Oracle® Database Appliance

X9-2 Deployment and User's Guide

Release 19.25 for Linux x86-64
G14676-02
December 2024

Oracle Database Appliance X9-2 Deployment and User’s Guide, Release 19.25 for Linux x86-64

G14676-02

Copyright © 2022, 2024, Oracle and/or its affiliates.

Primary Author: Aparna Kamath

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 xix
Documentation Accessibility xix
Related Documents xx
Conventions xx

1 Oracle Database Appliance Checklists

Checklist for System Requirements 1-1
Deployment Checklist for Oracle Database Appliance Bare Metal Systems 1-6
Checklist for Custom Network Address Configuration 1-8

2 Introduction to Oracle Database Appliance

About Oracle Database Appliance 2-1
Oracle Database Appliance Software Terminology 2-3
About the Browser User Interface 2-4

3 Readying Oracle Database Appliance for Deployment

Attaching Storage Cables to Oracle Database Appliance X9-2-HA 3-1
Attaching Network Cables to Oracle Database Appliance X9-2-HA 3-4
Attaching Network Cables to Oracle Database Appliance X9-2S or X9-2L 3-6
Attaching Power Cords and Initializing Components 3-9
Configuring Oracle Integrated Lights Out Manager 3-9
Powering On Oracle Database Appliance the First Time 3-11

4 Provisioning Oracle Database Appliance Bare Metal System

Verifying Cabling and Network Connections 4-1
Plumbing the Network 4-4
Downloading Oracle Database Appliance Software 4-6
Installing Oracle Database Appliance Software 4-7
Creating the Appliance 4-9

About Enabling Oracle ASMFD on Oracle Database Appliance 4-15

5 Implementing Multi-User Access on Oracle Database Appliance

About Multi-User Access on Oracle Database Appliance 5-2
About Users, Roles, Entitlements, and Operations on a Multi-User Access Enabled Oracle Database Appliance System 5-4
About Granting and Revoking Resource Access 5-5
Using Passwordless Multi-User Access on Oracle Database Appliance 5-6
- About Passwordless Multi-User Access on Oracle Database Appliance 5-7
- Provisioning Oracle Database Appliance with Passwordless Multi-User Access 5-9
- Creating, Viewing, and Deleting Users on Oracle Database Appliance with Passwordless Multi-User Access 5-10
- Changing the Password for a User Account on Oracle Database Appliance with Passwordless Multi-User Access 5-11
- Example JSON Files to Create a Single-Node Passwordless Multi-User Enabled Appliance 5-12
- Example JSON File to Create a High-Availability Passwordless Multi-User Enabled Appliance 5-17
- Example JSON Files to Create a Single-Node Passwordless Multi-User Enabled DB System 5-21
- Example JSON File to Create a High-Availability Passwordless Multi-User Enabled DB System 5-25

Provisioning Oracle Database Appliance with Multi-User Access 5-29
Creating, Viewing, and Deleting Users on Oracle Database Appliance with Multi-User Access 5-32
Activating a New User on Oracle Database Appliance with Multi-User Access 5-33
Granting and Revoking Resource Access on Oracle Database Appliance with Multi-User Access 5-34
Viewing Roles, Operations, and Entitlements on Oracle Database Appliance with Multi-User Access 5-35
Managing Databases and Database Homes on Oracle Database Appliance with Multi-User Access 5-36
Changing the Password for a User Account on Oracle Database Appliance with Multi-User Access 5-38
Resetting the Password for a Locked User Account on Oracle Database Appliance with Multi-User Access 5-38

6 Oracle Database Appliance Postinstallation Tasks

Configuring CPU Core Count 6-1
Modifying Oracle ASM Listener Port After Deployment 6-2
Securing Oracle ILOM Service Processors 6-2
Changing Oracle Database Appliance Passwords 6-3

Changing the DNS Server Address on Oracle Database Appliance 6-4

7 Re-imaging Oracle Database Appliance

Uninstalling Oracle Database Appliance Components 7-1
Reimaging an Oracle Database Appliance Baremetal System 7-4
Performing Secure Erase of Data on Storage Disks 7-6
Redeployment of Oracle Database Appliance 7-7

8 Patching Oracle Database Appliance

About Patching Oracle Database Appliance 8-2
About Patching Prechecks 8-4
About Enabling NUMA on Oracle Database Appliance 8-6
Minimizing Downtime When Patching Oracle Database Appliance 8-7
About Updating Databases and Database Homes 8-8
Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line 8-9
Patching Oracle Database Appliance DB Systems Using the Command-Line 8-25
Patching Oracle Database Appliance Bare Metal Systems Using the Browser User Interface 8-31
Patching Oracle Database Appliance DB Systems Using the Browser User Interface 8-33
Updating Oracle Database Appliance Repository with Database Clone Files For Bare Metal Systems 8-35
Patching Databases Using ODACLI Commands or the BUI 8-36
Patching Existing Database Homes Using ODACLI or the BUI 8-37
Patching Non-Provisioned Oracle Database Appliance Systems 8-38
Applying Additional Patches and Updates on Bare Metal Systems 8-39
- Enabling Kernel Updates 8-39
- Applying Out-of-Cycle Database Patches 8-40

Cleaning Up the Patch Repository 8-42
About Upgrading to a Different Database Home 8-42
Setting Dynamic Default Parameters 8-43

9 Managing Oracle Databases

About Administrative Groups and Users on Oracle Database Appliance 9-2
About Data Migration Options for Oracle Database Appliance 9-3
About Creating Database Homes on Oracle ACFS Storage 9-3
About Standard Edition High Availability for Oracle Database Appliance 9-5
About Enterprise Edition High Availability for Oracle Database Appliance 9-6
About Transparent Database Encryption (TDE) in Oracle Database Appliance 9-7
Storing Transparent Database Encryption Keys on Oracle Key Vault Server 9-9
- About Oracle Key Vault 9-9
- Creating TDE-Enabled Databases With User Credentials 9-12

Creating TDE-Enabled Databases Using Oracle Key Vault With Endpoints in Oracle Key Vault 9-16
Managing Oracle Key Vault Server Configuration Operations Using ODACLI Commands 9-22
Managing Database Operations on TDE-enabled Databases with TDE keys stored on Oracle Key Vault 9-24
Re-keying TDE Master Encryption Key for TDE-Enabled Databases 9-24
Performing Backup and Recovery of TDE-enabled Databases 9-25
Restoring TDE-Enabled Databases Using Oracle Key Vault User Credentials 9-30
Restoring TDE-Enabled Databases With Endpoints and Wallet Manually Created in Oracle Key Vault 9-38
Deleting TDE-enabled Databases 9-48
Creating an Oracle Key Vault User Using BUI 9-50
Creating Oracle Key Vault Server Configuration Object Using the BUI 9-50
Granting and Revoking Access on Oracle Key Vault Server Configuration Using BUI 9-51
Managing Oracle Key Vault Server Configuration Operations Using BUI 9-51
Manual Migration and Encryption of Data into TDE-Enabled Database 9-52
Migrating Tablespaces from Non-TDE Oracle ACFS Database to Oracle ACFS Database on Oracle Database Appliance 9-52
Migrating Tablespaces from Non-TDE Oracle ASM Database to Oracle ASM Database on Oracle Database Appliance 9-58
Migrating Tablespaces from Non-TDE Oracle ASM Database to Oracle ASM CDB Database on Oracle Database Appliance 9-65
Migrating Tablespaces from Non-TDE Oracle ACFS Database to Oracle ACFS CDB Database on Oracle Database Appliance 9-72
Working with Databases 9-80
Viewing Databases 9-81
Creating a Database Using the Browser User Interface 9-81
Managing Database Homes on Oracle ACFS Storage 9-83
Creating a Database Using ODACLI Commands 9-85
Cloning a Database from Backup 9-86
Cloning an Oracle ACFS Database 9-87
Modifying a Database 9-90
Moving a Database from One Oracle Home to Another 9-91
Upgrading a Database 9-92
Upgrading a Database Manually Using AutoUpgrade 9-93
Deleting a Database 9-100
Working with Database Homes 9-100
About Managing Multiple Oracle Homes on Oracle Database Appliance 9-101
Viewing Database Homes 9-101
Creating a Database Home 9-102
Deleting a Database Home 9-102
Migrating Databases 9-102
About Migrating Databases 9-103
Configuring a Static Listener 9-104

Migrating a Database 9-104
Registering a Database 9-106
About Registering a TDE-Enabled Database 9-107
About Managing Multiple Database Instances Using Instance Caging 9-113
Oracle EM Express and DB Console 9-113

10 Managing Storage

About Managing Storage 10-1
About Managing Oracle ASM Disks 10-1
Managing Storage on Single-Node Systems 10-2
About Storage on Oracle Database Appliance X9-2S and X9-2L 10-2
Adding NVMe Storage Disks 10-3
Replacing NVMe Storage Disks 10-5
Managing Storage on High-Availability Systems 10-6
About Storage Options for Oracle Database Appliance X9-2-HA 10-6
Adding Solid-State Drives (SSDs) for Data Storage 10-8
Adding the Storage Expansion Shelf 10-12

11 Managing Networks

About Network Infrastructure and VLANs on Oracle Database Appliance 11-1
About Oracle Database Appliance X9-2 Network Interfaces 11-2
Viewing Configured Networks and Network Interfaces 11-4
Creating a Network 11-4
Creating a Physical Network 11-5
Updating a Network 11-5
Adding a Network Interface 11-6
Deleting a Network Interface 11-6
Deleting a Network Interface Using Browser User Interface 11-6
Deleting a Network Using the Browser User Interface 11-7

12 Backup, Restore and Recover Databases

About Database Backup and Recovery Options 12-1
Configuring NFS Backup and Restore on Oracle Database Appliance 12-9
Configuring an NFS Backup Location 12-9
Backing Up the Database to NFS Location 12-12
Restoring the Database from NFS Backup 12-15
Configuring Agent Proxy Settings for Object Store Access 12-17
Creating a Database Backup Policy 12-18
Updating a Database with a Backup Policy 12-19

Updating a Backup Policy 12-19
Backing Up a Database Using the Browser User Interface 12-19
Viewing Backup Reports 12-20
Recovering a Database Using the Browser User Interface 12-20
Deleting a Backup Using the Browser User Interface 12-21
Deleting a Backup Policy 12-21
Using the CLI to Backup, Restore, and Recover 12-21
Preparing for Database Backup to Disk 12-22
Preparing for Database Backup to NFS Location 12-23
Preparing for Database Backup to Oracle Cloud 12-24
Backing Up a Database with the CLI 12-25
Recovering a Database with the CLI 12-26
Restoring a Database with the CLI 12-28
Restoring Archivelog with the CLI 12-31
Deleting Backups with the CLI 12-31

13 Using Oracle Data Guard for Disaster Management and Recovery on Oracle Database Appliance

About Oracle Data Guard on Oracle Database Appliance 13-2
Configuring Oracle Data Guard on Oracle Database Appliance 13-2
Viewing Oracle Data Guard Status on Oracle Database Appliance 13-12
Performing Oracle Data Guard Operations 13-12
Creating an Oracle Data Guard Network 13-14
Migrating an Oracle Database from One System to Another Using Integrated Oracle Data Guard 13-15
Patching Oracle Data Guard Databases on Oracle Database Appliance 13-20
Upgrading Oracle Data Guard Databases on Oracle Database Appliance 13-22
Enabling Oracle Active Data Guard on Existing Configurations 13-23
Registering Oracle Data Guard on Oracle Database Appliance 13-23
Deconfiguring Oracle Data Guard on Oracle Database Appliance 13-28

14 Managing CPU Pools on Oracle Database Appliance Bare Metal and KVM Deployments

About CPU Pools 14-1
Creating a CPU Pool in a Bare Metal or KVM Deployment 14-3
Remapping CPU Pools in a Bare Metal or KVM Deployment 14-4
Viewing CPU Pools in a Bare Metal or KVM Deployment 14-5
Modifying a CPU Pool in a Bare Metal or KVM Deployment 14-6
Deleting a CPU Pool in a Bare Metal or KVM Deployment 14-6
Viewing CPU Pool Audit Records in a Bare Metal or KVM Deployment 14-7

About Over-Subscribed CPU Pools 14-7

15 Managing an Oracle Database Appliance KVM Deployment

About Oracle Database Appliance KVM Deployment 15-2
About Oracle Database Appliance KVM Virtualization Architecture 15-3
About Virtual Machines and Oracle Database Appliance KVM Deployments 15-4
About KVM DB Systems on Oracle Database Appliance 15-7
Deploying Multiple Databases on DB Systems on Oracle Database Appliance 15-10
Managing DB Systems in KVM Deployment 15-15
- Creating a DB System in a KVM Deployment 15-16
- Listing DB Systems in a KVM Deployment 15-22
- Describing a DB System in a KVM Deployment 15-23
- Modifying a DB System in a KVM Deployment 15-24
- Attaching and Detaching a Network for a DB System in a KVM Deployment 15-26
- Starting a DB System in a KVM Deployment 15-27
- Stopping a DB System in a KVM Deployment 15-27
- Deleting a DB System in a KVM Deployment 15-28
- Managing Backup, Restore, and Recovery on a DB System in a KVM Deployment 15-29
- Managing Shared CPU Pool with DB System in a KVM Deployment 15-29
- Managing Oracle Data Guard on a DB System in a KVM Deployment 15-32

Setting Up and Configuring a KVM Deployment for Applications 15-33
Managing VM Storage in KVM Deployment 15-33
- Creating a VM Storage in a KVM Deployment 15-34
- Viewing VM Storage in a KVM Deployment 15-34
- Modifying VM Storage in a KVM Deployment 15-35
- Deleting a VM Storage in a KVM Deployment 15-36

Managing Virtual Networks in KVM Deployment 15-36
- Creating a Virtual Network in a KVM Deployment 15-37
- Viewing Virtual Networks in a KVM Deployment 15-38
- Starting and Stopping Virtual Networks in a KVM Deployment 15-38
- Modifying a Virtual Network in a KVM Deployment 15-39
- Deleting a Virtual Network in a KVM Deployment 15-39

Managing Virtual Disks in KVM Deployment 15-40
- Creating a Virtual Disk in a KVM Deployment 15-40
- Viewing Virtual Disks in a KVM Deployment 15-42
- Cloning a Virtual Disk in a KVM Deployment 15-42
- Modifying a Virtual Disk in a KVM Deployment 15-43
- Deleting a Virtual Disk in a KVM Deployment 15-44

Managing Virtual Machines in KVM Deployment 15-44
- Creating a Virtual Machine in a KVM Deployment 15-45
- Cloning a Virtual Machine in a KVM Deployment 15-48

Modifying a Virtual Machine in a KVM Deployment 15-49
Viewing Virtual Machines in a KVM Deployment 15-50
Starting and Stopping Virtual Machines in a KVM Deployment 15-51
Deleting a Virtual Machine in a KVM Deployment 15-51
About Overcommitting Memory or CPUs in an Oracle Database Appliance KVM System 15-52
Example JSON File to Create a Single-Node DB System 15-52
Example JSON File to Create a High-Availability DB System 15-62

16 Oracle Database Appliance Command-Line Interface

About Oracle Database Appliance Command-Line Interface 16-2
Managing ODACLI Privileges and Security with SUDO 16-4
ODACLI Command Changes with Multi-User Access on Oracle Database Appliance 16-6
odacli Network Commands 16-13
odacli configure-firstnet 16-13
odacli create-network 16-16
odacli delete-network 16-18
odacli delete-networkinterface 16-19
odacli describe-network 16-20
odacli describe-networkinterface 16-22
odacli list-networks 16-23
odacli list-networkinterfaces 16-23
odacli modify-network 16-24
odacli Multi-User Access Commands 16-26
odacli list-domains 16-27
odacli describe-domain 16-28
odacli list-tenants 16-29
odacli describe-tenant 16-29
odacli create-user 16-30
odacli activate-user 16-31
odacli list-users 16-32
odacli describe-user 16-33
odacli authorize-user 16-34
odacli delete-user 16-35
odacli list-user-roles 16-36
odacli describe-user-role 16-37
odacli list-user-entitlements 16-38
odacli describe-user-entitlement 16-39
odacli list-user-operations 16-40
odacli describe-user-operation 16-40
odacli list-resources-access 16-41
odacli describe-resource-access 16-42

odacli grant-resource-access 16-43
odacli revoke-resource-access 16-44
odacli reset-password 16-45
odacli change-password 16-46
odacli Apply Patch and Update Commands 16-46
odacli cleanup-patchrepo 16-48
odacli create-prepatchreport 16-49
odacli describe-prepatchreport 16-52
odacli list-prepatchreports 16-62
odacli delete-prepatchreport 16-63
odacli list-availablepatches 16-63
odacli describe-component 16-65
odacli describe-latestpatch 16-71
odacli list-agentconfig-parameters 16-72
odacli modify-agentconfig-parameters 16-74
odacli delete-agentconfig-parameters 16-76
odacli update-database 16-78
odacli update-dbhome 16-79
odacli update-dcsadmin 16-81
odacli update-dcsagent 16-81
odacli update-dcscomponents 16-82
odacli update-registry 16-83
odacli update-repository 16-91
odacli update-server 16-92
odacli update-storage 16-94
odacli update-server-postcheck 16-95
odaadmcli orachk 16-95
odacli Appliance Commands 16-96
odacli modify-asmport 16-96
odacli set-credential 16-99
odacli create-appliance 16-99
odacli describe-system 16-100
odacli ping-agent 16-102
odacli list-nodes 16-103
odacli list-featuretracking 16-103
odacli Backup and Recovery Commands 16-106
odacli create-backup 16-108
odacli create-backupconfig 16-109
odacli create-objectstoreswift 16-111
odacli describe-backupconfig 16-112
odacli delete-backup 16-112
odacli delete-backupconfig 16-113

odacli describe-backupreport 16-114
odacli describe-objectstoreswift 16-116
odacli delete-objectstoreswift 16-117
odacli describe-schedule 16-118
odacli irestore-database 16-118
odacli list-backupconfigs 16-124
odacli list-backupreports 16-125
odacli list-objectstoreswifts 16-125
odacli list-schedules 16-126
odacli list-tdebackupreports 16-127
odacli recover-database 16-128
odacli restore-archivelog 16-130
odacli restore-tdewallet 16-130
odacli modify-backupconfig 16-132
odacli modify-objectstoreswift 16-133
odacli modify-schedule 16-134
odacli Oracle Data Guard Commands 16-135
odacli configure-dataguard 16-135
odacli deconfigure-dataguard 16-140
odacli describe-dataguardstatus 16-141
odacli failover-dataguard 16-142
odacli list-dataguardstatus 16-143
odacli reinstate-dataguard 16-143
odacli register-dataguard 16-144
odacli switchover-dataguard 16-144
odacli CPU Core Commands 16-145
odacli list-cpucores 16-145
odacli describe-cpucore 16-146
odacli modify-cpucore 16-146
odacli CPU Pool Commands for Bare Metal and KVM 16-148
odacli create-cpupool 16-149
odacli remap-cpupools 16-150
odacli list-cpupools 16-151
odacli describe-cpupool 16-151
odacli modify-cpupool 16-152
odacli delete-cpupool 16-152
odacli list-auditrecords 16-153
odacli describe-auditrecord 16-154
odacli Database Commands 16-154
odacli clone-database 16-155
odacli create-database 16-165
odacli delete-database 16-172

odacli describe-database 16-173
odacli list-databases 16-175
odacli modify-database 16-175
odacli move-database 16-179
odacli register-database 16-180
odacli upgrade-database 16-183
odacli DBHome Commands 16-185
odacli configure-dbhome-storage 16-186
odacli list-dbhome-storages 16-187
odacli describe-dbhome-storage 16-188
odacli modify-dbhome-storage 16-189
odacli create-dbhome 16-190
odacli delete-dbhome 16-190
odacli describe-dbhome 16-191
odacli list-dbhomes 16-192
odacli Database Storage Commands 16-193
odacli create-dbstorage 16-193
odacli delete-dbstorage 16-195
odacli describe-dbstorage 16-195
odacli describe-dgstorage 16-197
odacli list-dbstorages 16-199
odacli list-dgstorages 16-199
odacli modify-dbstorage 16-200
odacli modify-dgstorage 16-202
odacli Diagnostics Collection Commands 16-203
odacli create-diagcollection 16-203
odacli describe-diagcollection 16-204
odacli list-diagcollections 16-205
odacli delete-diagcollection 16-205
odacli Job Commands 16-206
odacli create-jobdata-retention-policy 16-207
odacli create-purge-jobdata-job 16-208
odacli delete-jobdata-retention-policy 16-212
odacli describe-job 16-213
odacli list-jobs 16-215
odacli list-pendingjobs 16-219
odacli list-jobdata-retention-policy 16-219
odacli list-purge-jobdata-jobs 16-220
odacli list-scheduled-executions 16-221
odacli Oracle Key Vault Server Configuration Commands 16-222
odacli create-okvserverconfig 16-223
odacli copy-okvtdewallet 16-223

odacli delete-okvserverconfig 16-226
odacli describe-okvserverconfig 16-227
odacli list-okvserverconfigs 16-228
Log Commands 16-229
odaadmcli manage diagcollect 16-230
odacli create-auto-logclean-policy 16-231
odacli delete-auto-logclean-policy 16-232
odacli create-logcleanjob 16-233
odacli describe-logcleanjob 16-234
odacli list-auto-logclean-policy 16-235
odacli list-logcleanjobs 16-236
odacli list-logspaceusage 16-236
odacli Oracle Auto Service Request Commands 16-237
odacli configure-asr 16-238
odacli describe-asr 16-240
odacli delete-asr 16-240
odacli export-asrconfig 16-241
odacli modify-asr 16-242
odacli test-asr 16-243
Adaptive Classification and Redaction (ACR) Commands 16-244
odacli describe-acr 16-244
odacli enable-acr 16-245
odacli redact-acr 16-245
odacli list-acr 16-246
odacli disable-acr 16-246
odacli OS Commands 16-247
odacli list-osconfigurations 16-247
odacli modify-osconfigurations 16-248
odaadmcli Hardware Monitoring Commands 16-249
odaadmcli show localdisk 16-249
odaadmcli show cooling 16-250
odaadmcli show env_hw 16-250
odaadmcli show fs 16-251
odaadmcli show memory 16-252
odaadmcli show network 16-252
odaadmcli show power 16-253
odaadmcli show processor 16-254
odaadmcli show server 16-254
Storage Commands 16-255
odaadmcli asmops diskgroup 16-256
odaadmcli expand storage 16-257
odacli list-dgdisks 16-257

odacli validate-storagetopology 16-259
odaadmcli power disk 16-259
odaadmcli show controller 16-260
odaadmcli show disk 16-261
odaadmcli show diskgroup 16-262
odaadmcli show raidsyncstatus 16-263
odaadmcli show storage 16-264
odaadmcli stordiag 16-265

17 Oracle Database Appliance Command-Line Interface for KVM

About Command-Line Interface for Kernel-Based Virtual Machine (KVM) 17-1
About ODACLI Commands Supported on Oracle KVM Database System 17-3
odacli DB System Commands for KVM 17-5
odacli create-dbsystem 17-6
odacli modify-dbsystem 17-7
odacli modify-dbfileattributes 17-10
odacli list-dbsystems 17-11
odacli describe-dbsystem 17-11
odacli describe-dbsystem-image 17-18
odacli start-dbsystem 17-19
odacli stop-dbsystem 17-20
odacli delete-dbsystem 17-21
odacli VM Storage Commands for KVM 17-22
odacli create-vmstorage 17-23
odacli list-vmstorages 17-23
odacli describe-vmstorage 17-24
odacli modify-vmstorage 17-24
odacli delete-vmstorage 17-25
odacli Virtual Disk Commands for Oracle KVM 17-25
odacli create-vdisk 17-26
odacli list-vdisks 17-26
odacli describe-vdisk 17-27
odacli modify-vdisk 17-27
odacli clone-vdisk 17-28
odacli delete-vdisk 17-28
odacli CPU Pool Commands for Bare Metal and KVM 17-29
odacli create-cpupool 17-30
odacli remap-cpupools 17-31
odacli list-cpupools 17-32
odacli describe-cpupool 17-32
odacli modify-cpupool 17-33

odacli delete-cpupool 17-33
odacli list-auditrecords 17-34
odacli describe-auditrecord 17-35
odacli VM Network Commands for KVM 17-35
odacli create-vnetwork 17-36
odacli list-vnetworks 17-37
odacli describe-vnetwork 17-37
odacli start-vnetwork 17-38
odacli stop-vnetwork 17-38
odacli modify-vnetwork 17-39
odacli delete-vnetwork 17-39
odacli Virtual Machine Commands for Oracle KVM 17-40
odacli create-vm 17-40
odacli list-vms 17-42
odacli describe-vm 17-42
odacli modify-vm 17-43
odacli start-vm 17-45
odacli stop-vm 17-46
odacli migrate-vm 17-46
odacli clone-vm 17-47
odacli delete-vm 17-47

18 Configuring and Using Oracle Auto Service Request (Oracle ASR)

About Configuring Oracle ASR 18-1
Configuring Internal Oracle ASR 18-3
Registering with External Oracle ASR Manager 18-5
Configuring External Oracle ASR When Oracle ASR Manager Does Not Run on an Oracle Database Appliance 18-7
Testing Oracle ASR 18-11
Managing an Oracle ASR Configuration 18-12
Additional Resources for Oracle ASR 18-13

19 Troubleshooting Oracle Database Appliance

About Setting Up and Removing SSH Configuration 19-2
Viewing Oracle Database Appliance Error Correlation Reports 19-2
About Enabling Linux Kernel Core Extractor for Troubleshooting 19-3
Viewing Details About DCS Error Messages 19-5
Collecting Diagnostics Data Using the BUI 19-6
Resolving Errors When Updating DCS Components During Patching 19-6
Viewing Component Information on the Appliance 19-15

Errors When Logging into the Browser User Interface 19-18
Errors when re-imaging Oracle Database Appliance 19-19
Using Oracle Autonomous Health Framework for Running Diagnostics 19-19
About Installing Oracle Autonomous Health Framework 19-20
Using the Oracle ORAchk Health Check Tool 19-21
Generating and Viewing Oracle ORAchk Health Check Tool Reports in the Browser User Interface 19-22
Generating and Viewing Database Security Assessment Reports in the Browser User Interface 19-22
Running Oracle Trace File Analyzer (TFA) Collector Commands 19-23
Sanitizing Sensitive Information in Diagnostic Collections 19-25
Enabling Adaptive Classification and Redaction (ACR) 19-27
Sanitizing Sensitive Information in Oracle Trace File Analyzer Collections 19-28
Redacting and Sanitizing Entities in the BUI 19-28
Sanitizing Sensitive Information in Oracle ORAchk Output 19-29
Running the Disk Diagnostic Tool 19-31
Running the Oracle Database Appliance Hardware Monitoring Tool 19-31
Disabling the Browser User Interface 19-32
Preparing Log Files for Oracle Support Services 19-32

A Oracle Database Appliance Software Configuration Defaults

Directory Paths for Oracle Database Appliance A-1
Port Numbers and Protocols of Oracle Database Appliance Components A-2
Location of Log Files A-4
Oracle Groups and User Configurations for Oracle Database Appliance A-5

B Oracle Database Appliance Storage Defaults

About Database File Storage B-1
Oracle ACFS Mount Points and Storage Space B-2
Displaying Mounted Disk Details B-3
Usable Space on Oracle Database Appliance X9-2S and X9-2L B-4
Usable Space on Oracle Database Appliance X9-2-HA B-5
Usable Free Space with Oracle ASM B-6

C Creating an Appliance Using JSON File

Loading and Saving Appliance Configurations C-1
Readme for the Command odacli create-appliance C-1
Example JSON Files to Create a Single Node Appliance with the CLI C-2
Example JSON File to Create a High-Availability Appliance with the CLI C-7

D Database Shapes for Oracle Database Appliance

About Database Shapes and Classes D-1
Database Shapes for the OLTP Class D-2
Database Shapes for the In-Memory Class D-4
Database Shapes for the DSS Class D-5

Index

Preface

Oracle Database Appliance is an optimized, prebuilt database system that is easy to deploy, operate, and manage. By integrating hardware and software, Oracle Database Appliance eliminates the complexities of nonintegrated, manually assembled solutions. Oracle Database Appliance reduces the installation and software deployment times from weeks or months to just a few hours while preventing configuration and setup errors that often result in suboptimal, hard-to-manage database environments.

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This guide is intended for anyone who configures, maintains, or uses Oracle Database Appliance:

- System administrators
- Network administrators
- Database administrators
- Application administrators and users

This book does not include information about Oracle Database architecture, tools, management, or application development that is covered in the main body of Oracle Documentation, unless the information provided is specific to Oracle Database Appliance. Users of Oracle Database Appliance software are expected to have the same skills as users of any other Linux-based Oracle Database installations.

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 customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information about Oracle Database Appliance, go to `http://www.oracle.com/goto/oda/docs` and click the appropriate release.

For more information about using Oracle Database, go to `http://docs.oracle.com/database/` and select the database release from the menu.

For more information about Oracle Integrated Lights Out Manager 3.2, see https://docs.oracle.com/cd/E37444_01/.

For more details about other Oracle products that are mentioned in Oracle Database Appliance documentation, see the Oracle Documentation home page at `http://docs.oracle.com`.

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 the text.
italic	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.
# prompt	The pound (#) prompt indicates a command that is run as the root user.

1

Oracle Database Appliance Checklists

Use these checklists to ensure you have completed the tasks required for setting up, deploying, and patching Oracle Database Appliance.

- Checklist for System Requirements
  Before you begin the deployment, ensure that you have the system information described in this checklist ready. This checklist applies to baremetal and virtualized platform deployments.
- Deployment Checklist for Oracle Database Appliance Bare Metal Systems
  This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.
- Checklist for Custom Network Address Configuration
  Use the checklist to identify the IP addresses required for Oracle Database Appliance.

Checklist for System Requirements

Before you begin the deployment, ensure that you have the system information described in this checklist ready. This checklist applies to baremetal and virtualized platform deployments.

Table 1-1 Checklist for System Configuration Information for Oracle Database Appliance

Component	Information Required	Comments
Host Name	The name for the Oracle Database Appliance System. The name must conform with the RFC 952 standard, which allows alphanumeric characters and hyphens (-), but does not allow underscores (_). The name should not begin with a numeral or hyphen and should not end in a hyphen. Oracle recommends that you use all lowercase characters for the host name.	Use this space to note the values for your appliance. Host Name:
Domain Name	Your domain name. For example: `example.com`	Domain Name:
Region	The region where you plan to operate the Oracle Database Appliance system.	Region:
Timezone	The time zone where you plan to operate the Oracle Database Appliance system.	Timezone:

Table 1-1 (Cont.) Checklist for System Configuration Information for Oracle Database Appliance

Component	Information Required	Comments
Diskgroup Redundancy	Determine the redundancy level for DATA, RECO, and FLASH: If there are up to two disks, then you can select the disk group redundancy as Normal or Flex. If there are more than two disk drives, then you can select the redundancy as Normal (two way mirror), High (three way mirror), or Flex. If you select High redundancy, then DATA, RECO, and FLASH are all High Redundancy.	Use this space to note the values for your appliance.
Number of Enabled CPU Cores	Based on your capacity-on-demand (CoD) license, this value specifies the number of enabled CPU cores	Use this space to note the values for your appliance.
Backup	Determine the backup location. For external backup, you need an NFS device.	Backup location:
Percentage of Storage Reserved for Data	Determine the amount of reserves for DATA storage. The percentage must be a whole number between 10 and 90, and determines how the disks are partitioned between DATA and RECO. For example, if you specify 80, then 80% of storage is reserved for DATA and the remaining 20% is for RECO.	Use this space to note the values for your appliance.
System Password	The password set for the root user of the system, operating system users, database users, and pdbadmin. The password is also used to set the database `SYS` and `SYSTEM` passwords. Ensure that the password you provide is in compliance with common security practices.	You must provide a system password when you deploy your appliance.
DNS Server	(Optional) DNS server details. If configuring DNS, a minimum of one DNS Server IP is required.	DNS Server details:
NTP Server	(Optional) Network Time Protocol (NTP) server details.	NTP Server details:

Table 1-1 (Cont.) Checklist for System Configuration Information for Oracle Database Appliance

Component	Information Required	Comments
Network Information for the client access network	Obtain the following network information: • Node 0 Name: such as `example0` • Node 0 Public IP Address • Node 1 Name (For HA Models): such as `example1` • Node 1 Public IP Address (For HA Models) • Node 0 Virtual IP Name (For HA Models): for example, `example0-vip` • Node 0 Public Virtual IP Address (For HA Models) • Node 1 Virtual IP Name (For HA Models): for example, `example1-vip` • Node 1 Public Virtual IP Address (For HA Models) • SCAN Name (For HA Models): for example, `example-scan` • SCAN IP Addresses: Two SCAN IP addresses are required • Netmask • Gateway IP The Public and private network must not be in same subnet, they must be in non-related separate subnet.	Use this space to note the values for your appliance.
Network Information for the Oracle Integrated Lights Out Manager (ILOM) network	Obtain the following ILOM network information: • ILOM host name • ILOM IP address • Subnet mask • Gateway IP • ILOM Password: default is `changeme`. For HA systems, ensure that you have the information to configure two ILOM networks.	Use this space to note the values for your appliance.

Table 1-1 (Cont.) Checklist for System Configuration Information for Oracle Database Appliance

Component	Information Required	Comments
User and group information	Determine how you want to configure your users and groups and whether or not you want to allow operating system role separation. The default is two users with six groups. Operating System Role Separation: Yes/No Customization of Users and Groups: Yes/No Groups and IDs: • Grid Infrastructure User • Grid Infrastructure User ID • Database User • Database User ID • Install Group • Install Group ID • DBA Operating System Group • DBA Operating System Group ID • ASM Admin Group • ASM Admin Group ID • ASM Operating System Group • ASM Operating System Group ID • ASM DBA Group • ASM DBA Group ID	Use this space to note the values for your appliance.

Table 1-1 (Cont.) Checklist for System Configuration Information for Oracle Database Appliance

Component	Information Required	Comments
Initial Database Details (if you want to create one during deployment)	(Optional) If you choose to create an initial database during deployment, determine the following configuration details: • Database name • Database unique name • Database version (based on the database clones that are registered with the Oracle Database Appliance) • Determine the Oracle Database edition licensing that you have, either Enterprise Edition or Standard Edition. You cannot mix editions. • Database deployment, whether Oracle RAC or single-instance database • CDB (Container Database): Yes or No • PDB name • Shape (for example, odb1, odb2, and so on) • Database Class: Enterprise Edition, OLTP, DSS, or IMDB. Standard Edition OLTP only. • Data files on Flash storage: Yes/No • Database file size • Database characterset • National characterset • Database language • Storage (Oracle ASM or Oracle ACFS). • Database redundancy • Configure Oracle Enterprise Manager console. If you are using Enterprise Manager Cloud Control with this appliance, do not choose this option. • Password for the database The DB Name and DB Unique name can contain alphanumeric characters and must start with an alphabet. It can contain underscores (_), but cannot contain characters such as "! @%^&*()+=\\|`~[{]};:'\",<>/?".	Use this space to note the values for your appliance.

Table 1-1 (Cont.) Checklist for System Configuration Information for Oracle Database Appliance

Component	Information Required	Comments
Oracle Auto Service Request (Oracle ASR) information	(Optional) Do you want to configure and enable Oracle ASR at deployment or later? If required, then ensure that you have the following information. • ASR Type • ASR User Name • ASR Password • SNMP Version • Proxy Server Name • Proxy Port • Proxy User Name • Proxy Password	Oracle Auto Service Request (Oracle ASR) information for your appliance

Deployment Checklist for Oracle Database Appliance Bare Metal Systems

This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.

Table 1-2 Deployment Checklist for Bare Metal Systems

Component	Tasks
Register your hardware Support Identifier (SI)	Add your hardware Support Identifier (SI) to your My Oracle Support account profile. Your hardware SI is supplied when you purchase Oracle Database Appliance. If you acquire new software licenses, then you must also register your new software SIs. The SI registration process can take up to 24 hours to complete.
Plug in the Appliance	• Plug in all the cables including the power cords, and then wait for the SP OK light to become solid green. Do not power up yet. • If there is no video port available on your Oracle Database Appliance hardware model, then use the Server Management Serial Port to connect to a laptop, or determine the DHCP address assigned to it. **Note:** Serial port must be set to 115200, so make sure the baud rate speed matches.
Configure Oracle Integrated Lights Out Manager (Oracle ILOM) on `Node0` and `Node1`	Before you configure Oracle ILOM: • Connect an Ethernet cable from the NET MGT port to your site's management network. The NET MGT port is a 1 GbE port for Oracle ILOM and by default, is configured for DHCP. • Obtain the Oracle ILOM host name and IP address, the netmask, and the gateway addresses from the DHCP server. • Have a new password ready to replace the default Oracle ILOM password Log into the Oracle ILOM console and follow the procedure in the topic *Configuring Oracle Integrated Lights Out Manager*.

Table 1-2 (Cont.) Deployment Checklist for Bare Metal Systems

Component	Tasks
Power ON	• Power on each node and log in as `root` user with `welcome1` password. • Verify cabling: `# /opt/oracle/dcs/bin/odacli validate-storagetopology` • Verify interconnect networking. Run the commands `# ethtool p1p1` and `# ethtool p1p2`. Follow the steps described in the topic *Verifying Cabling and Network Connections*.
Plumb the network	Ensure that you have recorded the IP address and netmask address. Set up a temporary network connection to Oracle Database Appliance. Use the command `odacli configure-firstnet` on Node0, for both single- and multi-node systems. Log into the Oracle ILOM console and follow the procedure in the topic *Plumbing the Network*.
Download Oracle Database Appliance Software	Log into My Oracle Support and download the software files for deployment. Refer to the *Oracle Database Appliance Release Notes* for the latest software.
Install Oracle Database Appliance	Upload the zip files to a temporary location on Oracle Database Appliance and then update the repository and install the software. Follow the steps and run the commands described in the topic *Installing Oracle Database Appliance Software*.
Create Appliance	• Set the Browser User Interface password for the `oda-admin` user. • Log into the Browser User Interface and configure system, network, and users and groups. Ensure that you have the information mentioned in *Checklist for System Requirements* available to you. Follow the steps described in the topic *Creating an Appliance*.
Perform postinstallation tasks	Configure CPU core count. Change the passwords for Browser User Interface, Oracle users and groups, and Oracle ILOM. Follow the steps described in the chapter *Oracle Database Appliance Postinstallation Tasks*.
Cleanup Deployments	If your deployment failed for some reason, then run the cleanup script and redeploy the appliance. For steps to run the cleanup script, see the topic *Uninstalling Oracle Database Appliance Components*.

Related Topics

- Verifying Cabling and Network Connections
  After powering on Oracle Database Appliance, verify that the network connections are set up correctly.
- Configuring Oracle Integrated Lights Out Manager
  Configure Oracle Integrated Lights Out Manager (Oracle ILOM) to manage Oracle Database Appliance independent of the operating system.
- Installing Oracle Database Appliance Software
  Install Oracle Database Appliance software, before creating the appliance.

- Creating the Appliance
  Create the appliance using the Browser User Interface or ODACLI commands.
- Oracle Database Appliance Postinstallation Tasks
  Complete these administrative tasks after you have deployed software, but before the system is operational.
- Uninstalling Oracle Database Appliance Components
  Use the Oracle Database Appliance cleanup deploy utility, `/opt/oracle/oak/onecmd/cleanup.pl` to uninstall Oracle Database Appliance components.

Checklist for Custom Network Address Configuration

Use the checklist to identify the IP addresses required for Oracle Database Appliance.

> **Note:**
>
> Oracle does not recommend changing the default Host Private Address. You cannot change the private network after deploying the appliance.

Table 1-3 Default IP Address Requirements for Oracle Database Appliance

Type of IP	IP Address Default Values	Your Values As Applicable
Client Access Network	No default	No default
Additional Network	No default	No default
Oracle Integrated Lights Out Manager (ILOM)	No default	No default
Host Private Addresses	InfiniBand has bonded interface `ibbond0` . The system configures `icbond0` on non-InfiniBand High-Availability systems. The subnets 192.168.16.0/24 and 192.168.17.0/24 are configured.	Not applicable: the private addresses are defined before deployment and should not be changed

2

Introduction to Oracle Database Appliance

Oracle Database Appliance saves time and money by simplifying deployment, maintenance, and support of database solutions for organizations of every size. Oracle Database Appliance hardware models are optimized to run Oracle Database Standard Edition and Enterprise Edition.

- About Oracle Database Appliance
  Understand Oracle Database Appliance hardware models, deployment plans, and database options.
- Oracle Database Appliance Software Terminology
  Understand the software patches available for Oracle Database Appliance.
- About the Browser User Interface
  Use the Oracle Appliance Manager Browser User Interface to deploy and manage the appliance, databases, networks, jobs, CPU pools,compute instances, and database systems.

About Oracle Database Appliance

Understand Oracle Database Appliance hardware models, deployment plans, and database options.

About Oracle Database Appliance Hardware Models

Choose the Oracle Database Appliance hardware model that suits your business needs.

Oracle Database Appliance Hardware Model	Deployment Use Case
Oracle Database Appliance X9-2S, X9-2L, X9-2-HA	• Oracle Database Appliance X9-2S is a small configuration designed for smaller or entry-level deployments. • Oracle Database Appliance X9-2L is a large configuration designed for larger databases. • Oracle Database Appliance X9-2-HA is a large configuration designed for larger databases and high-availability.
Oracle Database Appliance X8-2S, X8-2M, X8-2-HA	• Oracle Database Appliance X8-2S is a small configuration designed for smaller or entry-level deployments. • Oracle Database Appliance X8-2M is a medium-sized configuration designed for performance. • Oracle Database Appliance X8-2-HA is a large configuration designed for larger databases and high-availability.

Oracle Database Appliance Hardware Model	Deployment Use Case
Oracle Database Appliance X7-2S, X7-2M, X7-2-HA	• Oracle Database Appliance X7-2S is a small configuration designed for smaller or entry-level deployments. • Oracle Database Appliance X7-2M is a medium-sized configuration designed for performance. • Oracle Database Appliance X7-2-HA is a large configuration designed for larger databases and high-availability.
Oracle Database Appliance X6-2S, X6-2M, X6-2L	• Oracle Database Appliance X6-2S is a small configuration designed for smaller or entry-level deployments. • Oracle Database Appliance X6-2M is a medium-sized configuration designed for performance. • Oracle Database Appliance X6-2L is a large configuration designed for larger databases.
Oracle Database Appliance X6-2-HA	Highly-available two-node Oracle RAC clusters that can support both Oracle RAC and single-instance databases.
Oracle Database Appliance X5-2	Highly-available two-node Oracle RAC clusters that can support both Oracle RAC and single-instance databases.

You cannot expand or reconfigure Oracle Database Appliance to a different configuration. For example, you cannot expand Oracle Database Appliance X7-S to Oracle Database Appliance X7-2M. For Oracle Database Appliance hardware configuration details, see the *Oracle Database Appliance Owner's Guide*.

About Oracle Database Appliance Deployment Plans

Oracle Database Appliance X8-2S and X8-2M are single node systems that you deploy as a bare metal platform.

Oracle Database Appliance X8-2-HA is a two-node system with the option to deploy as a bare metal platform or a virtualized platform.

Oracle Database Appliance X7-2S and X7-2M are single node systems that you deploy as a bare metal platform.

Oracle Database Appliance X7-2-HA is a two-node system with the option to deploy as a bare metal platform or a virtualized platform.

Oracle Database Appliance X6-2S, X6-2M, X6-2L are single node systems that you deploy as a bare metal platform.

Oracle Database Appliance X6-2-HA is a two-node system with the option to deploy as a bare metal platform or a virtualized platform.

Oracle Database Appliance X5-2 is a two-node system with the option to deploy as a bare metal platform or a virtualized platform.

About Database Deployment Options for Oracle Database Appliance

The following Oracle Database editions are available:

Database Edition	Supported Releases
Oracle Database Enterprise Edition	Oracle Database release 19c, 18c, 12.2.0.1, and 12.1.0.2
Oracle Database Standard Edition 2	Oracle Database release 19c, 18c, 12.2.0.1, and 12.1.0.2

Note:

It is important to understand your Oracle Database licensing before you create databases. You cannot provision Oracle Database Enterprise Edition and Standard Edition databases in the same bare metal platform, or the same ODA_BASE virtualized deployment, subject to Oracle Database licensing requirements. Review the *Oracle Database Licensing Information User Manual* for the Oracle Database release to review supported options and products.

Oracle Database Appliance Software Terminology

Understand the software patches available for Oracle Database Appliance.

The table describes the Oracle Database Appliance patches that are available for download and the purpose of the patch.

Note:

Some hardware models may not use the patches listed in the table.

Table 2-1 Software for Oracle Database Appliance

Patch	Description
Oracle Database Appliance patch bundle (quarterly patch release)	Use the patch bundle to update to the latest release after Oracle Database Appliance is deployed. The patch updates the Oracle Database Appliance server, storage, and database components.
Oracle Database Appliance Grid Infrastructure/Database Clone	Use to perform an initial deployment of Oracle Database Appliance and also to patch your deployment to the latest release. The bundle contains the latest Grid Infrastructure and Database components, Oracle Database Appliance Manager software, Oracle Linux and hardware drivers for deployment on an Oracle Database Appliance that is shipped from factory state. If you reimage Oracle Database Appliance with the Bare Metal ISO Image, download the Grid Infrastructure/Database Clone patch to deploy Oracle Database Appliance.
Oracle Database Clone	Use the Database Software Clone files to get the latest Database clone binaries for Oracle Database releases 19c, 12.2.0.1, and 12.1.0.2. The RDBMS clone files are needed to create Oracle databases and database homes.
Bare Metal ISO Image	Use to reimage the appliance and reset the system back to the factory configuration.

> **Note:**
>
> See the *Oracle Database Appliance Release Notes* for the patch numbers and download locations and the component versions available in the patch bundle.

> **Note:**
>
> Starting with Oracle Database 18c, Oracle provides quarterly updates in the form of Release Updates (Updates) and Release Update Revisions (Revisions). Oracle no longer releases patch sets. For more information, see My Oracle Support Note 2285040.1.
>
> Oracle Database releases are of type `version` and `version_full` releases. The `version` release is designated in the form major release version.0.0.0.0. The major release version is based on the last two digits of the year in which an Oracle Database version is released for the first time. The `version_full` release is an update of a version release and is designated based on the major release version, the quarterly release update version (Update), and the quarterly release update revision version (Revision).
>
> Prior to Oracle Database release 18c, changing the first or second digit in the release number was an upgrade, and changing the fourth digit indicated an update. With 18c and later releases, changing the first digit is an upgrade, and changing the second indicates an update.

About the Browser User Interface

Use the Oracle Appliance Manager Browser User Interface to deploy and manage the appliance, databases, networks, jobs, CPU pools,compute instances, and database systems.

The Browser User Interface provides a user-friendly interface to quickly view and perform many of the tasks needed to deploy, patch, and manage your bare metal system. The *Oracle Database Appliance Deployment and User's Guide* for your hardware model describes how to use the Browser User Interface to perform many of the tasks related to deployment, patching, and management of your appliance.

Use the links in the side navigation to view and manage the following:

- **Appliance**
  - Appliance information
  - System information
  - Disk group information including storage utilization such as free and used space
  - Information about all databases that use a storage disk group
  - All the patches and hardware and software components installed on the appliance
  - List of RPMs and RPM Drift information
- **Network**
  - View network details
  - Create, update, and delete networks

  - View interface details
- **Oracle ASR**
  - View Oracle Auto Service Request (Oracle ASR) configuration details
  - Configure, modify, test, and delete Oracle ASR
- **Repository Manager**
  - Update the patch repository with Oracle Database Appliance patches
  - View component details of the patch and versions that are installed on the appliance
  - View component details of the patch and versions that are available, but not yet installed, on the appliance
  - Run patch prechecks before applying the patch
  - Update the server and storage components
- **Compute Instances**
  - Create KVM deployment on Oracle Database Appliance
  - Create view, modify, and, delete virtual storage, virtul network, and virtual disks
  - Create view, modify, and, delete, start, and stop virtual machines
- **DB Systems**
  - Create DB System
  - List and describe DB System
  - Modify, delete, start, and stop DB System
- **CPU Pools**
  - Create view, modify, and, delete CPU pool for bare metal and KVM deployments

Use the links in the top navigation to view and manage the following:

- **Database**
  - Create a database
  - View database details
  - Apply a backup policy for the database
  - Create a manual backup
  - Recover a database from a backup
  - View and update the database backup schedule and update the archive log backup schedule for the database
  - Modify associated networks, database shape and class, and the associated backup policy for a database
  - Create, update, and delete backup policies
  - Create, update, and delete database homes
  - Create and delete database backup
  - Clone a database from a database backup or an Oracle ACFS database snapshot
  - Patch database homes
  - Move databases from one Oracle home to another

- **Credential Store**
  - Create Object Store Credentials for backup and recovery
  - View the details for Object Store Credentials, including the endpoint URL, user name, and tenant
  - Update and delete Object Store Credentials
  - Configure Oracle Key Vault Server
- **Monitoring**
  - View the status of hardware, storage, memory, cooling, networking, and power
  - View monitoring information for each node, in the case of High-Availability deployments
  - View system health status and drilldown to details of incidents
  - Click the hardware image to view details
- **Security**
  - View audit records and create and view database security assessment reports
- **Activity**
  - View a list of jobs, details, and status
  - Click a job description to view job details

Expand **Resources** in the upper right corner of the Browser User Interface to perform the following tasks:

- There are links to access the Oracle Database Appliance blogs, FAQs, and Documentation.
- Click **About** to view the Oracle Database Appliance release that is installed on the appliance.

Expand **Account** in the upper right corner of the Browser User Interface to perform the following tasks:

- Click **ILOM Setting** to change the root password. To enable first-time login and access to Oracle ILOM, a default Administrator (root) account and its password are provided with the system.
- Click **Change Password** to change the `oda-admin` password.
- Click **Sign Out** to sign out of the Browser User Interface. If you do not sign out, the system automatically signs you out after a period of inactivity. A warning will appear 2 minutes before the session expires.

Doc Search

You can access the Oracle Database Appliance documentation set from the Browser User Interface. There is a search box at the top right hand corner on the BUI. Search results are links to documentation pages shown as a new window in the BUI. When you click the **Help** button, links from the documentation relevant to the context of the tab are displayed.

Along with the search results from the Oracle Database Appliance documentation pages, BUI also provides relevant Frequently Asked Questions (FAQs) for the search query. Additionally, you can also search on DCS error codes such as DCS-10001, DCS-10032, and so on in the search box and get the links to documentation pages contains these error codes. When you specify the search query, relevant documentation links, FAQs and DCS error codes links are displayed under separate sections in a new window in the BUI.

For the latest updates to the documentation for a release, see the online Oracle Database Appliance documentation library at https://docs.oracle.com/en/engineered-systems/oracle-database-appliance/index.html.

3

Readying Oracle Database Appliance for Deployment

Before deploying Oracle Database Appliance, perform these setup tasks.

- Attaching Storage Cables to Oracle Database Appliance X9-2-HA
  Connect the storage cables for Oracle Database Appliance X9-2-HA.
- Attaching Network Cables to Oracle Database Appliance X9-2-HA
  Connect Oracle Database Appliance X9-2-HA to either a 10GBase-T (copper) or 25GbE SFP28 (fiber) network.
- Attaching Network Cables to Oracle Database Appliance X9-2S or X9-2L
  Connect Oracle Database Appliance X9-2S or X9-2L to either a 10GBase-T (copper) or 25GbE SFP28 (fiber) network.
- Attaching Power Cords and Initializing Components
  Attach power cords for Oracle Database Appliance.
- Configuring Oracle Integrated Lights Out Manager
  Configure Oracle Integrated Lights Out Manager (Oracle ILOM) to manage Oracle Database Appliance independent of the operating system.
- Powering On Oracle Database Appliance the First Time
  Use this procedure the first time you power on Oracle Database Appliance.

Attaching Storage Cables to Oracle Database Appliance X9-2-HA

Connect the storage cables for Oracle Database Appliance X9-2-HA.

Cabling the Storage for Oracle Database Appliance X9-2-HA

Connect the storage cables for Oracle Database Appliance X9-2-HA.

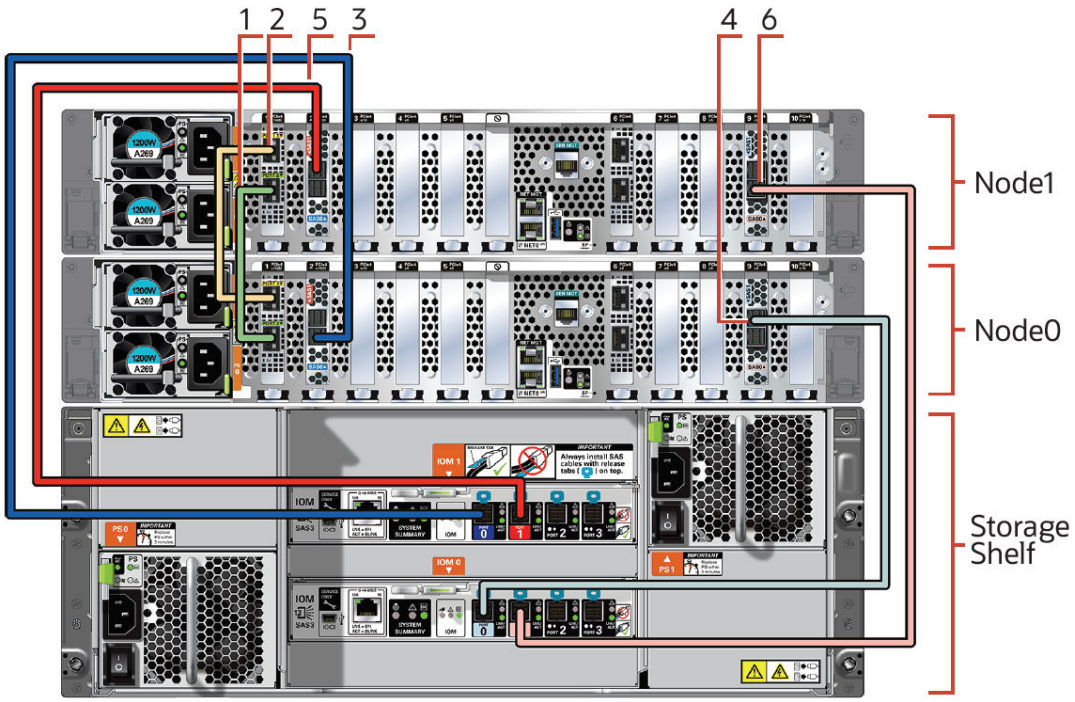


Table 3-1 Storage Cabling for Oracle Database Appliance X9-2-HA

Callout Number	Purpose	Start - Compute Node0	End - Compute Node1
1	Connect green SFP28 cable (Interconnect Port 25GbE)	Connect into green port (PORT 2) in PCIe slot 1	Connect into green port (PORT 2) in PCIe slot 1
2	Connect yellow SFP28 cable (Interconnect Port 25GbE)	Connect into yellow port (PORT 1) in PCIe slot 1	Connect into yellow port (PORT 1) in PCIe slot 1
3	Connect dark blue SAS cable	Connect into dark blue port (SAS0) in PCIe slot 2 in Node0	Connect into dark blue port in top IO Module (PORT 0)
4	Connect light blue SAS cable	Connect into light blue port (SAS1) in PCIe slot 9 in Node0	Connect into light blue port in bottom IO Module (PORT 0)
5	Connect dark red SAS cable	Connect into dark red port (SAS1) in PCIe slot 2 in Node1	Connect into dark red port in top IO Module (PORT 1)
6	Connect light red SAS cable	Connect into light red port (SAS0) in PCIe slot 9 in Node1	Connect into light red port in bottom IO Module (PORT 1)

Cabling the Storage Expansion Shelf

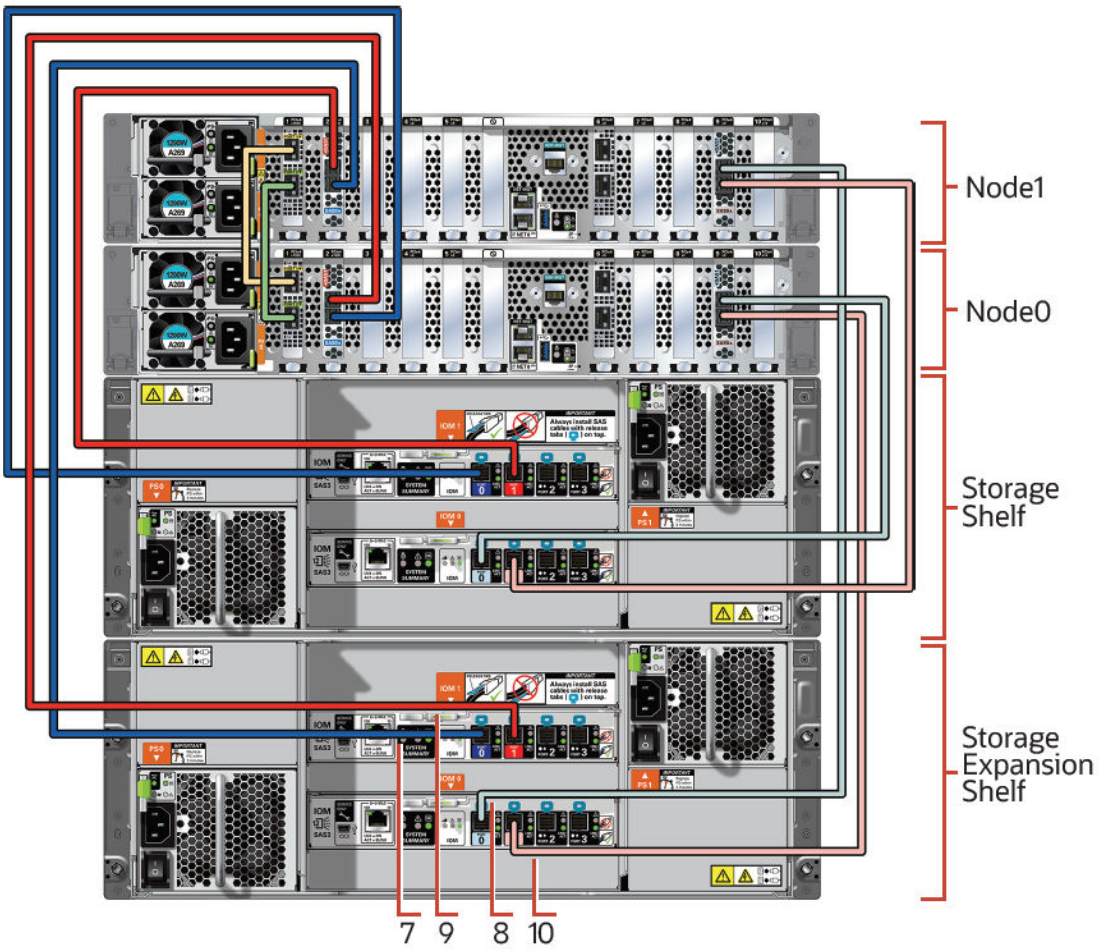


Table 3-2 Cabling the Storage Expansion Shelf for Oracle Database Appliance X9-2-HA

Callout Number	Purpose	Start - Compute Nodes	End - Expansion Shelf
7	Connect dark blue SAS cable	Connect into dark blue port (SAS0) in PCIe slot 2 in Node1	Connect into dark blue port in top IO Module (PORT 0)
8	Connect light blue SAS cable	Connect into light blue port (SAS1) in PCIe slot 9 in Node1	Connect into light blue port in bottom IO Module (PORT 0)
9	Connect dark red SAS cable	Connect into dark red port (SAS1) in PCIe slot 2 in Node0	Connect into dark red port in top IO Module (PORT 1)
10	Connect light red SAS cable	Connect into light red port (SAS0) in PCIe slot 9 in Node0	Connect into light red port in bottom IO Module (PORT 1)

Attaching Network Cables to Oracle Database Appliance X9-2-HA

Connect Oracle Database Appliance X9-2-HA to either a 10GBase-T (copper) or 25GbE SFP28 (fiber) network.

