLE.SECURITY.KB.ENCRYPTION.
ORACLE.SECURITY.TS.ENCRYPTION.BZuIPES7+k/tv0Zw0lDeIp4CAwAAAAAAAAAAAAAAAAAAAAAA
Dumping cert store of wallet:

There are 1 Certificate Requests in the list

Certificate request:
  DN: CN=oracle
  Type: NZDST_CERT_REQ
  PUB key size: 2048

There are 0 Certificates in the list

There are 0 TPs in the list

```

okvutil upload Command

The `okvutil upload` command uploads security objects to Oracle Key Vault.

These security objects can be Oracle wallets including auto-login wallets, Java keystores, credential files, user-defined keys, and other types of key storage files.

You can upload Oracle wallets from all currently supported releases of Oracle Database and other Oracle software products that use Oracle wallets. The `okvutil upload` command opens the wallet or Java keystore and uploads each item found as an individual security object into Oracle Key Vault. If you are uploading credential files, then Oracle Key Vault uploads them as whole files called opaque objects.

Syntax

Short format:

```

okvutil upload -l <location> -t <type> [ -o | -U <SSH-user> -L <length> [-i <SSH-private-key-id>] [-g <group>] [-d <description>][-v <verbosity_level>]

```

Long format:

```
okvutil upload --location location --type type [--overwrite | --ssh-user <SSH-user> --
length <length> [--item <SSH-private-key-id>] [--group group]
[--description description] [--verbose verbosity_level]
```

Parameters

Table B-6 okvutil upload Command Options

Parameter	Description
-o, --overwrite	If there are conflicts with the existing data in the Oracle Key Vault virtual wallet, then Key Vault replaces the existing data with new data that is sent by the endpoint. If there are no conflicts, then the overwrite operation is not necessary and is not performed. Use care if you plan to specify this option.
-l, --location	Specifies the location of an Oracle wallet file, Java keystore, or a text file containing user-defined and hex-encoded TDE master encryption identifier and key. For an Oracle wallet, the location is the directory that contains the .p12 or .sso files. If you are uploading a credential file as an opaque object, then ensure that this file is no larger than 120 kilobytes (KB).
-t, --type	Specifies the data type of the object being uploaded to Oracle Key Vault. It must be a value from the following list: <ul style="list-style-type: none"> • WALLET for an Oracle wallet • JKS for a Java keystore • JCEKS for a Java Cryptography Extension keystore (JCEKS) • SSH for an SSH key file, to be uploaded as an opaque object. The maximum size is 120 KB. • KERBEROS for a Kerberos keytab, to be uploaded as an opaque object. The maximum size is 120 KB. • TDE_KEY_BYTES for a user-defined key to be used as a TDE master encryption key. • SSH_PUBLIC_KEY for the SSH public key. • SSH_PRIVATE_KEY for the SSH private key. • OTHER for opaque objects, which are other files that store secrets. The maximum size is 120 KB. <p>The WALLET, JKS, and JCEKS types contain multiple objects. Oracle Key Vault uploads each of these objects individually. The SSH, KERBEROS, TDE_KEY_BYTES, SSH_PUBLIC_KEY, SSH_PRIVATE_KEY, and OTHER types, being opaque objects, are uploaded as single files.</p> <p>This setting is not case-sensitive.</p>
-g, --group	Is the name of a Key Vault virtual wallet to which the certificate store or secret store (or both) are added. This name is case-sensitive. The virtual wallet must already exist, and the user must have authorization to access it. If you omit this setting, then the default group, if there is one, is used. If there is no default group and you omit the -g, --group option, then the data uploaded will not be placed in a group. Only the default wallet assigned to the endpoint can be specified when the name status is PENDING.
-d, --description	Enables you to add a description, up to 2000 bytes. It is valid only if the -t type, --type parameter is set to SSH, KERBEROS, TDE_KEY_BYTES, or OTHER. Optional. Enclose this description in double quotation marks. If there are spaces within this description, then include escape characters with the quotation marks. For example: -d \"text with spaces\"
-v, --verbose	Refers to the verbosity level from 0 (none), 1 (debug), 2 (detailed debug).

Table B-6 (Cont.) okvutil upload Command Options

Parameter	Description
-i, --item	Refers to the unique ID of the SSH private key to be linked to the SSH public key on upload. This option is only valid if the type is SSH_PUBLIC_KEY.
-U, --ssh-user	Refers to the name of SSH user who owns the SSH public or private key. This option is only valid if the type is SSH_PUBLIC_KEY or SSH_PRIVATE_KEY.
-L, --length	The length (in bits) of the SSH public or private key to be uploaded. This option is only valid if the type is SSH_PUBLIC_KEY or SSH_PRIVATE_KEY.

Example: Uploading a Java Keystore Using the -v 2 Option

The following example shows how to use the `okvutil upload` command to upload a Java keystore. The `-v 2` option enables the command to list the items that are uploaded. The `okvutil` command prompts if necessary for passwords to connect to Oracle Key Vault and to open the Oracle wallet file.

```
$ okvutil upload -l ./fin_jceks.jck -t JCEKS -g fin_wal -v 2

okvutil version 21.5.0.0.0
Configuration file: /tmp/fin_okv/conf/okvclient.ora
Server: 192.0.2.254:5696
Standby Server: 127.0.0.1:5696
Uploading from /tmp/fin_okv/keystores/jks/keystore.jks
Enter source Java keystore password:
Uploading private key
Uploading trust point
Uploading trust point
Uploading private key
Uploading private key

Uploaded 3 private keys
Uploaded 0 secret keys
Uploaded 2 trust points

Upload succeeded
```

Example: Uploading a Password-Protected Wallet File

The following example shows how to use the `okvutil upload` command to upload a password-protected wallet file when there is no password for the endpoint to connect to Oracle Key Vault.

```
$ okvutil upload -l . -t WALLET -g FinanceWallet
Enter source wallet password: password

Upload succeeded
```

Example: Uploading a User-Defined Key to Use as a TDE Master Encryption Key

The following example shows how to upload a user-defined key.

```
$ okvutil upload -l /tmp/tde_key_bytes.txt -t TDE_KEY_BYTES -g
"FIN_DATABASE_VIRTUAL_WALLET" -d "\"This key was created for Financial database use on
1st April 2020\""
```

Related Topics

- [Uploading Oracle Wallets](#)
The `okvutil upload` command uploads wallets to Oracle Key Vault.
- [Step 1: Upload the User-Defined Key](#)
Use the `okvutil upload` command to upload user-defined master encryption keys to Oracle Key Vault.
- [Adding Security Objects to a Virtual Wallet](#)
You can add new security objects to a virtual wallet at any time as needed.
- [Uploading Oracle Wallets](#)
The `okvutil upload` command uploads wallets to Oracle Key Vault.

okvutil sign Command

The `okvutil sign` command generates a digital signature for a message or message digest by using the private key stored on the Oracle Key Vault server.

The `okvutil` utility can sign a message of length up to 32768 when provided on CLI. The `okvutil` can also sign a data of any size when provided in the form of a file. In addition to the signing message or the file in raw format it can also sign a message digest.

Syntax

Short format:

```
okvutil sign [-l | -m] [-M ] [-D ] [-i | -n ]
```

Long format:

```
okvutil sign [-l <location> | -m <message>] [-M <message-type>] [-D <digital-signature-  
algorithm>] [-i <object-id> | -n <name>]
```

Parameters

Table B-7 okvutil sign Command Options

Parameter	Description
-l	Specifies the file location containing the data.
-m	Specifies message to be signed.
-M	Specifies whether the message provided with the -m option is raw or digest (already hashed). The default value is RAW. The allowed values are: <ul style="list-style-type: none"> • RAW • DIGEST
-D	Specifies the digital signature algorithm. The default values is RSASSA_PKCS1_v1_5_SHA256. The allowed values are: <ul style="list-style-type: none"> • RSASSA_PSS_SHA256 • RSASSA_PSS_SHA384 • RSASSA_PSS_SHA512 • RSASSA_PKCS1_v1_5_SHA256 • RSASSA_PKCS1_v1_5_SHA384 • RSASSA_PKCS1_v1_5_SHA384 • RSASSA_PKCS1_v1_5_SHA384 • RSASSA_PKCS1_v1_5_SHA512

Table B-7 (Cont.) okvutil sign Command Options

Parameter	Description
-i	Specifies the KMIP ID (UUID) of the private key used for signing. This KMIP ID (UUID) cannot be used with option -n.
-n	Specifies the user-supplied name to locate private key that is used for signing. This cannot be used with option -i.

Example 2

The following example shows how to use the `okvutil sign` command.

```
okvutil sign -l ./lib/liborapkcs.so -i 78916DFB-CC03-4FDB-BFF8-F25C1D846EF6
Enter Oracle Key Vault endpoint password:
79A013DE36E7DBDCD718C37A5510C760429FDE782224F4A7D1442D5C34
64A03C04C6A75676985510D05F225C3E3C2D1484F9ADFE4D59AE8D984F
6C059491FE9D856DC7018C77128EF5EBDE54A0F678C10B33C2C6357BF7
E67895FE235D90F2343AEA5A925DA0D266E27BC5F261C94F2AA3A180C5
26D730BB3540D88967695047070CEB41D325F9E618BF628698710586BC5
C7D1C635E35E5B1B28DA10CD3D1575FA66E2C4BBD5B6DAC0B7AAD83D
69C390491BA4F7763BC6880FD214B07394970322A115C368CB0901FA36B
4131ACF53B635661DED63018BECB18E28BE3D33B7217092DEFA54437DD
46E6FA72891B3B157183CD51920BEF9D154243AAEC9379B
```

Example 3

The following example shows how to use the `okvutil sign` command.

```
okvutil sign -m
\"a152b1752be70662511cd615d4a2e8a9503f7a19ce6f8415ddee8024e56001ec\" -M DIGEST -D
RSASSA_PKCS1_v1_5_SHA256 -n private_key
Enter Oracle Key Vault endpoint
password:
79A013DE36E7DBDCD718C37A5510C760429FDE782224F4A7D1442D5C34
64A03C04C6A75676985510D05F225C3E3C2D1484F9ADFE4D59AE8D984F
6C059491FE9D856DC7018C77128EF5EBDE54A0F678C10B33C2C6357BF7
E67895FE235D90F2343AEA5A925DA0D266E27BC5F261C94F2AA3A180C5
26D730BB3540D88967695047070CEB41D325F9E618BF628698710586BC5
C7D1C635E35E5B1B28DA10CD3D1575FA66E2C4BBD5B6DAC0B7AAD83D
69C390491BA4F7763BC6880FD214B07394970322A115C368CB0901FA36B
4131ACF53B635661DED63018BECB18E28BE3D33B7217092DEFA54437DD
46E6FA72891B3B157183CD51920BEF9D154243AAEC9379B
```

 **Note:**

In a password-protected endpoint, the `okvutil sign` command does not display the password prompt for the endpoint password, when output is redirected to a file. Since you need to enter the endpoint password, do not redirect the output.

okvutil sign-verify Command

The `okvutil sign-verify` command verifies digital signature of a message or message digest using the public key or certificate stored on Oracle Key Vault Server.

The `okvutil utility` can verify digital signature of a message of length up to 32768 when provided on CLI and data of any size when provided in the form of a file. In addition to verifying digital signature of message or data in raw format `okvutil utility` can also verify digital signature of a message digest.

Syntax

Short format:

```
okvutil sign-verify [-l | -m ] [-M ] [-D ] -S [-i | -n ]
```

Long format:

```
okvutil sign-verify [-l <location> | -m <message>] [-M <message-type>] [-D <digital-signature-algorithm>] -S <signature> [-i <object-id> | -n <name>]
```

Parameters

Table B-8 okvutil sign-verify Command Options

Parameter	Description
-l	Specifies the file location containing the data.
-m	Specifies the message for which to verify the signature.
-M	Specifies whether the message provided with the -m option is raw or digest (already hashed). The default value is RAW. The allowed values are: <ul style="list-style-type: none"> RAW DIGEST
-D	Specifies the digital signature algorithm. The default values is RSASSA_PKCS1_v1_5_SHA256. The allowed values are: <ul style="list-style-type: none"> RSASSA_PSS_SHA256 RSASSA_PSS_SHA384 RSASSA_PSS_SHA512 RSASSA_PKCS1_v1_5_SHA256 RSASSA_PKCS1_v1_5_SHA384 RSASSA_PKCS1_v1_5_SHA384 RSASSA_PKCS1_v1_5_SHA512
-S	Specifies the file containing signature or the signature itself. If the value is a path to a valid file then the signature is read from the file otherwise the value is treated as signature itself.
-i	Specifies the KMIP ID (UUID) of the private key used for signing the file. This KMIP ID (UUID) cannot be used with option -n.
-n	Specifies the user-supplied name to locate private key that is used for signing the file. This cannot be used with option -i.

Example 1

The following example shows how to use the `okvutil sign-verify` command.

```
okvutil sign-verify -l
./lib/liborapkcs.so -n public_key -S
79A013DE36E7DBDCD718C37A5510C760429FDE782224F4A7D1442D5C34
64A03C04C6A75676985510D05F225C3E3C2D1484F9ADFE4D59AE8D984F
6C059491FE9D856DC7018C77128EF5EBDE54A0F678C10B33C2C6357BF7
E67895FE235D90F234AEA5A925DA0D266E27BC5F261C94F2AA3A180C5
26D730BB3540D88967695047070CEB41D325F9E618BF628698710586BC5
C7D1C635E35E5B1B28DA10CD3D1575FA66E2C4BBD5B6DAC0B7AAD83D
69C390491BA4F7763BC6880FD214B07394970322A115C368CB0901FA36B
4131ACF53B635661DED63018BECB18E28BE3D33B7217092DEFA54437DD
46E6FA72891B3B157183CD51920BEF9D154243AAEC9379B
Enter Oracle Key Vault endpoint password:
Signature Validity:Valid
```

Example 2

The following example shows how to use the `okvutil sign-verify` command.

```
okvutil sign-verify -m
\"a152b1752be70662511cd615d4a2e8a9503f7a19ce6f8415ddee8024e56001ec\" -M DIGEST -D
RSASSA_PKCS1_v1_5_SHA256 -i 5AFBC939-73D2-4F57-BF9E-8D253AEDCD8B -S
./digital_signature_file
Enter Oracle Key Vault endpoint password:
Signature Validity:Valid
```

Note:

If your message includes spaces, ensure to enclose the message using the escape character within the quotes. For example, `./bin/okvutilsign -i 3FF9E715-7648-4F4A-BF80-E3EC6543F0A0 -m \"Oracle key Vault\"`.

Example 3

The following example shows how to use `openssl` to verify signature generated by `okvutil sign` command.

1. `openssl` does not support verification of signatures in HEX format. As the signature generated by `okvutil sign` command is in HEX format therefore the signature can be converted into binary format using `xxd` or some other utility.

```
xxd -r -p ./digital_signature_file > ./digital_signature_file.bin
```

2. Verify signature in binary format using `openssl` and public key stored in file `key.pub`.

```
openssl dgst -sha256 -verify key.pub
-signature ./digital_signature_file.bin ./lib/liborapkcs.so
Verified OK
```

okvutil show Command

The `okvutil show` command displays the endpoint information.

The endpoint information displayed by this command includes the name, name status, unique identifier, description, default wallet, user who created the endpoint, creator node, cluster subgroup to which the endpoint belongs to, certificate expiration and common name of the certificate issuer.

For a stand-alone or HA Oracle Key Vault set-up, creator node value is displayed as '-' and cluster subgroup value as '**No Cluster Subgroup (from Creator Node)**' since these two values are applicable only in the cluster mode.

Syntax

Short format:

```
okvutil show -e
```

Long format:

```
okvutil show --endpoint
```

Parameters

Table B-9 okvutil show Command Options

Parameter	Description
-e, --endpoint	Specify this option with the <code>show</code> command, to view the endpoint information to which <code>okvutil</code> belongs. This parameter is mandatory to view the endpoint information.

Example

The following example shows how to use the `okvutil show` command.

```
$ ./okvutil show --endpoint
Enter Oracle Key Vault endpoint password:

Endpoint Information
Name: ORACLEDB_EP
Name Status: ACTIVE
Unique ID: F63E3F4A-C8FB-5560-E043-7A6BF00AA4A6
Description: Endpoint for Oracle Database for storing TDE Master Keys
Default Wallet: tdewallet
Created By: OKVADMIN
Creator Node: node1
Cluster Subgroup: boston_subgroup
Certificate Expiration: 30-MAY-2026 08:38:56
Common Name Of Certificate Issuer: CA
```

okvutil Common Errors

The `okvutil` common errors describes the error conditions and the reason for their occurrence.

The `okvutil` utility displays a message when an error condition is established.

Table B-10 okvutil Common Errors

Error Code	Description
10115	Displayed when the server is down or the port in the <code>okvclient.ora</code> configuration file is incorrect.
28759	Displayed when the SSL wallet location is incorrect.

Table B-10 (Cont.) okvutil Common Errors

Error Code	Description
28791	Displayed when the certificate cannot be verified due to certification error.
29106	Displayed when the password is incorrect.

C

Troubleshooting Oracle Key Vault

Oracle provides checklists, tips, instructions, and how-tos for common errors to help you smoothly install and deploy Oracle Key Vault.

- [Before You Start Troubleshooting](#)
Learn how to use the endpoint health check utility and troubleshoot the Oracle Key Vault server issues.
- [Common Oracle Key Vault Tasks](#)
Review these tasks for resolving common issues encountered when working with Oracle Key Vault.
- [okvutil and Endpoint Issues](#)
Learn how to run the endpoint health check utility to triage endpoint related issues
- [Multi-Master Cluster Issues](#)
Review these troubleshooting tips for common Multi-Master Cluster related errors when working with Oracle Key Vault.
- [Backup and Restore Issues](#)
Review these troubleshooting tips for common backup and restore related issues when working with Oracle Key Vault.
- [Certificate Related Issues](#)
Review these troubleshooting tips for common certificate-related issues when working with Oracle Key Vault.
- [Installation and Upgrade Issues](#)
Review these troubleshooting tips for common installation and upgrade issues when working with Oracle Key Vault.
- [Primary-Standby Configuration Issues](#)
Review these troubleshooting tips for commonly encountered primary-standby configuration related issues when working with Oracle Key Vault.
- [DBCS Endpoint Configuration Issues](#)
Review these troubleshooting tips for commonly encountered DBCS endpoint configuration related issues when working with Oracle Key Vault.
- [Server and Node Issues](#)
Review these troubleshooting tips for common server and node related errors when working with Oracle Key Vault.

Before You Start Troubleshooting

Learn how to use the endpoint health check utility and troubleshoot the Oracle Key Vault server issues.

- [Endpoint Related Issues](#)
To identify endpoint-related issues, use the endpoint health check utility.

Endpoint Related Issues

To identify endpoint-related issues, use the endpoint health check utility.

The endpoint health check utility performs several checks for the endpoint environment and configuration settings, identifies probable causes, and recommends possible actions to fix the issues.

- Verifies if the environment variables `ORACLE_HOME`, `ORACLE_SID`, `OKV_HOME`, `ORACLE_BASE`, and `JAVA_HOME` are set in the current session and `gen0` process.
- Verifies if multiple HSM libraries exist in the environment.
- Verifies if the symbolic link is created for `okvclient.ora` under `ORACLE_BASE/okv/$ORACLE_SID`.
- Verifies if the `wallet_root` path is set, if set, then verifies if the `okv` and `tde` folders are configured.
- Verifies if the database is up and running.
- Verify if the SSL path is set correctly in `okvclient.ora`.
- Verifies the permissions of the `liborapkcs` file under `/opt/oracle` path.
- Verifies the connectivity to the Oracle Key Vault server and provides solutions to fix the error.
- Verifies if the current Oracle Key Vault version is supported.

To run the endpoint health check script for Oracle Key Vault releases 21.5 or later:

1. Log in to the endpoint DB server.
2. Go to the `OKV_HOME` directory where `okvclient.jar` is installed.
3. Go to the `bin` directory under `OKV_HOME`.
4. Run the health check script.

```
./ep_healthcheck.sh
```

5. Follow the on-screen instructions to run the endpoint health check utility.
6. After the health check is complete, open the log file mentioned in the output.
7. Read the log file to find if any tests have FAILED. Fix the failed tests using the instructions in the log file or with the help of the Troubleshooting section.
8. After fixing the failed tests, re-run the health check utility and verify all tests are passed.



Note:

For Oracle Key Vault release 21.4 or earlier, use the instructions in [My Oracle Support Note 2859388.1](#).

Common Oracle Key Vault Tasks

Review these tasks for resolving common issues encountered when working with Oracle Key Vault.

- [How to Re-Enroll an Endpoint on an Endpoint Database](#)
You can re-enroll an endpoint again on an endpoint database using these steps.
- [How To Download Diagnostics From Oracle Key Vault Server](#)
Downloading the Oracle Key Vault diagnostics log for 21.5 or previous versions provides troubleshooting information for Oracle Key Vault issues.
- [How to Recover the root User Password](#)
You can reset the Oracle Key Vault's root user password when the current root password is forgotten using Oracle Key Vault server's terminal console.
- [How to Reset the support User Password](#)
You can reset the Oracle Key Vault support user password when the current password is forgotten using Oracle Key Vault server's console terminal.
- [How to Add SAN Details to the Console Certificate](#)
You can add SAN details in console certificate using these steps.
- [How to Check for the Patch Updates](#)
Check if additional patches are applicable to your environment. Oracle recommends that you check for patches after installing or upgrading to the Oracle Key Vault release.

How to Re-Enroll an Endpoint on an Endpoint Database

You can re-enroll an endpoint again on an endpoint database using these steps.

1. Log in to the Oracle Key Vault management console as a user who has either the system administrator role or privilege to manage the endpoint.
2. Navigate to the **Endpoints** tab.
3. Select the endpoint and re-enroll the endpoint.
4. Download the `okvclient.jar` file using the endpoint enrollment token.
5. Securely transfer `okvclient.jar` using `SCP` to the endpoint database server.
6. Set the `ORACLE_BASE`, `ORACLE_HOME`, `ORACLE_SID`, `JAVA_HOME`, and `OKV_HOME` environment variables as required.
7. Back up the `$OKV_HOME` directory and delete the files under `$OKV_HOME`:

```
$cp -R $OKV_HOME $OKV_HOME_bkp_date +%Y%m%d
```

8. Go to the `$OKV_HOME` directory and remove all the files.
9. Verify if the `okvclient.ora` file exists in `$ORACLE_BASE/okv/$ORACLE_SID`. If the file exists, rename it.

```
cd $ORACLE_BASE/okv/$ORACLE_SID
ls -ltr
mv okvclient.jar okvclient.jar_bkp_date +%Y%m%d
rm okvclient.lck
rm okv.pc.lck
```

10. Install the new `okvclient.jar` file:

```
$JAVA_HOME/bin/java -jar okvclient.jar -d $OKV_HOME -v -o
```

How To Download Diagnostics From Oracle Key Vault Server

Downloading the Oracle Key Vault diagnostics log for 21.5 or previous versions provides troubleshooting information for Oracle Key Vault issues.

To download the Oracle Key Vault diagnostics log requested by Oracle Support:

1. SSH into the Oracle Key Vault system as user `support`, then switch to user `root`:

```
ssh support@<OKV_IP_Address>  
su - root
```

2. Install the diagnostics package:

```
/usr/local/dbfw/bin/priv/dbfw-diagnostics-package.rb --install
```

3. Modify the diagnostics configuration to allow the utility to collect information about the appliance. Enable collection of all elements and files:

```
/usr/local/dbfw/bin/priv/dbfw-diagnostics-package.rb --enable ALL
```

4. Copy the diagnostics file outside Oracle Key Vault before deleting them.

```
scp/usr/local/dbfw/tmp/diagnostics_okv*.zip user@hostname:/tmp
```

5. Collect the diagnostic logs:

```
/usr/local/dbfw/bin/priv/dbfw-diagnostics-package.rb
```

For example, a diagnostics file looks like:

```
/usr/local/dbfw/tmp/  
diagnostics_okv0039f6043e9a_2017_06_08_17_48_07_75F4728E5644D781A215200EAAE  
ADF3B.zip
```

6. Delete the generated diagnostics zip file and remove the package using the following commands:

```
/usr/local/dbfw/bin/priv/dbfw-diagnostics-package.rb --clean  
/usr/local/dbfw/bin/priv/dbfw-diagnostics-package.rb --remove
```

How to Recover the root User Password

You can reset the Oracle Key Vault's root user password when the current root password is forgotten using Oracle Key Vault server's terminal console.

To reset the root user password of Oracle Key Vault Server:

1. Reboot the Oracle Key Vault server.

On the terminal console, when the **GRUB2** menu appears with the menu item for the Oracle Key Vault server entry, enter the edit mode by pressing `e` key.

```
Oracle Key Vault Server 21.8.0.0.0 [5.4.17-2136.328.3.el8uek.x86_64]
Oracle Key Vault Server 21.8.0.0.0 [5.4.17-2136.328.3.el8uek.x86_64] (rescue mode)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

The **GRUB** edit mode interrupts the boot process to display the kernel boot parameters as shown in the following image:

```
setparams 'Oracle Key Vault Server 21.8.0.0.0 [5.4.17-2136.328.3.el8uek.x86_64]'

load_video
set gfxpayload=keep
insmod gzio
insmod part_msdos
insmod ext2
# Search for root by UUID
search --no-floppy --fs-uuid --set=root df107610-b1c1-4de7-a1f0-67fdd856a8e5
linuxefi /vmlinuz-5.4.17-2136.328.3.el8uek.x86_64 root=/dev/mapper/vg_root-lv_ol8root ro crashkernel=auto rd.lvm.lv=vg\
root/lv_ol8root rd.lvm.lv=vg_root/lv_swap LANG=en_US.UTF-8 noip6 crashkernel=auto boot=UUID=df107610-b1c1-4de7-a1f0-67fdd85\
6a8e5
initrd /initramfs-5.4.17-2136.328.3.el8uek.x86_64.img
```

2. Go to the end of the line that starts at `linuxefi`. Press `Ctrl-e` to jump to the end of a line.
3. Enter `rd.break` to the end of the line that starts with `linux16`.
4. Press `Ctrl-x` to continue the boot process with changed kernel parameters. The **switch_root:/#** prompt displays.
5. Remount the file system with read write access to change the root password. By default, the file system is mounted as read-only at `/sysroot`.
6. Run the given command on the **switch_root:/#** prompt to make `/sysroot` writable:

```
mount -o remount,rw /sysroot
```

7. To enter `chroot` environment, run the following command:

```
chroot /sysroot
```

8. Use the `passwd` command to set the new root password. Follow the prompts to complete the password change.

```
passwd
```

9. Force SELinux file system relabeling process on the next system reboot:

```
touch /.autorelabel
```

Make sure the dot appears after the forward-slash.

10. Exit the `chroot` environment.

```
exit
```

11. Exit the `switch_root` prompt.

```
exit
```

The boot process continues. Wait for few minutes to complete the SELinux relabeling process. The system reboots automatically when the relabelling process is completed.

12. Once the system reboots, the root password resetting is completed.
13. Login with your new root password.

How to Reset the support User Password

You can reset the Oracle Key Vault support user password when the current password is forgotten using Oracle Key Vault server's console terminal.

To change the OS user support password on the Oracle Key Vault console terminal:

1. Login as root user.

Note:

You can login as root using console terminal.

2. Run the `passwd support` command to reset the support user password.

```
[root@okvserver ~]# passwd support
Changing password for user support.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@okvserver ~]#
```

There are no consequence in changing the support password.

After the password is set successfully, the following message is displayed on the console:

```
All authentication tokens updated successfully.
```

3. Login as support.

How to Add SAN Details to the Console Certificate

You can add SAN details in console certificate using these steps.

Example

In Oracle Key Vault, after uploading the console certificate, the browser displays **Invalid Certificate** due to no SAN details.

Probable Cause

- Currently, you cannot add the SAN details while generating console certificate from the Oracle Key Vault Management web console.

- SAN details cannot be added into console certificate in Oracle Key Vault 21.7 or previous versions.

Solution

1. Login with support user to the Oracle Key Vault server through ssh and switch to root user.
2. Create a temporary directory under /tmp directory like, `mkdir /tmp/console_cert`.

```
chmod 755 /tmp/console_cert
cp/etc/pki/tls/private/user_uploaded_ui.key public.key
```

3. Copy the configuration code.

```
consolecert.conf:
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req

[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = US
localityName = Locality Name (eg, city)
organizationalUnitName = Organizational Unit Name (eg, section)
commonName = Common Name (eg, YOUR name)
commonName_max = 64
emailAddress = Email Address
emailAddress_max = 70

[v3_req]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[alt_names]
DNS.1 = [eg. oracle.com]
IP.1 = [ip address]
```

4. Create `consolecert.conf` in `/tmp/console_cert`.
5. Edit the `consolecert.conf` file and add the `alt_names` under `alt_name` record.
6. Generate the **csr** using the given command.

```
openssl req -new -key public.key -out console_cert_okvname.csr -config
consolecert.conf
```

- a. Sign the generated `csr`,
7. Go to the console certificate page in the Oracle Key Vault management console.
 8. Click **Upload Certificate** to upload the signed certificate.