The SFP28 ports support 10Gb and 25Gb fibre and twinaxial cable (twinax), depending on the SFP modules used in these ports. For 10GBase-T public networking, use the Cat-6 network cables. To use the SFP28 ports, use the correct transceivers and cables to meet your switch bandwidth. The following sections show the cabling options for 10GBase and 25GbE SFP28 (fiber) network ports.

Connecting the Fiber and Copper Network Cables for Oracle Database Appliance X9-2-HA

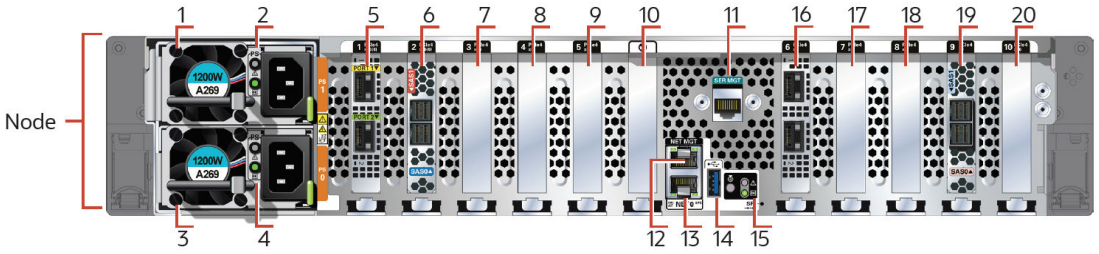


Table 3-3 Network Ports and Power Cabling

Callout	Description
1	Power Supply (PS) 1 with fan module
2	Power Supply (PS) 1 status indicators: Service Required LED: amber, AC OK LED: green
3	Power Supply (PS) 0 with fan module
4	Power Supply (PS) 0 status indicators: Service Required LED: amber, AC OK LED: green
5	PCIe card slot 1: Oracle Dual Port 25Gb Ethernet Adapter provides two ports with SFP28 connections for a private cluster interconnect between server nodes
6	PCIe card slot 2: provides two SAS3 connectors used to connect the servers to the storage shelf and the storage expansion shelf
7	PCIe card slot 3: filler panel
8	PCI card slots 4: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
9	PCIe card slot 5: filler panel
10	Not used
11	SER MGT port: RJ-45 serial port used to connect to the Oracle ILOM service processor
12	NET MGT port: 10/100/1000Base-T network interface port with RJ-45 connector used to connect to the Oracle ILOM service processor

Table 3-3 (Cont.) Network Ports and Power Cabling

Callout	Description
13	100/1000Base-T network interface port with RJ-45 connector: NET 0
14	USB 3.0 connector
15	System status indicators: Locate LED (white), Service Required LED (amber), Power/OK LED (green)
16	PCIe card slot 6: Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
17	PCIe card slot 7: filler panel
18	PCIe card slot 8: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
19	PCIe card slot 9: provides two SAS3 connectors used to connect the servers to the storage shelf and the storage expansion shelf
20	PCIe card slot 10: filler panel

Fiber Cables

For optical cables, you must purchase either Short Range (SR) or Long Range (LR) SFP transceivers for each of the network ports, and then plug in the appropriate optical cable. Currently, Oracle sells both the SR and LR SFP transceivers. In addition to these transceivers, you must purchase the appropriate LC-LC terminated fiber optic cables from a third-party vendor.

Table 3-4 Transceivers for the SFP Ports - 10Gb/1Gb

Name	ATO Part Number	x-option Part Number
SFP+ 10GBase-SR (Short Range Transceiver)	2129a	X2129A-N
SFP+ 10GBase-LR (Long Range Transceiver)	5562a-z	X5562A-Z

Table 3-5 Transceivers for the SFP Ports - 25Gb

Name	ATO Part Number	x-option Part Number
SFP28 25GBase-SR (Short Range Transceiver)	7118017	7118019
SFP28 25GBase-LR (Long Range Transceiver)	7118020	7118021

Copper Cables

You can purchase the following copper cables from Oracle. These cables have built-in SFP connectors.

Table 3-6 Networking Cables for the SFP Ports - 10Gb/1Gb

Name	Length	ATO Part Number	x-option Part Number
TwinAx passive copper cable: 1 meter	1m	7105137	X2130A-1M-N
TwinAx passive copper cable: 2 meter	2m	7105139	7105148
TwinAx passive copper cable: 3 meter	3m	7105140	X2130A-3M-N
TwinAx passive copper cable: 5 meter	5m	7105141	X2130A-5M-N

Table 3-7 Networking Cables for the SFP Ports - 25Gb

Name	Length	ATO Part Number	x-option Part Number
TwinAx passive copper cable: 1 meter	1m	7118359	7118367
TwinAx passive copper cable: 2 meter	2m	7118360	7118368
TwinAx passive copper cable: 3 meter	3m	7118361	7118369
TwinAx passive copper cable: 5 meter	5m	7118362	7118370

Attaching Network Cables to Oracle Database Appliance X9-2S or X9-2L

Connect Oracle Database Appliance X9-2S or X9-2L to either a 10GBase-T (copper) or 25GbE SFP28 (fiber) network.

The SFP28 ports support 10Gb and 25Gb fibre and twinaxial cable (twinax), depending on the SFP modules used in these ports. For 10GBase-T public networking ports that have RJ-45 connectors, use Cat-6 network cables. To use the SFP28 ports, use the correct transceivers and cables to meet your switch bandwidth. The following sections show the cabling options for 10GBase and 25GbE SFP28 (fiber) network ports.

Connecting Network Cables for Oracle Database Appliance X9-2S or X9-2L

Connect the Fiber and Copper Network Cables for Oracle Database Appliance X9-2S or X9-2L

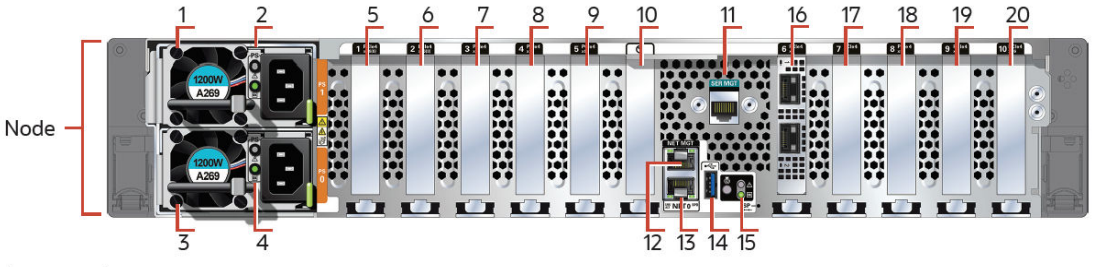


Table 3-8 Network Ports and Power Cabling

Callout	Description
1	Power Supply (PS) 1 with fan module
2	Power Supply (PS) 1 status indicators: Service Required LED: amber, AC OK LED: green
3	Power Supply (PS) 0 with fan module
4	Power Supply (PS) 0 status indicators: Service Required LED: amber, AC OK LED: green
5	PCIe card slot 1: filler panel
6	PCIe card slot 2: filler panel
7	PCIe card slot 3: filler panel
8	PCIe card slot 4: • Oracle Database Appliance X9-2S: filler panel • Oracle Database Appliance X9-2L: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
9	PCIe card slot 5: filler panel
10	Not used
11	SER MGT port: RJ-45 serial port used to connect to the Oracle ILOM service processor
12	NET MGT port: 10/100/1000Base-T network interface port with RJ-45 connector used to connect to the Oracle ILOM service processor
13	100/1000Base-T network interface port with RJ-45 connector: NET 0
14	USB 3.0 connector
15	System status indicators: Locate LED (white), Service Required LED (amber), Power/OK LED (green)
16	PCIe card slot 6: Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
17	PCIe card slot 7: • Oracle Database Appliance X9-2S: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter • Oracle Database Appliance X9-2L: filler panel
18	PCIe card slot 8: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
19	PCIe card slot 9: filler panel
20	PCIe card slot 10: • Oracle Database Appliance X9-2S: filler panel • Oracle Database Appliance X9-2L: Oracle Retimer PCIe card

Fiber Cables

For optical cables, you must purchase either Short Range (SR) or Long Range (LR) SFP transceivers for each of the network ports, and then plug in the appropriate optical cable.

Currently, Oracle sells both the SR and LR SFP transceivers. In addition to these transceivers, you must purchase the appropriate LC-LC terminated fiber optic cables from a third-party vendor.

Table 3-9 Transceivers for the SFP Ports - 10Gb/1Gb

Name	ATO Part Number	x-option Part Number
SFP+ 10GBase-SR (Short Range Transceiver)	2129a	X2129A-N
SFP+ 10GBase-LR (Long Range Transceiver)	5562a-z	X5562A-Z

Table 3-10 Transceivers for the SFP Ports - 25Gb

Name	ATO Part Number	x-option Part Number
SFP28 25GBase-SR (Short Range Transceiver)	7118017	7118019
SFP28 25GBase-LR (Long Range Transceiver)	7118020	7118021

Copper Cables

You can purchase the following copper cables from Oracle. These cables have built-in SFP connectors:

Table 3-11 Networking Cables for the SFP Ports - 10Gb/1Gb

Name	Length	ATO Part Number	x-option Part Number
TwinAx passive copper cable: 1 meter	1m	7105137	X2130A-1M-N
TwinAx passive copper cable: 2 meter	2m	7105139	7105148
TwinAx passive copper cable: 3 meter	3m	7105140	X2130A-3M-N
TwinAx passive copper cable: 5 meter	5m	7105141	X2130A-5M-N

Table 3-12 Networking Cables for the SFP Ports - 25Gb

Name	Length	ATO Part Number	x-option Part Number
TwinAx passive copper cable: 1 meter	1m	7118359	7118367
TwinAx passive copper cable: 2 meter	2m	7118360	7118368
TwinAx passive copper cable: 3 meter	3m	7118361	7118369
TwinAx passive copper cable: 5 meter	5m	7118362	7118370

Attaching Power Cords and Initializing Components

Attach power cords for Oracle Database Appliance.

> **Caution:**
>
> Before plugging in the power cords, ensure that the electrical outlets providing the power are grounded.

If you use only a single AC circuit, then connect both power cords for each component to that circuit. If you want to maintain N+1 power supply redundancy, then use two separate AC circuits. Connect one power cord from each AC circuit into each component.

For more information about cabling with the supplied Cable Management Arm, refer to *Oracle Database Appliance Owner's Guide*.

Configuring Oracle Integrated Lights Out Manager

Configure Oracle Integrated Lights Out Manager (Oracle ILOM) to manage Oracle Database Appliance independent of the operating system.

Deploying a new Oracle Database Appliance requires a direct connection to the system to configure the Oracle Integrated Lights Out Manager (ILOM) and the first network. If you use the serial port, ensure that the baud rate speeds match. After the first public network interface is configured, you can perform the remaining deployment steps on the Oracle Database Appliance system or from a remote system.

Configuring Oracle ILOM Using IP Address Assigned by DHCP

Ensure that the ILOM NIC is enabled and that `ipv4` is enabled.

1. Using a client system's browser, enter the IP address or host name assigned by DHCP into the browser address field and press **Enter**. For example, https://129.xxx.xx.xx.
2. At the login page, enter the default user name, `root` and the default password, `changeme`.

   The Oracle ILOM web interface is displayed.

Configuring Oracle ILOM Using Static IP Address

1. Using a client system, establish a secure shell (SSH) connection by entering the following on the command line:

   ```
   ssh -l root sp_ip_address
   ```

   where *`sp_ip_address`* is the IP address assigned by DHCP.
2. Enter the default user name, `root`, and the default password, `changeme`.

   The Oracle ILOM CLI prompt appears. Change the default password to a secure password of your choice.
3. Set the working directory.

   ```
   cd /SP/network
   ```

4. Specify a static Ethernet configuration as follows:

```
set pendingipaddress=xxx.xxx.xx.xx
set pendingipnetmask=yyy.yyy.yyy.y
set pendingipgateway=zzz.zzz.zz.zzz
set commitpending=true
set state=enabled
```

`xxx.xxx.xx.xx`, `yyy.yyy.yyy.y` and `zzz.zzz.zz.zzz` are the IP address, netmask, and gateway (respectively) for your ILOM and network configuration.

5. Verify changes and then exit:

```
show /SP/network
exit
```

Configuring Oracle ILOM Using the IPMI tool

To assign IP addresses to Oracle ILOM manually, follow these steps:

1. Open the IPMI tool.

```
# ipmitool -I open sunoem cli
```

2. Go to `SP/network` and verify that the port is enabled:

```
show /SP/network
```

3. If the port is not enabled, then enable the port.

```
# ipmitool -I open sunoem cli "set /SP/network state=enabled"
```

4. In `SP/network`, verify that `ipv4` is enabled.
5. If `ipv6` is enabled, then disable `ipv6` and enable `ipv4`.

```
# ipmitool -I open sunoem cli "set /SP/network/ipv6 state=disabled"
# ipmitool -I open sunoem cli "set /SP/network/ipv4 state=enabled"
```

Related Topics

- Deployment Checklist for Oracle Database Appliance Bare Metal Systems
  This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.

> **See Also:**
>
> *My Oracle Support Note 1395445.1: "ODA (Oracle Database Appliance) : ILOM configuration via Serial port"* at https://support.oracle.com/rs?type=doc&id=1395445.1

Powering On Oracle Database Appliance the First Time

Use this procedure the first time you power on Oracle Database Appliance.

When you power on Oracle Database Appliance for the first time, the system automatically defines your public network interface based on which of the public interface ports are connected. For Oracle Database Appliance X9-2-HA, the dual-port 10/25 GbE SFP28 PCIe adapter is used for the cluster interconnect.

> **Note:**
>
> After you connect power cords, the green SP OK light-emitting diode (LED) lights blink for a few minutes, and then turn to steady ON. The cooling fans also may turn on. However, these events do not indicate that the system is started. You must complete all of the steps in this section to turn on the appliance properly. Read through the entire section to understand the sequence required for initial system startup.

1. (For Oracle Database Appliance X9-2-HA) Turn on the attached storage shelf, or shelves, before powering on the server nodes.
2. Push the recessed power button to turn on the appliance.

> **Note:**
>
> The fully-initialized state is indicated by the green SP OK LEDs staying steadily lit. If any of the green LED lights on the server are still blinking, then wait for them to complete their initialization steps.

Refer to the figures and to the following callout table to identify the power button and the system initialization status indicator lights.

Figure 3-1 Front of Oracle Database Appliance Power Panel

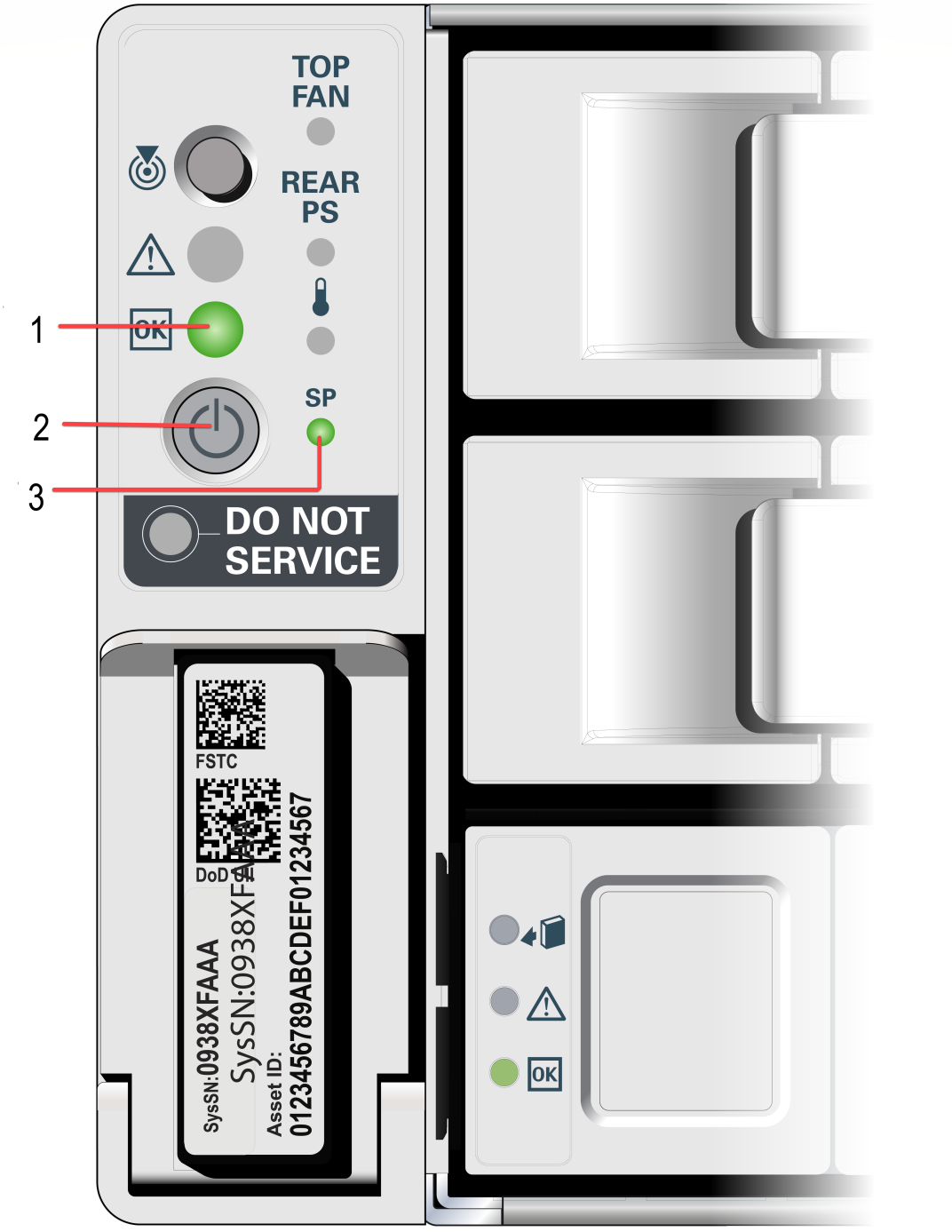


Table 3-13 Description of Callouts for Powering On Oracle Database Appliance

Callout	Function
1	Green Power/OK LED, located on the front panel. The light must be in the steady ON position (Green light does not blink) before you log on to the system.

Table 3-13 (Cont.) Description of Callouts for Powering On Oracle Database Appliance

Callout	Function
2	Recessed ON/OFF power button, which is located on the front panel of the appliance. Push this button only once. Do not repeatedly push the power button.
3	SP OK LED light, located on the front panel of the appliance.

Oracle Database Appliance does not use the DO NOT SERVICE indicator.

3. Wait for Oracle Database Appliance to complete startup.

> **Note:**
>
> Do not repeatedly push the power buttons. Startup can take several minutes to complete.

Oracle Database Appliance is ready for use when the green Power/OK LED (callout 1) on the front of the system remains steadily on.

4

Provisioning Oracle Database Appliance Bare Metal System

Understand the process to configure Oracle Database Appliance bare metal system.

The Browser User Interface is the preferred method of deploying your bare metal platform configuration. The Browser User Interface provides all of the fields necessary to provision Oracle Database Appliance, including configuring the system, network, database, and Oracle Auto Service Request (Oracle ASR). If you prefer to use the command-line interface, you must create a JSON file to configure the deployment options.

- Verifying Cabling and Network Connections
  After powering on Oracle Database Appliance, verify that the network connections are set up correctly.
- Plumbing the Network
  Plumb the Oracle Database Appliance network with the public internet protocol (IP) information assigned to a node, to enable provisioning of the Oracle Database Appliance software.
- Downloading Oracle Database Appliance Software
  Download Oracle Database Appliance software and copy to a temporary location before applying updates to your appliance.
- Installing Oracle Database Appliance Software
  Install Oracle Database Appliance software, before creating the appliance.
- Creating the Appliance
  Create the appliance using the Browser User Interface or ODACLI commands.
- About Enabling Oracle ASMFD on Oracle Database Appliance
  Understand how you can enable Oracle ASMFD on bare metal systems and DB systems on Oracle Database Appliance.

Verifying Cabling and Network Connections

After powering on Oracle Database Appliance, verify that the network connections are set up correctly.

Run the validation scripts only on Oracle Database Appliance HA models.

1. Log into the Oracle Database Appliance console and access the server node with user name as `root` and password `welcome1`.
2. Run validation scripts on both nodes, to verify cabling. For example:

```
# /opt/oracle/dcs/bin/odacli validate-storagetopology
INFO : ODA Topology Verification
INFO : Running on Node0
INFO : Check hardware type
SUCCESS : Type of hardware found : X9-2
INFO : Check for Environment
SUCCESS : Type of environment found : Bare Metal
```

```
INFO : Check number of Controllers
SUCCESS : Number of ahci controller found : 1
SUCCESS : Number of External SCSI controllers found : 2
INFO : Check for Controllers correct PCIe slot address
SUCCESS : Internal RAID controller :
SUCCESS : External LSI SAS controller 0 : 31:00.0
SUCCESS : External LSI SAS controller 1 : 9d:00.0
INFO : Check if JBOD powered on
SUCCESS : 2JBOD : Powered-on
INFO : Check for correct number of EBODS(2 or 4)
SUCCESS : EBOD found : 4
INFO : Check for External Controller 0
SUCCESS : Cable check for port 0 on controller 0
SUCCESS : Cable check for port 1 on controller 0
SUCCESS : Overall Cable check for controller 0
INFO : Check for External Controller 1
SUCCESS : Cable check for port 0 on controller 1
SUCCESS : Cable check for port 1 on controller 1
SUCCESS : Overall Cable check for controller 1
INFO : Check for overall status of cable validation on Node0
SUCCESS : Overall Cable Validation on Node0
SUCCESS : JBOD0 Nickname set correctly
SUCCESS : JBOD1 Nickname set correctly
```

3. If there is a storage expansion shelf connected to the appliance, then power it on. Run the `storagetopology` command on both nodes to make sure the cabling is correct. For example:

```
# /opt/oracle/dcs/bin/odacli validate-storagetopology
INFO : ODA Topology Verification
INFO : Running on Node0
INFO : Check hardware type
SUCCESS : Type of hardware found : X9-2
INFO : Check for Environment
SUCCESS : Type of environment found : Bare Metal
INFO : Check number of Controllers
SUCCESS : Number of ahci controller found : 1
SUCCESS : Number of External SCSI controllers found : 2
INFO : Check for Controllers correct PCIe slot address
SUCCESS : Internal RAID controller :
SUCCESS : External LSI SAS controller 0 : 31:00.0
SUCCESS : External LSI SAS controller 1 : 9d:00.0
INFO : Check if JBOD powered on
SUCCESS : 2JBOD : Powered-on
INFO : Check for correct number of EBODS(2 or 4)
SUCCESS : EBOD found : 4
INFO : Check for External Controller 0
SUCCESS : Cable check for port 0 on controller 0
SUCCESS : Cable check for port 1 on controller 0
SUCCESS : Overall Cable check for controller 0
INFO : Check for External Controller 1
SUCCESS : Cable check for port 0 on controller 1
SUCCESS : Cable check for port 1 on controller 1
SUCCESS : Overall Cable check for controller 1
INFO : Check for overall status of cable validation on Node0
SUCCESS : Overall Cable Validation on Node0
```

```
SUCCESS : JBOD0 Nickname set correctly
SUCCESS : JBOD1 Nickname set correctly
```

4. Verify the interconnect network:

```
# ethtool p1p1
Settings for p1p1:
Supported ports: [ FIBRE ]
Supported link modes: 1000baseT/Full
Supported pause frame use: Symmetric Receive-only
Supports auto-negotiation: Yes
Advertised link modes: 1000baseT/Full
10000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: 25000Mb/s <<<<<<<<<<<<<<<<<<<<<<< check speed
Duplex: Full
Port: Direct Attach Copper
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
Current message level: 0x00000000 (0)
Link detected: yes
# ethtool p1p2
Settings for p1p2:
Supported ports: [ FIBRE ]
Supported link modes: 1000baseT/Full
Supported pause frame use: Symmetric Receive-only
Supports auto-negotiation: Yes
Advertised link modes: 1000baseT/Full
10000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: 25000Mb/s <<<<<<<<<<<<<<<<<<<<<<< check speed
Duplex: Full
Port: Direct Attach Copper
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
Current message level: 0x00000000 (0)
Link detected: yes
```

Related Topics

- Deployment Checklist for Oracle Database Appliance Bare Metal Systems
  This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.

Plumbing the Network

Plumb the Oracle Database Appliance network with the public internet protocol (IP) information assigned to a node, to enable provisioning of the Oracle Database Appliance software.

> **Important:**
>
> On Oracle Database Appliance X9-2 and X10, the `odacli configure-firstnet` command supports bonding of ports across two supported network PCI cards of the same type.

> **Caution:**
>
> On Oracle Database Appliance X9-2 and X10, you cannot change the network bonding **after** the system is deployed. To change the network bonding, you must reimage and deploy the appliance again.

1. Connect to Oracle ILOM remote console, then log into Oracle Database Appliance as `root`.
2. Run the command `configure-firstnet` on **both** nodes. You can choose to enable Link Aggregation Control Protocol (LACP) on the bonding interface when you run `odacli configure-firstnet`.

```
# /opt/oracle/dcs/bin/odacli configure-firstnet
Use bonding on different network interfaces:
    bonding between ports on the same network card [0]
    bonding between Network Card 1 in slot 6 and Network Card 2 in slot 4
[1]
    bonding between Network Card 1 in slot 6 and Network Card 3 in slot 8
[2]
    bonding between Network Card 2 in slot 4 and Network Card 3 in slot 8
[3]
Please input your selection (0 - 3) [0]:
Select the Interface to configure the network on (btbond1 btbond3 btbond5)
[btbond1]:
Use LACP bonding on btbond1 (yes/no) [no]:
Configure DHCP on btbond1 (yes/no) [no]:
INFO: You have chosen Static configuration
Use VLAN on btbond1 (yes/no) [no]:
```

3. Complete the network configuration as prompted and configure the public network interface. Also provide the netmask and gateway IP address.

You use this network connection to transfer the software to the server node.

Example 4-1 Example of a Bonded Configuration

For a bonded configuration, answer `yes` to using a bonding public interface and configure the first network to use a `btbond3` interface without configuring DHCP.

```
# /opt/oracle/dcs/bin/odacli configure-firstnet
Select the Interface to configure the network on (btbond1 btbond3 btbond4)
[btbond1]:btbond3
Use LCAP bonding on btbond3 (yes/no) [no]:yes
INFO: network interface [btbond3] in LACP mode
Configure DHCP on btbond3 (yes/no) [no]:
INFO: You have chosen Static configuration
Use VLAN on btbond3 (yes/no) [no]:yes
Configure VLAN on btbond3, input VLAN ID [2 - 4094] 119
INFO: using network interface btbond3.119
Enter the IP address to configure : 10.31.129.36
Enter the Netmask address to configure : 255.255.255.240
Enter the Gateway address to configure [10.31.129.33] :
INFO: Restarting the network
Shutting down interface btbond1: [ OK ]
Shutting down interface btbond2: [ OK ]
Shutting down interface p3p1: [ OK ]
Shutting down interface p3p2: [ OK ]
Shutting down interface sfpbond1: [ OK ]
Shutting down loopback interface: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface btbond1: Determining if ip address 10.209.13.109 is
already in use for device btbond1...
[ OK ]
Bringing up interface btbond2: [ OK ]
Bringing up interface sfpbond1: [ OK ]
```

Example 4-2 Example of Configuring VLAN

```
# /opt/oracle/dcs/bin/odacli configure-firstnet
Select the Interface to configure the network on (btbond1 btbond3 btbond4)
[btbond1]:btbond3
Use LCAP bonding on btbond3 (yes/no) [no]:yes
INFO: network interface [btbond3] in LACP mode
Configure DHCP on btbond3 (yes/no) [no]:
INFO: You have chosen Static configuration
Use VLAN on btbond3 (yes/no) [no]:yes
Configure VLAN on btbond3, input VLAN ID [2 - 4094] 119
INFO: using network interface btbond3.119
Enter the IP address to configure : 10.31.129.36
Enter the Netmask address to configure : 255.255.255.240
Enter the Gateway address to configure [10.31.129.33] :
INFO: Restarting the network
Shutting down interface btbond1:                           [  OK  ]
Shutting down interface em1:                               [  OK  ]
Shutting down interface p1p1:                              [  OK  ]
Shutting down interface p1p2:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface btbond1:
```

```
Determining if ip address 192.0.2.24 is already in use for device btbond1...
                                                           [  OK  ]
Bringing up interface em1:                                 [  OK  ]
Bringing up interface p1p1:
Determining if ip address 192.0.2.24 is already in use for device
p1p1...                                                              [  OK  ]
Bringing up interface p1p2:
Determining if ip address 192.0.2.24 is already in use for device p1p2...
                                                           [  OK  ]
Bringing up interface btbond1.122:
Determining if ip address 192.0.2.24 is already in use for device
btbond1.122...
                                                           [  OK
INFO: Restarting the DCS agent
initdcsagent stop/waiting
initdcsagent start/running, process 32104
```

Configuring LACP after Deployment

Use the `-o` option with the `odacli configure-firstnet` command to enable or disable LACP on a bonding interface.

Enabling LACP on a bonding interface after deployment:

```
# odacli configure-firstnet -o
INFO: Configure network option
Select the Interface to configure the network on (btbond1 btbond3 btbond5)
[btbond1]:btbond3
Use LACP bonding on btbond3 (yes/no) [no]: yes
INFO: network interface [btbond3] setup in LACP mode
```

Disabling LACP on a bonding interface after deployment:

```
# odacli configure-firstnet -o
INFO: Configure network option
Select the Interface to configure the network on (btbond1 btbond3 btbond5)
[btbond1]:btbond3
Use LACP bonding on btbond3 (yes/no) [yes]: no
INFO: network interface [btbond3] setup in active-backup mode
```

Downloading Oracle Database Appliance Software

Download Oracle Database Appliance software and copy to a temporary location before applying updates to your appliance.

1. Download the software files from My Oracle Support to a temporary location on an external client. Refer to the release notes for details about the software for the latest release.

   For example, download the Oracle Database Appliance GI Clone for ODACLI/DCS stack (patch 30403673) and Oracle Database Appliance RDBMS Clone for ODACLI/DCS stack (patch 30403662) for 19.25:

   ```
   p30403673_1925000_Linux-x86-64.zip
   p30403662_1925000_Linux-x86-64.zip
   ```

2. Unzip the software — it contains README.html and one or more zip files for the patch.

```
unzip p30403673_1925000_Linux-x86-64.zip
unzip p30403662_1925000_Linux-x86-64.zip
```

The GI and RDBMS Clone files for release 19.25 are extracted.

```
odacli-dcs-19.25.0.0.0-date-GI-19.25.0.0.zip
odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip
```

3. Copy the software files from the external client to Oracle Database Appliance. Use the `scp` or `sftp` protocol to copy the bundle.

Example using `scp` command:

```
scp software_file root@oda_host:/tmp
```

Example using `sftp` command:

```
sftp root@oda_host
```

Enter the `root` password, and then copy the file.

```
put software_file
```

For example, for release 19.25:

```
put odacli-dcs-19.25.0.0.0-date-GI-19.25.0.0.zip
put odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip
```

The GI and RDBMS Clone files are copied to the temporary location on the appliance. You can update the repository, and create the appliance, as explained in the next topic.

Related Topics

•

Installing Oracle Database Appliance Software

Install Oracle Database Appliance software, before creating the appliance.

Ensure that the Oracle Database Appliance patches are downloaded and available for updating the repository, as described in the topic *Downloading Oracle Database Appliance Software*.

Starting with Oracle Database Appliance release 19.12, the sizes of volumes `/u01` and `/opt` are smaller than the corresponding sizes in earlier releases. The size of `/u01` is now 40GB and the size of `/opt` is now 30GB. A new ACFS file system, mounted at `/opt/oracle/oak/pkgrepos/orapkgs/clones`, is created during provisioning. The DATA disk group provides space of 150 GB for the file system. The Oracle Grid Infrastructure and Oracle Database clone files are stored on this file system.

Important:

If you want to install third-party software on your Oracle Database Appliance, then ensure that the software does not impact the Oracle Database Appliance software. The version lock on Oracle Database Appliance RPMs displays a warning if the third-party software tries to override Oracle Database Appliance RPMs. You must restore the affected RPMs before patching Oracle Database Appliance so that patching completes successfully.

Important:

To ensure that your deployment has the latest Oracle Database Appliance firmware, download the latest server patch and update the repository. Then, run the `odacli describe-component` command to check whether the firmware is current or out-of-date. If the firmware is not current, then run the `odacli update-server` and `odacli update-storage` commands before you deploy the Oracle Database Appliance software. If the firmware is current, then you can deploy the Oracle Database Appliance software as described in this topic, without updating the server and storage. Before updating the server or storage, ensure that you run the `odacli create-prepatchreport` command to generate the prechecks report, otherwise an error is displayed.

Note:

Public network gateway must be responsive to ping requests. Oracle Clusterware uses the public network gateway as ping target to facilitate the VIP failover in Oracle Database Appliance high-availability environment.

1. Verify the current system version by running the following command on **both** nodes:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-component -v
```

On a single-node Oracle Database Appliance system, run the command `odacli describe-component` without the `-v` option.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-component
```

2. Update the repository with Oracle Grid Infrastructure and Database software only. To prevent space issues in `/opt`, it is recommended that you add other repositories after the bare metal instance is provisioned.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/GI_clone_file,/tmp/DB_clone_file
```

For example, for release 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip,/tmp/odacli-dcs-19.25.0.0.0-
date-GI-19.25.0.0.zip
```

3. Confirm that the repository update is successful:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-job -i job_ID
```

4. Delete the software zip files from the temporary location on the appliance to save space:

```
[root@oda1 opt]# rm software_file
rm: remove regular file software_file? y
```

You have now updated the repository with the Oracle Database Appliance software, and can create your appliance, as described in the next topic.

Related Topics

- Deployment Checklist for Oracle Database Appliance Bare Metal Systems
  This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.

Creating the Appliance

Create the appliance using the Browser User Interface or ODACLI commands.

> **Note:**
>
> If you plan to enable multi-user access, then refer to the chapter *Implementing Multi-User Access on Oracle Database Appliance* before you create your appliance.

Creating the Appliance using ODACLI Commands

You can use the command `odacli create-appliance` and a JSON file that includes your appliance configuration settings to deploy the appliance instead of using the Browser User Interface. Review the readme in the `/opt/oracle/dcs/sample` directory carefully along with the provided JSON example files. Create a JSON file with the necessary changes based on your environment and requirements.

> **See Also:**
>
> *Creating an Appliance Using a JSON File* in this guide for example JSON files and the procedure to create a JSON file.

> **Note:**
>
> It is important to review the readme and the examples carefully before creating your JSON file. If you do not enter your network and Oracle ILOM information correctly based on your setup, you will lose network access to both the host and Oracle ILOM.

Creating the Appliance using the Browser User Interface (BUI)

Follow these steps to create the appliance using the BUI:

Ensure that ports 7070 and 7093 on the network between Oracle Database Appliance and the client machine, are open.

1. Navigate to the Browser User Interface. You are prompted to set the password for the `oda-admin` user.

   ```
   https://ODA-host-ip-address:7093/mgmt/index.html
   ```

2. Log into the Browser User Interface with the new password.

   When you launch the Browser User Interface on an unconfigured appliance, the Appliance page notifies you that the appliance is not configured and provides a link to the pages needed to configure and create the appliance.

   > **Note:**
   >
   > If you face any issues when logging into the Oracle Database Appliance Browser User Interface, then see the topic *Errors When Logging into the Browser User Interface* in the chapter *Troubleshooting Oracle Database Appliance*.

3. Click **Create Appliance**.
4. In the Create Appliance page, if you want to create the appliance from a saved configuration, click **Browse**, and select the JSON file. The information from the saved configuration file is loaded. You can retain or edit the fields. Otherwise, manually, provide the values to create the appliance.
5. In the System Information tab, specify the values as follows.
   a. **System Name**: Enter the host name.

      The host name can contain alphanumeric characters and dashes (-), but cannot start with a number or dash (-) or end with dash (-). Do not exceed 30 characters.

   b. **Domain Name**: Enter the domain name.

   c. **Region**: Select the region of the world where the Oracle Database Appliance is located.

   d. **Time Zone**: Select the time zone where the Oracle Database Appliance is located.

   e. (Optional) **DNS Servers**: Enter addresses for one or more DNS servers.

   f. (Optional) **NTP Servers**: Enter addresses for one or more NTP servers.

   g. **Diskgroup Redundancy**: If there are two disks, then you can select the disk group redundancy as Normal or Flex. To select the the redundancy as **High**, there must exist more than four disks, and a quorum disk. If five or more disks are configured, then the quorum disk is not required. Select **Normal**, **High**, or **Flex**, as per your deployment requirement.

The Flex parameter defines the disk group redundancy. Flex redundancy requires at least 3 storage devices, including a quorum. For a database to be placed on Oracle ASM, you can set the database redundancy during its creation. For a database to be placed on Oracle ACFS, the database inherits the redundancy of the ACFS file system. You can specify an ACFS file system redundancy during its creation. You can set the disk group redundancy only during appliance creation. The database redundancy can be set anytime you create the database.

The redundancy level for DATA, RECO, and FLASH can be set to Normal, Flex, or High based on whether there are two, four, or more disks.

Note: For Oracle Database Appliance hardware models other than X8-2, the redundancy level for REDO is always High.

h. **Data Storage Percentage**: Enter a whole number between 10 and 90 to define the percentage of storage reserved for DATA, the remainder is reserved for RECO. For example, if you enter 80, then 80% of the storage for DATA and 20% for RECO.

i. **Disk Group for Database Home Storage**: Select the disk group, whether DATA or RECO depending on your hardware model, and specify the size to be allocated. This chooses the disk group from which the ACFS storage required for database homes is allocated.

j. **System Password** and **Confirm Password**: Enter the system password in both fields.

The system password is the password set for UNIX and `root` users. The password must begin with an alpha character and cannot contain quotation marks. Do not exceed 30 characters.

k. Specify the **DNS Servers** and **NTP Servers** details.

Starting with Oracle Database Appliance release 19.16, if you specify DNS servers, then ensure that all host names (except SCAN) are resolvable to one and only one IP address through the given domain name and DNS servers. These include Client Access Network Host Name, VIP Name, and ILOM Host Name. This does not apply to SCAN names as they can resolve to multiple IP addresses. If any of the above host names does not resolve to an IP address or resolves to multiple IP addresses, then the the following error is displayed:

```
DCS-10045:Validation error encountered: Hostname 'hostname1' does not
resolve to a unique IP address.
```

6. In the Network Information tab, configure the primary client access network, virtual networks, and an Oracle Integrated Lights Out Manager (Oracle ILOM) network. You are only required to configure the client access network. The ILOM configuration is optional if you already configured the ILOM for the appliance and you do not need to make changes.

Specify the following, depending on whether it is a single-node or two-node Oracle Database Appliance deployment:

a. **Client Access Network Host Name**: For Node0, enter the host name for the primary client access network.

b. **Client Access Network IP Address**: For Node0, enter the IP address for the primary client access network.

c. **VIP Name for Node0**: Enter the name of the virtual IP network for Node0.

d. **VIP IP Address for Node0**: Enter the virtual IP address that resides on Node0 and is shared between the nodes.

e. **VIP Name for Node1**: Enter the name of the virtual IP network for Node0.

f. **VIP IP Address for Node1**: Enter the virtual IP address that resides on Node1 and is shared between the nodes.

g. (Optional) **ILOM Host Name**: Enter the name of the Oracle ILOM host for Node0 and Node1.

h. (Optional) **ILOM Network IP Address**: Enter the IP address for the ILOM for Node0 and Node1.

i. (Optional) **ILOM Network Subnet Mask**: Enter the subnet mask address for the ILOM.

j. (Optional) **ILOM Network Gateway**: Enter the gateway address for the ILOM.

k. **Client Access Network Subnet Mask**: Enter the subnet mask address for the primary client access network.

l. **Client Access Network Gateway**: Enter the gateway address for the primary client access network.

m. **Client Access Network Interface**: Enter the interface for the primary client access network.

n. **SCAN Name**: Specify the Single Client Access Name (SCAN).

o. **SCAN IP Address**: Specify the SCAN IP address. You can specify up to three SCAN IP addresses.

Specify the VIP address and SCAN only for high-availability deployments.

7. Determine how you want to configure your users and groups and whether or not you want to allow operating system role separation:

- Two users with six groups: Customize Users and Groups, select **No**. Allow OS Role Separation, select **Yes**. This is the default configuration.
- Two customized users with six customized groups: Customize Users and Groups, select **Yes**. Allow OS Role Separation, select **Yes**.
- Single user with two groups: Customize Users and Groups, select **No**. Allow OS Role Separation, select **No**
- Single user with six groups: Customize Users and Groups, select **Yes**. Allow OS Role Separation, select **No**. SAP deployments use this configuration.

8. Do you want to create an initial database? Select **Yes** and go to Step 9 or select **No** and go to Step 10.

9. Enter the following information to configure an initial database:

a. **DB Name**: Enter a name for the database.

The name must contain alphanumeric characters and cannot exceed 8 characters.

b. (Optional) **DB Unique Name**: Enter a globally unique name for the database.

Databases with the same DB Name within the same domain (for example, copies of a database created for reporting or a physical standby) must have a different DB Unique Name that is unique within the enterprise. The name must contain alphanumeric, underscore (_), dollar ($), and pound (#) characters, but must begin with an alphabetic character. No other special characters are permitted in a database name. The unique name cannot exceed 30 characters.

c. **DB Version**: Select a database bundle patch number.

d. **CDB**: Select **Yes** or **No** to specify whether or not you want a Container Database (CDB).

e. **PDB Name**: Enter a name for the pluggable database (PDB).

The name must begin with an alphanumeric character. The following characters are valid: alphanumeric characters, and underscore (_).

f. **PDB Admin User**: Enter an Admin user name for the pluggable database (PDB).

g. **Database Edition**: Select the Oracle Database edition, either Standard Edition and Enterprise Edition. Your license determines which database edition you are eligible to create in the appliance.

   For Oracle Database 19c Standard Edition, you can only create single-instance Oracle Database, with or without high-availability.

h. **SYS and PDB Admin User Password**: Provide a password for the database.

i. **Deployment**: Select a deployment type from the list. The options are RAC, RAC-One, or SI (single-instance database). If you select a single instance database, then select the node for the SI database deployment.

   For Standard Edition Oracle Database 19c or later, you can choose to enable high availability for single-instance databases. For Standard Edition Oracle Database 19.6 or later, if you choose to create a single-instance database, then the high availability feature is available by default. To disable high-availability, set the value in the **Enable High Availability** field to **No**.

   For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can choose to enable high availability for single-instance databases. For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, if you choose to create a single-instance database, then the high availability feature is disabled by default. To enable high-availability, set the value in the **Enable High Availability** field to **Yes**.

   If you select a single instance database, then you have the option to create the database on either Node0 or Node1. The default is Node0.

j. **Shape**: Select a database shape from the list.

k. In the **Database Class** field, select a database class from the drop-down list. If an option is not available in the list, it is not supported for the database edition on the Oracle Database Appliance or the version that you selected. The default is OLTP.

l. In the **Storage** field, select **ACFS** or **ASM** from the drop-down list. The default is Oracle ASM.

m. If you specified the disk group redundancy as **Flex**, then you can select the **Database Redundancy** value as **Mirror** or **High**.

n. **Data Files on Flash Storage**: Select **Yes** or **No**.

   This option is only available if the HA system has HDD storage drives.

o. **Configure EM Express**: Select **Yes** or **No**.

   Select **Yes** to configure the Oracle Enterprise Manager Database Express (EM Express) console for Oracle Database 12.2.1.0 or 12.1.0.2. Selecting Yes enables you to use the console to manage the database.

p. **Character set**: Select a character set.

q. **National Characterset**: Select a national characterset.

r. **Language**: Select the database language.

s. **Territory**: Select a territory or location from the list.

t. For Oracle Database Enterprise Edition 19c or later, you can choose to enable Transparent Database Encryption (TDE). Select **Yes** or **No** in the **Enable TDE** option. Specify and confirm the **TDE Password**. By default, the TDE option is disabled.

10. In the Diagnostics Information tab, configure and enable Oracle ASR or choose to enable trace file redaction.

    Enable Trace File Redaction: Select **Yes** if you want to redact sensitive information in diagnostics collections.

    You can configure and enable Oracle Auto Service Request (Oracle ASR) now or later:

    - To not enable Oracle ASR during deployment, select **No** and click **Submit**. After deployment, you can configure an internal Oracle ASR or register with an external Oracle ASR Manager from either the Browser User Interface or command-line interface.
      - Internal Oracle ASR: choose to configure Oracle ASR Manager on Oracle Database Appliance or use Oracle ASR Manager configured on another server in the same network as your appliance.
      - External Oracle ASR: If you already have Oracle ASR Manager configured elsewhere, you can register Oracle Database Appliance with your existing Oracle ASR Manager.
    - To enable Oracle ASR, select **Yes** and complete the fields:

    a. Select **Internal** or **External** in the **ASR Type** field.

    b. **ASR User Name**: Enter the e-mail address associated with the My Oracle Support account under which the server is registered.

    c. **Password**: Enter the password associated with the My Oracle Support account under which the server is registered.

    d. (Optional) Specify the **HTTPS Port Number** for the ASR manager. The default port number is 16162.

    e. **HTTP Proxy used for Upload to ASR**: Select **Yes** or **No**.

    f. **Proxy Server Name**: If you are using a proxy for upload, enter the proxy server name.

    g. **Proxy Port**: If you are using a proxy for upload, enter the proxy port.

    h. **(Optional) HTTP Proxy Requires Authentication**: If you are using a proxy for upload, select **Yes** if you require authentication. If you do not require authentication, select **No**.

    i. **Proxy User Name**: If you are using a proxy for upload, enter the proxy user name.

    j. **(Optional) Proxy Password**: If you are using a proxy for upload and require authentication, enter the proxy password.

11. Click **Submit**. When prompted, click **Yes** to confirm that you want to start the job to deploy the appliance.

12. Verify that the appliance is deployed. Run the `odacli describe-system` command. For high-availability deployments, run the command on both nodes.

The job to create the appliance takes time to complete. To monitor the job progress, click the **Activity** tab. Click the job number to view the tasks.
After deployment, the `root` and database users `SYS`, `SYSTEM`, and `PDBADMIN` are set to the system password. The `oracle` and `grid` passwords are set to the default password. Change these passwords to comply with your user security protocols.

Related Topics

- Errors When Logging into the Browser User Interface
  If you have problems logging into the Browser User Interface, then it may be due to your browser or credentials.

Related Topics

- Deployment Checklist for Oracle Database Appliance Bare Metal Systems
  This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.

Related Topics

- Reimaging an Oracle Database Appliance Baremetal System
  Reimage Oracle Database Appliance to perform a bare metal restore of Oracle Database Appliance.

Related Topics

- Implementing Multi-User Access on Oracle Database Appliance
  Understand how multi-user access can be implemented on Oracle Database Appliance, its advantages and restrictions and the associated life cycle management changes for your appliance.
- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

About Enabling Oracle ASMFD on Oracle Database Appliance

Understand how you can enable Oracle ASMFD on bare metal systems and DB systems on Oracle Database Appliance.

Oracle ASM Filter Driver (Oracle ASMFD) simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

By default, Oracle ASMFD is enabled on Oracle Database Appliance bare metal systems and DB systems. To disable Oracle ASMFD on Oracle Database Appliance bare metal system and DB system, specify `"enableAFD":"FALSE"` in the JSON file when you provision the bare metal system and DB system. If Oracle ASMFD is not enabled on Oracle Database Appliance bare metal system, then you cannot enable Oracle ASMFD on Oracle Database Appliance DB system. If Oracle ASMFD is enabled on Oracle Database Appliance bare metal system, then you can choose to enable or disable Oracle ASMFD on Oracle Database Appliance DB system by specifying the parameter `"enableAFD":"TRUE"` or `"enableAFD":"FALSE"` in the DB system JSON file during provisioning of the DB system.

For DB systems, if the `"enableAFD"` attribute is not specified in the JSON file during provisioning, then the Oracle ASMFD configuration on the DB system is the same as the bare metal system Oracle ASMFD configuration.

For more information about specifying the `"enableAFD"` attribute in the JSON files when provisioning bare metal and DB systems, see the example JSON files provided in the chapters *Creating an Appliance Using JSON File* and *Managing an Oracle Database Appliance KVM Deployment* in this book.

> **Note:**
>
> To identify if Oracle ASMFD is enabled or not, check whether the `/etc/oracleafd.conf` file exists or not. Oracle ASMFD is enabled if this file exists, otherwise, Oracle ASMFD is disabled.

5

Implementing Multi-User Access on Oracle Database Appliance

Understand how multi-user access can be implemented on Oracle Database Appliance, its advantages and restrictions and the associated life cycle management changes for your appliance.

> **Caution:**
>
> You can enable multi-user access **only** at the time of provisioning Oracle Database Appliance on bare metal systems and **cannot** rollback multi-user access after you provision and deploy your appliance. The Oracle Database Appliance administration model changes when you deploy the Multi-User Access feature. Evaluate your role separation requirements before using this feature. Deploying on a test system first can help with the evaluation and assessment of the new administration model.

> **Note:**
>
> You can enable multi-user access on Oracle Database Appliance DB systems when you create the DB system either using ODACLI commands or Browser User Interface (BUI).

- About Multi-User Access on Oracle Database Appliance
  Before provisioning your appliance, understand how multi-user access can enhance the security of the system and provide an efficient mechanism for role separation.
- About Users, Roles, Entitlements, and Operations on a Multi-User Access Enabled Oracle Database Appliance System
  Understand the users, roles, entitlements, and operations when you provision your appliance with multi-user access enabled.
- About Granting and Revoking Resource Access
  Muti-user access allows exclusive or shared access to resources. Review this example about shared resource access.
- Using Passwordless Multi-User Access on Oracle Database Appliance
  Understand how you can implement passwordless multi-user access on Oracle Database Appliance, its advantages and restrictions, and the associated life cycle management changes for your appliance.
- Provisioning Oracle Database Appliance with Multi-User Access
  You can enable multi-user access **only** when you provision Oracle Database Appliance, using CLI commands or the Browser User Interface.
- Creating, Viewing, and Deleting Users on Oracle Database Appliance with Multi-User Access
  After provisioning the appliance with multi-user access enabled, you can create users with specific entitlements.

- Activating a New User on Oracle Database Appliance with Multi-User Access
  Understand how to activate a new user on multi-user access enabled Oracle Database Appliance.
- Granting and Revoking Resource Access on Oracle Database Appliance with Multi-User Access
  You can grant and revoke resource access on the appliance.
- Viewing Roles, Operations, and Entitlements on Oracle Database Appliance with Multi-User Access
  You can view roles and entitlements on the appliance.
- Managing Databases and Database Homes on Oracle Database Appliance with Multi-User Access
  The custom user created on multi-user access Oracle Database Appliance can deploy and manage databases and database homes.
- Changing the Password for a User Account on Oracle Database Appliance with Multi-User Access
  Understand how to manage passwords on multi-user access Oracle Database Appliance.
- Resetting the Password for a Locked User Account on Oracle Database Appliance with Multi-User Access
  Understand how to reset the password on multi-user access Oracle Database Appliance.

About Multi-User Access on Oracle Database Appliance

Before provisioning your appliance, understand how multi-user access can enhance the security of the system and provide an efficient mechanism for role separation.

Currently, a single Oracle Database Appliance account with user name and password is used to securely connect to the appliance, run ODACLI commands, or log into BUI. The `root` user performs all administration on an Oracle Database Appliance. With multi-user access, you have the option of providing separate access to database administrators to manage databases. Display of resources within the Browser User Interface are also filtered as per user role. The `root` access is restricted to the Oracle Database Appliance system administrator to access system logs or debug issues that require `root` access.

When you enable multi-user access, you create multiple users with different roles that restrict them from accessing resources created by other users and also restrict the set of operations they can perform using ODACLI commands or Browser User Interface (BUI). The same user credentials that you set up, can be used for logging into the BUI and running ODACLI commands. The BUI also displays resources and information based on access to the set of resources. A separate **Multi-User Access Management** tab is available only to the `odaadmin` user to administer the users and resources in the system.

> **Note:**
>
> When you enable multi-user access, the Oracle Database Appliance administrator is `odaadmin`. This user has access to all the resources on the appliance and can run any operations using ODACLI or the BUI using the same set of credentials. When you do not enable multi-user access, the user name you use to log into BUI is `oda-admin`.

> **Note:**
>
> The authentication token support for ODACLI session management is linked to a multi-user access user account. Since `root` is an operating system administrative user and not a multi-user access user, auth token based session management system is not supported when a user logs in as `root`. Therefore, you must provide an Oracle Database Appliance account user name and password to run any ODACLI command.

Benefits of enabling multi-user access

- Multi-user access feature supports user lifecycle management such as creation, activation, updation, deactivation, deletion, and credential management.
- By using multi-user access, multiple departments such as finance and human resources within the same organization can use Oracle Database Appliance as a consolidation platform for hosting their databases in a secure manner as only the authorized users in the respective departments can access their departmental databases and perform lifecycle management operations on the databases.
- Organizations that have root access policy restrictions can use multi-user access and create separate users with restricted roles.
- Without multi-user access, all databases were created as the default database user chosen during provisioning. Hence, even in a `sudo`-based multi-user environment, the ODA administrator could not track usage of resources. Now such reporting is possible at the database level.
- Multi-user access feature supports token-based session management. A user enters the password only when they run the first `odacli` command. Subsequently, a token is generated and till it expires, the user is not required to enter the password again. Each time an `odacli` command is run, the existing token is refreshed with a new token that has the expiry of 120 mins or the value configured by the `odaadmin` user. This means if the `odacli` session is not idle for the expiry duration, the user does not need to enter the password again.
- Both Basic Auth and mTLS modes of authentication are supported. ODACLI and BUI uses Basic Auth. Users such as `oracle` and `grid` can also run certain operations on the DCS agent using mTLS-based authentication. Basic Auth is a password-based authentication scheme. mTLS is a certificate-based authentication scheme where both the client (user) and the server (DCS agent) mutually present and authenticate each other's certificate before the authentication is deemed complete.
- Multi-user access provides for user account locking on multiple failed login attempts and password expiration. You can also unlock and reset the account in case the password is forgotten.

> **Note:**
>
> You can enable multi-user access **only** at the time of provisioning Oracle Database Appliance and **cannot** rollback multi-user access after you provision and deploy your appliance. Provision the feature on your test system first, and then deploy this feature on your production system.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.
- ODACLI Command Changes with Multi-User Access on Oracle Database Appliance
  Understand the ODACLI command changes when you enable multi-user access.

About Users, Roles, Entitlements, and Operations on a Multi-User Access Enabled Oracle Database Appliance System

Understand the users, roles, entitlements, and operations when you provision your appliance with multi-user access enabled.

> **Note:**
>
> Note that in a multi-user access enabled deployment, the `oda-admin` user is not present. The ODA admin user name is `odaadmin`. The first user with administrative privileges is called `odaadmin`. This user can log into the BUI and run ODACLI commands. This user can also create other user accounts with roles and entitlements as required.

About Roles, Entitlements, and Operations

Each user can be assigned one or more roles in a multi-user enabled Oracle Database Appliance system. Each role encompasses a set of entitlements that authorizes a user to perform only a specific set of operations using ODACLI or BUI. Each entitlement in turn, is a group of a similar set of operations. For example, `PROVISIONDB-MGMT` is an entitlement that encompasses provisioning-related operations such as create-database, clone-database, delete-database, register-database. Similarly, `BACKUPDB-MGMT` encompasses backup related operations such as create-backup, delete-backup, irestore-database, recover-database , and so on. The `ODA-DB` role has access to a collection of entitlements such as `PROVISIONDB-MGMT`, `BACKUPDB-MGMT`, and `PATCHDB-MGMT`. A user with the `ODA-DB` role can perform all the database lifecycle management operations on the databases that they own. Additionally, if a user with the role of `ODA-DB` is granted the `ODA-GRID` role as well, this user can now perform Oracle Grid Infrastructure-related operations.

Roles can be internal or external. Internal roles are assigned to system users and are used internally for the purpose of administration of the Oracle Database Appliance system. For example, the `ODA-ADMINISTRATOR` role is assigned to the Oracle Database Appliance system administrator to manage the appliance or associated entities. Another example is a DB system communicating with the bare metal using a system user with the role of `ODA-DBVMINFRA`.

External roles can be granted to the new users created by `odaadmin`, the Oracle Database Appliance system administrator. For example, the `odaadmin` creates a new user `odadb1` with the role of `ODA-DB`. Now this user `odadb1` is entitled to create databases and perform lifecycle management operations because of the role granted to them. A user can have one or more roles.

The topic *ODACLI Command Changes with Multi-User Access on Oracle Database Appliance* describes the ODACLI commands that have changes for multi-user access and the entitlements required to run the commands.

Multi-User Access User Roles

When you enable multi-user access on Oracle Database Appliance, the following user roles are available:

- **ODA-ADMINISTRATOR:** This is an internal role assigned to the first user (`odaadmin`) created during the provisioning of an Oracle Database Appliance. This role entitles `odaadmin` to run all ODACLI commands or perform all Browser User Interface (BUI) operations. This role cannot be assigned to the new users that `odaadmin` creates. The `odaadmin` account is an administrator role that can run any operation (command) on any resource. For example, user `oda1` creates a database `db1` and user `oda2` creates a database `db2`. Each user can now perform lifecycle management operations on their respective database only. But, `odaadmin` can patch both databases by running `odacli` commands. This allows both DBAs (`oda1` and `oda2`) and an overall administrator (`odaadmin`) to perform functions specific to their role.
- **ODA-DB:** This is an external role available to `odaadmin` to assign to newly-created users. This role entitles the user to perform database management operations such as create, modify, restore, recover, backup, patch, clone, move, register, and delete.
- **ODA-OAKDROOT:** This is an internal role that is assigned to system user `oakdroot` created during provisioning and is used by OAKD to run certain operations such as get-disks and release-disks on the DCS agent. OAKD communicates with DCS through mTLS certificates-based authentication. Hence, this user does not have any password associated with it. Do not use this role to connect by SSH to the appliance or log into the Browser User Intercae to run ODACLI commands. None of the password management commands such as change-password apply to this user.
- **ODA-GRID:** This is an internal role assigned to the `grid` user. This role entitles the user to run Oracle Grid Infrastructure-related operations.
- **ODA-DBVMINFRA:** This is an internal role assigned to the DBVM user created on the bare metal system when the DB system is provisioned. This role entitles the user to synchronize metadata between the DB system and the bare metal system.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.
- ODACLI Command Changes with Multi-User Access on Oracle Database Appliance
  Understand the ODACLI command changes when you enable multi-user access.

About Granting and Revoking Resource Access

Muti-user access allows exclusive or shared access to resources. Review this example about shared resource access.

Muti-user access allows exclusive or shared access to resources. It is recommended that each user creates their own database home and then creates databases in the home. This provides for an efficient method of separation of duties where each user has exclusive access to their databases. However, in certain exceptional situations, such as lack of disk space, a user can request the administrator `odaadmin` to grant them shared access to a resource owned by another user.

For example, if user `oda1` wants to create a database of version 19c and there is already a database home `DBH2` of the same version created by another user `oda2`. With the consent of user `oda2`, user `oda1` can now request the `odaadmin` user to grant them shared access to

database home `DBH2`. Once the shared access is granted, user `oda1` can now create a database `db1` on the shared database home `DBH2` and manage it. Note that user `oda1` can connect to the database `db1` only through the SYS user password and not as a password-less connection based on operating system authentication, as the database home `DBH2` is still owned by user `oda2`. Oracle Database Appliance resources such as database home, database storage, and databases can be shared accross users in a similar fashion on an on-demand basis. However, there are restrictions to the secondary owner managing the shared resource.

Using Passwordless Multi-User Access on Oracle Database Appliance

Understand how you can implement passwordless multi-user access on Oracle Database Appliance, its advantages and restrictions, and the associated life cycle management changes for your appliance.

> **Caution:**
>
> You can enable passwordless multi-user access **only** at the time of provisioning Oracle Database Appliance on bare metal and DB systems and **cannot** rollback multi-user access after you provision and deploy your appliance. The Oracle Database Appliance administration model changes when you deploy the passwordless multi-user access feature. Evaluate your role separation requirements before using this feature. Deploying on a test system first can help with the evaluation and assessment of the new administration model.
> Existing bare metal and DB system deployments that do not have multi-user access enabled or have the earlier multi-user access feature which requires a password when you first run an odacli command, cannot be converted to a passwordless multi-user access deployment.

> **Note:**
>
> Starting with Oracle Database Appliance release 19.23, a passwordless flavor of the multi-user access feature is also available. You can provision a DB system with either flavor of multi-user access on a bare metal system that does not have multi-user access enabled. On such a bare metal system, you can still use role separation and resource ownership capabilities that multi-user access or multi-user access passwordless provides.

- About Passwordless Multi-User Access on Oracle Database Appliance
  Before provisioning your appliance, understand how passwordless multi-user access can enhance the security of the system and provide an efficient mechanism for role separation.
- Provisioning Oracle Database Appliance with Passwordless Multi-User Access
  You can enable passwordless multi-user access **only** when you provision Oracle Database Appliance using CLI commands. Passwordless multi-user access feature provisioning and management is not supported through the Browser User Interface (BUI).

- Creating, Viewing, and Deleting Users on Oracle Database Appliance with Passwordless Multi-User Access
  After provisioning the appliance with passwordless multi-user access enabled, you can create users with specific entitlements.
- Changing the Password for a User Account on Oracle Database Appliance with Passwordless Multi-User Access
  Understand how to manage passwords on multi-user access Oracle Database Appliance.
- Example JSON Files to Create a Single-Node Passwordless Multi-User Enabled Appliance
  Follow these JSON file examples to create a JSON file to deploy a single node appliance with the command `odacli create-appliance`.
- Example JSON File to Create a High-Availability Passwordless Multi-User Enabled Appliance
  Follow the JSON file example to create a JSON file to deploy a high-availability appliance with the command `odacli create-appliance`.
- Example JSON Files to Create a Single-Node Passwordless Multi-User Enabled DB System
  Follow the JSON file example to create a JSON file to deploy a single-node DB System, with role separation, with the command `odacli create-dbsystem`.
- Example JSON File to Create a High-Availability Passwordless Multi-User Enabled DB System
  Follow the JSON file example to create a JSON file to deploy a two-node DB System, with role separation, with the command `odacli create-dbsystem`.

About Passwordless Multi-User Access on Oracle Database Appliance

Before provisioning your appliance, understand how passwordless multi-user access can enhance the security of the system and provide an efficient mechanism for role separation.

Oracle Database Appliance supports enabling of passwordless multi-user access at the time of provisioning of Oracle Database Appliance, on bare metal systems and DB systems. You must specify the `"isMultiUserAccessPLEnabled": true` option in the JSON file used for provisioning bare metal systems and DB systems. For more information, see the example JSON file with this option in this chapter.

When you enable passwordless multi-user access, you create multiple users with different roles that restrict them from accessing resources created by other users and also restrict the set of operations they can perform using ODACLI commands.

> **Note:**
>
> You can enable passwordless multi-user access on Oracle Database Appliance bare metal and DB systems at the time of provisioning using ODACLI commands. You cannot use the Browser User Interface (BUI) to enable passwordless multi-user access on Oracle Database Appliance bare metal systems. However, you can use the BUI on the Oracle Database Appliance bare metal system to enable multi-user access on Oracle Database Appliance DB systems at the time of provisoning.

Benefits of enabling passwordless multi-user access

- If your corporate policies prevent direct usage of Oracle Database Appliance user account passwords by employing tools or software that allow access to Oracle Database Appliance through your corporate credentials, then you can configure passwordless multi-user

access and run commands on your appliance without the need for specifying your Oracle Database Appliance account password when you first run an ODACLI command or when your authentication token expires. This does not compromise the security of the system as all communication to the Oracle Database Appliance server is through a strong and secure system generated password, unique for every user.

- Passwordless multi-user access feature provides role separation and supports user lifecycle management such as creation, deletion, and credential management.
- By using passwordless multi-user access, multiple departments such as finance and human resources within the same organization can use Oracle Database Appliance as a consolidation platform for hosting their databases in a secure manner as only the authorized users in the respective departments can access their departmental databases and perform lifecycle management operations on the databases.
- Organizations that have root access policy restrictions can use passwordless multi-user access and create separate users with restricted roles.
- Without multi-user access, all databases were created as the default database user chosen during provisioning. Hence, even in a `sudo`-based multi-user environment, the ODA administrator could not track usage of resources. Now such reporting is possible at the database level.
- For passwordless multi-user access, you do not have to specify the Oracle Database Appliance account password when you first run an ODACLI command or when the authentication token expires. The Oracle Database Appliance system generates a strong and secure password for every user. This password is internally stored by the system and used when you run odacli commands.
- Both Basic Auth and mTLS modes of authentication are supported. ODACLI uses Basic Auth. Users such as `oracle` and `grid` can also run certain operations on the DCS agent using mTLS-based authentication. Basic Auth is a password-based authentication scheme. mTLS is a certificate-based authentication scheme where both the client (user) and the server (DCS agent) mutually present and authenticate each other's certificate before the authentication is deemed complete.

> **Note:**
>
> You can enable passwordless multi-user access **only** at the time of provisioning Oracle Database Appliance and **cannot** rollback passwordless multi-user access after you provision and deploy your appliance. Provision the feature on your test system first, and then deploy this feature on your production system.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.
- ODACLI Command Changes with Multi-User Access on Oracle Database Appliance
  Understand the ODACLI command changes when you enable multi-user access.

Provisioning Oracle Database Appliance with Passwordless Multi-User Access

You can enable passwordless multi-user access **only** when you provision Oracle Database Appliance using CLI commands. Passwordless multi-user access feature provisioning and management is not supported through the Browser User Interface (BUI).

Provision your appliance as described in the chapter *Provisioning Oracle Database Appliance Bare Metal System*.

> **Caution:**
>
> You can enable passwordless multi-user access **only** at the time of provisioning Oracle Database Appliance on bare metal systems and **cannot** rollback multi-user access after you provision and deploy your appliance. The Oracle Database Appliance administration model changes when you deploy the passwordless multi-user access feature. Evaluate your role separation requirements before using this feature. Deploying on a test system first can help with the evaluation and assessment of the new administration model.

Overall Steps in Provisioning Oracle Database Appliance with Passwordless Multi-User Access Enabled

1. Enable passwordless multi-user access.
   - To provision your appliance using a JSON file, add the attribute `"isMultiUserAccessPLEnabled": true` in the `prov_req.json` file. If the attribute is set to `false` or does not exist in the `prov_req.json` file, then passwordless multi-user access is not enabled during provisioning of the appliance. Note that you can set either multi-user access or passwordless multi-user access in the JSON file, **not** both.