How to Check for the Patch Updates

Check if additional patches are applicable to your environment. Oracle recommends that you check for patches after installing or upgrading to the Oracle Key Vault release.

To download and update the patches for Oracle Key Vault:

1. Log in to <https://support.oracle.com>
2. Select the **Patches & Updates** tab.
3. Select **Product or Family (Advanced)** tab.

 **Note:**

By default, the **Number/Name or Bug Number (simple)** option is selected.

4. Select Oracle Key Vault from the **Product is** drop-down list.
5. Select the current release number from the **Release is** drop-down list.
6. Click **Search**. The search results are displayed.
7. In the **Patch Name** column, click the applicable patch number to your release.

 **Note:**

You can check for the release in the **Release** column.

A corresponding **Patch number** page appears.

8. Click **Read Me** to open the readme file in a browser.
9. Click **Download** to open the **File Download** page.
10. Click the **Download File Metadata** link.
11. Click **Download** to download the XML metadata file.

okvutil and Endpoint Issues

Learn how to run the endpoint health check utility to triage endpoint related issues

- [Database Wallet Status Not Open or Not Found, TDE HEARTBEAT Check Failed](#)
Database wallet status for Oracle Key Vault is closed. When trying to open the Database wallet it fails with error ORA-28365, ORA-28407, or HSM heartbeat check failed errors.
- [Oracle Key Vault Server Communication or Connection Failed Error](#)
Oracle Key Vault displays a server communication or connection failed error when you try to fetch data using `okvutil` or when installing `okvclient.jar`.
- [Could Not Store Private Key Errors on Wallet Upload](#)
When uploading Java keystores, using `okvutil`, Oracle Key Vault displays could not store private key errors.
- [RESTful Services Endpoint Provisioning Command Failure](#)
The endpoint provisioning using RESTful services utility fails due to incorrect path or directory.

- [Uploading Certificate File Failure](#)
When uploading a console certificate an error is displayed .
- [Error in Uploading the Java Keystore](#)
Oracle Key Vault displays an error when uploading the Java keystore.
- [SSL layer Error while migrating MYSQL Database Keys to Oracle Key Vault](#)
An SSL layer error displays when the MySQL database is migrated to Oracle Key Vault.
- [Rotation or Set Key Failure in Windows Environment](#)
In Oracle Key Vault error displays while rotating or setting a new key.
- [Rotation or Set Key Fails with ORA-03113](#)
In Oracle Key Vault, while rotating the key an error displays.

Related Topics

- [Before You Start Troubleshooting](#)
Learn how to use the endpoint health check utility and troubleshoot the Oracle Key Vault server issues.

Database Wallet Status Not Open or Not Found, TDE HEARTBEAT Check Failed

Database wallet status for Oracle Key Vault is closed. When trying to open the Database wallet it fails with error ORA-28365, ORA-28407, or HSM heartbeat check failed errors.

Example

1. Failed to open keystore on endpoint DB

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
"*";
*
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "*"
*
ERROR at line 1:
ORA-28365: wallet is not open
```

2. Failed to fetch the keys from v\$encryption keys on endpoint DB.

```
SQL> SELECT KEY_ID value FROM V$ENCRYPTION_KEYS;
*
ERROR at line 1:
ORA-28407: Hardware Security Module failed with PKCS#11 error
CKR_GENERAL_ERROR(5)
```

3. Failed to open key store on endpoint DB.

```
administer key management set keystore open identified by "*";
*
ERROR at line 1:
ORA-28353: failed to open wallet
```

4. Endpoint DB alert log shows the following error messages:

```
HSM connection lost, closing wallet  
kzthsmccl: HSM heartbeat check failed
```

Probable Cause 1

When the endpoint DB does not use the *WALLET_ROOT* configuration. This can happen if the environment variables *ORACLE_BASE*, *ORACLE_HOME*, *ORACLE_SID*, *OKV_HOME*, and *JAVA_HOME* are not set.

Solution

1. For Oracle RAC environments, check if the environment variables *ORACLE_BASE*, *ORACLE_HOME*, *ORACLE_SID*, *OKV_HOME*, and *JAVA_HOME* are set in the configuration.

- a. If the variables are not set, then set them using *srvctl* environment.

```
srvctl setenv database -db db_unique_name -t  
"ORACLE_UNQNAME=db_unique_name, ORACLE_BASE=/u01/opt/oracle"
```

- b. You can restart the database instance using the *srvctl* command.

```
srvctl getenv database -db db_unique_name
```

2. If you get the error when running a third-party script for RMAN backup, then check if the *ORACLE_BASE*, *ORACLE_HOME*, *ORACLE_SID*, *OKV_HOME*, and *JAVA_HOME* environment variables are exported in the script. Also check if the user running the script has set the environment variables in the profile.
3. If the environment variables are not set, then set the variables in the user profile and script, and then run the backup.
4. If the endpoint health check utility reports that the environment variables are not set in the *gen0* environment, then set the variables using the command line and restart the database.

- a. Log in to SQLPLUS as the *SYSDBA* user and shutdown the database.

```
./sqlplus sys / as sysdba  
  
shutdown immediate
```

- b. Exit from SQLPLUS and restart the database service.

```
su - oracle  
snrctl start
```

- c. Log in to SQLPLUS and start the database.

```
startup
```

5. Check if the issue is resolved.

Probable Cause 2

Oracle Key Vault server KMIP service is not available.

Solution

1. Check if the Oracle Key Vault server can successfully list the keys:

```
$OKV_HOME/bin/okvutil list -v 4
```

2. If the keys do not list, then check the availability of the Oracle Key Vault server by accessing the Oracle Key Vault management console.
3. If the Oracle Key Vault server is up and running, then verify the KMIP service status using one of these methods:
 - If the REST services utility is configured on the endpoint DB server then use the REST services command `okv status get` to get the server information and verify the status of the `KMIP` service.
 - If the SNMP monitoring is configured, then use SNMP monitoring to find the status of the `KMIP` service.
 - Log in to the Oracle Key Vault management console as the System Administrator. On the Oracle Key Vault home page, check if there is an alert **stop responding** process alert.
 - Log in to the Oracle Key Vault management console as the System Administrator. Navigate to the **System, Status** page and check the `KMIP` service status.
4. If the `KMIP` service status is down, then follow restart the `KMIP` service:
 - a. Log in to the Oracle Key Vault server using SSH as the `support` user and then switch to the `root` user.

```
$ ssh support@okv_server_IP_address  
$ su - root
```

- b. Run the following command to verify the status of the `KMIP` process.

```
ps -eaf | grep kmip
```

- c. If the `KMIP` process is not running, then restart the `KMIP` service:

```
systemctl restart kmip
```

- d. Verify the status of the `KMIP` process:

```
ps -eaf | grep kmip
```

- e. On the endpoint DB server, check if `okvutil` lists the keys:

```
OKV_HOME/bin/okvutil list -v 4
```

- f. Check if the issue is resolved.

5. If the status of the `KMIP` service is up and running, then from the endpoint DB server run the following command:

```
curl -v telnet://OKV_SERVER_IP:5696
```

6. If this command does not connect to the Oracle Key Vault server, then contact your network administrator to verify that port 5696 is open. If not, ensure that port 5696 is open.

7. Check if the issue is resolved.

Probable Cause 3

Symbolic link to `okvclient.ora` does not exist or points to an incorrect configuration.

Solution

Perform the following steps:

1. On the endpoint DB server, verify whether the symbolic link for `okvclient.ora` exists.

```
cd $ORACLE_BASE/okv/$ORACLE_SID
ls -ltr okvclient.ora
```

OR

```
cd $ORACLE_HOME/okv/$ORACLE_SID
ls -ltr okvclient.ora
```

Command output should include the `okvclient.ora` symbolic link information, if it exists.
For example:

```
lrwxrwxrwx 1 oracle oinstall 64 Aug 3 2020 okvclient.ora -> /u02/app/
oracle/okv/conf/okvclient.ora.
```

2. If the `okvclient.ora` symbolic link does not exist or is pointing to a path other than `$OKV_HOME/conf`, then either correct the symbolic link, or reinstall the Oracle Key Vault client. See, [How to Re-Enroll an Endpoint on an Endpoint Database](#).
3. Verify that `okvclient.ora` exists and points to `$OKV_HOME/conf`.
4. Check if the issue is resolved.

Probable Cause 4

The auto-login wallet is not configured properly.

Solution

Perform the following steps:

1. Verify the wallet status by connecting to the endpoint DB and run the following SQL query.

```
select * from v$encryption_wallet
```

2. For Oracle Databases 12c and earlier:
If the auto-login is configured and wallet for OKV/HSM is not open and for FILE is open, then verify whether the auto-login wallet location is set in `sqlnet.ora` file. If not, then set the location, and restart the database (downtime required). Verify whether the issue is resolved

For Oracle database 12.1.0.2 or earlier the `sqlnet.ora` entry looks like,

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=HSM) (METHOD_DATA=(DIRECTORY=/
home/oracle/wallet_okv)))
```

For 12.2.0.1 and above the sqlnet.ora entry looks like,

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=OKV) (METHOD_DATA=(DIRECTORY=/home/oracle/wallet_okv)))
```

3. For the Oracle databases 18c or later:
If the `WALLET_ROOT` is configured, then the auto-login wallet should exist under the `WALLET_ROOT_directory/tde` directory and Oracle Key Vault endpoint software installation is under the `WALLET_ROOT_directory/okv` directory. If any of them do not exist, apply the necessary configuration changes to address this.
4. Check if the issue is resolved.

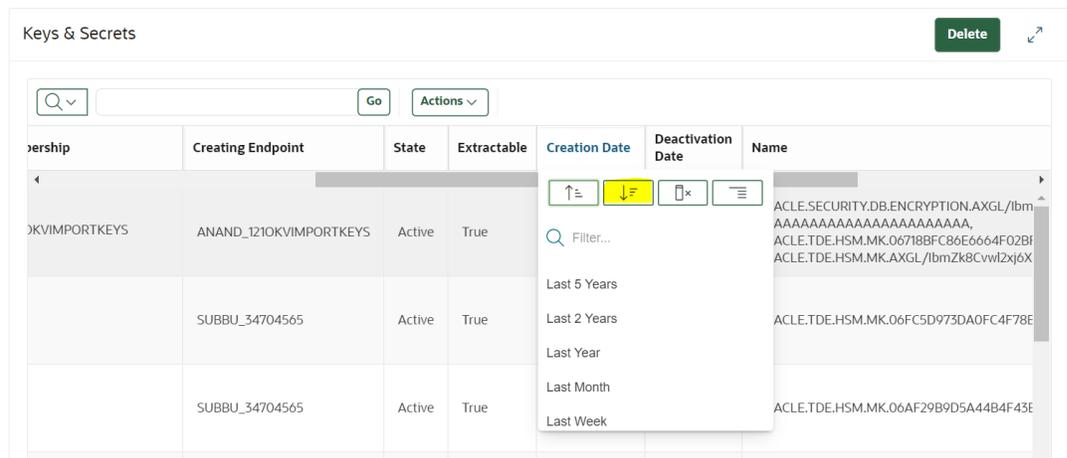
Probable Cause 5

The default wallet is not configured properly for the endpoint database.

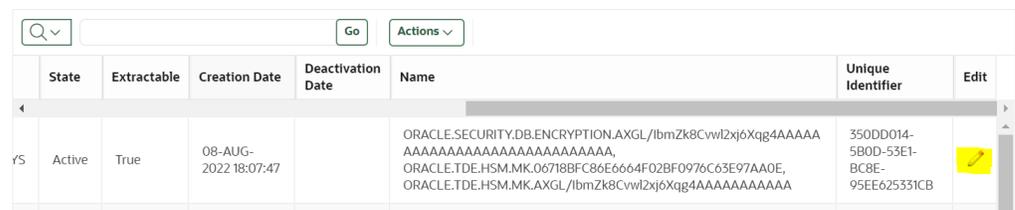
Solution

Verify whether the default wallet is assigned to the endpoint:

1. Log in to the Oracle Key Vault management console as the Key Administrator user.
2. Verify whether the latest keys are added to the wallet. Go to the **Keys & Secrets** page, and sort the keys in descending order by **Creation Date**.



- a. Check whether the latest keys created from that particular endpoint, **Creating Endpoint** are assigned to the wallet.
- b. If the key is not assigned to any wallet then, edit the key by clicking on the **pencil icon**.



- c. On the edit page, click on **Add Wallet Membership** and select the wallet assigned to the endpoint and save it.

- d. Verify whether the key has wallet membership by revisiting the **Keys & Secrets** page.
3. Go to the **endpoints** page.
4. Search for the endpoint then click on the endpoint name.
5. Check the default wallet setting. If no default wallet is assigned, then click **Choose Wallet** to select the desired wallet.
6. Click **Save**.

7. Revisit the endpoint edit page to verify whether the default wallet is assigned correctly.
8. Verify if the endpoint has **READ/MODIFY** and **MANAGE** permissions to the wallet from the **Access to Wallets** section.
9. Once verified, re-enroll the endpoint and install the new `okvclient.jar` on the endpoint DB server at the `OKV_HOME` path. See, [How to Re-Enroll an Endpoint on an Endpoint Database](#).
10. Check if the issue is resolved.

Oracle Key Vault Server Communication or Connection Failed Error

Oracle Key Vault displays a server communication or connection failed error when you try to fetch data using `okvutil` or when installing `okvclient.jar`.

Example

Running `okvutil list` on endpoint server returns the following errors:

1.


```
bash$ ./okvutil list
Enter Oracle Key Vault endpoint password:
Error: Server Communication Error
```
2.


```
bash$ ./okvutil list
Enter Oracle Key Vault endpoint password:
Error: Server Connect Failed
```

Probable Cause 1

Port 5696 is blocked between the endpoint DB and the Oracle Key Vault server.

Solution

1. Check the endpoint connectivity to the Oracle Key Vault server:

```
$OKV_HOME/bin/okvutil list -v 4
```

2. If the endpoint connection to the Oracle Key Vault server fails, then check if port 5696 is open:

```
curl -v telnet://OKV_SERVER_IP:5696
```

3. If the preceding command fails to connect to the Oracle Key Vault server, then contact your network administrator to verify that port 5696 is open. If not, ensure that port 5696 is open.
4. Check if the issue is resolved.

Probable Cause 2

File ownership of `okvclient.ora` is incorrect.

Solution

1. Check the file ownership of the `okvclient.ora` file.

```
ls -l $OKV_HOME/conf/okvclient.ora
```

2. If `okvclient.ora` ownership is set to a different user than expected, reset the file ownership to the correct user. The `okvclient.ora` file ownership may have changed if the `okvutil` or other endpoint software was run with an unintended user (say, `root`).
3. Check if the issue is resolved.

Probable Cause 3

Oracle Key Vault server, CA, or endpoints certificates have expired.

Solution

1. Check if the endpoint certificates have expired.
 - a. Log in to the Oracle Key Vault management console as a user with System Administrator role or privilege.
 - b. Navigate to the **Endpoints** tab, and search for the endpoint. Check the expiration date under the **Endpoint Certificate Expiration Date** column. You can also check the Oracle Key Vault Home page for the endpoint certificate expiration alerts.
 - c. If the endpoint certificate has expired, select and re-enroll the endpoint. See, [How to Re-Enroll an Endpoint on an Endpoint Database](#).
 - d. Check if the issue is resolved.
2. Check if the Oracle Key Vault server or node certificates have expired.
 - a. Log in to the Oracle Key Vault management console as a user with the System Administrator role.
 - b. To check the server or node **certificate expiration** status, navigate to **System, Status** and check the **Server or Node Certificate Expiration Date**. You can also check the Oracle Key Vault Home page for server or node certificate expiration alerts.
 - c. Regenerate the server certificate if it has expired. Go to **System**, click **Settings**.
 - d. Go to **Service Certificates** and select **Manage Server or Node Certificate** to regenerate the new server certificate.

3. Check if the Oracle Key Vault CA certificate has expired.
 - a. Log in to the Oracle Key Vault management console as a user who has the System Administrator role.
 - b. To check the CA certificate expiration status, navigate to **System, Status** and check the **CA Certificate Expiration Date**. You can also check the Oracle Key Vault Home page for any alerts for the CA certificate expiration.
 - c. If the Oracle Key Vault CA certificate has expired, you must manually recover and regenerate the CA certificate, For 21.5 and later See [section 17.6, Managing the Oracle Key Vault CA Certificate](#) after it has expired, for details on how to do so. For Oracle Key Vault 21.4 and earlier, contact Oracle Support.
4. Check if the issue is resolved.

Probable Cause 4

The `SSL_WALLET_LOC` in `okvclient.ora` is set to an incorrect location.

Solution

1. Endpoint certificate rotation places the new endpoint certificates in a new directory under `$OKV_HOME/ssl` and `SSL_WALLET_LOC` is modified to point to this new directory. Check if the `SSL_WALLET_LOC` is correctly pointing to the most recently created directory under `$OKV_HOME/ssl`.
2. Go to the `$OKV_HOME/conf` directory, open `okvclient.ora` and verify if `SSL_WALLET_LOC` is pointing to the most recently created directory under `$OKV_HOME/ssl`.
3. If not, then modify the `SSL_WALLET_LOC` setting appropriately and save `okvclient.ora`.
4. Check if the issue is resolved.

Probable Cause 5

During an endpoint reenrollment, Oracle Key Vault endpoint software was upgraded in-place instead of installing the endpoint software in an empty directory.

Solution

1. During endpoint re-enrollment, if the existing `$OKV_HOME` directory is not removed before installing new `okvclient.jar`, then the installation process only upgrades the endpoint software. The endpoint certificates and `okvclient.ora` file are not updated in such a case. See, [How to Re-Enroll an Endpoint on an Endpoint Database](#).
2. Check if the issue is resolved.

Probable Cause 6

Oracle Key Vault server KMIP service is not available.

Solution

1. Run the following command to check if it can list the keys successfully:

```
$OKV_HOME/bin/okvutil list -v4
```

2. If the preceding command fails to list the keys then check the availability of the Oracle Key Vault server by accessing the Oracle Key Vault management console.

3. If the Oracle Key Vault server is up, then verify the status of the KMIP service using one of these methods:
 - a. If the REST services utility is configured on the endpoint DB server, then run the REST services command `okv server status get` to get the server information and verify the status of the KMIP service.
 - b. If the SNMP monitoring is configured, then use the SNMP monitoring to determine the status of the KMIP service.
 - c. Log in to the Oracle Key Vault management console as a user who has the System Administrator role. In the Oracle Key Vault home page, check if there is an alert reporting the KMIP process as `stop responding`.
 - d. Log in to the Oracle Key Vault management console as a user who has the System Administrator role. Navigate to the **System, Status** page and check the status of the KMIP service.
4. If the status of the KMIP service is down, then restart the KMIP service.
 - Log in to the Oracle Key Vault server through SSH as the support user and then switch user `su` to `root`.

```
ssh support@okv_server_IP_address
su - root
```

5. Run the following command to verify the status of the KMIP process.

```
ps -eaf | grep kmip
```

6. If the preceding command shows that the KMIP process is not running, then restart the KMIP service:

```
systemctl stop kmip
systemctl start kmip
```

7. Verify the status of the KMIP process:

```
ps -eaf | grep kmip
```

8. On the endpoint DB server, check if `okvutil` lists the keys:

```
$OKV_HOME/bin/okvutil list -v4
```

9. Check if the issue is resolved.

Probable Cause 7

A strict IP check is enabled for the endpoint. The endpoint IP address is different from the IP address that was saved during endpoint enrollment.

Solution

1. Check IP address of the endpoint that is currently saved in the Oracle Key Vault server. Log in to the Oracle Key Vault management console as a user who has the System Administrator role or the privilege to manage this endpoint.

Navigate to the **Endpoints** page, search for the endpoint and verify the IP address saved for the endpoint.

2. Click on the endpoint name to open the **Endpoint Details** page.
3. Select **False** for the **Strict IP Check Enforced** option.
4. In the endpoint DB server, check if `okvutil` lists the keys.

```
$OKV_HOME/bin/okvutil list -v4
```

5. If deselecting the strict IP check resolves the issue, then:
 - a. Check if more than one IP address is configured for the endpoint DB server. In such a case, `Strict IP Check Enforced` cannot be enabled.
 - b. If the endpoint DB server is configured with only one IP address, then re-enroll the endpoint. See, [How to Re-Enroll an Endpoint on an Endpoint Database](#) and re-enable the `Strict IP Check Enforced` for the endpoint.
6. The `Strict IP` check should be re-enabled if the endpoint DB server is reconfigured and its connecting from one IP address only.
7. Check if the issue is resolved.

Probable Cause 8

There is a time lag between the endpoint DB server and the Oracle Key Vault server.

Solution

1. Check if there is a time drift between the endpoint DB server and the Oracle Key Vault server.
2. If yes, resolve the time drift. Consider using the NTP configuration to synchronize the system clock of the Oracle Key Vault server.
3. Check if the issue is resolved.

Could Not Store Private Key Errors on Wallet Upload

When uploading Java keystores, using `okvutil`, Oracle Key Vault displays could not store private key errors.

Example

`okvutil upload` command returns the following error:

```
$ okvutil upload -l ./fin_jceks.jck -t JCEKS -g fin_wal -v2
okvutil version 21.5.0.0.0
Configuration file: /tmp/fin_okv/conf/okvclient.ora
Server: 192.0.2.254:5696
Uploading from /tmp/fin_okv/keystores/jks/keystore.jks
Enter source Java keystore password:
Uploading private key
Uploading private key

Upload Failed
WARNING: Could not store private key error
```

Probable Cause

This error may occur when you upload two Java keystores using `okvutil` with the same file name but different contents, or if you use the same alias, like `-alias slserver`, in each `keytool` command.

Solution

1. When you download two keystores with the same alias, the `okvutil` download process ignores the second one because the JKS aliases must be unique.
2. To fix this error, upload the second keystore using a unique alias.

RESTful Services Endpoint Provisioning Command Failure

The endpoint provisioning using RESTful services utility fails due to incorrect path or directory.

Example

```
$ okv admin endpoint provision --generate-json-input
Error:/usr/bin/java does not exist.
```

Probable Cause

The RESTful service command to provision an endpoint fails if the soft link `/usr/bin/java` does not exist or points to an incorrect Java directory.

Solution

Ensure that you use Java version 1.7.21 or later. Create a soft link to the Java home directory:

```
ln -s Java_home_directory/bin/java /usr/bin/java
```

Uploading Certificate File Failure

When uploading a console certificate an error is displayed .

Example

Uploading a console certificate from Oracle Key Vault Management console returns the error.

```
ORA-20101: Failed to upload certificate file
```

Probable Cause

Mandatory certificates have expired.

Solution

1. Log in to the Oracle Key Vault server as a support user.
2. Switch to the `root` user.
3. Regenerate the certificates:

```
# /usr/local/bin/gensslcert create-certs
```

4. Log in to the Oracle Key Vault management console as SYSADMIN.
5. Regenerate the console certificate request from the **Console Certificate** page.
6. Download and sign the console certificate.
7. Upload the signed console certificate to the **Console Certificate** page.
8. Verify if the issue is resolved.

Error in Uploading the Java Keystore

Oracle Key Vault displays an error when uploading the Java keystore.

Example

Uploading Java Keystore using `okvutil` fails with below error,

```
SEVERE: Error occurred while determining entry type.  
java.security.UnrecoverableKeyException: Cannot recover key
```

Probable Cause

The Java keystore (JKS) has multiple passwords. One password protects the complete keystore. Each private key also gets an additional password that can be different from the other keystore password. To upload a JKS from `okvutil`, all the key passwords should be the same as the keystore password.

Solution

1. Set both the keystore and the key passwords to the same value.
2. Upload JKS to Oracle Key Vault.
3. Reset the passwords again if you want them to be different.
To change the keystore password:

```
$ keytool -storepasswd -keystore keystorename  
Enter keystore password: <old password>  
New keystore password: <new password>  
Re-enter new keystore password: <new password>
```

To change an individual key password:

```
$keytool -keypasswd -keystore keystorename -alias <alias>  
Enter keystore password: <keystore password>  
Enter key password for <alias> <old key password>  
New key password for <alias>: <new key password>  
Re-enter new key password for <alias>: <new key password>
```

4. Upload the entire keystore as an **Other** type instead of JKS, if the passwords cannot be changed. The password is stored as a binary file and you must track the passwords elsewhere.

 **Note:**

You get a centralized repository for backup and distribution, but cannot manage keys and certifications at an object-level granularity.

SSL layer Error while migrating MYSQL Database Keys to Oracle Key Vault

An SSL layer error displays when the MySQL database is migrated to Oracle Key Vault.

Example

Following errors are seen in the mysql database trace or log file:

```
reported: 'Error setting the certificate file.'  
reported: 'Could not initialize ssl layer'  
reported: 'keyring_okv initialization failure. Please check that the  
keyring_okv_conf_dir points to a readable directory and that the directory  
contains Oracle Key Vault configuration file and ssl materials.'
```

Probable Cause

Corrupted certificate file.

Solution

1. Check if Oracle Key Vault is operational.
2. Edit the certificate files copied from the SSL folder extracted from `okvclient.jar`.
3. Delete any blank line at the end of the certificate file and click **Save**.
4. Verify if the issue is resolved.

Rotation or Set Key Failure in Windows Environment

In Oracle Key Vault error displays while rotating or setting a new key.

Example

Rotating or setting a new key fails with ORA-46627: keystore password mismatch

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY xxxxxxxx;  
ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY xxxxxxxx  
* ERROR at line 1: ORA-46627: keystore password mismatch
```

Probable Cause

`ewallet.p12` under `ssl` folder does not have permissions to access the Oracle Key Vault server.

Solution

1. Open properties window for the `OKV_HOME` folder.
2. Disable the Inheritance permissions for `OKV_HOME/ssl` folder.
3. Keep the inheritance permissions enabled for `OKV_HOME/ssl/ewallet.p12`.

4. Verify, if the set key is successful.

Rotation or Set Key Fails with ORA-03113

In Oracle Key Vault, while rotating the key an error displays.

Example

Rotating or setting a new key from an endpoint database fails with the error.

```
ORA-03113 END-OF-FILE ON COMMUNICATION CHANNEL
```

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY xxxxxxxx;  
ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY xxxxxxxx  
* ERROR at line 1: ORA-03113 END-OF-FILE ON COMMUNICATION CHANNEL
```

Probable Cause

Persistent cache wallet is corrupted and not accessible to update.

Solution

1. Rename the `ewallet.p12` under `$OKV_HOME/conf` or `$ORACLE_BASE/okv/$ORACLE_SID`.
2. Login to sqlplus, and query `v$encryption_keys` to create a new persistent cache wallet.

```
SQL> select key_id from v$encryption_keys
```

3. Verify if the `ewallet.p12` is created under `$OKV_HOME/conf` or `$ORACLE_BASE/okv/$ORACLE_SID`.
4. Verify if the set key is successful.

Multi-Master Cluster Issues

Review these troubleshooting tips for common Multi-Master Cluster related errors when working with Oracle Key Vault.

- [Heartbeat Lag or High Replication Lag in Multi-Master Cluster Environment](#)
Heartbeat lag or replication lag is very high or undetermined in a multi-master cluster environment.
- [Cluster Node Pairing Failure](#)
Cluster node pairing fails in multi-master cluster environment.
- [Adding a Node to Cluster Fails with Invalid Certificate or Certificate Expired Error](#)
When adding a node to the cluster certificate expired or invalid certificate error is displayed.
- [How to Diagnose Oracle Key Vault Cluster Issues](#)
Based on the issue, use the paths in this topic to troubleshoot the Oracle Key Vault cluster issues.

Heartbeat Lag or High Replication Lag in Multi-Master Cluster Environment

Heartbeat lag or replication lag is very high or undetermined in a multi-master cluster environment.

Example

Cluster Monitoring page on Oracle Key Vault Management console shows,

```
Heartbeat Lag or Replication Lag higher than 120 sec and increasing consistently.
```

Probable Cause 1

This issue may occur because of network connectivity failures. This issue may occur because of network connectivity failures, the required ports are not open between cluster nodes or intermittent network performance issues.

Solution

1. On each cluster node, log in to the Oracle Key Vault management console as a System Administrator and navigate to the **Cluster Monitoring** page.
2. Check the heartbeat lag and replication lags.
3. If the heartbeat lag or replication lag is high or **Cluster Services Status** is down (a down red arrow), Click **Restart Cluster Services** to restart the cluster services. Wait for a few minutes and refresh the page.
4. If the lag is still high, for each cluster node, log in to the cluster node through SSH as the support user, switch user `su` to `root`, and ping the problematic nodes.

```
curl -v telnet://other-node-ip-address:7093
curl -v telnet://other-node-ip-address:7443
```

5. If connectivity fails, then resolve network connection issues to ensure that the preceding commands are successful.
6. Wait for few minutes for the network connectivity to be restored.
7. Check if the issue is resolved.

Probable Cause 2

Cluster node replication has failed.

Solution

1. Identify the cluster nodes that have a replication lag.
2. Reboot the problematic cluster node(s).
3. Check if the issue is resolved.

Probable Cause 3

Oracle Key Vault CA certificate, node certificates, nodes have expired.

Solution

1. On all cluster nodes, check if the Oracle Key Vault node certificates has expired.

- a. Log in to the Oracle Key Vault management console as System Administrator.
 - b. To check the node certificate expiration status, navigate to **System, Status** and check the **Node Certificate Expiration Date**. You can also check the Oracle Key Vault Home page for any node certificate expiration alerts.
 - c. Regenerate the node certificate if it has expired. Go to **System**, select **Settings**.
 - d. Select **Service Certificates** and click **Manage Node Certificate** to regenerate the new node certificate.
2. On all cluster nodes, check if the Oracle Key Vault CA certificate has expired.
 - a. Log in to the Oracle Key Vault management console as System Administrator.
 - b. To check the CA certificate expiration status, go to **System**, select **Status** and check the **CA Certificate Expiration Date**. You can also check the Oracle Key Vault Home page for any CA certificate expiration alerts.
 - c. In Oracle Key Vault release 21.5 and later, you cannot start a CA certificate rotation if the CA has already expired. You must generate a new CA certificate manually and re-enroll all endpoints instead. See [Managing CA Certificate Rotation](#). For Oracle Key Vault 21.4 and earlier, contact Oracle Support.
 3. Check if the issue is resolved.

Cluster Node Pairing Failure

Cluster node pairing fails in multi-master cluster environment.

Example

Adding a node to cluster fails,
Status of the node in Cluster Management page on the Oracle Key Vault Management console shows **Pairing** for indefinite amount of time.

Probable Cause 1

The NTP settings are different on the nodes being paired.

Solution

1. Log in to the Oracle Key Vault management console of the controller node as System Administrator.
2. Log in to the Oracle Key Vault management console on both the nodes.
3. Go to the **System, Settings** page.
4. In the Network Services area, click **NTP** to display the **System Time** window.

System Time

System Time
Save

Set Manually

Use Network Time Protocol

Synchronize After Save Synchronize Periodically

NTP Server 1

Address

Server Time (in UTC) 24-JUN-2021 21:20:32 (-0.000074 seconds difference)

NTP Server 2

Address

Server Time (in UTC)

NTP Server 3

5. Verify if the pairing status of the nodes has changed to **ACTIVE** after some time.
6. Check if the issue is resolved.

Probable Cause 2

The controller node and the new node are not reachable through the required ports.

Solution

1. Log in to all the nodes through SSH as the support user. Switch user `su` to `root` and ping the problematic nodes with the required ports.

```
curl -v telnet://ipaddress:7093
curl -v telnet://ipaddress:7443
```

2. If connectivity fails, then enable the required ports between nodes. See, Oracle Key Vault Installation and Upgrade Guide for the network port requirements.
3. Verify if the pairing status has changed to **ACTIVE**.
4. Check if the issue is resolved.

Adding a Node to Cluster Fails with Invalid Certificate or Certificate Expired Error

When adding a node to the cluster certificate expired or invalid certificate error is displayed.

Example

```
Adding a node to cluster fails with,
OAV-4783 Invalid certificate certificate has expired
```

Probable Cause

Invalid and the AV Certificates are expired on the node.

Solution

1. Log in to the controller node using SSH as a support user and switch to the root user.

 **Note:**

If you add node 2 as a peer read/write node to node 1, then node 1 is the controller node.

2. Go to the certificates directory.

```
cd /usr/local/dbfw/etc
```

3. Check the expiration date of the platform CA certificate by running the following command . If the date displayed in the **notAfter** field is in the past, the certificate has expired.

```
# openssl x509 -in /usr/local/dbfw/etc/ca.crt -enddate -noout
```

As an example:

```
# openssl x509 -in /usr/local/dbfw/etc/ca.crt -enddate -noout  
notAfter=Jan 2 15:06:34 2023 GMT
```

If the certificate has expired, regenerate the platform certificates. For more information about platform certificates, see [Managing Oracle Key Vault Platform Certificates](#).

How to Diagnose Oracle Key Vault Cluster Issues

Based on the issue, use the paths in this topic to troubleshoot the Oracle Key Vault cluster issues.

- [Cluster-Node Pairing Issues](#)
The error logs provide the path to look for issues related to cluster-node pairing. Different log files capture the issue details at different locations. Refer these log files to understand more about issues related to cluster-node pairing.

Cluster-Node Pairing Issues

The error logs provide the path to look for issues related to cluster-node pairing. Different log files capture the issue details at different locations. Refer these log files to understand more about issues related to cluster-node pairing.

General Troubleshooting from Oracle Key Vault Management Console

- Monitoring Page - displays the nodes status.
- Verify the **Cluster Monitoring** page for replication lag below 120 seconds or recommended value, if objects like endpoints, or wallets (except node) are in the **Pending** state and not changing to **Active** state.

Table C-1 Controller Node Key Files

File Path	Description
/var/log/messages	Location that reports errors.
/var/log/debug	Details operations such as, HTTPS requests. The file records errors that occur during pairing.
/var/okv/log/mmha/ welcome_node_status.txt	Records the timestamps and steps run by the controller node. Also records errors that occur during the recent pairing on the controller node.
/var/okv/log/mmha/ okv_replicate_to_new_node*.log	Look for the most recent pairing with <code>ls -lrt grep okv_replicate_to_new_node</code> . This may have more details than <code>welcome_node_status.txt</code>
/var/okv/log/mmha/ okv_new_node_added*.log	Same as <code>okv_replicate_to_new_node*.log</code> except that it is used at a later step in the pairing, after <code>okv_load_node_tune_restored_bkp*</code> has finished on the candidate node.
/var/okv/og/mmha/output_date_ogg_*	These files are the output of OGG commands

Table C-2 Candidate Node Key Files

File Path	Description
/var/log/messages	Location that reports errors.
/var/log/debug	Detail operations such as, HTTPS requests. The file records errors that occur during pairing.
/var/lib/oracle/ogg/okv/ load_node_info.txt	Generated by the candidate node. The controller node attempts to retrieve this file via HTTPS and verifies the information as one of the starting steps in the pairing process.
/var/okv/log/mmha/ new_node_status.txt	Timestamps and steps are written here. If there is an error during pairing on the candidate node, then it is located here.
/var/okv/log/mmha/ okv_load_node_from_backup*.log	Occurs after <code>okv_replicate_to_new_node</code> on the controller node. More details in <code>new_node_status.txt</code> .
/var/okv/log/mmha/ okv_load_node_tune_restored_bkp*.log	Occurs after <code>okv_load_node_from_backup</code> (after a reboot). More details than <code>new_node_status.txt</code> .
/var/okv/log/db/ output*okv_backup_restore.log	Backup restores during pairing records the output here.

Backup and Restore Issues

Review these troubleshooting tips for common backup and restore related issues when working with Oracle Key Vault.

- [Oracle Key Vault Backup Failed Error](#)
During Oracle Key Vault backup operation, the backup operation fails with `Failed to pack files` error.

- [Unable to Schedule a New Backup](#)
Oracle Key Vault failed to schedule a new backup on the management console.
- [Remote Backup Failed](#)
Remote backup fails on the Oracle Key Vault management console.
- [Backup Restore Failure](#)
Oracle Key Vault backup restore operation fails with `Backup Restoration Failed` error.

Oracle Key Vault Backup Failed Error

During Oracle Key Vault backup operation, the backup operation fails with `Failed to pack files` error.

Example

Oracle Key Vault Backup failed with:

```
Error: Failed to pack files
```

Probable Cause

Oracle Key Vault backup failed due to the **Failed to pack files** error.

Solution

1. Login to the Oracle Key Vault management server through ssh as support.
2. Switch to root user.
3. Go to `/var/okv/log/db/` open the latest backup run log named as `output*okv_backup_run.log`.
4. If you get a `permissions issues to access a file error`, then fix the permissions and run the backup again.
5. Check if the issue is resolved.

Unable to Schedule a New Backup

Oracle Key Vault failed to schedule a new backup on the management console.

Example

Scheduling a new backup on Oracle Key Vault Management console fails with:

```
Error: Unable to schedule new backup
```

Probable Cause

There may be an old backup still in progress or in a non-responsive state for several days.

Solution

1. Restart the Oracle Key Vault server and verify if the old backup has completed.
2. Schedule the new backup.
3. Check if the issue is resolved.

Remote Backup Failed

Remote backup fails on the Oracle Key Vault management console.

Example

Remote Backup status on the Oracle Key Vault management console displayed as:

```
Error: Remote Backup Failed
```

Probable Cause 1

There may be a remote backup in **Ongoing** status for several days.

Solution

1. Restart the Oracle Key Vault server and verify if the old backup has completed.
2. Check if the issue is resolved.

Probable Cause 2

Backup remote destination is not accessible.

Solution

1. Verify if the remote destination is accessible and `scp` is working in the configured remote destination path.
2. If `scp` is successfully working:
 - a. Verify if the Oracle Key Vault server is upgraded, or if the server, CA certificate, or node certificate is rotated recently. Verify if the remote backup destination is not configured again.
 - b. Log in to the Oracle Key Vault management console as a system administrator.
 - c. Go to **System, Settings**.
 - d. Go to the **Backup and Restore** page.
 - e. Edit **Remote backup destination** and copy the public key.
 - f. Paste the public key in the `known_hosts` file on the remote destination server.
3. If the server, CA certificate, or node certificate is rotated, then the public key of the Oracle Key Vault server or node has changed.
4. Copy the new public key that appears in the **Public Key** field in the **Backup Destination Details** page.
5. Paste the public key in the appropriate configuration file, such as `authorized_keys`, on the backup destination server.
6. Go to **System, Settings**.
7. In the **System Configuration** area, select **Backup and Restore**.
8. Click **Manage Backup Destination** to view all backup destinations.
9. Click **Create**. The **Public Key** field displays the new public key.
10. Copy this public key to the `authorized_key` file of the user specified in the backup destination configuration.

11. Apply the change for each backup destination.
12. Check if the issue is resolved.

Backup Restore Failure

Oracle Key Vault backup restore operation fails with `Backup Restoration Failed` error.

Example

Oracle Key Vault restore from a remote backup failed.

```
Error: Backup Restoration Failed
```

Probable Cause

Restoration fails, if the remote server is not accessible.

Solution

1. Verify if the backup file from the remote backup destination is accessible by the Oracle Key Vault server.
2. If not, ensure that the Oracle Key Vault server can access the backup file from the remote backup destination.
3. Check if the issue is resolved.
- [Backup Size is Growing Exponentially](#)
Oracle Key Vault backup size is exponentially growing every time you take a backup.

Backup Size is Growing Exponentially

Oracle Key Vault backup size is exponentially growing every time you take a backup.

Example

```
Oracle Key Vault restoration failure.
```

Restoration fails, if the remote server is not accessible.

Solution

1. Use RMAN to delete the OBSOLETE archive logs as well as the database backups.
2. Connect to the Oracle Key Vault server using SSH as support, switch to root user and then switch to ORACLE user.
3. Connect using RMAN utility and run the following commands to list the OBSOLETE redo logs and backups taken until then and the obsolete backups and archived logs.

```
rman target /  
RMAN> list backup summary;  
RMAN> report obsolete;
```

4. Delete the obsolete backups using the following command from RMAN.

```
:RMAN> delete noprompt obsolete;
```

5. If this action plan generates other warning messages please contact the support team.
6. This will not delete the backup taken by the system on the REMOTE location. Those backup files need to be managed manually.

Certificate Related Issues

Review these troubleshooting tips for common certificate-related issues when working with Oracle Key Vault.

- [Updating to Current Certificate Issuer](#)
While the Oracle Key Vault CA certificate rotation is in progress, the endpoint's status remains as **Updating in Progress** for many days. The CA certificate rotation process may be stalled if there are several endpoints in the **Updating in Progress** state.

Updating to Current Certificate Issuer

While the Oracle Key Vault CA certificate rotation is in progress, the endpoint's status remains as **Updating in Progress** for many days. The CA certificate rotation process may be stalled if there are several endpoints in the **Updating in Progress** state.

Example

Probable Cause 1

No recent activity from the endpoint.

Solution

1. In the endpoint, go to `$OKV_HOME/bin`, and run the `okvutil list` command multiple times.

```
$OKV_HOME/bin/okvutil list -v 4
```

2. If the preceding command returns data then:
 - a. Verify if `$OKV_HOME/ssl` is updated with the new certificates. A new directory is created under `$OKV_HOME/ssl` that contains `ewallet.p12`.
 - b. Verify the endpoint status in the Oracle Key Vault management console.
 - c. If the endpoint status still shows `Update in Progress`, then contact Oracle support.

Note:

In a multi-master cluster environment, the endpoint may not connect to the node where the new endpoint certificates are generated.

3. If the `okvutil` command fails with an error, re-enroll the endpoint, download and install the `okvclient.jar` file. See, [How to Re-Enroll an Endpoint on an Endpoint Database](#).
4. Verify if the certificate rotation proceeds.
5. Check if the issue is resolved.

Probable Cause 2

The endpoint is no longer in use.

Solution

1. Check if the endpoint is not in use. If so, delete or re-enroll the endpoint.
2. Repeat the same action for all the inactive endpoints.
3. Verify if the certificate rotation proceeds.
4. Check if the issue is resolved.

Installation and Upgrade Issues

Review these troubleshooting tips for common installation and upgrade issues when working with Oracle Key Vault.

- [Oracle Key Vault Installation Failure](#)
Oracle Key Vault installation is failing.
- [Oracle Key Vault Upgrade Failure](#)
Oracle Key Vault upgrade fails with file descriptor (FD) open error.
- [Oracle Key Vault Management Console is Not Accessible After Installation](#)
The Oracle Key Vault management console is not accessible after rebooting the Oracle Key Vault server as part of the installation on VMWare virtual machine.
- [Oracle Key Vault Upgrade Failure](#)
Oracle Key Vault upgrade fails with an error.
- [Unable to boot after installation of Oracle Key Vault on VMWare VM](#)
Unable to boot the Virtual Machine (VM) after Oracle Key Vault installation.
- [Operation Failed on Network Information Screen After Upgrade from 21.x to 21.5 and Later.](#)
Network information screen is displaying **Operation failed** error.

Oracle Key Vault Installation Failure

Oracle Key Vault installation is failing.

Probable Cause

Oracle Key Vault Installation may fail due to various reasons listed with solutions in the **Solutions** section.

Solution

1. Check if the physical server has RAID enabled software.

Note:

Oracle Key Vault does not support the use of Software RAID. Ensure that RAID is configured only at the hardware level.

2. On the Oracle Key Vault server, check the network settings. Also check if there is a delay in copying the `iso` files from the NFS location to the disk.
3. Check system configuration including disk size, number of disks, disk configuration, RAM size, and number of CPU cores. Ensure that the server meets the minimum system configuration requirements of Oracle Key Vault.
4. Verify if the checksum matches with the local `iso` file.
 - a. Copy the checksum of the `iso` from the specific location where you downloaded the `iso` file.
 - b. Generate the checksum for the downloaded `iso` file.
 - c. Both the checksum values should be the same. If not, download the correct `iso` file.

```
sha256sum isoname.iso
```

- d. Check if the issue is resolved.

Oracle Key Vault Upgrade Failure

Oracle Key Vault upgrade fails with file descriptor (FD) open error.

Probable Cause 1

Oracle Key Vault upgrade version is not supported.

Solution

Verify that the Oracle Key Vault upgrade from the current release to the version you want to upgrade is supported. See, *Release Notes* of specific Oracle Key Vault release for supported upgrade paths.

Probable Cause 2

Oracle Key Vault certificates have expired.

Solution

1. Check if the Oracle Key Vault server or node certificates have expired.
 - a. Log in to the Oracle Key Vault management console as a user who has the system administrator role.
 - b. To check the server or node certificate expiration status, go to **System, Status** and check the **Server or Node Certificate Expiration Date**.
 - c. Regenerate the server certificate if it has expired. Go to **System, Status**.
 - d. Click **Service Certificates** and select **Manage Server or Node Certificate** to regenerate the new server certificate.
2. Check if the Oracle Key Vault CA certificate has expired.
 - a. Log in to the Oracle Key Vault management console as a system administrator.
 - b. To check the CA certificate expiration status, navigate to **System, Status** and check the **CA Certificate Expiration Date**.
 - c. If the CA certificate has expired, contact Oracle support.

3. Verify if all the pre-upgrade steps are performed as described in the [Oracle Key Vault Installation and Upgrade guide](#).

 **Note:**

For Oracle Key Vault 21.5 or higher, see the steps listed in [Managing CA Certificate Rotation](#) after it has expired, to rotate the expired CA certificate.

Considerations for Multi-Master Cluster Deployment

1. Log in to each node's management console, and check if the replication lag or heartbeat lag is high in other nodes. If yes, then resolve this issue first. See, [Heartbeat Lag or High Replication Lag in Multi-Master Cluster Environment](#).
2. Based on the upgrade scenario, certain restrictions may apply when upgrading nodes of a read/write pair. This includes the order in which the nodes are disabled, upgraded, and then re-enabled after the upgrade. See, [Oracle Key Vault Installation and Upgrade guide](#) of the target version for specific instructions.
3. Verify if the node is not disabled beyond the maximum node disabled duration limit.

Considerations for Primary-Standby Environment

1. Verify if the upgrade is performed first on standby and then on the primary. See, [Upgrading a Pair of Primary-Standby Oracle Key Vault Servers](#).
2. Retry the upgrade. If the issue persists, contact Oracle Support.

Oracle Key Vault Management Console is Not Accessible After Installation

The Oracle Key Vault management console is not accessible after rebooting the Oracle Key Vault server as part of the installation on VMWare virtual machine.

Example

On the Oracle Key Vault server, the `/var/log/messages` shows the error,

```
OAV-46501: invalid IP Address.  
java.sql.SQLException: OAV-46501: invalid IP Address.
```

Probable Cause

The HTTPd server daemon on the Oracle Key Vault server is not started. Oracle Key Vault faced issues in identifying the network adapter on VMware.

Solution

1. Change the network adapter settings in VMware. Change the Virtual Machine from VMXNET3 to e1000.
2. Reboot the Oracle Key Vault server.

Oracle Key Vault Upgrade Failure

Oracle Key Vault upgrade fails with an error.

Example

Oracle Key Vault upgrade fails with,

Open FDs on: Volume: lv_tmp, Process: sshd

Sample Error Output

```
[root@server ]# /usr/bin/ruby /images/upgrade.rb --confirm
Power loss during upgrade may cause data loss. Do not power
off during upgrade.
Verifying boot partition before upgrade
Verifying upgrade preconditions
The Oracle base has been set to /var/lib/oracle

AVDF::Installer::Upgrade::LVOpenFD

Open FDs on: Volume: lv_tmp, Process: sshd, File(s): /tmp/ssh-<xxxxxx>/
agent.1662

Failed to apply update: Verifying pre-upgrade conditions failed.
Failed to apply update: /images/upgrade/lib/preconditions.rb:2189:in
'verify_all'
/images/upgrade.rb:203:in 'upgrade'
/images/upgrade.rb:282:in '<main>'
```

Probable Cause

This error is due to a running SSH process.

Solution

1. Verify if any SSH process is running on the Oracle Key Vault server:

```
$ ps -eaf | grep -i "ssh"
```

2. Stop the running SSH process and retry the upgrade.

Unable to boot after installation of Oracle Key Vault on VMWare VM

Unable to boot the Virtual Machine (VM) after Oracle Key Vault installation.

Example

After reboot, the following error message appears on VM console.

```
PXE-M0F: Exiting Intel PXE ROM
          Operating system not Found
```

Probable Cause

The virtual disk is set to 2 TB and is not allowing the server to boot.

Solution

1. Set the virtual disc size to 1.9 TB.
2. In vSphere 5.5.x and 6.0.x, large capacity virtual disks have these conditions and limitations:
 - The maximum supported VMDK size on an VMFS-5 data store is increased to 62 TB. However, the maximum supported VMDK size on VMFS-3 is still 2 TB.
 - The maximum supported size of a VMDK on NFS is the lesser of 62 TB and 1% less than the maximum file size supported by the NFS filesystem.

Operation Failed on Network Information Screen After Upgrade from 21.x to 21.5 and Later.

Network information screen is displaying **Operation failed** error.

Example

The network information screen is displaying **Operation failed** error while fetching the network interfaces details.

Probable Cause

The network configuration file name is not changed to the supported naming convention on 21.5 or later.

Solution

1. Go to `/etc/sysconfig/network-scripts/`.
2. List all the files and find if the `ifcfg*` files have `d1` appended at the end. For example, `/etc/sysconfig/network-scripts/ifcfg-bond0-eno3d1`.
3. Rename the files to the correct name, `mv ifcfg-bond0-eno3d1 ifcfg-bond0-eno3`.
4. Reboot the server.

Primary-Standby Configuration Issues

Review these troubleshooting tips for commonly encountered primary-standby configuration related issues when working with Oracle Key Vault.

- [Write Operations Fail in Restricted Mode](#)
When the primary or standby server is in read-only restricted mode, write operations like creation of key fails.
- [Fast-Start Failover Suspended](#)
Oracle Key Vault displays a Fast-Start Failover suspended error.

- [How to Verify Primary-Standby Status](#)
After the Primary-Standby nodes pairing, verify the status of the primary-standby nodes.

Write Operations Fail in Restricted Mode

When the primary or standby server is in read-only restricted mode, write operations like creation of key fails.

In Primary-Standby configuration, the status of the node on the Oracle Key Vault management console is, Read-Only Restricted.

Probable Cause

The recommended ports are not open between the primary and standby servers.

Solution

1. Verify the alerts on the Oracle Key Vault management console home page.
2. Log in to the Oracle Key Vault server through SSH as the support user. Switch to the `root` user and verify if the network connectivity between the primary and standby servers is successful or not.
3. Verify ports 1521, 1522, 7443, and 5696.

```
curl -v telnet://ipaddress:port
```

4. If the command fails, fix the port issue and ensure that the ports are available and open.
5. Reboot the Oracle Key Vault server.
6. Verify if the issue is resolved.

Fast-Start Failover Suspended

Oracle Key Vault displays a Fast-Start Failover suspended error.

Example

On Oracle Key Vault Server, you may see the error in `/var/log/messages` or when you run the `show_configuration` command from `dgmgrl`.

```
ORA-16818: Fast-Start Failover suspended
```

Probable Cause

An ORA-16818: Fast-Start Failover suspended appears because of a fast start fail over operation failure.

Solution

1. Gracefully shut down the primary server in a controlled manner using the **Power Off** option instead of manually turning off the computer. When you power off the Primary Server, you cannot perform a fast start failover function, resulting in an ORA-16818 error.
2. In a graceful shutdown operation, the primary server's failover status goes into a suspended state with the standby waiting indefinitely for the primary server to be available. This is the expected behavior for a Fast-Start Failover (FSFO) operation.

 **Note:**

Avoid a split-brain scenario, as defined by Oracle Data Guard.

3. By design, an FSFO operation error occurs only when the primary server shuts down unexpectedly.
4. If you perform a `SHUTDOWN IMMEDIATE` or `SHUTDOWN NORMAL` command in SQL*Plus, then the FSFO does not occur because the database shuts down gracefully.

How to Verify Primary-Standby Status

After the Primary-Standby nodes pairing, verify the status of the primary-standby nodes.

1. Check the Primary-Standby nodes status from the Oracle Key Vault management console.
 - a. Log in to the Oracle Key Vault management console.
 - b. On the **Primary-Standby** page, go to **System, Settings**.
 - c. Check the nodes status.
2. Check the status from Primary-Standby nodes.
 - a. Log in to the Primary-Standby nodes as a support user using SSH, and then switch to the `root` user.
 - b. Run these commands to check the status.

```
/usr/local/dbfw/bin/setup_ha.rb --status
/usr/local/dbfw/bin/setup_ha.rb --dg_status
sudo -u oracle /usr/local/dbfw/bin/setup_ha.rb --ha_role
```

Sample Output

```
$ /usr/local/dbfw/bin/setup_ha.rb --status
DGMGRL for Linux: Version 12.1.0.2.0 - 64bit Production

Copyright (c) 2000, 2013, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
Connected as SYSDBG.
DGMGRL>
Configuration - DBFWDB

Protection Mode: MaxAvailability
Members:
DBFWDB_HA2 - Primary database
DBFWDB_HA1 - (*) Physical standby database

(*) Fast-Start Failover target

Properties:
FastStartFailoverThreshold = '120'
OperationTimeout = '30'
TraceLevel = 'USER'
FastStartFailoverLagLimit = '30'
```

```

CommunicationTimeout = '60'
ObserverReconnect = '30'
FastStartFailoverAutoReinstate = 'TRUE'
FastStartFailoverPmyShutdown = 'FALSE'
BystandersFollowRoleChange = 'ALL'
ObserverOverride = 'FALSE'
ExternalDestination1 = ''
ExternalDestination2 = ''
PrimaryLostWriteAction = 'CONTINUE'

```

Fast-Start Failover: ENABLED

```

Threshold: 120 seconds
Target: DBFWDB_HA1
Observer: *****
Lag Limit: 30 seconds (not in use)
Shutdown Primary: FALSE
Auto-reinstate: TRUE
Observer Reconnect: 30 seconds
Observer Override: FALSE

```

Configuration Status:
SUCCESS

DGMGRL>
Database - DBFWDB_HA1

```

Role: PHYSICAL STANDBY
Intended State: APPLY-ON
Transport Lag: 0 seconds (computed 1 second ago)
Apply Lag: 0 seconds (computed 1 second ago)
Average Apply Rate: 2.00 KByte/s
Active Apply Rate: (unknown)
Maximum Apply Rate: (unknown)
Real Time Query: ON
Instance(s):
dbfwdb

```

```

Properties:
DGConnectIdentifier = 'DBFWDB_HA1'
ObserverConnectIdentifier = ''
LogXptMode = 'sync'
RedoRoutes = ''
DelayMins = '0'
Binding = 'optional'
MaxFailure = '0'
MaxConnections = '1'
ReopenSecs = '10'
NetTimeout = '30'
RedoCompression = 'DISABLE'
LogShipping = 'ON'
PreferredApplyInstance = ''
ApplyInstanceTimeout = '0'
ApplyLagThreshold = '0'
TransportLagThreshold = '0'
TransportDisconnectedThreshold = '30'

```

```

ApplyParallel = 'AUTO'
StandbyFileManagement = 'AUTO'
ArchiveLagTarget = '0'
LogArchiveMaxProcesses = '4'
LogArchiveMinSucceedDest = '1'
DbFileNameConvert = ''
LogFileNameConvert = ''
FastStartFailoverTarget = 'DBFWDB_HA2'
InconsistentProperties = '(monitor)'
InconsistentLogXptProps = '(monitor)'
SendQEntries = '(monitor)'
LogXptStatus = '(monitor)'
RecvQEntries = '(monitor)'
StaticConnectIdentifier = 'DBFWDB_HA1_DGMGRL'
StandbyArchiveLocation = 'USE_DB_RECOVERY_FILE_DEST'
AlternateLocation = ''
LogArchiveTrace = '0'
LogArchiveFormat = '%t_%s_%r.dbf'
TopWaitEvents = '(monitor)'

```

```

Database Status:
SUCCESS

```

```

DGMGRL>
Database - DBFWDB_HA2

```

```

Role: PRIMARY
Intended State: TRANSPORT-ON
Instance(s):
dbfwdb

```

```

Properties:
DGConnectIdentifier = 'DBFWDB_HA2'
ObserverConnectIdentifier = ''
LogXptMode = 'sync'
RedoRoutes = ''
DelayMins = '0'
Binding = 'optional'
MaxFailure = '0'
MaxConnections = '1'
ReopenSecs = '10'
NetTimeout = '30'
RedoCompression = 'DISABLE'
LogShipping = 'ON'
PreferredApplyInstance = ''
ApplyInstanceTimeout = '0'
ApplyLagThreshold = '0'
TransportLagThreshold = '0'
TransportDisconnectedThreshold = '30'
ApplyParallel = 'AUTO'
StandbyFileManagement = 'AUTO'
ArchiveLagTarget = '0'
LogArchiveMaxProcesses = '4'
LogArchiveMinSucceedDest = '1'
DbFileNameConvert = ''
LogFileNameConvert = ''

```

```
FastStartFailoverTarget = 'DBFWDB_HA1'
InconsistentProperties = '(monitor)'
InconsistentLogXptProps = '(monitor)'
SendQEntries = '(monitor)'
LogXptStatus = '(monitor)'
RecvQEntries = '(monitor)'
StaticConnectIdentifier = 'DBFWDB_HA2_DGMGRL'
StandbyArchiveLocation = 'USE_DB_RECOVERY_FILE_DEST'
AlternateLocation = ''
LogArchiveTrace = '0'
LogArchiveFormat = '%t_%s_%r.dbf'
TopWaitEvents = '(monitor)'
```

```
Database Status:
SUCCESS
```

```
DGMGRL>
```

- c. Run these commands to check Oracle Data Guard status. Make sure the Observer is running.