```
"isRoleSeparated": true,
        "isMultiUserAccessPLEnabled": true,
        "osUserGroup": {
            "groups": [{
                    "groupId": 1001,
                    "groupName": "oinstall",
                    "groupRole": "oinstall"
                },

...
```

2. Connect to the Oracle Database Appliance network and run the `odacli configure-firstnet` command using the Oracle ILOM console.
3. Log into Oracle Database Appliance as the root user and run the `odacli create-appliance -r prov.json` command with `"isMultiUserAccessPLEnabled": true` in the `prov.json` file.
4. After the provisoning job is submitted, log out as `root` and log in as the `odaadmin` user using a PAM utility or passwordless SSH and check the status of the provisioning job and ensure that it is successful.

5. The system creates Oracle Database Appliance accounts for `odaadmin`, `oracle`, and `grid` users. These are Oracle Database Appliance system users and their accounts are activated during created. The user `odaadmin` is created with the role of `ODA-ADMINISTRATOR` while the `oracle` and `grid` users are created with the role of `ODA_DB` and `ODA_GRID` respectively.
6. The system configures the multi-user access repository with a list of roles and entitlements, used for assigning to the users in the system.
7. You can now log into the appliance using the PAM utility that is integrated with the appliance. The PAM utility prompts you for your corporate credentials and on successful authentication, logs you into the appliance as an Oracle Database Appliance user as per the mapping defined by the PAM utility administrator. Alternatively, you can set up passwordless SSH to log into the appliance.
8. After the bare metal system is provisioned, log in as the `odaadmin` user and run the `odacli create-dbsystem -p` *`prov.json`* command with `"isMultiUserAccessPLEnabled": true` in the `prov.json` file.
9. After the DB system is provisioned, log in as the `odaadmin` user through the PAM utility or passwordless SSH and create additional users, if required.
10. Log into the DB system as the `oracle` user through the PAM utility or passwordless SSH and create databases for the `oracle` user.
11. Log into DB system as a custom user through the PAM utility or passwordless SSH and create databases for the custom user.

Related Topics

- Provisioning Oracle Database Appliance Bare Metal System
  Understand the process to configure Oracle Database Appliance bare metal system.
- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Creating, Viewing, and Deleting Users on Oracle Database Appliance with Passwordless Multi-User Access

After provisioning the appliance with passwordless multi-user access enabled, you can create users with specific entitlements.

After you provision your appliance with multi-user access enabled, do the following:

Creating, Viewing, and Deleting Users with ODACLI Commands

1. Connect to the appliance as the `odaadmin` user.

   ```
   ssh odaadmin@oda-box hostname/IP
   ```

2. Run any ODACLI command and provide the `odaadmin` password when prompted.
3. On successful authentication, create a user with the following command:

   ```
   odacli create-user –u username -r comma-separated role names
   ```

For example, create a user `dbuser1` with lifecycle management privileges for dabatases:

```
odacli create-user -u dbuser1 -r ODA-DB
```

The `odaadmin` user creates `dbuser1` and assigns a temporary password.

4. After the user is created successfully, the `dbuser1` can log into the appliance with the temporary password.
5. The `dbuser1` is in the `Inactive` state. Activate the user with the following command:

   ```
   odacli activate-user
   ```

   You are prompted to change the password. Enter the temporary password, the new password, and confirm the new password.
6. Use the new password to connect by SSH into the appliance and run ODACLI commands or connect to the Browser User Interface.
7. The `odaadmin` user can view all the users in the system:

   ```
   # odacli list-users
   ```

8. The `odaadmin` user can view details for a user in the system:

   ```
   # odacli describe-user -u user_id
   ```

9. Delete a user in the system. Note that only the `odaadmin` user can delete a user in the system.

   ```
   # odacli delete-user -u user_id
   ```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Changing the Password for a User Account on Oracle Database Appliance with Passwordless Multi-User Access

Understand how to manage passwords on multi-user access Oracle Database Appliance.

> **Note:**
>
> If your appliance is configured as passwordless multi-user enabled, then all user accounts are preactivated during creation and never get locked because the system internally generates the password for each user and provides it to the DCS agent for authentication when required for running ODACLI commands. For passwordless multi-user enabled systems, do **not** run any scripts for unlocking the `odaadmin` account or the `odacli reset-password` command to reset the password. Run the `odacli change-password` command to change the system generated password for your Oracle Database Appliance account.

Changing the Password on Passwordless Multi-User Access Enabled System Using ODACLI Commands

- You can change password for an Oracle Database Appliance user, whose account is active. This command resets the password of this user account to a new randomly generated password.

```
odacli change-password
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Example JSON Files to Create a Single-Node Passwordless Multi-User Enabled Appliance

Follow these JSON file examples to create a JSON file to deploy a single node appliance with the command `odacli create-appliance`.

> **Note:**
>
> If you do not enter your network and Oracle ILOM information correctly based on your setup, then network access is lost to both the host and Oracle ILOM.

When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

Note: To configure Oracle ASR during provisioning, always specify the asrType parameter in the JSON file.

Example 5-1 JSON File to Create a Single Node Oracle Database Appliance with Role Separation and Passwordless Multi-User Access

The following is an example of a JSON file that creates a single node appliance on Oracle Database Appliance. The example uses role separation and enables passwordless multi-user access.

```
   {
 "instance" : {
   "name" : "odambox",
   "instanceBaseName" : "odambox-c",
   "dbEdition" : "EE",
   "timeZone" : "UTC",
   "ntpServers" : ["xx.x.x.x"],
   "dnsServers" : ["x.x.x.xx","xx.x.x.xx","xx.x.x.xx"],
   "domainName" : "example.com",
   "isRoleSeparated" : true,
   "isMultiUserAccessPLEnabled" : true,
   "osUserGroup" : {
     "groups" : [ {
       "groupId" : 1001,
```

```
      "groupName" : "oinstall",
      "groupRole" : "oinstall"
    }, {
      "groupId" : 1002,
      "groupName" : "dbaoper",
      "groupRole" : "dbaoper"
    }, {
      "groupId" : 1003,
      "groupName" : "dba",
      "groupRole" : "dba"
    }, {
      "groupId" : 1004,
      "groupName" : "asmadmin",
      "groupRole" : "asmadmin"
    }, {
      "groupId" : 1005,
      "groupName" : "asmoper",
      "groupRole" : "asmoper"
    }, {
      "groupId" : 1006,
      "groupName" : "asmdba",
      "groupRole" : "asmdba"
    } ],
    "users" : [ {
      "userId" : 1000,
      "userName" : "oracle",
      "userRole" : "oracleUser"
    }, {
      "userId" : 1001,
      "userName" : "grid",
      "userRole" : "gridUser"
    } ]
  }
},
 "users": [
              {
                  "id": 1000,
                  "name": "ugrid",
                  "role": "gridUser"
              },
              {
                  "id": 1001,
                  "name": "uoracle",
                  "role": "oracleUser"
              }
          ]
      },
      "multiUserAccess": {
          "isMultiUserAccessPLEnabled": true
      }
  },
"sysOraHomeStorage": {
      "volumeSize": "160",
      "diskgroup": "DATA"
  }
"nodes" : [ {
```

```
  "nodeNumber" : "0",
  "nodeName" : "odambox",
  "network" : [ {
     "nicName" : "btbond1",
     "ipAddress" : "xx.x.x.xx",
     "subNetMask" : "xxx.xxx.xxx.x",
     "gateway" : "xx.x.x.x",
     "networkType" : [ "Public" ],
     "isDefaultNetwork" : true
    },
  ],
  "ilom" : {
    "ilomName":"odambox-c",
    "ipAddress":"xx.x.x.xx",
    "subNetMask":"xxx.xxx.xx.x",
    "gateway":"xx.x.x.x"
   }
} ],
"grid" : {
  "diskGroup" : [ {
    "diskGroupName" : "DATA",
    "redundancy" : "NORMAL",
    "diskPercentage" :80
  }, {
    "diskGroupName" : "RECO",
    "redundancy" : "NORMAL",
    "diskPercentage" :20
  } ],
"scan" : null,
"vip": null,
  "language" : "en",
  "enableAFD":"TRUE"
},
"database" : {
  "dbName" : "myTestDb",
  "databaseUniqueName":"myTestDb_sea1kj",
  "dbEdition" : "EE",
  "dbVersion" : "19.23.0.0",
  "dbHomeId":null,
  "instanceOnly" : false,
  "isCdb" : true,
  "pdBName" : "pdb1",
  "pdbAdminuserName" : "pdbuser",
  "enableTDE":true,
  "adminPassword" : "password",
  "dbType" : "SI",
  "dbTargetNodeNumber" : null,
  "dbClass" : "OLTP",
  "dbShape" : "odb1",
  "dbStorage" : "ACFS",
  "dbCharacterSet" : {
    "characterSet" : "AL32UTF8",
    "nlsCharacterset" : "AL16UTF16",
    "dbTerritory" : "AMERICA",
    "dbLanguage" : "AMERICAN"
  },
```

```
    "dbConsoleEnable" : false,
    "backupConfigId":null,
    "rmanBkupPassword": null
  },
  "asr" :{
    "asrType": "INTERNAL",
    "userName":"john.smith@example.com",
    "password":"password",
    "proxyServerName":"www-proxy.example.com",
    "proxyPort":"80",
    "proxyUserName":"proxy-user",
    "proxyPassword":"proxy-password",
    "httpsPortNumber":"16163"
  }
}
```

Example 5-2 JSON File to Create a Single Node Oracle Database Appliance without Role Separation and with Passwordless Multi-User Access

The following is an example of a JSON file that creates an Oracle Database Appliance without using role separation and with passwordless multi-user access. This example creates two groups (oinstall and dba) and one user (oracle).

```
{
  "instance" : {
    "name" : "odambox",
    "instanceBaseName" : "odambox-c",
    "dbEdition" : "EE",
    "timeZone" : "UTC",
    "ntpServers" : ["xx.x.x.xx"],
    "dnsServers" : ["xx.x.x.xx","xx.x.x.xx","xx.x.x.xx"],
    "domainName" : "example.com",
    "isRoleSeparated" : false,
    "isMultiUserAccessPLEnabled" : true,
    "osUserGroup" : {
      "groups" : [ {
        "groupId" : 1001,
        "groupName" : "oinstall",
        "groupRole" : "oinstall"
      }, {
        "groupId" : 1002,
        "groupName" : "dba",
        "groupRole" : "dba"
      } ],
        "multiUserAccess": {
            "isMultiUserAccessPLEnabled": true
        }
    },
  "nodes" : [ {
    "nodeNumber" : "0",
    "nodeName" : "odambox",
    "network" : [ {
       "nicName" : "btbond1",
       "ipAddress" : "xx.x.x.xx",
       "subNetMask" : "xxx.xxx.xxx.x",
       "gateway" : "xx.x.x.x",
```

```
      "networkType" : [ "Public" ],
      "isDefaultNetwork" : true
     },
  ],
  "ilom" : {
    "ilomName":"odambox-c",
    "ipAddress":"xx.x.x.xx",
    "subNetMask":"xxx.xxx.xxx.x",
    "gateway":"xx.x.x.x"
   }
} ],
"grid" : {
  "diskGroup" : [ {
    "diskGroupName" : "DATA",
    "redundancy" : "NORMAL",
    "diskPercentage" :80
  }, {
    "diskGroupName" : "RECO",
    "redundancy" : "NORMAL",
    "diskPercentage" :20
  } ],
"scan" : null,
"vip": null,
  "language" : "en",
  "enableAFD":"TRUE"
},
"database" : {
  "dbName" : "myTestDb",
  "databaseUniqueName":"myTestDb_sea1kj",
  "dbEdition" : "EE",
  "dbVersion" : "19.21.0.0",
  "dbHomeId":null,
  "instanceOnly" : false,
  "isCdb" : true,
  "pdBName" : "pdb1",
  "pdbAdminuserName" : "pdbuser",
  "enableTDE":true,
  "adminPassword" : "password",
  "dbType" : "SI",
  "dbTargetNodeNumber" : null,
  "dbClass" : "OLTP",
  "dbShape" : "odb1",
  "dbStorage" : "ACFS",
  "dbCharacterSet" : {
    "characterSet" : "AL32UTF8",
    "nlsCharacterset" : "AL16UTF16",
    "dbTerritory" : "AMERICA",
    "dbLanguage" : "AMERICAN"
  },
  "dbConsoleEnable" : false,
  "backupConfigId":null,
  "rmanBkupPassword": null
},
"asr" :{
  "asrType": "INTERNAL",
  "userName":"john.smith@example.com",
```

```
    "password":"password",
    "proxyServerName":"www-proxy.example.com",
    "proxyPort":"80",
    "proxyUserName":"proxy-user",
    "proxyPassword":"proxy-password",
    "httpsPortNumber":"16163"
  }
}
```

Example JSON File to Create a High-Availability Passwordless Multi-User Enabled Appliance

Follow the JSON file example to create a JSON file to deploy a high-availability appliance with the command `odacli create-appliance`.

> **Note:**
>
> It is important to review the readme and the examples carefully before creating your JSON file. If you do not enter your network and Oracle ILOM information correctly based on your setup, then network access is lost to both the host and Oracle ILOM.

Example 5-3 JSON File to Create a High-Availability Oracle Database Appliance with Role Separation and with Passwordless Multi-User Access

The following is an example of a JSON file that creates a high-availability appliance on Oracle Database Appliance bare metal platform. The example uses role separation and with passwordless multi-user access. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
   "instance":{
      "name":"odahabox",
      "instanceBaseName":"odahabox",
      "dbEdition":"EE",
      "timeZone":"UTC",
      "ntpServers" : ["10.0.3.14"],
      "dnsServers" : ["10.0.4.10","10.0.4.11","10.0.4.12"],
      "domainName":"example.com",
      "isRoleSeparated":true,
      "isMultiUserAccessPLEnabled" : true,
      "osUserGroup":{
         "groups":[
            {
               "groupId":1001,
               "groupName":"oinstall",
               "groupRole":"oinstall"
            },
            {
               "groupId":1002,
               "groupName":"dbaoper",
               "groupRole":"dbaoper"
            },
```

```
            {
               "groupId":1003,
               "groupName":"dba",
               "groupRole":"dba"
            },
            {
               "groupId":1004,
               "groupName":"asmadmin",
               "groupRole":"asmadmin"
            },
            {
               "groupId":1005,
               "groupName":"asmoper",
               "groupRole":"asmoper"
            },
            {
               "groupId":1006,
               "groupName":"asmdba",
               "groupRole":"asmdba"
            }
         ],
         "users":[
            {
               "userId":101,
               "userName":"grid",
               "userRole":"gridUser"
            },
            {
               "userId":102,
               "userName":"oracle",
               "userRole":"oracleUser"
            }
         ]
      },
      "users": [
                {
                    "id": 1000,
                    "name": "ugrid",
                    "role": "gridUser"
                },
                {
                    "id": 1001,
                    "name": "uoracle",
                    "role": "oracleUser"
                }
            ]
        },
        "multiUserAccess": {
            "isMultiUserAccessPLEnabled": true
        }
    },
      "objectStoreCredentials":null
  },
  "sysOraHomeStorage": {
       "volumeSize": "160",
       "diskgroup": "DATA"
```

```
 }
"nodes":[
   {
      "nodeNumber":"0",
      "nodeName":"odahaboxc1n2",
      "network":[
         {
            "nicName":"btbond1",
            "ipAddress":"10.31.98.133",
            "subNetMask":"255.255.240.0",
            "gateway":"10.31.96.1",
            "networkType":[
               "Public"
            ],
            "isDefaultNetwork":true
         }
      ],
      "ilom":{
         "ilomName":"odahabox2-c",
         "ipAddress":"10.31.16.140",
         "subNetMask":"255.255.240.0",
         "gateway":"10.31.16.1"
      }
   },
   {
      "nodeNumber":"1",
      "nodeName":"odahaboxc1n1",
      "network":[
         {
            "nicName":"btbond1",
            "ipAddress":"10.31.98.132",
            "subNetMask":"255.255.240.0",
            "gateway":"10.31.96.1",
            "networkType":[
               "Public"
            ],
            "isDefaultNetwork":true
         }
      ],
      "ilom":{
         "ilomName":"odahabox1-c",
         "ipAddress":"10.31.16.139",
         "subNetMask":"255.255.240.0",
         "gateway":"10.31.16.1"
      }
   }
],
"grid":{
   "diskGroup":[
      {
         "diskGroupName":"DATA",
         "redundancy":"HIGH",
         "diskPercentage":80
      },
      {
         "diskGroupName":"RECO",
```

```
            "redundancy":"HIGH",
            "diskPercentage":20
         },
      ],
      "scan":{
         "scanName":"odahaboxc1-scan",
         "ipAddresses":[
            "10.31.98.182",
            "10.31.98.183"
         ]
      },
      "vip":[
         {
            "nodeNumber":"0",
            "vipName":"odahaboxc1n2-vip",
            "ipAddress":"10.31.98.159"
         },
         {
            "nodeNumber":"1",
            "vipName":"odahaboxc1n1-vip",
            "ipAddress":"10.31.98.158"
         }
      ],
      "language":"en",
      "enableAFD":"TRUE"
   },
   "database":{
      "dbName":"myTestDb",
      "databaseUniqueName":"myTestDb_sea1kj",
      "dbVersion":"19.21.0.0",
      "dbHomeId":null,
      "instanceOnly":false,
      "isCdb":true,
      "pdBName":"pdb1",
      "pdbAdminuserName":"pdbuser",
      "enableTDE":true,
      "adminPassword":"password",
      "dbType":"RAC",
      "dbTargetNodeNumber":null,
      "dbClass":"OLTP",
      "dbShape":"odb1",
      "dbStorage":"ACFS",
      "dbCharacterSet":{
         "characterSet":"AL32UTF8",
         "nlsCharacterset":"AL16UTF16",
         "dbTerritory":"AMERICA",
         "dbLanguage":"AMERICAN"
      },
      "dbConsoleEnable":false,
      "backupConfigId":null
   },
   "asr":null
}
```

Example JSON Files to Create a Single-Node Passwordless Multi-User Enabled DB System

Follow the JSON file example to create a JSON file to deploy a single-node DB System, with role separation, with the command `odacli create-dbsystem`.

Use the example JSON file to create a file for your environment.

> **Note:**
>
> It is important to review the readme and the examples carefully before creating your JSON file.

Example 5-4 JSON File to Create a Single-Node Oracle KVM Database System with Role Separation and Passwordless Multi-User Access

The following is an example of a JSON file that creates a single-node Oracle KVM Database System on Oracle Database Appliance bare metal platform. The example uses role separation and passwordless multi-user access. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
```

```
                },
                {
                    "name": "asmoper",
                    "id": 1005,
                    "role": "asmoper"
                },
                {
                    "name": "asmdba",
                    "id": 1006,
                    "role": "asmdba"
                }
            ],
            "users": [
                {
                    "name": "grid",
                    "id": 1000,
                    "role": "gridUser"
                },
                {
                    "name": "oracle",
                    "id": 1001,
                    "role": "oracleUser"
                }
            ]
        "users": [
                {
                    "id": 1000,
                    "name": "ugrid",
                    "role": "gridUser"
                },
                {
                    "id": 1001,
                    "name": "uoracle",
                    "role": "oracleUser"
                }
            ]
        },
        "multiUserAccess": {
            "isMultiUserAccessPLEnabled": true
        }
    },
        }
    },
    "database": {
        "name": "db19",
        "uniqueName": "db19",
        "domainName": "example.com",
        "version": "19.25.0.0.241015",
        "edition": "EE",
        "type": "SI",
        "dbClass": "OLTP",
        "shape": "odb2",
        "role": "PRIMARY",
        "targetNodeNumber": null,
        "enableDbConsole": false,
        "enableFlashStorage": false,
```

```
        "enableFlashCache": false,
        "enableUnifiedAuditing": true,
        "enableEEHA": true,
        "enableSEHA": false,
        "redundancy": null,<<< if diskgroup redundancy is FLEX, then database
redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can be null
        "characterSet": {
            "characterSet": "AL32UTF8",
            "nlsCharacterset": "AL16UTF16",
            "dbTerritory": "AMERICA",
            "dbLanguage": "ENGLISH"
        },
        "enableTDE": false,
         "isCdb": true,
            "pdbName": "pdb1",
            "pdbAdminUser": "pdbadmin"
    },
    "network": {
        "domainName": "test_domain",
        "ntpServers": [
            "xx.xxx.xx.xxx"
        ],
        "dnsServers": [
            "xx.xxx.xx.xxx"
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xx.xxx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xx.x",
                "number": 0
            }
        ],
  "publicVNetwork": "vnet1"
    },
    "grid": {
        "language": "en",
        "enableAFD": true
    }
}
```

Example 5-5 JSON File to Create a Single-Node Oracle KVM Database System without Role Separation and with Passwordless Multi-User Access

The following is an example of a JSON file that creates a single-node Oracle KVM Database System on Oracle Database Appliance bare metal platform, without role separation and with passwordless multi-user access. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
```

```
            "diskGroup": "DATA",
            "cpuPoolName": "shared_dbsystem_cpupool",
        "database": {
            "name": "db19",
            "uniqueName": "db19",
            "domainName": "example.com",
            "version": "19.25.0.0.241015",
            "edition": "EE",
            "type": "SI",
            "dbClass": "OLTP",
            "shape": "odb4",
            "role": "PRIMARY",
            "targetNodeNumber": null,
            "enableDbConsole": false,
            "enableUnifiedAuditing": true,
            "redundancy": "HIGH",<<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
                "dbLanguage": "ENGLISH"
            },
            "enableTDE": false,
            "isCdb": true,
                "pdbName": "pdb1",
                "pdbAdminUser": "pdbadmin"
        },
         "users": [
                    {
                        "id": 1000,
                        "name": "ugrid",
                        "role": "gridUser"
                    },
                    {
                        "id": 1001,
                        "name": "uoracle",
                        "role": "oracleUser"
                    }
                ]
            },
            "multiUserAccess": {
                "isMultiUserAccessPLEnabled": true
            }
        },
        "network": {
            "domainName": "test_domain",
            "ntpServers": [],
            "dnsServers": [
                "xx.xxx.xx.xxx"
            ],
            "nodes": [
                {
                    "name": "node1",
                    "ipAddress": "xx.xx.xx.xxx",
```

```
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xx.x",
                "number": 0
            }
        ],
  "publicVNetwork": "vnet1"
    },
    "grid": {
        "language": "en",
         "enableAFD": true
    }
}
```

Example JSON File to Create a High-Availability Passwordless Multi-User Enabled DB System

Follow the JSON file example to create a JSON file to deploy a two-node DB System, with role separation, with the command `odacli create-dbsystem`.

Use the example JSON file to create a file for your environment.

Example 5-6 JSON File to Create Two-Node Oracle KVM Database System without Role Separation and with Passwordless Multi-User Access

The following is an example of a JSON file that creates two-node Oracle KVM Database System on Oracle Database Appliance bare metal platform, without role separation and with passwordless multi-user access. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_system",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        },
    "database": {
        "name": "dbtest",
        "uniqueName": "dbtest",
        "domainName": "test_domain",
        "version": "19.25.0.0.241015",
        "edition": "EE",
        "type": "SI",
        "dbClass": "OLTP",
        "shape": "odb2",
        "role": "PRIMARY",
        "targetNodeNumber": "0",
        "enableDbConsole": false,version
        "enableUnifiedAuditing": true,
        "enableEEHA": true,
        "redundancy": "MIRROR", <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
```

```
    "characterSet": {
        "characterSet": "AL32UTF8",
        "nlsCharacterset": "AL16UTF16",
        "dbTerritory": "AMERICA",
        "dbLanguage": "AMERICAN"
    },
     "enableTDE": false,
     "isCdb": true,
        "pdbName": "pdb1",
        "pdbAdminUser": "pdbadmin"
},
 "users": [
            {
                "id": 1000,
                "name": "ugrid",
                "role": "gridUser"
            },
            {
                "id": 1001,
                "name": "uoracle",
                "role": "oracleUser"
            }
        ]
    },
    "multiUserAccess": {
        "isMultiUserAccessPLEnabled": true
    }
},
"network": {
    "domainName": "test_domain",
    "ntpServers": [],
    "dnsServers": [
        "xx.xxx.xx.xxx"
    ],
    "nodes": [
        {
            "name": "node1",
            "ipAddress": "xx.xx.xxx.xx",
            "netmask": "xxx.xxx.xxx.x",
            "gateway": "xx.xx.xxx.x",
            "number": 0,
            "vipName": "node1-vip",
            "vipAddress": "xx.xx.xxx.xx"
        },
        {
            "name": "node2",
            "ipAddress": "xx.xx.xxx.xx",
            "netmask": "xxx.xxx.xxx.x",
            "gateway": "xx.xx.xxx.x",
            "number": 1,
            "vipName": "node2-vip",
            "vipAddress": "xx.xx.xxx.xx"
        }
    ],
    "publicVNetwork": "vnet1",
    "scanName": "test-scan",
```

```
        "scanIps": [
            "xx.xx.xxx.xx",
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en",
         "enableAFD": true
    }
}
```

Example 5-7 JSON File to Create Two-Node Oracle KVM Database System with Role Separation and with Passwordless Multi-User Access

The following is an example of a JSON file that creates two-node Oracle KVM Database System on Oracle Database Appliance bare metal platform. The example uses role separation and with passwordless multi-user access. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_system",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
                    "id": 1005,
                    "role": "asmoper"
                },
```

```
                {
                    "name": "asmdba",
                    "id": 1006,
                    "role": "asmdba"
                }
            ],
            "users": [
                {
                    "name": "grid",
                    "id": 1000,
                    "role": "gridUser"
                },
                {
                    "name": "oracle",
                    "id": 1001,
                    "role": "oracleUser"
                }
            ]
        }
    },
     "users": [
                {
                    "id": 1000,
                    "name": "ugrid",
                    "role": "gridUser"
                },
                {
                    "id": 1001,
                    "name": "uoracle",
                    "role": "oracleUser"
                }
            ]
        },
        "multiUserAccess": {
            "isMultiUserAccessPLEnabled": true
        }
    },
    "database": {
        "name": "dbtest",
        "uniqueName": "dbtest",
        "domainName": "test_domain",
        "version": "19.25.0.0.241015",
        "edition": "EE",
        "type": "SI",
        "dbClass": "OLTP",
        "shape": "odb2",
        "role": "PRIMARY",
        "targetNodeNumber": "0",
        "enableDbConsole": false,
        "enableUnifiedAuditing": true,
        "enableEEHA": true,
        "redundancy": null, <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
        "characterSet": {
            "characterSet": "AL32UTF8",
```

```
            "nlsCharacterset": "AL16UTF16",
            "dbTerritory": "AMERICA",
            "dbLanguage": "AMERICAN"
        },
        "enableTDE": false,
        "isCdb": true,
            "pdbName": "pdb1",
            "pdbAdminUser": "pdbadmin"
    },
    "network": {
        "domainName": "test_domain",
        "ntpServers": [],
        "dnsServers": [
            "xx.xxx.xx.xxx"
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xx.xx.xxx.xx"
            },
            {
                "name": "node2",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xx.xx.xxx.xx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "test-scan",
        "scanIps": [
            "xx.xx.xxx.xx",
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en",
         "enableAFD": true
    }
}
```

Provisioning Oracle Database Appliance with Multi-User Access

You can enable multi-user access **only** when you provision Oracle Database Appliance, using CLI commands or the Browser User Interface.

Provision your appliance as described in the chapter *Provisioning Oracle Database Appliance Bare Metal System*.

> **Caution:**
>
> You can enable multi-user access **only** at the time of provisioning Oracle Database Appliance on bare metal systems and **cannot** rollback multi-user access after you provision and deploy your appliance. The Oracle Database Appliance administration model changes when you deploy the Multi-User Access feature. Evaluate your role separation requirements before using this feature. Deploying on a test system first can help with the evaluation and assessment of the new administration model.

> **Note:**
>
> You can specify the token expiration duration, password expiration duration, maximum failed login attempts, and other details when you provision multi-user access enabled Oracle Database Appliance with JSON file. You can also specify these values as a one-time activity from the Browser User Interface when you configure multi-user access and provision Oracle Database Appliance using the Browser User Interface.

Overall Steps in Provisioning Oracle Database Appliance with Multi-User Access Enabled

1. Enable multi-user access.
   - If you provision your appliance using a JSON file, then add the attribute `"isMultiUserAccessEnabled": true` in the `prov_req.json` file. If the attribute is set to `false` or does not exist in the `prov_req.json` file, then multi-user access is not enabled during provisioning of the appliance.

```
"isRoleSeparated": true,
        "isMultiUserAccessEnabled": true,
        "osUserGroup": {
            "groups": [{
                    "groupId": 1001,
                    "groupName": "oinstall",
                    "groupRole": "oinstall"
                },

...
```

You can also set the multi-user access attributes by adding the following in the JSON file:

```
},
  "asr": null,
  "multiUserAccess": {
    "dcsUserPasswdExpDurationInDays": 90,
      "tokenExpirationInMins": 120,
    "maxNumFailedLoginAttempts": 3
  }
}
```

The values for these attributes are as follows:

  - **Token expiration duration in minutes:** The minimum value you can specify is 10 minutes, the maximum value is 600 minutes, and the default is 120 minutes.
  - **Password expiration duration in days:** The minimum value you can specify is 30 days, the maximum value is 180 days, and the default is 90 days.
  - **Maximum failed login attempts allowed:** The minimum value you can specify is 2, the maximum value is 5, and the default is 3.
- If you create the appliance using the Browser User Interface (BUI), then select the **Enable Multi-User Access (N/A for DB System)** checkbox in the BUI login page.

2. Provide passwords for `odaadmin`, `oracle`, and `grid` users. These are Oracle Database Appliance system users and their accounts are activated during created. The user `odaadmin` is created with the role of `ODA-ADMINISTRATOR` while the `oracle` and `grid` users are created with the role of `ODA_DB` and `ODA_GRID` respectively.
3. The system configures the multi-user access repository with a list of roles and entitlements, used for assigning to the users in the system.
4. You can now log into the appliance with the newly-created user credentials and deploy databases.

Provisioning Oracle Database Appliance Using the Browser User Interface with Multi-User Access Enabled

1. Access the Browser User Interface:

   `https://`*`host-ip-address`*`:7093/mgmt/index.html`

2. For the first login, since the `odaadmin` role is not configured, you are prompted to provide the ODA password and enable multi-user access.
3. Select **Enable Multi-User Access (N/A for DB System)** and provide a strong password for the ODA user.
4. Click **Configure Multi-User Settings** and then set the **User Password Expiry Duration (In Days)**, **Session Expiration for CLI (minutes)**, and **Maximum Failed Login Attempts**. Click **Save** to save these settings and return to the Login page on the BUI.
5. Click **Submit**. A confirmation message is displayed on successful creation of the user.
6. Click **OK**. You are redirected to the Login page.
7. Specify the **User Name** and **ODA Password** and click **Login**. Note that the ODA admin user name is `odaadmin` if multi-user access is enabled. If multi-user access is not enabled, then the ODA admin user name is `oda-admin`.
8. In the Create Appliance page, specify the details for creating the appliance. See the topic *Creating the Appliance* for detailed information about the information you need to provide.
9. Select **Assign same password for admin, oracle, grid users** if you want to specify the same password for all users. Otherwise, specify different passwords for the `system admin`, `oracle`, and `grid` users.
10. Click **Submit** to create the appliance with multi-user access enabled.
11. The job is submitted and a confirmation page appears with a link to the job. Click the link to view the job progress, tasks, and status. After you close the Job confirmation page, you can click the **Activity** tab to monitor the job progress. Click the job number to view the tasks and status details. Click **Refresh** to refresh the page.

Related Topics

- Provisioning Oracle Database Appliance Bare Metal System
  Understand the process to configure Oracle Database Appliance bare metal system.
- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Creating, Viewing, and Deleting Users on Oracle Database Appliance with Multi-User Access

After provisioning the appliance with multi-user access enabled, you can create users with specific entitlements.

After you provision your appliance with multi-user access enabled, do the following:

Creating, Viewing, and Deleting Users with ODACLI Commands

1. Connect to the appliance as the `odaadmin` user.

   ```
   ssh odaadmin@oda-box hostname/IP
   ```

2. Run any ODACLI command and provide the `odaadmin` password when prompted.
3. On successful authentication, create a user with the following command:

   ```
   odacli create-user -u username -r comma-separated role names
   ```

   For example, create a user `dbuser1` with lifecycle management privileges for dabatases:

   ```
   odacli create-user -u dbuser1 -r ODA-DB
   ```

   The `odaadmin` user creates `dbuser1` and assigns a temporary password.
4. After the user is created successfully, the `dbuser1` can log into the appliance with the temporary password.
5. The `dbuser1` is in the `Inactive` state. Activate the user with the following command:

   ```
   odacli activate-user
   ```

   You are prompted to change the password. Enter the temporary password, the new password, and confirm the new password.
6. Use the new password to connect by SSH into the appliance and run ODACLI commands or connect to the Browser User Interface.
7. The `odaadmin` user can view all the users in the system:

   ```
   # odacli list-users
   ```

8. The `odaadmin` user can view details for a user in the system:

   ```
   # odacli describe-user -u user_id
   ```

9. Delete a user in the system. Note that only the `odaadmin` user can delete a user in the system.

   ```
   # odacli delete-user -u user_id
   ```

Creating, Viewing, and Deleting Users with Browser User Interface

1. Log into the Browser User Interface as the `odaadmin` user:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Multi-User Access** tab.
3. Click the Users link on the left-hand pane.
4. Click **Create User**.
5. In the Create User page, provide the **User ID**, specify the **Role**, and provide the **ODA Password** for this user. Note that the same user credentials work for login for BUI and ODACLI commands.
6. Optionally, click **Generate mTLS Certificate** to enable mTLS-based authentication.
7. Click **Create**.
8. The job is submitted and a confirmation page appears with a link to the job. Click the link to view the job progress, tasks, and status. After you close the Job confirmation page, you can click the **Activity** tab to monitor the job progress. Click the job number to view the tasks and status details. Click **Refresh** to refresh the page.
9. In the **Multi-User Access** tab, on the Users page, click on the link for the user whose details you want to view.
10. To delete a user, log in as the the `odaadmin` user. In the **Actions** drop-down list, select **Delete**. Note that only a user of type **Custom** can be deleted.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Activating a New User on Oracle Database Appliance with Multi-User Access

Understand how to activate a new user on multi-user access enabled Oracle Database Appliance.

Activating the New User on Multi-User Access Enabled System Using ODACLI Commands

1. After a new user is created by `odaadmin` successfully, the new user, for example, `dbuser1` can log into the appliance with the temporary password.
2. Activate the user with the following command:

   ```
   odacli activate-user
   ```

You are prompted to change the password. Enter the temporary password, the new password, and confirm the new password.

Activating the New User on Multi-User Access Enabled System Using Browser User Interface

1. After a new user is created by `odaadmin`, log into the Browser User Interface as the new user:

   `https://host-ip-address:7093/mgmt/index.html`

2. Specify the **User Name** and the temporary password in the **ODA Password** field.
3. Since this is a new account, the Account Status is `Inactive`. You are prompted to specify and confirm a new password.
4. Specify and confirm the **Password** and click **Submit**.
5. On successful password change, log into the Browser User Interface with the new password.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Granting and Revoking Resource Access on Oracle Database Appliance with Multi-User Access

You can grant and revoke resource access on the appliance.

Granting and Revoking Resource Access with ODACLI Commands

- Grant or revoke access to a resource in a multi-user access enabled system:

  ```
  # odacli grant-resource-access -ri resource_ID -u user_name
  # odacli revoke-resource-access -ri resource_ID -u user_name
  ```

- View access to a DCS resource in a multi-user access system:

  ```
  # odacli describe-resource-access -ri resource_ID
  ```

- View access to all DCS resources defined in a multi-user access system:

  ```
  # odacli list-resources-access -ao -rn resource_name -rt resource_type
  ```

Granting and Revoking Resource Access with Browser User Interface

1. Log into the Browser User Interface as `odaadmin`:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Multi-User Access** tab.
3. Click the Resources link on the left-hand pane.
4. Click on a Resource to view more details.

5. For a Resource, in the Actions drop down list, select **Grant Resource Access** to grant the user shared access to a resource. Select the **User Name** from the drop-down list and click **Grant**. Click **Yes** to confirm and submit the job.
6. Select **Revoke Resource Access** to revoke access to a resource from a user. Select the **User Name** from the drop-down list and click **Revoke**. Click **Yes** to confirm and submit the job.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Viewing Roles, Operations, and Entitlements on Oracle Database Appliance with Multi-User Access

You can view roles and entitlements on the appliance.

Viewing Roles, Operations, and Entitlements with ODACLI Commands

> **Note:**
>
> For Oracle Database Appliance release 19.13, the multi-user access feature is available for standalone Oracle Database Appliance systems. During provisioning, a single domain and tenancy is created by default and all users are created within the default domain and tenancy.

- View all the roles defined in the system:

```
# odacli list-user-roles
```

- View details for a user role in the system:

```
# odacli describe-user-role -n role_name
```

- View all the entitlements defined in the system:

```
# odacli list-user-entitlements
```

- View details for an entitlement in the system:

```
# odacli describe-user-entitlement -n entitlement_name
```

- View all the operations defined in the system:

```
# odacli list-user-operations
```

- View details for an operation in the system:

```
# odacli describe-user-operation -n operation_name
```

- View the domains defined in the system. In this release, this is the default domain.

  ```
  # odacli list-domains
  ```

- View details for a domain in the system:

  ```
  # odacli describe-domain -dn domain_name
  ```

- View the tenants in a multi-user access enabled domain. In this release, this is the default tenancy.

  ```
  # odacli list-tenants
  ```

- View details for a tenant in a multi-user access enabled domain:

  ```
  # odacli describe-tenant -tn tenant_name
  ```

Viewing Roles, Operations, and Entitlements with Browser User Interface

1. Log into the Browser User Interface as `odaadmin`:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Multi-User Access** tab.
3. Click the Roles link on the left-hand pane. The roles defined in the system are displayed. These roles cannot be edited or updated.
4. Click on a Role to view more details.
5. Click the Entitlements link on the left-hand pane. The entitlements defined in the system are displayed. These entitlements cannot be edited or updated.
6. Click on an Entitlement to view more details.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Managing Databases and Database Homes on Oracle Database Appliance with Multi-User Access

The custom user created on multi-user access Oracle Database Appliance can deploy and manage databases and database homes.

After you create a custom `dbuser1` on an Oracle Database Appliance with multi-user access enabled, manage databases as follows:

Creating and Listing Databases and Database Homes Using ODACLI Commands

1. Connect to the appliance as `dbuser1`.

   ```
   ssh dbuser1@oda-box hostname/IP
   ```

2. Create a database:

   ```
   odacli create-database -n dbName -v dbVersion
   ```

3. Run the `odacli list-databases` to view the databases owned by `dbuser1`:

   ```
   odacli list-databases
   ```

   Another user with the `ODA-DB` role cannot use the resource owned by `dbuser1` to create a database home, thus ensuring role separation.

4. Use the `-all` option on the appliance when multi-user access is enabled to view all the databases in the system.

   ```
   odacli list-databases -all
   ```

5. Use the `-all` option on the appliance when multi-user access is enabled to view all the database homes in the system.

   ```
   odacli list-dbhomes -all
   ```

Creating and Listing Databases and Database Homes Using Browser User Interface

1. Log into the Browser User Interface as `dbuser1`:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Database** tab.
3. Click **Show All Databases**. A list of all databases in the system is displayed.
   - **View Details:** View the details of the database.
   - **Modify:** Modify the database
   - **Move:** Move the database
   - **Upgrade:** Upgrade the database
   - **Delete:** Delete the database.
   - **Grant Access:** Grant access privilege to the database to a user.
   - **Revoke Access:** Revoke access privilege to the database from a user.
   - **View Pre-patch reports:** View prechecks report for patching.
4. Click the Database Home link on the left hand pane. A list of all database homes in the system is displayed.
5. For a database home, you can click the **Actions** dropdown list, and select from the following options:
   - **View Database:** View the details of the database home.
   - **Delete:** Delete the database.
   - **Grant Access:** Grant access privilege to the database home to a user.
   - **Revoke Access:** Revoke access privilege to the database home from a user.
   - **View Pre-patch reports:** View prechecks report for patching.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Changing the Password for a User Account on Oracle Database Appliance with Multi-User Access

Understand how to manage passwords on multi-user access Oracle Database Appliance.

Changing the Password on Multi-User Access Enabled System Using ODACLI Commands

- You can change password for an Oracle Database Appliance user, whose account is active:

  ```
  odacli change-password
  ```

Changing the Password on Multi-User Access Enabled System Using Browser User Interface

1. Log into the Browser User Interface as the user whose password you want to change:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. To change the Account password at any time: Click on the Account drop down list in the top right-hand side of Browser User Interface and select **Change Password**.
3. Specify the **Old Password** and also specify and confirm the new **Password** and click **Submit**.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Resetting the Password for a Locked User Account on Oracle Database Appliance with Multi-User Access

Understand how to reset the password on multi-user access Oracle Database Appliance.

Resetting Password for a Locked User Account on Multi-User Access Enabled System Using ODACLI Commands

> **Note:**
>
> If your appliance is configured as passwordless multi-user enabled, then all user accounts are preactivated during creation and never get locked because the system internally generates the password for each user and provides it to the DCS agent for authentication when required for running ODACLI commands. For passwordless multi-user enabled systems, do **not** run any scripts for unlocking the `odaadmin` account or the `odacli reset-password` command to reset the password. Run the `odacli change-password` command to change the system generated password for your Oracle Database Appliance account.

- Unlock the `odaadmin` user account that is locked due to multiple failed login attempts or password expiry.
  1. Log in as `root`.
  2. Run the following:

     ```
     /opt/oracle/dcs/bin/resetCredsForOdaAdmin.sh
     ```

     A temporary password is assigned to the `odaadmin` user.
  3. Log in as the `odaadmin` user with the temporary password.
  4. Run the following command:

     ```
     odacli reset-password
     ```

     You are prompted to provide the temporary password and specify and confirm the new password. After the command runs successfully, the user account is unlocked.
- Unlock any non-admin user account that is locked due to multiple failed login attempts or password expiry.
  1. Log in as `odaadmin`.
  2. Run the following command:

     ```
     odacli authorize-user
     ```

     After you provide a temporary password, the account is unlocked.
  3. Log in as the user whose account was locked, with the temporary password.
  4. Run the following command:

     ```
     odacli reset-password
     ```

     You are prompted to provide the old password, the temporary password and specify and confirm the new password. After the command runs successfully, the user account is unlocked and reactivated.

Resetting Password for a Locked User Account on Multi-User Access Enabled System Using Browser User Interface

- Unlock the non-admin user account that is locked due to multiple failed login attempts or password expiry as follows:
  1. Log into the Browser User Interface as the `odaadmin` user:

     `https://host-ip-address:7093/mgmt/index.html`

  2. In the **Multi-User Access** tab, on the Users page, click on the link for the user whose password you want to reset. Note that you can reset the password for users of type **Custom** only. The Account Status for the user is `LockedFailedLogin`.
  3. In the Actions drop down list, select **Authorize Password Reset**.
  4. In the Authorize Password Reset page, specify the **Old Password**, specify and confirm the **Temporary ODA Password**, and click **Authorize**.
  5. Now, log into the Browser User Interface as the user whose account is being unlocked. Specify the **User Name** and the temporary password in the **ODA Password** field.
  6. Since the account was locked, the Account Status is `CredentialReset`. You are prompted to specify and confirm a new password.
  7. Specify and confirm the **Password** and click **Submit**.
  8. On successful password change, log into the Browser User Interface with the new password.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

6

Oracle Database Appliance Postinstallation Tasks

Complete these administrative tasks after you have deployed software, but before the system is operational.

- Configuring CPU Core Count
  Oracle Database Appliance is delivered with all cores on each server enabled. Follow this procedure to reduce the number of cores, if required.
- Modifying Oracle ASM Listener Port After Deployment
  Understand how you can modify Oracle ASM listener port on Oracle Database Appliance after deployment.
- Securing Oracle ILOM Service Processors
  Change the Oracle ILOM default password after completing Oracle Database Appliance deployment.
- Changing Oracle Database Appliance Passwords
  After deploying your appliance, ensure that you change the following passwords for securing your system.
- Changing the DNS Server Address on Oracle Database Appliance
  If the DNS server in the data center changes, understand how you can update the DNS entries on on Oracle Database Appliance server so that the network functions correctly.

Related Topics

- Deployment Checklist for Oracle Database Appliance Bare Metal Systems
  This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.

Configuring CPU Core Count

Oracle Database Appliance is delivered with all cores on each server enabled. Follow this procedure to reduce the number of cores, if required.

1. To reduce the number of cores, run the following command on Node 0 only:

   Set the cores in multiples of 2, for example, 12:

   ```
   [root@oak1 opt]# /opt/oracle/dcs/bin/odacli modify-cpucore --cores 12
   ```

   When you run the command, the nodes are restarted to apply the changes.
2. Check if the job completed successfully:

   ```
   [root@oak1 opt]# /opt/oracle/dcs/bin/odacli describe-job -i job_ID
   ```

3. Verify that the core count is updated on both nodes:

```
[root@oak1 opt]# /opt/oracle/dcs/bin/odacli describe-cpucore
```

Modifying Oracle ASM Listener Port After Deployment

Understand how you can modify Oracle ASM listener port on Oracle Database Appliance after deployment.

In earlier Oracle Database Appliance releases, you provisioned your bare metal system with Oracle ASM listener port 1525 as the default and as the Oracle ASM discovery address port from DB systems. Starting with Oracle Database Appliance release 19.24, you can customize Oracle ASM listener port number as a postinstallation task. All DB systems that you provision after changing the Oracle ASM listener port number use the new Oracle ASM port number.

Use the following command to modify the Oracle ASM listener port number:

```
odacli modify-asmport -p any_unused_port_between_1024_and_65536
```

For example:

```
# odacli modify-asmport -p 1528
```

> **Note:**
>
> You can run the `odacli modify-asmport` command only on deployments where the bare metal system is on Oracle Database Appliance release 19.24 and the DB system is on release 19.24 and you use the DB 23ai clones available with Oracle Database Appliance release 19.24.

Related Topics

- Modifying Oracle ASM Listener Port After Deployment
  Understand how you can modify Oracle ASM listener port on Oracle Database Appliance after deployment.

Securing Oracle ILOM Service Processors

Change the Oracle ILOM default password after completing Oracle Database Appliance deployment.

Do not change the default password until after you have completed software deployment on the Oracle Database Appliance.

Changing Oracle ILOM Password from the Console

1. In the Oracle ILOM console, from the **Administration** menu, select **User Management**, and then navigate to the **User Accounts** subtab.
2. Select **root** user and click **Edit**.
3. Change the `root` user password.

Changing Oracle ILOM Password Using CLI Commands

1. Connect to the Oracle ILOM service processor (SP) through SSH:

   ```
   # ssh -l root SP-ipaddr
   ```

2. Set the new password:

   ```
   -> set /SP/users/root password=new_password
   Changing password for user /SP/users/root/password...
   Enter new password again: *********
   New password was successfully set for user /SP/users/root
   ```

Changing Oracle Database Appliance Passwords

After deploying your appliance, ensure that you change the following passwords for securing your system.

Changing the Oracle Installation Owner Passwords

During deployment, the root and database users SYS, SYSTEM and PDBADMIN are set to the system password. After deployment, the oracle and grid passwords are also set to the system password. Change the passwords to comply with your enterprise user security protocols. Refer to the *Oracle Database Appliance Security Guide* and *Oracle Database Security Guide* for information about the required configuration and best practices to secure database systems.

Changing the oda-admin User Password through the Command-Line

1. Log in to the appliance as `root`.
2. Run the `odacli set-credential` command to reset the password. Enter the new password when prompted.

   ```
   # odacli set-credential --username oda-admin
   Enter 'user' password:
   Retype 'user' password:
   ```

Changing the the oda-admin User Password through the Browser User Interface

1. Log into the Browser User Interface using the user name `oda-admin`.
2. Click **About**, then **User Settings** in the upper right corner of the Browser User Interface.
3. Enter the password in the Password field and the Password Confirmation field, then click **Submit**.

   A confirmation message is displayed.
4. Click **About**, then click **Sign Out**.
5. Log back into the Browser User Interface with the new password.

> **Note:**
>
> The `oda-admin` password expiration period is 90 days.

Changing the DNS Server Address on Oracle Database Appliance

If the DNS server in the data center changes, understand how you can update the DNS entries on on Oracle Database Appliance server so that the network functions correctly.

Follow these steps to change the DNS server address on Oracle Database Appliance:

1. Log in as the `root` user.
2. Open the `/etc/resolv.conf` file in a text editor.
3. Locate the name server setting for the DNS server and change it to the new value. For example:

```
search        example.com
nameserver    10.7.7.3
```

4. Save the file and close the editor.

7

Re-imaging Oracle Database Appliance

Bare metal operating system re-imaging installs Oracle Database Appliance operating system software on the local (boot) drive.

Bare Metal is a non-virtualized Oracle Database Appliance configuration. Oracle Database Appliance ships from the factory with a bare metal configuration, default ISO image and Appliance Manager installed. Use the OS ISO Image to restore the OS to the "shipped from factory" state. Use only when necessary. Reimaging does not patch the firmware or update the component versions; it only re-images the local system disk from an operating system perspective. After imaging is completed, provision the bare metal system.

- Uninstalling Oracle Database Appliance Components
  Use the Oracle Database Appliance cleanup deploy utility, `/opt/oracle/oak/onecmd/cleanup.pl` to uninstall Oracle Database Appliance components.
- Reimaging an Oracle Database Appliance Baremetal System
  Reimage Oracle Database Appliance to perform a bare metal restore of Oracle Database Appliance.
- Performing Secure Erase of Data on Storage Disks
  With this release, you can securely erase data from storage devices. Running the secure erase tool deletes the data on the storage disk permanently.
- Redeployment of Oracle Database Appliance
  Understand the process to redeploy Oracle Database Appliance.

Related Topics

- Errors when re-imaging Oracle Database Appliance
  Understand how to troubleshoot errors that occur when re-imaging Oracle Database Appliance.

Uninstalling Oracle Database Appliance Components

Use the Oracle Database Appliance cleanup deploy utility, `/opt/oracle/oak/onecmd/cleanup.pl` to uninstall Oracle Database Appliance components.

About the Cleanup Utility

Use the cleanup deploy utility to do the following:

- Uninstall Oracle Auto Service Request (Oracle ASR)
- Uninstall Oracle Trace File Analyzer (TFA)
- Uninstall Oracle ORAchk Health Check Tool
- Uninstall Oracle Grid Infrastructure and the Oracle stack
- Reset or delete the Oracle Linux udev rules
- Delete users and groups that were created when the appliance was created.

The script removes the firstnet config and the client access VLAN UNIX_DB, but does not delete any other VLANs. The script also resets any disabled CPU cores and enables all of the cores.

The cleanup utility runs in two modes: default and force mode. In the default mode, the commands used as part of cleanup do not use the force option. The Oracle ASM disk status is changed from MEMBER to FORMER, but ASM header on the disk is not erased. The default mode can be used only for cleaning up or re-provisioning of the same system.

> **Note:**
>
> For High-Availability systems, run the cleanup utility on both nodes sequentially. Ensure that the cleanup utility has completed on the first node, and then start the process on the second node.

Running the Cleanup Utility for a Baremetal Deployment

```
# perl /opt/oracle/oak/onecmd/cleanup.pl [-griduser grid_user] [-dbuser
db_user] [-groups comma separated list of groups] [-omausers
dbuser1,dbuser2,dbuser3][-erasedata][-nodpr] [-nossh] [f]
```

When the *grid_user* and *db_user* are the same (`roleSeparation=false`), you must run the script for each user (`-gridUser` and `-dbUser`).

For example:

```
cleanup.pl -griduser grid -dbuser oracle -omausers dbuser1,dbuser2,dbuser3
```

Table 7-1 Command Options for Cleanup Utility

Option	Description
grid_user	Describes the Oracle Grid Infrastructure user name. The default user is `grid`.
db_user	Describes the database user name. The default user is `oracle` Example with `grid` and `oracle` users: `cleanup.pl -griduser grid -dbuser oracle`
groups	Describes the comma-separated list of groups. The default groups are `oinstall,dba,dbaoper,asmadmin,asmoper,asmdba`.
omausers	Describes the comma-separated list of custom users created on a multi-user access enabled Oracle Database Appliance environment. All custom users created after provisioning a multi-user access enabled environment must be removed completely before you reprovision the appliance.

Table 7-1 (Cont.) Command Options for Cleanup Utility

Option	Description
`erasedata`	Completely erases all disks on the Oracle Database system. It is mandatory to run this option if intention of cleanup is to reuse or move these disks on other systems. This option needs to be run on the first node when running on high-availability systems. Example of secure erase: `cleanup.pl -erasedata`
`cleanDefNet`	Cleans up the default public network.
`checkHeader`	Checks for `OAK/ASM` header on disks after successfully running the cleanup script. Use this to validate if the `OAK/ASM` header was erased by the cleanup script or not. Example of checking disk header: `cleanup.pl -checkHeader`
`nodpr`	Forces regular cleanup even if Data Preserving Reprovisioning environment is detected.Wipes out all data on Oracle ASM disks. You cannot run the command `odacli restore-node -g` after this operation. On Data Preserving Reprovisioning environments, the default cleanup mode is Data Preserving Reprovisioning mode which cleans the appliance without erasing Oracle SM disks to allow the command `odacli restore-node -g` to be reattempted. Using the `-nodpr` option allows you to override this behaviour and force regular cleanup. Example: `cleanup.pl -nodpr`
`nossh`	Does not attempt to set up SSH on high-availability deployments. Use this option in cases such as when the scripts using **cleanup.pl** are not updated to handle stopping DCS agent, or if you have forgotten the SSH password, or if SSH is not set up on the system. Example: `cleanup.pl -nossh`

Table 7-1 (Cont.) Command Options for Cleanup Utility

Option	Description

`f`	In the force mode, all commands use the force option. The ASM disk header is erased. The `oakd` header on the disk is erased in both default and force modes.
	Example of default mode:
	`cleanup.pl`
	Example of force mode:
	`cleanup.pl -f`

> **Note:**
>
> The cleanup utility for bare metal deployments produces a log file. The utility tries to create the log file at `/opt/oracle/oak/log/`*`hostname`*`/cleanup/cleanup_`*`timestamp`*`.log`. If the log cannot be created in that location, then the log is created at `/tmp/cleanup_`*`timestamp`*`.log`.

Related Topics

- Deployment Checklist for Oracle Database Appliance Bare Metal Systems
  This checklist provides an overview of the tasks to setup and deploy Oracle Database Appliance bare metal systems for the first time.

Reimaging an Oracle Database Appliance Baremetal System

Reimage Oracle Database Appliance to perform a bare metal restore of Oracle Database Appliance.

Bare metal restore uses Oracle Integrated Lights Out Manager (Oracle ILOM) to re-image the Oracle Database Appliance node. The ILOM must be configured before performing a bare metal restore or reimage. Generally, the ILOM is configured as part of readying for deploying Oracle Database Appliance.

Download the Oracle Database Appliance Bare Metal ISO image to your local machine before you launch the Oracle ILOM console.

Follow these steps to reimage your appliance. For Oracle Database Appliance High-Availability deployment, follow these steps on **both** nodes.

1. Open a browser and connect to Oracle Integrated Lights Out Manager (ILOM) on Node 0 as root.

   https://*ilom-ip-address*

2. Launch the Remote Console.

   a. Expand **Remote Control** in the left navigation.

b. Click the **Redirection** tab.

c. Click **Launch** for the Remote Console in the Actions menu.

   The state of the system determines what appears on the Console page.

3. Add the image.

   a. Click the **KVMS** tab, then select **Storage**.

   b. Click **Add**.

   c. Browse to the Oracle Database Appliance Bare Metal ISO Image, highlight the image, then click **Select**.

   d. Click **Connect**.

      The mounting of the ISO image is successful when the **Connect** button changes to a **Disconnect** button.

   e. Click **OK**

      The CD-ROM icon in the top right corner is highlighted.

4. Configure the CD-ROM as the next boot device.

   a. Expand **Host Management** in the left menu of the ILOM Remote Console tab.

   b. Click **Host Control**.

   c. Select **CDROM** from the Next Boot Device menu, then click **Save**.

5. Power cycle the node.

   a. Click **Power Control** in the **Host Management** menu.

   b. Select **Power Cycle** , then click **Save**.

   When the node comes back after the power cycle, re-imaging starts automatically. The Oracle Linux page appears, followed by the Running Post-Install scripts page.

   The Running Post-Install scripts page is a static page and might give the impression that the reimaging process is not progressing. The post-install process during re-imaging will wait until the synchronization between the partitions of the two mirrored local hard disks is complete, which can take 15 to 20 minutes to complete.

   To check the progress of re-synchronization, press the ALT-F2 key combination to open a second console and enter the following command:

   ```
   # cat /proc/mdstat
   ```

   When the re-synchronization is complete, re-imaging is completed and the machine restarts.

After the machine restarts, the system is ready for you to deploy the Oracle software on the appliance to create an Oracle Database Appliance bare metal platform.

Related Topics

- Creating the Appliance
  Create the appliance using the Browser User Interface or ODACLI commands.

Performing Secure Erase of Data on Storage Disks

With this release, you can securely erase data from storage devices. Running the secure erase tool deletes the data on the storage disk permanently.

You may want to consolidate storage disks between appliances. For example, you can use X7-2 disks on X6-2 models, and may want to consolidate all X6-2 disks on one system, and use all new X7-2 storage disks on another system. In such cases, use secure erase to erase headers from disks before re-imaging the disks, since the storage disks across the appliances may have different partition ratios.

> **Note:**
>
> The secure erase tool conforms to the NIST 800-88 standard, also called NIST Special Publication 800-88 (NIST SP 800-88), Guidelines for Media Sanitization. Running the secure erase tool removes data from storage disks permanently. If you have any data on the disk, then take a backup of your storage disk before running the secure erase tool.

Follow the steps to run the secure erase tool:

1. Stop `oakd` and the database and Oracle Grid Infrastructure before running the secure erase tool.

   On bare metal systems, run the command:

   ```
   # odaadmcli stop oak
   ```

   On Virtualized Platforms:

   ```
   # oakcli stop oak
   ```

   To stop Oracle Clusterware resources:

   ```
   # crsctl stop crs -f
   ```

2. Run the secure erase tool:

   ```
   # /opt/oracle/oak/bin/odaeraser.py
   ```

3. You can also run the secure erase tool when running the cleanup tool:

   ```
   # /opt/oracle/oak/onecmd/cleanup.pl -erasedata
   ```

Example 7-1 Options for the Secure Erase Tool

Option	Description
`-a, --all`	Erases all disks. For example: `# /opt/oracle/oak/bin/odaeraser.py --all`
`--disk disk1,disk2`	Erases specified disks. For example: `# /opt/oracle/oak/bin/odaeraser.py --disk e0_pd_00,e0_pd_01`
`--dryrun`	Runs the tool in the test mode
`--type HDD|SSD|NVMe`	Erases the type of disk specified
`-v, --verbose`	Displays verbose output
`-h, --help`	Displays all CLI options for the tool

Redeployment of Oracle Database Appliance

Understand the process to redeploy Oracle Database Appliance.

After cleaning up the system and erasing the header, you can continue deployment of Oracle Database Appliance as follows:

1. Download Oracle Database Appliance software.
2. Install Oracle Database Appliance software.
3. Create the appliance.

For the above steps, run the procedures as described in this guide.

8

Patching Oracle Database Appliance

To keep Oracle Database Appliance running with the latest software, check for and apply Oracle Database Appliance patch bundles when they are released.

- About Patching Oracle Database Appliance
  Use the Oracle Database Appliance Patch Bundle to patch your appliance.
- About Patching Prechecks
  Patching prechecks are designed to detect and flag problems that might be encountered during patching.
- About Enabling NUMA on Oracle Database Appliance
  Understand how you can enable NUMA on bare metal, KVM, and DB Systems on Oracle Database Appliance.
- Minimizing Downtime When Patching Oracle Database Appliance
  Understand best practices to minimize downtime when you apply patches to your Oracle Database Appliance deployment.
- About Updating Databases and Database Homes
  Understand how to update a database and database home using either the Browser User Interface or `odacli` commands.
- Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line
  Follow these steps to apply patches to your Oracle Database Appliance bare metal deployment and existing Oracle Database homes, using CLI commands.
- Patching Oracle Database Appliance DB Systems Using the Command-Line
  Follow these steps to apply patches to your Oracle Database Appliance DB system deployment and existing Oracle Database home in the DB system, using CLI commands.
- Patching Oracle Database Appliance Bare Metal Systems Using the Browser User Interface
  Upload the Oracle Database Appliance Server Patch to the patch repository, deploy the patch bundle using the Browser User Interface, and then update the DCS admin, DCS components, and DCS agent using CLI commands.
- Patching Oracle Database Appliance DB Systems Using the Browser User Interface
  Upload the Oracle Database Appliance Server Patch to the patch repository, deploy the patch bundle using the Browser User Interface, and then update the DCS agent and components using CLI commands.
- Updating Oracle Database Appliance Repository with Database Clone Files For Bare Metal Systems
  Follow these steps to update the Oracle Database Appliance repository with Oracle Database clone files for the latest release using CLI commands.
- Patching Databases Using ODACLI Commands or the BUI
  Use ODACLI commands or the Browser User Interface to patch databases to the latest release in your deployment.
- Patching Existing Database Homes Using ODACLI or the BUI
  Use ODACLI or BUI to patch database homes in your deployment to the latest release.

- Patching Non-Provisioned Oracle Database Appliance Systems
  Oracle Database Appliance supports patching of non-provisioned single-node or high availablity systems by running `odacli update-server` command.
- Applying Additional Patches and Updates on Bare Metal Systems
  Configure additional patch repositories and patch your operating system and databases to ensure that your deployment has the latest updates and security fixes.
- Cleaning Up the Patch Repository
  Use the Browser User Interface or Command Line Interface to delete obsolete or old patches from the repository.
- About Upgrading to a Different Database Home
  Understand how to upgrade to a different database home using either the Browser User Interface or `odacli` commands.
- Setting Dynamic Default Parameters
  You can set dynamic default parameters using the BUI.

About Patching Oracle Database Appliance

Use the Oracle Database Appliance Patch Bundle to patch your appliance.

Patches offer new features and may improve the functionality of existing features.

About Out of Place Patching

Starting with Oracle Database Appliance release 19.11, Oracle Database Appliance uses the out-of-place patching model. This means that a new software Oracle home is created when patching an Oracle Grid Infrastructure or Oracle Database home. Starting with Oracle Database Appliance release 19.11, the patches for Oracle Grid Infrastructure and Oracle Database are available as images. All procedures remain the same.

About MySQL Server and DCS Agent

Starting with Oracle Database Appliance release 19.10, DCS Agent uses MySQL server as its metadata store. MySQL server is automatically installed and configured, during provisioning, when creating the appliance. Similarly, during patching, the `odacli update-dcscomponents` command automatically installs and configures MySQL server. MySQL server processes do not run with `root` operating system user account. Instead, MySQL server processes run with a new operating system user account called `odamysql`. Note that the `odamysql` operating system user is created only for ownership purposes and the `odamysql` user account owns only the MySQL directories. The `odamysql` operating system user cannot log into the appliance. The `odamysql` operating system user account for MySQL ensures role separation between the user owning Oracle databases on the appliance and the metadata used for DCS operations.

About Oracle Database Appliance Patch Bundles

The Oracle Database Appliance Patch Bundle contains the latest patches for DCS Admin, DCS Components, OAK, Oracle ILOM, firmware, enhancements, and fixes.

The Oracle Database Appliance Release Notes include information about the latest Oracle Database Appliance patch bundle and a list of component versions in the patch bundle. Oracle recommends updating the appliance with the latest Oracle Database Appliance software version to take advantage of new features, fixes, and the latest supported component versions. See the Oracle Database Appliance Release Notes for the upgrade paths.

> **Caution:**
>
> Do not patch Oracle Database Appliance using generic patches for Oracle Grid Infrastructure or Oracle Linux. Also do not use individual infrastructure patches, such as firmware patches. You must only use Oracle Database Appliance patches. If you use patches that are not intended for Oracle Database Appliance, or if you use OPatch, or a similar patching tool, then the metadata maintained by Oracle Database Appliance is not updated, and you cannot complete future patch updates. If you apply out-of-cycle Oracle Database Release Update (RU), then ensure that you also follow the recommendations described in the Readme for the RU.

The patch bundle provides all relevant patches for the entire system, including the following:

- Oracle Database Appliance server
- BIOS
- Hardware drivers
- Hardware Management Pack (HMP) and firmware drivers for various components
- Oracle Appliance Manager
- Oracle Linux
- Java Runtime Environment (JRE)
- Oracle Integrated Lights Out Manager (Oracle ILOM)
- Oracle Auto Service Request (Oracle ASR)
- Oracle Intelligent Platform Management Interface (Oracle IPMI)
- Network Card Patches for relevant hardware models

About Updating Oracle Linux RPMs

While not recommended, you can update some Oracle Linux RPMs for Oracle Database Appliance. Do not update or customize Oracle Linux kernel, Oracle Optimal Flexible Architecture, Oracle InfiniBand, or related software. Other software may be installed, updated, or customized. However, the Oracle Database Appliance update may not carry newer version dependencies of customized components. Therefore, you might be required to remove and subsequently reapply site-specific changes to successfully update Oracle Database Appliance in the future.

> **Caution:**
>
> For Oracle Database Appliance, do not update the following:
>
> - Oracle Linux Kernel (`kernel`)
> - Oracle Optimal Flexible Architecture (`ofa`)
> - Oracle RDMA packages (`oracle-rdma-release`)
>
> For storage, do not apply any RPM updates.

About Patching Prechecks

Patching prechecks are designed to detect and flag problems that might be encountered during patching.

The prechecks view the state of the appliance, including the version and hardware specific upgrades, verify that the necessary conditions and checks are complete before attempting an upgrade, and identify conditions that might cause an upgrade to fail. You can avoid potential patching problems by correcting the conditions and resolving the issues before applying the patch.

The prechecks commands are available on Oracle Database Appliance release 12.2.1.2.0 or later. Update the DCS Agent to ensure that the latest updated prechecks are available when you run these commands.

Components Verified by Patching Prechecks

Check	Component Verified
System patching prechecks	Checks that the DCS Agent is online on all nodes Checks that there is sufficient space for the update Validates the minimum agent version and validates the patching tag

Check	Component Verified
Server patching prechecks	For operating system: • Validates supported versions • Validates minimum supported versions • Validates patching tag • Validates whether patch location is available • Verifies OS patch • Validates command execution For Oracle ILOM: • Validates supported versions • Validates minimum supported versions • Validates patching tag • Validates whether patch location is available • Checks for Oracle ILOM patch version • Validates Patch location • Validates command execution • Validates connectivity to Oracle ILOM For Oracle Grid Infrastructure: • Validates supported Oracle Grid Infrastructure versions • Validates available space • Validates whether Oracle Clusterware is running • Validates patching tag • Validates whether system is provisioned • Validates whether Oracle ASM is online • Validates the minimum agent version • Validates the central inventory • Validates patching locks • Validates whether clones location exists • Validates DB start dependencies • Validates DB stop dependencies • Validates command execution For Oracle ORAchk: • Runs the ORAchk tool that performs operating system level prechecks. • Validates command execution For server: • Validates whether server patching can be performed in local mode or not • Validates command execution Confirms that the bare metal system is upgraded, in the case of DB systems. Verifies and confirms that the database start/stop dependencies are correct. Confirms that patching locks are not present. Verifies that the central inventory is accurate.

Check	Component Verified
Database home patching prechecks	Confirms that Oracle Grid Infrastructure is installed Confirms that the DCS Agent is at the required version Confirms that Oracle Clusterware is running on all nodes Confirms that Oracle Grid Infrastructure is upgraded Validates that there is sufficient disk space available Checks if the disk group for database homes on Oracle ACFS is configured Checks whether the clone file is present in the repository Checks whether the `/u01/app/odaorabase` directory exists. If the directory exists, then the check confirms that there is a valid symbolic link.
Database home patching with out-of-place patching prechecks	The check creates a database home and verifies that all databases can be moved from the source home to new home. This check validates that the `--local` option can be used for the database home.
Storage prechecks	Validates storage patch location and tag Validates patching location and tag Validates command execution Validates Oracle ASM disk groups status Validates rolling patch (for high-availability deployment)

About Enabling NUMA on Oracle Database Appliance

Understand how you can enable NUMA on bare metal, KVM, and DB Systems on Oracle Database Appliance.

NUMA (Non-Uniform Memory Access) can help facilitate efficient use of the underlying NUMA hardware. On Oracle Database Appliance, you can enable NUMA for databases on bare metal and DB systems, CPU pools, and for KVM applications.

About Enabling NUMA During Provisioning of Oracle Database Appliance

When you provision Oracle Database Appliance with release 19.25, NUMA is automatically enabled on the database, DB system and CPU pools. Hence, when you create a database, DB system, or CPU pool on an Oracle Database Appliance release 19.25 deployment, NUMA is automatically enabled and no additional configuration is required. NUMA is not enabled by default for application VMs.

About Enabling NUMA After Patching Oracle Database Appliance

NUMA is enabled by default starting with Oracle Database Appliance release 19.12. When you patch to Oracle Database Appliance release 19.25 from 19.11 or earlier, the `odacli update-server` command sets NUMA at the system level. New databases created after patching are NUMA-enabled. For existing databases, enable NUMA by running the command `alter system set "enable_numa_support"=true` on the database.

After patching the bare metal system to release 19.25 from 19.11 or earlier, as a one-time activity, you must run the command `odacli remap-cpupools` to enable NUMA on all existing CPU pools (BM, VM, dbsystem).

For existing DB systems, run the command `odacli modify-dbsystem --enable-numa` to enable NUMA for the dbsystem and the database of the DB system.

NUMA is not enabled by default for application VMs. For new and existing application VMs, run the command `odacli modify-vm --enable-numa` to enable NUMA for the VM configuration. You must also enable NUMA for the components inside the VM such as operating system and application running inside the VM. The command `odacli modify-dbsystem --enable-numa` is only supported on application VMs with even number of CPU cores so that CPU cores can be evenly distributed to NUMA nodes. In Oracle Database Appliance, one CPUcore is equal to 2 vCPU.

See the topic *Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line* for more details about the steps to patch your deployment and enable NUMA.

Related Topics

- Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line
  Follow these steps to apply patches to your Oracle Database Appliance bare metal deployment and existing Oracle Database homes, using CLI commands.
- Patching Oracle Database Appliance DB Systems Using the Command-Line
  Follow these steps to apply patches to your Oracle Database Appliance DB system deployment and existing Oracle Database home in the DB system, using CLI commands.
- Remapping CPU Pools in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to remap CPU pools in a bare metal or KVM deployment.
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Minimizing Downtime When Patching Oracle Database Appliance

Understand best practices to minimize downtime when you apply patches to your Oracle Database Appliance deployment.

Recommendations for Minimizing Downtime During Patching

- To minimize database patching downtime, it is recommended that you use Oracle RAC or Oracle RAC One Node on an Oracle Database Appliance high-availability system.

- If available, use Oracle Data Guard and patch the standby system first. After you patch the standby system, switch the workload to the standby system, patch the primary system, and then switch the workload back to the primary system. For more information, see *Patching Oracle Data Guard Databases on Oracle Database Appliance* in this guide.
- Always use Oracle Database Appliance Backup and Recovery (ODABR) to back up the system disks to ensure easy restore, if the patching operation fails. ODABR restores the system disk to pre-patching state.
- The `odacli update-storage` command has a `--rolling` option for storage devices (SSD and HDD) to minimize patching downtime. There could be other shared storage component firmware that require a system restart and do not support rolling. Oracle Database Appliance storage patching can be deferred, but it is not recommended. Note that the `--rolling` option cannot be used with the `odacli update-storage` command on single-node Oracle Database Appliance systems.

Minimizing Downtime using the `--local` Option During Patching

To minimize downtime, use the `--local` option with the `odacli update-server` and `odacli update-dbhome` commands. Update the DCS components and then run the `odacli create-prepatchreport` command. There is no downtime during the prepatch process. After analyzing the prepatch report, use the `--local` option to continue patching Oracle Database Appliance.

> **Note:**
>
> Before patching databases, ensure that the server patching operation on both nodes has completed successfully. Do not run the `odacli update-dbhome` and `odacli update-database` commands before the `odacli update-server` operation completes successfully on both nodes.

> **See Also:**
>
> *Oracle Database High Availability Overview and Best Practices Guide* in the Oracle Database 19c Documentation Library

About Updating Databases and Database Homes

Understand how to update a database and database home using either the Browser User Interface or `odacli` commands.

About Updating Databases and Database Homes

Oracle Database Appliance provides the following options to patch or update databases and database homes:

- **Patch or update each database home.** The database home is patched out-of-place. When patching database home, Oracle Database Appliance automatically provisions a destination database home if needed, records the source database home, target version, and destination database home mapping in the metadata repository, and then moves all databases running in the source database home to the destination database home. Use the `odacli update-dbhome` command or the BUI to patch all databases from a single source database home to a single destination database home. Use this option if there are no constraints on your system for space usage or for provisioning a new database home.

- **Patch or update selected databases in a database home to a specific database home.** You must specify the target database and destination database home when patching. You can use the `odacli update-database` command or the BUI to patch a specific database to a specific database home or to control when and how many database homes are provisioned. Using the update-database workflow also provides an option to run the operation again, if the patching operation failed. The abort and revert options available with the update-database workflow are not supported when the target database is a single-instance database.

Before patching the database or database home, upload the Oracle Database clone files for the database version, to the repository, and then create the database home. See *Patching Oracle Database Appliance Using the CLI* for the procedure to update the repository with the latest Oracle Database clone files.

> **Important:**
>
> The `odacli update-database` command that enabled association of a backup configuration with database was deprecated in Oracle Database Appliance release 19.6. Starting with Oracle Database Appliance release 19.17, the `odacli update-database` command enables database patching and is similar to the patching functionality that the `odacli update-dbhome` command provides, with a few differences as described in the *Oracle Database Appliance Deployment Guide* for your hardware model.

Before you run the `odacli update-dbhome` command, you must create the prepatch report for the database home by running the command `odacli create-prepatchreport -d`. Similarly, before you run the `odacli update-database` command, you must create the prepatch report either for the source database home or target database by running the command `odacli create-prepatchreport -d` or `odacli create-prepatchreport -db` respectively. The source and destination database homes must be consistent in the prepatch report and when specifying them in the `odacli update-database` command. If you do not create the prepatch report before the patch operation, then the `odacli update-database` and `odacli update-dbhome` commands fail with an error message prompting you to run the patching pre-checks.

For more information about the options for the `update-dbhome` and `odacli update-database` commands, see the chapter *Oracle Database Appliance Command-Line Interface*.

Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line

Follow these steps to apply patches to your Oracle Database Appliance bare metal deployment and existing Oracle Database homes, using CLI commands.

To patch your Oracle Database Appliance deployment and your existing database homes to the current release, you must download the Oracle Database Appliance Server patch, Oracle Grid Infrastructure clone files, and Oracle Database clone files and update the repository on the bare metal system.

Important:

Oracle recommends always patching from within the previous four Oracle Database Appliance releases because these patching paths are tested. If you patch from any earlier Oracle Database Appliance release, then you may encounter errors.

Note:

Run the steps in this procedure in the same order as documented. Run the `odacli update-dcsadmin`, `odacli update-dcscomponents`, and `odacli update-dcsagent` commands before running the `odacli update-server` command.

Note:

Note that for DCS Agent update to be complete, both the `odacli update-dcscomponents` and `odacli update-dcsagent` commands must be run. Ensure that both commands are run in the order specified.

Important:

Ensure that there is sufficient space on your appliance to download the patches.

Important:

If you want to install third-party software on your Oracle Database Appliance, then ensure that the software does not impact the Oracle Database Appliance software. The version lock on Oracle Database Appliance RPMs displays a warning if the third-party software tries to override Oracle Database Appliance RPMs. You must restore the affected RPMs before patching Oracle Database Appliance so that patching completes successfully.

Important:

If you have configured Oracle ASR and are patching your appliance to Oracle Database Appliance release 19.25, then after updating the DCS agent, delete the existing Oracle ASR configuration and reconfigure it, before you run the `odacli update-server` command to patch your appliance.

Note:

Public network gateway must be responsive to ping requests. Oracle Clusterware uses the public network gateway as ping target to facilitate the VIP failover in Oracle Database Appliance high-availability environment.

> **See Also:**
>
> *Enabling Custom SSH Key Management on Oracle Database Appliance* in this guide for information on how to manage SSH keys on bare metal systems and DB systems

The Oracle Database Appliance Backup Recovery (ODABR) tool helps recover a system to the pre-patch state, should any potential errors occur during the patching process that may render the system unusable. Download and install the latest version of the Oracle Database Appliance Backup Recovery (ODABR) tool to enable creation of snapshots before patching. See My Oracle Support Note 2466177.1 for the procedure to install ODABR.
https://support.oracle.com/rs?type=doc&id=2466177.1

Follow these steps to apply patches to your Oracle Database Appliance and update existing database homes.

1. Download the Oracle Database Appliance Server Patch for the ODACLI/DCS stack (patch 37081701), Oracle Grid Infrastructure clone files (patch 30403673), and Oracle Database clone files (patch 30403662) from My Oracle Support to a temporary location on an external client. Refer to the release notes for details about the patch numbers and software for the latest release.

   For example, download the server patch for 19.25:

   ```
   p37081701_1925000_Linux-x86-64.zip
   ```

2. Unzip the software — it contains README.html and one or more zip files for the patch.

   ```
   unzip p37081701_1925000_Linux-x86-64.zip
   ```

   The zip file contains the following software file:

   ```
   oda-sm-19.25.0.0.0-date-server.zip
   ```

3. Copy all the software files from the external client to Oracle Database Appliance. For High-Availability deployments, copy the software files to only one node. The software files are copied to the other node during the patching process. Use the `scp` or `sftp` protocol to copy the bundle.

   Example using `scp` command:

   ```
   # scp software_file root@oda_host:/tmp
   ```

   Example using `sftp` command:

   ```
   # sftp root@oda_host
   ```

   Enter the `root` password, and copy the files.

   ```
   put software_file
   ```

4. Update the repository with the server software file:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
software_file
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/oda-
sm-19.25.0.0.0-date-server.zip
```

5. Confirm that the repository update is successful:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-job -i job_ID
```

6. Update DCS admin:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsadmin -v 19.25.0.0.0
```

7. Update the DCS components:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
19.25.0.0.0
```

The `odacli update-dcscomponents` command updates the Zookeeper, MySQL, and updates other DCS components such as the DCS Agent, DCS CLI, and DCS Controller on Oracle Database Appliance.

If the DCS components are updated, then the message `"status" : "Success"` is displayed on the command line. For failed updates, fix the error and then proceed with the update by re-running the `odacli update-dcscomponents` command. See the topic *Resolving Errors When Updating DCS Components During Patching* about more information about DCS components checks errors.

> **Note:**
>
> Note that for DCS Agent update to be complete, both the `odacli update-dcscomponents` and `odacli update-dcsagent` commands must be run. Ensure that both commands are run in the order specified in this procedure.

8. Update the DCS agent:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsagent -v 19.25.0.0.0
```

9. Update the repository with the Oracle Grid Infrastructure clone file and the Oracle Database clone file:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
software_file
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
odacli-dcs-19.25.0.0.0-date-GI-19.25.0.0.zip
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip
```

10. Run patching pre-checks. You must run the `odacli create-prepatchreport` command before you patch your server; otherwise, the `odacli update-server` command fails with an error message prompting you to run the patching pre-checks. Note that `odacli create-prepatchreport` supports the `-l` option, which runs the prechecks on the local node only. However, to be able to patch the system by running the command `odacli update-server`, the command `odacli create-prepatchreport` must be run without the `-l` option, so that the prechecks can be performed on both nodes before patching is started on any of the two nodes.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -s -v
version
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -s -v
19.25.0.0.0
```

Note that running the `odacli create-prepatchreport` command creates the target Oracle Grid Infrastructure home, which the server patching operation uses.

11. Verify that the patching pre-checks ran successfully:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-prepatchreport
```

For example:

```
[root@oda1 opt]# odacli describe-prepatchreport -i 677d73d2-5eed-46e5-a06f-
d5ca2649d36e

Patch pre-check report
------------------------------------------------------------------------
                 Job ID: 677d73d2-5eed-46e5-a06f-d5ca2649d36e
            Description: Patch pre-checks for [OS, ILOM, GI, ORACHKSERVER,
SERVER]
                 Status: SUCCESS
                Created: September 8, 2024 3:37:19 PM CST
                 Result: All pre-checks succeeded
Node Name
---------------
n1

Pre-Check Status Comments
------------------------------ --------
-------------------------------------
__OS__
Validate supported versions Success Validated minimum supported versions.
Validate patching tag Success Validated patching tag: 19.25.0.0.0
```

```
Is patch location available Success Patch location is available.
Verify OS patch Success Verified OS patch
Validate command execution Success Validated command execution

__ILOM__
Validate ILOM server reachable Success Successfully connected with ILOM
                                       server using public IP and USB
                                       interconnect
Validate supported versions Success Validated minimum supported versions.
Validate patching tag Success Validated patching tag: 19.25.0.0.0.
Is patch location available Success Patch location is available.
Checking Ilom patch Version Success Successfully verified the versions
Patch location validation Success Successfully validated location
Validate command execution Success Validated command execution
 __GI__
Validate GI metadata Success Successfully validated GI metadata
Validate supported GI versions Success Validated minimum supported
versions.
Validate available space Success Validated free space under /u01
Is clusterware running Success Clusterware is running
Validate patching tag Success Validated patching tag: 19.25.0.0.0.
Is system provisioned Success Verified system is provisioned
Validate ASM in online Success ASM is online
Validate kernel log level Success Successfully validated the OS log
                                       level
Validate minimum agent version Success GI patching enabled in current
                                       DCSAGENT version
Validate Central Inventory Success oraInventory validation passed
Validate patching locks Success Validated patching locks
Validate clones location exist Success Validated clones location
Validate DB start dependencies Success DBs START dependency check passed
Validate DB stop dependencies Success DBs STOP dependency check passed
Validate space for clones Success Clones volume is already created
volume
Evaluate GI patching Success Successfully validated GI patching
Validate command execution Success Validated command execution
 __ORACHK__
Running orachk Success Successfully ran Orachk
Validate command execution Success Validated command execution

__SERVER__
Validate local patching Success Successfully validated server local
                                       patching
Validate command execution Success Validated command execution
Node Name
---------------
n2

Pre-Check Status Comments
------------------------------ --------
-------------------------------------
__OS__
Validate supported versions Success Validated minimum supported versions.
Validate patching tag Success Validated patching tag: 19.25.0.0.0.
Is patch location available Success Patch location is available.
Verify OS patch Success Verified OS patch
```

```
Validate command execution Success Validated command execution

__ILOM__
Validate ILOM server reachable Success Successfully connected with ILOM
                                       server using public IP and USB
                                       interconnect
Validate supported versions Success Validated minimum supported versions.
Validate patching tag Success Validated patching tag: 19.25.0.0.0.
Is patch location available Success Patch location is available.
Checking Ilom patch Version Success Successfully verified the versions
Patch location validation Success Successfully validated location
Validate command execution Success Validated command execution
__GI__
Validate GI metadata Success Successfully validated GI metadata
Validate supported GI versions Success Validated minimum supported
versions.
Validate available space Success Validated free space under /u01
Is clusterware running Success Clusterware is running
Validate patching tag Success Validated patching tag: 19.25.0.0.0.
Is system provisioned Success Verified system is provisioned
Validate ASM in online Success ASM is online
Validate kernel log level Success Successfully validated the OS log
                                       level
Validate minimum agent version Success GI patching enabled in current
                                       DCSAGENT version
Validate Central Inventory Success oraInventory validation passed
Validate patching locks Success Validated patching locks
Validate clones location exist Success Validated clones location
Validate DB start dependencies Success DBs START dependency check passed
Validate DB stop dependencies Success DBs STOP dependency check passed
Validate space for clones Success Clones volume is already created
volume
Evaluate GI patching Success Successfully validated GI patching
Validate command execution Success Validated command execution
__ORACHK__
Running orachk Success Successfully ran Orachk
Validate command execution Success Validated command execution

__SERVER__
Validate local patching Success Successfully validated server local
                                       patching
Validate command execution Success Validated command execution
```

Use the command `odacli describe-prepatchreport` to view details of the pre-patch report. Fix the warnings and errors mentioned in the report and proceed with the server patching.

12. Apply the server update. Note that you can use the `--local` option with the `odacli update-server` command to update the local node.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-server -v version
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-server -v 19.25.0.0.0
```

13. Confirm that the server update is successful:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-job -i job_ID
```

> **Note:**
>
> After successful server update, it is recommended that you delete any ODABR snapshot you created before starting the server update. If you want to retain any ODABR snapshot when you update the storage and other components, then consider that the presence of active ODABR snapshots may affect system performance. See My Oracle Support Note 2466177.1 for the procedure to create and delete ODABR snapshots: https://support.oracle.com/rs?type=doc&id=2466177.1

14. Before you update the storage components, run the `odacli create-prepatchreport` command with the `-st` option.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -st -v
version
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -st -v
19.25.0.0.0
```

15. Verify that the patching pre-checks ran successfully:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-prepatchreport
```

For example:

```
[root@oda1 opt]# odacli describe-prepatchreport -i 95887f92-7be7-4865-
a311-54318ab385f2

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  95887f92-7be7-4865-a311-54318ab385f2
            Description:  Patch pre-checks for [STORAGE]
                 Status:  SUCCESS
                Created:  September 8, 2024 12:52:37 PM HKT
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status
Comments
------------------------------ --------
--------------------------------------
__STORAGE__
Validate patching tag           Success   Validated patching tag:
```

```
19.25.0.0.0.
Patch location validation       Success   Verified patch
location
Patch tag validation            Success   Verified patch
tag
Storage patch tag validation    Success   Verified storage patch
location
Verify ASM disks status         Success   ASM disks are
online
Validate rolling patch          Success   Rolling mode patching allowed
as
                                          there is no expander and
controller

upgrade.
Validate command execution      Success   Validated command
execution

Node Name
---------------
node2

Pre-Check                      Status
Comments
------------------------------ --------
-------------------------------------
__STORAGE__
Validate patching tag           Success   Validated patching tag:
19.25.0.0.0.
Patch location validation       Success   Verified patch
location
Patch tag validation            Success   Verified patch
tag
Storage patch tag validation    Success   Verified storage patch
location
Verify ASM disks status         Success   ASM disks are
online
Validate rolling patch          Success   Rolling mode patching allowed
as
                                          there is no expander and
controller

upgrade.
Validate command execution      Success   Validated command
execution
```

Use the command `odacli describe-prepatchreport` to view details of the pre-patch report. The pre-patch report also indicates whether storage patching can be rolling or not, based on whether an Expander or Controller update is also required.

Fix the warnings and errors mentioned in the report and proceed with the storage components patching.

16. Update the storage components.

Specify the `--rolling` option to patch shared disks in a rolling fashion. Note that if you patch from an Oracle Database Appliance release that requires the expander to be patched, then you cannot use the `--rolling` option during storage patching.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-storage -v version --
rolling
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-storage -v 19.25.0.0.0
--rolling
```

17. Starting with Oracle Database Appliance release 19.11, Oracle Database homes are not created on the local disk. Oracle Database homes are created on an Oracle ACFS-managed file system. For a database user oracle, the new database homes are created under `/u01/app/odaorahome/oracle/`.

    If the database home is not already configured on Oracle ACFS, then before patching the database home, configure the database home storage with the `odacli configure-dbhome-storage` command. For example:

    ```
    [root@oda1 opt]# odacli configure-dbhome-storage -dg DATA
    ```

    The command does not cause storage allocation or creation of volumes or file systems. The command only sets the disk group location in the metadata.

    For information about managing database homes on Oracle ACFS, see the topic *Managing Database Home Storage*.

18. To patch existing Oracle Database Homes to release 19.25, ensure that the repository is updated with the Oracle Database clone files. If you have not updated the repository with the Oracle Database clone files, then update the repository. Note that you must update the DCS agent before you update the repository with the Oracle Database clone files.

    ```
    [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
    DB_software_file
    ```

    ```
    [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
    odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip
    ```

    > **Note:**
    >
    > You have the option to patch each database home using the `odacli update-dbhome` command or patch selected databases in a database home to a specific database home using the `odacli update-database` command.

19. **To patch all databases in a specified database home:**

    a. If you want to patch each database home by running the `odacli update-dbhome` command, then create the prepatch report first. Run the patching pre-checks and review the prepatch report. You must run the `odacli create-prepatchreport` command before you patch your Oracle Database homes; otherwise, the `odacli`

`update-dbhome` command fails with an error message prompting you to run the patching pre-checks.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport --
dbhome --dbhomeid DB_Home_ID -v release_number
[root@oda1 opt]# odacli describe-prepatchreport -i Job_ID


[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport --
dbhome --dbhomeid d03d5187-1c20-470a-9d51-49a0d6f60772 -v 19.25.0.0.0
[root@oda1 opt]# odacli describe-prepatchreport -i 6f040374-d271-4711-
a7db-e6c04597ad69

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  6f040374-d271-4711-a7db-e6c04597ad69
            Description:  Patch pre-checks for [DB, ORACHKDB]: DbHome
is OraDB19000_home1
                 Status:  SUCCESS
                Created:  September 8, 2024 8:59:48 AM HKT
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status
Comments
------------------------------ --------
--------------------------------------
__DB__
Validate DB Home ID            Success   Validated DB Home
ID:

d03d5187-1c20-470a-9d51-49a0d6f60772
Validate patching tag          Success   Validated patching tag:
19.25.0.0.0.
Is system provisioned          Success   Verified system is
provisioned
Validate minimum agent version Success   Validated minimum agent
version
Is GI upgraded                 Success   Validated GI is
upgraded
Validate available space for   Success   Validated free space required
under
db                                       /u01/app/
odaorahome
Validate dbHomesOnACFS         Success   User has configured diskgroup
for
configured                               Database homes on
ACFS
Validate Oracle base           Success   Successfully validated Oracle
Base
Is DB clone available          Success   Successfully validated clone
file
```

```
exists
Evaluate DBHome patching with   Success   Successfully validated
updating
RHP                                       dbhome with RHP.  and local
patching
                                          is
possible
Validate command execution      Success   Validated command
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution

Node Name
---------------
node2

Pre-Check                      Status
Comments
------------------------------ --------
--------------------------------------
__DB__
Validate DB Home ID             Success   Validated DB Home
ID:

d03d5187-1c20-470a-9d51-49a0d6f60772
Validate patching tag           Success   Validated patching tag:
19.25.0.0.0.
Is system provisioned           Success   Verified system is
provisioned
Validate minimum agent version  Success   Validated minimum agent
version
Is GI upgraded                  Success   Validated GI is
upgraded
Validate available space for    Success   Validated free space required
under
db                                        /u01/app/
odaorahome
Validate dbHomesOnACFS          Success   User has configured diskgroup
for
configured                                Database homes on
ACFS
Validate Oracle base            Success   Successfully validated Oracle
Base
Is DB clone available           Success   Successfully validated clone
file

exists
Evaluate DBHome patching with   Success   Successfully validated
updating
RHP                                       dbhome with
RHP.
Validate command execution      Success   Validated command
```

```
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution
```

If the disk group for database homes on Oracle ACFS is not set up as described in step 17, then the prepatch report lists the error `The system is not set up to create database homes on ACFS`. Run the `odacli configure-dbhome-storage` command to configure the disk group for database homes on Oracle ACFS.

Note that running the `odacli create-prepatchreport` command creates the target Oracle Grid Infrastructure home, which the database patching operation uses.

b. If the pre-patch report does not show any errors, then update the database home to the latest release. You need not use the `-f` option with the `odacli update-dbhome` command. Note that you can use the `--local` option with the `odacli update-dbhome` command to update the local node.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --id
DB_Home_ID -v release_number
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --id 7c67c5b4-
f585-4ba9-865f-c719c63c0a6e -v 19.25.0.0.0
```

20. **To patch a specific database to a specific database home:**

a. If you want patch selected databases in a database home to a specific database home using the `odacli update-database` command, then create the prepatch report first. Run the patching pre-checks and review the prepatch report. You must run the `odacli create-prepatchreport` command before you patch your Oracle database; otherwise, the `odacli update-database` command fails with an error message prompting you to run the patching pre-checks.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -db -
dbid DATABASE_ID -to a68f9947-DB_HOME_ID
```

For example, for 19.25:

```
[root@oda1 ~]# odacli describe-prepatchreport -i efa93519-
a365-4e9a-9f25-ad3c653cccef

Patch pre-check report
------------------------------------------------------------------------
                 Job ID: efa93519-a365-4e9a-9f25-ad3c653cccef
            Description: Patch pre-checks for [SINGLEDB,
ORACHKSINGLEDB]: Target DB is db10181, Destination DbHome is
OraDB19000_home8
                 Status: SUCCESS
                Created: September 8, 2024 11:19:03 AM PDT
                 Result: All pre-checks succeeded
```

```
Node Name
---------------
node1

Pre-Check Status Comments
------------------------------ --------
-------------------------------------
__SINGLEDB__
Is system provisioned Success Verified system is provisioned
Validate dbHomesOnACFS Success User has configured disk group for
configured Database homes on ACFS
Validate Oracle base Success Successfully validated Oracle Base
Evaluate DB clone availability Success Successfully validated clone file
                                          exists
Evaluate DB patching with RHP Success Successfully validated patching DB
                                          with RHP.
Validate command execution Success Validated command execution

__ORACHK__
Running orachk Success Successfully ran Orachk
Validate command execution Success Validated command execution

Node Name
---------------
node2

Pre-Check Status Comments
------------------------------ --------
-------------------------------------
__SINGLEDB__
Is system provisioned Success Verified system is provisioned
Validate dbHomesOnACFS Success User has configured disk group for
configured Database homes on ACFS
Validate Oracle base Success Successfully validated Oracle Base
Evaluate DB clone availability Success Successfully validated clone file
                                          exists
Evaluate DB patching with RHP Success Successfully validated patching DB
                                          with RHP.
Validate command execution Success Validated command execution

__ORACHK__
Running orachk Success Successfully ran Orachk
Validate command execution Success Validated command execution
```

b. If the pre-patch report does not show any errors, then update the databases to the latest release.

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-database -i
database_ID/-n database_name -to destination_dbhome_ID
```

For example, for 19.25:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-database -i
c05458c7-0f1d-4260-8e62-003b3ec26d74/-n db10181 -to dc98eaa9-289c-4710-
```

```
a898-ca6287a07a9c
[root@oda1 ~]# odacli describe-job -i
d442a7d4-594f-48e2-8b02-9124811bb7b4

Job details
----------------------------------------------------------------
                     ID: d442a7d4-594f-48e2-8b02-9124811bb7b4
            Description: DB Patching: database ID is
c05458c7-0f1d-4260-8e62-003b3ec26d74
                 Status: Success
                Created: September 8, 2024 3:23:43 PM PDT
                Message:

Task Name Node Name Start Time End Time Status
---------------------------------------- -------------------------
---------------------------------- ----------------------------------
----------
Creating wallet for DB Client node1 September 8, 2024 3:24:42 PM PDT
September 8, 2024 3:24:42 PM PDT Success
Patch databases by RHP node1 September 8, 2024 3:24:42 PM PDT September
8, 2024 3:29:34 PM PDT Success
Updating database metadata node2 September 8, 2024 3:29:34 PM PDT
September 8, 2024 3:29:34 PM PDT Success
Set log_archive_dest for Database node1 September 8, 2024 3:29:34 PM
PDT September 8, 2024 3:29:37 PM PDT Success
Generating and saving BOM node1 September 8, 2024 3:29:37 PM PDT
September 8, 2024 3:31:44 PM PDT Success
Generating and saving BOM node2 September 8, 2024 3:29:37 PM PDT
September 8, 2024 3:31:44 PM PDT Success
TDE parameter update node2 September 8, 2024 3:32:25 PM PDT September
8, 2024 3:32:25 PM PDT Success
```

21. Confirm that the patch jobs are successful:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli list-jobs
```

22. If you are patching your appliance from Oracle Database Appliance release 19.11 or earlier, then run the following steps. You must also run these steps if your appliance was previously patched to Oracle Database Appliance release 19.12 or later, but you did not run these steps when patching to release 19.12.

    a. After patching the bare metal systems and database homes, enable NUMA (Non-Uniform Memory Access) on existing databases of release 19.12 or earlier, by running the following SQL*Plus command:

    ```
    SQL> alter system set "_enable_numa_support"=true scope=spfile sid='*';
    ```

    Restart the database after running the command to ensure that the changes take effect.

    For more information about enabling NUMA, see the topic *About Enabling NUMA on Oracle Database Appliance*.

    b. If you have deployed DB systems, VMs, or CPU pools, perform the following steps:

i. Run the `odacli remap-cpupools` command:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli remap-cpupools
```

If the `odacli remap-cpupools` command fails with the error `DCS-10001:Internal error encountered: Not enough physical CPUs available for CPU Pool 'cpupool_name' on node 'node_name'`, run the `odacli remap-cpupools` command again with the `--use_reserved-cores` option. For information about the command options, see the chapter *Oracle Database Appliance Command-Line Interface for KVM*.

ii. If you have DB systems in your deployment, then run the `odacli modify-dbsystem` command. If you have VMs in your KVM deployment and you want to enable NUMA for the VMs, then run the `odacli modify-vm --enable-numa` command for VMs. That is:
For DB systems:

```
odacli modify-dbsystem --enable-numa
```

For VMs:

```
odacli modify-vm --enable-numa
```

> **Note:**
>
> For bare metal systems and DB systems on Oracle Database Appliance X7-2S and X8-2S, the database `init.ora` parameter `_enable_numa_support` is set to `FALSE` even when NUMA is enabled. This is expected because these hardware models are single socket systems.

Related Topics

- Patching Existing Database Homes Using ODACLI or the BUI
  Use ODACLI or BUI to patch database homes in your deployment to the latest release.

Related Topics

- Oracle Database Appliance Release Notes

Related Topics

- Resolving Errors When Updating DCS Components During Patching
  Understand how to troubleshoot errors when updating DCS components during patching.

Related Topics

- About Enabling NUMA on Oracle Database Appliance
  Understand how you can enable NUMA on bare metal, KVM, and DB Systems on Oracle Database Appliance.
- Patching Oracle Database Appliance DB Systems Using the Command-Line
  Follow these steps to apply patches to your Oracle Database Appliance DB system deployment and existing Oracle Database home in the DB system, using CLI commands.
- Managing Database Homes on Oracle ACFS Storage
  Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to manage database home storage in Oracle Database Appliance.

- **About Creating Database Homes on Oracle ACFS Storage**
  When you create new Oracle Database homes, they are created on Oracle ACFS file system which are stored on Oracle Database Appliance storage disks. You can create new Oracle Database homes using the `odacli create-database` or `odacli create-dbhome` commands, or when you patch an existing database home.
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Patching Oracle Database Appliance DB Systems Using the Command-Line

Follow these steps to apply patches to your Oracle Database Appliance DB system deployment and existing Oracle Database home in the DB system, using CLI commands.

To patch your Oracle Database Appliance DB systems deployment and your existing database homes to the current release, you must download the Oracle Database Appliance Server Patch and update the bare metal system repository.

> **Note:**
>
> Run the commands in this topic in the same order as documented. Run the `odacli update-dcsadmin`, `odacli update-dcscomponents`, and `odacli update-dcsagent` commands before running the `odacli update-server` command.

> **Note:**
>
> Note that for DCS agent update to be complete, both the `odacli update-dcscomponents` and `odacli update-dcsagent` commands must be run. Ensure that both commands are run in the order specified.

> **Note:**
>
> Public network gateway must be responsive to ping requests. Oracle Clusterware uses the public network gateway as ping target to facilitate the VIP failover in Oracle Database Appliance high-availability environment.

Ensure that you have patched your Oracle Database Appliance bare metal deployment to release 19.25 before you patch your DB system.

Follow these steps to apply patches to your Oracle Database Appliance DB system deployment and update the existing database home in the DB system.

1. Ensure that the Oracle Database Appliance Server Patch for the ODACLI/DCS stack, Oracle Grid Infrastructure clone files, and Oracle Database clone files are available in the repository. Run the `odacli list-availablepatches` command in the bare metal system to view whether the clone files are available in the system.

   ```
   [root@oda1 opt]# odacli list-availablepatches
   ```

2. Connect to the DB system.
3. Update DCS admin:

   ```
   [root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dcsadmin -v
   19.25.0.0.0
   ```

4. Update the DCS components:

   ```
   [root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
   19.25.0.0.0
   ```

   If the DCS components are updated, then the message `"status" : "Success"` is displayed on the command line. For failed updates, fix the error and then proceed with the update by re-running the `odacli update-dcscomponents` command. See the topic *Resolving Errors When Updating DCS Components During Patching* about more information about DCS components checks errors.

   > **Note:**
   >
   > Note that for DCS agent update to be complete, both the `odacli update-dcscomponents` and `odacli update-dcsagent` commands must be run. Ensure that both commands are run in the order specified.

5. Update the DCS agent:

   ```
   [root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dcsagent -v
   19.25.0.0.0
   ```

6. Run patching pre-checks. You must run the `odacli create-prepatchreport` command before you patch your server; otherwise, the `odacli update-server` command fails with an error message prompting you to run the patching pre-checks.

   ```
   [root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -s -
   v version
   ```

   For example, for 19.25:

   ```
   [root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -s -
   v 19.25.0.0.0
   ```

7. Verify that the patching pre-checks ran successfully:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli describe-prepatchreport
```

For example:

```
[root@dbsystem1 ~]# odacli describe-prepatchreport -i
f7e6fb22-1fc0-4aa1-937c-9669dbbb8ae6

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  f7e6fb22-1fc0-4aa1-937c-9669dbbb8ae6
            Description:  Patch pre-checks for [OS, GI, ORACHKSERVER,
SERVER]
                 Status:  SUCCESS
                Created:  September 8, 2024 3:38:03 PM CET
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status
Comments
------------------------------ --------
--------------------------------------
__OS__
Validate supported versions     Success   Validated minimum supported
versions.
Validate patching tag           Success   Validated patching tag:
19.25.0.0.0.
Is patch location available     Success   Patch location is
available.
Verify OS patch                 Success   Verified OS
patch
Validate command execution      Success   Validated command
execution

__GI__
Validate GI metadata            Success   Successfully validated GI
metadata
Validate supported GI versions  Success   Validated minimum supported
versions.
Validate available space        Success   Validated free space
under /u01
Is clusterware running          Success   Clusterware is
running
Validate patching tag           Success   Validated patching tag:
19.25.0.0.0.
Is system provisioned           Success   Verified system is
provisioned
Validate BM versions            Success   Validated BM server
components

versions
```

```
Validate kernel log level       Success   Successfully validated the OS
log

level
Validate minimum agent version  Success   GI patching enabled in
current
                                          DCSAGENT
version
Validate Central Inventory      Success   oraInventory validation
passed
Validate patching locks         Success   Validated patching
locks
Validate clones location exist  Success   Validated clones
location
Evaluate GI patching            Success   Successfully validated GI
patching
Validate command execution      Success   Validated command
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution

__SERVER__
Validate local patching         Success   Successfully validated server
local

patching
Validate command execution      Success   Validated command
execution

Node Name
---------------
node2

Pre-Check                      Status
Comments
------------------------------ --------
--------------------------------------
__OS__
Validate supported versions     Success   Validated minimum supported
versions.
Validate patching tag           Success   Validated patching tag:
19.25.0.0.0.
Is patch location available     Success   Patch location is
available.
Verify OS patch                 Success   Verified OS
patch
Validate command execution      Success   Validated command
execution

__GI__
Validate GI metadata            Success   Successfully validated GI
metadata
```

```
Validate supported GI versions  Success   Validated minimum supported
versions.
Validate available space        Success   Validated free space
under /u01
Is clusterware running          Success   Clusterware is
running
Validate patching tag           Success   Validated patching tag:
19.25.0.0.0.
Is system provisioned           Success   Verified system is
provisioned
Validate BM versions            Success   Validated BM server
components

versions
Validate kernel log level       Success   Successfully validated the OS
log

level
Validate minimum agent version  Success   GI patching enabled in
current
                                          DCSAGENT
version
Validate Central Inventory      Success   oraInventory validation
passed
Validate patching locks         Success   Validated patching
locks
Validate clones location exist  Success   Validated clones
location
Evaluate GI patching            Success   Successfully validated GI
patching
Validate command execution      Success   Validated command
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution

__SERVER__
Validate local patching         Success   Successfully validated server
local

patching
Validate command execution      Success   Validated command
execution
```

Use the command `odacli describe-prepatchreport` to view details of the pre-patch report. Fix the warnings and errors mentioned in the report and proceed with the server patching.

8. Apply the server update. Note that you can use the `--local` option with the `odacli update-server` command to update the local node.

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-server -v version
```

For example, for 19.25:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-server -v
19.25.0.0.0
```

You can use the `-f` option to override precheck failures and complete the server patching.

9. Confirm that the server update is successful:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli describe-job -i job_ID
```

10. **To patch existing Oracle Database Homes to release 19.25:**

    a. On the bare metal system, run the command `odacli describe-dbsystem-image` to check that the repository is updated with the required Oracle Database clone files. If the database clone file is not displayed in the command output, then run the command `odacli update-repository` on the bare metal system to update the repository with the database clone files:

    ```
    [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
    DB_software_file
    ```

    ```
    [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
    odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip
    ```

    b. Run the patching pre-checks and review the prepatch report. You must run the `odacli create-prepatchreport` command before you patch your Oracle Database homes; otherwise, the `odacli update-dbhome` command fails with the error message prompting you to run the patching pre-checks.

    ```
    [root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport
    --dbhome --dbhomeid DB_Home_ID -v release_number
    [root@dbsystem1 opt]# odacli describe-prepatchreport -i Job_ID
    ```

    ```
    [root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport
    --dbhome --dbhomeid 7c67c5b4-f585-4ba9-865f-c719c63c0a6e -v 19.25.0.0.0
    [root@dbsystem1 opt]# odacli describe-prepatchreport -i
    39ef1eeb-70d3-47ad-b3f5-48960ca0607b
    ```

    Alternatively, you can use the odacli update-database to update the databases in your deployment:

    ```
    [root@dbsystem1 opt]# odacli create-prepatchreport --database --
    database-id database_ID --to-home dest_dbhome_ID
    [root@dbsystem1 opt]# odacli update-database --id database_ID --to-home
    dest_dbhome_ID
    ```

    c. If there are no errors, then update the database home to the latest release. You need not use the `-f` option with the `odacli update-dbhome` command. Note that you can use the `--local` option with the `odacli update-dbhome` command to update the local

node. You can also use the `--node` option to specify the node containing the database to update.

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --id
DB_Home_ID -v release_number
```

For example, for 19.25:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --id
7c67c5b4-f585-4ba9-865f-c719c63c0a6e -v 19.25.0.0.0
```

11. Confirm that the patch jobs are successful:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli list-jobs
```

Related Topics

- About Enabling NUMA on Oracle Database Appliance
  Understand how you can enable NUMA on bare metal, KVM, and DB Systems on Oracle Database Appliance.
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli describe-dbsystem-image
  Use the `odacli describe-dbsystem-image` command to view the supported Oracle Grid Infrastructure and Oracle Database versions and the availability of the clone files and Database System KVM image in the Oracle Database Appliance system.

Patching Oracle Database Appliance Bare Metal Systems Using the Browser User Interface

Upload the Oracle Database Appliance Server Patch to the patch repository, deploy the patch bundle using the Browser User Interface, and then update the DCS admin, DCS components, and DCS agent using CLI commands.

Updating the Patch Repository Using the Browser User Interface

1. Download the Oracle Database Appliance patches from My Oracle Support and save them in a directory on the appliance. See the *Oracle Database Appliance Release Notes* for a list of available patches and links to download the patches.

2. Log into the Browser User Interface.

   ```
   https://Node0-host-ip-address:7093/mgmt/index.html
   ```

3. Click **Repository Manager**.
4. In the Update Patch Repository page, enter the absolute file path and patch name, then click **Update Repository** to start the update repository job. You can list more than one file to update to the repository with a comma separated list, without spaces, or you can update each file, one at a time.
5. Click **Activity** to monitor the progress. When the job completes successfully, the Patch Repository is updated.

Updating the DCS Admin, DCS Components, and DCS Agent After Updating the Repository and Before Patching

Before patching your appliance using the Browser User Interface, you must update the DCS admin, DCS agent, and DCS components manually through CLI commands.

Follow these steps:

1. In the BUI, click **Appliance** and then click **Infrastructure Patching** on the lefthand pane.
2. Using ODACLI commands, update the DCS admin, DCS components, and DCS agent:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsadmin  -v 19.25.0.0.0
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
   19.25.0.0.0
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsagent -v 19.25.0.0.0
   ```

   If the DCS components are updated, then the message `"status" : "Success"` is displayed on the command line. For failed updates, fix the error and then proceed with the update.

   > **Note:**
   >
   > You must log into the Browser User Interface again after running the `odacli update-dcscomponents` command.

Running Patching Prechecks and Patching Server, and Storage Using the Browser User Interface

1. Do not have any jobs running or pending during the update window.
2. In the BUI, click the **Appliance** tab, and then click **Infrastructure Patching** on the lefthand pane.
3. In the SERVER section, view the Component Details. If you just updated the Patch Repository, click **Refresh**.

   After the patch is uploaded to the Patch Repository, the Component Details on the page are updated with the Installed Version and Available Version for each component.
4. Click **Precheck** to run patching prechecks. You must run the patching prechecks before updating the server and other components. Click **View Pre-check Reports** to view the patching prechecks report. If there are no errors in the report, then click **Apply Patch** to begin the job to patch the server components. For high-availability systems, when updating

the server on the bare metal system, you can select the **Node to Update**. You can choose the node that you want to update or you can choose to update both nodes.

When the job finishes, go to the next step. Click **View Jobs** for the job status.

5. In the STORAGE section, click **Refresh** to refresh the Component Details. Click **Precheck** to run patching prechecks. You must run the patching prechecks before updating the storage components. Click **View Pre-check Reports** to view the patching prechecks report. If there are no errors in the report, then click **Apply Patch** to begin the job to patch the storage components. Click **View Jobs** for the job status.

   For high-availability environment, you can select the **Rolling** check box to perform rolling patching of storage components.

All patching-related information is logged in the `dcs-agent` log file at `/opt/oracle/dcs/log/dcs-agent.log`.

Patching Oracle Database Appliance DB Systems Using the Browser User Interface

Upload the Oracle Database Appliance Server Patch to the patch repository, deploy the patch bundle using the Browser User Interface, and then update the DCS agent and components using CLI commands.

Updating the Patch Repository Using the Browser User Interface

1. Download the Oracle Database Appliance patches from My Oracle Support and save them in a directory on the appliance. See the *Oracle Database Appliance Release Notes* for a list of available patches and links to download the patches.
2. Log into the Browser User Interface within the bare metal system, with the `oda-admin` user name and password.

   `https://Node0-host-ip-address:7093/mgmt/index.html`

3. Click the **Appliance** tab, and then click **Infrastructure Patching** on the lefthand pane.
4. In the **Update Repository** tab, on the Update Patch Repository page, enter the absolute file path and patch name, then click **Update Repository** to start the update repository job. You can list more than one file to update to the repository with a comma separated list, without spaces, or you can update each file, one at a time.
5. Click **Activity** to monitor the progress. When the job completes successfully, the Patch Repository is updated.

Updating the DCS Admin, DCS Components, and DCS Agent After Updating the Repository and Before Patching

Before patching your DB system using the Browser User Interface, you must update the DCS admin, DCS agent, and DCS components manually, through CLI commands.

Follow these steps:

1. In the BUI, click **Appliance** and then click **Infrastructure Patching** on the lefthand pane.
2. Using ODACLI, update the DCS admin, DCS components, and DCS agent:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsadmin  -v 19.25.0.0.0
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
```

```
19.25.0.0.0
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsagent -v 19.25.0.0.0
```

If the DCS components are updated, then the message `"status" : "Success"` is displayed on the command line. For failed updates, fix the error and then proceed with the update.

> **Note:**
>
> You must log into the Browser User Interface again after running the `odacli update-dcscomponents` command.

Running Patching Prechecks and Patching Server and Storage Components Using the Browser User Interface

1. Do not have any jobs running or pending during the update window.
2. Log into the Browser User Interface within the DB system with the `oda-admin` user name and password.

   `https://Node0-host-ip-address:7093/mgmt/index.html`

3. Click the **Appliance** tab, and then click **Infrastructure Patching** on the lefthand pane.
4. In the SERVER section, view the Component Details. If you just updated the Patch Repository, click **Refresh**.

   After the patch is uploaded to the Patch Repository, the Component Details on the page are updated with the Installed Version and Available Version for each component.
5. Click **Precheck** to run patching prechecks. You must run the patching prechecks before updating the server and other components. Click **View Pre-check Reports** to view the patching prechecks report. If there are no errors in the report, then click **Apply Patch** to begin the job to patch the server components. For high-availability systems, when updating the server on the bare metal system, you can select the **Node to Update**. You can choose the node that you want to update or you can choose to update both nodes.