```
su oracle
dgmgrl /
show configuration;
show database 'DBFWDB_HA1';
show database 'DBFWDB_HA2';
```

Sample Output

```
$ dgmgrl /
DGMGRL for Linux: Version 12.1.0.2.0 - 64bit Production

Copyright (c) 2000, 2013, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
Connected as SYSDBG.
DGMGRL> show configuration;

Configuration - DBFWDB

Protection Mode: MaxAvailability
Members:
DBFWDB_HA2 - Primary database
DBFWDB_HA1 - (*) Physical standby database

Fast-Start Failover: ENABLED

Configuration Status:
SUCCESS (status updated 25 seconds ago)

DGMGRL> show database 'DBFWDB_HA1';

Database - DBFWDB_HA1

Role: PHYSICAL STANDBY
```

```
Intended State: APPLY-ON
Transport Lag: 0 seconds (computed 1 second ago)
Apply Lag: 0 seconds (computed 1 second ago)
Average Apply Rate: 2.00 KByte/s
Real Time Query: ON
Instance(s):
dbfwdb
```

```
Database Status:
SUCCESS
```

```
DGMGRL> show database 'DBFWDB_HA2';
```

```
Database - DBFWDB_HA2
```

```
Role: PRIMARY
Intended State: TRANSPORT-ON
Instance(s):
dbfwdb
```

```
Database Status:
SUCCESS
```

DBCS Endpoint Configuration Issues

Review these troubleshooting tips for commonly encountered DBCS endpoint configuration related issues when working with Oracle Key Vault.

- [SSH Tunnel Add Failure](#)
Adding an SSH tunnel fails on the Oracle Key Vault management Console

SSH Tunnel Add Failure

Adding an SSH tunnel fails on the Oracle Key Vault management Console

Probable Cause

The failure may be due to invalid settings for IP address, port, or user name.

Solution

Check if any of the following settings are invalid:

- Invalid IP address
- Invalid port
- Invalid user name
- The public SSH Oracle Key Vault key is not copied to the `authorized_keys` file of the Oracle Key Vault user on the Database as a Service instance.
- The Database as a Service instance is not reachable because of network overload.

To fix these issues, check your input values, your connection, and then retry.

Server and Node Issues

Review these troubleshooting tips for common server and node related errors when working with Oracle Key Vault.

- [SSL Client Error Message](#)
The alert and trace log display the SSL client error message when the Server Domain Name does not contain the expected Security Identifier (SID) name.
- [Incorrect Value Returned for Custom Attributes of Integer Type](#)
In certain scenarios, an invalid value is returned for an integer type custom attribute.
- [Not Receiving Email Alerts](#)
Even after configuring SMTP successfully, administrators are not receiving email alerts from the Oracle Key Vault server.
- [Oracle Key Vault Server and NTP Server Date and Time Not Synchronized](#)
Learn two methods to resolve Oracle Key Vault date and time unsynchronized issue.
- [Failed To Enable FIPS mode](#)
Enabling FIPS fails with an error on the Oracle Key Vault management console
- [Non-Oracle Database Endpoints Fails to Connect to Oracle Key Vault](#)
Non-Oracle Database endpoints such as MySQL, Mongo DB, fails to connect to Oracle Key Vault server during CA certificate rotation.

SSL Client Error Message

The alert and trace log display the SSL client error message when the Server Domain Name does not contain the expected Security Identifier (SID) name.

Alert log from Oracle Key Vault Server shows the following error message.

```
SSL Client: Server DN does not contain expected SID name
```

Probable Cause

These messages are from earlier SSL configurations.

Solution

Ignore these messages.

Incorrect Value Returned for Custom Attributes of Integer Type

In certain scenarios, an invalid value is returned for an integer type custom attribute.

This problem scenario applies to values that were created from an Oracle Key Vault version 21.2.0.0.0 or earlier. Retrieving an integer type custom attribute created by RESTful services utility using C or Java SDK may return an invalid value. It is also possible that a custom attribute results in an invalid value.

Probable Cause

In Oracle Key Vault versions 21.2.0.0.0 or earlier, when an integer type custom attribute is added or modified using RESTful services utility, the attribute value is stored in a representation that is different from the representation that is used by other interfaces, for example, C or JAVA SDK, and KMIP.

This means that a value created or modified using the RESTful service utility cannot be retrieved correctly with C or JAVA SDK and KMIP interfaces. Likewise, a value created or modified using C or JAVA SDK and KMIP interfaces could not be retrieved correctly using RESTful service utility and the Oracle Key Vault management console.

 **Note:**

The values that are created or modified using Oracle Key Vault 21.3.0.0.0 or later are always returned correctly.

The invalid values for a custom attribute of integer types are returned in these cases:

- The value was created from an Oracle Key Vault version 21.2.0.0.0 or earlier and the value has never been modified after upgrade to Oracle Key Vault 21.3.0.0.0 or later.
- The value was created or modified using RESTful services utility, but the value is retrieved using C/ or JAVA SDK or KMIP clients including PKCS#11 library.
- The value was created or modified using C or JAVA SDK or KMIP client, but the value is retrieved using REST CLI or the Oracle Key Vault management console.

The correct value is returned when the value is created and retrieved using the same interface.

In a multi-master cluster, Oracle Key Vault version in this section refers to the cluster version of the deployment.

Solution

To identify the values that are suspect and establish a correct value for them use the following procedure after the upgrade to Oracle Key Vault version 21.3.0.0.0 or later. A value for the custom attribute of integer type is considered suspect for cross utility use if it was created prior to Oracle Key Vault 21.3.0.0.0. This includes the values that may already be stored using the correct representation. Because from the value itself, it is not feasible to determine the representation, all such values are considered suspect and must be corrected.

1. Identify suspect values for the custom attributes of integer type:
 - a. Log in to the Oracle Key Vault server through SSH as user support, then switch user su to root.

```
ssh support@okv_server_IP_address
su - root
```

- b. Run the script to generate a report with the suspected values:

```
/usr/bin/su - okv -c
    /usr/local/okv/bin/gen_custom_attr_suspect_values
```

A report with the list of suspected values is generated:

```
/tmp/suspect_values_for_custom_attribute_integer_type.txt
```

For each entry in the report, following values are shown:

- **Creating Endpoint:** Endpoint that created the value.
- **Unique ID of Object:** Unique ID (UUID) of the object.

- **Custom Attribute Name:** Name of the custom attribute.
- **Index:** Index of the value.
- **SDK Value:** Value as retrieved by the C/JAVA SDK interfaces.
- **REST Value:** Value as retrieved by the RESTful services utility.

Between SDK and REST values, one of the values will be the correct value.

2. Update the suspect value with the chosen correct value.
For each suspect entry:

- a. Review the SDK and REST values.
- b. Determine the correct value of the custom attribute from the two possible values. One of the values will be the correct value of the custom attribute.
In some cases, a '-' may be shown for one of the values. In such cases, the correct value would be the one that is shown as the integer value.
- c. Update the custom attribute with the chosen correct value. You can use any interface to update this value.
You must update the custom attribute value even when one of the value is shown as '-'.

This step may require the coordination between the root user and the Oracle Key Vault users who can update the suspect values. A user who has the Key Administrator role can update all suspect values. In addition, a user or an endpoint who has read/write access on an object can modify the suspect custom attribute value for that object. You can determine such users using the endpoint information shown under 'Creating Endpoint'.

It is recommended to complete this procedure by establishing the correct value for all suspect values in one iteration. However, in case if it becomes necessary, above procedure could be repeated and then it will include only the remaining suspect values in the report.

It is recommended to verify the completion of this procedure by executing Step 1 again and ensuring that the generated report does not contain any suspect value entries.

Not Receiving Email Alerts

Even after configuring SMTP successfully, administrators are not receiving email alerts from the Oracle Key Vault server.

Probable Cause

Requires restart of tomcat service

Solution

1. Login to the Oracle Key Vault server or node through ssh and switch user to root.
2. Restart tomcat service.

```
$service tomcat status
$service tomcat stop
$service tomcat start
```

3. Verify if the tomcat service is up and running.

```
$service tomcat status  
ps -eaf | grep tomcat
```

4. Send a test email and see if the email is received.

Oracle Key Vault Server and NTP Server Date and Time Not Synchronized

Learn two methods to resolve Oracle Key Vault date and time unsynchronized issue.

Probable Cause

The time on the Oracle Key Vault server does not match with the time on the NTP server.

Solution

Perform the following steps to synchronize the time on the NTP and Oracle Key Vault server:

1. Log in to the Oracle Key Vault management console as a system administrator role.
2. Select the **System** tab, then **Settings** from the left navigation side bar.
3. In the Network Services area, Select **NTP** to display the **System Time** page.
4. Click **Apply Server** to perform a re-synchronization of the clock on the Oracle Key Vault server with the NTP server.

Alternatively, you can also perform the following steps to synchronize the time on the NTP and the Oracle Key Vault server:

1. Log in to the Oracle Key Vault server through SSH as user `support`, then switch user `su` to `root`.

```
ssh support@okv_server_IP_address  
su - root
```

2. Run the following command to perform a re-synchronization of the Oracle Key Vault server clock:

```
/bin/chronyc makestep
```

Failed To Enable FIPS mode

Enabling FIPS fails with an error on the Oracle Key Vault management console

Example

Failed to enable FIPS mode error seen on the Oracle key vault management console while enabling FIPS.

Probable Cause

The Oracle Key Vault server was upgraded from release 21.5 or previous releases and the FIPS was disabled before the upgrade.

Solution

1. As the support user, log in to the Oracle Key Vault server using SSH, and then switch user `su` to `root`.

```
ssh support@okv_server_IP_address  
su - root
```

2. Run the following command:

```
/usr/local/okv/bin/okv_fps_disable
```

3. Restart the Oracle Key Vault server using the `reboot` command.
4. As the `sysadmin` user, log in to the Oracle Key Vault management console to re-enable FIPS.

Non-Oracle Database Endpoints Fails to Connect to Oracle Key Vault

Non-Oracle Database endpoints such as MySQL, Mongo DB, fails to connect to Oracle Key Vault server during CA certificate rotation.

Example

Server communication or SSL peer certificate validation failed error seen on the Oracle Key Vault during CA certificate rotation.

Probable Cause

CA certificate rotation is not complete.

Solution

1. Log in to the Oracle Key Vault Management console, verify if the CA certificate rotation is still in progress.
2. Select and re-enroll all non-oracle endpoints from the **Endpoint** page if the certificate rotation is still in progress.
3. If certificate rotation is still in progress, then on endpoints page re-enroll all non-oracle endpoints selecting all of them.
4. When the CA certificate rotation is complete and the new CA certificate is active perform the following steps:
 - a. Go to the **Endpoint** page and copy the token id of the desired non-oracle endpoint.
 - b. Download the `okvclient.jar` file.
 - c. Copy the `okvclient.jar` file to the endpoint system.
 - d. Extract the `ssl` certificates and use it for the respective Database configuration.
5. After copying the `ssl` certificates, reboot the Database.
6. Verify if the issue is fixed.

D

Security Technical Implementation Guides Compliance Standards

Oracle Key Vault follows the Security Technical Implementation Guides (STIG)-based compliance standards.

- [About Security Technical Implementation Guides](#)
A Security Technical Implementation Guide (STIG) is a methodology followed by the U.S. Department of Defense (DOD).
- [Enabling and Disabling STIG Rules on Oracle Key Vault](#)
You can enable STIG rules on Oracle Key Vault by enabling Strict mode.
- [Current Implementation of STIG Rules on Oracle Key Vault](#)
Oracle has developed a security-hardened configuration of Oracle Key Vault that supports U.S. Department of Defense Security Technical Implementation Guide (STIG) recommendations.
- [Current Implementation of Database STIG Rules](#)
The current implementation of the database STIG rules encompass a wide range of rules.
- [Current Implementation of Operating System STIG Rules](#)
This topic contains information on the current implementation of operating system STIG guidelines for Oracle Key Vault.

About Security Technical Implementation Guides

A Security Technical Implementation Guide (STIG) is a methodology followed by the U.S. Department of Defense (DOD).

STIG is designed to reduce the attack surface of computer systems and networks, thereby ensuring a lockdown of highly confidential information stored within the DOD network. STIGs provide secure configuration standards for the DOD's Information Assurance (IA) and IA-enabled devices and systems. STIGs are created by the Defense Information Systems Agency (DISA).

For over a decade, Oracle has worked closely with the DOD to develop, publish, and maintain a growing list of STIGs for a variety of core Oracle products and technologies including:

- Oracle Database
- Oracle Solaris
- Oracle Linux
- Oracle WebLogic

When STIGs are updated, Oracle analyzes the latest recommendations in order to identify new ways to improve the security of its products by:

- Implementing new and innovative security capabilities that are then added to future STIG updates
- Delivering functionality to automate the assessment and implementation of STIG recommendations

- Improving "out of the box" security configuration settings based upon STIG recommendations

Enabling and Disabling STIG Rules on Oracle Key Vault

You can enable STIG rules on Oracle Key Vault by enabling Strict mode.

- [Enabling STIG Rules on Oracle Key Vault](#)
You enable STIG rules (strict mode) from the command line.
- [Disabling STIG Rules on Oracle Key Vault](#)
You disable STIG rules (strict mode) from the command line.

Enabling STIG Rules on Oracle Key Vault

You enable STIG rules (strict mode) from the command line.

1. Log in to the operating system of the Oracle Key Vault server as the root user.
2. Run the following command as `root`:

```
/usr/local/dbfw/bin/stig --enable
```

Note:

The setting `PASS_WARN_AGE` indicates how many days prior to password expiration that a warning will be issued for the users. Starting with Oracle Key Vault 21.6 as per the DOD requirement, the `PASS_WARN_AGE` value is set to 7, after enabling the STIG. Previously, `PASS_WARN_AGE` value was set to 60.

Disabling STIG Rules on Oracle Key Vault

You disable STIG rules (strict mode) from the command line.

1. Log in to the operating system of the Key Vault server as the root user.
2. Run the following command as `root`:

```
/usr/local/dbfw/bin/stig --disable
```

Current Implementation of STIG Rules on Oracle Key Vault

Oracle has developed a security-hardened configuration of Oracle Key Vault that supports U.S. Department of Defense Security Technical Implementation Guide (STIG) recommendations.

[Table D-1](#) lists the three vulnerability categories that STIG recommendations address.

Table D-1 Vulnerability Categories

Category	Description
CAT I	Any vulnerability, the exploitation of which will, directly and immediately result in loss of Confidentiality, Availability, or Integrity.

Table D-1 (Cont.) Vulnerability Categories

Category	Description
CAT II	Any vulnerability, the exploitation of which has a potential to result in loss of Confidentiality, Availability, or Integrity.
CAT III	Any vulnerability, the existence of which degrades measures to protect against loss of Confidentiality, Availability, or Integrity.

Current Implementation of Database STIG Rules

The current implementation of the database STIG rules encompass a wide range of rules.

Table D-2 shows the current implementation of Database STIG rules on Oracle Key Vault.

Table D-2 Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
DG0004-ORACLE11	Application object owner accounts should be disabled when not performing installation or maintenance actions.	CAT II	No	No	Currently not supported
DG0008-ORACLE11	Application objects should be owned by accounts authorized for ownership.	CAT II	No	Yes	<p>The below list of users are either Oracle Key Vault application users or those that appear to be from the Oracle Database installation.</p> <ul style="list-style-type: none"> • APPQOSSYS • DBSFUSER • REMOTE_SCHEDULER_AGENT • APEX_210200 • FLOWS_FILES • AVRULEOWNER • APEX_LISTENER • ORDS_METADATA • AUDSYS • GSMADMIN_INTERNAL • SECURELOG • ORACLE_OCM • MANAGEMENT • AVREPORTUSER • KEYVAULT • AVSYS • ACMGR

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
DG0014-ORACLE11	Default demonstration and sample database objects and applications should be removed.	CAT II	No	No	Implemented by default. All default demonstration and sample database objects have been removed.
DG0071-ORACLE11	New passwords must be required to differ from old passwords by more than four characters.	CAT II	No	No	Currently not supported
DG0073-ORACLE11	DBMS failed login account lock	CAT II	Yes	No	Implemented in strict mode. <code>FAILED_LOGIN_ATTEMPTS</code> is set to 3 for Oracle Key Vault user database profiles by the STIG script.
DG0075-ORACLE11	DBMS links to external databases	CAT II	No	No	Implemented by default
DG0077-ORACLE11	Production databases should be protected from unauthorized access by developers on shared production/development host systems.	CAT II	No	No	Implemented by default
DG0116-ORACLE11	Database privileged role assignments should be restricted to <i>IAO-authorized</i> DBMS accounts.	CAT II	No	Yes	See DG0116-ORACLE11 STIG Guideline Guideline.
DG0117-ORACLE11	Administrative privileges should be assigned to database accounts via database roles.	CAT II	No	No	Currently not supported
DG0121-ORACLE11	Application users privileges should be restricted to assignment using application user roles.	CAT II	No	No	Currently not supported
DG0123-ORACLE11	Access to DBMS system tables and other configuration or metadata should be restricted to DBAs.	CAT II	No	No	Currently not supported

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
DG0125-ORACLE11	DBMS account passwords should be set to expire every 60 days or more frequently.	CAT II	Yes	No	Implemented in strict mode. the <code>PASSWORD_LIFE_TIME</code> is set to 35 for Oracle Key Vault user database profiles by the STIG script.
DG0126-ORACLE11	Password reuse should be prevented where supported by the DBMS.	CAT II	No	No	Implemented by default
DG0128-ORACLE11	DBMS default accounts should be assigned custom passwords.	CAT I	Yes	No	Implemented in strict mode. Accounts such as <ul style="list-style-type: none"> • APPQOSSYS • AUDSYS • DBSNMP • DIP • DVF • GSMADMIN_INTERNAL • GSMCATUSER • GSMUSER • LBACSYS • MDDATA • ORACLE_OCM • OUTLN • SYSBACKUP • SYSDG • SYSKM • XDB are assigned a random password in the script.
DG0133-ORACLE11	Unlimited account lock times should be specified for locked accounts.	CAT II	Yes	No	Implemented in strict mode. The <code>PASSWORD_LOCK_TIME</code> is set to <code>UNLIMITED</code> for Oracle Key Vault user database profiles by the STIG script.
DG0141-ORACLE11	Attempts to bypass access controls should be audited.	CAT II	Yes	No	Implemented in strict mode. <code>DROP ANY SYNONYM</code> and <code>DROP ANY INDEXTYPE</code> are audited.
DG0142-ORACLE11	Changes to configuration options must be audited.	CAT II	No	No	Implemented by default
DG0192-ORACLE11	Remote database or other external access should use fully-qualified names.	CAT II	No	No	Currently not supported

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
DO0231-ORACLE11	Application owner accounts should have a dedicated application tablespace.	CAT II	No	No	Currently not supported
DO0250-ORACLE11	Fixed user and public database links should be authorized for use.	CAT II	No	No	Implemented by default
DO0270-ORACLE11	A minimum of two Oracle redo log groups/files should be defined and configured to be stored on separate, archived physical disks or archived directories on a RAID device.	CAT II	No	No	Currently not supported
DO0350-ORACLE11	Oracle system privileges should not be directly assigned to unauthorized accounts.	CAT II	No	No	Currently not supported
DO3475-ORACLE11	Execute permission should be revoked from PUBLIC for restricted Oracle packages.	CAT II	No	No	Currently not supported
DO3536-ORACLE11	The IDLE_TIME profile parameter should be set for Oracle profiles IAW DoD policy.	CAT II	Yes	No	Implemented in strict mode. IDLE_TIME is set to 15 for Oracle Key Vault user database profiles by the STIG script.
DO3540-ORACLE11	The Oracle SQL92_SECURITY parameter should be set to TRUE.	CAT II	No	No	Implemented by default
DO3609-ORACLE11	System privileges granted using the WITH ADMIN OPTION should not be granted to unauthorized user accounts.	CAT II	No	No	Currently not supported
DO3610-ORACLE11	Required object auditing should be configured.	CAT II	No	No	Currently not supported

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
DO3689-ORACLE11	Object permissions granted to PUBLIC should be restricted.	CAT II	No	No	Currently not supported
DO3696-ORACLE11	The Oracle RESOURCE_LIMIT parameter should be set to TRUE.	CAT II	No	No	Implemented by default
O121-BP-021900	The Oracle REMOTE_OS_AUTHENTICATION parameter must be set to FALSE.	CAT I	No	No	Implemented by default
O121-BP-022000	The Oracle REMOTE_OS_ROLES parameter must be set to FALSE.	CAT I	No	No	Implemented by default
O121-BP-022700	The Oracle Listener must be configured to require administration authentication.	CAT I	No	No	Implemented by default
O121-C1-004500	DBA OS accounts must be granted only those host system privileges necessary for the administration of the DBMS.	CAT I	No	Yes	In Oracle Key Vault, only OS user <i>oracle</i> can connect to the Database as SYSDBA. Oracle user is granted only necessary privileges.
O121-C1-011100	Oracle software must be evaluated and patched against newly found vulnerabilities.	CAT I	No	Yes	Apply latest Oracle Key Vault release bundle patch which patches OS, Database, Java on the Oracle Key Vault server.
O121-C1-015400	The DBMS, when using PKI-based authentication, must enforce authorized access to the corresponding private key.	CAT I	No	No	Implemented by default

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
O121-C1-019700	The DBMS must employ cryptographic mechanisms preventing the unauthorized disclosure of information during transmission unless the transmitted data is otherwise protected by alternative physical measures.	CAT I	No	No	Implemented by default
O121-N1-015601	Applications must obscure feedback of authentication information during the authentication process to protect the information from possible exploitation or use by unauthorized individuals.	CAT I	No	Yes	All passwords in Oracle Key Vault are either stored in Oracle Wallet or encrypted in the database.
O121-N1-015602	When using command-line tools such as Oracle SQL*Plus, which can accept a plain-text password, users must use an alternative login method that does not expose the password.	CAT I	No	No	Cannot fully comply. However, it is highly unlikely and it is not recommended to login to the Oracle Key Vault server and run scripts that needs passwords to be passed in it. Please contact Oracle support for any such requirement.
O121-OS-004600	Use of the DBMS software installation account must be restricted to DBMS software installation.	CAT I	No	No	Implemented by default
O121-BP-021300	Oracle instance names must not contain Oracle version numbers.	CAT II	No	No	Implemented by default
O121-BP-021400	Fixed user and public database links must be authorized for use.	CAT II	No	No	Implemented by default

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
O121-BP-022100	The Oracle <code>SQL92_SECURITY</code> parameter must be set to <code>TRUE</code> .	CAT II	No	No	Implemented by default
O121-BP-022200	The Oracle <code>REMOTE_LOGIN_PASSWORDFILE</code> parameter must be set to <code>EXCLUSIVE</code> or <code>NONE</code> .	CAT II	No	No	Currently not supported
O121-BP-022300	System privileges granted using the <code>WITH ADMIN OPTION</code> must not be granted to unauthorized user accounts.	CAT II	No	No	Implemented by default
O121-BP-022400	System privileges must not be granted to <code>PUBLIC</code> role.	CAT II	No	No	Implemented by default
O121-BP-022500	Oracle roles granted using the <code>WITH ADMIN OPTION</code> must not be granted to unauthorized accounts.	CAT II	No	No	Implemented by default
O121-BP-022600	Object permissions granted to <code>PUBLIC</code> role must be restricted.	CAT II	No	No	Implemented by default
O121-BP-022800	Application role permissions must not be assigned to the Oracle <code>PUBLIC</code> role.	CAT II	No	No	Implemented by default
O121-BP-023000	Connections by mid-tier web and application systems to the Oracle DBMS must be protected, encrypted, and authenticated according to database, web, application, enclave, and network requirements.	CAT II	No	Yes	All communications between Oracle Key Vault and endpoints is over TLS.

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
O121-BP-023200	Unauthorized database links must not be defined and left active.	CAT II	No	No	Implemented by default
O121-BP-023600	Only authorized system accounts must have the SYSTEM table space specified as the default table space.	CAT II	No	No	Implemented by default
O121-BP-023900	The Oracle <code>_TRACE_FILES_PUBLIC</code> parameter if present must be set to FALSE.	CAT II	No	No	Implemented by default
O121-BP-025200	Credentials stored and used by the DBMS to access remote databases or applications must be authorized and restricted to authorized users.	CAT II	No	No	Implemented by default
O121-BP-025700	DBMS data files must be dedicated to support individual applications.	CAT II	No	No	Implemented by default
O121-BP-025800	Changes to configuration options must be audited.	CAT II	No	No	Implemented by default
O121-BP-026600	Network client connections must be restricted to supported versions.	CAT II	No	No	Implemented by default

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
O121-C2-003000	The DBMS must enforce Discretionary Access Control (DAC) policy allowing users to specify and control sharing by named individuals, groups of individuals, or by both, limiting propagation of access rights and including or excluding access to the granularity of a single user.	CAT II	No	No	Implemented by default
O121-C2-003400	DBMS processes or services must run under custom and dedicated OS accounts.	CAT II	No	No	Implemented by default
O121-C2-003600	A single database connection configuration file must not be used to configure all database clients.	CAT II	No	Yes	Clients of Oracle Key Vault cannot access its internal database.
O121-C2-006700	A DBMS utilizing Discretionary Access Control (DAC) must enforce a policy that includes or excludes access to the granularity of a single user.	CAT II	No	No	Implemented by default
O121-C2-006900	The DBMS must allow designated organizational personnel to select specific events that can be audited by the database.	CAT II	No	No	Implemented by default
O121-C2-011500	Default demonstration, sample databases, database objects, and applications must be removed.	CAT II	No	No	Implemented by default

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
O121-C2-011600	Unused database components, DBMS software, and database objects must be removed.	CAT II	No	No	Implemented by default
O121-C2-011700	Unused database components that are integrated in the DBMS and cannot be uninstalled must be disabled.	CAT II	No	No	Currently not supported
O121-C2-014600	The DBMS must support organizational requirements to enforce password encryption for storage.	CAT II	No	No	Implemented by default
O121-C2-015100	DBMS passwords must not be stored in compiled, encoded, or encrypted batch jobs or compiled, encoded, or encrypted application source code.	CAT II	No	No	Implemented by default
O121-C1-015000	DBMS default accounts must be assigned custom passwords.	CAT I	No	No	Implemented by default
O121-C2-002100	The DBMS must automatically disable accounts after a period of 35 days of account inactivity.	CAT II	Yes	No	Implemented in strict mode. <code>PASSWORD_LIFE_TIME</code> is set to 35 for Oracle Key Vault user database profiles by the STIG script.
O121-C2-004900	The DBMS must verify account lockouts and persist until reset by an administrator.	CAT II	Yes	No	None

Table D-2 (Cont.) Current Implementation of Database STIG Rules

STIG ID	Title	Severity	Addressed by Script	Addressed by Documentation	Notes
O121-C2-013800	The DBMS must support organizational requirements to disable user accounts after a defined time period of inactivity set by the organization.	CAT II	Yes	No	Implemented in strict mode. <code>PASSWORD_LIFE_TIME</code> is set to 35 for Oracle Key Vault user database profiles by the STIG script.
O121-C2-015200	The DBMS must enforce password maximum lifetime restrictions.	CAT II	Yes	No	Implemented in strict mode. <code>EFFECTIVE_LIFE_TIME (PASSWORD_LIFE_TIME + PASSWORD_GRACE_TIME)</code> is set to 35 for Oracle Key Vault user database profiles by the STIG script.

When you apply the database STIG rules in Oracle Key Vault, the default values listed below applies to the user account profile parameters.

Table D-3 User Account Profile Parameters

Parameter	Description	Default Value	Default Value in Strict STIG mode
Failed Login Attempts	Number of consecutive failed attempts to log in to the user account before the account is locked.	3	3
Password Life Time (in days)	Number of days the same password can be used for authentication.	180	35
Password Grace Time (in days)	Number of days after the grace period begins during which a warning is issued, and login is allowed.	5	0
Password Reuse Max	Number of password changes required before the current password can be reused.	UNLIMITED	UNLIMITED
Password Reuse Time (in days)	Number of days before which a password cannot be reused.	365	365

Table D-3 (Cont.) User Account Profile Parameters

Parameter	Description	Default Value	Default Value in Strict STIG mode
Password Lock Time (in days)	Number of days an account will be locked after the specified number of consecutive failed login attempts. After the time passes, then the account becomes unlocked.	1	UNLIMITED

You can configure user account profile parameters to use non-default values even when Oracle Key Vault is operating with STIG rules enabled (strict mode).

If you configure a value that is less strict than the corresponding non-STIG default value, a warning message, as shown below, is displayed indicating that the current value is not STIG compliant.

The above warning is displayed when the values are set to less stricter values than the STIG recommendation. In the above figure, **Failed Login Attempts** is set the **5** where as the STIG recommended value is **3** and hence the warning is displayed.

The **Save Defaults** operation resets the user account profile parameters to their default values. If STIG rules are enabled (strict mode), then profile parameters values are restored to the default values used for the strict STIG mode configuration. Otherwise, default values that apply to STIG configuration are restored.

When you enable strict STIG mode on a multi-master cluster node, user account profile parameter values are changed to the strict STIG mode default values. However, these changes are applied only to the current node. When enabling strict STIG mode, Oracle recommends that you first enable the strict STIG mode on all cluster nodes individually. Afterward, you can optionally configure the non-default values from any cluster node.

- [Additional STIG Guidelines Notes](#)
Learn about additional advice regarding STIG guidelines.

Related Topics

- [Managing User Account Profile Parameters](#)
You can manage the user account profile parameters in Oracle Key Vault.

Additional STIG Guidelines Notes

Learn about additional advice regarding STIG guidelines.

- [DG0116-ORACLE11 STIG Guideline](#)
This section provides information about the STIG guideline for DG0116-ORACLE11.

DG0116-ORACLE11 STIG Guideline

This section provides information about the STIG guideline for DG0116-ORACLE11.

Table D-4 Accounts and Role Assignments in Oracle Key Vault

Account	Role Assignment
OKV#OGG_DB_USER	<ul style="list-style-type: none"> • SELECT_CATALOG_ROLE • XDBADMIN
AV_ADMIN	<ul style="list-style-type: none"> • AQ_ADMINISTRATOR_ROLE • SELECT_CATALOG_ROLE • XDBADMIN

Current Implementation of Operating System STIG Rules

This topic contains information on the current implementation of operating system STIG guidelines for Oracle Key Vault.

[Current Implementation of Operating System STIG Rules](#) shows the current implementation of Operating System STIG Rules on Oracle Key Vault.

Table D-5 Operating System STIG Guideline Set Reference

Reference	Detail
Document	Oracle Linux 8 Security Technical Implementation Guide
Version	1
Release	5
Release date	January 13, 2023
Document link	Oracle Linux Security Technical Implementation Guide

Table D-6 User Action - Definition and Guidelines

User Action	Description of the Guideline
None	The guideline is implemented by default and no user action is required.

Table D-6 (Cont.) User Action - Definition and Guidelines

User Action	Description of the Guideline
Enable <i>strict</i> mode	The guideline can be implemented by switching the appliance to <i>strict</i> mode.
	 See Also: Enabling STIG Rules on Oracle Key Vault
Site policy	The guideline can be implemented depending on local policy and it requires administrator action. See the Notes column for additional information on implementation.
Administrative task	The guideline implementation is an administrator configuration action after installation or upgrade. It can also be a regularly used and defined administrative procedure.

Table D-7 Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-010000	CAT I	-	OL 8 must be a vendor-supported release.	Implemented by default
OL08-00-010140	CAT I	-	OL 8 operating systems booted with United Extensible Firmware Interface (UEFI) must require authentication upon booting into single-user mode and maintenance.	Not applicable
OL08-00-010150	CAT I	-	OL 8 operating systems booted with a BIOS must require authentication upon booting into single-user and maintenance modes.	Not applicable

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-010370	CAT I	-	YUM must be configured to prevent the installation of patches, service packs, device drivers, or OL 8 system components that have not been digitally signed using a certificate that is recognized and approved by the organization.	Implemented by default
OL08-00-010460	CAT I	-	There must be no <code>shosts.equiv</code> files on the OL 8 operating system.	Implemented by default
OL08-00-010470	CAT I	-	There must be no <code>.shosts</code> files on the OL 8 operating system.	Implemented by default
OL08-00-010820	CAT I	-	Unattended or automatic logon via the OL 8 graphical user interface must not be allowed.	Not applicable
OL08-00-010830	CAT I	-	OL 8 must not allow users to override SSH environment variables.	Implemented by default
OL08-00-020330	CAT I	-	OL 8 must not allow accounts configured with blank or null passwords.	Implemented by default
OL08-00-020331	CAT I	-	OL 8 must not allow blank or null passwords in the <code>system-auth</code> file.	Implemented by default
OL08-00-020332	CAT I	-	OL 8 must not allow blank or null passwords in the <code>password-auth</code> file.	Implemented by default
OL08-00-040000	CAT I	-	OL 8 must not have the <code>telnet-server</code> package installed.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-040010	CAT I	-	OL 8 must not have the <code>rsh-server</code> package installed.	Implemented by default
OL08-00-040171	CAT I	-	The x86 Ctrl-Alt-Delete key sequence in OL 8 must be disabled if a graphical user interface is installed.	Not applicable
OL08-00-040190	CAT I	-	The Trivial File Transfer Protocol (TFTP) server package must not be installed if not required for OL 8 operational support.	Implemented by default
OL08-00-040200	CAT I	-	The root account must be the only account having unrestricted access to the OL 8 system.	Implemented by default
OL08-00-040360	CAT I	-	A File Transfer Protocol (FTP) server package must not be installed unless mission essential on OL 8.	Implemented by default
OL08-00-010049	CAT II	-	OL 8 must display a banner before granting local or remote access to the system via a graphical user logon.	Not applicable
OL08-00-010110	CAT II	-	OL 8 must encrypt all stored passwords with a FIPS 140-2 approved cryptographic hashing algorithm.	Implemented by default
OL08-00-010120	CAT II	-	OL 8 must employ FIPS 140-2 approved cryptographic hashing algorithms for all stored passwords.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-010130	CAT II	-	The OL 8 shadow password suite must be configured to use a sufficient number of hashing rounds.	Implemented by default
OL08-00-010151	CAT II	-	OL 8 operating systems must require authentication upon booting into rescue mode.	Implemented by default
OL08-00-010152	CAT II	-	OL 8 operating systems must require authentication upon booting into emergency mode.	Implemented by default
OL08-00-010159	CAT II	-	The OL 8 <code>pam_unix.so</code> module must be configured in the <code>system-auth</code> file to use a FIPS 140-2 approved cryptographic hashing algorithm for system authentication.	Implemented by default
OL08-00-010160	CAT II	-	The OL 8 <code>pam_unix.so</code> module must be configured in the <code>password-auth</code> file to use a FIPS 140-2 approved cryptographic hashing algorithm for system authentication.	Implemented by default
OL08-00-010161	CAT II	-	OL 8 must prevent system daemons from using Kerberos for authentication.	Implemented by default
OL08-00-010162	CAT II	-	The <code>krb5-workstation</code> package must not be installed on OL 8.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-010163	CAT II	-	The krb5-server package must not be installed on OL 8.	Implemented by default
OL08-00-010200	CAT II	-	OL 8 must be configured so that all network connections associated with SSH traffic are terminate after a period of inactivity.	Implemented by default
OL08-00-010210	CAT II	-	The OL 8 /var/log/messages file must have mode 0640 or less permissive.	Implemented by default
OL08-00-010220	CAT II	-	The OL 8 /var/log/messages file must be owned by root.	Implemented by default
OL08-00-010230	CAT II	-	The OL 8/var/log/messages file must be group-owned by root.	Implemented by default
OL08-00-010240	CAT II	-	The OL 8 /var/log directory must have mode 0755 or less permissive.	Implemented by default
OL08-00-010250	CAT II	-	The OL 8 /var/log directory must be owned by root.	Implemented by default
OL08-00-010260	CAT II	-	The OL 8 /var/log directory must be group-owned by root.	Implemented by default
OL08-00-010294	CAT II	-	The OL 8 operating system must implement DoD-approved TLS encryption in the OpenSSL package.	Implemented by default
OL08-00-010372	CAT II	-	OL 8 must prevent the loading of a new kernel for later execution.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-010373	CAT II	-	OL 8 must enable kernel parameters to enforce Discretionary Access Control (DAC) on symlinks.	Implemented by default
OL08-00-010374	CAT II	-	OL 8 must enable kernel parameters to enforce Discretionary Access Control (DAC) on hardlinks.	Implemented by default
OL08-00-010381	CAT II	-	OL 8 must require users to re-authenticate for privilege escalation and changing roles.	Implemented by default
OL08-00-010382	CAT II	-	OL 8 must restrict privilege elevation to authorized personnel.	Implemented by default
OL08-00-010480	CAT II	-	The OL 8 SSH public host key files must have mode 0644 or less permissive.	Implemented by default
OL08-00-010500	CAT II	-	The OL 8 SSH daemon must perform strict mode checking of home directory configuration files.	Implemented by default
OL08-00-010520	CAT II	-	The OL 8 SSH daemon must not allow authentication using known host's authentication.	Implemented by default
OL08-00-010521	CAT II	-	The OL 8 SSH daemon must not allow Kerberos authentication, except to fulfill documented and validated mission requirements.	Implemented by default
OL08-00-010522	CAT II	-	The OL 8 SSH daemon must not allow GSSAPI authentication, except to fulfill documented and validated mission requirements.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-010543	CAT II	-	OL 8 must use a separate file system for <code>/tmp</code> .	Implemented by default
OL08-00-010550	CAT II	-	OL 8 must not permit direct logons to the root account using remote access via SSH.	Implemented by default
OL08-00-010561	CAT II	-	OL 8 must have the <code>rsyslog</code> service enabled and active.	Implemented by default
OL08-00-010571	CAT II	-	OL 8 must prevent files with the <code>setuid</code> and <code>setgid</code> bit set from being executed on the <code>/boot</code> directory.	Implemented by default
OL08-00-010630	CAT II	-	OL 8 file systems must not execute binary files that are imported via Network File System (NFS).	Not applicable
OL08-00-010640	CAT II	-	OL 8 file systems must not interpret character or block special devices that are imported via NFS.	Not applicable
OL08-00-010650	CAT II	-	OL 8 must prevent files with the <code>setuid</code> and <code>setgid</code> bit set from being executed on file systems that are imported via Network File System (NFS).	Not applicable
OL08-00-010760	CAT II	-	All OL 8 local interactive user accounts must be assigned a home directory upon creation.	Implemented by default
OL08-00-020010	CAT II	-	OL 8 systems below version 8.2 must automatically lock an account when three unsuccessful logon attempts occur.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-020011	CAT II	-	OL 8 systems, versions 8.2 and above, must automatically lock an account when three unsuccessful logon attempts occur.	Implemented by default
OL08-00-020012	CAT II	-	OL 8 systems below version 8.2 must automatically lock an account when three unsuccessful logon attempts occur during a 15-minute time period.	Implemented by default
OL08-00-020013	CAT II	-	OL 8 systems, versions 8.2 and above, must automatically lock an account when three unsuccessful logon attempts occur during a 15-minute time period.	Implemented by default
OL08-00-020014	CAT II	-	OL 8 systems below version 8.2 must automatically lock an account until the locked account is released by an administrator when three unsuccessful logon attempts occur during a 15-minute time period.	Not applicable
OL08-00-020018	CAT II	-	OL 8 systems below version 8.2 must prevent system messages from being presented when three unsuccessful logon attempts occur.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-020019	CAT II	-	OL 8 systems, versions 8.2 and above, must prevent system messages from being presented when three unsuccessful logon attempts occur.	Implemented by default
OL08-00-020020	CAT II	-	OL 8 systems below version 8.2 must log user name information when unsuccessful logon attempts occur.	Implemented by default
OL08-00-020021	CAT II	-	OL 8 systems, versions 8.2 and above, must log user name information when unsuccessful logon attempts occur.	Implemented by default
OL08-00-020022	CAT II	-	OL 8 systems below version 8.2 must include root when automatically locking an account until the locked account is released by an administrator when three unsuccessful logon attempts occur during a 15-minute time period.	Implemented by default
OL08-00-020039	CAT II	-	OL 8 must have the tmux package installed.	Implemented by default
OL08-00-020100	CAT II	-	OL 8 must ensure the password complexity module is enabled in the password-auth file.	Implemented by default
OL08-00-020140	CAT II	-	OL 8 must require the maximum number of repeating characters of the same character class be limited to four when passwords are changed.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-020150	CAT II	-	OL 8 must require the maximum number of repeating characters be limited to three when passwords are changed.	Implemented by default
OL08-00-020160	CAT II	-	OL 8 must require the change of at least four character classes when passwords are changed.	Implemented by default
OL08-00-020180	CAT II	-	OL 8 passwords for new users or password changes must have a 24 hours/1 day minimum password lifetime restriction in <code>/etc/shadow</code> .	Implemented by default
OL08-00-020190	CAT II	-	OL 8 passwords for new users or password changes must have a 24 hours/1 day minimum password lifetime restriction in <code>/etc/login.defs</code> .	Implemented by default
OL08-00-020200	CAT II	enable strict mode	OL 8 user account passwords must have a 60-day maximum password lifetime restriction.	Implemented in strict mode
OL08-00-020210	CAT II	enable strict mode	OL 8 user account passwords must be configured so that existing passwords are restricted to a 60-day maximum lifetime.	Implemented in strict mode
OL08-00-020230	CAT II	-	OL 8 passwords must have a minimum of 15 characters.	Implemented by default
OL08-00-020231	CAT II	enable strict mode	OL 8 passwords for new users must have a minimum of 15 characters.	Implemented in strict mode

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-020263	CAT II	-	The OL 8 lastlog command must be owned by root.	Implemented by default
OL08-00-020264	CAT II	-	The OL 8 lastlog command must be group-owned by root.	Implemented by default
OL08-00-020300	CAT II	-	OL 8 must prevent the use of dictionary words for passwords.	Implemented by default
OL08-00-020310	CAT II	-	OL 8 must enforce a delay of at least four seconds between logon prompts following a failed logon attempt.	Implemented by default
OL08-00-020350	CAT II	-	OL 8 must display the date and time of the last successful account logon upon an SSH logon.	Implemented by default
OL08-00-020351	CAT II	-	OL 8 default permissions must be defined in such a way that all authenticated users can read and modify only their own files.	Implemented by default
OL08-00-030000	CAT II	-	The OL 8 audit system must be configured to audit the execution of privileged functions and prevent all software from executing at higher privilege levels than users executing the software.	Implemented by default
OL08-00-030020	CAT II	-	The OL 8 System Administrator (SA) and Information System Security Officer (ISSO) (at a minimum) must be alerted of an audit processing failure event.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-030040	CAT II	-	The OL 8 System must take appropriate action when an audit processing failure occurs.	Implemented by default
OL08-00-030060	CAT II	-	The OL 8 audit system must take appropriate action when the audit storage volume is full.	Implemented by default
OL08-00-030061	CAT II	-	The OL 8 audit system must audit local events.	Implemented by default
OL08-00-030062	CAT II	-	OL 8 must label all offloaded audit logs before sending them to the central log server.	Implemented by default
OL08-00-030063	CAT II	-	OL 8 must resolve audit information before writing to disk.	Implemented by default
OL08-00-030080	CAT II	-	OL 8 audit logs must be owned by root to prevent unauthorized read access.	Implemented by default
OL08-00-030100	CAT II	-	The OL 8 audit log directory must be owned by root to prevent unauthorized read access.	Implemented by default
OL08-00-030121	CAT II	-	The OL 8 audit system must protect auditing rules from unauthorized change.	Implemented by default
OL08-00-030122	CAT II	-	The OL 8 audit system must protect logon UIDs from unauthorized change.	Implemented by default
OL08-00-030130	CAT II	-	OL 8 must generate audit records for all account creation events that affect /etc/shadow.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-030140	CAT II	-	OL 8 must generate audit records for all account creation events that affect /etc/security/opasswd.	Implemented by default
OL08-00-030150	CAT II	-	OL 8 must generate audit records for all account creation events that affect /etc/passwd.	Implemented by default
OL08-00-030160	CAT II	-	OL 8 must generate audit records for all account creation events that affect /etc/gshadow.	Implemented by default
OL08-00-030170	CAT II	-	OL 8 must generate audit records for all account creation events that affect /etc/group.	Implemented by default
OL08-00-030171	CAT II	-	OL 8 must generate audit records for all account creations, modifications, disabling, and termination events that affect /etc/sudoers.	Implemented by default
OL08-00-030172	CAT II	-	OL 8 must generate audit records for all account creations, modifications, disabling, and termination events that affect /etc/sudoers.d/.	Implemented by default
OL08-00-030180	CAT II	-	The OL 8 audit package must be installed.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-030181	CAT II	-	OL 8 audit records must contain information to establish what type of events occurred, the source of events, where events occurred, and the outcome of events.	Implemented by default
OL08-00-030190	CAT II	-	OL 8 must generate audit records for any use of the <code>su</code> command.	Implemented by default
OL08-00-030200	CAT II	-	The OL 8 audit system must be configured to audit any use of the <code>setxattr</code> , <code>fsetxattr</code> , <code>lsetxattr</code> , <code>removexattr</code> , <code>fremovexattr</code> , and <code>lremovexattr</code> system calls.	Implemented by default
OL08-00-030250	CAT II	-	OL 8 must generate audit records for any use of the <code>chage</code> command.	Implemented by default
OL08-00-030260	CAT II	-	OL 8 must generate audit records for any uses of the <code>chcon</code> command.	Implemented by default
OL08-00-030280	CAT II	-	OL 8 must generate audit records for any use of the <code>ssh-agent</code> command.	Implemented by default
OL08-00-030290	CAT II	-	OL 8 must generate audit records for any use of the <code>passwd</code> command.	Implemented by default
OL08-00-030300	CAT II	-	OL 8 must generate audit records for any use of the <code>mount</code> command.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-030301	CAT II	-	OL 8 must generate audit records for any use of the <code>umount</code> command.	Implemented by default
OL08-00-030302	CAT II	-	OL 8 must generate audit records for any use of the <code>mount</code> syscall.	Implemented by default
OL08-00-030310	CAT II	-	OL 8 must generate audit records for any use of the <code>unix_update</code> command.	Implemented by default
OL08-00-030311	CAT II	-	OL 8 must generate audit records for any use of the <code>postdrop</code> command.	Implemented by default
OL08-00-030312	CAT II	-	OL 8 must generate audit records for any use of the <code>postqueue</code> command.	Implemented by default
OL08-00-030313	CAT II	-	OL 8 must generate audit records for any use of the <code>semanage</code> command.	Implemented by default
OL08-00-030314	CAT II	-	OL 8 must generate audit records for any use of the <code>setfiles</code> command.	Implemented by default
OL08-00-030315	CAT II	-	OL 8 must generate audit records for any use of the <code>userhelper</code> command.	Implemented by default
OL08-00-030316	CAT II	-	OL 8 must generate audit records for any use of the <code>setsebool</code> command.	Implemented by default
OL08-00-030317	CAT II	-	OL 8 must generate audit records for any use of the <code>unix_chkpwd</code> command.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-030320	CAT II	-	OL 8 must generate audit records for any use of the <code>ssh-keysign</code> command.	Implemented by default
OL08-00-030330	CAT II	-	OL 8 must generate audit records for any use of the <code>setfacl</code> command.	Implemented by default
OL08-00-030340	CAT II	-	OL 8 must generate audit records for any use of the <code>pam_timestamp_check</code> command.	Implemented by default
OL08-00-030350	CAT II	-	OL 8 must generate audit records for any use of the <code>newgrp</code> command.	Implemented by default
OL08-00-030360	CAT II	-	OL 8 must generate audit records for any use of the <code>init_module</code> and <code>finit_module</code> system calls.	Implemented by default
OL08-00-030361	CAT II	-	OL 8 must generate audit records for any use of the <code>rename</code> , <code>unlink</code> , <code>rmdir</code> , <code>renameat</code> , and <code>unlinkat</code> system calls.	Implemented by default
OL08-00-030370	CAT II	-	OL 8 must generate audit records for any use of the <code>gpasswd</code> command.	Implemented by default
OL08-00-030390	CAT II	-	OL 8 must generate audit records for any use of the <code>delete_module</code> syscall.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-030400	CAT II	-	OL 8 must generate audit records for any use of the <code>crontab</code> command.	Implemented by default
OL08-00-030410	CAT II	-	OL 8 must generate audit records for any use of the <code>chsh</code> command.	Implemented by default
OL08-00-030420	CAT II	-	OL 8 must generate audit records for any use of the <code>truncate</code> , <code>ftruncate</code> , <code>creat</code> , <code>open</code> , <code>openat</code> , and <code>open_by_handle_at</code> system calls.	Implemented by default
OL08-00-030480	CAT II	-	OL 8 must generate audit records for any use of the <code>chown</code> , <code>fchown</code> , <code>fchownat</code> , and <code>lchown</code> system calls.	Implemented by default
OL08-00-030490	CAT II	-	OL 8 must generate audit records for any use of the <code>chmod</code> , <code>fchmod</code> , and <code>fchmodat</code> system calls.	Implemented by default
OL08-00-030550	CAT II	-	OL 8 must generate audit records for any use of the <code>sudo</code> command.	Implemented by default
OL08-00-030560	CAT II	-	OL 8 must generate audit records for any use of the <code>usermod</code> command.	Implemented by default
OL08-00-030570	CAT II	-	OL 8 must generate audit records for any use of the <code>chacl</code> command.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-030580	CAT II	-	OL 8 must generate audit records for any use of the <code>kmod</code> command.	Implemented by default
OL08-00-030600	CAT II	-	OL 8 must generate audit records for any attempted modifications to the <code>lastlog</code> file.	Implemented by default
OL08-00-030610	CAT II	-	OL 8 must allow only the Information System Security Manager (ISSM) (or individuals or roles appointed by the ISSM) to select which auditable events are to be audited.	Implemented by default
OL08-00-030620	CAT II	-	OL 8 audit tools must have a mode of 0755 or less permissive.	Implemented by default
OL08-00-030630	CAT II	-	OL 8 audit tools must be owned by root.	Implemented by default
OL08-00-030640	CAT II	-	OL 8 audit tools must be group-owned by root.	Implemented by default
OL08-00-030670	CAT II	-	OL 8 must have the packages required for offloading audit logs installed.	Implemented by default
OL08-00-030700	CAT II	-	OL 8 must take appropriate action when the internal event queue is full.	Implemented by default
OL08-00-040001	CAT II	-	OL 8 must not have any automated bug reporting tools installed.	Implemented by default
OL08-00-040002	CAT II	-	OL 8 must not have the <code>sendmail</code> package installed.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-040021	CAT II		OL 8 must not have the asynchronous transfer mode (ATM) kernel module installed if not required for operational support.	Implemented by default
OL08-00-040022	CAT II		OL 8 must not have the Controller Area Network (CAN) kernel module installed if not required for operational support.	Implemented by default
OL08-00-040023	CAT II		OL 8 must not have the stream control transmission protocol (SCTP) kernel module installed if not required for operational support.	Implemented by default
OL08-00-040080	CAT II	-	OL 8 must be configured to disable the ability to use USB mass storage devices.	Implemented by default
OL08-00-040111	CAT II	-	OL 8 Bluetooth must be disabled.	Implemented by default
OL08-00-040129	CAT II	-	OL 8 must mount /var/log/audit with the nodev option.	Not applicable
OL08-00-040130	CAT II	-	OL 8 must mount /var/log/audit with the nosuid option.	Not applicable
OL08-00-040131	CAT II	-	OL 8 must mount /var/log/audit with the noexec option.	Not applicable

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-040160	CAT II	-	All OL 8 networked systems must have and implement SSH to protect the confidentiality and integrity of transmitted and received information, as well as information during preparation for transmission.	Implemented by default
OL08-00-040161	CAT II	-	OL 8 must force a frequent session key renegotiation for SSH connections to the server.	Implemented by default
OL08-00-040209	CAT II	-	OL 8 must prevent IPv4 Internet Control Message Protocol (ICMP) redirect messages from being accepted.	Implemented by default
OL08-00-040210	CAT II	-	OL 8 must prevent IPv6 Internet Control Message Protocol (ICMP) redirect messages from being accepted.	Not applicable
OL08-00-040220	CAT II	-	OL 8 must not send Internet Control Message Protocol (ICMP) redirects.	Implemented by default
OL08-00-040230	CAT II	-	OL 8 must not respond to Internet Control Message Protocol (ICMP) echoes sent to a broadcast address.	Implemented by default
OL08-00-040239	CAT II	-	OL 8 must not forward IPv4 source-routed packets.	Implemented by default
OL08-00-040240	CAT II	-	OL 8 must not forward IPv6 source-routed packets.	Not applicable

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-040249	CAT II	-	OL 8 must not forward IPv4 source-routed packets by default.	Implemented by default
OL08-00-040250	CAT II	-	OL 8 must not forward IPv6 source-routed packets by default.	Not applicable
OL08-00-040260	CAT II	-	OL 8 must not enable IPv6 packet forwarding unless the system is a router.	Not applicable
OL08-00-040261	CAT II	-	OL 8 must not accept router advertisements on all IPv6 interfaces.	Not applicable
OL08-00-040262	CAT II	-	OL 8 must not accept router advertisements on all IPv6 interfaces by default.	Not applicable
OL08-00-040270	CAT II	-	OL 8 must not allow interfaces to perform Internet Control Message Protocol (ICMP) redirects by default.	Implemented by default
OL08-00-040279	CAT II	-	OL 8 must ignore IPv4 Internet Control Message Protocol (ICMP) redirect messages.	Implemented by default
OL08-00-040280	CAT II	-	OL 8 must ignore IPv6 Internet Control Message Protocol (ICMP) redirect messages.	Not applicable
OL08-00-040281	CAT II	-	OL 8 must disable access to the network <code>bpf</code> syscall from unprivileged processes.	Implemented by default
OL08-00-040283	CAT II	-	OL 8 must restrict exposed kernel pointer addresses access.	Implemented by default
OL08-00-040284	CAT II	-	OL 8 must disable the use of user namespaces.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-040285	CAT II	-	OL 8 must use reverse path filtering on all IPv4 interfaces.	Implemented by default
OL08-00-040286	CAT II	-	OL 8 must enable hardening for the Berkeley Packet Filter Just-in-time compiler.	Implemented by default
OL08-00-040290	CAT II	-	OL 8 must be configured to prevent unrestricted mail relaying.	Not applicable
OL08-00-040340	CAT II	-	OL 8 remote X connections for interactive users must be disabled unless to fulfill documented and validated mission requirements.	Implemented by default
OL08-00-040341	CAT II	-	The OL 8 SSH daemon must prevent remote hosts from connecting to the proxy display.	Implemented by default
OL08-00-040350	CAT II	-	If the Trivial File Transfer Protocol (TFTP) server is required, the OL 8 TFTP daemon must be configured to operate in secure mode.	Not applicable
OL08-00-040390	CAT II	-	OL 8 must not have the <code>tuned</code> package installed if not required for operational support.	Implemented by default
OL08-00-010171	CAT III	-	OL 8 must have the <code>policycoreutils</code> package installed.	Implemented by default
OL08-00-010292	CAT III	-	The OL 8 SSH server must be configured to use strong entropy.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-010375	CAT III	-	OL 8 must restrict access to the kernel message buffer.	Implemented by default
OL08-00-010376	CAT III	-	OL 8 must prevent kernel profiling by unprivileged users.	Implemented by default
OL08-00-010390	CAT III	-	OL 8 must have the package required for multifactor authentication installed.	Implemented by default
OL08-00-010440	CAT III	-	YUM must remove all software components after updated versions have been installed on OL 8.	Implemented by default
OL08-00-010541	CAT III	-	OL 8 must use a separate file system for <code>/var/log</code> .	Implemented by default
OL08-00-020024	CAT III	-	OL 8 must limit the number of concurrent sessions to 10 for all accounts and/or account types.	Implemented by default
OL08-00-020110	CAT III	-	OL 8 must enforce password complexity by requiring that at least one uppercase character be used.	Implemented by default
OL08-00-020120	CAT III	-	OL 8 must enforce password complexity by requiring that at least one lowercase character be used.	Implemented by default
OL08-00-020130	CAT III	-	OL 8 must enforce password complexity by requiring that at least one numeric character be used.	Implemented by default
OL08-00-020170	CAT III	-	OL 8 must require the change of at least 8 characters when passwords are changed.	Implemented by default

Table D-7 (Cont.) Current Implementation of Operating System STIG Guidelines for Oracle Key Vault

STIG ID	Severity	User Action	Title	Notes
OL08-00-020220	CAT III	-	OL 8 must be configured in the <code>password-auth</code> file to prohibit password reuse for a minimum of five generations.	Implemented by default
OL08-00-020280	CAT III	-	All OL 8 passwords must contain at least one special character.	Implemented by default
OL08-00-030741	CAT III	-	OL 8 must disable the <code>chrony</code> daemon from acting as a server.	Implemented by default
OL08-00-030742	CAT III	-	OL 8 must disable network management of the <code>chrony</code> daemon.	Implemented by default
OL08-00-040024	CAT III	-	OL 8 must disable the transparent inter-process communication (TIPC) protocol.	Implemented by default
OL08-00-040025	CAT III	-	OL 8 must disable mounting of <code>cramfs</code> .	Implemented by default
OL08-00-040026	CAT III	-	OL 8 must disable IEEE 1394 (FireWire) Support.	Implemented by default

E

Managing Oracle Key Vault Platform Certificates

This chapter describes how to manage Oracle Key Vault Platform certificates by monitoring and rotating platform certificates before they expire.

- [Overview of Oracle Key Vault Platform Certificates](#)
Oracle Key Vault platform certificates are used when adding a new node to an Oracle Key Vault multi-master cluster, or when adding systems to a primary-standby deployment. Platform certificates are also used when shipping redo between read/write nodes in the cluster.
- [Monitoring Oracle Key Vault Platform Certificate Expiration](#)
You can proactively set alerts and monitor the expiration dates of the Oracle Key Vault platform certificates and rotate them before they expire.
- [Rotating Platform Certificates](#)
You must rotate the platform certificates by logging in to the Oracle Key Vault system and running a series of commands.

Overview of Oracle Key Vault Platform Certificates

Oracle Key Vault platform certificates are used when adding a new node to an Oracle Key Vault multi-master cluster, or when adding systems to a primary-standby deployment. Platform certificates are also used when shipping redo between read/write nodes in the cluster.

They are also used when shipping redo between read/write nodes in the cluster. These certificates are different from the Oracle Key Vault service certificates and have different expiration dates. They are also managed using a different process than Oracle Key Vault service certificates. If you do not rotate the Oracle Key Vault platform certificates before they expire, you cannot add a new node to the Oracle Key Vault multi-master cluster. The redo shipping between Oracle Key Vault read/write nodes may also be impacted, causing each node of the read/write pair to go into read-only restricted mode. You cannot upgrade an Oracle Key Vault system with expired platform certificates. You must rotate the Oracle Key Vault platform certificates before they expire.

Rotating the Oracle Key Vault platform certificates does not rotate the Oracle Key Vault service certificates and does not impact endpoint communication with Oracle Key Vault. Similarly, rotating Oracle Key Vault service certificates does not rotate the platform certificates.

Monitoring Oracle Key Vault Platform Certificate Expiration

You can proactively set alerts and monitor the expiration dates of the Oracle Key Vault platform certificates and rotate them before they expire.

- [Finding the Expiration Date of Platform Certificates](#)
You can find the expiration date of the platform certificates on the Oracle Key Vault management console.

- [Monitoring Platform Certificates Expiration Using Platform Certificate Expiration Alerts](#)
You can set expiration alerts as reminders to rotate the platform certificates before they expire.

Finding the Expiration Date of Platform Certificates

You can find the expiration date of the platform certificates on the Oracle Key Vault management console.

1. Log in to the Oracle Key Vault management console as a System Administrator. In a multi-master cluster environment, you can log in to any node of the cluster.
2. Select the **System** tab, then select **Status** from the navigation side bar.
3. Check the **Platform Certificates Expiration Date** field to determine when the platform certificates will expire.
4. Check the **Platform Certificates Expiring In** field to determine how many days are left for the certificates to expire.

Oracle Key Vault raises an alert for the platform certificates expiration when the platform certificates expiration date falls within the alert threshold period.

Note:

The expiration date shown is the minimum of the expiration dates of the platform certificate used when adding new multi-master cluster nodes or configuring a primary-standby deployment and the one that is used for redo shipping.

Monitoring Platform Certificates Expiration Using Platform Certificate Expiration Alerts

You can set expiration alerts as reminders to rotate the platform certificates before they expire.

Expiration of the platform certificates could result in redo shipping failures between read/write nodes in a multi-master cluster or between the primary and standby of a primary-standby deployment, resulting in the systems going into read-only restricted mode. It may also prevent the addition of a new node to the Oracle Key Vault cluster and prevent upgrade of the Oracle Key Vault system. Ensure that you rotate the Oracle Key Vault platform certificates before their expiration date. To avoid this scenario, Oracle recommends that you configure the Platform Certificate Expiration alert as a reminder to rotate the Oracle Key Vault platform certificates before they expire. This alert is separate from those monitoring expiration of the Oracle Key Vault service certificates (CA, server/node, and endpoint certificates).

Rotating Platform Certificates

You must rotate the platform certificates by logging in to the Oracle Key Vault system and running a series of commands.

In the case of a multi-master cluster, different steps may need to be run on different nodes of the cluster. In the case of a primary-standby environment, the steps may need to be run on both primary and standby.

- [Rotating Platform Certificates on a Standalone Oracle Key Vault Server](#)
Rotate the platform certificates on a standalone Oracle Key Vault server to replace the existing certificates near their expiration date with the new ones.
- [Rotating Platform Certificates in a Multi-Master Cluster Environment](#)
Learn how to rotate platform certificates in a multi-master cluster environment.

Rotating Platform Certificates on a Standalone Oracle Key Vault Server

Rotate the platform certificates on a standalone Oracle Key Vault server to replace the existing certificates near their expiration date with the new ones.

1. SSH into the Oracle Key Vault system as the support user.