   When the job finishes, go to the next step. Click **View Jobs** for the job status.
6. In the STORAGE section, click **Refresh** to refresh the Component Details. Click **Precheck** to run patching prechecks. You must run the patching prechecks before updating the storage components. Click **View Pre-check Reports** to view the patching prechecks report. If there are no errors in the report, then click **Apply Patch** to begin the job to patch the storage components. Click **View Jobs** for the job status.

   For high-availability environment, you can select the **Rolling** check box to perform rolling patching of storage components.
7. Click **Activity** to monitor the progress, and verify that the job completes successfully.

Updating Oracle Database Appliance Repository with Database Clone Files For Bare Metal Systems

Follow these steps to update the Oracle Database Appliance repository with Oracle Database clone files for the latest release using CLI commands.

To create new databases of a specific release version, you must update the Oracle Database Appliance repository with the Oracle Database clone files for a release. Follow these steps to update the repository with Oracle Database clone files for the latest release.

1. Download the Oracle Database clone software file for a release, for example, `p30403662_1925000_Linux-x86-64.zip`, and extract the Oracle Database clone software:

   For example, download the Oracle Database clone software patch for 19.25:

   ```
   unzip /tmp/p30403662_1925000_Linux-x86-64.zip
   odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip
   ```

2. Copy all the software files from the external client to Oracle Database Appliance. For High-Availability deployments, copy the software files to only one node. The software files are copied to the other node during the patching process. Use the `scp` or `sftp` protocol to copy the bundle.

   Example using `scp` command:

   ```
   # scp software_file root@oda_host:/tmp
   ```

   Example using `sftp` command:

   ```
   # sftp root@oda_host
   ```

   Enter the `root` password, and copy the files.

   ```
   put software_file
   ```

3. Update the repository with latest release database software:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
   odacli-dcs-db-clone-file.zip/
   ```

   For example, for 19.25:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
   odacli-dcs-19.25.0.0.0-date-DB-19.25.0.0.zip
   ```

   Follow the same steps to update the repository with the database clone software for other supported database versions.

4. Confirm that the repository update is successful:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli list-jobs
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-job -i job_ID
   ```

5. Delete the software zip files from the temporary location on the appliance to save space:

```
[root@oda1 opt]# rm software_file
rm: remove regular file software_file? y
```

Patching Databases Using ODACLI Commands or the BUI

Use ODACLI commands or the Browser User Interface to patch databases to the latest release in your deployment.

Before patching the database home, upload the Oracle Database clone files for the database version, to the repository. See *Updating Oracle Database Appliance Repository with Database Clone Files Using the CLI* for the procedure to update the repository with the latest Oracle Database clone files.

> **Important:**
>
> You must run the `odacli create-prepatchreport` command before you patch the Oracle databases; otherwise, the `odacli update-database` command fails with an error message prompting you to run the patching pre-checks.

Patching Databases on Oracle Database Appliance using ODACLI Commands

Run the following command to patch a database using the CLI:

```
odacli update-database [-a] [-dp] [-f] [-i db_id] [-imp] [-l] [-n db_name] [-
ni node] [-r] [-to db_home_id] [-j] [-h]
```

For more information about the options for the `update-database` command, see the chapter *Oracle Database Appliance Command-Line Interface*.

Patching Databases on Oracle Database Appliance using BUI

1. Log into the Browser User Interface with the `oda-admin` user name and password.

   ```
   https://Node0-host-ip-address:7093/mgmt/index.html
   ```

2. Navigate to the **Database** tab.
3. Select the database you want to patch.
4. Click **Update**.
5. If you select **Apply Data Patch**, then the Data Patch for the specified database is applied and you cannot select any other options.
6. On a high-availability system, you can also select the node in the **Select Node to Update** list.
7. Select **Ignore Missing Patches** to ignore missing patches.
8. Select **Force Run** to force the operation to run.
9. Select the destination database home.
10. In the **Patching Options**

- **Abort**: To abort previously unfinished or failed patching operation.
- **Revert**: To revert previously unfinished or failed patching operation.
- **None**: To patch the database.

11. Click **Update**.
12. If you have not run the pre-checks earlier, then an error is displayed when you submit the job to update the database.
13. In the Database page, select the Database and then click **Precheck** to run pre-checks for patching the database.

    Click **Activity** for job status.
14. In the Database page, for the database to be patched, click **Actions** and select **View Pre-patch reports** to view the pre-check report. Fix any errors, and then select Action as **Apply** to patch the database.
15. Verify that the patching job completes successfully.

Patching Existing Database Homes Using ODACLI or the BUI

Use ODACLI or BUI to patch database homes in your deployment to the latest release.

Before patching the database, upload the Oracle Database clone files for the database version, to the repository. See *Updating Oracle Database Appliance Repository with Database Clone Files Using the CLI* for the procedure to update the repository with the latest Oracle Database clone files.

Patching Database Homes on Oracle Database Appliance using ODACLI Commands

Run the following command to patch a database home using the CLI:

```
odacli update-dbhome -i dbhome_id -v version [-f] [-imp] [-p] [-l] [-u
node_number] [-j] [-h]
```

For more information about the options for the `update-dbhome` command, see the chapter *Oracle Database Appliance Command-Line Interface*.

Patching Database Homes on Oracle Database Appliance using BUI

1. Log into the Browser User Interface with the `oda-admin` user name and password.

   `https://Node0-host-ip-address:7093/mgmt/index.html`
2. Navigate to the **Database Home** tab.
3. Select the database home you want to patch.
4. Select the **Patch Version** for the database home.
5. To patch multiple database homes, select each database home to be patched and the patch version for each database home.
6. Select the **Node to Update**. You can choose the node that you want to update or you can choose to update all nodes.
7. Click **Patch**. Select **Precheck** to run pre-checks before patching the database.

   Click **Activity** for job status.

8. On the Patch page, for the database to be patched, click **Actions** and select **View Pre-patch reports** to view the pre-check report. Fix any errors, and then select Action as **Apply** to patch the database.
9. Select **Ignore Precheck Failures** to ignore failures reported in the prechecks reported. It is recommended that you fix errors reported in the precheck results.
10. Select **Ignore Missing Patches** to ignore missing patches.
11. Verify that the patching job completes successfully.

Related Topics

- Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line
  Follow these steps to apply patches to your Oracle Database Appliance bare metal deployment and existing Oracle Database homes, using CLI commands.

Related Topics

- Oracle Database Appliance Release Notes

Patching Non-Provisioned Oracle Database Appliance Systems

Oracle Database Appliance supports patching of non-provisioned single-node or high availablity systems by running `odacli update-server` command.

A non-provisioned Oracle Database Appliance system does not have Oracle Grid Infrastructure or Oracle Database installed or running on it. You can upgrade Oracle ILOM and storage firmware on an unprovisioned system without running the `odacli create-appliance` command by reimaging the system using the bare metal ISO image of Oracle Database release 19.21 or later. Then patch the system with the server patches of the same release as the bare metal ISO image by running the `odacli update-server` command. When you reimage your system, the DCS infrastructure is updated automatically.

Follow these steps to update the firmware:

1. Reimage the Oracle Database Appliance using the bare metal ISO image for Oracle Database Appliance release 19.21 or later.
2. Update the repository with the server patch for the Oracle Database Appliance release you want to patch to, for example 19.21 or later.
3. For running `odacli update-server` on Oracle Database Appliance imaged with ISO of release 19.21:

   ```
   odacli update-server -v 19.21.0.0.0 -sko -f
   ```

   For Oracle Database Appliance release 19.21, you can use the `-sko` option to skip ORAchk failures and proceed with the server patching, and the `-f` option to override precheck failures and complete the server patching.
4. For running `odacli update-server` on Oracle Database Appliance imaged with ISO of release 19.11 or later, follow these steps:
   a. Create the pre-patch report for the `odacli update-server` command by specifying the `-s` option.

      ```
      odacli create-prepatchreport -v 19.11.0.0.0 -s
      ```

Fix the warnings and errors mentioned in the report and proceed with the server patching.

b. Update the server:

```
odacli update-server -v 19.11.0.0.0 -f
```

For Oracle Database Appliance release 19.11 or later, you can use the `-f` option to override precheck failures and complete the server patching.

Applying Additional Patches and Updates on Bare Metal Systems

Configure additional patch repositories and patch your operating system and databases to ensure that your deployment has the latest updates and security fixes.

- Enabling Kernel Updates
  Oracle Ksplice enables you to update your systems with the latest kernel security and bug fix updates.
- Applying Out-of-Cycle Database Patches
  The out-of-cycle patching feature enables administrators to patch databases in the Oracle Database Appliance deployment with the latest Oracle Database Release Update, outside of the Oracle Database Appliance releases.

Enabling Kernel Updates

Oracle Ksplice enables you to update your systems with the latest kernel security and bug fix updates.

You can download and install the latest Oracle Ksplice updates (uptrack-updates) with either the local ULN repository or ULN channel.

Follow these steps to install Oracle KSplice:

1. Register your server with Unbreakable Linux Network (ULN). By default, you are registered for the Oracle Linux Latest channel for your operating system and hardware.
2. Log in to Unbreakable Linux Network:

   https://www.oracle.com/in/linux/
3. Ensure that you have access to the Internet on the server where you want to use Ksplice. For example, if you are using a proxy server, then set the proxy server and port values in the shell with commands similar to the following:

   ```
   # export http_proxy=http://proxy.company.com:port
   # export https_proxy=http://proxy.company.com:port
   ```
4. Register the server:

   ```
   # uln_register
   ```
5. Set up the local repository as described in the Oracle Ksplice user documentation. This enables you to download the latest Ksplice update packages at regular intervals without requiring a direct connection to the Oracle Uptrack server or to ULN.

6. To update a system to a specific effective kernel version, install the `uptrack-updates` package for the current kernel.

```
yum -y install uptrack-updates-'uname -r'
```

You can also download the `uptrack-updates` RPM from the Ksplice channel and install the RPM manually. For example:

```
# rpm -Uvh uptrack-
updates-4.14.35-1902.11.3.1.el7uek.x86_64-20200325-0.noarch.rpm
```

> **See Also:**
>
> *Oracle Linux Ksplice User's Guide* for information about configuring Oracle Ksplice at:
> https://docs.oracle.com/en/operating-systems/oracle-linux/

Applying Out-of-Cycle Database Patches

The out-of-cycle patching feature enables administrators to patch databases in the Oracle Database Appliance deployment with the latest Oracle Database Release Update, outside of the Oracle Database Appliance releases.

You can apply Oracle Database Release Update (RU) patches manually using OPatch. Refer to the relevant section in this topic to run the OPatch for your database version.

> **Note:**
>
> The OPatch commands and procedures described in this topic are examples, and may need modifications based on a specific Oracle Database Release Update (RU). Always refer to the Readme of the Oracle Database Release Update (RU) for additional commands and steps.

> **Note:**
>
> It is strongly recommended that you patch your deployment through the Oracle Database Appliance releases, whenever possible.

> **Note:**
>
> You can apply out-of-cycle patches to baremetal and DB system deployments using this procedure.

Using OPatchauto to Manually Patch Oracle Database Release 12c and Later

1. Confirm that the `dcs-agent` and other components are updated to the latest release.

   ```
   # odacli describe-component
   # odacli list-dbhomes
   ```

2. Update OPatch to the latest version, if available:
   a. Download the latest OPatch version from My Oracle Support.
   b. Copy the files to the local host and extract them to the *ORACLE_HOME*`/OPatch` directory.
   c. Update the OPatch tool as `oracle` user.

      ```
      /bin/rm -rf $ORACLE_HOME/OPatch/
      su - oracle -c /usr/bin/unzip -q -o OPatch_zip_location/
      p6880880_122010_Linux-x86-64.zip -d $ORACLE_HOME
      ```

3. Download the latest database patches from My Oracle Support and copy the files to the local host.

   ```
   scp patchfilename root@hostname:directory
   ```

4. Unzip patches to an empty directory, for example, `/tmp/patchesDir` as `oracle` user. This creates sub directories under `/tmp/patchesDir` with bug numbers. If the patch you are applying is a Release Update (RU), then the patch directory has multiple sub-directories.

   ```
   /usr/bin/unzip -o location_of_patches/p29301631_180000_Linux-x86-64.zip -
   d /tmp/patchesDir
   ```

5. For Oracle Database Appliance High-Availability deployments, set up SSH Keys for secure connection.
6. Analyze the patches.

   ```
   $ORACLE_HOME/OPatch/opatchauto apply directory_with_patches_extracted -
   analyze -oh $ORALCE_HOME -log /tmp/opatchAutoAnalyzePatch.log
   ```

7. As `root` user, apply the patches.

   ```
   $ORACLE_HOME/OPatch/opatchauto apply directory_to_extract_patches -
   oh $ORACLE_HOME -inplace
   ```

8. When patching is completed on both nodes, run `utlrp.sql` and `catcon.pl` scripts to recompile invalid objects in the database.
9. Update the registry to ensure that system component values are registered.

   ```
   # odacli update-registry -n dbhome -f
   ```

Cleaning Up the Patch Repository

Use the Browser User Interface or Command Line Interface to delete obsolete or old patches from the repository.

About Cleaning Up the Patch Repository

After patching Oracle Database Appliance with the latest patches, old unused clone files may remain and consume space in the repository. Cleanup repository deletes the unused clone/patch files to reclaim space. You can clean up the repository using the Browser User Interface or ODACLI commands.

Cleaning Up the Patch Repository Using ODACLI command

```
# odacli cleanup-patchrepo [-cl clones] | [-v version [-comp component]]| [-l
local|-n NodeID]
```

Specify the clone file names and the node from which you want to delete the old patches.

For details about the command options, see the chapter *Oracle Database Appliance Command-Line Interface*.

Cleaning Up the Patch Repository Using the Browser User Interface

Follow these steps to clean up the patch repository:

1. In the Browser User Interface, click **Patch Manager**.
2. Click the **Cleanup Repository** tab, and then click **Refresh** to list the available patches in the repository.
3. Select the Components, such as old Oracle Grid Infrastructure or Oracle Database clone files that you want to clean up from the repository, and the Patch Version, for example, an older release of Oracle Database Appliance.
4. Click **Cleanup Clone Files** to clean up clone files for selected components from the repository at `/opt/oracle/oak/pkgrepos/orapkgs/clones/`.
5. Click **Cleanup Repository** to submit the cleanup repository job.
6. Click **Activity** to monitor the progress. When the job completes successfully, the Patch Repository is updated.

Related Topics

- odacli cleanup-patchrepo
  Use the `odacli cleanup-patchrepo` command to delete obsolete or old patches from the repository.

About Upgrading to a Different Database Home

Understand how to upgrade to a different database home using either the Browser User Interface or `odacli` commands.

Prerequisites for Upgrading Database Homes

You can upgrade database homes of the following releases:

- Oracle Database 12.2 to 19c
- Oracle Database 12.1 to 19c

Before upgrading to a different database home, upload the Oracle Database clone files for the database version, to the repository, and then create the database home. See *Patching Oracle Database Appliance Using the CLI* for the procedure to update the repository with the latest Oracle Database clone files.

About Upgrading Database Homes Using the Browser User Interface

Follow these steps to upgrade to a different database home using the Browser User Interface.

1. Log into the Browser User Interface with the `oda-admin` user name and password.

   `https://Node0-host-ip-address:7093/mgmt/index.html`

2. Click the **Database** tab, and then select a database, click **Actions**, and then click **Upgrade**.
3. Verify that the database upgrade job completes successfully.

About Upgrading Database Homes Using the Command-Line Interface

Run the following command to upgrade to a different database home using the CLI:

```
odacli upgrade-database -i Comma separated list of database ids -from source
dbhome id -to destination dbhome id [-j] [-h]
```

For more information about the options for the `upgrade-database` command, see the chapter *Oracle Database Appliance Command-Line Interface*.

Setting Dynamic Default Parameters

You can set dynamic default parameters using the BUI.

Setting Dynamic Default Parameters Using the Browser User Interface

Follow these steps:

1. From the Appliance page, click **Parameter Repository**.
2. The Parameter Repository page lists all the parameters in the repository. Currently, parameters of type RDBMS are supported.
3. Select **Update Parameter** to create a new configuration property.
4. In the Update Parameter page, specify the **Parameter Name** and **Value**, and provide an optional **Description**. Specify whether you want to create the parameter for the **Bare Metal** or **DB System**. You can also specify the whether the parameter applies to **All DB Systems** or you can select the DB system for which you want to create the parameter.
5. Click **OK**.
6. Confirm the action to submit the job. Monitor the job progress and check that it completes successfully.
7. To update a parameter, select the **Update** option from the **Actions** menu.

8. You can edit the Value, Description, and select whether the parameter applies to **Bare Metal** or **DB System**. You can also select the DB system for which you want to update the parameter or choose to apply the parameter to **All DB Systems**.
9. Click **Update**.
10. To delete a parameter, select the **Delete** option from the **Actions** menu. You can delete the parameter from the **Bare Metal** or **DB System**. You can also select the DB system for which you want to delete the parameter or choose to delete the parameter from **All DB Systems**.
11. Click **Delete**.

9

Managing Oracle Databases

Manage the Oracle Databases on your Oracle Database Appliance.

- About Administrative Groups and Users on Oracle Database Appliance
  Oracle Database Appliance Browser User Interface deployment creates operating system groups and users whose members are granted system administration privileges on the appliance.
- About Data Migration Options for Oracle Database Appliance
  Oracle Database Appliance supports the use of standard Oracle Database loading and migration tools.
- About Creating Database Homes on Oracle ACFS Storage
  When you create new Oracle Database homes, they are created on Oracle ACFS file system which are stored on Oracle Database Appliance storage disks. You can create new Oracle Database homes using the `odacli create-database` or `odacli create-dbhome` commands, or when you patch an existing database home.
- About Standard Edition High Availability for Oracle Database Appliance
  Oracle Database Appliance supports Standard Edition High Availability solution with Oracle Grid Infrastructure that provides cluster-based failover for Oracle Database 19c Standard Edition.
- About Enterprise Edition High Availability for Oracle Database Appliance
  Oracle Database Appliance supports Enterprise Edition High Availability solution with Oracle Grid Infrastructure that provides cluster-based failover for Oracle Database Enterprise Edition.
- About Transparent Database Encryption (TDE) in Oracle Database Appliance
  Transparent Data Encryption (TDE) enables you to encrypt sensitive data that you store in tables and tablespaces.
- Storing Transparent Database Encryption Keys on Oracle Key Vault Server
  Understand how you can create Transparent Database Encryption (TDE) enabled databases on Oracle Database Appliance and store the TDE keys on Oracle Key Vault server.
- Working with Databases
  Use the Browser User Interface to display a list of databases, database details, and create and delete databases. You can use CLI commands to manage your databases.
- Working with Database Homes
  Use the Browser User Interface to display a list of database homes, details, and create and delete database homes.
- Migrating Databases
  Review these topics to learn how to prepare for and migrate an entire database to your Oracle Database Appliance.
- About Registering a TDE-Enabled Database
  Understand the process to register Transparent Data Encryption (TDE) enabled databases.
- About Managing Multiple Database Instances Using Instance Caging
  Use instance caging to manage your system resources on Oracle Database Appliance.

- Oracle EM Express and DB Console
  You can use Oracle Enterprise Manager Database Express (EM Express), or the Database Control Console (DB Console) to manage your database.

About Administrative Groups and Users on Oracle Database Appliance

Oracle Database Appliance Browser User Interface deployment creates operating system groups and users whose members are granted system administration privileges on the appliance.

During configuration, two administrative accounts are created for Oracle Database Appliance: the user `grid`, with a user ID (UID) of 1001, and the user `oracle`, with a UID of 1000. The user `grid` is the Oracle Grid Infrastructure installation owner. The user `oracle` is the Oracle Database installation owner, and the owner of all Oracle Database homes (Oracle homes). By default, these users are members of operating system groups whose members are granted privileges to start up and administer Oracle Database and Oracle Automatic Storage Management.

The following table describes the Oracle system privileges groups, and information about the operating system authentication groups:

Table 9-1 Operating System Groups and Users on Oracle Database Appliance

Oracle System Privileges	Group Name	Group ID (GID)	*grid* is a member	*oracle* is a member
Oracle Inventory group (OINSTALL)	`oinstall`	1001	yes (primary group)	yes (primary group)
OSOPER for dbaoper group	`dbaoper`	1002	yes	yes
OSDBA group	`dba`	1003	no	yes
OSASM group for Oracle ASM	`asmadmin`	1004	yes	no
OSOPER for ASM group	`asmoper`	1005	yes	no
OSDBA for ASM group	`asmdba`	1006	yes	yes

To change the Group Name and GID from the default values on Oracle Database Appliance bare metal platforms, change the default values from the Browser User Interface during the deployment. If you create an initial database during deployment, then the password for the SYS and SYSTEM users is the password that you set in the Browser User Interface.

To change the Group Name and GID from the default values on the Oracle Database Appliance Virtualized Platform, use the `-advance` parameter with the command `oakcli deploy`. If you create an initial database during deployment, then the password for the SYS and SYSTEM users is the ROOT password from the Configurator.

> **Note:**
>
> Change the password for both users as soon as possible after configuration to prevent unauthorized access to your database using these privileged accounts.

About Data Migration Options for Oracle Database Appliance

Oracle Database Appliance supports the use of standard Oracle Database loading and migration tools.

If you are loading data or migrating data from an existing database to Oracle Database Appliance, then you can use the standard Oracle Database loading and migration tools. These tools include the following:

- Oracle GoldenGate
- SQL*Loader
- Oracle Data Pump
- transportable tablespaces
- RMAN

You can also use the RMAN utility to back up and recover databases on Oracle Database Appliance.

About Creating Database Homes on Oracle ACFS Storage

When you create new Oracle Database homes, they are created on Oracle ACFS file system which are stored on Oracle Database Appliance storage disks. You can create new Oracle Database homes using the `odacli create-database` or `odacli create-dbhome` commands, or when you patch an existing database home.

Oracle Database homes are created on an Oracle ACFS-managed file system. For a database user `oracle`, the new database homes are created under `/u01/app/odaorahome/oracle/`.

The mount `/u01/app/odaorahome` is a shared file system for a high-availability system. On a high-availability system, `/u01/app/odaorahome/` is a shared ORACLE_HOME between the nodes. Existing database homes continue to remain on the local disks.

The ORACLE_BASE location has also changed to an Oracle ACFS-based file system. On high-availability system, the ORACLE_BASE is not shared between the nodes. Each node has its own Oracle ACFS file system for ORACLE_BASE. The new ORACLE_BASE for the database homes are created under the mount `/u01/app/odaorabase/`.

The disk space for the Oracle ACFS file system is located in the DATA or RECO disk group. You must specify the disk group for the Oracle ACFS file system. For Oracle Database Appliance X8-2 hardware model and single-node systems, the Oracle ACFS file system must be located in the DATA disk group. You can specify the initial size of the ORACLE_HOME volume. The default value is 80 GB. You can set a value from 25 GB to 1024 GB. The initial volume size for ORACLE_BASE is fixed at 10GB. Note that for these volumes, AUTOEXTEND is off. You can extend the volume size with the `odacli modify-dbhome-storage` command.

Changes When Provisioning Oracle Database Appliance Release 19.25

As part of the provisioning request, you must specify the disk group and size of the volume in the json file in the sysOraHomeStorage section.

```
"sysOraHomeStorage" : {
  "diskgroup" : "DATA",
```

```
    "volumeSize" : "80"
  },
```

If you do not specify the values in the json file, then the following error is displayed during provisioning:

```
DCS-10045:Validation error encountered: Missing argument : sysOraHomeStorage
```

If the values are invalid, then the following errors may be displayed during provisioning:

```
DCS-10024:Input parameter 'Disk Group' contains unacceptable value 'DATAL'.
Allowed values are '[DATA, RECO]'. (case insensitive)
DCS-10045:Validation error encountered: Required volumeSize value in
sysOraHomeStorage should be in the range of 25GB to 1024GB.
```

Changes When Patching to Oracle Database Appliance Release 19.25

If you have patched your appliance from Oracle Database Appliance release 19.6 or later to 19.25, then run the `odacli configure-dbhome-storage` command **before** creating a new database home. Otherwise, the following error is displayed when creating a new database home:

```
# odacli create-dbhome -v 19.8.0.0.200714
DCS-10601:The system is not set up to create database homes on ACFS.
```

Once the disk group for the volumes are setup, new database homes are created on the Oracle ACFS file system.

Related Topics

- odacli configure-dbhome-storage
  Use the `odacli configure-dbhome-storage` command to configure the database home storage.
- odacli list-dbhome-storages
  Use the `odacli list-dbhome-storages` command to view the database home storage resources after the volumes for ORACLE_HOME and ORACLE_BASE are created.
- odacli describe-dbhome-storage
  Use the `odacli describe-dbhome-storage` command to view information such as how much space is available (at a volume or diskgroup level).
- odacli modify-dbhome-storage
  Use the `odacli modify-dbhome-storage` command to modify database home storage and add additional disk space.
- Managing Database Homes on Oracle ACFS Storage
  Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to manage database home storage in Oracle Database Appliance.

About Standard Edition High Availability for Oracle Database Appliance

Oracle Database Appliance supports Standard Edition High Availability solution with Oracle Grid Infrastructure that provides cluster-based failover for Oracle Database 19c Standard Edition.

About Standard Edition High Availability for Oracle Database 19c

With Standard Edition High Availability, when there is an instance or a node failure, the database automatically fails over to the other node, after attempting to restart on the local node. The database is restarted on the surviving node, thereby providing high availability for Oracle Standard Edition databases.

If your Oracle Database deployment contains Standard Edition 2 Oracle Real Application Cluster (Oracle RAC) databases of releases earlier than 19c, then they must be converted to a Standard Edition High Availability configuration as part of the upgrade to Oracle Database 19c.

For more details about this feature, refer to the Oracle Database Documentation Library at https://docs.oracle.com/en/database/oracle/oracle-database/index.html.

About Standard Edition High Availability for Oracle Database Appliance Release 19.25

Standard Edition High Availability for Oracle Database 19c is supported on Oracle Database Appliance for high availability deployments, that is, deployments that have two server nodes, shared storage, and server interconnects.

Oracle Database Appliance enables automatic configuration for failover at deployment time for Oracle Database 19c Standard Edition single-instance databases. Both nodes have Oracle Homes for Standard Edition single-instance databases.

Single-instance Oracle Database 19c Standard Edition databases created using the ODACLI commands or the Browser User Interface (BUI) have Standard Edition High Availability enabled by default. You can also choose to disable the High Availability option. To enable Standard Edition High Availability for existing Standard Edition Oracle RAC databases of releases earlier than 19c, you must convert these Oracle RAC or Oracle One Node databases to single-instance databases, and then upgrade to single-instance Oracle Database 19c Standard Edition database with Standard Edition High Availability enabled by default.

About Provisioning, Upgrading, and Relocating Standard Edition High Availability Oracle Database

To create a single-instance Oracle Database 19c with Standard Edition High Availability, use the following command:

```
odacli create-database -u db_unique_name -n db_name -dh db_home -y SI -g
target_node
```

To upgrade a Standard Edition Oracle RAC Database to Oracle Database 19c Standard Edition, convert the Oracle RAC Database to a single-instance Oracle Database:

```
odacli modify-database -in db_name -y single-instance
```

Then, upgrade the single-instance database to Oracle Database 19c with Standard Edition High Availability as follows:

```
odacli upgrade-database -i db_id -to destination_dbhome -sh
```

To relocate a Standard Edition High Availability Oracle Database from one node to another outside of a failover, use the following command, specifying either the target node or the target host in the command:

```
odacli modify-database -in db_name -g target_node
```

or:

```
odacli modify-database -in db_name -th target_host
```

For detailed information about these ODACLI command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- Oracle Database Appliance Command-Line Interface
  The command-line interface has different classes of tools to manage Oracle Database Appliance.

About Enterprise Edition High Availability for Oracle Database Appliance

Oracle Database Appliance supports Enterprise Edition High Availability solution with Oracle Grid Infrastructure that provides cluster-based failover for Oracle Database Enterprise Edition.

About Enterprise Edition High Availability for Oracle Database

With Enterprise Edition High Availability, when there is an instance or a node failure, the database automatically fails over to the other node, after attempting to restart on the local node. The database is restarted on the surviving node, thereby providing high availability for Oracle Enterprise Edition databases.

For more details about this feature, refer to the Oracle Database Documentation Library at https://docs.oracle.com/en/database/oracle/oracle-database/index.html.

About Enterprise Edition High Availability for Oracle Database Appliance Release 19.25

Enterprise Edition High Availability for Oracle Database 19.25 is supported on Oracle Database Appliance for high availability deployments, that is, deployments that have two server nodes, shared storage, and server interconnects.

Oracle Database Appliance enables automatic configuration for failover at deployment time for Oracle Database 19.25 Enterprise Edition single-instance databases. Both nodes have Oracle Homes for Enterprise Edition single-instance databases.

To enable Enterprise Edition High Availability for existing Enterprise Edition Oracle RAC databases of release 19.25, you must convert these Oracle RAC databases to single-instance databases, and then upgrade to single-instance Oracle Database 19.25 Enterprise Edition database with Enterprise Edition High Availability enabled by default.

About Provisioning, Upgrading, and Relocating Enterprise Edition High Availability Oracle Database

To create a single-instance Oracle Database 19.25 with Enterprise Edition High Availability, use the following command:

```
odacli create-database -u db_unique_name -n db_name -dh db_home -y SI -g
target_node -ha
```

To upgrade an Enterprise Edition Oracle RAC Database to Oracle Database 19.25 Enterprise Edition, convert the Oracle RAC Database to a single-instance Oracle Database:

```
odacli modify-database -in db_name -y single-instance
```

Then, upgrade the single-instance database to Oracle Database 19.25 with Enterprise Edition High Availability as follows:

```
odacli upgrade-database -i db_id -to destination_dbhome -ha
```

To relocate an Enterprise Edition High Availability Oracle Database from one node to another outside of a failover, use the following command, specifying either the target node or the target host in the command:

```
odacli modify-database -in db_name -g target_node
```

or:

```
odacli modify-database -in db_name -th target_host
```

For detailed information about these ODACLI command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- Oracle Database Appliance Command-Line Interface
  The command-line interface has different classes of tools to manage Oracle Database Appliance.

About Transparent Database Encryption (TDE) in Oracle Database Appliance

Transparent Data Encryption (TDE) enables you to encrypt sensitive data that you store in tables and tablespaces.

After the data is encrypted, this data is transparently decrypted for authorized users or applications when they access this data. TDE helps protect data stored on media (also called data at rest) in the event that the storage media or data file is stolen.

Oracle Database uses authentication, authorization, and auditing mechanisms to secure data in the database, but not in the operating system data files where data is stored. To protect these data files, Oracle Database provides Transparent Data Encryption (TDE).

TDE encrypts sensitive data stored in data files. To prevent unauthorized decryption, TDE stores the encryption keys in a security module external to the database, called a keystore.

Oracle Database Appliance provides the option to create TDE-enabled databases with the ability to store TDE keys either in software keystore on Oracle Database Appliance or on Oracle Key Vault. When the TDE keys are stored in software keystore, Oracle Database Appliance also provides the option to backup and restore TDE wallets.

Based on the keystore used, TDE on Oracle Database Appliance can be configured as any of the following types:

- **TDE keys stored in software keystore:** In this type of TDE method, the TDE keys are stored in a wallet on Oracle Database Appliance.
- **TDE keys stored in a wallet on Oracle Key Vault:** In this type of TDE method, the TDE keys are stored in a wallet on Oracle Key Vault. Note that if you do not want to provide the credentials of users in Oracle Key Vault, you can use endpoints already created on Oracle Key Vault, and directly specify the endpoints when you create or restore the database on Oracle Database Appliance. Otherwise, you can create Oracle Key Vault server configuration on Oracle Database Appliance with the Oracle Key Vault user details and use these user credentials to create and restore the database on Oracle Database Appliance. You can configure TDE keys with Oracle Key Vault keystore in one of the following ways:
  - **Using credentials of a user in Oracle Key Vault:** In this configuration, the TDE keys are stored in a wallet on Oracle Key Vault and accessed through Oracle Key Vault user credentials.
  - **Using endpoints and wallet already created on Oracle Key Vault:** In this configuration, the TDE keys are stored in a wallet on Oracle Key Vault and accessed through Oracle Key Vault end points.

Note:

To enable Transparent Data Encryption (TDE), you must deploy Oracle Database Enterprise Edition with Oracle Database 19c or later. Transparent Data Encryption (TDE) is an Advanced Security Option (ASO) available with Oracle Database Enterprise Edition.

See Also:

Using Transparent Data Encryption in the *Oracle Database Advanced Security Guide*

Caution:

When you enable TDE on databases, either with software keystore or with Oracle Key Vault keystore using endpoints and wallet already created on Oracle Key Vault, you are prompted to set a password for the TDE wallet. Provide a strong password for security compliance. Set the password carefully, and ensure that this password is available to you at all times for database management operations. Failure to provide the TDE wallet password when prompted, results in not being able to access the database tables.

Storing Transparent Database Encryption Keys on Oracle Key Vault Server

Understand how you can create Transparent Database Encryption (TDE) enabled databases on Oracle Database Appliance and store the TDE keys on Oracle Key Vault server.

- About 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.
- Creating TDE-Enabled Databases With User Credentials
  Understand how you can create TDE-enabled databases on Oracle Database Appliance using the credentials of a user in Oracle Key Vault. This method supports creation of TDE-enabled databases on bare metal and DB systems.
- Creating TDE-Enabled Databases Using Oracle Key Vault With Endpoints in Oracle Key Vault
  Understand how you can create TDE-enabled databases on Oracle Database Appliance using endpoints and wallets already in Oracle Key Vault. This method supports creation of TDE-enabled databases on bare metal and DB systems.
- Managing Oracle Key Vault Server Configuration Operations Using ODACLI Commands
  Understand how you can manage Oracle Key Vault server configuration operations on Oracle Database Appliance on bare metal and DB systems.
- Managing Database Operations on TDE-enabled Databases with TDE keys stored on Oracle Key Vault
  Understand how you can manage TDE-enabled databases with TDE keys stored on Oracle Key Vault on Oracle Database Appliance on bare metal and DB systems.
- Creating an Oracle Key Vault User Using BUI
  Create a user with Oracle Key Vault server configuration entitlement on a multi-user access enabled Oracle Database Appliance.
- Creating Oracle Key Vault Server Configuration Object Using the BUI
  Understand how you can create TDE-enabled database on Oracle Database Appliance with TDE keys stored in Oracle Key Vault on bare metal and DB systems using the BUI.
- Granting and Revoking Access on Oracle Key Vault Server Configuration Using BUI
  Understand how you can grant and revoke access to Oracle Key Vault server configuration on Oracle Database Appliance on bare metal and DB systems using BUI.
- Managing Oracle Key Vault Server Configuration Operations Using BUI
  Understand how you can manage Oracle Key Vault server configuration operations on Oracle Database Appliance on bare metal and DB systems.
- Manual Migration and Encryption of Data into TDE-Enabled Database
  Understand how you can migrate transportable tablespaces to TDE-enabled database on Oracle Database Appliance.

About 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 or 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) for compatibility with KMIP-based clients, for example MongoDB.

Terminology used when storing TDE keys on Oracle Key Vault for TDE-enabled databases on Oracle Database Appliance

Understand these concepts about using Oracle Key Vault server with databases on Oracle Database Appliance:

- **Role separation:** Oracle recommends that you assign different users for database administration and TDE key operations. You must not provide the user who has the database administration role with the credentials of the Oracle Key Vault server. This ensures the database user with data file access cannot open the database without knowing TDE wallet passwords. You can set up this role separation by enabling multi-user access feature on Oracle Database Appliance. For more information about configuring multi-user access, see the topic *Implementing Multi-User Access on Oracle Database Appliance*. On Oracle Database Appliance, the credentials of Oracle Key Vault user need not be shared with the ODA-DB user. The ODA-Adminstrator grants the resource, that is, the Oracle Key Vault server configuration, access to the intended DB user and the DB user can use the Oracle Key Vault server configuration when creating the TDE-enabled database to store the TDE keys on the Oracle Key Vault server.
- **Oracle Key Vault server configuration:** The term refers to an Oracle Database Appliance resource that stores the metadata of the Oracle Key Vault. The Oracle Key Vault server configuration has details such as the name of the user created to manage the TDE keys on the Oracle Key Vault server and the IP address or host name of the Oracle Key Vault server. The Oracle Key Vault server configuration on Oracle Database Appliance is identified by a name, to differentiate Oracle Key Vault server configurations. This name is unique within the Oracle Database Appliance system and the value is case-sensitive. When creating the Oracle Key Vault server configuration, you must provide the credentials of the user on Oracle Key Vault server. It is recommended that this user is a least privileged user with only `Create Endpoint` privilege. The user name and password you specify when you create the Oracle Key vault server configuration correspond to the user on the Oracle Key vault server. The password you provide during creation of this object is not stored on Oracle Database Appliance. The Oracle Key Vault client auto-login wallet is created using these credentials and all requests to the Oracle Key Vault server use this client auto-login wallet. The database automatically reads the auto-login wallet for the required ADMINISTER KEY MANAGEMENT SQL commands.
- **A user with ODA-DB role:** A user with ODA-DB role in a multi-user access-enabled or multi-user access-enabled passwordless environment can create and maintain the

database. This user, is granted the ODA-DB role and is entitled to perform all database related operations such as creation, patching, modification, deletion, backup, and recovery. These operations are restricted to the database that belongs to the user. For more details, see the chapter *Implementing Multi-User Access on Oracle Database Appliance* in this guide.

- **Oracle Key Vault user:** The Oracle Key Vault user name and password that you provide when you create the Oracle Key Vault server configuration must correspond to an existing user and the password on the Oracle Key Vault server. Otherwise, there can be failures in accessing Oracle Key Vault. This user can also perform create, delete, describe, and list operations on the Oracle Key Vault server configuration objects that they create.
- **Virtual Wallet:** A virtual wallet, created on Oracle Key Vault, is a container for objects that must be stored securely. For TDE-enabled databases on Oracle Database Appliance that use Oracle Key Vault as the keystore, the TDE keys of the database are stored inside this virtual wallet.

Requirements for configuring TDE-enabled databases with keys stored in Oracle Key Vault

- The Oracle Key Vault server version must be 21.7.0 or later.
- It is recommended that you set up Network Time Protocol on Oracle Database Appliance. The database operations interacting with Oracle Key Vault may fail if the clock on Oracle Database Appliance and Oracle Key Vault server are not synchronized. You must ensure that the time is consistent across the servers or use Network Time Protocol on Oracle Database Appliance.
- You cannot create a starter database with TDE configured using Oracle Key Vault when you provision Oracle Database Appliance on bare metal systems and select to create a starter database. This is because creation of TDE database that uses Oracle Key Vault to store TDE keys requires Oracle Key Vault server configuration to already exist. To create a database with TDE configured using Oracle Key Vault on DB system, first create a DB system without a starter database and then create the database with TDE configured using Oracle Key Vault as described in the topic *Creating TDE-Enabled Database on Oracle Database Appliance Using Oracle Key Vault*.
- For configuring TDE-enabled databases with keys stored in Oracle Key Vault with the credentials of a user in Oracle Key Vault, on multi-user access-enabled systems, you must create a new user with the `ODA-OKVCONFIGADMIN` role which has the entitlements `ODA-OKVCONFIGVIEWMGMT` and `ODA-OKVCONFIGMODIFYMGMT`. The `ODA-OKVCONFIGVIEWMGMT` entitlement enables you to perform operations such as Describe and List on the Oracle Key Vault server configuration object. The `ODA-OKVCONFIGMODIFYMGMT` entitlement enables you to perform operations such as Create and Delete on the Oracle Key Vault server configuration object. The `ODA-OKVCONFIGVIEWMGMT` entitlement is included in the `ODA-DB` role, and any DB user with the `ODA-DB` role can list and describe the Oracle Key Vault server configuration object and select the required Oracle Key Vault server configuration object to create the database. For non-multi user access-enabled systems, there is no need to create a new user.

Related Topics

- Implementing Multi-User Access on Oracle Database Appliance
  Understand how multi-user access can be implemented on Oracle Database Appliance, its advantages and restrictions and the associated life cycle management changes for your appliance.
- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

- ODACLI Command Changes with Multi-User Access on Oracle Database Appliance
  Understand the ODACLI command changes when you enable multi-user access.

Creating TDE-Enabled Databases With User Credentials

Understand how you can create TDE-enabled databases on Oracle Database Appliance using the credentials of a user in Oracle Key Vault. This method supports creation of TDE-enabled databases on bare metal and DB systems.

Follow these steps to create a TDE-enabled database with TDE keys stored on Oracle Key Vault on Oracle Database Appliance bare metal and DB systems:

Step 1: Create the user on Oracle Database Appliance corresponding to the user on Oracle Key Vault. This step applies to multi-user access-enabled systems only.

If the user with the `ODA-OKVCONFIGADMIN` role does not exist, or if you want to assign a different user for the database you are creating, then create the user.

1. Create a new user with the `ODA-OKVCONFIGADMIN` role. For example:

   ```
   odacli create-user -u okvuser1 -r ODA-OKVCONFIGADMIN
   ```

   The user is created and assigned a temporary password.
2. After the user is created successfully, the `okvuser1` can log into the appliance with the temporary password.
3. The `okvuser1` is in the `Inactive` state. If the appliance is configured with multi-user access enabled, then activate the user with the following command. Note that if the appliance is configured with passwordless multi-user access, then you do not need to activate the user. For more information about configuring multi-user access, the topic *Implementing Multi-User Access on Oracle Database Appliance*.

   ```
   odacli activate-user
   ```

   You are prompted to change the password. Enter the temporary password, the new password, and confirm the new password.
4. Use the new password to connect by SSH into the appliance and run ODACLI commands or connect to the Browser User Interface.

Step 2: Create an Oracle Key Vault server configuration

1. The Oracle Key Vault user that creates the Oracle Key Vault server configuration must be a user with the least privilege of `Create Endpoint` on the Oracle Key Vault server.
2. For multi-user access-enabled systems, log in as the user with the `ODA-OKVCONFIGADMIN` role, for example, the `okvuser1`. For non-multi user access enabled systems, log in as the `root` user.
3. The Oracle Key Vault password prompt displays twice, to accept and confirm the Oracle Key Vault password. Specify the password of the Oracle Key Vault user on the Oracle Key Vault server, which enables the user to log into the Oracle Key Vault server. Specify the user name with the `-u` option in the `odacli create-okvserverconfig` command. The name of the Oracle Key Vault server configuration is case-sensitive and no two Oracle Key

Vault server configurations can have same name on Oracle Database Appliance. The length of the Oracle Key Vault server configuration cannot exceed 128 characters.

```
odacli create-okvserverconfig -n okvobj1 -u epokvusr1 -ip xxx.xx.xxx.xxx
Enter OKV user password :
Retype OKV user password :
Job details
----------------------------------------------------------------
                     ID:  730d2469-6efa-4ed1-8386-513e97c866ea
            Description:  Create OKV Server Config: okvobj1
                 Status:  Created
                Created:  October 20, 2024 11:28:39 AM PDT
                Message:  Create OKV Server Config.

Task Name                                Start
Time                                 End
Time                                   Status
---------------------------------------
---------------------------------------
---------------------------------------- ----------------


 odacli describe-job -i 730d2469-6efa-4ed1-8386-513e97c866ea

Job details
----------------------------------------------------------------
                     ID:  730d2469-6efa-4ed1-8386-513e97c866ea
            Description:  Create OKV Server Config: okvobj1
                 Status:  Success
                Created:  October 20, 2024 11:28:39 AM PDT
                Message:  Create OKV Server Config.

Task Name                                Node Name                 Start
Time                                 End
Time                                   Status
---------------------------------------- -------------------------
---------------------------------------
---------------------------------------- ----------------
OKV Password Store creation              n1                        October
20, 2024 11:28:39 AM PDT         October 20, 2024 11:29:04 AM PDT
Success
Download of OKV REST package             n1                        October
20, 2024 11:28:42 AM PDT         October 20, 2024 11:28:44 AM PDT
Success
Edit of OKV REST package                 n1                        October
20, 2024 11:28:44 AM PDT         October 20, 2024 11:28:45 AM PDT
Success
Generate OKV client wallets              n1                        October
20, 2024 11:28:45 AM PDT         October 20, 2024 11:28:47 AM PDT
Success
```

The command creates an Oracle Key Vault server configuration `okvobj1` for the Oracle Key Vault server user `epokvusr1` on the specified server.

Step 3: Share Oracle Key Vault server configuration for multi user access-enabled systems only

1. For multi-user access-enabled systems, the DB user who creates the database using the Oracle Key Vault server configuration must have shared access to the Oracle Key Vault server configuration created by the Oracle Key Vault user. The `odaadmin` user must share the Oracle Key Vault server configuration with the DB user using the `odacli grant-resource-access` command. The DB user can then create the database with the Oracle Key Vault server configuration. An example where the `odaadmin` user shares the `okvobj1` resource with DB user `dbusr1` is as follows:

   ```
   su odaadmin

   odacli grant-resource-access -ri d017d35a-0757-460a-98f0-9f8d0944c872 -u
   dbusr1
   Shared access of resource granted to user successfully...
   ```

   To get the resource ID of the Oracle Key Vault server configuration, run the `odacli list-okvserverconfigs` command. For example:

   ```
   odacli list-okvserverconfigs

   ID                                       Name                 User
   Name         IP address          Host Name           Description
   ---------------------------------------- --------------------
   ---------------- ------------------ --------------------
   -------------------
   d017d35a-0757-460a-98f0-9f8d0944c872     okvobj1
   epadmin           xxx.xx.xx.xxx       Null                 OKV Server for
   sales databases
   ```

Step 4: Create the database

For multi-user access-enabled systems, log in as the DB user and use the `odacli create-database` command with the options `-t`, `-osc` and `-kt` to create a database with TDE configured using Oracle Key Vault server. For non-multi user access-enabled systems, you can run the same command as `root` user.
Determine the Oracle Key Vault server you want to store the TDE keys on and provide the value with the `-osc` option in the `odacli create-database` command. The DB user can select the required Oracle Key Vault server configuration from the list of Oracle Key Vault server configurations displayed with the `odacli list-okvserverconfigs` command. For non-multi user access enabled systems, run the `odacli list-okvserverconfigs` command as `root` user.

- The option `-t` indicates that the database must be configured with TDE, using the Oracle Key Vault credentials represented by the Oracle Key Vault server configuration specified using the `-osc` option.
- The option `-kt` specifies the the type of kesytore used in TDE configuration. The value of the keystore can be either `software` or `okv`. The default is `software`. The values are case-insensitive. The `OKVServerConfigName` attribute of the database is set to the name of the Oracle Key Vault server configuration specified in the create database request.
- The password for TDE wallet is randomly generated by the system, and you do not need to specify it.

An example to create a database `kokvdb2` which uses Oracle Key Vault credentials represented by Oracle Key Vault server configuration `kovobj1` is as follows:

```
odacli create-database -n kokvdb2 -t -kt OKV -osc kovobj1
Enter SYS and SYSTEM user password:
Retype SYS and SYSTEM user password:

Job details
----------------------------------------------------------------
                     ID:  94a51f1a-2d4c-4eac-b0c2-7871d88c0df8
            Description:  Database service creation with DB name: kokvdb2
                 Status:  Created
                Created:  July 15, 2024 3:00:53 PM CST
                Message:

Task Name                                Start
Time                                End Time
Status
---------------------------------------
---------------------------------------
---------------------------------------- ----------------



 odacli describe-job -i 94a51f1a-2d4c-4eac-b0c2-7871d88c0df8

Job details
----------------------------------------------------------------
                     ID:  94a51f1a-2d4c-4eac-b0c2-7871d88c0df8
            Description:  Database service creation with DB name: kokvdb2
                 Status:  Success
                Created:  July 15, 2024 3:00:53 PM CST
                Message:

Task Name                                Start
Time                                End Time
Status
---------------------------------------
---------------------------------------
---------------------------------------- ----------------
Validating dbHome available space        July 15, 2024 3:01:19 PM
CST             July 15, 2024 3:01:19 PM CST             Success
...
TDE configuration using OKV              July 15, 2024 3:17:37 PM
CST             July 15, 2024 3:20:36 PM CST             Success
TDE Wallet directory creation            July 15, 2024 3:17:38 PM
CST             July 15, 2024 3:17:46 PM CST             Success
Setting Database parameter for OKV       July 15, 2024 3:17:46 PM
CST             July 15, 2024 3:19:07 PM CST             Success
Auto login TDE Wallet creation           July 15, 2024 3:19:07 PM
CST             July 15, 2024 3:19:09 PM CST             Success
OKV TDE Wallet creation                  July 15, 2024 3:19:09 PM
CST             July 15, 2024 3:19:12 PM CST             Success
OKV Endpoint configuration               July 15, 2024 3:19:12 PM
CST             July 15, 2024 3:19:43 PM CST             Success
Setting TDE Master Encryption Key        July 15, 2024 3:19:43 PM
```

```
CST             July 15, 2024 3:20:13 PM CST             Success
Create Users tablespace                  July 15, 2024 3:20:36 PM
CST             July 15, 2024 3:20:39 PM CST             Success
Clear all listeners from Database        July 15, 2024 3:20:39 PM
CST             July 15, 2024 3:20:40 PM CST             Success
Copy Pwfile to Shared Storage            July 15, 2024 3:20:43 PM
CST             July 15, 2024 3:20:47 PM CST             Success
```

On successful database creation, the Oracle Key Vault client software is installed in the `/etc/OKV/db_unique_name/okv` directory. The ownership of all the files is set to `db_user:oinstall` and the permission is set to 750. The autologin TDE wallets are created in the `/etc/OKV/db_unique_name/tde` and `/etc/OKV/db_unique_name/tde_seps` directories. The ownership of these directories is recursively set to `db_user:oinstall` and the permission to 750.

Related Topics

- Implementing Multi-User Access on Oracle Database Appliance
  Understand how multi-user access can be implemented on Oracle Database Appliance, its advantages and restrictions and the associated life cycle management changes for your appliance.
- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Creating TDE-Enabled Databases Using Oracle Key Vault With Endpoints in Oracle Key Vault

Understand how you can create TDE-enabled databases on Oracle Database Appliance using endpoints and wallets already in Oracle Key Vault. This method supports creation of TDE-enabled databases on bare metal and DB systems.

> **Note:**
>
> In this Oracle Database Appliance release, this feature is supported on Oracle RAC and Oracle single-instance databases only. This feature is not available with Oracle RAC One Node databases and Oracle single-instance database with high-availability enabled.

To create a TDE-enabled database using Oracle Key Vault with endpoints in Oracle Key Vault, you must complete the following tasks:

1. Log into Oracle Key Vault and create virtual TDE wallet corresponding to a database.
2. Create endpoints corresponding to each instance. For a single instance database, you must create one endpoint and for Oracle RAC database, create two endpoints.
3. Set the default wallet for the created endpoint.
4. Download the `okvclient.jar` file corresponding to each endpoint. For single instance database, download one file, and for Oracle RAC database, there are two files available for download, one per instance..

Step 1: Obtain the `okvclient.jar` file using Oracle Key Vault REST commands

You can create the `okvclient.jar` file either through Oracle Key Vault REST commands or Oracle Key Vault server user interface.

To obtain the `okvclient.jar` file using Oracle Key Vault REST commands, log in as the `oracle` user, and then follow these steps:

1. Create working directories, such as `/tmp/work` and `/tmp/work/client_wallet`. Navigate to the `/tmp/work` directory and download and unzip the RESTful Service utility from the Oracle Key Vault server. When you unzip the utility, you can observer three directories `bin`, `lib`, and `conf`.

   ```
   mkdir /tmp/work/
   mkdir /tmp/work/client_wallet
   cd /tmp/work
   curl -Ok https://okv_server_ip:5695/okvrestclipackage.zip
    % Total % Received % Xferd Average Speed Time Time Time Current
                                    Dload Upload Total Spent Left Speed
   100 3750k 100 3750k 0 0 2500k 0 0:00:01 0:00:01 --:--:-- 2500k
   ```

2. Unzip `okvrestclipackage.zip`. Edit `conf/okvrestcli.ini` and add the Oracle Key Vault server IP address, user name, and path to the directory where the client autologin wallets are created. In the following example, `/tmp/work/client_wallet` is the directory path where client autologin wallets are created.

   ```
   #Provide absolute path for log_property, okv_client_config properties
           [Default]
           log_property=./conf/okvrestcli_logging.properties
           server=okv_server_ip
           okv_client_config=./conf/okvclient.ora
           user=okv_user
           client_wallet=/tmp/work/client_wallet
   ```

3. Edit the `bin/okv` file and uncomment the line `export OKV_RESTCLI_CONFIG=$OKV_RESTCLI_DIR/conf/okvrestcli.ini` and also set the `JAVA_HOME` variable:

   ```
   #!/bin/bash
       export OKV_RESTCLI_DIR=$(dirname "${0}")/..
       export OKV_RESTCLI_CONFIG=$OKV_RESTCLI_DIR/conf/okvrestcli.ini
       export JAVA_HOME=/opt/oracle/dcs/java/1.8.0_xxx
       if [ -z "$JAVA_HOME" ]
       then
         echo "JAVA_HOME environment variable is not set."
         exit 1
       fi

       if [ -z "$OKV_RESTCLI_CONFIG" ]
       then
         echo "OKV_RESTCLI_CONFIG environment variable is not set."
         exit 1
       fi
   ```

```
export OKV_RESTCLI_JAR=$OKV_RESTCLI_DIR/lib/okvrestcli.jar
$JAVA_HOME/bin/java -jar $OKV_RESTCLI_JAR "$@"
```

4. Generate client auto login wallets. You are prompted to provide the Oracle Key Vault user password.

   ```
   /tmp/work/bin/okv admin client-wallet add --client-wallet /tmp/work/
   client_wallet --wallet-user okv_user
   Password:
   ```

5. Create the TDE wallet corresponding to the database, if the TDE wallet is not already created using the Oracle Key Vault user interface.

   ```
   /tmp/work/bin/okv manage-access wallet create --wallet wallet_name --
   unique FALSE
   ```

6. Create the endpoint corresponding to the wallet. For Oracle RAC database, create two endpoints. Create the endpoint only if it is not already created using the Oracle Key Vault user interface.

   ```
   /tmp/work/bin/okv admin endpoint create --endpoint endpoint_name --type
   ORACLE_DB --platform LINUX64 --strict-ip-check TRUE
   ```

7. Set the default wallet for the endpoint. For Oracle RAC database, the default wallet for both endpoints must be the same TDE wallet. Set the default wallet only if it is not already set using the Oracle Key Vault user interface.

   ```
   /tmp/work/bin/okv manage-access wallet set-default --wallet wallet_name --
   endpoint endpoint_name
   ```

8. Download the `okvclient.jar` command. For Oracle RAC database, you must download the `okvclient.jar` corresponding to both the endpoints. After the download is complete, copy the `okvclient.jar` corresponding to the remote instance to the remote node.

   ```
   /tmp/work/bin/okv admin endpoint download --endpoint <endpoint_name> --
   location <location>
   ```

9. While downloading the endpoint, if the `Endpoint token is already consumed` message is observed as below, then re-enroll the endpoint.

   ```
   /tmp/work/bin/okv admin endpoint download --endpoint endpoint_name --
   location location
   /tmp/work/log/okv0.log.0 (Permission denied)
   {
     "result" : "Failure",
     "message" : "Endpoint token is already consumed"
   }

   /tmp/work/bin/okv admin endpoint re-enroll --endpoint <endpoint_name>
   /tmp/work/log/okv0.log.0 (Permission denied)
   {
     "result" : "Success"
   }
   ```

Step 2: Create the database

1. For multi user access-enabled and multi user access passwordless systems, log in as the DB user and run the `odacli create-database` command. For non-multi user access-enabled systems, log in as the `root` user and run the `odacli create-database` command. The option `--enable-tde/-t` indicates that the database must be configured with TDE. The option `--keystore-type/-kt` specifies the the type of keystore to use in TDE configuration. The option `--okvclient-path/-ocp` specifies the absolute path to the `okvclient.jar` file corresponding to the endpoint.

   For single-instance database, the option `--okvclient-path/-ocp` takes only one value. For Oracle RAC database, the option `--okvclient-path/-ocp` takes two values corresponding to each instance. The values must be comma-separated. The first path corresponds to the local instance whereas the second path corresponds to the remote instance. You are prompted to provide the TDE password.

   On successful database creation, the Oracle Key Vault client software is installed in the `/etc/OKV/db_unique_name/okv` directory. The local auto login TDE wallets are created in the `/etc/OKV/db_unique_name/tde` directory. The `okvclient.jar` file in the source location is deleted.

   An example to create a single-instance database named `okvdb` without using the Oracle Key Vault server configuration object, is as follows:

```
odacli create-database -n okvdb -y SI -t -kt OKV -ocp /tmp/okvclient.jar
Enter SYS and SYSTEM user password:
Retype SYS and SYSTEM user password:
Enter TDE wallet password:
Retype TDE wallet password:

 Job details
----------------------------------------------------------------
                     ID:  94a51f1a-2d4c-4eac-b0c2-7871d88c0df8
            Description:  Database service creation with DB name: okvdb
                 Status:  Created
                Created:  July 15, 2024 3:00:53 PM CST
                Message:

Task Name                                Start
Time                                 End
Time                                   Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------



 odacli describe-job -i 94a51f1a-2d4c-4eac-b0c2-7871d88c0df8

Job details
----------------------------------------------------------------
                     ID:  94a51f1a-2d4c-4eac-b0c2-7871d88c0df8
            Description:  Database service creation with DB name: okvdb
                 Status:  Success
                Created:  July 15, 2024 3:00:53 PM CST
                Message:
```

```
Task Name                                Start
Time                                 End
Time                                   Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Validating dbHome available space        July 15, 2024 3:01:19 PM
CST             July 15, 2024 3:01:19 PM CST             Success
...
TDE configuration using OKV              July 15, 2024 3:17:37 PM
CST             July 15, 2024 3:20:36 PM CST             Success
TDE Wallet directory creation            July 15, 2024 3:17:38 PM
CST             July 15, 2024 3:17:46 PM CST             Success
Setting Database parameter for OKV       July 15, 2024 3:17:46 PM
CST             July 15, 2024 3:19:07 PM CST             Success
Auto login TDE Wallet creation           July 15, 2024 3:19:07 PM
CST             July 15, 2024 3:19:09 PM CST             Success
Installing the OKV client software       July 15, 2024 3:19:09 PM
CST             July 15, 2024 3:19:42 PM CST             Success
Setting TDE Master Encryption Key        July 15, 2024 3:19:43 PM
CST             July 15, 2024 3:20:13 PM CST             Success
Create Users tablespace                  July 15, 2024 3:20:36 PM
CST             July 15, 2024 3:20:39 PM CST             Success
Clear all listeners from Database        July 15, 2024 3:20:39 PM
CST             July 15, 2024 3:20:40 PM CST             Success
Copy Pwfile to Shared Storage            July 15, 2024 3:20:43 PM
CST             July 15, 2024 3:20:47 PM CST             Success
```

An example to create an Oracle RAC database named `okvdbrac` without using the Oracle Key Vault server configuration object, is as follows. Specify the path to the `okvclient.jar` file for the local database instance first, followed by the path to the `okvclient.jar` file for the remote database instance in the request.