```
ssh support@OKV_SERVER_IP_ADDRESS
```

2. Switch to user root:

```
su - root
```

3. Check the validity of the platform certificates that are used when adding new nodes to a multi-master cluster or configuring a primary-standby deployment as follows:

```
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/ca.crt  
openssl x509 -noout -enddate -in /etc/pki/tls/certs/localhost_internal.crt  
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/cert.crt  
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/avs/  
avs_apex_client.crt  
openssl x509 -noout -enddate -in /etc/pki/tls/certs/localhost.crt
```

If the dates shown indicate that any of the certificates have expired or will soon expire (as indicated by the platform certificate alerts), you must rotate the certificate by performing the following steps:

- For Oracle Key Vault systems at release 21.1, 21.2, 21.3, 21.4, and 21.5, download the **Oracle Audit Vault and Database Firewall patch** for bug **34378212**. SCP the downloaded zip file to the Oracle Key Vault server and unzip the file:

```
scp <downloaded_patch_file> support@OKV_SERVER_IP_ADDRESS:/tmp  
ssh support@OKV_SERVER_IP_ADDRESS  
su - root  
cd /tmp  
unzip <zip_file_for_patch_34378212> -d /tmp  
mkdir /root/gensslcert  
cp /tmp/gensslcert.avs.tar.gz /root/gensslcert  
cd /root/gensslcert  
tar xvfz gensslcert.avs.tar.gz  
export GENSSLCERT_HOME="/root/gensslcert"  
echo $GENSSLCERT_HOME
```

- If the Oracle Key Vault system is at release 21.6 or higher, then run the following command:

```
export GENSSLCERT_HOME="/usr/local/bin/"  
echo $GENSSLCERT_HOME
```

- Run the following command to regenerate the platform certificate used when adding new nodes to a multi-master cluster or configuring a primary-standby deployment:

```
"$GENSSLCERT_HOME"/gensslcert destroy-certs create-ca
```

4. Restart the necessary services by running the following commands:

```
systemctl reload httpd
systemctl restart controller
```

5. Update the platform CA certificate bundle, using the following command:

```
cat /usr/local/dbfw/etc/ha_partner.crt /usr/local/dbfw/etc/ca.crt
> /etc/pki/tls/certs/dbfw-ca.crt
systemctl reload httpd
systemctl restart controller
systemctl start monitor
```

6. Check the validity of the platform certificate used for redo shipping by performing the following steps. It is sufficient to check the validity on one multi-master cluster node only.

```
su oracle
cd /var/lib/oracle/dbfw/network/admin/avwallet
orapki wallet display -wallet /var/lib/oracle/dbfw/network/admin/avwallet
-complete
orapki wallet export -wallet . -dn "DC=com,CN=avserver,OU=db,O=oracle -
cert /tmp/avserver_check_cert_validity.cert
openssl x509 -in /tmp/avserver_check_cert_validity.cert -subject -enddate -
noout
```

If the certificate has expired or is expiring soon, then you must rotate it using the steps starting with step 8.

7. Switch back to the `root` user using the command:

```
exit
```

8. For Oracle Key Vault releases 21.1 to 21.4, contact Oracle Support for help with rotating the platform redo shipping certificate. For Oracle Key Vault release 21.5 and later, run the following commands to regenerate the platform redo shipping certificate:

```
cd /opt/avdf/lib/ruby/avdf
ruby update_agent_cert_task.rb
```

9. Run the following commands to verify the validity of the regenerated certificates:

```
openssl x509 -startdate -enddate -noout -in /usr/local/dbfw/etc/ca.crt
openssl x509 -startdate -enddate -noout -in /etc/pki/tls/certs/
localhost_internal.crt
openssl x509 -startdate -enddate -noout -in /usr/local/dbfw/etc/cert.crt
openssl x509 -startdate -enddate -noout -in /usr/local/dbfw/etc/avs/
avs_apex_client.crt
openssl x509 -startdate -enddate -noout -in /etc/pki/tls/certs/
localhost.crt
```

10. Switch to user `oracle` to verify the validity of the redo shipping certificate, using the following commands:

```
su oracle
cd /var/lib/oracle/dbfw/network/admin/avwallet
orapki wallet display -wallet /var/lib/oracle/dbfw/network/admin/avwallet
-complete
orapki wallet export -wallet . -dn "DC=com,CN=avserver,OU=db,O=oracle -
cert /tmp/avserver_check_cert_validity.cert
openssl x509 -in /tmp/avserver_check_cert_validity.cert -subject -
startdate -enddate -noout
```

11. Switch back to user `root` using the following command:

```
exit
```

12. Restart the database service on the system using the following command:

```
systemctl restart dbfwdb
```

13. Verify the validity of the platform certificates from the Oracle Key Vault management console.

Rotating Platform Certificates in a Multi-Master Cluster Environment

Learn how to rotate platform certificates in a multi-master cluster environment.

- [Rotate Platform CA Certificate on Read/Write Multi-Master Cluster Nodes](#)
Rotate the platform certificates on read/write multi-master cluster nodes to replace the existing certificates near their expiration date with the new ones.
- [Rotate Platform CA Certificate on Read-Only Multi-Master Cluster Nodes](#)
Learn how to rotate platform certificates on each read-only Oracle Key Vault multi-master cluster node using the steps described in this topic.
- [Rotate Platform Certificate Used For Redo Shipping On Any One Multi-Master Cluster Node](#)
In a multi-master cluster environment, if the platform certificate for redo shipping is expiring or has expired, you must rotate it on one node of the multi-master cluster, and then transfer the certificate to all of the other multi-master cluster nodes.
- [Transfer the Rotated Redo Shipping Platform Certificate to Other Multi-Master Cluster Nodes](#)
Learn how to transfer the redo shipping platform certificate to other multi-master cluster node after rotating it on one multi-master cluster node.

Rotate Platform CA Certificate on Read/Write Multi-Master Cluster Nodes

Rotate the platform certificates on read/write multi-master cluster nodes to replace the existing certificates near their expiration date with the new ones.

In this section, Node A and Node B refer to the two nodes of a given read/write pair. Implement these steps on each set of read/write pairs in turn.

1. (Node A) SSH into Node A as user support:

```
ssh support@OKV_NODEA_IP
```

2. (Node A) Switch to user root on Node A :

```
su - root
```

3. Check the validity of the platform certificates that are used when adding new nodes to a multi-master cluster or configuring a primary-standby deployment using the following commands:

```
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/ca.crt
openssl x509 -noout -enddate -in /etc/pki/tls/certs/localhost_internal.crt
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/cert.crt
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/avs/
avs_apex_client.crt
openssl x509 -noout -enddate -in /etc/pki/tls/certs/localhost.crt
```

If the dates indicate that any of the certificates have expired or will soon expire (as indicated by the platform certificate alerts), you must rotate the certificates using the following steps:

- (Node A)
 - For Oracle Key Vault systems at release 21.1, 21.2, 21.3, 21.4, and 21.5, download the **Oracle Audit Vault and Database Firewall patch** for bug **34378212**. SCP the downloaded zip file to the Oracle Key Vault system and unzip the file using the following commands:

```
scp <zip_file_for_patch_34378212> support@OKV_NODEA_IP:/tmp
ssh support@OKV_NODEA_IP
su - root
cd /tmp
unzip <zip_file_for_patch_34378212> -d /tmp
mkdir /root/gensslcert
cp /tmp/gensslcert.avs.tar.gz /root/gensslcert
cd /root/gensslcert
tar xvfz gensslcert.avs.tar.gz
export GENSSLCERT_HOME=/root/gensslcert
echo $GENSSLCERT_HOME
```

- If the Oracle Key Vault system is at release 21.6 or higher, then run the following command:

```
export GENSSLCERT_HOME="/usr/local/bin/"
echo $GENSSLCERT_HOME
```

- Run the following command on Node A (at release 21.1 or higher) to regenerate the platform certificate used when adding new nodes to a multi-master cluster or configuring a primary-standby deployment:

```
"$GENSSLCERT_HOME"/gensslcert destroy-certs create-ca
systemctl reload httpd
systemctl restart controller
```

4. (Node A) Transfer a copy of the regenerated certificate to Node B using SCP:

```
scp /usr/local/dbfw/etc/ca.crt support@OKV_NODEB_IP:/tmp/ha_partner.crt
```

5. (Node B)

- For Oracle Key Vault systems at release 21.1, 21.2, 21.3, 21.4, and 21.5, download the **Oracle Audit Vault and Database Firewall patch** for bug **34378212**. SCP the downloaded zip file to the Oracle Key Vault system and unzip the file using the following commands:

```
scp <zip_file_for_patch_34378212> support@OKV_NODEB_IP:/tmp
ssh support@OKV_NODEB_IP
su - root
cd /tmp
unzip <zip_file_for_patch_34378212> -d /tmp
mkdir /root/gensslcert
cp /tmp/gensslcert.avs.tar.gz /root/gensslcert
cd /root/gensslcert
tar xvfz gensslcert.avs.tar.gz
export GENSSLCERT_HOME=/root/gensslcert
echo $GENSSLCERT_HOME
```

- If the Oracle Key Vault system is at release 21.6 or higher, then run the following command:

```
export GENSSLCERT_HOME="/usr/local/bin/"
echo $GENSSLCERT_HOME
```

6. (Node B) Run the following commands on Node B:

```
cp /tmp/ha_partner.crt /usr/local/dbfw/etc/ha_partner.crt
systemctl reload httpd
systemctl restart controller
"$GENSSLCERT_HOME"/gensslcert destroy-certs create-ca
systemctl reload httpd
systemctl restart controller
```

7. (Node B) Transfer a copy of the regenerated certificate to Node A using SCP:

```
scp /usr/local/dbfw/etc/ca.crt support@OKV_NODEA_IP:/tmp/ha_partner.crt
```

8. (Node A) SSH into Node A as the support user, then switch to user root:

```
ssh support@OKV_NODEA_IP
su - root
```

9. (Node A) Run the following commands on Node A:

```
cp /tmp/ha_partner.crt /usr/local/dbfw/etc/ha_partner.crt
systemctl reload httpd
systemctl restart controller
cat /usr/local/dbfw/etc/ha_partner.crt /usr/local/dbfw/etc/ca.crt
> /etc/pki/tls/certs/dbfw-ca.crt
systemctl reload httpd
```

```
systemctl restart controller
systemctl restart monitor
```

10. (Node B) SSH into Node B as the `support` user, then switch to user `root`:

```
ssh support@OKV_NODEB_IP
su - root
```

11. (Node B) Run the following commands on Node B:

```
cat /usr/local/dbfw/etc/ha_partner.crt /usr/local/dbfw/etc/ca.crt
> /etc/pki/tls/certs/dbfw-ca.crt
systemctl reload httpd
systemctl restart controller
Systemctl restart monitor
```

Repeat steps 1 to 11 on each pair of read/write nodes in the multi-master cluster.

Rotate Platform CA Certificate on Read-Only Multi-Master Cluster Nodes

Learn how to rotate platform certificates on each read-only Oracle Key Vault multi-master cluster node using the steps described in this topic.

1. SSH in to the node as user `support`.

```
ssh support@OKV_NODE_IP
```

2. Switch to user `root`.

```
su - root
```

3. Check the validity of the platform certificates that are used when adding new nodes to a multi-master cluster or configuring a primary-standby deployment using the following commands:

```
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/ca.crt
openssl x509 -noout -enddate -in /etc/pki/tls/certs/localhost_internal.crt
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/cert.crt
openssl x509 -noout -enddate -in /usr/local/dbfw/etc/avs/
avs_apex_client.crt
openssl x509 -noout -enddate -in /etc/pki/tls/certs/localhost.crt
```

4. For Oracle Key Vault release 21.1 to release 21.5, download the **Oracle Audit Vault and Database Firewall patch** for bug **34378212**. SCP the downloaded zip file to the Oracle Key Vault server and unzip the file using the following commands:

```
scp <zip_file_for_patch_34378212> support@OKV_NODE_IP_ADDRESS:/tmp
ssh support@OKV_NODE_IP_ADDRESS
su - root
cd /tmp
unzip <zip_file_for_patch_34378212> -d /tmp
mkdir /root/gensslcert
cp /tmp/gensslcert.avs.tar.gz /root/gensslcert
```

```
cd /root/gensslcert
tar xvfz gensslcert.avs.tar.gz
```

```
export GENSSLCERT_HOME="/root/gensslcert"
echo $GENSSLCERT_HOME
```

- If the Oracle Key Vault system is at release 21.6 or later, then run the following commands:

```
export GENSSLCERT_HOME="/usr/local/bin/"
echo $GENSSLCERT_HOME
```

5. Run the following command to regenerate the platform certificate used when adding new nodes to a multi-master cluster or configuring a primary-standby deployment:

```
"$GENSSLCERT_HOME"/gensslcert destroy-certs create-ca
```

6. Reload and restart the `httpd` and `controller` services:

```
systemctl reload httpd
systemctl restart controller
```

7. Update the platform CA certificate bundle:

```
cat /usr/local/dbfw/etc/ha_partner.crt /usr/local/dbfw/etc/ca.crt
> /etc/pki/tls/certs/dbfw-ca.crt
systemctl reload httpd
systemctl restart controller
systemctl start monitor
```

Repeat these steps on each read-only node in the multi-master cluster.

Rotate Platform Certificate Used For Redo Shipping On Any One Multi-Master Cluster Node

In a multi-master cluster environment, if the platform certificate for redo shipping is expiring or has expired, you must rotate it on one node of the multi-master cluster, and then transfer the certificate to all of the other multi-master cluster nodes.

Follow these steps to check the expiration date of the redo shipping platform certificate and to rotate the certificate if required.

1. Choose one node of the multi-master cluster on which to rotate the redo shipping platform certificate.
2. SSH into the Oracle Key Vault node as the `support` user.

```
ssh support@OKV_NODE_IP_ADDRESS
```

3. Switch to user `root`.

```
su - root
```

4. Check the validity of the platform certificate used for redo shipping using the following commands. It is sufficient to check this on one multi-master cluster node only.

```
su oracle
cd /var/lib/oracle/dbfw/network/admin/avwallet
orapki wallet display -wallet /var/lib/oracle/dbfw/network/admin/avwallet
-complete
orapki wallet export -wallet . -dn "DC=com,CN=avserver,OU=db,O=oracle" -
cert /tmp/avserver_check_cert_validity.cert
openssl x509 -in /tmp/avserver_check_cert_validity.cert -subject -enddate -
noout
```

If the certificate has expired or will soon expire, then you must rotate it using the subsequent steps.

5. Switch back to the user `root` using the following command:

```
exit
```

6. Rotate the platform certificate used for redo shipping using the following commands, still as user `root`:

```
cd /opt/avdf/lib/ruby/avdf
ruby update_agent_cert_task.rb
```

7. Switch to user `oracle` to verify the validity of the certificate, using the following commands:

```
su oracle
cd /var/lib/oracle/dbfw/network/admin/avwallet
orapki wallet display -wallet /var/lib/oracle/dbfw/network/admin/avwallet
-complete
orapki wallet export -wallet . -dn "DC=com,CN=avserver,OU=db,O=oracle" -
cert /tmp/avserver_check_cert_validity.cert
openssl x509 -in /tmp/avserver_check_cert_validity.cert -subject -
startdate -enddate -noout
```

8. For Oracle Key Vault releases 21.1 to 21.4, contact Oracle Support for help with rotating the platform redo shipping certificate. For Oracle Key Vault release 21.5 and later, run the following commands to regenerate the platform redo shipping certificate:

```
cd /opt/avdf/lib/ruby/avdf
ruby update_agent_cert_task.rb
```

9. Restart the database service on the system using the following command:

```
systemctl restart dbfwdb
```

10. Verify the validity of the platform certificates from the Oracle Key Vault management console.
11. Transfer the rotated redo shipping platform certificate to all other multi-master cluster nodes using the steps detailed in the section [Transfer the Rotated Redo Shipping Platform Certificate to Other Multi-Master Cluster Nodes](#).

Transfer the Rotated Redo Shipping Platform Certificate to Other Multi-Master Cluster Nodes

Learn how to transfer the redo shipping platform certificate to other multi-master cluster node after rotating it on one multi-master cluster node.

1. As the `support` user, SSH into the Oracle Key Vault node on which the redo shipping platform certificate was rotated.

```
ssh support@OKV_NODE_IP_ADDRESS
```

2. Switch to user `root`.

```
su - root
```

3. Create a bundle containing the rotated platform certificate using the following steps:

```
cd /var/lib/oracle/dbfw/network/admin/avwallet  
zip -r avwallet_regenerated.zip *
```

4. Use SCP to copy the bundle to all other multi-master cluster nodes.

```
scp avwallet_regenerated.zip  
support@<OKV_OTHER_CLUSTER_NODE_IP_ADDRESS>:/tmp
```

5. Repeat the following set of steps on each of the other multi-master cluster nodes:

- a. SSH (as the `support` user) in to the Oracle Key Vault node to which the regenerated redo shipping certificate was transferred.

```
ssh support@OKV_OTHER_CLUSTER_NODE_IP_ADDRESS
```

- b. Switch to user `root`.

```
su - root
```

- c. Unzip the `avwallet_regenerated` zip file that was copied using SCP using the following command:

```
cd /tmp  
unzip avwallet_regenerated.zip -d avwallet_regenerated  
chmod -R 775 avwallet_regenerated
```

- d. Take a backup of the old platform certificate using the following command:

```
cd /var/lib/oracle/dbfw/network/admin  
cp -Rp avwallet avwallet_expired
```

- e. Copy the new wallet, ensuring to preserve permissions :

```
cp /tmp/avwallet_regenerated/* avwallet/
```

- f. Restart services using the following commands:

```
service dbfwlistener restart
service dbfwdb restart
```

- g. Switch to user `oracle` to verify the validity of the certificate, using the following commands:

```
su oracle
cd /var/lib/oracle/dbfw/network/admin/avwallet
orapki wallet display -wallet /var/lib/oracle/dbfw/network/admin/
avwallet -complete
orapki wallet export -wallet . -dn "DC=com,CN=avserver,OU=db,O=oracle" -
cert /tmp/avserver_check_cert_validity.cert
openssl x509 -in /tmp/avserver_check_cert_validity.cert -subject -
startdate -enddate -noout
```

Verify the validity of the platform certificates from the Oracle Key Vault management console.

Glossary

appliance

The format in which Oracle Key Vault is made available. The Oracle Key Vault software appliance includes the operating system, the software that implements the Oracle Key Vault functionality, the database, the replication software, and other related components. Oracle Key Vault is delivered as a software image that is installed on a standalone computer, or machine, supplied by the user. Oracle provides all updates for the software on the appliance, including the operating system. Do not load additional software on the Oracle Key Vault appliance. You can deploy an Oracle Key Vault appliance as a standalone server, a member of a primary-standby configuration, or a node in a multi-master cluster.

Audit Manager

An Oracle Key Vault administrative role that enables a user to manage audit lifecycle and policies and to separate the role of auditing from the role of managing the Oracle Key Vault server.

auto-login wallet

An Oracle wallet file that can be accessed without a password. An auto-login wallet is stored in a `cwallet.sso` file.

candidate node

During [node induction](#), an Oracle Key Vault server to be added to a multi-master cluster. A candidate node must be a freshly installed Oracle Key Vault appliance, except when it is the [initial node](#), in which case it provides the entirety of the cluster's initial data. A candidate node must be at the same release and patch level as the multi-master cluster to which it is being added.

After the server has been inducted into a cluster, it is called a [node](#). After a successful node induction, you can configure the server to use the cluster-wide configuration settings. The [cluster data set](#) is then replicated to the node.

cluster data set

The set of all [security objects](#) managed by the cluster. When creating the cluster, the initial node provides all of the security objects that will be part of the initial cluster data set.

cluster link

A link that represents the outbound network connection (to the [node](#)) and the inbound replication process (from the node). You can enable or disable the link to manage node data replication.

cluster subgroup

A group of one or more [nodes](#) that is a subgroup of a cluster. Each node in a cluster can belong to only one subgroup. The node is assigned to a subgroup when the node is added to the multi-master cluster. The assignment is for each node, and members of a [read/write pair](#) can be in different subgroups.

The subgroup implements a notion of [endpoint](#) affinity. Endpoints are also a part of subgroups. The endpoint's subgroup is assigned when the endpoint is created. It is used when you set the endpoint's node search order in the [endpoint node scan list](#). Nodes in the same subgroup as the endpoint are considered local to the endpoint. The local subgroup is scanned first before communicating with nodes that are not in the local subgroup.

The cluster topology can change when you add or remove new nodes to and from the cluster. The endpoints get this information with the response messages for the operations the endpoint initiated. Oracle Key Vault periodically sends the updated endpoint node scan list back to the endpoint even if there is no change to cluster topology. This is to account for any lost messages.

controller node

A [node](#) that controls or manages a cluster reconfiguration change, such as adding, enabling, disabling, or removing nodes. A node is only a controller node while the change is being made. During [node induction](#), the controller node provides the server certificate and the data that is used to initialize the [candidate node](#).

Each concurrent operation will have its own controller node. One controller node can only control one cluster configuration transaction at a time.

credential file

A file that contains sensitive information such as user IDs, passwords, and keys. The file, such as a Kerberos keytab file, is stored as an opaque object, which means that its individual contents are not interpreted by Oracle Key Vault. The entire file is uploaded and downloaded as an object.

See also [security object](#).

default wallet

A special [virtual wallet](#) that is associated with an [endpoint](#), into which all the endpoint's [security objects](#) can be automatically uploaded.

deleted node

A [node](#) that has been disassociated from the cluster, either by using the **Delete** or **Force Delete** buttons on the Oracle Key Vault management console. If it has been disabled for longer than the [Maximum Disable Node duration](#), then you must delete the node.

Once a node has been deleted, you cannot re-associate it with the cluster. If it is to be inducted into the cluster, then you must re-image it and then convert into a freshly installed server.

You can use the **Delete** option under normal operating circumstances. Only use the **Force Delete** option if the node is unreachable when the **Delete** option does not work.

endpoint

A computer system such as a database server, an application server, and other information systems, where keys are used to access encrypted data and credentials are used to authenticate to other systems.

endpoint administrator

Owner of an [endpoint](#). Endpoint administrators can be typically system, security, or database administrators, but they can be any personnel charged with deploying, managing and maintaining security within an enterprise. They are responsible for enrolling endpoints and controlling endpoint access to [security objects](#).

endpoint group

A collection of [endpoints](#) that are created to share a set of [security objects](#).

endpoint node scan list

A list of [nodes](#) to which an [endpoint](#) can connect.

heartbeat lag

A monitored metric that determines the health of the multi-master cluster. This is an indication of the [node](#) and network health. It is the time since the current node received a heartbeat message from a given node. A heartbeat is sent out from each node every two minutes. Every heartbeat should be received on each other node shortly thereafter.

A higher heartbeat lag indicates that the user operations that require conflict resolution like creating a wallet will take longer. Heartbeat lags between any two nodes affect the operations cluster wide. If the heartbeat lag is high, ensure that the cluster services are active and that replication is active. Disable and then re-enable the links between the two nodes between which the heartbeat lag is significant.

initial node

The first, or initial, [node](#) of an [Oracle Key Vault Multi-Master Cluster](#). You create a multi-master cluster by converting a single Oracle Key Vault server to become the initial node. The Oracle Key Vault server can be a clean installed Oracle Key Vault server, or it can already be in

service with active data. A standalone server or a member of a primary-standby configuration can be converted to be the initial node of a cluster. If you want to use a member of a primary-standby configuration, then you must first break the primary-standby relationship splitting the pair.

If the initial node has been active and therefore has data, then Oracle Key Vault uses this data as the [cluster data set](#) to initialize the cluster.

Initialization can occur only once in the life of the cluster.

installation passphrase

A password that is specified during the Oracle Key Vault installation. The installation passphrase is used to log in to Oracle Key Vault and complete the post-installation tasks. The installation passphrase can only be changed on the Oracle Key Vault management console after installation but before post-installation. After you complete the post-installation process, this option no longer appears on the management console.

JAVA_HOME

The environment variable that points to the location of Java files (JDK/JRE) in the system. This allows Java applications to look up the `JAVA_HOME` variable in order to operate.

Java keystore file

A file that can hold multiple [security objects](#) such as keys and certificates. It uses the Java Keystore File (JKS) format.

Key Administrator

An Oracle Key Vault administrator role that enables a user to manage the key lifecycle and control access to all [security objects](#) within Oracle Key Vault. This is a highly sensitive role and should be granted with care.

keystore

A generalized term for a container that stores encryption keys including but not limited to TDE master encryption keys.

Management Information Base (MIB)

See [MIB](#).

master encryption key

See [TDE master encryption key](#).

maximum disable node duration

The time, in hours, that a node may remain in the disabled state. If the node has been disabled for a longer duration, it can no longer be enabled.

The default maximum disable node duration is 24 hours.

MIB

Management information base; a text file that, if Oracle Key Vault is monitored through SNMP, describes the variables that contain the information that SNMP can access. The variables described in a MIB, which are also called MIB objects, are the items that can be monitored using SNMP. There is one MIB for each element that is monitored.

name resolution time

A monitored metric used to determine the health of the multi-master cluster. It is the average time taken to ascertain that there is no name conflict in the cluster or to resolve the name conflict after an attempt to use conflicting names took place.

node

An Oracle Key Vault server that has been converted to be a member of an Oracle Key Vault multi-master cluster. It is known as an Oracle Key Vault cluster node or simply a node.

node induction

The process of converting an Oracle Key Vault server to be a node in the multi-master cluster.

The initial node in a cluster provides the initial [cluster data set](#). Subsequently, only new Oracle Key Vault servers can be inducted to the multi-master cluster, and the current data in the multi-master cluster is loaded into the new nodes.

OKV_HOME

The environment variable that points to the location in which the Oracle Key Vault [endpoint](#) software will reside. It contains sub-directories for endpoint software such as the configuration files, log files, libraries, binaries, and other files that the endpoint software utility needs.

online master encryption key

A TDE-generated master encryption key that is stored in Oracle Key Vault. The online master encryption key enables Oracle Key Vault administrators to have full control over the TDE master encryption keys that Key Vault protects. When a key rotation is performed on the online master encryption key, the change is reflected in all other [nodes](#) in a cluster. In previous releases, the term for online master encryption key was TDE direct connection.

Opaque Object

A [security object](#) that Oracle Key Vault cannot interpret.

Oracle Key Vault appliance

See [appliance](#).

Oracle Key Vault multi-master cluster

A distributed set of Oracle Key Vault [nodes](#) that are grouped together so that they all communicate with one another. Some pairs of nodes are configured as [read/write pairs](#). In a read-write pair, an update to one node is replicated to the other node, and the update must be verified on the other node before the update is considered successful.

All nodes in the multi-master cluster connect to all other nodes. Data updated in a read-write pair is replicated to all nodes.

Oracle Key Vault node

See [node](#).

Oracle Key Vault server

An Oracle Key Vault server that is a standalone installation of the Oracle Key Vault appliance. It provides all the core functionality related to endpoints and wallets.

Oracle wallet file

A container that can hold multiple [security objects](#) such as keys and certificates. It uses the PKCS#12 cryptographic standard.

You can manage Oracle wallets in Oracle Key Vault just like other security objects. Optionally, you can encrypt them and protect them with a password. An Oracle wallet that can be accessed without a password is called an [auto-login wallet](#).

See also [password-protected wallet](#).

ORACLE_BASE

The environment variable that points to the root of the Oracle Database directory tree. The Oracle Base directory is the top level directory that you can use to install the various Oracle software products. You can use the same Oracle base directory for multiple installations. For example, `/u01/app/oracle` is an Oracle base directory created by the `oracle` user.

ORACLE_HOME

The environment variable that points to the directory path to install Oracle components (for example, `/u01/app/oracle/product/18.3.0/db_n`). You are prompted to enter an Oracle home in the **Path** field of the Specify File Locations window.

`ORACLE_HOME` corresponds to the environment in which Oracle Database products run. If you install an OFA-compliant database, using Oracle Universal Installer defaults, then the Oracle home (known as `$ORACLE_HOME` in this guide) is located beneath `$ORACLE_BASE`. The default Oracle home is `db_n` where `n` is the Oracle home number. It contains subdirectories for Oracle Database software executable files and network files.

ORACLE_SID

The environment variable that represents the Oracle System ID (SID), which uniquely identifies a particular database on a system. For this reason, you cannot have more than one database with the same SID on a computer system.

When using Oracle Real Application Clusters, you must ensure that all instances that belong to the same database have a unique SID.

oraenv

Along with `coraenv`, a Unix/ Linux command line utility that sets the required environment variables (`ORACLE_SID`, `ORACLE_HOME` and `PATH`) to allow a user to connect to a given database instance. If these environment variables are not set, then commands such as `sqlplus`, `imp`, `exp` will not work (or not be found).

Use `coraenv` when using the C Shell and `oraenv` when using a Bourne, Korn, or Bash shell.

password-protected wallet

An encrypted Oracle wallet that has a user-defined password stored in an `ewallet.p12` file.

PKCS#11 library

A library that allows an Oracle TDE database to connect to Oracle Key Vault to manage the master encryption keys.

PKCS#12 file

In cryptography, PKCS#12 defines an archive file format for storing many cryptographic objects as a single file. Wallet files are stored in PKCS#12 format.

read-only node

A [node](#) that is not part of a replication pair. Most data cannot be directly updated using the Oracle Key Vault management console, or with Oracle Key Vault client software. Critical data such as keys, wallets, and certificates in a read-only node is only updated through replication from [read-write nodes](#).

read-only restricted mode

A [node](#) enters read-only restricted mode when it has no [read/write pair](#), or if its [read-write peer](#) is unavailable. The Oracle Key Vault console displays a warning that the node is operating in

read-only restricted mode. In read-only restricted mode, updates using the Oracle Key Vault management console, or Oracle Key Vault client software are restricted. However, you can still perform system configuration on the node.

When the node is a member of a [read-write pair](#), this indicates the other node has been disabled but not deleted from the cluster, or the heartbeat is not detected for other reasons.

read/write mode

A node is in read/write mode when it is available for endpoint and wallet data updates using the Oracle Key Vault management console, or Oracle Key Vault client software. The node must be a member of a [read/write pair](#), and the [read/write peer](#) must be online and active.

When both nodes in the pair are available, both nodes can accept updates, and all updates to one node are synchronously replicated to the peer. If one of the nodes in the pair becomes unavailable, then the remaining node enters [read-only restricted mode](#) and will not accept any data updates until the peer is restored.

The node state is displayed on the Monitoring page of the **Cluster** tab of the node management console. The **Cluster** tab of the node management console displays the type and status of all nodes in the cluster.

read/write node

An active, connected, member of a read/write pair of [nodes](#).

read/write pair

A pair of [nodes](#) that operates with bidirectional synchronous replication. You create the read/write pair by pairing a new node with a read-only node. You can update data, including the [endpoint](#) and wallet data, in either node by using the Oracle Key Vault management console, or Oracle Key Vault client software. The updates are replicated immediately to the other node in the pair. Updates are replicated asynchronously to all other nodes.

A node can be a member of at most one bidirectional synchronous pair.

A multi-master cluster requires at least one [read-write pair](#) to be fully operational. It can have a maximum of 8 read/write pairs.

read/write peer

The specific member of one, and only one, [read-write pair](#) in the cluster. Each read-write pair consists of only two [nodes](#). You configure nodes as peers by setting **Add Candidate Node as Read-Write Peer** to **Yes** on the [controller node](#) during induction of the [candidate node](#). Peers are identified on the Cluster Management Configuration page.

If one member of the pair is deleted, then the peer automatically becomes a [read-only node](#).

recovery passphrase

A secret token that is created during the installation of an Oracle Key Vault [appliance](#). The recovery passphrase created for the [initial node](#) is subsequently used by the cluster and propagated to all other [nodes](#) in the cluster.

You enter the existing recovery passphrase on both the controller page and the candidate page during induction of any nodes into the cluster. Because there is only one recovery passphrase, you must use that same recovery passphrase when the recovery passphrase is required.

replication

The process of replicating data changes that were made to a [read-write node](#) to all other [nodes](#). The [read-write peer](#) is updated immediately. Replication is used to distribute the data to all other nodes in the cluster.

replication lag

A monitored metric that determines the health of the multi-master cluster. It is the time taken for an object to be replicated to another [node](#).

A higher replication lag indicates that the Oracle Key Vault operations like changing the access permissions for an [endpoint](#) on the wallet will take longer to replicate. Depending on the operation, a replication lag may or may not have a cluster-wide impact. If the replication lag is significant between two nodes, then you should disable and re-enable the cluster links.

security object

An object that contains critical data provided by the [user](#). A security object can be of the following types:

- private encryption key
- Oracle wallet
- Java keystore
- Java Cryptography Extension keystore
- certificate
- credential file

software appliance

A self-contained preconfigured product that can be installed on supported hardware dedicated for a specific purpose.

sqlnet.ora

An Oracle Database configuration file for the client or server. By default, the `sqlnet.ora` file resides in `$ORACLE_HOME/network/admin` directory. It specifies the following connection information:

- Client domain to append to unqualified service names or net service names
- Order of naming methods for the client to use when resolving a name
- Logging and tracing features to use
- Route of connections
- External naming parameters
- Oracle Advanced Security parameters

System Administrator

An Oracle Key Vault administrator role that enables a user to create [users](#), [endpoints](#) and their respective groups, configure system settings and alerts, and generally administer Oracle Key Vault. This is a highly sensitive role and should be granted with care.

TDE master encryption key

A key that encrypts the data encryption keys for tables and tablespaces.

template

A collection of attributes for [security objects](#). When a security object is created using a template, then the attributes in the template are automatically assigned to the new object.

user

A staff member who uses Oracle Key Vault. Users can be administrators, auditors, or ordinary users with no administrative roles.

user group

A named collection of Oracle Key Vault [users](#). A user group can collectively be granted privileges or roles.

virtual wallet

A container for [security objects](#) such as public and private encryption keys, TDE master encryption keys, passwords, credentials, and certificates in Oracle Key Vault. The main purpose of a virtual wallet is to enable sharing of keys among [endpoints](#).

Index

A

- about managing, [9-24](#)
- access control, [2-4](#)
 - access grants for virtual wallets, [2-4](#)
 - how configuration works, [2-4](#)
- access control options, [2-4](#)
- access grants, [2-4](#)
- adding user to user group, [9-26](#)
- administration users
 - multi-masters clusters effect on, [9-6](#)
- administrative roles
 - about, [2-6](#)
 - about managing, [9-11](#)
 - Audit Manager
 - about, [2-8](#)
 - granting or changing, [9-11](#)
 - Key Administrator
 - about, [2-8](#)
 - revoking, [9-15](#)
 - separation of duty, [2-5](#)
 - System Administrator
 - about, [2-7](#)
- administrators
 - recovering credentials, [18-33](#)
- alerts, [1-12](#)
 - about, [22-25](#)
 - configuring, [22-40](#)
 - types of, [22-25](#)
 - viewing open alerts, [22-43](#)
- architecture, [3-3](#)
- archiving
 - credential files, [17-1](#)
- Audit Manager
 - about, [2-8](#)
- Audit Manager role
 - multi-master cluster effect on, [9-6](#)
- audit trail contents, [22-47](#)
- auditing
 - about, [22-47](#)
 - audit trail contents, [22-47](#)
 - deleting audit records, [9-3](#), [22-58](#)
 - exporting audit records to file, [9-3](#), [22-58](#)
 - setting for cluster, [18-28](#)
 - setting for node, [18-23](#)
 - syslog file, [22-48](#)

- auditing (*continued*)
 - viewing audit records, [22-57](#)
- Automatic Storage Management
 - uploading keystores
 - about, [14-9](#)
 - copying keystore to, [14-10](#)
 - procedure, [14-9](#)

B

- backing up data
 - about, [21-1](#)
 - best practices, [21-26](#)
 - changing schedule, [21-11](#)
 - deleting schedule, [21-11](#)
 - finding information about, [21-21](#)
 - multi-master cluster environment, [21-12](#)
 - primary-standby deployment, [21-11](#)
 - recovery passphrase, [21-12](#)
 - removing local backups, [21-22](#)
 - removing old backups, about, [21-17](#)
 - removing old backups, adding policy to existing destination, [21-19](#)
 - removing old backups, changing schedule, [21-19](#)
 - removing old backups, creating schedule, [21-18](#)
 - removing old backups, deleting schedule, [21-21](#)
 - resuming backup destination policy, [21-20](#)
 - scheduling backup, [21-9](#)
 - suspending backup destination policy, [21-20](#)
- backup destinations
 - about, [21-2](#)
 - changing settings, [21-6](#)
 - creating remote, [21-4](#)
 - deleting remote, [21-7](#)
- Backup Restore Failed, [C-30](#)
- backup scheduling, [21-9](#)
 - about, [21-8](#)
 - types, [21-8](#)
- backups
 - console certificates, [20-4](#)
 - Oracle ZFS Storage Appliance, [21-22](#), [21-23](#)
 - protecting with recovery passphrase, [21-12](#)
 - reports, [22-79](#)

- backups (*continued*)
 - types, [21-8](#)
- benefits
 - centralizing key lifecycle management, [1-2](#)
 - centralizing key storage, [1-2](#)
 - fighting security threats, [1-2](#)
- C**

- C SDK, [1-17](#)
- candidate nodes, [3-10](#)
- centralized storage
 - Java keystores, [1-6](#)
 - Oracle wallet files, [1-6](#)
- centralized storage and management of security objects, [1-10](#)
- centrally managed passwords, [17-9](#)
- certificates
 - rotation, checking overall status, [19-27](#)
 - rotation, checking status for endpoints, [19-28](#)
 - setting batch size for endpoint rotations, [19-25](#)
 - setting rotation sequence for multimaster cluster nodes, [19-26](#)
 - setting validity of self-signed CA certificates, [19-15](#)
- Certificate File Failure, [C-19](#)
- certificates, [20-1](#)
 - post-rotation tasks, [19-28](#)
 - See also console certificates
- changepwd command (okvutil), [B-3](#)
- changing a user group description, [9-27](#)
- Classic mode network interface
 - checking status of, [18-4](#)
- cluster node types
 - about, [3-6](#)
- cluster nodes
 - about, [3-4](#)
 - limitation, [3-4](#)
- cluster size and availability guidance, [3-13](#)
- cluster subgroups
 - about, [3-5](#)
- clusters
 - creating first node, [4-2](#)
 - deleting a node, [4-10](#)
 - disabling a node, [4-9](#)
 - disabling node replication, [4-12](#)
 - enabling a node, [4-10](#)
 - enabling node replication, [4-12](#)
 - force deleting a node, [4-11](#)
 - management information, [3-19](#), [4-13](#)
 - monitoring information, [4-14](#)
 - read-only, creating, [4-6](#)
 - read-write pair of nodes, creating, [4-4](#)
 - read-write pairs of nodes, creating, [4-8](#)
 - restarting cluster services, [4-12](#)
- clusters (*continued*)
 - setting up, about, [4-2](#)
 - terminating node pairing, [4-8](#)
- Commercial National Security Algorithm (CNSA)
 - about, [18-37](#)
 - backup and restore operations, [18-39](#)
 - running scripts, [18-38](#)
 - upgrading primary-standby Oracle Key Vault servers, [18-41](#)
 - upgrading standalone Oracle Key Vault server, [18-39](#)
- configuration files
 - endpoint configuration file, [13-11](#)
- configuration parameters
 - endpoints, [12-21](#)
 - setting globally for endpoints, [12-22](#)
 - setting per endpoint, [12-25](#)
- configuring a primary-standby deployment, [23-2](#)
- conflicts in names of objects, [4-16](#)
- console certificates, [20-1](#)
 - about managing, [20-1](#)
 - backup data restored, [20-4](#)
 - downloading CA request, [20-1](#)
 - having signed, [20-3](#)
 - primary-standby environments, [20-4](#)
 - RESTful services, [20-4](#)
 - uploading, [20-3](#)
- controller nodes
 - about, [3-9](#)
- Could Not Store Private Key Errors, [C-18](#)
- Create Endpoint Group privilege
 - endpoint privileges
 - separation of duty, [2-5](#)
 - granting or changing, [9-13](#)
 - separation of duty, [2-5](#)
- Create Endpoint privilege
 - about, [2-9](#)
 - granting or changing, [9-12](#)
 - separation of duty, [2-5](#)
- creating a user group, [9-24](#)
- creating user accounts, [9-7](#)
- credential files
 - about archiving and downloading, [17-1](#)
 - change to content guidance, [17-3](#)
 - downloading
 - guidance, [17-3](#)
 - procedure, [17-3](#)
 - overwriting danger of, [17-3](#)
 - sharing with multiple endpoints guidance, [17-3](#)
 - uploading
 - guidance, [17-3](#)
 - procedure, [17-2](#)
- credentials
 - guidance for SSH, [17-4](#)
- critical data, [3-5](#)

D

dashboard

- status panes, [18-2](#)
- viewing, [18-2](#)

data

- backing up, about, [21-1](#)
- backup removal schedule, adding to existing destination, [21-19](#)
- backup removal schedule, changing, [21-19](#)
- backup removal schedule, creating, [21-18](#)
- backup removal schedule, deleting, [21-21](#)
- backup removal schedule, finding information about, [21-21](#)
- backup removal schedule, resuming, [21-20](#)
- backup removal schedule, suspending, [21-20](#)
- backup removal, about, [21-17](#)
- backups, deleting local, [21-22](#)
- restoring, about, [21-1](#)

Database as a Service

- about configuring for Key Vault, [6-2](#)
- configuring instance, [6-2](#)
- creating low privileged user, [6-3](#)
- deleting SSH tunnel, [6-11](#)
- disabling SSH tunnel, [6-10](#)
- enrolling instance as endpoint
 - about, [6-13](#)
 - installing Oracle Key Vault software onto, [6-15](#)
 - post-installation tasks, [6-17](#)
 - preparing environment, [6-15](#)
 - registering, [6-13](#)
- resuming access to Oracle Key Vault, [6-21](#)
- reverse SSH tunnel in multi-master cluster, [6-8](#)
- reverse SSH tunnel in primary-standby configuration, [6-9](#)
- SSH tunnel between Oracle Key Vault and DBaaS instance, [6-5](#)
- SSH tunnel not active, [6-11](#)
- suspending access to Oracle Key Vault
 - about, [6-19](#)
 - procedure, [6-20](#)
- users
 - low privileged user for DBaaS, [6-3](#)
 - viewing SSH tunnel details, [6-9](#)

deleting user accounts, [9-10](#)

deleting user groups, [9-28](#)

deployment

- architecture, [2-2](#)
- overview, [1-17](#)

deployment scenarios

- cluster size and availability, [3-13](#)
- mid-size cluster, [3-15](#)
- two data centers, [3-15](#)
- two nodes, [3-14](#)

deployments

- credential files, archiving and downloading, [17-1](#)
 - Java keystores, uploading and downloading, [11-20](#)
 - JKS and JCEKS keystores, archiving and downloading, [11-23](#)
 - migrating standalone Key Vault server to multi-master cluster, [3-12](#)
 - online master encryption keys for TDE wallets, [1-4](#)
 - Oracle wallets, uploading and downloading, [11-20](#)
 - primary-standby to multi-master cluster, [3-12](#)
 - recommendations for, [4-17](#)
- ### diagnostic reports, [22-16](#), [22-22](#)
- about, [22-16](#), [22-22](#)
 - configuring generation utility, [22-17](#)
 - generating file, [22-19](#)

diagnostics

- accessing with okvutil diagnostics, [B-3](#)

DNS

- configuring NTP servers for non-multi-master clusters, [18-9](#)
- nodes, [18-19](#)

DNS settings

- multi-master clusters, [18-27](#)

download command (okvutil), [B-4](#)

Download Diagnostics, [C-4](#)

downloading

- credential files, [17-3](#)
- JKS and JCEKS keystores, [11-23](#), [11-25](#)
- wallets, [11-22](#)

downtime, minimizing, [18-42](#)

Dual NIC mode network interface

- checking status of, [18-4](#)

dual NIC network mode

- changing for nodes, [18-21](#)
- changing for standalone environment, [18-12](#)

E

effective group membership, LDAP users, [8-1](#)

email addresses

- changing, [9-22](#)
- disabling email notifications, [9-23](#)

email notification

- about, [22-11](#)
- configuring, [22-12](#)
- disabling, [22-15](#)
- testing, [22-14](#)

emergency system recovery, [2-15](#)

endpoint, [12-17](#)

endpoint administrators

- about, [2-16](#)

- endpoint groups, [12-32](#)
 - access grant to virtual wallet, [12-35](#)
 - adding endpoint too, [12-36](#)
 - creating, [12-32](#)
 - deleting, [12-38](#)
 - modifying details, [12-34](#)
 - modifying virtual wallets from Keys & Wallets tab, [10-9](#)
 - multi-master clusters, effect on, [12-32](#)
 - naming guidelines, [2-15](#)
 - removing access to virtual wallets from Keys & Wallets tab, [10-8](#)
 - removing endpoint, [12-37](#)
- endpoint node scan lists
 - about, [3-19](#)
- endpoint privileges
 - about managing, [9-11](#)
- EndPoint Related Issues, [C-2](#)
- endpoint self-enrollment, about, [12-4](#)
- endpoints, [12-32](#)
 - about, [12-12](#)
 - about managing, [12-1](#)
 - adding access to virtual wallet, [12-29](#)
 - adding to an endpoint group, [12-36](#)
 - adding using administrator-initiated enrollment, [12-6](#)
 - adding using self-enrollment, [12-10](#)
 - adding using self-enrollment, about, [12-10](#)
 - adding using self-enrollment, procedure for, [12-10](#)
 - administrators for, [12-1](#)
 - alternative for individual, [12-13](#)
 - associating default wallet with, [12-27](#)
 - configuration file, [13-11](#)
 - configuration parameters, about, [12-21](#)
 - DBaaS
 - enrolling, [6-13](#)
 - registering, [6-13](#)
 - default wallet, setting for, [12-27](#)
 - deleting, [12-12](#), [12-13](#)
 - details
 - about, [12-18](#)
 - modifying, [12-19](#)
 - diagnostics, [B-3](#), [B-12](#), [B-14](#)
 - downloading software, [13-3](#)