```
odacli create-database -n okvdb -y RAC -t -kt OKV -ocp /tmp/dir1/
okvclient.jar,/tmp/dir2/okvclient.jar
Enter SYS and SYSTEM user password:
Retype SYS and SYSTEM user password:
Enter TDE wallet password:
Retype TDE wallet password:

Job details
----------------------------------------------------------------
                     ID:  cb611ec2-ff87-44f9-a79b-3ab1a290644f
            Description:  Database service creation with DB name: okvdbrac
                 Status:  Created
                Created:  October 20, 2024 6:13:28 AM PDT
                Message:

Task Name                                Start
Time                                 End
Time                                   Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
```

```
odacli describe-job -i cb611ec2-ff87-44f9-a79b-3ab1a290644f

Job details
----------------------------------------------------------------
                     ID:  cb611ec2-ff87-44f9-a79b-3ab1a290644f
            Description:  Database service creation with DB name: okvdbrac
                 Status:  Success
                Created:  October 20, 2024 6:13:28 AM PDT
                Message:

Task Name                                Node Name                 Start
Time                                 End
Time                                   Status
---------------------------------------- ------------------------
----------------------------------------
---------------------------------------- ----------------
Validating dbHome available space        n1                        October
20, 2024 6:13:49 AM PDT          October 20, 2024 6:13:49 AM PDT
Success
Validating dbHome available space        n2                        October
20, 2024 6:13:49 AM PDT          October 20, 2024 6:13:49 AM PDT
Success
Setting up SSH equivalence               n1                        October
20, 2024 6:13:51 AM PDT          October 20, 2024 6:13:54 AM PDT
Success
Setting up SSH equivalence               n1                        October
20, 2024 6:13:54 AM PDT          October 20, 2024 6:13:57 AM PDT
Success
Creating ACFS database home              n1                        October
20, 2024 6:13:58 AM PDT          October 20, 2024 6:13:58 AM PDT
Success
Validating dbHome available space        n1                        October
20, 2024 6:13:59 AM PDT          October 20, 2024 6:13:59 AM PDT
Success
.
.
.
Setting TDE Master Encryption Key        n1                       October
20, 2024 6:34:15 AM PDT          October 20, 2024 6:34:56 AM PDT
Success
Create Users tablespace                  n1                       October
20, 2024 6:34:57 AM PDT          October 20, 2024 6:34:59 AM PDT
Success
Clear all listeners from Database        n1                       October
20, 2024 6:35:00 AM PDT          October 20, 2024 6:35:01 AM PDT
Success
Deleting OKV Client Software Jar         n1                       October
20, 2024 6:35:03 AM PDT          October 20, 2024 6:35:04 AM PDT
Success
Deleting OKV Client Software Jar         n2                       October
20, 2024 6:35:04 AM PDT          October 20, 2024 6:35:04 AM PDT
Success
Encrypt all Tablespaces                  n1                       October
```

```
20, 2024 6:35:04 AM PDT          October 20, 2024 6:36:15 AM PDT
Success
```

Related Topics

- Implementing Multi-User Access on Oracle Database Appliance
  Understand how multi-user access can be implemented on Oracle Database Appliance, its advantages and restrictions and the associated life cycle management changes for your appliance.
- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Managing Oracle Key Vault Server Configuration Operations Using ODACLI Commands

Understand how you can manage Oracle Key Vault server configuration operations on Oracle Database Appliance on bare metal and DB systems.

You can create, list, delete, and describe Oracle Key Vault server configuration on Oracle Database Appliance bare metal and DB systems. Note that you run these operations only when you create TDE-enabled databases that allow Oracle Database Appliance to store the user credentials of the Oracle Key Vault user, described in the *Creating TDE-Enabled Databases With User Credentials* topic.

Listing Oracle Key Vault server configurations:

Use the `odacli list-okvserverconfigs` command to view all the Oracle Key Vault server configurations.

```
odacli list-okvserverconfigs

ID                                       Name                 User
Name          IP address          Host Name           Description
---------------------------------------- --------------------
--------------- ------------------ -------------------- ------------------
d017d35a-0757-460a-98f0-9f8d0944c872     okvobj1
epadmin           xxx.xx.xx.xxx       Null                 OKV server for
Sales database
b6336173-666b-4d81-88bd-5ae384419114     okvobj2
epadmin2          xxx.xx.xx.xxx       Null                 OKV server for
Sales database
```

Describing an Oracle Key Vault server configuration:

Use the `odacli describe-okvserverconfig` command to view a Oracle Key Vault server configuration.

For example:

```
odacli describe-okvserverconfig -n dokvobj1
OKV Server Config details
----------------------------------------------------------------
                     ID: c196bd0a-4a42-43a6-8578-9929ceb28525
                   Name: dokvobj1
               UserName: user1
```

```
              IP Address: xxx.xxx.xx.xxx
               Host Name:
             Description: OKV server config for Sales
             CreatedTime: October 4, 2024 9:41:43 AM HKT
             UpdatedTime: October 4, 2024 9:42:10 AM HKT
                   State: CONFIGURED
```

Deleting Oracle Key Vault server configuration:

Use the `odacli delete-okvserverconfig` command to delete Oracle Key Vault server configuration. You can delete the Oracle Key Vault server configuration only if all databases that used the Oracle Key Vault server configuration are already deleted.

```
odacli delete-okvserverconfig -n okvobj1
Job details
----------------------------------------------------------------
                     ID:  7446c9b4-a927-431f-8e0b-099eec8116e7
            Description:  Delete OKV Server Config: dokvobj1
                 Status:  Running
                Created:  October 4, 2024 9:44:29 AM HKT
                Message:  Delete OKV Server Config

Task Name                                Start
Time                               End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
OKV Password Store deletion              October 4, 2024 9:44:29 AM
HKT                                                      Running

 odacli describe-job -i 7446c9b4-a927-431f-8e0b-099eec8116e7

Job details
----------------------------------------------------------------
                     ID:  7446c9b4-a927-431f-8e0b-099eec8116e7
            Description:  Delete OKV Server Config: dokvobj1
                 Status:  Success
                Created:  October 4, 2024 9:44:29 AM HKT
                Message:  Delete OKV Server Config

Task Name                                Node Name                 Start
Time                               End Time
Status
---------------------------------------- ------------------------
----------------------------------------
---------------------------------------- ----------------
OKV Password Store deletion              oda1                      October 4,
2024 9:44:29 AM HKT          October 4, 2024 9:44:45 AM HKT
Success
OKV Server Config directory deletion     oda1                      October 4,
2024 9:44:45 AM HKT          October 4, 2024 9:44:45 AM HKT
Success
OKV Server Config deletion               oda1                      October 4,
2024 9:44:45 AM HKT          October 4, 2024 9:44:46 AM HKT           Success
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Managing Database Operations on TDE-enabled Databases with TDE keys stored on Oracle Key Vault

Understand how you can manage TDE-enabled databases with TDE keys stored on Oracle Key Vault on Oracle Database Appliance on bare metal and DB systems.

You can re-key TDE master encryption key or restore, delete, perform database backup and recovery, and configure Oracle Data Guard on TDE-enabled database on Oracle Key Vault on Oracle Database Appliance bare metal and DB systems.

- Re-keying TDE Master Encryption Key for TDE-Enabled Databases
  Understand how you can re-key TDE master encryption key of TDE-enabled database on Oracle Database Appliance on bare metal and DB systems, that uses Oracle Key Vault to store TDE keys.
- Performing Backup and Recovery of TDE-enabled Databases
  Understand how you can back up and recover TDE-enabled database with TDE keys stored on Oracle Key Vault on Oracle Database Appliance on bare metal and DB systems.
- Restoring TDE-Enabled Databases Using Oracle Key Vault User Credentials
  Understand how you can restore TDE-enabled databases using database backups, on Oracle Database Appliance bare metal and DB systems, with credentials of a user on Oracle Key Vault.
- Restoring TDE-Enabled Databases With Endpoints and Wallet Manually Created in Oracle Key Vault
  Understand how you can restore TDE-enabled databases on Oracle Database Appliance using the endpoints and wallets created manually on Oracle Key Vault. This method supports creation of TDE-enabled databases on bare metal and DB systems.
- Deleting TDE-enabled Databases
  Understand how you can delete a TDE-enabled database on Oracle Database Appliance on bare metal and DB systems, that uses Oracle Key Vault to store TDE keys.

Re-keying TDE Master Encryption Key for TDE-Enabled Databases

Understand how you can re-key TDE master encryption key of TDE-enabled database on Oracle Database Appliance on bare metal and DB systems, that uses Oracle Key Vault to store TDE keys.

Use the `odacli modify-database` command to perform re-key of TDE master encryption key with the option `-rkt`. For TDE configured using Oracle Key Vault, you are prompted to specify the password during the re-key operation. For example:

```
odacli modify-database -n kokvdb2 -rkt
{
  "jobId" : "dbedf89d-b60f-4209-bbcf-0db61e4eb350",
  "status" : "Created",
  "message" : "Modify database",
  "reports" : [ ],
  "createTimestamp" : "July 15, 2024 15:56:01 PM CST",
  "resourceList" : [ {
    "resourceId" : "7a1c996b-ee8d-49bc-8759-cbbfcee3bcce",
```

```
    "resourceType" : null,
    "resourceNewType" : "Db",
    "jobId" : "dbedf89d-b60f-4209-bbcf-0db61e4eb350",
    "updatedTime" : null
  } ],
  "description" : "Modify database: kokvdb2",
  "updatedTime" : "July 15, 2024 15:56:01 PM CST",
  "jobType" : null
}

 odacli describe-job -i dbedf89d-b60f-4209-bbcf-0db61e4eb350

Job details
----------------------------------------------------------------
                     ID:  dbedf89d-b60f-4209-bbcf-0db61e4eb350
            Description:  Modify database: kokvdb2
                 Status:  Success
                Created:  July 15, 2024 3:56:01 PM CST
                Message:  Modify database

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
TDE Wallet Re-Key                        July 15, 2024 3:56:04 PM
CST             July 15, 2024 3:56:12 PM CST             Success
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Performing Backup and Recovery of TDE-enabled Databases

Understand how you can back up and recover TDE-enabled database with TDE keys stored on Oracle Key Vault on Oracle Database Appliance on bare metal and DB systems.

Performing backup of TDE-enabled database with TDE keys stored on Oracle Key Vault

Use the `odacli create-backup` command to back up TDE-enabled database with TDE keys stored on Oracle Key Vault. The TDE wallet is not present on Oracle Database Appliance, but resides in Oracle Key Vault server, and hence it is not backed up with the database backup. As a result, the value of TDE wallet backup location attribute of the database backup report is set to null.

```
[odaadmin@oda1 bin]$ odacli describe-job -i be94d1f7-ad6e-48a5-94a3-
f0a52578048b

Job details
----------------------------------------------------------------
                     ID:  be94d1f7-ad6e-48a5-94a3-f0a52578048b
            Description:  Create Regular-L0 Backup[TAG:auto][Db:db15][NFS:/
nfs_backup/orabackups/dbsfb225975e/database/1851419729/db15]
                 Status:  Success
```

```
                Created:  March 26, 2024 4:14:19 AM PDT
                Message:

Task Name                                Node Name                Start
Time                                End Time                    Status
---------------------------------------- ------------
----------------------------------------
--------------------------------         ------
Validate backup config                   oda1          March 26, 2024 4:14:21
AM PDT           March 26, 2024 4:14:21 AM PDT           Success
NFS location existence validation        oda1          March 26, 2024 4:14:21
AM PDT           March 26, 2024 4:14:22 AM PDT           Success
Recovery Window validation               oda1          March 26, 2024 4:14:22
AM PDT           March 26, 2024 4:14:24 AM PDT           Success
Archivelog deletion policy configuration oda1          March 26, 2024 4:14:24
AM PDT           March 26, 2024 4:14:26 AM PDT           Success
Database backup                          oda1          March 26, 2024 4:14:26
AM PDT           March 26, 2024 4:15:21 AM PDT           Success


[odaadmin@oda1 bin]$ odacli describe-backupreport -i d2326683-9766-4933-bee1-
a2b59c9d7e9e
{
  "id" : "d2326683-9766-4933-bee1-a2b59c9d7e9e",
  "dbResId" : "96a98156-b657-403e-a87c-357c2069c285",
  "tag" : "auto",
  "dbId" : "1851419729",
  "dbName" : "db15",
  "dbUniqueName" : "db15",
  "backupType" : "Regular-L0",
  "keepDays" : null,
  "backupLocation" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/db",
  "cfBackupHandle" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/db/c-1851419729-20240326-03",
  "spfBackupHandle" : "/nfs_backup/orabackups/dbsfb225975e/database/
1851419729/db15/db/c-1851419729-20240326-03",
  "pitrTimeStamp" : "March 26, 2024 04:15:06 AM PDT",
  "pitrSCN" : "1675444",
  "resetLogsTimeStamp" : "March 26, 2024 03:01:39 AM PDT",
  "resetLogsSCN" : "1575481",
  "oraHomeVersion" : "19.23.0.0.240116",
  "sqlPatches" : "35926646,34774667,35943157",
  "backupLogLoc" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/rmanlog/2024-03-26/rman_backup_auto_2024-03-26_04-14-29.0478.log",
  "tdeWalletLoc" : null,
  "dbConfigLoc" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/dbconfig/2024-03-26/DBCONFIG_auto_2024-03-26_04-15-19.0472.tar.gz",
  "name" : "Backup_Report_db15",
  "createTime" : "March 26, 2024 04:14:26 AM PDT",
  "state" : {
    "status" : "CONFIGURED"
  },
  "updatedTime" : "March 26, 2024 04:14:26 AM PDT",
  "backupReportLogDetail" : "/nfs_backup/orabackups/dbsfb225975e/database/
1851419729/db15/rmandetaillogreport/2024-03-26/
```

```
rman_list_backup_detail_auto_2024-03-26_04-15-16.0129.log",
  "dbInfo" : {
    "dbClass" : "OLTP",
    "dbType" : "RAC",
    "dbShape" : "odb2",
    "dbEdition" : "EE",
    "dbStorage" : "ASM",
    "dbRedundancy" : null,
    "pdbName" : null,
    "isCdb" : false
  },
  "dbDataSize" : "2441M",
  "dbRedoSize" : "16403M",
  "rmanBackupPieces" : "/nfs_backup/orabackups/dbsfb225975e/database/
1851419729/db15/backuppieces/2024-03-26/backupPieces_auto_96a98156-b657-403e-
a87c-357c2069c285_20240326041518.json",
  "compressionAlgo" : "BASIC",
  "cpuPool" : null,
  "numberOfCores" : null,
  "keystoreType" : "OKV",
}
```

Note that since TDE wallet backup is not supported, if the keystore type is `OKV`, if you specify the component as `tdewallet` in the `odacli create-backup` command, an error is encountered.

```
[odaadmin@oda1 bin]$ odacli create-backup -n db15 -c tdewallet
DCS-10144:TDE wallet backup operation is not supported for TDE databases with
OKV keystore type.
```

Performing recovery of database with TDE configured using Oracle Key Vault with credentials of Oracle Key Vault user

Use the `odacli recover-database` command to recover a database with TDE configured using Oracle Key Vault. For example, to recover a database `kokvdb2`, run the following command:

```
odacli recover-database -n kokvdb2 -t latest
{
  "jobId" : "8a50d65f-2269-41ff-b030-51fe4ecc82b5",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "July 15, 2024 18:26:26 PM CST",
  "resourceList" : [ ],
  "description" : "Create recovery-latest for DB : kokvdb2",
  "updatedTime" : "July 15, 2024 18:26:26 PM CST",
  "jobType" : null
}

odacli describe-job -i 8a50d65f-2269-41ff-b030-51fe4ecc82b5

Job details
----------------------------------------------------------------
                     ID:  8a50d65f-2269-41ff-b030-51fe4ecc82b5
            Description:  Create recovery-latest for DB : kokvdb2
```

```
                 Status:  Success
                Created:  July 15, 2024 6:26:26 PM CST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Database recovery validation             July 15, 2024 6:26:30 PM
CST             July 15, 2024 6:27:50 PM CST             Success
Database recovery                        July 15, 2024 6:27:50 PM
CST             July 15, 2024 6:30:07 PM CST             Success
Enable block change tracking             July 15, 2024 6:30:07 PM
CST             July 15, 2024 6:30:12 PM CST             Success
Database opening                         July 15, 2024 6:30:12 PM
CST             July 15, 2024 6:30:15 PM CST             Success
Database restart                         July 15, 2024 6:30:15 PM
CST             July 15, 2024 6:31:31 PM CST             Success
Recovery metadata persistence            July 15, 2024 6:31:31 PM
CST             July 15, 2024 6:31:31 PM CST             Success
```

The TDE wallets are stored on the Oracle Key Vault server and not on Oracle Database Appliance, and so, ODACLI commands cannot be used for restore and recovery operations. If the keystore type is `OKV`, and you specify the component as `tdewallet` in the `odacli recover-database` command, an error is encountered.

```
[odaadmin@oda1 bin]$ odacli recover-database -n db15 -tl /nfs_backup/
ewallet.p12 -t latest
DCS-10045:Validation error encountered: Usage of TDE wallet related
parameters is not allowed for a TDE database with OKV keystore type.

[odaadmin@oda1 bin]$ odacli restore-tdewallet -n db15
Enter TDE wallet password:
DCS-10144:Restore TDE wallet operation is not supported for TDE databases
with OKV keystore type.
```

Performing recovery of database with TDE configured using Oracle Key Vault with endpoints in Oracle Key Vault

The `odacli recover-database` command prompts for the TDE password. Before database recovery, the local auto-login wallet is regenerated using the Administer Key Management command. You must specify the TDE password for database recovery.

```
odacli recover-database -n kokvdb2 -t latest
Enter the TDE password:


odacli describe-job -i 8a50d65f-2269-41ff-b030-51fe4ecc82b5

Job details
----------------------------------------------------------------
                     ID:  8a50d65f-2269-41ff-b030-51fe4ecc82b5
            Description:  Create recovery-latest for DB : kokvdb2
```

```
                 Status:  Success
                Created:  July 15, 2024 6:26:26 PM CST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Auto login TDE Wallet creation           July 15, 2024 6:26:30 PM
CST             July 15, 2024 6:27:50 PM CST
Database recovery validation             July 15, 2024 6:26:30 PM
CST             July 15, 2024 6:27:50 PM CST             Success
Database recovery                        July 15, 2024 6:27:50 PM
CST             July 15, 2024 6:30:07 PM CST             Success
Enable block change tracking             July 15, 2024 6:30:07 PM
CST             July 15, 2024 6:30:12 PM CST             Success
Database opening                         July 15, 2024 6:30:12 PM
CST             July 15, 2024 6:30:15 PM CST             Success
Database restart                         July 15, 2024 6:30:15 PM
CST             July 15, 2024 6:31:31 PM CST             Success
Recovery metadata persistance            July 15, 2024 6:31:31 PM
CST             July 15, 2024 6:31:31 PM CST             Success
```

Errors during database recovery

The TDE wallets are stored on the Oracle Key Vault server and not on Oracle Database Appliance, and so, ODACLI commands cannot be used for restore and recovery operations. If the keystore type is `OKV`, and you specify the component as `tdewallet` in the `odacli recover-database` command, an error is encountered.

```
[odaadmin@oda1 bin]$ odacli recover-database -n db15 -tl /nfs_backup/
ewallet.p12 -t latest
DCS-10045:Validation error encountered: Usage of TDE wallet related
parameters is not allowed for a TDE database with OKV keystore type.

[odaadmin@oda1 bin]$ odacli restore-tdewallet -n db15
Enter TDE wallet password:
DCS-10144:Restore TDE wallet operation is not supported for TDE databases
with OKV keystore type.
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Restoring TDE-Enabled Databases Using Oracle Key Vault User Credentials

Understand how you can restore TDE-enabled databases using database backups, on Oracle Database Appliance bare metal and DB systems, with credentials of a user on Oracle Key Vault.

Prerequisites

- To restore a database with primary role in an Oracle Data Guard configuration, TDE wallet for the target database must be created on the Oracle Key Vault server before you run the restore command. This TDE wallet is a copy of the wallet of the source database.
- To restore a database with a standby role in an Oracle Data Guard configuration, the TDE wallet copy must **not** be created. The TDE wallet of the primary database must be used for the standby database.

Step 1: Create the Oracle Key Vault user. This step applies to multi-user access-enabled systems only.

If the user with the `ODA-OKVCONFIGADMIN` role does not exist, or if you want to assign a different user for the database you are restoring, then create the user.

1. Create a new user with the `ODA-OKVCONFIGADMIN` role. For example:

   ```
   odacli create-user -u okvuser1 -r ODA-OKVCONFIGADMIN
   ```

   The user is created and assigned a temporary password.
2. After the user is created successfully, the `okvuser1` can log into the appliance with the temporary password.
3. The `okvuser1` is in the `Inactive` state. If the appliance is configured with multi-user access enabled, then activate the user with the following command. Note that if the appliance is configured with passwordless multi-user access, then you do not need to activate the user. For more information about configuring multi-user access, the topic *Implementing Multi-User Access on Oracle Database Appliance*.

   ```
   odacli activate-user
   ```

   You are prompted to change the password. Enter the temporary password, the new password, and confirm the new password.
4. Use the new password to connect by SSH into the appliance and run ODACLI commands or connect to the Browser User Interface.

Step 2: Create an Oracle Key Vault server configuration

Create an Oracle Key Vault server configuration for the target database. Create a new Oracle Key Vault server configuration only if an Oracle Key Vault server configuration for the same Oracle Key Vault server does not already exist. If a configuration already exists, then use the existing configuration.

1. The Oracle Key Vault user that creates the Oracle Key Vault server configuration must be a user with atleast the privilege of `Create Endpoint` on the Oracle Key Vault server.

2. For multi-user access-enabled systems, log in as the user with the `ODA-OKVCONFIGADMIN` role, for example, the `okvuser1`. For non-multi user access enabled systems, log in as the `root` user.
3. Oracle Key Vault password prompt displays twice, to accept and confirm the Oracle Key Vault password. This is the password of the Oracle Key Vault user on the Oracle Key Vault server, which enables the user to log into the Oracle Key Vault server. Specify the user name with the `-u` option in the `odacli create-okvserverconfig` command. The name of the Oracle Key Vault server configuration is case-sensitive and no two Oracle Key Vault server configurations can have same name on Oracle Database Appliance. The length of the Oracle Key Vault server configuration cannot exceed 128 characters.

```
odacli create-okvserverconfig -n okvobj1 -u epokvusr1 -ip xxx.xx.xxx.xxx
Enter OKV user password :
Retype OKV user password :
Job details
----------------------------------------------------------------
                     ID: 8878fc02-3d64-4519-bf52-fcb50839f89f
            Description: Create OKV Server Config: okvcfg5
                 Status: Success
                Created: October 14, 2024 14:04:29 UTC
                Message: Create OKV Server Config.

Task Name                                Node Name                 Start
Time                                 End
Time                                   Status
---------------------------------------- ------------------------
----------------------------------------
---------------------------------------- ----------------
OKV Password Store creation                           oda1          October
14, 2024 14:04:29 UTC           October 14, 2024 14:04:56 UTC
Success
Download of OKV REST package                          oda1          October
14, 2024 14:04:31 UTC           October 14, 2024 14:04:32 UTC
Success
Edit of OKV REST package                              oda1          October
14, 2024 14:04:32 UTC           October 14, 2024 14:04:32 UTC
Success
Generate OKV client wallets                           oda1          October
14, 2024 14:04:32 UTC           October 14, 2024 14:04:35 UTC
Success
```

The command creates an Oracle Key Vault server configuration `okvobj1` for the Oracle Key Vault server user `epokvusr1` on the specified server.

Step 3: Copy TDE keys from source wallet to target wallet on Oracle Key Vault server using ODACLI commands

Use the `odacli copy-okvtdewallet` command to copy the TDE wallet to the Oracle Key Vault server. Specify the source wallet name (`--source-wallet-name | -swn`), target wallet Oracle Key Vault server configuration object name (`--target-okvserverconfig | -tosc`), and the target wallet name (`--target-wallet-name | -twn`) in the command. A new wallet with the name specified for the target wallet name (`--target-wallet-name | -twn`) parameter is created in the target Oracle Key Vault server. This wallet contains the TDE keys copied from the source wallet.

If the source wallet is on a different Oracle Key Vault server than the target wallet, then specify the source wallet Oracle Key Vault server details such as the host name (`--source-hostname | -shn`) or the IP address (`--source-ip | -sip`) of the server and the name of the user that owns the wallet on the Oracle Key Vault server (`--source-username | -su`). You must provide a password when prompted in this case. This is the password of the Oracle Key Vault user that owns the source wallet on the Oracle Key Vault server. For multi-user access enabled systems, run this command as the Oracle Key Vault user on Oracle Database Appliance.

There are three scenarios when you copy the TDE wallet:

Scenario 1: Copy the TDE wallet on the same Oracle Key Vault server with the same Oracle Key Vault user

In this case, the source TDE wallet and the target TDE wallet are on the same Oracle Key Vault server, and both are owned by the same user on the OKV server. For example:

```
odacli copy-okvtdewallet -swn tdeokv_on_oda8m013-c -tosc okvobj1 -twn tdeokvss
Job details
----------------------------------------------------------------
                     ID:  50df7dcc-8e17-48db-9206-879c44e57c20
            Description:  Copy wallet in OKV server
                 Status:  Success
                Created:  June 20, 2024 2:56:02 AM CST
                Message:

Task Name                                Start
Time                                 End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Creating a TDE wallet copy in the OKV    June 20, 2024 2:56:03 AM
CST             June 20, 2024 2:57:17 AM CST             Success
server
```

Scenario 2: Copy the TDE wallet on the same Oracle Key Vault server but using a different Oracle Key Vault user

In this case, the source and the target wallet are on the same Oracle Key Vault server but are owned by different users on the Oracle Key Vault server. The owner of the source wallet in the Oracle Key Vault must provide read and manage access permissions on the wallet to the owner of the target wallet on Oracle Key Vault before running the ODACLI command to copy the wallet.

Follow these steps to provide read and manage wallet permissions to the target wallet Oracle Key Vault user:

1. Log into the Oracle Key Vault server as the source wallet Oracle Key Vault user.
2. Navigate to the **Keys and Wallets** tab. Select the source wallet checkbox and click the Edit icon.
3. Navigate to the **Wallet Access Settings** section and click **Add**.
4. In the dropdown list for **Type**, select **Users**, and then select the checkbox for the target wallet Oracle Key Vault user.
5. In the **Access Level** section, select **Read Only** and **Manage Wallet**.
6. Click **Save**.

After you provide the required permissions to the target wallet user, run the `odacli copy-okvtdewallet` command.

```
odacli copy-okvtdewallet -swn tdeokv_on_oda8m013-c -tosc okvobj2 -twn tdeokvdu
Job details
----------------------------------------------------------------
                     ID:  50df7dcc-8e17-48db-9206-879c44e57c20
            Description:  Copy wallet in OKV server
                 Status:  Success
                Created:  June 20, 2024 2:56:02 AM CST
                Message:

Task Name                                Start
Time                                End Time
Status
---------------------------------------
---------------------------------------
---------------------------------------- ----------------
Creating a TDE wallet copy in the OKV    June 20, 2024 2:56:03 AM
CST             June 20, 2024 2:57:17 AM CST             Success
server
```

Scenario 3: Copy the TDE wallet to a different Oracle Key Vault server

In this case, the source wallet and the target wallet are on different Oracle Key Vault servers. Specify the source wallet Oracle Key Vault server details in the `odacli copy-okvtdewallet` command. Specify the password of the source wallet Oracle Key Vault user when prompted. For example:

```
odacli copy-okvtdewallet -swn tdeokv_on_oda8m013-c -sip xxx.xx.x.xxx -su
epokvusr1 -tosc okvobj3 -twn tdeokvds
Enter OKV user password:
Retype OKV user password:

Job details
----------------------------------------------------------------
                     ID:  50df7dcc-8e17-48db-9206-879c44e57c20
            Description:  Copy wallet in OKV server
                 Status:  Success
                Created:  June 20, 2024 2:56:02 AM CST
                Message:

Task Name                                Start
Time                                End Time
Status
---------------------------------------
---------------------------------------
---------------------------------------- ----------------
Creating a TDE wallet copy in the OKV    June 20, 2024 2:56:03 AM
CST             June 20, 2024 2:57:17 AM CST             Success
server
```

Alternate method for Step 3: Copying the TDE keys from the source database wallet to the target database wallet using the Oracle Key Vault user interface

Instead of using ODACLI commands, you can create a new wallet for the target database in the Oracle Key Vault server and copy the TDE keys of the source database to the wallet using the Oracle Key Vault user interface as follows:

1. Log into the Oracle Key Vault server.
2. Navigate to the **Keys & Wallets** tab.
3. The Oracle Key Vault user must be provided read, write, and manage wallet permissions from the **Access Settings** section of the source database wallet.
4. To create a new wallet, click **Create**.
5. Specify the new name for the wallet of the target database.
6. Copy the TDE keys of the source database when creating the new wallet. Search for the name of the source database in the **Add Wallet Contents** section. When the keys are displayed, select all **TDE Master Encryption Key** checkboxes for the database.
7. To save the wallet, click **Save**.
8. Run this step only if the Oracle Key Vault administrator has created the wallet for the Oracle Key Vault user. Ensure that the administrator provides read-write and manage-wallet access for the wallet to the user after creation.
    a. Log into the Oracle Key Vault server as the Oracle Key Vault administrator.
    b. In the **Keys & Wallets** tab, select the checkbox for the newly-created wallet and click on the **Edit** option for the wallet.
    c. In the **Wallet Access Settings**, click **Add**.
    d. In the **Select Endpoint/User Group** section, select **Users** from the dropdown list.
    e. Select the checkbox for the Oracle Key Vault user to be provided access to the wallet.
    f. In the **Access Level** section, select **Read and Modify** and **Manage Wallet** options.
    g. To save the access settings for the wallet, click **Save**.

Step 4: Restoring TDE-enabled databases with TDE keys stored on Oracle Key Vault

Use the `odacli irestore-database` command to restore a TDE-enabled database with TDE keys stored on Oracle Key Vault. Specify the options `--enable-tde`, `--okv-server-config`, `--okv-wallet-name`, and `keystore-type` in the command.

The option `--enable-tde,-t` indicates that the restored database must be configured with TDE, using the Oracle Key Vault server configuration specified in the `--okv-server-config` option. The keystore of such irestored database is set to `OKV`. The `okvServerConfigName` attribute of the restored database object is set to the name of the Oracle Key Vault server configuration specified in the irestore request. Specify the wallet name for the target database with the `--okv-wallet-name` parameter. This wallet must be present in the Oracle Key Vault server and the TDE keys must be copied to the wallet before running the irestore operation.

Since the wallet is already created in the Oracle Key Vault server, specify the name of the wallet, the Oracle Key Vault server configuration where the wallet is created, and the keystore type as `OKV` in this case in the `odacli irestore-database` command:

```
odacli irestore-database -r /tmp/db1Backupreport.json -t -osc okvobj1 -kt OKV
-own db2_wallet -n okvdb15
```

```
Enter SYS and SYSTEM user password:
Enter SYS and SYSTEM user password:

Job details
----------------------------------------------------------------
                     ID:  dc44f3f0-5af3-4402-ad23-e923e1446371
            Description:  Database service recovery with DB name: okvdb15
                 Status:  Success
                Created:  July 5, 2024 6:54:00 AM EDT
                Message:

Task Name                                Node Name                 Start
Time                                 End Time
Status
---------------------------------------- -------------------------
----------------------------------------
---------------------------------------- ----------------
Check if cluster ware is running          oda1                     July 5,
2024 6:54:23 AM EDT              July 5, 2024 6:54:23 AM EDT
Success
Check if cluster ware is running          oda2                    July 5, 2024
6:54:23 AM EDT              July 5, 2024 6:54:24 AM EDT
Success
Creating DbStorage for DbRestore          oda1                    July 5, 2024
6:54:25 AM EDT              July 5, 2024 6:54:42 AM EDT
Success
Validating DiskSpace for DATA             oda1                     July 5,
2024 6:54:25 AM EDT              July 5, 2024 6:54:27 AM EDT
Success
Setting up SSH equivalence                oda1                     July 5,
2024 6:54:27 AM EDT              July 5, 2024 6:54:34 AM EDT
Success
Configuring user access to ACFS           oda1                     July 5,
2024 6:54:38 AM EDT              July 5, 2024 6:54:39 AM EDT
Success
filesystems for
okvdb15

Audit directory creation                  oda1                    July 5, 2024
6:54:40 AM EDT              July 5, 2024 6:54:40 AM EDT
Success
Audit directory creation                  oda2                    July 5, 2024
6:54:41 AM EDT              July 5, 2024 6:54:41 AM EDT
Success
Configure TDE for RestoreDb TaskFlow      oda1                    July 5, 2024
6:54:46 AM EDT              July 5, 2024 6:56:07 AM EDT
Success
Auxiliary Instance Creation               oda1                     July 5, 2024
6:54:50 AM EDT              July 5, 2024 6:55:30 AM EDT
Success
TDE Wallet directory creation             oda1                     July 5, 2024
6:55:30 AM EDT              July 5, 2024 6:55:31 AM EDT
Success
OKV Endpoint configuration                oda1                     July 5, 2024
6:55:31 AM EDT              July 5, 2024 6:56:02 AM EDT
Success
```

```
Auto login TDE Wallet creation           oda1                   July 5, 2024
6:56:02 AM EDT             July 5, 2024 6:56:04 AM EDT
Success
Password based TDE Wallet open           oda1                    July 5,
2024 6:56:04 AM EDT             July 5, 2024 6:56:07 AM EDT
Success
Create pfile for Auxiliary Instance      oda1                    July 5,
2024 6:56:07 AM EDT             July 5, 2024 6:56:08 AM EDT
Success
Rman duplicate                           oda1                    July 5,
2024 6:56:08 AM EDT             July 5, 2024 7:01:22 AM EDT
Success
Creating pfile from spfile               oda1                    July 5,
2024 7:01:23 AM EDT             July 5, 2024 7:01:24 AM EDT
Success
Set PFile Ownership                      oda1                    July 5,
2024 7:01:24 AM EDT             July 5, 2024 7:01:24 AM EDT
Success
Customize Db Parameters                  oda1                    July 5,
2024 7:01:25 AM EDT             July 5, 2024 7:01:36 AM EDT
Success
Shutdown And Start database              oda1                    July 5,
2024 7:01:36 AM EDT             July 5, 2024 7:06:51 AM EDT
Success
Create spfile for restore db             oda1                    July 5,
2024 7:06:51 AM EDT             July 5, 2024 7:06:53 AM EDT
Success
Set PFile Ownership                      oda1                    July 5,
2024 7:06:53 AM EDT             July 5, 2024 7:06:53 AM EDT
Success
Shutdown And Mount database              oda1                    July 5,
2024 7:06:54 AM EDT             July 5, 2024 7:08:22 AM EDT
Success
Re-Create control file                   oda1                    July 5,
2024 7:08:23 AM EDT             July 5, 2024 7:09:21 AM EDT
Success
Removing Disabled Redo Threads           oda1                    July 5,
2024 7:09:21 AM EDT             July 5, 2024 7:09:23 AM EDT
Success
Updating DB attributes                   oda1                    July 5,
2024 7:09:23 AM EDT             July 5, 2024 7:09:25 AM EDT
Success
Register Database taskflow               oda1                    July 5,
2024 7:09:30 AM EDT             July 5, 2024 7:14:51 AM EDT
Success
Create SPFile in shared loc              oda1                    July 5,
2024 7:09:30 AM EDT             July 5, 2024 7:09:40 AM EDT
Success
Delete Local Spfile                      oda1                    July 5,
2024 7:09:40 AM EDT             July 5, 2024 7:09:41 AM EDT
Success
Register DB with clusterware             oda1                    July 5,
2024 7:09:41 AM EDT             July 5, 2024 7:11:19 AM EDT
Success
Add Startup Trigger to Open all PDBS     oda1                    July 5,
2024 7:11:20 AM EDT             July 5, 2024 7:11:20 AM EDT
```

```
Success
Set SysPassword and Create PwFile         oda1                    July 5,
2024 7:11:21 AM EDT             July 5, 2024 7:11:24 AM EDT
Success
Enable block change tracking              oda1                    July 5,
2024 7:11:25 AM EDT             July 5, 2024 7:11:37 AM EDT
Success
Creating pfile                            oda1                    July 5,
2024 7:11:37 AM EDT             July 5, 2024 7:11:39 AM EDT
Success
Updating db env                           oda1                    July 5,
2024 7:11:39 AM EDT             July 5, 2024 7:11:40 AM EDT
Success
Enable DbSizing Template                  oda1                    July 5,
2024 7:11:40 AM EDT             July 5, 2024 7:13:14 AM EDT
Success
Update Database Global Name               oda1                    July 5,
2024 7:13:14 AM EDT             July 5, 2024 7:13:17 AM EDT
Success
Create tns entry                          oda1                    July 5,
2024 7:13:18 AM EDT             July 5, 2024 7:13:19 AM EDT
Success
Create tns entry                         oda2                     July 5,
2024 7:13:19 AM EDT             July 5, 2024 7:13:21 AM EDT
Success
Running datapatch                         oda1                    July 5,
2024 7:13:21 AM EDT             July 5, 2024 7:13:52 AM EDT
Success
Set CPU pool                              oda1                    July 5,
2024 7:13:52 AM EDT             July 5, 2024 7:13:52 AM EDT
Success
Reset Associated Networks for Database    oda1                    July 5,
2024 7:14:54 AM EDT             July 5, 2024 7:15:00 AM EDT
Success
Reset Associated Networks                oda2                     July 5,
2024 7:15:00 AM EDT             July 5, 2024 7:15:01 AM EDT
Success
Set log_archive_dest for Database         oda1                    July 5,
2024 7:15:01 AM EDT             July 5, 2024 7:15:07 AM EDT
Success
Setting Database parameter for OKV        oda1                    July 5,
2024 7:15:07 AM EDT             July 5, 2024 7:16:43 AM EDT
Success
Enable New Tablespace Encryption          oda1                    July 5,
2024 7:16:45 AM EDT             July 5, 2024 7:16:47 AM EDT
Success
Copy Pwfile to Shared Storage             oda1                    July 5,
2024 7:16:47 AM EDT             July 5, 2024 7:16:58 AM EDT
Success
Configure All Candidate Nodes             oda1                    July 5,
2024 7:16:58 AM EDT             July 5, 2024 7:17:01 AM EDT
Success
```

Restore the database by copying the wallet using ODACLI commands and providing the wallet name to the command. After copying the TDE keys of the wallet of the source database with

the `odacli copy-okvtdewallet` command, specify the target Oracle Key Vault wallet name in the `odacli irestore-database` command.

```
odacli irestore-database -r /tmp/db1Backupreport.json -t -osc okvobj1 -kt OKV
-own db2_wallet
Enter SYS and SYSTEM user password:
Enter SYS and SYSTEM user password:
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Restoring TDE-Enabled Databases With Endpoints and Wallet Manually Created in Oracle Key Vault

Understand how you can restore TDE-enabled databases on Oracle Database Appliance using the endpoints and wallets created manually on Oracle Key Vault. This method supports creation of TDE-enabled databases on bare metal and DB systems.

Prerequisites

- To restore a database with primary role in an Oracle Data Guard configuration, TDE wallet for the target database must be created on the Oracle Key Vault server before you run the restore command. This TDE wallet is a copy of the wallet of the source database.
- To restore a database with a standby role in an Oracle Data Guard configuration, the TDE wallet copy must **not** be created. The TDE wallet of the primary database must be used for the standby database.
- The TDE wallet created must be set as the default wallet for the endpoints created.
- The `okvclient.jar` file corresponding to each endpoint must be downloaded and copied to the Oracle Database Appliance node on which you submit the database restore request.

Step 1: Copying the TDE keys from the source database wallet to the target database wallet using the Oracle Key Vault user interface

Create a new wallet for the target database in the Oracle Key Vault server and copy the TDE keys of the source database to the wallet as follows:

1. Log into the Oracle Key Vault server.
2. Navigate to the **Keys & Wallets** tab.
3. The Oracle Key Vault user must be provided read, write, and manage wallet permissions from the **Access Settings** section of the source database wallet.
4. To create a new wallet, click **Create**.
5. Specify the new name for the wallet of the target database.
6. Copy the TDE keys of the source database when creating the new wallet. Search for the name of the source database in the **Add Wallet Contents** section. When the keys are displayed, select all **TDE Master Encryption Key** checkboxes for the database.
7. To save the wallet, click **Save**.
8. Run this step only if the Oracle Key Vault administrator has created the wallet for the Oracle Key Vault user. Ensure that the administrator provides read-write and manage-wallet access for the wallet to the user after creation.

a. Log into the Oracle Key Vault server as the Oracle Key Vault administrator.

b. In the **Keys & Wallets** tab, select the checkbox for the newly-created wallet and click on the **Edit** option for the wallet.

c. In the **Wallet Access Settings**, click **Add**.

d. In the **Select Endpoint/User Group** section, select **Users** from the dropdown list.

e. Select the checkbox for the Oracle Key Vault user to be provided access to the wallet.

f. In the **Access Level** section, select **Read and Modify** and **Manage Wallet** options.

g. To save the access settings for the wallet, click **Save**.

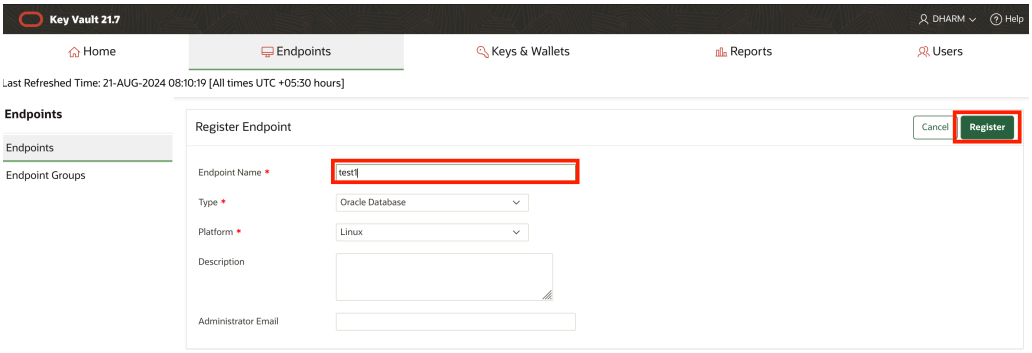
Step 2: Creating endpoints and setting the default wallet

Follow these steps:

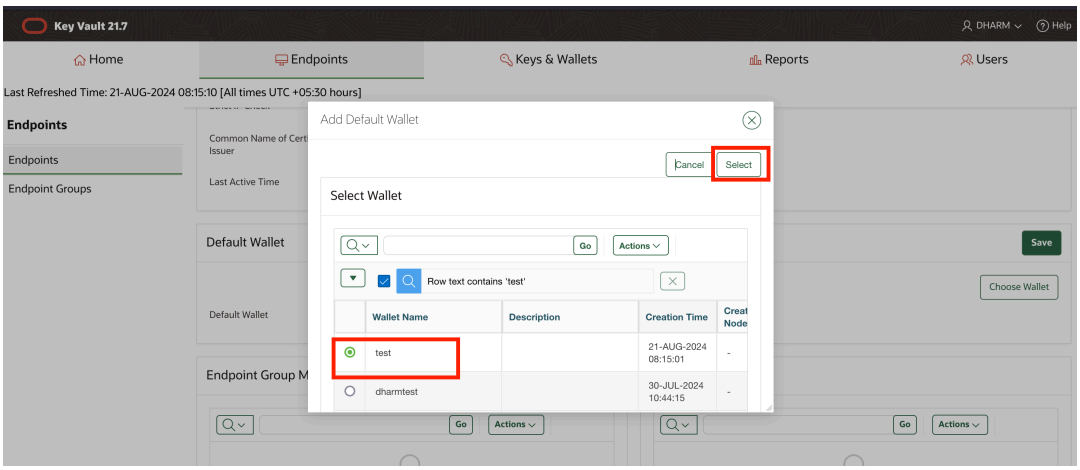
1. Create endpoints corresponding to each instance. For Oracle RAC database, create two endpoints corresponding to two instances. For single-instance database, create one endpoint corresponding to one instance.

   a. In the Oracle Key Vault home page, navigate to the **Endpoints** tab and click **Add**.

   b. Specify the **Endpoint Name**. Select the **Type** as **Oracle Database**, **Platform** as **Linux**, and click **Register**.



2. In the Default Wallet page, click **Choose Wallet** and set the default wallet for the created endpoints.



Step 3: Downloading the TDE wallet to Oracle Key Vault server using Oracle Key Vault commands

For multi user access-enabled and multi user access passwordless systems, log in as the Oracle Key Vault server user. For non-multi user access-enabled systems, log in as the `oracle` user and then follow these steps:

1. Create working directories such as `/tmp/work` and `/tmp/work/client_wallet`. Navigate to the `/tmp/work` directory. Download and extract the RESTful Service Utility from Oracle Key Vault server. You can view three directories `bin`, `lib`, and `conf`.

```
mkdir /tmp/work/
mkdir /tmp/work/client_wallet
cd /tmp/work
curl -Ok https://okv_server_ip:5695/okvrestclipackage.zip
% Total % Received % Xferd Average Speed Time Time Time Current
                                Dload Upload Total Spent Left Speed
100 3750k 100 3750k 0 0 2500k 0 0:00:01 0:00:01 --:--:-- 2500k
```

2. Extract the contents of `okvrestclipackage.zip` file. Edit the `conf/okvrestcli.ini` file and add the Oracle Key Vault server IP address, user name, and path to the directory where the client auto login wallets are created. In the following example, the client auto login wallets are created in the `/tmp/work/client_wallet` directory.

```
#Provide absolute path for log_property, okv_client_config properties
        [Default]
        log_property=./conf/okvrestcli_logging.properties
        server=<okv_server_ip>
        okv_client_config=./conf/okvclient.ora
        user=<okv_user>
        client_wallet=/tmp/work/client_wallet
```

3. Edit the `bin/okv` file and uncomment the line `export OKV_RESTCLI_CONFIG=$OKV_RESTCLI_DIR/conf/okvrestcli.ini` file. Set `JAVA_HOME` as follows:

```
#!/bin/bash
    export OKV_RESTCLI_DIR=$(dirname "${0}")/..
    export OKV_RESTCLI_CONFIG=$OKV_RESTCLI_DIR/conf/okvrestcli.ini
    export JAVA_HOME=/opt/oracle/dcs/java/1.8.0_xxx
    if [ -z "$JAVA_HOME" ]
    then
      echo "JAVA_HOME environment variable is not set."
      exit 1
    fi

    if [ -z "$OKV_RESTCLI_CONFIG" ]
    then
      echo "OKV_RESTCLI_CONFIG environment variable is not set."
      exit 1
    fi

    export OKV_RESTCLI_JAR=$OKV_RESTCLI_DIR/lib/okvrestcli.jar
    $JAVA_HOME/bin/java -jar $OKV_RESTCLI_JAR "$@"
```

4. Generate the client auto login wallets. Specify the Oracle Key Vault password when prompted.

```
/tmp/work/bin/okv admin client-wallet add --client-wallet /tmp/work/
client_wallet --wallet-user okv_user
    Password:
```

5. Create the endpoint corresponding to the wallet. For Oracle RAC database, you must create two endpoints specific to each database instance. Run this step only if the endpoints are not already created using the Oracle Key Vault Server user interface.

```
/tmp/work/bin/okv admin endpoint create --endpoint endpoint_name --type
ORACLE_DB --platform LINUX64 --strict-ip-check TRUE
```

6. Set the wallet created in Step 1 as the default wallet for the endpoints. For Oracle RAC database, the default wallet for both the endpoints must be same TDE wallet. Run this step only if the TDE wallet is not already set as the default wallet for the endpoints using the Oracle Key Vault Server user interface.

```
/tmp/work/bin/okv manage-access wallet set-default --wallet <wallet_name>
--endpoint <endpoint_name>
```

7. Download the `okvclient.jar` file. For Oracle RAC database, download the `okvclient.jar` file corresponding to both endpoints. After downloading the files, copy the `okvclient.jar` corresponding to the remote instance to the remote node.

```
/tmp/work/bin/okv admin endpoint download --endpoint <endpoint_name> --
location <location>
```

When you download the endpoints, if you encounter a message such as `Endpoint token is already consumed`, then you must enroll the endpoint again as follows:

```
/tmp/work/bin/okv admin endpoint download --endpoint endpoint_name --
location location
/tmp/work/log/okv0.log.0 (Permission denied)
{
  "result" : "Failure",
  "message" : "Endpoint token is already consumed"
}

/tmp/work/bin/okv admin endpoint re-enroll --endpoint <endpoint_name>
/tmp/work/log/okv0.log.0 (Permission denied)
{
  "result" : "Success"
}
}
```

Step 4: Restoring TDE-enabled databases with TDE keys stored on Oracle Key Vault

For multi-user access-enabled systems, log in as database user. For non-multi user access enabled systems, log in as the `root` user. Use the `odacli irestore-database` command to restore a TDE-enabled database with TDE keys stored on Oracle Key Vault. Specify the options `--enable-tde`, `--okvclient-path`, and `keystore-type` in the command.

The option `--enable-tde,-t` indicates that the restored database must be configured with TDE. Use the option `--keystore-type,-kt` to specify the type of keystore for the TDE configuration. Use the option `--okvclient-path,-ocp` to specify the absolute path to the `okvclient.jar` file corresponding to the database endpoint. For single-instance database, you can specify one value in the `--okvclient-path,-ocp` option, whereas for Oracle RAC database, you can specify two values. Provide the TDE password when you are prompted. After the restore operation of the database is successful, the Oracle Key Vault endpoint software is installed in the `/etc/OKV/db_unique_name/okv` location. The local auto login TDE wallets are created in the `/etc/OKV/db_unique_name/tde` location.

To restore a single-instance database `okvdb` without using the Oracle Key Vault server configuration object, run the `odacli irestore-database` command as follows:

```
odacli irestore-database -r /tmp/bkpreport.json -n okvdb8 -y SI -t -kt OKV -
ocp /tmp/okvclient.jar
Enter SYS and SYSTEM user password:
Retype SYS and SYSTEM user password:
Enter TDE wallet password:

Job details
----------------------------------------------------------------
                     ID:  dce7bde6-7fc4-48ec-891b-649525f1ce4f
            Description:  Database service recovery with DB name: okvdb8
                 Status:  Success
                Created:  August 28, 2024 2:50:46 PM CST
                Message:

Task Name                                Node Name                 Start
Time                                 End Time
Status
---------------------------------------- ------------------------
----------------------------------------
---------------------------------------- ----------------
Restore Database                         scaoda10ha13c2n1          August 28,
2024 2:51:09 PM CST          August 28, 2024 2:58:11 PM CST
Failure
Restore Database                         scaoda10ha13c2n1          August 28,
2024 2:51:10 PM CST          August 28, 2024 2:58:10 PM CST
Failure
Check if cluster ware is running         scaoda10ha13c2n1          August 28,
2024 2:51:14 PM CST          August 28, 2024 2:51:14 PM CST
Success
Check if cluster ware is running         scaoda10ha13c2n2          August 28,
2024 2:51:14 PM CST          August 28, 2024 2:51:14 PM CST
Success
Creating DbStorage for DbRestore         scaoda10ha13c2n1          August 28,
2024 2:51:14 PM CST          August 28, 2024 2:51:21 PM CST
Success
Validating DiskSpace for DATA            scaoda10ha13c2n1          August 28,
2024 2:51:14 PM CST          August 28, 2024 2:51:15 PM CST
Success
Setting up SSH equivalence               scaoda10ha13c2n1          August 28,
2024 2:51:15 PM CST          August 28, 2024 2:51:19 PM CST
Success
Configuring user access to ACFS          scaoda10ha13c2n1          August 28,
2024 2:51:20 PM CST          August 28, 2024 2:51:20 PM CST
```

```
Success
filesystems for
okvdb8st

Audit directory creation                scaoda10ha13c2n1          August 28,
2024 2:51:20 PM CST          August 28, 2024 2:51:21 PM CST
Success
Audit directory creation                scaoda10ha13c2n2          August 28,
2024 2:51:21 PM CST          August 28, 2024 2:51:21 PM CST
Success
Configure TDE for RestoreDb TaskFlow    scaoda10ha13c2n1          August 28,
2024 2:51:22 PM CST          August 28, 2024 2:51:39 PM CST
Success
Auxiliary Instance Creation             scaoda10ha13c2n2          August 28,
2024 2:51:24 PM CST          August 28, 2024 2:51:36 PM CST
Success
TDE Wallet directory creation           scaoda10ha13c2n1          August 28,
2024 2:51:36 PM CST          August 28, 2024 2:51:36 PM CST
Success
Installing OKV Client Software          scaoda10ha13c2n2          August 28,
2024 2:51:36 PM CST          August 28, 2024 2:51:37 PM CST
Success
Auto login TDE Wallet creation          scaoda10ha13c2n1          August 28,
2024 2:51:37 PM CST          August 28, 2024 2:51:38 PM CST
Success
Password based TDE Wallet open          scaoda10ha13c2n1          August 28,
2024 2:51:38 PM CST          August 28, 2024 2:51:39 PM CST
Success
Install Object Store Swift module       scaoda10ha13c2n2          August 28,
2024 2:51:58 PM CST          August 28, 2024 2:52:16 PM CST
Success
Restoring Spfile From Casper            scaoda10ha13c2n2          August 28,
2024 2:52:16 PM CST          August 28, 2024 2:52:38 PM CST
Success
Customize Db Parameters                 scaoda10ha13c2n2          August 28,
2024 2:52:38 PM CST          August 28, 2024 2:52:47 PM CST
Success
Create spfile for restore db            scaoda10ha13c2n2          August 28,
2024 2:52:47 PM CST          August 28, 2024 2:52:48 PM CST
Success
Restoring control file                  scaoda10ha13c2n2          August 28,
2024 2:52:48 PM CST          August 28, 2024 2:53:11 PM CST
Success
Mounting db                             scaoda10ha13c2n2          August 28,
2024 2:53:11 PM CST          August 28, 2024 2:53:35 PM CST
Success
Validating backup for RestoreDB         scaoda10ha13c2n2          August 28,
2024 2:53:35 PM CST          August 28, 2024 2:53:41 PM CST
Success
Restoring DB for migration              scaoda10ha13c2n2          August 28,
2024 2:53:41 PM CST          August 28, 2024 2:54:26 PM CST
Success
Change DBID and/or DBName               scaoda10ha13c2n2          August 28,
2024 2:54:27 PM CST          August 28, 2024 2:54:28 PM CST
Success
Register Database taskflow              scaoda10ha13c2n1          August 28,
```

```
2024 2:54:31 PM CST          August 28, 2024 2:57:08 PM CST
Success
Create SPFile in shared loc             scaoda10ha13c2n2          August 28,
2024 2:54:31 PM CST          August 28, 2024 2:54:37 PM CST
Success
Delete Local Spfile                     scaoda10ha13c2n2          August 28,
2024 2:54:37 PM CST          August 28, 2024 2:54:37 PM CST
Success
Register DB with clusterware            scaoda10ha13c2n2          August 28,
2024 2:54:37 PM CST          August 28, 2024 2:55:23 PM CST
Success
Set SysPassword and Create PwFile       scaoda10ha13c2n1          August 28,
2024 2:55:23 PM CST          August 28, 2024 2:55:25 PM CST
Success
Enable block change tracking            scaoda10ha13c2n2          August 28,
2024 2:55:26 PM CST          August 28, 2024 2:55:28 PM CST
Success
Creating pfile                          scaoda10ha13c2n2          August 28,
2024 2:55:28 PM CST          August 28, 2024 2:55:29 PM CST
Success
Updating db env                         scaoda10ha13c2n2          August 28,
2024 2:55:29 PM CST          August 28, 2024 2:55:30 PM CST
Success
Enable DbSizing Template                scaoda10ha13c2n2          August 28,
2024 2:55:30 PM CST          August 28, 2024 2:56:20 PM CST
Success
Create tns entry                        scaoda10ha13c2n1          August 28,
2024 2:56:20 PM CST          August 28, 2024 2:56:21 PM CST
Success
Create tns entry                        scaoda10ha13c2n2          August 28,
2024 2:56:21 PM CST          August 28, 2024 2:56:22 PM CST
Success
Running datapatch                       scaoda10ha13c2n2          August 28,
2024 2:56:22 PM CST          August 28, 2024 2:56:23 PM CST
Success
Set CPU pool                            scaoda10ha13c2n1          August 28,
2024 2:56:23 PM CST          August 28, 2024 2:56:23 PM CST
Success
Reset Associated Networks for Database  scaoda10ha13c2n2          August 28,
2024 2:57:10 PM CST          August 28, 2024 2:57:13 PM CST
Success
Reset Associated Networks               scaoda10ha13c2n1          August 28,
2024 2:57:13 PM CST          August 28, 2024 2:57:14 PM CST
Success
Set log_archive_dest for Database       scaoda10ha13c2n2          August 28,
2024 2:57:14 PM CST          August 28, 2024 2:57:18 PM CST
Success
Setting Database parameter for OKV      scaoda10ha13c2n1          August 28,
2024 2:57:18 PM CST          August 28, 2024 2:58:05 PM CST
Success
Enable New Tablespace Encryption        scaoda10ha13c2n2          August 28,
2024 2:58:07 PM CST          August 28, 2024 2:58:08 PM CST
Success
Deleting OKV Client Software Jar        scaoda10ha13c2n2          August 28,
2024 2:58:09 PM CST          August 28, 2024 2:58:09 PM CST
Success
```

```
Copy Pwfile to Shared Storage            scaoda10ha13c2n1          August 28,
2024 2:58:09 PM CST           August 28, 2024 2:58:10 PM CST           Success
```

To restore an Oracle RAC database, run the `odacli irestore-database` command as follows. Specify the `okvclient.jar` file for the local database instance first, followed by the path to the `okvclient.jar` file for the remote database instance in the request. In the following example, the `okvclient.jar` file in the `/tmp/dir1` location belongs to the local node and the `okvclient.jar` file in the `/tmp/dir2` location belongs to the remote node.