 - endpoint node scan lists, [3-19](#)
 - enrolling and provisioning, [13-3](#)
 - enrollment
 - about, [13-1](#)
 - administrator initiated, about, [12-4](#)
 - types of enrollment, [12-4](#)
 - enrollment in multi-master cluster, [12-5](#)
 - enrollment process
 - about, [13-1](#)
 - enrollment types, [12-4](#)
- endpoints (*continued*)
 - guidance on enrolling across deployments, [3-15](#)
 - installing software for new enrollment, [13-5](#)
 - Java home, how determined, [13-9](#)
 - limitations of TDE endpoint integration, [11-12](#)
 - modifying virtual wallets from Keys & Wallets tab, [10-9](#)
 - multi-master clusters, effect on, [12-3](#)
 - naming guidelines, [2-15](#)
 - nodes available for connection, [3-19](#)
 - not using Oracle Key Vault client software, [13-10](#)
 - okvclient.ora file, [13-11](#)
 - okvutil utility for provisioning, [B-1](#)
 - one or more endpoints, [12-12](#)
 - Oracle Cloud Infrastructure database instance
 - about, [6-1](#)
 - password, changing, [B-3](#)
 - post-installation for new enrollment, [13-7](#)
 - preparing environment for new enrollment, [13-4](#)
 - privileges for managing, [2-9](#)
 - provisioning
 - about, [13-1](#)
 - reenrolling, [12-14](#)
 - removing access to virtual wallets from Keys & Wallets tab, [10-8](#)
 - removing from an endpoint group, [12-37](#)
 - reports, [22-77](#)
 - revoking access to virtual wallet, [12-30](#)
 - rotation, [12-15](#), [19-35](#)
 - setting configuration parameters globally, [12-22](#)
 - setting configuration parameters per endpoint, [12-25](#)
 - setting extractable attribute value globally, [12-25](#)
 - setting extractable attribute value per endpoint, [12-26](#)
 - settings, about, [12-21](#)
 - suspending, [12-13](#)
 - TDE endpoint management, [13-11](#)
 - unmodifiable okvclient.ora parameters, [13-13](#)
 - upgrading for enrolled, [13-19](#)
 - upgrading for unenrolled
 - downloading Oracle Key Vault okvclient.jar software, [13-15](#)
 - installing Oracle Key Vault okvclient.jar software, [13-16](#)
 - post-installation tasks, [13-18](#)
 - preparing environment, [13-15](#)
 - upgrading from unenrolled endpoint, [13-14](#)
 - wallet items, viewing, [12-31](#)
 - See also* endpoint groups

enrolling endpoints
 about, [13-1](#)
 administrator initiated
 about, [12-4](#)
 self-initiated
 about, [12-4](#)

environment variables
 JAVA_HOME, how determined during client installation, [13-9](#)
 OKV_HOME
 non-database utilities, [13-10](#)
 set during installation, [13-9](#)
 okvclient.ora location of, [13-9](#)
 persistent master encryption key cache, [11-3](#)
 sqlnet.ora file, [13-10](#)

Error
 Object is Unstorable in Container error, [B-4](#)

EXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN parameter, [11-8](#)

external keystore password uploads
 about, [17-8](#)

external keystore password uploads to large deployments, [17-9](#)
 changing passwords, [17-13](#)
 example script for using passwords, [17-10](#)
 sharing secrets, [17-12](#)

extractable attribute
 setting globally, [12-25](#)
 setting per endpoint, [12-26](#)

extractable attribute value of symmetric and private keys
 about, [10-27](#)
 setting
 symmetric keys
 preventing existing symmetric keys from being extracted, [10-28](#)

F

failovers
 restoring primary-standby after, [23-10](#)

Fast-Start Failover (FSFO) Failure, [C-37](#)

FIPS 140–2, [2-16](#)

FIPS mode, [2-16](#)
 disabling, [18-20](#)
 disabling for non-multi-master clusters, [18-11](#)
 enabling, [18-20](#)
 enabling for non-multi-master clusters, [18-11](#)

FIPS-Inside
 See FIPS mode

G

granting access to objects or users, [2-4](#)

I

Inaccessible Oracle Key Vault, [C-34](#)

initial node
 creating, [4-2](#)
 creation of, [3-7](#)

Initial Triaging Steps, [C-26](#)

interfaces, [1-16](#)
 C client, [1-17](#)
 Java client, [1-17](#)
 management console, [1-16](#)
 okvutil endpoint utility, [1-17](#)
 RESTful services, [1-17](#)

Invalid certificate, [C-25](#)

J

Java keystores
 downloading, [B-4](#)
 uploading, [B-9](#)

Java SDK, [1-17](#)

JAVA_HOME environment variable
 how determined during client installation, [13-9](#)
 location determined during installation, [13-9](#)

JKS and JCEKS keystores
 downloading
 JKS and JCEKS keystores, [11-23](#)
 procedure, [11-25](#)
 uploading
 procedure, [11-24](#)

JKS and JCKS keystores
 change to content guidance, [11-25](#)
 downloading
 guidance, [11-25](#)
 overwriting danger of, [11-25](#)
 sharing with multiple endpoints guidance, [11-25](#)
 uploading
 guidance, [11-25](#)

K

Kerberos keytabs
 downloading, [B-4](#)

Key Administrator role
 about, [2-8](#)
 multi-master cluster effect on, [9-6](#)

key lifecycle management, [1-11](#)

key management reports for Oracle endpoints, [22-74](#)

key rotation, [1-4](#)

keys
 activating, [10-25](#)
 deactivating, [10-25](#)
 deleting, [10-26](#)
 finding for Key Vault, [B-7](#), [B-16](#)

keys (*continued*)

- multi-master clusters, effect on, [10-24](#)
- reports, [22-75](#)
- revoking, [10-26](#)
- state of, managing, [10-24](#)

keystores

- Automatic Storage Management
 - about uploading from, [14-9](#)
 - copying keystore to, [14-10](#)
 - procedure for uploading from, [14-9](#)

KMIP Protocol, [1-14](#)

L

LDAP configuration, [8-1](#)

- about, [8-1](#)
- about logging in as LDAP user, [8-8](#)
- creating the provider connection, [8-5](#)
- deleting, [8-11](#)
- disabling, [8-11](#)
- enabling, [8-9](#)
- LDAP directory server preparation tasks, [8-4](#)
- logging in as LDAP user, [8-8](#)
- mapping LDAP groups to Oracle Key Vault user groups, [8-6](#)
- modifying, [8-10](#)
- privilege grants for LDAP users, [8-3](#)
- testing, [8-11](#)

LDAP groups

- about, [8-12](#)
- creating group mappings, [8-13](#)
- deleting mapping, [8-17](#)
- modifying group mappings, [8-14](#)
- validating group mappings, [8-16](#)

LDAP users

- about, [8-17](#)
- about validation, [8-18](#)
- effective group membership, [8-1](#)
- finding information about, [8-18](#)
- modifying by regular users who have manage wallet privileges, [8-20](#)
- modifying using Key Administrator role, [8-20](#)
- modifying, about, [8-19](#)
- removing from Oracle Key Vault, [8-21](#)
- validating, [8-19](#)

list command (okvutil), [B-7](#)

list common errors(okvutil), [B-16](#)

local backup destinations

- about, [21-2](#)

logging in

- as LDAP user, [8-8](#)
- as LDAP user, about, [8-8](#)

M

Manage Endpoint Group privilege

- about, [2-11](#), [2-12](#)
- endpoint privileges
 - separation of duty, [2-5](#)
- granting or changing, [9-14](#)
- revoking, [9-15](#)
- separation of duty, [2-5](#)

Manage Endpoint privilege

- about, [2-10](#)
- granting or changing, [9-13](#)
- revoking, [9-15](#)
- separation of duty, [2-5](#)

management console

- about, [1-16](#)

Management Information Base (MIB) variables, [22-6](#)

master encryption keys

- persistent master encryption key cache, [1-12](#), [11-3](#)
- TDE,
 - See persistent master encryption key cache
 - user-defined key as, [11-26](#)

maximum disable node duration

- multi-master clusters, [18-27](#)

Microsoft Active Directory

- See LDAP configuration

monitoring

- diagnostic reports, [22-16](#), [22-22](#)
- email notification, [22-11](#)
- remote monitoring, [22-2](#)
- reports, [22-73](#)
- SNMP, [22-2](#)
- syslog configuration, [22-15](#)

monitoring information for clusters, [4-14](#)

multi-master cluster

- addition of second node, [7-8](#)

multi-master cluster configuration

- Oracle Audit Vault integration, [22-72](#)

multi-master clusters, [3-3](#)

- about managing, [4-2](#)
- addition of new server to cluster, [3-10](#)
- addition of nodes, [3-10](#)
- administration users, effect on, [9-6](#)
- Audit Manager role, affect on, [9-6](#)
- auditing for cluster
 - setting, [18-28](#)
- auditing for individual nodes
 - setting, [18-23](#)
- backup and restore operations, [21-1](#)
- benefits, [3-2](#)
- building and managing, about, [3-7](#)
- candidate node, [3-10](#)
- changing recovery passphrase, [18-34](#)
- cluster node, [3-4](#)

multi-master clusters (*continued*)

- cluster subgroups, [3-5](#)
- controller node, [3-9](#)
- critical data, [3-5](#)
- difference from primary-standby configuration, [23-4](#)
- DNS for individual nodes
 - configuring, [18-19](#)
- DNS settings, [18-27](#)
- downtime, minimizing, [18-42](#)
- effect on role management, [9-11](#)
- endpoint enrollment, [12-5](#)
- endpoint groups, effect on, [12-32](#)
- endpoints, [12-5](#)
- endpoints, effect on, [12-3](#)
- expansion of
 - about, [3-9](#)
 - addition of more nodes, [3-10](#)
 - controller node, [3-9](#)
- FIPS mode for individual nodes, setting, [18-20](#)
- host name network setting for individual nodes, [18-18](#)
- inconsistency resolution, [3-18](#)
- initial node, [3-7](#)
- Key Administrator role, effect on, [9-6](#)
- keys, effect on, [10-24](#)
- maximum disable node duration, [18-27](#)
- mid-size cluster, [3-15](#)
- migrating standalone Key Vault server to, [3-12](#)
- mode types, [3-6](#)
- multi-master clusters
 - expansion of
 - candidate nodes, [3-10](#)
- name conflict resolution, [3-18](#)
- network services for individual nodes, [18-18](#)
- node limitations, [3-4](#)
- operations permitted on modes, [3-7](#)
- Oracle Key Vault management console,
 - setting timeout, [18-29](#)
- overview, [3-1](#)
- primary-standby to multi-master cluster, [3-12](#)
- read-only mode, [3-6](#)
- read-only node, [3-6](#)
- read-only restricted mode, [3-6](#)
- read-write mode, [3-6](#)
- read-write node, [3-5](#)
- reconfiguration changes, [3-9](#)
- RESTful services enablement, [18-28](#)
- restore operations, [21-15](#)
- reverse SSH tunnels, [6-8](#)
- security objects, effect on, [10-24](#)
- size and availability, [3-13](#)
- SNMP settings, [18-29](#)
- SNMP settings for individual node, [18-24](#)

multi-master clusters (*continued*)

- syslog destination
 - clearing, [22-16](#)
 - setting, [22-15](#)
- syslog settings, [18-27](#)
- syslog settings, node, [18-21](#)
- System Administrator role, effect on, [9-5](#)
- system settings, [18-16](#)
- system settings for individual nodes, [18-17](#)
- system time for cluster
 - setting, [18-26](#)
- system time for individual nodes
 - setting, [18-20](#)
- system users, effect on, [9-7](#)
- two data centers, [3-15](#)
- two nodes, [3-14](#)
- user accounts, effect on, [9-4](#)
- user groups, changing description, [9-27](#)
- user groups, creating in, [9-24](#)
- user groups, deleting, [9-28](#)
- user groups, effect on, [9-24](#)
- user groups, removing users from, [9-28](#)
- user groups, renaming, [9-27](#)
- users, effect on, [9-6](#)
- virtual wallet user access to, [9-16](#)

MySQL integration with Oracle Key Vault, [14-11](#)

N

naming conflicts

- about, [4-16](#)
- accepting suggested name, [4-17](#)
- changing suggested name, [4-17](#)
- finding, [4-16](#)

network details

- configuring for non-multi-master clusters, [18-4](#)

network interface

- checking status of, [18-4](#)

network services

- configuring for non-multi-master clusters, [18-6](#)

node certificates

- about, [19-32](#)
- configuring, [19-6](#), [19-33](#)
- finding expiration date, [19-11](#)
- guidelines for rotating, [19-34](#)
- rotating, [19-33](#)

nodes

- creating first node, [4-2](#)
- deleting, [4-10](#)
- disabling, [4-9](#)
- disabling replication, [4-12](#)
- enabling, [4-10](#)
- enabling replication, [4-12](#)
- force deleting, [4-11](#)
- restarting cluster services for, [4-12](#)
- terminating pairing of, [4-8](#)

- NTP servers
 configuring DNS for non-multi-master clusters, [18-9](#)
- O**
-
- OASIS Key Management Interoperability Protocol (KMIP)
 Oracle Key Vault implementation of, [1-14](#)
- objects
 naming guidelines, [2-15](#)
- OKV_HOME environment variable
 non-database utilities, [13-10](#)
- okvclient.jar
 downloading for installation on endpoint, [13-3](#)
- okvclient.ora file
 about, [13-11](#)
 unmodifiable parameters, [13-13](#)
- okvutil errors
 common errors, [B-16](#)
- okvutil utility, [1-16](#), [1-17](#)
 about, [1-16](#), [1-17](#)
 changepwd command, [B-3](#)
 diagnostics command, [B-3](#)
 download command, [B-4](#)
 list command, [B-7](#)
 sign command, [B-12](#)
 sign-verify command, [B-14](#)
 syntax, [B-2](#)
 upload command, [B-9](#)
 used to manage endpoints, [B-1](#)
- online master encryption keys, [1-4](#)
 about using with Oracle Key Vault, [1-4](#)
 centralized management of TDE keys, [1-4](#)
 Oracle Data Guard connection, [14-5](#)
 Oracle GoldenGate, [14-3](#)
- operations, restrictions and conditions of, [A-1](#)
- options for access control, [2-4](#)
- Oracle Active Data Guard
 support for data moves, [14-11](#)
- Oracle Audit Vault
 checking monitoring for multi-master cluster node, [18-25](#)
 checking monitoring for non-multi-master clusters, [18-15](#)
- Oracle Audit Vault integration
 checking environment, [22-68](#)
 deleting integration, [22-71](#)
 integration steps, [22-68](#)
 multi-master cluster integration, [22-72](#)
 primary-standby integration, [22-72](#)
 resuming monitoring operation, [22-71](#)
 suspending monitoring operation, [22-71](#)
 viewing collected audit data, [22-70](#)
- Oracle Cloud Infrastructure database instance endpoints
 about, [6-1](#)
- Oracle Data Guard
 migrating Oracle wallets, [14-5](#)
 online master encryption keys connection, [14-5](#)
 reverse migrating wallets, [14-6](#)
 uploading wallets to Oracle Key Vault, [14-4](#)
- Oracle Data Pump support for data moves, [14-11](#)
- Oracle GoldenGate
 online master encryption keys with
 about, [14-3](#)
 TDE wallet migration
 about, [14-3](#)
 wallets used with, [14-2](#)
- Oracle Key Vault
 administering cluster environments, [18-16](#)
 benefits, [1-2](#)
 deployment architecture, [2-2](#)
 deployment overview, [1-17](#)
 key management, about, [1-1](#)
 standards and protocols, [1-14](#)
 who should use, [1-8](#)
- Oracle Key Vault Backup Failed, [C-28](#)
- Oracle Key Vault client
 C SDK, [1-17](#)
 Java SDK, [1-17](#)
- Oracle Key Vault client software
 endpoints not using, [13-10](#)
- Oracle Key Vault compute instance
 about, [5-1](#)
 about provisioning, [5-4](#)
 benefits, [5-2](#)
 finding image, [5-5](#)
 launching process, [5-5](#)
 launching, about, [5-4](#)
 migrating data out of compute instance, [5-10](#)
 migrations, about, [5-9](#)
 post-installation tasks, [5-6](#)
 post-launch tasks, [5-6](#)
 prerequisites, [5-5](#)
 restarting, [5-7](#)
 starting, [5-7](#)
 stopping, [5-7](#)
 system settings, [5-8](#)
 terminating, [5-9](#)
 transitioning data into a compute instance, [5-10](#)
- Oracle Key Vault compute instances
 backup operations, [5-8](#)
 restore operations, [5-8](#)
- Oracle Key Vault concepts, [2-1](#)
- Oracle Key Vault endpoint utility, [1-16](#), [1-17](#)
 about, [1-16](#), [1-17](#)
 See also okvutil utility

Oracle Key Vault features

- Advanced Cluster File System encryption key management, [1-15](#)
- audit and monitoring services, external support for, [1-15](#)
- backup and restore support for security objects, [1-13](#)
- centralized storage and management of security objects, [1-10](#), [1-14](#)
- database release and platform support, [1-14](#)
- DBaaS endpoint support, [1-15](#)
- HSM integration, [1-16](#)
- key lifecycle management, [1-11](#)
- MySQL integration, [1-15](#)
- persistent master encryption key cache, [1-12](#)
- primary-standby environment support, [18-36](#)
- reporting and alerts, [1-12](#)
- RESTful service support, [1-13](#)
- separation of duties, [1-12](#)

Oracle Key Vault general system administration

- about, [18-2](#)

Oracle Key Vault Installation Failed, [C-32](#)

Oracle Key Vault interfaces, [1-16](#), [1-17](#)

- management console, [1-16](#)
- RESTful services, [1-17](#)

Oracle Key Vault keys

- finding, [B-7](#)

Oracle Key Vault maintenance

- dashboard, [18-2](#)
- system settings, [18-17](#)

Oracle Key Vault management console

- about, [1-16](#)
- timeout for multi-master cluster nodes, [18-29](#)
- timeout for Web sessions for non-multi-master clusters, [18-15](#)

Oracle Key Vault Multi-Master Cluster, [A-1](#)

Oracle Key Vault state

- viewing, [18-2](#)

Oracle Key Vault Upgrade Failed, [C-33](#)

Oracle Key Vault use cases, [1-4](#)

Oracle Real Application Clusters

- support for data moves, [14-11](#)
- wallets, [14-1](#)

Oracle Recovery Manager (RMAN) support for data moves, [14-11](#)

Oracle wallets

- downloading
 - about, [11-20](#)
- uploading
 - about, [11-20](#)

Oracle ZFS Storage Appliance

- used for Oracle Key Vault backups, [21-22](#), [21-23](#)

P

passphrases, [18-32](#)

- changing in clusters environment, [18-34](#)
- changing in non-clusters environment, [18-33](#)
- recovering system, [18-32](#)
- See also passwords

passwords, [18-32](#)

- about changing, [9-17](#), [9-18](#)
- centrally managed, [17-9](#)
- changing endpoint password, [B-3](#)
- changing password automatically, [9-19](#)
- changing password manually, [9-19](#)
- changing support user, [9-30](#)
- changing your own, [9-18](#)
- controlling manual password reset operations, about, [9-20](#)
- controlling manual password reset operations, configuration, [9-21](#)
- See also passphrases

persistent master encryption key cache, [11-2](#)

- about, [11-2](#)
- architecture, [11-2](#)
- caching master encryption keys in-memory, [11-3](#)
- contents of, listing, [11-9](#)
- environment variables, importance of setting, [11-3](#)
- modes of operation
 - first mode, [11-4](#)
 - Oracle Key Vault first mode, [11-4](#)
- Oracle Database deployments, [11-10](#)
- PEXPIRE PKCS11 PERSISTENT CACHE ON DATABASE SHUTDOWN parameter, [11-8](#)
- PKCS11_CACHE_TIMEOUT parameter, [11-6](#)
- PKCS11_CONFIG_PARAM_REFRESH_INTERVAL parameter, [11-7](#)
- PKCS11_PERSISTENT_CACHE_FIRST parameter, [11-7](#)
- PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW parameter, [11-8](#)
- PKCS11_PERSISTENT_CACHE_TIMEOUT parameter, [11-6](#)
- refresh window, [11-5](#)
- storage location, [11-3](#)
- PKCS11_CACHE_TIMEOUT parameter, [11-6](#)
- PKCS11_CONFIG_PARAM_REFRESH_INTERVAL parameter, [11-7](#)
- PKCS11_PERSISTENT_CACHE_FIRST parameter, [11-7](#)
- PKCS11_PERSISTENT_CACHE_REFRESH_WINDOW parameter, [11-8](#)
- PKCS11_PERSISTENT_CACHE_TIMEOUT parameter, [11-6](#)

powering off Oracle Key Vault for non-multi-master clusters, [18-16](#)

powering off Oracle Key Vault nodes, [18-25](#)

primary servers
 role in primary-standby configuration, [23-4](#)

primary-standby
 restore operations, [21-15](#)

primary-standby configuration
 about, [23-2](#)
 benefits, [23-3](#)
 best practices, [23-19](#)
 changing SNMP settings on standby server, [22-4](#)
 checking TDE wallet migration for logical standby, [14-8](#)
 configuring primary server, [17-5–17-7](#), [23-5](#)
 configuring standby server, [23-6](#)
 difference from multi-master clusters, [23-4](#)
 disabling, [23-11](#)
 downtime, minimizing, [18-42](#)
 enabling primary-standby on primary, [23-8](#)
 migrating TDE wallets to Oracle Key Vault for standby, [14-7](#)
 Oracle Audit Vault integration, [22-72](#)
 persistent master encryption key cache
 downtime, minimizing, [18-42](#)
 primary server
 configuring, [17-5–17-7](#), [23-5](#)
 enabling for primary-standby, [23-8](#)
 primary server role, [23-4](#)
 read-only restricted mode
 downtime, minimizing, [18-42](#)
 read-only restricted mode disabled, [23-13](#)
 read-only restricted mode enabled, [23-13](#)
 read-only restricted mode impact, [23-12](#)
 read-only restricted mode state during network failure, [23-16](#)
 read-only restricted mode state during primary server failure, [23-16](#)
 read-only restricted mode state during standby server failure, [23-16](#)
 read-only restricted mode states, [23-15](#)
 read-only restricted mode, disabling, [23-17](#)
 read-only restricted mode, enabling, [23-17](#)
 read-only restricted mode, recovering from, [23-18](#)
 restoring primary-standby after, [23-10](#)
 reverse SSH tunnels, [6-9](#)
 standby server
 configuring, [23-6](#)
 standby server role, [23-4](#)
 switching servers, [23-9](#)
 unpairing, [23-11](#)

primary-standby environments
 console certificates, [20-4](#)

primary-standby server
 moving to multi-master cluster, [3-12](#)

Primary-Standby Status, [C-38](#)

privileges, [2-4](#)
 access control options, [2-4](#)
 access grants for virtual wallets, [2-4](#)
 See also access control

R

re-enroll an endpoint, [C-3](#)

read-only mode
 about, [3-6](#)

read-only nodes
 about, [3-6](#)
 creating, [4-6](#)

read-only restricted mode
 about, [3-6](#)
 disabling, [23-17](#)
 enabling, [23-17](#)
 notifications, [23-19](#)
 primary-standby configuration with read-only restricted mode enabled, [23-13](#)
 primary-standby configuration without read-only restricted mode enabled, [23-13](#)
 primary-standby configuration, impact on, [23-12](#)
 recovering primary-standby, [23-18](#)

read-only restricted mode states
 network failure in primary-standby configuration, [23-16](#)
 primary server failure, [23-16](#)
 primary-standby configuration, [23-15](#)
 standby server failure, [23-16](#)

read-write mode
 about, [3-6](#)

read-write nodes
 about, [3-5](#)

read-write pair of nodes
 creating, [4-4](#)

read-write pairs of nodes
 creating additional, [4-8](#)

rebooting Oracle Key Vault for non-multi-master clusters, [18-16](#)

rebooting Oracle Key Vault nodes, [18-25](#)

recovery passphrase
 about recovering, [18-32](#)
 changing in clusters environment, [18-34](#)
 changing in non-clusters environment, [18-33](#)
 protecting the backup, [21-12](#)

rekey operation, [1-4](#)

remote backup destination
 about, [21-2](#)

remote backup destinations
 changing settings, [21-6](#)
 creating, [21-4](#)
 deleting, [21-7](#)

Remote Backup Failed, [C-29](#)

- remote monitoring
 - about, [22-2](#)
 - changing settings on standby server, [22-4](#)
 - changing user name and password, [22-4](#)
 - granting SNMP access to users, [22-3](#)
 - remotely monitoring using SNMP, [22-5](#)
 - removing user from a user group, [9-28](#)
 - renaming a user group, [9-27](#)
 - reporting, [1-12](#)
 - reports
 - about, [22-73](#)
 - endpoint reports, [22-77](#)
 - key management reports for Oracle endpoints reports, [22-74](#)
 - keys reports, [22-75](#)
 - notification report, [22-79](#)
 - secrets reports, [22-76](#)
 - system reports, [22-79](#)
 - user reports, [22-78](#)
 - wallets reports, [22-75](#)
 - restarting Oracle Key Vault for non-multi-master clusters, [18-16](#)
 - restarting Oracle Key Vault nodes, [18-25](#)
 - RESTful command-line interface commands reports, [22-79](#)
 - RESTful services
 - about, [1-17](#)
 - console certificates, [20-4](#)
 - disabling for non-multi-master clusters, [18-14](#)
 - enabling for non-multi-master clusters, [18-14](#)
 - multi-master clusters, enablement, [18-28](#)
 - restoring data
 - about, [21-13](#)
 - best practices, [21-26](#)
 - multi-master clusters, [21-15](#)
 - primary-standby deployment, [21-15](#)
 - procedure, [21-13](#)
 - system state after, [21-16](#)
 - third-party certificates, [21-15](#)
 - roles
 - about, [2-6](#)
 - root user
 - about, [2-16](#)
 - Roots of Trust (RoT), [1-16](#)
- ## S
-
- secrets
 - guidance for SSH, [17-4](#)
 - reports, [22-76](#)
 - secure user management, [9-20](#)
 - security objects
 - activating, [10-25](#)
 - adding to virtual wallets, [10-6](#)
 - deactivating, [10-25](#)
 - deleting, [10-26](#)
 - security objects (*continued*)
 - details of, about, [10-29](#)
 - downloading to different types, [B-4](#)
 - modifying details of, [10-34](#)
 - multi-master clusters, effect on, [10-24](#)
 - removing from virtual wallets, [10-6](#)
 - revoking, [10-26](#)
 - searching for object items, [10-30](#)
 - state of, managing, [10-24](#)
 - viewing details of, [10-31](#)
 - virtual wallets, creating for, [10-2](#)
 - virtual wallets, modifying, [10-4](#)
 - self-enrollment, for endpoints, [12-10](#)
 - separation of duties, [1-12](#)
 - how Oracle Key Vault manages, [2-5](#)
 - users, [9-2](#), [9-3](#), [9-5](#)
 - server certificates
 - about, [19-32](#)
 - configuring, [19-6](#), [19-33](#)
 - finding expiration date, [19-11](#)
 - guidelines for rotating, [19-34](#)
 - rotating, [19-33](#)
 - sign signatures
 - accessing with okvutil sign, [B-12](#)
 - sign-verify
 - accessing with okvutil sign-verify, [B-14](#)
 - Single Sign-On
 - Creating User, [7-5](#)
 - Single Sign-On
 - See IDP
 - SNMP
 - about, [22-2](#)
 - changing settings on standby server, [22-4](#)
 - changing user name and password, [22-4](#)
 - example of simplified remote monitoring, [22-8](#)
 - granting access to user, [22-3](#)
 - Management Information Base (MIB) variables, [22-6](#)
 - remotely monitoring Oracle Key Vault, [22-5](#)
 - SNMP settings
 - multi-master clusters, [18-29](#)
 - nodes, [18-24](#)
 - split-brain scenarios, [23-2](#)
 - sqlnet.ora file
 - environment variables and, [13-10](#)
 - SSH
 - guidance for credentials, [17-4](#)
 - guidance for secrets, [17-4](#)
 - SSH key files
 - downloading from Key Vault to a wallet, [B-4](#)
 - SSH Tunnel Add Failure, [C-42](#)
 - SSH tunnels
 - creating between Oracle Key Vault and DBaaS instance, [6-5](#)
 - deleting, [6-11](#)
 - disabling, [6-10](#)

SSH tunnels (*continued*)

- multi-master clusters, [6-10](#), [6-11](#)
- not active, [6-11](#)
- reverse SSH tunnel in multi-master cluster, [6-8](#)
- reverse SSH tunnel in primary-standby configuration, [6-9](#)
- viewing details, [6-9](#)

SSL Client Error, [C-43](#)

SSL Layer Error, [C-21](#)

standby servers

- role in primary-standby configuration, [23-4](#)

support user

- about, [2-16](#)

syslog

- configuring for non-multi-master clusters, [18-12](#)

syslog configuration

- audit records, [22-48](#)
- destination
 - clearing, [22-16](#)
 - setting, [22-15](#)

syslog settings

- multi-master cluster node, [18-21](#)
- multi-master clusters, [18-27](#)

System Administrator role

- about, [2-7](#)
- multi-master cluster effect on, [9-5](#)

system diagnostics

- See diagnostic reports

System Metrics

- CPU and Memory Usage, [22-22](#)

system recovery, [2-15](#), [18-32](#)

system time

- setting for non-multi-master clusters, [18-9](#)

system users

- multi-master cluster effect on, [9-7](#)

T

TDE direct connect

- See online master encryption keys

TDE master encryption keys

- centralized management, [1-4](#)

TDE wallets

- Oracle GoldenGate, [14-3](#)

TDE-enabled databases, [11-11](#)

- about configuring Key Vault for, [11-11](#)
- configuring environment for, [11-12](#)
- integrating TDE with Key Vault, [11-15](#)
- limitations of TDE endpoint integration, [11-12](#)

third-party certificates

- restoring data, [21-15](#)

time

- setting for cluster, [18-26](#)
- setting for node, [18-20](#)

time (*continued*)

- setting for non-multi-master clusters, [18-9](#)

Transparent Data Encryption, [11-11](#)

- downtime, minimizing for TDE heartbeat, [18-42](#)
- endpoint management, [13-11](#)
 - See also TDE-enabled databases

transportable tablespaces support for data moves, [14-11](#)

types of backups, [21-8](#)

U

Unable to boot the Virtual Machine, [C-35](#)

Unable to schedule new backup, [C-28](#)

Upgrade Failure, [C-34](#)

upgrading endpoint software

- unenrolled endpoint, [13-14](#)

upload command (okvutil), [B-9](#)

uploading

- credential files, [17-2](#)
- JKS and JCEKS keystores, [11-23](#), [11-24](#)
- wallet to Oracle RAC, [14-1](#)
- wallets, [11-21](#)

Uploading Java Keystore, [C-20](#)

use cases, [1-4](#)

- centralized storage, [1-6](#)
- key rotation, [1-4](#)
- online management of keys and secret data, [1-8](#)
- storage of credential files, [1-7](#)

user accounts

- multi-master clusters, effect on, [9-4](#)

user groups, [9-24](#)

- adding a user, [9-26](#)
- changing description, [9-27](#)
- creating, [9-24](#)
- deleting, [9-28](#)
- granting access to virtual wallet, [9-26](#)
- modifying virtual wallets from Keys & Wallets tab, [10-9](#)
- multi-master clusters, effect on, [9-24](#)
- naming guidelines, [2-15](#)
- removing access to virtual wallets from Keys & Wallets tab, [10-8](#)
- removing access to virtual wallets from User's tab, [10-11](#)
- removing user from, [9-28](#)
- renaming, [9-27](#)
- revoking access to virtual wallets, [10-11](#)

user-defined keys

- about, [11-26](#)
- activating, [11-28](#)
- uploading to Oracle Key Vault, [11-27](#)

users, [9-24](#)

- about changing password, [9-18](#)

users (*continued*)

- about user accounts, [9-2](#), [9-3](#), [9-5](#)
- administrative roles, about, [9-11](#)
- administrative roles, granting or changing, [9-11](#)
- administrative roles, revoking, [9-15](#)
- changing own password, [9-18](#)
- changing password automatically, [9-19](#)
- changing password manually, [9-19](#)
- changing passwords, about, [9-17](#)
- changing support user password, [9-30](#)
- changing user email address, [9-22](#)
- controlling manual password reset operations, about, [9-20](#)
- controlling manual password reset operations, configuration, [9-21](#)
- Create Endpoint Group privilege, granting or changing, [9-13](#)
- Create Endpoint privilege, granting or changing, [9-12](#)
- creating accounts, [9-7](#)
- deleting accounts, [9-10](#)
- disabling email notifications, [9-23](#)
- endpoint administrators
 - about, [2-16](#)
- endpoint privileges, about, [9-11](#)
- endpoint privileges, revoking, [9-15](#)
- granting access to virtual wallet, [9-16](#)
- Manage Endpoint Group privilege, granting or changing, [9-14](#)
- Manage Endpoint privilege, granting or changing, [9-13](#)
- modifying virtual wallets from Keys & Wallets tab, [10-9](#)
- multi-master cluster effect on, [9-6](#)
- naming guidelines, [2-15](#)
- removing access to virtual wallets from Keys & Wallets tab, [10-8](#)
- removing access to virtual wallets from User's tab, [10-10](#)
- reports, [22-78](#)
- root user
 - about, [2-16](#)
- support user
 - about, [2-16](#)
- view account details, [9-9](#)
 - See also user groups

V

viewing user account details, [9-9](#)

virtual wallets

- about, [10-2](#)
- access management from Keys and Wallets tab, [10-7](#)
- adding endpoint access to, [12-29](#)

virtual wallets (*continued*)

- adding security objects to, [10-6](#)
- creating, [10-2](#)
- deleting, [10-7](#)
- endpoint group access grant, [12-35](#)
- granting access to from Keys & Wallets tab, [10-8](#)
- granting access to from Users tab, [10-10](#)
- granting user access to, [9-16](#), [9-26](#)
- granting user group access to from User's tab, [10-11](#)
- modifying, [10-4](#)
- modifying from Keys & Wallets tab, [10-9](#)
- naming guidelines, [2-15](#)
- removing security objects from, [10-6](#)
- removing user access to from Users tab, [10-10](#)
- revoking endpoint access, [12-30](#)
- revoking user group access from, [10-11](#)

W

wallet not open error, [C-9](#)

wallets

- checking TDE wallet migration for logical standby
 - about, [14-8](#)
- downloading
 - guidance, [11-23](#)
 - procedure, [11-22](#)
- downloading from Key Vault to a wallet, [B-4](#)
- endpoint group access grant, [12-35](#)
- endpoints, associating, [12-27](#)
- endpoints, viewing wallet items for, [12-31](#)
- key rotation guidance, [11-23](#)
- migrating existing TDE wallet to Key Vault
 - about, [11-16](#)
 - procedure, [11-17](#)
- migrating TDE to Key Vault for logical standby database
 - about, [14-7](#)
- migrating to Oracle Data Guard, [14-5](#)
- Oracle GoldenGate use with, [14-2](#)
- Oracle Real Application Clusters environment, [14-1](#)
- overwriting danger of, [11-23](#)
- reports, [22-75](#)
- restoring database contents previously encrypted by TDE
 - about, [11-19](#)
- reverse migrating in Oracle Data Guard
 - about, [14-6](#)
- setting default for endpoint, [12-27](#)
- sharing with multiple endpoints guidance, [11-23](#)

wallets (*continued*)

- uploading
 - guidance, [11-23](#)
 - procedure, [11-21](#)
- uploading contents to Key Vault server, [B-9](#)

wallets (*continued*)

- uploading in Oracle Data Guard
 - about, [14-4](#)
 - procedure, [14-5](#)