```
odacli irestore-database -r /tmp/bkpreport.json -n okvdb7 -y RAC -t -kt OKV -
ocp /tmp/dir1/okvclient.jar,/tmp/dir2/okvclient.jar
Enter SYS and SYSTEM user password:
Retype SYS and SYSTEM user password:
Enter TDE wallet password:

Job details
----------------------------------------------------------------
                     ID:  79b2d96c-0156-444b-b965-185b311f5110
            Description:  Database service recovery with DB name: okvdb7
                 Status:  Success
                Created:  August 28, 2024 12:58:39 AM CST
                Message:

Task Name                                Node Name                 Start
Time                               End Time
Status
---------------------------------------- ------------------------
----------------------------------------
---------------------------------------- ----------------
Check if cluster ware is running         scaoda10ha13c2n1          August 28,
2024 12:59:07 AM CST          August 28, 2024 12:59:07 AM CST
Success
Check if cluster ware is running         scaoda10ha13c2n2          August 28,
2024 12:59:07 AM CST          August 28, 2024 12:59:07 AM CST
Success
Creating DbStorage for DbRestore         scaoda10ha13c2n1          August 28,
2024 12:59:07 AM CST          August 28, 2024 12:59:14 AM CST
Success
Validating DiskSpace for DATA            scaoda10ha13c2n1          August 28,
2024 12:59:08 AM CST          August 28, 2024 12:59:08 AM CST
Success
Setting up SSH equivalence               scaoda10ha13c2n1          August 28,
2024 12:59:09 AM CST          August 28, 2024 12:59:12 AM CST
Success
Configuring user access to ACFS          scaoda10ha13c2n1          August 28,
2024 12:59:13 AM CST          August 28, 2024 12:59:13 AM CST
Success
filesystems for
okvdb7

Audit directory creation                 scaoda10ha13c2n1          August 28,
2024 12:59:14 AM CST          August 28, 2024 12:59:14 AM CST
Success
Audit directory creation                 scaoda10ha13c2n2          August 28,
2024 12:59:14 AM CST          August 28, 2024 12:59:14 AM CST
```

```
Success
Configure TDE for RestoreDb TaskFlow     scaoda10ha13c2n1          August 28,
2024 12:59:15 AM CST          August 28, 2024 12:59:32 AM CST
Success
Auxiliary Instance Creation              scaoda10ha13c2n1          August 28,
2024 12:59:17 AM CST          August 28, 2024 12:59:28 AM CST
Success
TDE Wallet directory creation            scaoda10ha13c2n1          August 28,
2024 12:59:28 AM CST          August 28, 2024 12:59:28 AM CST
Success
Installing OKV Client Software           scaoda10ha13c2n1          August 28,
2024 12:59:28 AM CST          August 28, 2024 12:59:29 AM CST
Success
Installing OKV Client Software           scaoda10ha13c2n2          August 28,
2024 12:59:30 AM CST          August 28, 2024 12:59:31 AM CST
Success
Auto login TDE Wallet creation           scaoda10ha13c2n1          August 28,
2024 12:59:31 AM CST          August 28, 2024 12:59:32 AM CST
Success
Password based TDE Wallet open           scaoda10ha13c2n1          August 28,
2024 12:59:32 AM CST          August 28, 2024 12:59:32 AM CST
Success
Install Object Store Swift module        scaoda10ha13c2n2          August 28,
2024 12:59:51 AM CST          August 28, 2024 1:00:09 AM CST
Success
Restoring Spfile From Casper             scaoda10ha13c2n1          August 28,
2024 1:00:09 AM CST           August 28, 2024 1:00:28 AM CST
Success
Customize Db Parameters                  scaoda10ha13c2n1          August 28,
2024 1:00:28 AM CST           August 28, 2024 1:00:34 AM CST
Success
Create spfile for restore db             scaoda10ha13c2n1          August 28,
2024 1:00:34 AM CST           August 28, 2024 1:00:35 AM CST
Success
Restoring control file                   scaoda10ha13c2n1          August 28,
2024 1:00:35 AM CST           August 28, 2024 1:00:58 AM CST
Success
Mounting db                              scaoda10ha13c2n1          August 28,
2024 1:00:58 AM CST           August 28, 2024 1:01:23 AM CST
Success
Validating backup for RestoreDB          scaoda10ha13c2n1          August 28,
2024 1:01:23 AM CST           August 28, 2024 1:01:29 AM CST
Success
Restoring DB for migration               scaoda10ha13c2n1          August 28,
2024 1:01:29 AM CST           August 28, 2024 1:10:54 AM CST
Success
Re-Create control file                   scaoda10ha13c2n1          August 28,
2024 1:10:54 AM CST           August 28, 2024 1:11:36 AM CST
Success
Change DBID and/or DBName                scaoda10ha13c2n1          August 28,
2024 1:11:36 AM CST           August 28, 2024 1:14:34 AM CST
Success
Removing Disabled Redo Threads           scaoda10ha13c2n1          August 28,
2024 1:14:34 AM CST           August 28, 2024 1:14:36 AM CST
Success
Updating DB attributes                   scaoda10ha13c2n1          August 28,
```

```
2024 1:14:36 AM CST          August 28, 2024 1:14:37 AM CST
Success
Customize DB for Type RAC/RacOne         scaoda10ha13c2n1          August 28,
2024 1:14:37 AM CST          August 28, 2024 1:14:44 AM CST
Success
Enable cluster ware for rac/racone db    scaoda10ha13c2n1          August 28,
2024 1:14:45 AM CST          August 28, 2024 1:14:45 AM CST
Success
Building Cluster DB Views for Rac and    scaoda10ha13c2n1          August 28,
2024 1:14:45 AM CST          August 28, 2024 1:14:56 AM CST
Success
RacOne


Register Database taskflow               scaoda10ha13c2n1          August 28,
2024 1:14:58 AM CST          August 28, 2024 1:20:24 AM CST
Success
Create SPFile in shared loc              scaoda10ha13c2n1          August 28,
2024 1:14:59 AM CST          August 28, 2024 1:15:05 AM CST
Success
Delete Local Spfile                      scaoda10ha13c2n1          August 28,
2024 1:15:05 AM CST          August 28, 2024 1:15:05 AM CST
Success
Register DB with clusterware             scaoda10ha13c2n1          August 28,
2024 1:15:05 AM CST          August 28, 2024 1:16:33 AM CST
Success
Add Startup Trigger to Open all PDBS     scaoda10ha13c2n1          August 28,
2024 1:16:33 AM CST          August 28, 2024 1:16:33 AM CST
Success
Set SysPassword and Create PwFile        scaoda10ha13c2n1          August 28,
2024 1:16:34 AM CST          August 28, 2024 1:16:38 AM CST
Success
Enable block change tracking             scaoda10ha13c2n1          August 28,
2024 1:16:38 AM CST          August 28, 2024 1:16:54 AM CST
Success
Creating pfile                           scaoda10ha13c2n1          August 28,
2024 1:16:55 AM CST          August 28, 2024 1:16:56 AM CST
Success
Updating db env                          scaoda10ha13c2n1          August 28,
2024 1:16:56 AM CST          August 28, 2024 1:16:56 AM CST
Success
Enable DbSizing Template                 scaoda10ha13c2n1          August 28,
2024 1:16:56 AM CST          August 28, 2024 1:18:38 AM CST
Success
Update Database Global Name              scaoda10ha13c2n1          August 28,
2024 1:18:39 AM CST          August 28, 2024 1:18:40 AM CST
Success
Create tns entry                         scaoda10ha13c2n1          August 28,
2024 1:18:40 AM CST          August 28, 2024 1:18:42 AM CST
Success
Create tns entry                         scaoda10ha13c2n2          August 28,
2024 1:18:42 AM CST          August 28, 2024 1:18:43 AM CST
Success
Running datapatch                        scaoda10ha13c2n1          August 28,
2024 1:18:43 AM CST          August 28, 2024 1:19:19 AM CST
Success
```

```
Set CPU pool                              scaoda10ha13c2n1          August 28,
2024 1:19:19 AM CST           August 28, 2024 1:19:19 AM CST
Success
Reset Associated Networks for Database    scaoda10ha13c2n1          August 28,
2024 1:20:26 AM CST           August 28, 2024 1:20:30 AM CST
Success
Reset Associated Networks                 scaoda10ha13c2n2          August 28,
2024 1:20:30 AM CST           August 28, 2024 1:20:31 AM CST
Success
Set log_archive_dest for Database         scaoda10ha13c2n1          August 28,
2024 1:20:31 AM CST           August 28, 2024 1:20:34 AM CST
Success
Auto login TDE Wallet creation            scaoda10ha13c2n1          August 28,
2024 1:20:34 AM CST           August 28, 2024 1:20:36 AM CST
Success
Setting Database parameter for OKV        scaoda10ha13c2n1          August 28,
2024 1:20:36 AM CST           August 28, 2024 1:22:17 AM CST
Success
Enable New Tablespace Encryption          scaoda10ha13c2n1          August 28,
2024 1:22:19 AM CST           August 28, 2024 1:22:20 AM CST
Success
Deleting OKV Client Software Jar          scaoda10ha13c2n1          August 28,
2024 1:22:20 AM CST           August 28, 2024 1:22:20 AM CST
Success
Deleting OKV Client Software Jar          scaoda10ha13c2n2          August 28,
2024 1:22:20 AM CST           August 28, 2024 1:22:20 AM CST           Success
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Deleting TDE-enabled Databases

Understand how you can delete a TDE-enabled database on Oracle Database Appliance on bare metal and DB systems, that uses Oracle Key Vault to store TDE keys.

Use the `odacli delete-database` command to delete TDE-enabled database with TDE keys stored on Oracle Key Vault. Use the `-n` option to specify the name of the database to be deleted.

```
odacli delete-database -n kokvdb1
{
  "jobId" : "6f001840-5002-43ee-9067-e22930e36d58",
  "status" : "Running",
  "message" : null,
  "reports" : [ {
    "taskId" : "TaskDcsJsonRpcExt_2259",
    "taskName" : "Validate DB 9aeeeb5f-afc5-4051-8bd1-00c62048ca01 for
deletion",
    "nodeName" : "scaoda6m006",
    "taskResult" : "",
    "startTime" : "July 15, 2024 15:40:50 PM CST",
    "endTime" : "July 15, 2024 15:40:50 PM CST",
    "duration" : "00:00:00.13",
    "status" : "Success",
```

```
    "taskDescription" : null,
    "parentTaskId" : "TaskSequential_2257",
    "jobId" : "6f001840-5002-43ee-9067-e22930e36d58",
    "tags" : [ ],
    "reportLevel" : "Info",
    "updatedTime" : "July 15, 2024 15:40:50 PM CST"
  } ],
  "createTimestamp" : "July 15, 2024 15:40:49 PM CST",
  "resourceList" : [ ],
  "description" : "Database service deletion with DB name: kokvdb1 with ID :
9aeeeb5f-afc5-4051-8bd1-00c62048ca01",
  "updatedTime" : "July 15, 2024 15:40:50 PM CST",
  "jobType" : null
}

 odacli describe-job -i 6f001840-5002-43ee-9067-e22930e36d58

Job details
----------------------------------------------------------------
                     ID:  6f001840-5002-43ee-9067-e22930e36d58
            Description:  Database service deletion with DB name: kokvdb1
with ID : 9aeeeb5f-afc5-4051-8bd1-00c62048ca01
                 Status:  Success
                Created:  July 15, 2024 3:40:49 PM CST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Validate DB                              July 15, 2024 3:40:50 PM
CST             July 15, 2024 3:40:50 PM CST             Success
9aeeeb5f-
afc5-4051-8bd1-00c62048ca01

for
deletion

Deleting the RMAN logs                   July 15, 2024 3:40:50 PM
CST             July 15, 2024 3:40:51 PM CST             Success
OKV Endpoint deletion                    July 15, 2024 3:40:51 PM
CST             July 15, 2024 3:40:56 PM CST             Success
OKV Wallet deletion                      July 15, 2024 3:40:56 PM
CST             July 15, 2024 3:41:05 PM CST             Success
OKV Files deletion                       July 15, 2024 3:41:05 PM
CST             July 15, 2024 3:41:05 PM CST             Success
Database Deletion By RHP                 July 15, 2024 3:41:05 PM
CST             July 15, 2024 3:42:37 PM CST             Success
Unregister DB From Cluster               July 15, 2024 3:42:37 PM
CST             July 15, 2024 3:42:37 PM CST             Success
Kill PMON Process                        July 15, 2024 3:42:37 PM
CST             July 15, 2024 3:42:37 PM CST             Success
Database Files Deletion                  July 15, 2024 3:42:37 PM
CST             July 15, 2024 3:42:49 PM CST             Success
```

```
Delete File Groups of Database kokvdb1   July 15, 2024 3:42:49 PM
CST             July 15, 2024 3:42:58 PM CST             Success
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Creating an Oracle Key Vault User Using BUI

Create a user with Oracle Key Vault server configuration entitlement on a multi-user access enabled Oracle Database Appliance.

Creating an Oracle Key Vault User with Browser User Interface

1. Log into the Browser User Interface as the `odaadmin` user:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Multi-User Access** tab.
3. Click the Users link on the left-hand pane.
4. Click **Create User**.
5. In the Create User page, provide the **User ID**, specify the **Role** as `ODA-OKVCONFIGADMIN`, and provide the **ODA Password** for this user. Note that the same user credentials work for login for BUI and ODACLI commands.
6. Click **Create**.
7. The job is submitted and a confirmation page appears with a link to the job. Click the link to view the job progress, tasks, and status. After you close the Job confirmation page, you can click the **Activity** tab to monitor the job progress. Click the job number to view the tasks and status details. Click **Refresh** to refresh the page.
8. The account is created with Inactive state. Log into the BUI with the user credentials. You are prompted to change the password. Change the password and log into the BUI with the new password.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Creating Oracle Key Vault Server Configuration Object Using the BUI

Understand how you can create TDE-enabled database on Oracle Database Appliance with TDE keys stored in Oracle Key Vault on bare metal and DB systems using the BUI.

Follow these steps to create a TDE-enabled database with TDE keys stored on Oracle Key Vault on Oracle Database Appliance bare metal and DB systems:

1. Log into the Browser User Interface:
   `https://host-ip-address:7093/mgmt/index.html`
2. Click **Credential Store**, then click **Oracle Key Vault Server Config**.
3. The existing Oracle Key Vault server configurations are displayed.
4. Click **Create** and create an Oracle Key Vault server configuration.

5. In the Create OKV Server Config page, specify the **OKV Server Config Name**, **OKV Server Host**, **OKV Server IP address**, **OKV User Name**, and **Password**.
6. Click **Create**.
7. Confirm the creation request job and verify that the job completed successfully.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Granting and Revoking Access on Oracle Key Vault Server Configuration Using BUI

Understand how you can grant and revoke access to Oracle Key Vault server configuration on Oracle Database Appliance on bare metal and DB systems using BUI.

Granting and Revoking Access to Oracle Key Vault server configurations

Follow these steps:

1. Log into the Browser User Interface:
   `https://host-ip-address:7093/mgmt/index.html`
2. Click **Credential Store**, then click **Oracle Key Vault Server Config**.
3. You can view all existing Oracle Key Vault server configurations.
4. To grant access to an existing Oracle Key Vault server configuration, select the Oracle Key Vault server configuration, and from the **Actions** drop-down list, select **Grant Access**.
5. In the Grant Resource Access page, select the **User Name** to whom you want to grant access to the resource.
6. Click **Grant**.
7. Click **Yes** to confirm the action.
8. To revoke access to an existing Oracle Key Vault server configuration, select the Oracle Key Vault server configuration, and from the **Actions** drop-down list, select **Revoke Access**.
9. In the Revoke Resource Access page, select the **User Name** from whom you want to revoke access to the resource.
10. Click **Revoke**.
11. Click **Yes** to confirm the action.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Managing Oracle Key Vault Server Configuration Operations Using BUI

Understand how you can manage Oracle Key Vault server configuration operations on Oracle Database Appliance on bare metal and DB systems.

You can create, list, delete, and describe Oracle Key Vault server configuration on Oracle Database Appliance bare metal and DB systems using the Browser User Interface (BUI).

Listing and Deleting Oracle Key Vault server configurations:

Follow these steps:

1. Log into the Browser User Interface:
   `https://host-ip-address:7093/mgmt/index.html`
2. Click **Credential Store**, then click **Oracle Key Vault Tasks**.
3. You can view all existing Oracle Key Vault server configurations.
4. To delete an existing Oracle Key Vault server configuration, select the Oracle Key Vault server configuration, and from the **Actions** drop-down list, select **Delete**.

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Manual Migration and Encryption of Data into TDE-Enabled Database

Understand how you can migrate transportable tablespaces to TDE-enabled database on Oracle Database Appliance.

- Migrating Tablespaces from Non-TDE Oracle ACFS Database to Oracle ACFS Database on Oracle Database Appliance
  Understand how you can migrate tablespaces from non-TDE Oracle ACFS database to Oracle ACFS database with Oracle Key Vault enabled on Oracle Database Appliance.
- Migrating Tablespaces from Non-TDE Oracle ASM Database to Oracle ASM Database on Oracle Database Appliance
  Understand how you can migrate tablespaces from non-TDE Oracle ASM database to Oracle ASM database with Oracle Key Vault enabled on Oracle Database Appliance.
- Migrating Tablespaces from Non-TDE Oracle ASM Database to Oracle ASM CDB Database on Oracle Database Appliance
  Understand how you can migrate tablespaces from non-TDE Oracle ASM database to Oracle ASM CDB database with Oracle Key Vault enabled on Oracle Database Appliance.
- Migrating Tablespaces from Non-TDE Oracle ACFS Database to Oracle ACFS CDB Database on Oracle Database Appliance
  Understand how you can migrate tablespaces from non-TDE Oracle ACFS database to Oracle ACFS CDB database with Oracle Key Vault enabled on Oracle Database Appliance.

Migrating Tablespaces from Non-TDE Oracle ACFS Database to Oracle ACFS Database on Oracle Database Appliance

Understand how you can migrate tablespaces from non-TDE Oracle ACFS database to Oracle ACFS database with Oracle Key Vault enabled on Oracle Database Appliance.

Ensure that there is a source and destination database. The two databases can be on one Oracle Database Appliance server or on different servers. The source database must be an non TDE-enabled database, either an Oracle Database Appliance database or any Oracle database with compatible Endian. The destination database must be an Oracle Database Appliance database created with ODACLI commands.

Step 1: Export the tablespace from the source database

1. Create a tablespace TBS01 and table TAB1 on source database DBNOTDE Create an extra tablespace TBS01 on the source database, and then, create table TAB1 on the new tablespace.

```
SQL> select TABLESPACE_NAME, ENCRYPTED from dba_tablespaces ;

 TABLESPACE_NAME           STATUS    ENC
------------------------------ --------- ---
SYSTEM                  ONLINE    NO
SYSAUX                  ONLINE    NO
UNDOTBS1                ONLINE    NO
TEMP                    ONLINE    NO
USERS                   ONLINE    NO

###
### Checks if a set of tablespaces (to be transported) is self-contained
###

SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('tbs01',true) ;

PL/SQL procedure successfully completed.
```

2. Create the database directory in the source database DBNOTDE to be used for the export. Create a database directory object in the destination database to be used by `expdp`.

```
###
### login as oracle user
###

$ mkdir -pv /u01/app/odaorabase0/oracle/dpdump/ttsps_dir
mkdir: created directory '/u01/app/odaorabase0/oracle/dpdump/ttsps_dir'

###
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.

SQL>
```

3. Export the TBS01 tablespace from souce database DBNOTDE. First, set the tablespace to be copied in READ ONLY mode. Next, run `expdp` on the designated tablespace. Review the log files for `expdp` stored under the database directory `ttsps_dir`.

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###
```

```
SQL> ALTER TABLESPACE TBS01 READ ONLY;

Tablespace altered.


###
### Run expdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$ expdp system TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES
DIRECTORY=ttsps_dir DUMPFILE=expttps.dmp

Export: Release 19.0.0.0.0 - Production on Mon Nov 6 12:47:00 2023
Version 19.25.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01":  system/********
TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES DIRECTORY=ttsps_dir
DUMPFILE=expttps.dmp
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
******************************************************************************
***
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /u01/app/odaorabase0/oracle/dpdump/ttsps_dir/expttps.dmp
******************************************************************************
***
Datafiles required for transportable tablespace TBS01:
  /u02/app/oracle/oradata/dbnotde/DBNOTDE/datafile/
o1_mf_tbs01_lnlgn65j_.dbf
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Mon
Nov 6 12:47:19 2023 elapsed 0 00:00:14
```

4. Move data files to the destination database server and edit the status of the data file in the source database DBNOTDE to `READ WRITE`. Move the data file to the TDE-enabled database server in the location `/u02/app/oracle/oradata/dbtde/DBTDE/datafile`.

```
###
### login as oracle user
###

$ cp /u02/app/oracle/oradata/dbnotde/DBNOTDE/datafile/
o1_mf_tbs01_lnlgn65j_.dbf /u02/app/oracle/oradata/dbtde/DBTDE/datafile/
o1_mf_tbs01_lnlgn65j_.dbf
```

```
###
### In source database, set the tablespace back to read write mode
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> ALTER TABLESPACE TBS01 READ WRITE;
```

Step 2: Import the tablespaces into a TDE-enabled database

1. Create a TDE-enabled database with Oracle Key Vault, DBTDE as the destination database, as described in the earlier procedure. Copy the data file and tablespace of the source database to this database using `expdp` and `impdp`. Create the database on an Oracle Database Appliance using the Browser User Interface (BUI) or ODACLI commands. For example:

```
# odacli create-database -n dbtde -t -kt okv -osc adminobj1 -dh
e2a65596-6435-4410-9e8d-6b21a6005779 -r ACFS

# odacli describe-database -n dbtde
Database details
----------------------------------------------------------------
                     ID: e261b9f8-8212-499b-840e-eb6ea3e09447
            Description: dbtde
                DB Name: dbtde
                DB Type: SI
                    CDB: false
                Storage: ACFS
                Home ID: e2a65596-6435-4410-9e8d-6b21a6005779
  TDE Wallet Management: ODA
           TDE Enabled: true
         KeystoreType : OKV
```

2. Create the database directory in the Oracle Key Vault-enabled destination database for the import operation. The source database expdp dump file and output for impdp is stored in the database directory object. Copy the expdp dump file from the source database server to this server, and use the impdp dump file to add tablespace metadata to destination database.

```
###
### login as sysdba user and run the following:
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.
```

3. Import the TBS01 tablespace to the destination database DBTDE. Copy the `expdp` dump file of the source database to the database directory `ttsps_dir`. The `impdp` uses the dump

file to add tablespace metadata to destination database. TRANSPORT_DATAFILES references the location of the data file in the destination database.

```
###
### Run impdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$ impdp system  TRANSPORT_DATAFILES=\'/u02/app/oracle/oradata/dbtde/DBTDE/
datafile/o1_mf_tbs01_lnlgn65j_.dbf\' DIRECTORY=ttsps_dir
dumpfile=expttps.dmp

Import: Release 19.0.0.0.0 - Production on Mon Nov 6 12:53:37 2023
Version 19.25.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01":  system/********
TRANSPORT_DATAFILES='/u02/app/oracle/oradata/dbtde/DBTDE/datafile/
o1_mf_tbs01_lnlgn65j_.dbf' DIRECTORY=ttsps_dir dumpfile=expttps.dmp
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Mon
Nov 6 12:53:46 2023 elapsed 0 00:00:03
```

4. Enable and test encryption on the destination Oracle Key Vault-enabled database.

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###


SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME                STATUS    ENC
------------------------------ --------- ---
SYSTEM                         ONLINE    YES
SYSAUX                         ONLINE    YES
UNDOTBS1                       ONLINE    YES
TEMP                           ONLINE    YES
USERS                          ONLINE    YES
TBS01                          READ ONLY NO


SQL> alter tablespace TBS01 read write ;
```

```
Tablespace altered.

SQL> alter tablespace TBS01 encryption online encrypt ;

Tablespace altered.
SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME                  STATUS ENC
------------------------------ --------- ---
SYSTEM                         ONLINE   YES
SYSAUX                         ONLINE   YES
UNDOTBS1                       ONLINE   YES
TEMP                           ONLINE   YES
USERS                          ONLINE   YES
TBS01                          ONLINE   YES

###
### Retrieve the data from TAB1
###


SQL> select OWNER, table_name from tab1 where rownum < 3;

OWNER
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$

SYS
ICOL$


###
### When the wallet is closed or unavailable,  same SQL does not retrieve
data
### To simulate wallet absence rename the cwallet.sso wallet under
<wallet_root>/tde to some other name like cwallet.sso.orig
### and close the wallet. To return auto-login wallet , rename file back
to  cwallet.sso and run the SQL again.
###

SQL> show parameter wallet_root

NAME                     TYPE    VALUE
------------------------------------ -----------
------------------------------
wallet_root              string  /etc/OKV/dbtde


SQL> ! mv /etc/OKV/dbtde/tde/cwallet.sso /etc/OKV/dbtde/tde/
cwallet.sso.orig
```

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE ;

SQL> select OWNER, table_name from tab1 where rownum < 3;
select OWNER, table_name from tab1 where rownum < 3
                               *
ERROR at line 1:
ORA-28365: wallet is not open


SQL> ! mv /etc/OKV/dbtde/tde/cwallet.sso.orig /etc/OKV/dbtde/tde/
cwallet.sso

SQL> select OWNER, table_name from tab1 where rownum < 3
  2  ;

OWNER
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$

SYS
ICOL$
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Migrating Tablespaces from Non-TDE Oracle ASM Database to Oracle ASM Database on Oracle Database Appliance

Understand how you can migrate tablespaces from non-TDE Oracle ASM database to Oracle ASM database with Oracle Key Vault enabled on Oracle Database Appliance.

Ensure that there is a source and destination database. The two databases can be on one Oracle Database Appliance server or on different servers. The source database must be an non TDE-enabled database, either an Oracle Database Appliance database or any Oracle database with compatible Endian. The destination database must be an Oracle Database Appliance database created with ODACLI commands and with Oracle Key Vault enabled on it.

Step 1: Export the tablespace from the source database

1. Create a tablespace TBS01 and table TAB1 on source database ASMNOTDE Create an extra tablespace TBS01 on the source database, and then, create table TAB1 on the new tablespace.

```
SQL>  select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;
TABLESPACE_NAME            STATUS    ENC
------------------------------ --------- ---
SYSTEM                 ONLINE    NO
SYSAUX                 ONLINE    NO
UNDOTBS1               ONLINE    NO
```

```
TEMP                     ONLINE    NO
USERS                    ONLINE    NO
TBS01                    ONLINE    NO

SQL> SQL> select FILE_NAME from dba_data_files where
TABLESPACE_NAME='TBS01' ;

FILE_NAME
--------------------------------------------------------------------------------
-----
+DATA/ASMNOTDE/DATAFILE/tbs01.374.1152267015

###
### Checks if a set of tablespaces (to be transported) is self-contained
###

SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('tbs01',true) ;

PL/SQL procedure successfully completed.
```

2. Create the database directory in the source database ASMNOTDE to use for the export. Create a database directory object in the destination database for `expdp`.

```
###
### login as oracle user
###

$ mkdir -pv /u01/app/odaorabase0/oracle/dpdump/ttsps_dir
mkdir: created directory ‘/u01/app/odaorabase0/oracle/dpdump/ttsps_dir’


###
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.
```

3. Export the TBS01 tablespace from the souce database ASMNOTDE. First, set the tablespace to be copied in READ ONLY mode. Next, run `expdp` on the designated tablespace. Review the dump files for `expdp` stored under the database directory `ttsps_dir`.

```
###
### Set the tablespaces that you need to be exported in read-only mode
### login as Oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following

SQL> ALTER TABLESPACE TBS01 READ ONLY;

Tablespace altered.
```

```
SQL>

###
### Run expdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$ expdp system TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES
DIRECTORY=ttsps_dir DUMPFILE=expttpsasm.dmp

Export: Release 19.0.0.0.0 - Production on Tue Nov 7 10:12:19 2023
Version 19.25.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01":  system/********
TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES DIRECTORY=ttsps_dir
DUMPFILE=expttpsasm.dmp
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
******************************************************************************
***
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /u01/app/odaorabase0/oracle/dpdump/ttsps_dir/expttpsasm.dmp
******************************************************************************
***
Datafiles required for transportable tablespace TBS01:
  +DATA/ASMNOTDE/DATAFILE/tbs01.374.1152267015
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Tue
Nov 7 10:12:49 2023 elapsed 0 00:00:20
```

4. Move data files to the destination database server and edit the status of the data file in the source database ASMNOTDE to `READ WRITE`.

```
###
### login to grid user
### Copying datafiles from ASM to  filesystem
###

$ asmcmd cp +DATA/ASMNOTDE/DATAFILE/tbs01.295.1165369089 /home/grid/
tbs01.dbf
copying +DATA/ASMNOTDE/DATAFILE/tbs01.295.1165369089 -> /home/grid/
tbs01.dbf


###
### In source database, set the tablespace back to read write mode
```

```
### login as Oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> ALTER TABLESPACE TBS01 READ WRITE;
```

Step 2: Import the tablespaces into a TDE-enabled database

1. Create a TDE-enabled database with Oracle Key Vault, ASMTDE as the destination database. Copy the data file and tablespace of the source database to this database using `expdp` and `impdp`. Create the database on an Oracle Database Appliance using the Browser User Interface (BUI) or ODACLI commands. For example:

```
# odacli create-database -n asmtde -t -kt okv -osc adminobj1 -dh
e2a65596-6435-4410-9e8d-6b21a6005779 -r ASM

#  odacli describe-database -n asmtde
Database details
----------------------------------------------------------------
                     ID: 5dfa97d9-d774-4a4a-93a2-d21a36faf283
            Description: asmtde
                DB Name: asmtde
                DB Type: SI
                    CDB: false
                Storage: ASM
                Home ID: e2a65596-6435-4410-9e8d-6b21a6005779
  TDE Wallet Management: ODA
           TDE Enabled: true
         KeystoreType : OKV
```

2. Copy the data file from file system to the destination Oracle ASM location of ASMTDE.

```
###
### login to grid user
###

$ asmcmd cp /home/grid/tbs01.dbf +DATA/ASMTDE/DATAFILE/tbs01 --dest_dbname
ASMTDE
copying /home/grid/tbs01.dbf -> +DATA/ASMTDE/DATAFILE/tbs01

ASMCMD> ls --permission
User                                    Group  Permission  Name
dbusr1@66d5786e6a057fd3bfa6ca745d993aef         rw-------
SYSAUX.292.1165368127
dbusr1@66d5786e6a057fd3bfa6ca745d993aef         rw-------
SYSTEM.279.1165368093
dbusr1@66d5786e6a057fd3bfa6ca745d993aef         rw-------
UNDOTBS1.290.1165368143
dbusr1@66d5786e6a057fd3bfa6ca745d993aef         rw-------
USERS.288.1165368875
  grid@66d5786e6a057fd3bfa6ca745d993aef         rw-------  tbs01 => +DATA/
ASMTDE/DATAFILE/tbs01.294.1165425531
  grid@66d5786e6a057fd3bfa6ca745d993aef         rw-------
tbs01.294.1165425531
```

```
ASMCMD> chown dbusr1@66d5786e6a057fd3bfa6ca745d993aef tbs01
ASMCMD> ls --permission
User                                        Group  Permission  Name
dbusr1@66d5786e6a057fd3bfa6ca745d993aef          rw-------
SYSAUX.292.1165368127
dbusr1@66d5786e6a057fd3bfa6ca745d993aef          rw-------
SYSTEM.279.1165368093
dbusr1@66d5786e6a057fd3bfa6ca745d993aef          rw-------
UNDOTBS1.290.1165368143
dbusr1@66d5786e6a057fd3bfa6ca745d993aef          rw-------
USERS.288.1165368875
dbusr1@66d5786e6a057fd3bfa6ca745d993aef          rw-------  tbs01 => +DATA/
OKVDB2/DATAFILE/tbs012.294.1165425531
dbusr1@66d5786e6a057fd3bfa6ca745d993aef          rw-------
tbs01.294.1165425531
```

3. Create the database directory in the ASMTDE database for the import.

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.
```

4. Import the TBS01 tablespace to the destination database ASMTDE. Copy the `expdp` dump file of the source database to the database directory `ttsps_dir`. The `impdp` uses the dump file to add tablespace metadata to destination database. TRANSPORT_DATAFILES references the location of the data file in the destination database.

```
###
### Run impdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$ impdp system  TRANSPORT_DATAFILES=\'/u02/app/oracle/oradata/dbtde/DBTDE/
datafile/o1_mf_tbs01_lnlgn65j_.dbf\' DIRECTORY=ttsps_dir
dumpfile=expttps.dmp

Import: Release 19.0.0.0.0 - Production on Mon Nov 6 12:53:37 2023
Version 19.25.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01":  system/********
TRANSPORT_DATAFILES='/u02/app/oracle/oradata/dbtde/DBTDE/datafile/
```

```
o1_mf_tbs01_lnlgn65j_.dbf' DIRECTORY=ttsps_dir dumpfile=expttps.dmp
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed
```

5. Enable and test encryption on the destination Oracle Key Vault-enabled database.

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME            STATUS    ENC
------------------------------ --------- ---
SYSTEM                 ONLINE    NO
SYSAUX                 ONLINE    NO
UNDOTBS1               ONLINE    NO
TEMP                   ONLINE    NO
USERS                  ONLINE    YES
TBS01                  READ ONLY NO

6 rows selected.

SQL> alter tablespace TBS01 read write ;

Tablespace altered.

SQL>     alter tablespace TBS01 encryption online encrypt ;

Tablespace altered.

SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME            STATUS    ENC
------------------------------ --------- ---
SYSTEM                 ONLINE    NO
SYSAUX                 ONLINE    NO
UNDOTBS1               ONLINE    NO
TEMP                   ONLINE    NO
USERS                  ONLINE    YES
TBS01                  ONLINE    YES

6 rows selected.

###
### Retrieve the data from TAB1
###

SQL> select OWNER, table_name from tab1 where rownum < 3

OWNER
```

```
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$

SYS
ICOL$

###
### When the wallet is closed or unavailable,  same SQL does not retrieve
data
### To simulate wallet absence rename the cwallet.sso wallet under
<wallet_root>/tde to some other name like cwallet.sso.orig
### and close the wallet. To return auto-login wallet , rename file back
to  cwallet.sso and run the SQL again.
###

SQL> show parameter wallet_root

NAME                     TYPE    VALUE
------------------------------------ -----------
------------------------------
wallet_root              string  /etc/OKV/asmtde


SQL> ! mv /etc/OKV/asmtde/tde/cwallet.sso /etc/OKV/asmtde/tde/
cwallet.sso.orig

SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE ;

SQL> select OWNER, table_name from tab1 where rownum < 3;
select OWNER, table_name from tab1 where rownum < 3
                              *
ERROR at line 1:
ORA-28365: wallet is not open


SQL> ! mv /etc/OKV/asmtde/tde/cwallet.sso.orig /etc/OKV/asmtde/tde/
cwallet.sso

SQL> select OWNER, table_name from tab1 where rownum < 3
  2  ;

OWNER
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$
```

```
SYS
ICOL$
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Migrating Tablespaces from Non-TDE Oracle ASM Database to Oracle ASM CDB Database on Oracle Database Appliance

Understand how you can migrate tablespaces from non-TDE Oracle ASM database to Oracle ASM CDB database with Oracle Key Vault enabled on Oracle Database Appliance.

Ensure that there is a source and destination database. The two databases can be on one Oracle Database Appliance server or on different servers. The source database must be an non TDE-enabled database, either an Oracle Database Appliance database or any Oracle database with compatible Endian. The destination database must be an Oracle Database Appliance database created with ODACLI commands and with Oracle Key Vault enabled on it.

Step 1: Export the tablespace from the source database

1. Create a tablespace TBS01 and table TAB1 on source database ASMNOTDE Create an extra tablespace TBS01 on the source database, and then, create table TAB1 on the new tablespace.

```
SQL> SQL> select FILE_NAME from dba_data_files where
TABLESPACE_NAME='TBS01' ;

FILE_NAME
--------------------------------------------------------------------------------
-----
+DATA/ASMNOTDE/DATAFILE/tbs01.374.1152267015

###
### Checks if a set of tablespaces (to be transported) is self-contained
###

SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('tbs01',true) ;

PL/SQL procedure successfully completed.
```

2. Create the database directory in the source database ASMNOTDE to use for the export. Create a database directory object in the destination database for `expdp`.

```
###
### login as oracle user
###

$ mkdir -pv /u01/app/odaorabase0/oracle/dpdump/ttsps_dir
mkdir: created directory '/u01/app/odaorabase0/oracle/dpdump/ttsps_dir'


###
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
```

```
###

SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.
```

3. Export the TBS01 tablespace from the souce database ASMNOTDE. First, set the tablespace to be copied in READ ONLY mode. Next, run `expdp` on the designated tablespace. Review the dump files for `expdp` stored under the database directory `ttsps_dir`.

```
###
### Set the tablespaces that you need to be exported in read-only mode
### login as Oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following

SQL> ALTER TABLESPACE TBS01 READ ONLY;

Tablespace altered.

SQL>

###
### Run expdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$ expdp system TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES
DIRECTORY=ttsps_dir DUMPFILE=expttpsasm.dmp

Export: Release 19.0.0.0.0 - Production on Tue Nov 7 10:12:19 2023
Version 19.25.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01":  system/********
TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES DIRECTORY=ttsps_dir
DUMPFILE=expttpsasm.dmp
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
******************************************************************************
***
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /u01/app/odaorabase0/oracle/dpdump/ttsps_dir/expttpsasm.dmp
******************************************************************************
***
```

```
Datafiles required for transportable tablespace TBS01:
  +DATA/ASMNOTDE/DATAFILE/tbs01.374.1152267015
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Tue
Nov 7 10:12:49 2023 elapsed 0 00:00:20
```

4. Move data files to the destination database server and edit the status of the data file in the source database ASMNOTDE to `READ WRITE`.

```
###
### login to grid user
### Copying datafiles from ASM to  filesystem
###

$ asmcmd cp +DATA/ASMNOTDE/DATAFILE/tbs01.295.1165369089 /home/grid/
tbs01.dbf
copying +DATA/ASMNOTDE/DATAFILE/tbs01.295.1165369089 -> /home/grid/
tbs01.dbf


###
### In source database, set the tablespace back to read write mode
### login as Oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> ALTER TABLESPACE TBS01 READ WRITE;
```

Step 2: Import the tablespaces into a TDE-enabled database

1. Create a TDE-enabled database with Oracle Key Vault, ASMTDE as the destination database. Copy the data file and tablespace of the source database to this database using `expdp` and `impdp`. Create the database on an Oracle Database Appliance using the Browser User Interface (BUI) or ODACLI commands. For example:

```
# odacli create-database -n asmtde -t -kt okv -osc adminobj1 -dh
e2a65596-6435-4410-9e8d-6b21a6005779 -r ASM --cdb -d pdbadmin -p pdb1

#  odacli describe-database -n asmtde
Database details
----------------------------------------------------------------
                     ID: 5dfa97d9-d774-4a4a-93a2-d21a36faf283
            Description: asmtde
                DB Name: asmtde
                DB Type: SI
                    CDB: true
               PDB Name: PDB2
    PDB Admin User Name: pdbadmin
                Storage: ASM
                Home ID: e2a65596-6435-4410-9e8d-6b21a6005779
  TDE Wallet Management: ODA
           TDE Enabled: true
         KeystoreType : OKV
```

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> col PDB_NAME for a20
SQL> select  PDB_NAME,GUID from  dba_pdbs  ;

PDB_NAME         GUID
-------------------- --------------------------------
PDB1             1599A96E256A8B99E063BE6B1F0AC02B
PDB$SEED         15998A6665985353E063BE6B1F0A494A
```

2. Copy the data file from file system to the destination Oracle ASM location of ASMTDE. Ensure that you set the relevant permissions after copying the file.

```
###
### login to grid user
###

$ asmcmd cp /home/grid/tbs01.dbf +DATA/ASMTDE/
1599A96E256A8B99E063BE6B1F0AC02B/DATAFILE/tbs01 --dest_dbname ASMTDE
copying /home/grid/tbs01.dbf ->  +DATA/ASMTDE/
1599A96E256A8B99E063BE6B1F0AC02B/DATAFILE/tbs01


$ asmcmd

ASMCMD> pwd
+DATA/ASMTDE/1599A96E256A8B99E063BE6B1F0AC02B/DATAFILE
ASMCMD>  ls --permission
User                                    Group  Permission  Name
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
SYSAUX.332.1165769683
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
SYSTEM.333.1165769683
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
UNDOTBS1.331.1165769683
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
USERS.335.1165769685
  grid@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------  tbs01 => +DATA/
ASMTDE/DATAFILE/tbs01.338.1165770781


ASMCMD> chown dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6 tbs01
ASMCMD>
ASMCMD> ls --permission
User                                    Group  Permission  Name
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
SYSAUX.332.1165769683
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
SYSTEM.333.1165769683
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
UNDOTBS1.331.1165769683
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------
```

```
USERS.335.1165769685
dbusr1@00eea686ffb0ff1bbf28c0b8362cccf6          rw-------  tbs01 => +DATA/
ASMTDE/DATAFILE/tbs01.338.1165770781
```

3. Create tnsnames service for PDB1.

```
###
### Edit the file  $ORACLE_HOME/network/admin/tnsnames.ora  with the
following content
###

PDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <server-name>)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb1.test)
```

4. Create the directory in PDB1 for the import.

```
###
### login as oracle user
###

$ mkdir -pv /u01/app/odaorabase0/oracle/dpdump/ttsps_dir
mkdir: created directory ‘/u01/app/odaorabase0/oracle/dpdump/ttsps_dir’

###
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> ALTER SESSION SET CONTAINER=PDB1 ;
SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.

SQL>
```

5. Import the tablespace to PDB1. Copy the `expdp` dump file of the source database to the database directory `ttsps_dir`. The `impdp` uses the dump file to add tablespace metadata to destination database. TRANSPORT_DATAFILES references the location of the data file in the destination database.

```
###
### Run impdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$  impdp system@pdb1  TRANSPORT_DATAFILES=\'+DATA/ASMTDE/
1599A96E256A8B99E063BE6B1F0AC02B/DATAFILE/tbs01\' DIRECTORY=ttsps_dir
dumpfile=expttpsasm.dmp

Import: Release 19.0.0.0.0 - Production on Mon Apr 8 17:52:09 2024
```

```
Version 19.23.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01":  system/********@pdb1
TRANSPORT_DATAFILES='+DATA/ASMTDE/1599A96E256A8B99E063BE6B1F0AC02B/
DATAFILE/tbs01' DIRECTORY=ttsps_dir dumpfile=expttpsasm.dmp
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Mon
Apr 8 17:52:18 2024 elapsed 0 00:00:03
```

6. Enable and test encryption on the destination Oracle Key Vault-enabled database PDB1.

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL>  alter session set container=pdb1 ;

Session altered.

SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME            STATUS    ENC
------------------------------ --------- ---
SYSTEM                 ONLINE    NO
SYSAUX                 ONLINE    NO
UNDOTBS1               ONLINE    NO
TEMP                   ONLINE    NO
USERS                  ONLINE    NO
TBS01                  READ ONLY NO

6 rows selected.

SQL> alter tablespace TBS01 read write ;

Tablespace altered.

SQL>   alter tablespace TBS01 encryption online encrypt ;

Tablespace altered.

SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME            STATUS    ENC
```

```
------------------------------ --------- ---
SYSTEM                         ONLINE    NO
SYSAUX                         ONLINE    NO
UNDOTBS1                       ONLINE    NO
TEMP                           ONLINE    NO
USERS                          ONLINE    NO
TBS01                          ONLINE    YES

6 rows selected.

###
### Retrieve the data from TAB1
###

SQL> select OWNER, table_name from tab1 where rownum < 3
  2  ;

OWNER
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$

SYS
ICOL$


###
### When the wallet is closed or unavailable,  same SQL does not retrieve
data
### To simulate wallet absence rename the cwallet.sso wallet under
<wallet_root>/tde to some other name like cwallet.sso.orig
### and close the wallet. To return auto-login wallet , rename file back
to  cwallet.sso and run the SQL again.
###

SQL> show parameter wallet_root

NAME                                 TYPE    VALUE
------------------------------------ -----------
------------------------------
wallet_root                          string  /etc/OKV/asmtde


SQL> ! mv /etc/OKV/asmtde/tde/cwallet.sso /etc/OKV/asmtde/tde/
cwallet.sso.orig

SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE ;

SQL> select OWNER, table_name from tab1 where rownum < 3;
select OWNER, table_name from tab1 where rownum < 3
                               *
ERROR at line 1:
```

```
ORA-28365: wallet is not open


SQL> ! mv /etc/OKV/asmtde/tde/cwallet.sso.orig /etc/OKV/asmtde/tde/
cwallet.sso

SQL> select OWNER, table_name from tab1 where rownum < 3
  2  ;

OWNER
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$

SYS
ICOL$
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Migrating Tablespaces from Non-TDE Oracle ACFS Database to Oracle ACFS CDB Database on Oracle Database Appliance

Understand how you can migrate tablespaces from non-TDE Oracle ACFS database to Oracle ACFS CDB database with Oracle Key Vault enabled on Oracle Database Appliance.

Ensure that there is a source and destination database. The two databases can be on one Oracle Database Appliance server or on different servers. The source database must be an non TDE-enabled database, either an Oracle Database Appliance database or any Oracle database with compatible Endian. The destination database must be an Oracle Database Appliance database created with ODACLI commands and with Oracle Key Vault enabled on it.

Step 1: Export the tablespace from the source database

1. Create a tablespace TBS01 and table TAB1 on source database DBNOTDE Create an extra tablespace TBS01 on the source database, and then, create table TAB1 on the new tablespace.

```
###
### login as Oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> select TABLESPACE_NAME, ENCRYPTED from dba_tablespaces ;

 TABLESPACE_NAME           STATUS    ENC
------------------------------ --------- ---
SYSTEM                 ONLINE    NO
SYSAUX                 ONLINE    NO
```

```
UNDOTBS1                ONLINE    NO
TEMP                    ONLINE    NO
USERS                   ONLINE    NO

SQL> create tablespace TBS01 datafile size 10g ;
Tablespace created.

SQL>
SQL> create table tab1 tablespace TBS01 as select a.* from dba_tables
a ,dba_users b, dba_users c ;
Table created.

SQL> select count(*) from tab1 ;

  COUNT(*)
----------
   3028228

SQL>


SQL> select TABLESPACE_NAME, ENCRYPTED from dba_tablespaces ;

 TABLESPACE_NAME           STATUS    ENC
------------------------------ --------- ---
SYSTEM                  ONLINE    NO
SYSAUX                  ONLINE    NO
UNDOTBS1                ONLINE    NO
TEMP                    ONLINE    NO
USERS                   ONLINE    NO
TBS01                   ONLINE    NO


SQL> select FILE_NAME from dba_data_files where TABLESPACE_NAME='TBS01' ;

FILE_NAME
--------------------------------------------------------------------------------
-----
/u02/app/oracle/oradata/dbnotde/DBNOTDE/datafile/o1_mf_tbs01_lnlgn65j_.dbf

SQL>


###
### Checks if a set of tablespaces (to be transported) is self-contained
###

SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('tbs01',true) ;

PL/SQL procedure successfully completed.
```

2. Create the database directory in the source database DBNOTDE to be used for the export. Create a database directory object in the destination database to be used by `expdp`.

```
###
### login as oracle user
```

```
###

$ mkdir -pv /u01/app/odaorabase0/oracle/dpdump/ttsps_dir
mkdir: created directory '/u01/app/odaorabase0/oracle/dpdump/ttsps_dir'

###
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.

SQL>
```

3. Export the TBS01 tablespace from souce database DBNOTDE. First, set the tablespace to be copied in READ ONLY mode. Next, run `expdp` on the designated tablespace. Review the log files for `expdp` stored under the database directory `ttsps_dir`.

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###


SQL> ALTER TABLESPACE TBS01 READ ONLY;

Tablespace altered.


###
### Run expdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$ expdp system TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES
DIRECTORY=ttsps_dir DUMPFILE=expttps.dmp

Export: Release 19.0.0.0.0 - Production on Mon Nov 6 12:47:00 2023
Version 19.25.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01":  system/********
TRANSPORT_TABLESPACES=TBS01 TRANSPORT_FULL_CHECK=YES DIRECTORY=ttsps_dir
DUMPFILE=expttps.dmp
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
```

```
Processing object type TRANSPORTABLE_EXPORT/TABLE
Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
******************************************************************************
***
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /u01/app/odaorabase0/oracle/dpdump/ttsps_dir/expttps.dmp
******************************************************************************
***
Datafiles required for transportable tablespace TBS01:
  /u02/app/oracle/oradata/dbnotde/DBNOTDE/datafile/
o1_mf_tbs01_lnlgn65j_.dbf
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Mon
Nov 6 12:47:19 2023 elapsed 0 00:00:14
```

4. Move data files to the destination database server and edit the status of the data file in the source database DBNOTDE to `READ WRITE`. Move the data file to the TDE-enabled database server in the PDB2 data files location.

```
###
### login as oracle user
###

$ cp /u02/app/oracle/oradata/dbnotde/DBNOTDE/datafile/
o1_mf_tbs01_lnlgn65j_.dbf /u02/app/oracle/oradata/cdbtde/CDBTDE/
092546916A493FE5E063E730850AF5BE/datafile/

###
### In source database, set the tablespace back to read write mode
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> ALTER TABLESPACE TBS01 READ WRITE;
```

Step 2: Import the tablespaces into a TDE-enabled database

1. Create a TDE-enabled database with Oracle Key Vault, CDBTDE as the destination database. Copy the data file and tablespace of the source database to this database using `expdp` and `impdp`. Create the database on an Oracle Database Appliance using the Browser User Interface (BUI) or ODACLI commands. For example:

```
# odacli create-database -n okvdb2 -t -kt okv -osc adminobj1 -dh
e2a65596-6435-4410-9e8d-6b21a6005779 -r ACFS --cdb -d pdbadmin -p pdb2

# odacli describe-database -n cdbtde
Database details
----------------------------------------------------------------
                     ID: c1ba7c6a-2808-4a20-bf23-e35cb62c3fd0
            Description: cdbtde
                DB Name: cdbtde
                DB Type: SI
                    CDB: true
               PDB Name: PDB2
    PDB Admin User Name: pdbadmin
```

```
                Storage: ACFS
                Home ID: e2a65596-6435-4410-9e8d-6b21a6005779
  TDE Wallet Management: ODA
            TDE Enabled: true
          KeystoreType : OKV

###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
### Take a note of PDB2 GUID which helps to determine the path of PDB2
datafiles
###

SQL> col PDB_NAME for a20
SQL> select  PDB_NAME,GUID from  dba_pdbs  ;

PDB_NAME         GUID
-------------------- --------------------------------
PDB2             092546916A493FE5E063E730850AF5BE
PDB$SEED         15998A6665985353E063BE6B1F0A494A
```

2. Create tnsnames service for PDB2.

```
###
### Edit the file  $ORACLE_HOME/network/admin/tnsnames.ora  with the
following content
###

PDB2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <server-name>)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb2.test)
```

3. Create directory in PDB2 for the import.

```
###
### login as oracle user
###

$ mkdir -pv /u01/app/odaorabase0/oracle/dpdump/ttsps_dir
mkdir: created directory '/u01/app/odaorabase0/oracle/dpdump/ttsps_dir'

###
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###


SQL> ALTER SESSION SET CONTAINER=PDB2 ;
SQL> CREATE DIRECTORY ttsps_dir AS '/u01/app/odaorabase0/oracle/dpdump/
ttsps_dir';

Directory created.
```

```
SQL>
```

4. Import the TBS01 tablespace to the destination database PDB2 container on CDBTDE. Copy the `expdp` dump file of the source database to the database directory `ttsps_dir`. The `impdp` uses the dump file to add tablespace metadata to destination database. TRANSPORT_DATAFILES references the location of the data file in the destination database.

```
###
### Run impdb as SYSTEM user, password for SYSTEM need to be entered when
prompted
###

$ impdp system@pdb2  TRANSPORT_DATAFILES=\'/u02/app/oracle/oradata/cdbtde/
CDBTDE/092546916A493FE5E063E730850AF5BE/datafile/
o1_mf_tbs01_lnlgn65j_.dbf\' DIRECTORY=ttsps_dir dumpfile=exp1.dmp

Import: Release 19.0.0.0.0 - Production on Thu Nov 9 21:42:25 2023
Version 19.25.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/
unloaded
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01":  system/********@pdb2
TRANSPORT_DATAFILES='/u02/app/oracle/oradata/cdbtde/CDBTDE/
092546916A493FE5E063E730850AF5BE/datafile/o1_mf_tbs01_lnlgn65j_.dbf'
DIRECTORY=ttsps_dir dumpfile=exp1.dmp
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/STATISTICS/MARKER
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Thu
Nov 9 21:42:52 2023 elapsed 0 00:00:19
```

5. Enable and test encryption on the destination Oracle Key Vault-enabled database PDB1.

```
###
### login as oracle user
### Connect to database using SQLPLUS as sysdba user ( sqlplus / as
sysdba ) and run the following
###

SQL> alter session set container=pdb2 ;

Session altered.

SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME            STATUS    ENC
```

```
------------------------------ --------- ---
SYSTEM                         ONLINE    NO
SYSAUX                         ONLINE    NO
UNDOTBS1                       ONLINE    NO
TEMP                           ONLINE    NO
USERS                          ONLINE    NO
TBS01                          READ ONLY NO


SQL> alter tablespace TBS01 read write ;

Tablespace altered.

SQL>  alter tablespace TBS01 encryption online encrypt ;

Tablespace altered.

SQL> select TABLESPACE_NAME, STATUS , ENCRYPTED from dba_tablespaces ;

TABLESPACE_NAME                STATUS    ENC
------------------------------ --------- ---
SYSTEM                         ONLINE    NO
SYSAUX                         ONLINE    NO
UNDOTBS1                       ONLINE    NO
TEMP                           ONLINE    NO
USERS                          ONLINE    NO
TBS01                          ONLINE    YES


SQL> select FILE_NAME from dba_data_files where TABLESPACE_NAME='TBS01' ;

FILE_NAME
--------------------------------------------------------------------------------
-----
/u02/app/oracle/oradata/cdbtde/CDBTDE/092546916A493FE5E063E730850AF5BE/
datafile/
o1_mf_tbs01_lnx8h50f_.dbf


###
### Retrieve the data from TAB1
###

SQL> select OWNER, table_name from tab1 where rownum < 3

OWNER
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$

SYS
ICOL$
```

```
###
### When the wallet is closed or unavailable,  same SQL does not retrieve
data
### To simulate wallet absence rename the cwallet.sso wallet under
<wallet_root>/tde to some other name like cwallet.sso.orig
### and close the wallet. To return auto-login wallet , rename file back
to  cwallet.sso and run the SQL again.
###

SQL> show parameter wallet_root

NAME                     TYPE    VALUE
------------------------------------ -----------
------------------------------
wallet_root              string  /etc/OKV/cdbtde


SQL> ! mv /etc/OKV/cdbtde/tde/cwallet.sso /etc/OKV/cdbtde/tde/
cwallet.sso.orig

SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE ;

SQL> select OWNER, table_name from tab1 where rownum < 3;
select OWNER, table_name from tab1 where rownum < 3
                              *
ERROR at line 1:
ORA-28365: wallet is not open


SQL> ! mv /etc/OKV/cdbtde/tde/cwallet.sso.orig /etc/OKV/cdbtde/tde/
cwallet.sso

SQL> select OWNER, table_name from tab1 where rownum < 3
  2  ;

OWNER
--------------------------------------------------------------------------------
-----
TABLE_NAME
--------------------------------------------------------------------------------
-----
SYS
TS$

SYS
ICOL$
```

Related Topics

- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.

Working with Databases

Use the Browser User Interface to display a list of databases, database details, and create and delete databases. You can use CLI commands to manage your databases.

> **Note:**
>
> Oracle Database Appliance enables unified auditing for databases created in new database homes. Unified audit trail captures audit information and places them in one location and in one format. This consolidated view enables auditors to co-relate audit information from different components. Having a single audit trail also improves management and security of the audit trail. For more information about unified audit trail for Oracle Database, see *Oracle Database Security Guide*.

- Viewing Databases
  Use the Oracle Appliance Manager Browser User Interface to display a list of Oracle databases and database details, create, upgrade, and delete a database in Oracle Database Appliance.
- Creating a Database Using the Browser User Interface
  Use the Oracle Appliance Manager Browser User Interface to create a database in Oracle Database Appliance.
- Managing Database Homes on Oracle ACFS Storage
  Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to manage database home storage in Oracle Database Appliance.
- Creating a Database Using ODACLI Commands
  Create a database from the command-line interface.
- Cloning a Database from Backup
  Use the Browser User Interface to clone a database from a backup.
- Cloning an Oracle ACFS Database
  Create a database from an existing Oracle ACFS database using ODACLI commands or BUI.
- Modifying a Database
  Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to modify a database in Oracle Database Appliance.
- Moving a Database from One Oracle Home to Another
  Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to move a database from one Oracle home to another of the same database version.
- Upgrading a Database
  Use the Oracle Appliance Manager Browser User Interface to upgrade an Oracle database to a different database home version.
- Upgrading a Database Manually Using AutoUpgrade
  Use AutoUpgrade to manually upgrade an earlier release of Oracle database on Oracle Database Appliance.
- Deleting a Database
  Use the Oracle Appliance Manager Browser User Interface to delete an Oracle database.

Viewing Databases

Use the Oracle Appliance Manager Browser User Interface to display a list of Oracle databases and database details, create, upgrade, and delete a database in Oracle Database Appliance.

1. Log into the Browser User Interface:

   `https://host name or ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.
3. (Optional) Click the database name, in blue font, to display more details about the database.
4. (Optional) Click **Actions** next to a database entry to view more details, upgrade or delete the database.

Creating a Database Using the Browser User Interface

Use the Oracle Appliance Manager Browser User Interface to create a database in Oracle Database Appliance.

Ensure that the repository is updated with Oracle RDBMS Clone files for a database version, before creating the database as described in *Updating Oracle Database Appliance Repository with Database Clone Files Using the CLI*.

> **Note:**
>
> For Standard Edition Oracle Database 19c or later, you cannot create Oracle RAC Or Oracle RAC One Node Database. You can only create single-instance Oracle Database. For Standard Edition Oracle Database 19.6 or later, you can choose to enable high availability for single-instance database. For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can choose to enable high availability for single-instance databases.

The Browser User Interface provides a quick and easy method of creating new databases. The Create New Database page in the Browser User Interface is populated with default options for most of the configuration options. Drop-down lists enable you to quickly view and select from a list of available options. You can use the drop-down list to create a new database Oracle Database Home (ORACLE_HOME) for the database or select an existing ORACLE_HOME that you created earlier.

Oracle Database 19.25 is supported on both Oracle Automatic Storage Management (Oracle ASM) and Oracle ASM Cluster file system (Oracle ACFS). When databases are created in Oracle ACFS, each database is configured with its own Oracle ACFS file system for the datafiles and uses the following naming convention: `/u02/app/db user/oradata/db unique name`. The default size of this mount point is 100 GB.

The fields in the Browser User Interface adjust, depending on the database version you select. Follow these steps to create a database:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.
3. Click **Create Database** to display the Create Database page.
4. Select **Create Database** on the Create Database page.
5. Enter the following information on the Database Information page to configure the database:
    a. In the **DB Name** field, enter a name for the database. The name *db1* appears in the field as an example of a database name, but the field is not populated. You must enter a name.

       The name must contain lowercase alphanumeric characters and cannot exceed 8 characters. The Oracle system identifier (SID) is always set to the database name.
    b. (Optional) In the **DB Unique Name** field, enter a name to define a unique name for the database.
    c. In the **Use Existing DB Home** field, select **Yes** or **No**.
    d. In the **DB Version** field, select a database bundle patch number from the drop-down list.

       Available database versions with available clones appear in the list in descending order. Supported versions that are not available in the repository appear in light gray font at the bottom of the list. Search is available for the list. To add a supported version to the repository, download the RDBMS clone file and update the repository.
    e. For the **CDB** option on Oracle Database releases later than 12.1, select **Yes** or **No**, depending on whether or not you want the database to be a container database (CDB). The default is **Yes**.
    f. In the **PDB Name** field, enter a name for the pluggable database (PDB).

       Alphanumeric characters and underscore (_) are valid. The name must begin with an alphanumeric character and cannot exceed 30 characters. The name *pdb1* appears in the field in italic text as an example, but it is not populated. You must enter a name.
    g. In the **PDB Admin User** field, enter a name.

       The field shows *pdbadmin* as an example, you must enter a name. Alphanumeric characters and underscore (_) are valid.
    h. In the Database Edition field, enter the edition for which you have a license, either **Enterprise Edition** or **Standard Edition**.

       For Standard Edition Oracle Database 19c or later, you can only create single-instance Oracle Database. For Standard Edition Oracle Database 19.6 or later, you can choose to enable high availability for single-instance database. For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can choose to enable high availability for single-instance databases.
    i. For single-instance Oracle Database deployment, specify the **Node**, either Node0 or Node1. The default is Node0. Specify whether you want to **Enable High Availability**.
    j. In the **Shape** field, select a database shape from the drop-down list. The shape determines the number of cores and total memory allocated to the database. The default is `odb1`.
    k. In the **Database Class** field, select a database class from the drop-down list. If an option is not available in the list, it is not supported for the database edition on the Oracle Database Appliance or the version that you selected. The default is OLTP.
    l. In the **Storage** field, select **ACFS** or **ASM** from the drop-down list. The default is Oracle ASM.

m. If the disk group redundancy was set to **Flex** during the appliance creation, then you can select the **Database Redundancy** value as **Mirror** or **High**.

n. Select the **Networks** you want to associate with the database.

o. Select the **CPU Pool Name** to associate with the database.

p. For the **Configure EM Express** or **Configure EM Console** option, select **Yes** or **No**. The default is **No**.

   Select **Yes** to configure the Oracle Enterprise Manager Database Express (EM Express) console for Oracle Database 19.25. Selecting **Yes** enables you to use the console to manage the database.

q. In the **SYS and PDB Admin User Password** field, enter the password for the `SYS`, `SYSTEM`, and `PDB Admin`.

   The password must begin with an alpha character and cannot exceed 30 characters. Quotation marks are not allowed.

r. In the **Confirm Password** field, enter the password again to confirm.

s. In the **Characterset** field, select an option from the drop-down list. The default is AL32UTF8.

t. In the **National Characterset** field, select an option from the drop-down list. The default is AL16UTF16.

u. In the **Language** field, select a database language from the drop-down list. The default is American.

v. In the **Territory** field, select a territory or location for the database from the drop-down list. The default is America.

6. Click **Next**.

7. In the TDE Information page, for Oracle Database Enterprise Edition 19c or later, you can choose to enable Transparent Database Encryption (TDE). Select **Yes** or **No** in the **Enable TDE** option. Specify and confirm the **TDE Password**. By default, the TDE option is disabled.

8. Click **Create**. When prompted, click **Yes** to confirm that you want to start the job to create the database.

The job is submitted and a confirmation page appears with a link to the job. Click the link to view the job progress, tasks, and status.
After you close the Job confirmation page, you can click the **Activity** tab to monitor the job progress. Click the job number to view the tasks and status details. Click **Refresh** to refresh the page.

Managing Database Homes on Oracle ACFS Storage

Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to manage database home storage in Oracle Database Appliance.

Managing Database Home Storage Using the Browser User Interface

1. Log into the Browser User Interface:
   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.

3. Click Database Home Storage on the left pane.

4. If Database Home Storage is not configured, then click on the **Configure Database Home Storage** button.
5. Select the Disk Group Name and specify the Size in GB.
6. Click **Create**. When prompted, click **Yes** to confirm that you want to start the job to configure the database home storage.
7. You can also view a list of all Database Home storages from the Database tab and view details for each database.
8. To modify Database Home Storage, select **Modify** in the **Actions** drop down list.

Managing Database Home Storage Using ODACLI Commands

1. Use the `odacli configure-dbhome-storage` command to configure the database home storage:

   ```
   # odacli configure-dbhome-storage -dg DATA
   ```

2. Use the `odacli list-dbhome-storages` command to list the database home storages:

   ```
   # odacli list-dbhome-storages
   ```

3. Use the `odacli describe-dbhome-storages` command to describe a database home storage:

   ```
   odacli describe-dbhome-storage -i 67622ce6-0a00-4fec-b948-7a0ba7922311
   ```

   For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli configure-dbhome-storage
  Use the `odacli configure-dbhome-storage` command to configure the database home storage.
- odacli list-dbhome-storages
  Use the `odacli list-dbhome-storages` command to view the database home storage resources after the volumes for ORACLE_HOME and ORACLE_BASE are created.
- odacli describe-dbhome-storage
  Use the `odacli describe-dbhome-storage` command to view information such as how much space is available (at a volume or diskgroup level).
- odacli modify-dbhome-storage
  Use the `odacli modify-dbhome-storage` command to modify database home storage and add additional disk space.

Creating a Database Using ODACLI Commands

Create a database from the command-line interface.

> **Note:**
>
> If you provisioned the appliance without creating an initial database, then you must create a Oracle home. If the version of the database home is different from the migrated database, create a database home for the migrated database. You might want to create a database home specifically for the migrated database.

> **Caution:**
>
> When you enable Transparent Data Encryption (TDE) on databases, with software keystore or Oracle Key Vault keystore with endpoints in Oracle Key Vault, you are prompted to set a password for the TDE wallet. Provide a strong password for security compliance. Enter this password carefully when setting it for the first time, and ensure that this password is available to you at all times for database management operations. Failure to provide the TDE wallet password when prompted, will cause an error in accessing encrypted data.

This example creates a new database named `PRODDB` with database version 19.25, and a new database home, if a database home does not exist.

1. Use the `odacli list-dbhomes` command to display a list of database homes and verify that a database home exists for the database version.

   Note the ID if you want to use an existing database home or if you want to get more details. Use the `odacli describe-dbhomes` command to get more details about a specific database home.

   ```
   # odacli list-dbhomes
   ID                                       Name                 DB Version
   ---------------------------------------- -----------------    ----------
   b727bf80-c99e-4846-ac1f-28a81a725df6 OraDB199_home1  19.25.0.0.0

   (continued)
   Home Location
   ----------------------------------------
   /u01/app/orauser/product/19.0.0.0/dbhome_1
   ```

2. Create a database home if a database home does not exist for the database version.

   If you want to create a database home for specifically for the migrated database, you can use the `odacli create-dbhome` command, or you can create the database home when you create the database. The example creates a database home for database version 19.25.0.0.0.

   ```
   # odacli  create-dbhome -v 19.25.0.0.0
   ```

3. Create a database. You can use an existing database home ID or you can create a database home when creating the database.

To create a database named `PRODDB` that uses an existing database home with ID `b727bf80-c99e-4846-ac1f-28a81a725df6`:

```
# odacli create-database -n PRODDB -io -dh b727bf80-c99e-4846-
ac1f-28a81a725df6
```

To create a database named `PRODDB` with database version `19.25.0.0.0` and a database home:

```
# odacli create-database -n PRODDB -v 19.25.0.0.0 -io
```

To create a TDE-enabled database named `PRODDB`:

```
# odacli create-database -n PRODDB -t
```

Cloning a Database from Backup

Use the Browser User Interface to clone a database from a backup.

When you backup a database in Oracle Database Appliance, a backup report is created. Save the backup report as a JSON file and you can use the file to create a database on the same appliance, or a different appliance.

The following are the prerequisites to cloning a database from a backup:

1. The source database backup location must be Object Store or External FRA (Network File System).
2. If you use Object Store backup location, then obtain Object Store Credentials for the host.
3. If you use an External FRA (Network File System) backup location, then obtain the Network File System (NFS) mount point location.
4. Create a backup policy using the object store credentials or NFS mount point as the backup destination.
5. Attach the backup policy to the database that you want to backup. If the source database does not have TDE enabled, then providing the Backup Encryption password is mandatory when attaching Objectstore backup policy. However, the Backup Encryption password is optional when attaching the NFS backup policy. If the source database has TDE enabled, then you need not specify the Backup Encryption password, irrespective of backup destination.
6. Create a manual backup of the database and save the backup report generated when the backup completes.

Follow these steps to create a database from backup:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.
3. Click **Create Database** to display the Create Database page.
4. Click **Clone Database from Backup**, then click **Next** to display the Clone Database from Backup page.
5. Select the Backup Destination from which you want to create the database. If your backup destination is Object Store:

a. Select Backup Destination as **ObjectStore**.

b. Select your **Object Store Credential Name**.

c. Enter the password in the **Backup Encryption Passwords** field and the **Confirm Backup Encryption Passwords** field.

If your backup destination is Network File System (NFS):

a. Select Backup Destination as **External FRA**.

b. Enter the password in the **Backup Encryption Passwords** field and the **Confirm Backup Encryption Passwords** field.

6. Click **Browse** and select the backup report from which you want to create the database.

   When the backup report is loaded, Database Information and TDE Information steps are displayed on different pages and populated based on the backup report. You can edit some of the fields.

   For Standard Edition Oracle Database 19c or later, you cannot clone Oracle RAC or Oracle RAC One Node Database. You can only clone a single-instance Oracle Database. For Standard Edition Oracle Database 19.6 or later, you can choose to enable high-availability for single-instance database. For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can choose to enable high availability for single-instance databases.

7. In the **DB Name** field, enter a name for the database.

   The name must contain lowercase alphanumeric characters and cannot exceed 8 characters. The Oracle system identifier (SID) is always set to the database name.

8. Enter the password in the **SYS and PDB Admin User Password** field and the **Confirm Password** field.

9. Specify the Networks.

10. If your source database has Transparent Database Encryption (TDE) enabled, then you can enable TDE on the cloned database. If the source database has TDE enabled, then the backup report has the TDE wallet backup location and the **TDE Wallet Backup Location** field in the BUI displays this value. Specify and confirm the **TDE Password**.

11. You can choose an Oracle Key Vault server configuration from the **Select OKV Server Config** drop-down list, and specify the **OKV Wallet Location**.

12. Click **Create**.

13. Click **Yes** to confirm that you want to clone a database from the selected Object Store or External FRA.

    When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.

14. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

    ```
    # odacli describe-job -i jobId
    ```

Cloning an Oracle ACFS Database

Create a database from an existing Oracle ACFS database using ODACLI commands or BUI.

Cloning enables to:

- Create a database from another database without bringing down the source database

- Create multiple databases from a gold image, thus ensuring standardized mass deployments
- Optimize space utilization, by use of Oracle ACFS snapshots in cloning
- Create different types of databases, from a single source database type. For example, you can create single-instance databases, Oracle RAC databases, or Oracle RAC One Node databases from any type of source database
- Depending on the available memory and CPU cores, specify a supported database shape such as `odb1s`, `odb2`, and so on, to create any class of database from any source database.
- Create clone databases on a standby system. Since standby systems are not production systems, they are a better choice for seeding test or development environments.
- Run clone database from a database home that is different from that of the source database.
- Create clone database from a read-only source database.
- Create read-only clone database from any source database.

Following are the prerequisites to clone a database:

- Ensure that Oracle Clusterware is running on all nodes, and the source database is up and running.
- The source database must use Oracle ACFS storage.
- The source database must not be a multitenant container database (CDB).
- The source database must be opened in either read-write mode, or read-only mode if it is a primary database. A standby database can be in mounted or read-only open mode.
- The source database must not be in backup mode.
- The source database must be in archive mode.
- All data files of the source database must be in the same Oracle ACFS directory.

> **Note:**
>
> For Standard Edition Oracle Database 19c or later, you cannot clone Oracle RAC or Oracle RAC One Node Database. You can only clone a single-instance Oracle Database. For Standard Edition Oracle Database 19.6 or later, you can choose to enable high-availability for single-instance database.

Cloning an Oracle ACFS Database Using Command Line Interface

1. Run the `odacli clone-database` command.

```
odacli clone-database --databaseUniqueName --dbname --dbshape --dbtype --
sourcedbname --associated-networks --cpupool --dbhomeid --disable-ha --
enable-ha --newhome --read-only --tdepassword--json
```

For cloning, you can create a new database home or select an existing database home. To create a new database home, specify the `--newhome` option. To use an existing database home, specify the `--dbhomeid`. Specify the unique name of the database, the name for the new database, the database shape, the type of database, and the source database name.

For TDE-enabled database, also specify the `--tdepassword` option. The TDE password of the new database must be the same as the TDE password of the source database.

For information about all the command options, see the *ODACLI Command Reference* chapter.

Cloning an Oracle ACFS Database Using the Browser User Interface

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.
3. Click **Create Database** to display the Create Database page.
4. Click **Clone a Database**, then click **Next** to display the Clone Database page.
5. In the Clone Database page, you can create a new database home or select an existing database home.
6. To create a new database home, select the **Create New DB Home** and then provide the values as described in this procedure.
7. To specify an existing database home, select the **Source Database Name** from which you want to create the database and the **DB Home Name**.

   For Oracle Database 19c Standard Edition, you can only create single-instance Oracle Database, with or without high-availability. For Standard Edition Oracle Database 19.6 or later, you can choose to enable high-availability for single-instance database. For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can choose to enable high availability for single-instance databases.
8. Select the **Database Shape** (template) for your new database. The database shape you select determines the total memory allocated to the database.
9. In the **DB Name** field, enter a name for the database.

   The name must contain lowercase alphanumeric characters and cannot exceed 8 characters. The Oracle system identifier (SID) is always set to the database name.
10. Specify the **DB Unique Name** for the database. If the database unique name is not provided, then the name of the database is set to the database name.
11. Specify the Networks.
12. Enter the password in the **SYS and PDB Admin User Password** field and the **Confirm Password** field. For Transparent Database Encryption (TDE) enabled database, specify the TDE wallet password. The TDE password is the same as the TDE password of the source database.
13. Click **Create**.
14. Click **Yes** to confirm that you want to clone a database from the selected source database.

    When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
15. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

    ```
    # odacli describe-job -i jobId
    ```

Related Topics

- odacli clone-database
  Use the `odacli clone-database` command to clone a new database from a source database.

Modifying a Database

Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to modify a database in Oracle Database Appliance.

Modifying a Database Using the Browser User Interface

1. Log into the Browser User Interface:
   `https://host-ip-address:7093/mgmt/index.html`
2. Click the **Database** tab.
3. For the database that you want to modify, select **Modify** in the Actions drop down list.
4. In the Modify Database page, you can attach or detach associated networks and backup policy, and modify database shape and class. You can also change TDE wallet management from `EXTERNAL` to `ODA` for Oracle Database 19c and later databases.
5. To modify the **Database Class** or **Database Shape**, select the new value in the drop down list.
6. To attach or detach networks, modify the values in the **Attach Networks** and **Detach Networks** fields.
7. To remove the associated backup policy from the database, do not select any value in the **Select Back up Policy** drop down list.
8. If you configured Oracle Flex disk group when you created the database, you can also modify the database redundancy. If the database you are modifying is an Oracle ACFS Database, then the database redundancy for all cloned Oracle ACFS databases is modified.
9. You can choose to change the **Backup Encryption Password**. If the database has TDE enabled, then the RMAN Backup Encryption password is not used.
10. Note that if the database uses Oracle Key Vault server configuration, then you cannot modify any TDE options.
11. Select the **Specify TDE Option** to perform these operations:
    - Select **Re-key TDE Master Encryption Key** to re-key the TDE master encryption key of the database. Specify the **Current TDE Wallet** password.
    - Select **Change Password for TDE Wallet** and specify the **Current TDE Password** and the **New TDE Password**.
    - If the TDE wallet management is set to `EXTERNAL`, then you can view and select the **Change TDE Wallet Management** option, and set it to `ODA`.
12. For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can choose to **Enable High Availability** for single-instance databases.
13. For Standard Edition Oracle Database 19.6 or later, you can choose to **Enable High Availability** for single-instance databases.
14. Click **Modify**. When prompted, click **Yes** to confirm that you want to start the job to modify the database.

Modifying a Database Using ODACLI Commands

1. Use the `odacli modify-database` command to modify the configuration of a database, such as backup configuration, database class, database type, TDE key, or to change the TDE wallet management from `EXTERNAL` to `ODA`.

   ```
   # odacli modify-database -s database_shape  -cl database_class -i dbid
   ```

   For example:

   ```
   # odacli modify-database -i 1941d594-c777-4eca-9fce-18b778d5c153 -s odb2 -
   cl DSS
   ```

   For example, the following command re-keys the TDE master encryption key of the database after accepting the current TDE Wallet password.

   ```
   # odacli modify-database -n testdb -rkt
   Enter TDE wallet password:
   ```

   For more information about the `odacli modify-database` command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli modify-database
  Use the `odacli modify-database` command to modify the configuration of a database, such as backup configuration, database class, and database type, and to change TDE wallet management from `EXTERNAL` to `ODA`.

Moving a Database from One Oracle Home to Another

Use the Oracle Appliance Manager Browser User Interface or ODACLI commands to move a database from one Oracle home to another of the same database version.

Moving a Database Using the Browser User Interface

1. Log into the Browser User Interface:
   `https://host-ip-address:7093/mgmt/index.html`
2. Click the **Database** tab.
3. For the database that you want to modify, select **Move** in the Actions drop down list.
4. Select the **Destination Database Home** where you want to move the database. The destination database home must be of the same base version. Select **Ignore Missing Patches** to ignore missing patches. For high-availability environment, you can select the **Non-Rolling** check box to perform non-rolling move of the database home.
5. Click **Move**. When prompted, click **Yes** to confirm that you want to start the job to move the database.

Moving a Database Using ODACLI Commands

1. Use the `odacli move-database` command to move a database from one Oracle home to another home of the same database version.

   ```
   # odacli move-database -i database_ID -dh destination_database_home_ID
   ```

   For more information about the `odacli move-database` command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli move-database
  Use the command `odacli move-database` to move a database from one Oracle home to another home of the same database version.

Upgrading a Database

Use the Oracle Appliance Manager Browser User Interface to upgrade an Oracle database to a different database home version.

Before upgrading to a different database home, you must upload the Oracle RDBMS clone files to the repository and create the database home.

> **Note:**
>
> You cannot upgrade Oracle RAC or Oracle RAC One Node Database to a destination database home of Standard Edition Oracle Database 19c or later. You must first convert Oracle RAC Or Oracle RAC One Node Database to single-instance Oracle Database using the `odacli modify-database` command and then upgrade the single-instance Oracle Database to a destination database home of Standard Edition 19c or later.

1. Log into the Browser User Interface:

   ```
   https://host name or ip-address:7093/mgmt/index.html
   ```

2. Click the **Database** tab.
3. Click **Actions** next to the database that you want, then select **Upgrade**.
4. Select the destination database home version from the list of available versions for the upgrade.

   When upgrading single-instance Standard Edition Oracle Database 18.9 or later to Standard Edition Oracle Database 19c or later, you can also choose to enable or disable High Availability.

   When upgrading single-instance Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can also choose to enable or disable High Availability.
5. You can also change the TDE wallet management option if TDE is enabled for the source database and the TDE Wallet Management is set to **External**. Select **Change TDE Wallet Management** to set the TDE Wallet Management to **ODA**.
6. Click **Upgrade**.

Upgrading a Database Manually Using AutoUpgrade

Use AutoUpgrade to manually upgrade an earlier release of Oracle database on Oracle Database Appliance.

Starting with Oracle Database Appliance release 19.17, you cannot manage releases of Oracle Database earlier than 19c using ODACLI or BUI. To upgrade Oracle databases of release 12.1, 12.2, and 18c on Oracle Database Appliance, use OPatch to patch the databases or use Oracle Database Upgrade Assistant (DBUA) or AutoUpgrade to upgrade these databases manually with Oracle Database Appliance 19.17 or later.

Ensure the following:

- The Oracle Database Appliance server must be on Oracle Database Appliance release 19.19 or later.
- The repository must be updated with the target database clone version on the source and target Oracle Database Appliance systems.

1. Create the destination DB home for the database:

```
#odacli create-dbhome -v 19.18.0.0.230117

[root@oda1 ~]# odacli list-dbhomes
ID                                       Name                 DB
Version                     DB Edition Home
Location                                             Status
---------------------------------------- --------------------
------------------------------ ----------
------------------------------------------------------------ ----------
f90adcc1-f64a-41ce-b72d-154db155b1fa     OraDB19000_home1
19.18.0.0.230117               EE         /u01/app/odaorahome/oracle/
product/19.0.0.0/dbhome_1    CONFIGURED
562a7428-9ea7-4878-9005-62c9d732a12b     OraDB12102_home1
12.1.0.2.220719                EE         /u01/app/odaorahome/oracle/
product/12.1.0.2/dbhome_1    CONFIGURED
```

2. Disable SSHCleanerJob and configure SSH user equivalence between both nodes in case of Oracle Database Appliance High-Availability systems.

```
[root@oda1 ~]# odacli list-schedules|grep "Name\|SSH"
ID                                       Name
Description
CronExpression                 Disabled
44ad4fe2-4893-4c7d-a61c-15845cb74aa5     SSHCleanerJob             SSH
cleaner job to clean up stale SSH keys         0 0/30 * 1/1 * ?
*             false

[root@oda1 ~]# odacli modify-schedule -i 44ad4fe2-4893-4c7d-
a61c-15845cb74aa5 -d
Modify job schedule success

[root@oda1 ~]# odacli list-schedules|grep "Name\|SSH"
ID                                       Name
Description
CronExpression                 Disabled
44ad4fe2-4893-4c7d-a61c-15845cb74aa5     SSHCleanerJob             SSH
```

```
cleaner job to clean up stale SSH keys         0 0/30 * 1/1 * ?
*             true

[oracle@oda1 ~]$ /u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1/
deinstall/sshUserSetup.sh -user oracle -hosts "oda1 oda2" -
noPromptPassphrase
```

3. Create the AutoUpgrade configuration file as the `oracle` operating system user on the first node.

```
[oracle@oda1 ~]$ cat autoupgrade.conf
global.autoupg_log_dir=/u01/app/odaorabase/oracle/autoupgrade
upg1.dbname=italy
upg1.start_time=NOW
upg1.source_home=/u01/app/odaorahome/oracle/product/12.1.0.2/dbhome_1
upg1.target_home=/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1
upg1.sid=italy11
upg1.log_dir=/u01/app/odaorabase/oracle/autoupgrade/italy
upg1.upgrade_node=localhost
upg1.target_version=19
upg1.run_utlrp=yes
upg1.timezone_upg=no
```

4. Change the Fast Recovery Area to Oracle ACFS if the database was created on Oracle ASM.

   a. Verify whether the dbstorage of the database is ACFS or ASM.

```
# odacli list-databases

ID                                       DB Name    DB Type  DB
Version           CDB        Class    Shape    Storage    Status
DbHomeID
---------------------------------------- ---------- --------
-------------------- ---------- -------- -------- ----------
------------ ----------------------------------------
5c183783-a198-413e-b96e-2c941fe1a895     italy      RAC
12.1.0.2.220719      true       OLTP     odb2     ACFS
CONFIGURED   562a7428-9ea7-4878-9005-62c9d732a12b
8f90d26d-c17a-45e3-abbc-67c981c24a3f     hun        RAC
19.18.0.0.230117     true    OLTP     EE       odb1     ASM
CONFIGURED   f90adcc1-f64a-41ce-b72d-154db155b1fa

[root@oda1 ~]# odacli list-dbhomes
ID                                       Name                 DB
Version                     DB Edition Home
Location                                             Status
---------------------------------------- --------------------
------------------------------ ----------
------------------------------------------------------------ ----------
f90adcc1-f64a-41ce-b72d-154db155b1fa     OraDB19000_home1
19.18.0.0.230117               EE         /u01/app/odaorahome/oracle/
product/19.0.0.0/dbhome_1    CONFIGURED
562a7428-9ea7-4878-9005-62c9d732a12b     OraDB12102_home1
12.1.0.2.220719                EE         /u01/app/odaorahome/oracle/
product/12.1.0.2/dbhome_1    CONFIGURED
```

```
[oracle@oda1 ~]$ export ORACLE_HOME=/u01/app/odaorahome/oracle/product/
12.1.0.2/dbhome_1
[oracle@oda1 ~]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@oda1 ~]$ export ORACLE_SID=italy1
[oracle@oda1 ~]$ sqlplus / as sysdba

SQL> show parameter db_recovery_file_dest

ACFS:

NAME                     TYPE     VALUE
------------------------------------ -----------
------------------------------
db_recovery_file_dest             string     /u03/app/oracle/
fast_recovery_area/
db_recovery_file_dest_size         big integer 53862M

ASM:

NAME                     TYPE     VALUE
------------------------------------ -----------
------------------------------
db_recovery_file_dest             string
+RECO(FG$FILEGROUP_TEMPLATE_MIRROR)
db_recovery_file_dest_size         big integer 200G
```

b. If the dbstorage is ASM, then create a vmstorage temporarily or use an existing one and create with a slightly larger size than `db_recovery_file_dest_size`.

```
[root@oda1 ~]# odacli create-vmstorage -n tempfra -r mirror -s 250G
[root@oda1 ~]# odacli list-vmstorages
Name                  Disk group       Volume name      Volume
device                   Size        Used        Used %
Available   Mount Point
Created                   Updated
--------------------  ---------------  ---------------
------------------------------  ----------  ----------  ----------
----------  ----------------------------------
------------------------  ------------------------
tempfra               DATA             TEMPFRA          /dev/asm/
tempfra-18             250.00 GB   1.09 GB     0.44%       248.91
GB   /u05/app/sharedrepo/tempfra          2023-04-18 19:05:55 CEST
2023-04-18 19:05:55 CEST
```

c. Create a folder on the vmstorage and change ownership of the folder to `oracle:dba`.

```
[root@oda1 ~]# mkdir /u05/app/sharedrepo/tempfra/italy
[root@oda1 ~]# chown oracle:dba /u05/app/sharedrepo/tempfra/italy
```

d. Change `db_recovery_file_dest` to ACFS.

```
SQL> alter system set db_recovery_file_dest='/u05/app/sharedrepo/
tempfra/italy/' scope=both;
```

5. Run prechecks and review the findings.

```
[oracle@oda1 ~]$ /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/jdk/bin/java -jar /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/rdbms/admin/autoupgrade.jar -config ~/autoupgrade.conf -mode
analyze
AutoUpgrade 22.4.220712 launched with default internal options
Processing config file ...
+--------------------------------+
| Starting AutoUpgrade execution |
+--------------------------------+
1 CDB(s) plus 2 PDB(s) will be analyzed
Type 'help' to list console commands
upg> Job 100 completed
------------------ Final Summary --------------------
Number of databases            [ 1 ]

Jobs finished                  [1]
Jobs failed                    [0]

Please check the summary report at:
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
status.html
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
status.log
```

6. Run AutoUpgrade in fixup mode to fix issues that could prevent a successful upgrade (if any).

```
[oracle@oda1 ~]$ /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/jdk/bin/java -jar /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/rdbms/admin/autoupgrade.jar -config ~/autoupgrade.conf -mode
fixups
AutoUpgrade 22.4.220712 launched with default internal options
Processing config file ...
+--------------------------------+
| Starting AutoUpgrade execution |
+--------------------------------+
1 CDB(s) plus 2 PDB(s) will be processed
Type 'help' to list console commands
upg> Job 101 completed
------------------ Final Summary --------------------
Number of databases            [ 1 ]

Jobs finished                  [1]
Jobs failed                    [0]

Please check the summary report at:
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
status.html
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
status.log
```

7. Re-run prechecks and review the findings.

```
[oracle@oda1 ~]$ /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/jdk/bin/java -jar /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/rdbms/admin/autoupgrade.jar -config ~/autoupgrade.conf -mode
analyze
AutoUpgrade 22.4.220712 launched with default internal options
Processing config file ...
+--------------------------------+
| Starting AutoUpgrade execution |
+--------------------------------+
1 CDB(s) plus 2 PDB(s) will be analyzed
Type 'help' to list console commands
upg> Job 102 completed
------------------- Final Summary --------------------
Number of databases            [ 1 ]

Jobs finished                  [1]
Jobs failed                    [0]

Please check the summary report at:
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
status.html
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
status.log
```

8. Upgrade the database.

```
[oracle@oda1 ~]$ /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/jdk/bin/java -jar /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/rdbms/admin/autoupgrade.jar -config ~/autoupgrade.conf -mode
deploy

AutoUpgrade 22.4.220712 launched with default internal options
Processing config file ...
+--------------------------------+
| Starting AutoUpgrade execution |
+--------------------------------+
1 CDB(s) plus 2 PDB(s) will be processed
Type 'help' to list console commands

upg> Job 103 completed
------------------- Final Summary --------------------
Number of databases            [ 1 ]

Jobs finished                  [1]
Jobs failed                    [0]
Jobs restored                  [0]
Jobs pending                   [0]

---- Drop GRP at your convenience once you consider it is no longer needed
----
Drop GRP from italy1: drop restore point AUTOUPGRADE_9212_ROME121020

Please check the summary report at:
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
```

```
status.html
/u01/app/odaorabase/oracle/autoupgrade/cfgtoollogs/upgrade/auto/status/
status.log

Note: once you confirmed that database was running as expected from all
perspective and downgrade surely would not be needed, drop the restore
point which was created by AutoUpgrade tool as the output also instructed

---- Drop GRP at your convenience once you consider it is no longer needed
----
Drop GRP from italy1: drop restore point AUTOUPGRADE_9212_ROME121020
```

9. Enable SSHCleaner job and remove SSH user equivalence in case of Oracle Database Appliance high-availability system.

```
[root@oda1 ~]# odacli list-schedules|grep "Name\|SSH"
ID                                       Name
Description
CronExpression                 Disabled
44ad4fe2-4893-4c7d-a61c-15845cb74aa5     SSHCleanerJob             SSH
cleaner job to clean up stale SSH keys         0 0/30 * 1/1 * ?
*             true

[root@oda1 ~]# odacli modify-schedule -i 44ad4fe2-4893-4c7d-
a61c-15845cb74aa5 -e
Modify job schedule success

[root@oda1 ~]# odacli list-schedules|grep "Name\|SSH"
ID                                       Name
Description
CronExpression                 Disabled
44ad4fe2-4893-4c7d-a61c-15845cb74aa5     SSHCleanerJob             SSH
cleaner job to clean up stale SSH keys         0 0/30 * 1/1 * ?
*             false

Remove local and remote node from /home/oracle/.ssh/authorized_keys files
on both nodes as oracle OS user using "vi" command

[oracle@oda1 ~]$ vi /home/oracle/.ssh/authorized_keys
[oracle@oda2 ~]$ vi /home/oracle/.ssh/authorized_keys

Remove all id* files under /home/oracle/.ssh as oracle OS user on both
nodes

[oracle@oda1 ~]$ rm /home/oracle/.ssh/id*
[oracle@oda2 ~]$ rm /home/oracle/.ssh/id*
```

10. Sync up the DCS metadata of the database.

```
[root@oda1 ~]# odacli list-dbhomes
ID                                       Name                 DB
Version                    DB Edition Home
Location                                              Status
---------------------------------------- --------------------
------------------------------ ----------
---------------------------------------------------------- ----------
```

```
f90adcc1-f64a-41ce-b72d-154db155b1fa     OraDB19000_home1
19.18.0.0.230117               EE         /u01/app/odaorahome/oracle/
product/19.0.0.0/dbhome_1    CONFIGURED
562a7428-9ea7-4878-9005-62c9d732a12b     OraDB12102_home1
12.1.0.2.220719                EE         /u01/app/odaorahome/oracle/
product/12.1.0.2/dbhome_1    CONFIGURED

[root@oda1 ~]# odacli list-databases
ID                                       DB Name    DB Type  DB
Version           CDB     Class    Edition  Shape    Storage  Status
DB Home ID
---------------------------------------- ---------- --------
------------------- ------- -------- -------- -------- --------
------------ ----------------------------------------
222a1d47-24ea-4a00-82f0-20d7fe17f59e     italy    RAC
12.1.0.2.220719      true    OLTP     EE       odb2     ACFS
CONFIGURED   562a7428-9ea7-4878-9005-62c9d732a12b

[root@oda1 ~]# odacli describe-dbstorage -n italy|grep DBUnique

          DBUnique Name: rome

[root@oda1 ~]# odacli update-registry -n db -u rome

Job details
----------------------------------------------------------------
                     ID:  fc54b821-c407-4174-8a1a-c90ba66e6cd2
            Description:  Discover Components : db
                 Status:  Created
                Created:  April 18, 2023 12:04:55 PM CEST
                Message:

[root@oda1 ~]# odacli describe-job -i fc54b821-c407-4174-8a1a-c90ba66e6cd2

Job details
----------------------------------------------------------------
                     ID:  fc54b821-c407-4174-8a1a-c90ba66e6cd2
            Description:  Discover Components : db
                 Status:  Success
                Created:  April 18, 2023 12:04:55 PM CEST
                Message:

Task Name                                Node Name                 Start
Time                          End Time                          Status
---------------------------------------- -------------------------
---------------------------------- ----------------------------------
----------
Discover DBHome                          oda1                April 18,
2023 12:05:01 PM CEST     April 18, 2023 12:05:04 PM CEST     Success
Discover DBHome                          oda1                April 18,
2023 12:05:04 PM CEST     April 18, 2023 12:05:07 PM CEST     Success
Discover DB: rome                       oda1                April 18, 2023
12:05:09 PM CEST     April 18, 2023 12:05:20 PM CEST     Success

[root@oda1 ~]# odacli list-databases
ID                                       DB Name    DB Type  DB
```

```
Version           CDB     Class    Edition  Shape    Storage  Status
DB Home ID
---------------------------------------- ---------- --------
-------------------- ------- -------- -------- -------- --------
------------ ----------------------------------------
222a1d47-24ea-4a00-82f0-20d7fe17f59e     italy    RAC
19.18.0.0.230117     true    OLTP     EE       odb2     ACFS
CONFIGURED   f90adcc1-f64a-41ce-b72d-154db155b1fa
```

11. If you edited db_recovery_file_dest in step 4, then revert the change.

```
[oracle@oda1 ~]$ export ORACLE_HOME=/u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_1
[oracle@oda1 ~]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@oda1 ~]$ export ORACLE_SID=italy1
[oracle@oda1 ~]$ sqlplus / as sysdba

SQL> alter system set
db_recovery_file_dest='+RECO(FG$FILEGROUP_TEMPLATE_MIRROR)' scope=both;
```

12. Take a full backup.

```
[oracle@oda1 ~]# odacli create-backup -n italy -bt Regular-L0
```

13. Remove the temporary vmstorage.

```
[oracle@oda1 ~]# odacli delete-vmstorage -n tempfra
```

Deleting a Database

Use the Oracle Appliance Manager Browser User Interface to delete an Oracle database.

1. Log into the Browser User Interface:

   `https://host name or ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.
3. Click **Actions** next to the database that you want, then select **Delete**.
4. In the Confirmation dialog, select **Force Delete** to force the deletion and then click **OK** to confirm the action.

Working with Database Homes

Use the Browser User Interface to display a list of database homes, details, and create and delete database homes.

- About Managing Multiple Oracle Homes on Oracle Database Appliance
  Create and manage one or more Oracle home directories and Oracle databases on Oracle Database Appliance.
- Viewing Database Homes
  Use the Browser User Interface to display a list of database homes and database home details, including databases associated with a DB home.

- Creating a Database Home
  Use the Browser User Interface to create database homes in Oracle Database Appliance.
- Deleting a Database Home
  Use the Browser User Interface to delete an Oracle database home.

About Managing Multiple Oracle Homes on Oracle Database Appliance

Create and manage one or more Oracle home directories and Oracle databases on Oracle Database Appliance.

Oracle home is the directory in which you install Oracle Database binaries, and from which Oracle Database runs. Oracle Database Appliance supports multiple Oracle homes, including support of different release Oracle Database homes. You can create multiple Oracle databases on a given Oracle home. Use Oracle Appliance Manager Browser User Interface to create and manage multiple Oracle homes and databases on Oracle Database Appliance. Oracle Database Appliance Manager automatically creates an Oracle Database Oracle home that is compliant with Oracle's Optimal Flexible Architecture (OFA) standards.

Check the *Oracle Database Appliance Release Notes* to obtain information about the specific Oracle software releases supported for your Oracle Database Appliance platform.

When you use ODACLI commands to create multiple homes on Oracle Database Appliance, the commands start the Oracle Home cloning process. In Oracle Database Appliance deployments, the user `oracle` is the default software installation owner.

You can use the Browser User Interface or the command-line interface to create and manage databases.

Use ODACLI commands to create, list, describe, and delete databases on Oracle Database Appliance. The `odacli create-database` command enables you to create a database with minimal user input. When you run this command without any additional options, the command creates a new database home (ORACLE_HOME). You can create a database in an existing home by using the `--dbhomeid` option. To find the *dbhomeid*, use the `odacli list-dbhomes` command.

Alternatively, you can use the Browser User Interface to create list, describe and delete databases and database homes. You can display a list of all Database Homes that includes the database home name, ID, version, the date and time that the database home was created and the location on the same page. You can also create and delete database homes on the Browser User Interface.

> **Caution:**
>
> Do not apply Oracle Database patches directly to Oracle Databases on Oracle Database Appliance. Only use Oracle Database Appliance patch bundles, which are tested to work across the whole software stack. If a one-off database patch is required, it may be applied to the Oracle Home. When you apply the Oracle Database Appliance patch bundle, it may cause a conflict during future patching events and you might need to roll back and then re-apply the patch.

Viewing Database Homes

Use the Browser User Interface to display a list of database homes and database home details, including databases associated with a DB home.

1. Log in to the Browser User Interface:

   `https://ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.
3. Click **Database Home** on the left menu to view all database homes. The name, ID, version, location and date and time stamp of when the database home was created appears on the page.
4. (Optional) Click **Actions** next to a database home entry, then **View Databases** to see the databases that are associated with the database home.

Creating a Database Home

Use the Browser User Interface to create database homes in Oracle Database Appliance.

Before you can create a database home, the Oracle Database Appliance RDBMS Clone file image must be in the repository. Follow these steps to create a database home:

1. Log in to the Browser User Interface:

   `https://ip-address:7093/mgmt/index.html`

2. Click the **Database** tab, then click **Database Home**.
3. Click **Create Database Home**.
4. Select a database version from the list of available versions, then select the database edition, either **Enterprise Edition** or **Standard Edition**, per your licensing agreement.
5. Click **Create**. When prompted, click **Yes** to confirm that you want to start the job.

Deleting a Database Home

Use the Browser User Interface to delete an Oracle database home.

You can delete a database home (DB Home) if it is not associated with any databases.

1. Log into the Browser User Interface:

   `https://host name or ip-address:7093/mgmt/index.html`

2. Click the **Database** tab.
3. Click **Database Home** on the left menu to view all database homes. The name, ID, version, location and date and time stamp of when the database home was created appears on the page.
4. Click **Actions** next to a database home entry, then **Delete**, and then confirm the action to delete a database home.

Migrating Databases

Review these topics to learn how to prepare for and migrate an entire database to your Oracle Database Appliance.

- About Migrating Databases
  You can migrate an entire active container database (CDB) or non-CDB database to an Oracle Database Appliance machine by using the RMAN duplicate command.
- Configuring a Static Listener
  Configure a static listener before you duplicate a database.

- Migrating a Database
  Use the `RMAN Duplicate` command to migrate the entire database to the appliance.
- Registering a Database
  Use the `odacli register-database` command to register the migrated database with the appliance.

About Migrating Databases

You can migrate an entire active container database (CDB) or non-CDB database to an Oracle Database Appliance machine by using the RMAN duplicate command.

When using RMAN Duplicate, ensure to have network connectivity from source and target databases:

- Source database: The existing database to be migrated.
- Target database: The new database created on an Oracle Database Appliance environment.

If you do not have network connectivity between the source and the target environments, you can use the offline migration method. Offline migration uses RMAN backup sets for duplication, which does not require connectivity to the primary database.

At a high level, the procedure involves the following steps:

1. **Deploy or update Oracle Database Appliance to the latest version.**

   Confirm that the provisioning completed successfully. On bare metal systems, use the command `odacli list-jobs` and the command `odacli describe-job` to verify the status.
2. **Create an instance only database from the command-line interface.**
   - On bare metal systems, use the command `odacli create-database` with the `instanceonly` flag on an Oracle Database Appliance machine. The new database is the target database.

   Creating an instance only database also creates the following:
   - ACFS Filesystem used to store the database files
   - Directory structures that are required by the database instance/rman duplicate command
   - Password file for the SYS user
3. **Configure a static listener.**
4. **Migrate the existing database to the target database using the backup and restore operations.**
5. **Register the migrated database with the appliance.**

> **Note:**
>
> You can only migrate and register a database of the same storage type. For example, to migrate and register an Oracle ACFS database, you must create an Oracle ACFS database and then migrate and register it. Similarly, to to migrate an Oracle ASM database, you must create an Oracle ASM database and then migrate it.

Configuring a Static Listener

Configure a static listener before you duplicate a database.

The static listener is only required when using RMAN Duplicate command.

Perform the following steps to manually configure the `listener.ora` file:

1. Log in as a grid user.
2. Navigate to `/u01/app/19.0.0.0/grid/network/admin/` directory.
3. Edit the default `listener.ora` file and append the static database entry.

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
         (GLOBAL_DBNAME=db_unique_name with domain)
         (ORACLE_HOME=db home)
         (SID_NAME=db unique name)
          (ENVS="TNS_ADMIN=database home/network/admin"))
 )
```

   For example, the following file is edited for an 19.25.0.0.0 database named `PRODDB` in the `example.com` domain:

```
SID_LIST_LISTENER=
 (SID_LIST=
    (SID_DESC=
      (GLOBAL_DBNAME=PRODDB.example.com)
       (ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_2)
       (SID_NAME=PRODDB)
            (ENVS="TNS_ADMIN=/u01/app/oracle/product/19.0.0.0/
dbhome_2/network/admin"))
 )
```

4. Save the file.
5. Restart the listener.

```
srvctl stop listener -l listener
srvctl start listener -l listener
```

Migrating a Database

Use the `RMAN Duplicate` command to migrate the entire database to the appliance.

Before migrating the database, verify that a network connection exists between the source and destination databases.

1. Verify that the `sysdba` connection to the auxiliary database is working. Use the password that you defined when you created the database with the `--instanceonly` flag.

```
sqlplus sys/password@hostname:PORT/GLOBAL_DBNAME as sysdba << EOF
select 1 from dual;
```

```
exit;
EOF
```

The following example uses the Welcome_12 password for myhost on port 1521 for PRODDB.example.com.

```
sqlplus sys/Welcome_12@myhost:1521/PRODDB.example.com as sysdba << EOF
select 1 from dual;
exit;
EOF
```

2. As oracle user, set the ORACLE_HOME and ORACLE_SID environment variables.

```
ORACLE_HOME=path of Oracle Home against which the AUX DB is created
ORACLE_SID=database unique name
```

3. Display a list of all database storage configured on the appliance and database storage identifiers (ID).

   Use the `odacli list-dbstorages` to display the storage IDs for all configured filesystems.

```
# odacli list-dbstorages
ID Type                                DBUnique  Name          Status
-------------------------------------  --------  ----------- ----------
68d13446-f26c-49ee-ab75-a393732aa88a   Asm       rdb1        Configured
ff2023d9-338d-4cff-8bb4-e73a89e32ce4   Acfs      PRODDB      Configured
```

4. Display the database storage details, including the DATA, RECO and REDO locations, for each database storage ID.

   For example, use the `odacli describe-dbstorage` command to display information about the ACFS storage for ID 9fe39332-cc1a-4b4b-8393-165524a6ef6b.

```
# odacli describe-dbstorage -i ff2023d9-338d-4cff-8bb4-e73a89e32ce4
DBStorage details
----------------------------------------------------------------
ID: ff2023d9-338d-4cff-8bb4-e73a89e32ce4
DB Name: PRODDB
DBUnique Name: PRODDB
DB Resource ID: c5b77384-634e-4dc8-b10b-fa2831d2c59b
Storage Type: Acfs
DATA Location: /u02/app/oracle/oradata/PRODDB
RECO Location: /u03/app/oracle/fast_recovery_area/
REDO Location: /u03/app/oracle/redo/
State: ResourceState(status=Configured)
Created: June 22, 2019 12:07:12 PM SGT
UpdatedTime: June 22, 2019 12:26:39 PM SGT
```

5. Duplicate the database.

   Use the RMAN duplicate database command to duplicate the database.

```
rman target sys/Welcome__12@source database hostname:PORT
(Continued)
```

```
/Service Name auxiliary sys/Welcome_12@Hostname of the target
database:1521/service name
RUN {
SET NEWNAME FOR DATABASE TO NEW;
duplicate target database to PRODDB from active database
SPFILE
SET cluster_database 'false'
SET audit_file_dest '/u01/app/oracle/admin/PRODDB/adump'
SET db_create_file_dest '/u02/app/oracle/oradata/PRODDB'
SET db_create_online_log_dest_1 '/u03/app/oracle/redo'
SET db_recovery_file_dest '/u03/app/oracle/fast_recovery_area'
SET control_files '/tmp/control_PRODDB.ctl' nofilenamecheck;
}
```

Registering a Database

Use the `odacli register-database` command to register the migrated database with the appliance.

The `dbclass`, `dbshape`, `servicename` and password are mandatory for registering the database. The `dbclass` and `dbshape` determine the `sga_target` and `pga_target` settings. The database `init.ora` parameters are reset as part of the `odacli register-database` command. Review the `init.ora` parameters after registration and ensure that the parameters set correctly.

Follow these steps to register a database:

1. Ensure that the `sqlpatches` are in the Oracle Database Appliance environment. If the patches are not in the environment, copy the `$OH/sqlpatch` directories from the source database home to the `$OH` in Oracle Database Appliance before executing the `odacli register-database` command

   The `odacli register-database` command invokes the `datapatch` utility for applying the post patch SQLs. If the source database has any `sqlpatches` applied that are not present in the Oracle Database Appliance environment, then the `datapatch` will fail.

2. Register the database with Oracle Database Appliance.

```
# odacli register-database -c OLTP -s odb1 -sn example_service
Enter SYS, SYSTEM and PDB Admin user password:
Retype SYS, SYSTEM and PDB Admin user password:
{
  "jobId" : "317b430f-ad5f-42ae-bb07-13f053d266e2",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "August 08, 2021 05:55:49 AM EDT",
  "description" : "Database service registration with
                   db service name: example_service",
  "updatedTime" : "August 08, 2018 05:55:49 AM EDT"
}
# odacli describe-job -i "317b430f-ad5f-42ae-bb07-13f053d266e2"

Job details
----------------------------------------------------------------
           ID: 317b430f-ad5f-42ae-bb07-13f053d266e2
```

```
   Description: Database service registration with
             db service name: example_service
         Status: Success
        Created: August 08, 2018 5:55:49 AM EDT
        Message:

Task Name                      Start Time
------------------------------ ------------------------------
restore control file           August 08, 2018 5:55:49 AM EDT
move spfile to right location  August 08, 2018 5:56:08 AM EDT
register DB with clusterware   August 08, 2018 5:56:13 AM EDT
reset db parameters            August 08, 2018 5:57:05 AM EDT
Running DataPatch              August 08, 2018 5:57:36 AM EDT

(Continued)
  End Time                     Status
------------------------------ --------
August 08, 2018 5:56:08 AM EDT   Success
August 08, 2018 5:56:13 AM EDT   Success
August 08, 2018 5:57:05 AM EDT   Success
August 08, 2018 5:57:36 AM EDT   Success
August 08, 2018 5:57:49 AM EDT   Success
```

3. Use the `odacli list-databases` command to view the registered database.

About Registering a TDE-Enabled Database

Understand the process to register Transparent Data Encryption (TDE) enabled databases.

Starting with Oracle Database Appliance release 19.12, a new attribute called `TDE Wallet Management` is added to the database. The `TDE Wallet Management` value can be either `EXTERNAL`, which means TDE is not configured by Oracle Database Appliance, or the value can be `ODA`, which means TDE is configured by Oracle Database Appliance. If the database is not TDE-enabled, then the value of the `TDE Wallet Management` attribute is NULL. If a database of release 18c or later is registered, then its TDE wallet management attribute value is `ODA` by default. If a database of release earlier than 18c is registered, then the TDE wallet management attribute value is `EXTERNAL` by default. The examples with sample output are provided at the end of this topic.

For Oracle Database release 19c or later, if the `TDE Wallet Management` value is `EXTERNAL`, then you can change this value to `ODA` using the `--change-tdewallet-mgmt` option in the `odacli modify-database` command. You must provide the TDE password when prompted.

For Oracle Database release 12c, the `TDE Wallet Management` value can be changed from `EXTERNAL` to `ODA`while upgrading the database to 19c or later using the `--change-tdewallet-mgmt` option in the `odacli upgrade-database` command.

> **Note:**
>
> Oracle Database Appliance does not support the conversion of a non-TDE database to a TDE-enabled database.

> **Note:**
>
> In this Oracle Database Appliance release, you cannot register a TDE-enabled database with TDE Keys stored on Oracle Key Vault.

Prerequisites to Register a TDE-Enabled Database

Following are the prerequisites to register a TDE-enabled database:

- You can use either the software keystore or the hardware keystore to configure TDE.
- If TDE is configured using the software keystore, then you must use only the `SQLNET.ENCRYPTION_WALLET_LOCATION` parameter.
- If TDE is configured using the software keystore, then for bare metal deployments, password-protected wallet (`ewallet.p12`) and autologin TDE wallet (`cwallet.sso`) must be available in the location `/opt/oracle/dcs/commonstore/wallets/tde/db_uniquename/`.
- If autologin TDE wallet (`cwallet.sso`) is not present but password-protected wallet (`ewallet.p12`) is present, then TDE password must be provided in the register database request.
- For Oracle Database Appliance DB systems deployments, if TDE is configured using Software keystore, then TDE must be configured using the `WALLET_ROOT` parameter and TDE wallets must be at location `+DATA/DB_UNIQUE_NAME/tde`.
- The instance name of the database being registered must be same as the database name. This condition also applies to the databases that do not have TDE enabled.

When `TDE Wallet Management` value is `EXTERNAL`

Following are the restrictions when the `TDE Wallet Management` value is `EXTERNAL`:

- TDE wallet re-key using ODACLI commands is not supported. You can connect to the database and re-key the TDE wallet.

```
# odacli modify-database -in db_name -rkt
Enter TDE wallet password:
DCS-10040:Operation 'Re-Key of TDE wallet' is not supported: TDE wallet
management is not ODA.
```

- TDE wallet password change using ODACLI commands is not supported.

```
# odacli modify-database -in db_name -ctp
Enter current TDE wallet password:
Enter new TDE wallet password:
Retype new TDE wallet password:
DCS-10040:Operation 'Password change of TDE wallet' is not supported: TDE
wallet management is not ODA.
```

- TDE wallet backup is not supported.

```
# odacli create-backup -in database_name -c tdewallet
DCS-10040:Operation 'Backup of TDE wallet' is not supported: TDE wallet
management is not ODA.
```

- TDE wallet is not backed up while performing database backup.

```
# odacli create-backup -in database_name -bt regular-l0
{
  "jobId" : "49153a90-d4bd-45e7-b3b7-46078621b895",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "August 24, 2021 05:59:53 AM UTC",
  "resourceList" : [ ],
  "description" : "Create regular-l0 Backup[TAG:auto][Db:db_name]
[OSS:example]",
  "updatedTime" : "August 24, 2021 05:59:53 AM UTC"
}

# odacli describe-job -i 49153a90-d4bd-45e7-b3b7-46078621b895

Job details
----------------------------------------------------------------
                     ID:  49153a90-d4bd-45e7-b3b7-46078621b895
            Description:  Create regular-l0 Backup[TAG:auto][Db:db_name]
[OSS:example]
                 Status:  Success
                Created:  August 24, 2021 5:59:53 AM UTC
                Message:

Task Name                                Start
Time                          End Time                          Status
----------------------------------------
---------------------------------- ----------------------------------
----------
Validate backup config                   August 24, 2021 5:59:56 AM
UTC      August 24, 2021 5:59:56 AM UTC      Success
Container validation                     August 24, 2021 5:59:56 AM
UTC      August 24, 2021 5:59:57 AM UTC      Success
libopc existence check                   August 24, 2021 5:59:57 AM
UTC      August 24, 2021 5:59:57 AM UTC      Success
Backup Validations                       August 24, 2021 5:59:57 AM
UTC      August 24, 2021 6:00:02 AM UTC      Success
Recovery Window validation               August 24, 2021 6:00:02 AM
UTC      August 24, 2021 6:00:05 AM UTC      Success
Archivelog deletion policy configuration August 24, 2021 6:00:05 AM
UTC      August 24, 2021 6:00:08 AM UTC      Success
Database backup                          August 24, 2021 6:00:08 AM
UTC      August 24, 2021 6:02:06 AM UTC      Success

# odacli describe-backupreport -i b75c7ffa-68f8-4eea-8f12-bfe30b8c92f7 |
grep -i tde
  "tdeWalletLoc" : null,
```

- Restore of TDE wallet is not supported.

```
# odacli restore-tdewallet -in db_name
Enter TDE wallet password:
```

```
DCS-10040:Operation 'Restore of TDE wallet' is not supported: TDE wallet
management is not ODA.
```

- Cloning of TDE-enabled database is not supported.

```
odacli clone-database -n db_name2 -f db_name -u db_name2
Enter SYS user password:
Retype SYS user password:
DCS-10040:Operation 'Cloning of TDE database' is not supported: TDE wallet
management is not ODA.
```

- When you move the database from one database home to another database home, the `SQLNET.ENCRYPTION_WALLET_LOCATION` parameter is updated in the destination database home.
- When you upgrade the database, the `SQLNET.ENCRYPTION_WALLET_LOCATION` parameter is updated in the new database home.
- When you update the database home, the `SQLNET.ENCRYPTION_WALLET_LOCATION` parameter is updated in the new database home.
- When you delete a database, the TDE wallet is not deleted.
- For Oracle Database Appliance DB systems, only databases that have TDE configured using the `WALLET_ROOT` parameter can be registered.

Example 9-1 Sample output of TDE-enabled database registered in bare metal deployments

```
# odacli describe-database -in mydb
  Database details
  ----------------------------------------------------------------
                     ID: 47de99d6-62cc-4623-a38c-1ac880e7082c
            Description: mydb
                DB Name: mydb
             DB Version: 19.11.0.0.210420
                DB Type: SI
                DB Role: PRIMARY
    DB Target Node Name: node1
              DB Edition: EE
                    DBID: 2987837625
Instance Only Database: false
                    CDB: false
               PDB Name:
    PDB Admin User Name:
High-Availability Enabled: false
                  Class: OLTP
                  Shape: odb1
                Storage: ASM
          DB Redundancy:
           CharacterSet: AL32UTF8
  National CharacterSet: AL16UTF16
               Language: AMERICAN
              Territory: AMERICA
                Home ID: 6717bed4-104a-415b-8d26-c0de634a2c85
        Console Enabled: false
            TDE Enabled: false
  TDE Wallet Management: External
```

```
     Level 0 Backup Day: Sunday
     AutoBackup Enabled: true
                Created: May 18, 2021 3:33:13 AM UTC
        DB Domain Name: example_domain
    Associated Networks: Public-network
          CPU Pool Name:
```

Example 9-2 Sample output of TDE-enabled database registered in Oracle Database Appliance DB System deployments

```
# odacli describe-database -in mydb
odacli describe-database -in mydb
Database details
----------------------------------------------------------------
                     ID: 47de99d6-62cc-4623-a38c-1ac880e7082c
            Description: mydb
                DB Name: mydb
             DB Version: 19.11.0.0.210420
                DB Type: SI
                DB Role: PRIMARY
    DB Target Node Name: node1
             DB Edition: EE
                   DBID: 2987837625
Instance Only Database: false
                    CDB: false
                PDB Name:
    PDB Admin User Name:
High-Availability Enabled:false
                  Class: OLTP
                  Shape: odb1
                Storage: ASM
          DB Redundancy:
           CharacterSet: AL32UTF8
  National CharacterSet: AL16UTF16
               Language: AMERICAN
              Territory: AMERICA
                Home ID: 6717bed4-104a-415b-8d26-c0de634a2c85
        Console Enabled: false
            TDE Enabled: false
  TDE Wallet Management: ODA
     Level 0 Backup Day: Sunday
     AutoBackup Enabled: true
                Created: May 18, 2021 4:33:13 AM UTC
        DB Domain Name: example_domain
    Associated Networks: Public-network
          CPU Pool Name:
```

Example 9-3 Sample output of TDE-enabled database registered in Oracle Database Appliance when password-protected wallet (ewallet.p12) exists but not the autologin TDE wallet (cwallet.sso)

```
# odacli register-database  -sn example_service --dbclass OLTP --dbshape odb2
Enter SYS user password:
DCS-10011:Input parameter 'TDE wallet password' cannot be NULL.Required to
generate Autologin wallet at /opt/oracle/dcs/commonstore/wallets/tde/db_name
```

```
odacli register-database  -sn example_service --dbclass OLTP --dbshape odb2 -
tp
Enter SYS, SYSTEM and PDB Admin user password:
Retype SYS, SYSTEM and PDB Admin user password:
Enter TDE wallet password
Retype TDE wallet password:
{
  "jobId" : "4903b711-f144-4aed-9bf6-31c79dbce261",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "August 25, 2021 05:42:13 AM UTC",
  "resourceList" : [ ],
  "description" : "Database service registration with db service name:
example_service",
  "updatedTime" : "August 25, 2021 05:42:14 AM UTC"
}

# odacli describe-job -i 4903b711-f144-4aed-9bf6-31c79dbce261

Job details
----------------------------------------------------------------
                     ID:  4903b711-f144-4aed-9bf6-31c79dbce261
            Description:  Database service registration with db service name:
example_service
                 Status:  Success
                Created:  August 25, 2021 5:42:13 AM UTC
                Message:

Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
TDE parameter validate at destination    August 25, 2021 5:42:14 AM UTC
August 25, 2021 5:42:14 AM UTC      Success
Enable OMF parameters                    August 25, 2021 5:42:17 AM UTC
August 25, 2021 5:42:17 AM UTC      Success
Setting db character set                 August 25, 2021 5:42:17 AM UTC
August 25, 2021 5:42:18 AM UTC      Success
Move Spfile to right location            August 25, 2021 5:42:18 AM UTC
August 25, 2021 5:42:24 AM UTC      Success
Enable DbSizing Template                 August 25, 2021 5:42:24 AM UTC
August 25, 2021 5:43:15 AM UTC      Success
Copy Pwfile to Shared Storage            August 25, 2021 5:43:16 AM UTC
August 25, 2021 5:43:20 AM UTC      Success
Running DataPatch                        August 25, 2021 5:43:20 AM UTC
August 25, 2021 5:43:32 AM UTC      Success
configuring TDE                          August 25, 2021 5:43:32 AM UTC
August 25, 2021 5:44:51 AM UTC      Success
Reset Associated Networks                August 25, 2021 5:44:52 AM UTC
August 25, 2021 5:44:55 AM UTC      Success
```

About Managing Multiple Database Instances Using Instance Caging

Use instance caging to manage your system resources on Oracle Database Appliance.

Oracle Database provides a method for managing CPU allocations on a multi-CPU server that runs multiple database instances. This method is called instance caging. Instance caging uses an initialization parameter to limit the number of CPUs that an instance can use simultaneously.

Instance caging and Oracle Database Resource Manager (the Resource Manager) work together to support your desired service levels across multiple instances. Consolidation can minimize idle resources, maximize efficiency, and lower costs.

Oracle Database Appliance templates are already tuned for the size of each database instance workload. They are designed to run on a specific number of cores. Instance caging ensures that each database workload is restricted to the set of cores allocated by the template, enabling multiple databases to run concurrently with no performance degradation, up to the capacity of Oracle Database Appliance. You can select database template sizes larger than your current needs to provide for planned growth.

> **Note:**
>
> Oracle strongly recommends that you use the Oracle Database Appliance templates, because they implement best practices and are configured specifically for Oracle Database Appliance.
>
> The Oracle Database Appliance Manager interface refers to the database sizing templates as database classes.

By default, instance caging is not enabled on Oracle Database Appliance. To enable instance caging, set the initialization parameter, `RESOURCE_MANAGER_PLAN`, for each database on Oracle Database Appliance. The parameter specifies the plan to be used by the Resource Manager for the current instance. Setting this parameter directs the Resource Manager to allocate core resources among databases. If a plan is not specified with this parameter, then the Resource Manager and instance caging are not enabled.

Instance caging allocation of core resources is enabled in accordance with the Oracle Database Appliance database template size that you select for each database. The `CPU_COUNT` initialization parameter is set in the template. Use the `CPU_COUNT` setting that matches the size of each database to consolidate, and follow the standard instructions for configuring instance caging.

Oracle EM Express and DB Console

You can use Oracle Enterprise Manager Database Express (EM Express), or the Database Control Console (DB Console) to manage your database.

The EM Express console is available for supported releases of Oracle Database. The consoles are web-based tools for managing Oracle Databases.

The EM Express console provides the following features:

- Support for basic administrative tasks, such as storage and user management
- Comprehensive solutions for performance diagnostics and tuning
- Performance advisors in a graphic user interface
- Oracle Database utilities in a graphic user interface, such as SQL*Loader and Oracle Recovery Manager (RMAN)

EM Express is built inside the database server and cannot perform actions outside the database.

10

Managing Storage

Understand the storage options and how to manage storage for your Oracle Database Appliance deployment.

- About Managing Storage
  Understand Oracle Database Appliance storage options.
- About Managing Oracle ASM Disks
  Understand the Oracle ASM disk management features that Oracle Database Appliance supports.
- Managing Storage on Single-Node Systems
  Understand the storage options for your Oracle Database Appliance X9-2S and X9-2L systems.
- Managing Storage on High-Availability Systems
  Understand the storage for your Oracle Database Appliance X9-2-HA system.

About Managing Storage

Understand Oracle Database Appliance storage options.

Oracle Database Appliance uses raw storage to protect data in the following ways:

- Fast Recovery Area (FRA) backup. FRA is a storage area (directory on disk or Oracle ASM diskgroup) that contains redo logs, control file, archived logs, backup pieces and copies, and flashback logs.
- Mirroring. Double or triple mirroring provides protection against mechanical issues.

The amount of available storage is determined by the location of the FRA backup (external or internal) and if double or triple mirroring is used. External NFS storage is supported for online backups, data staging, or additional database files.

Oracle Database Appliance X9-2L and X9-2-HA models provide storage expansion options from the base configuration. In addition, on Oracle Database Appliance X9-2-HA multi-node platforms, you can add an optional storage expansion shelf.

The redundancy level for FLASH is based on the DATA and RECO selection. If you choose High redundancy (triple mirroring), then FLASH is also High redundancy.

About Managing Oracle ASM Disks

Understand the Oracle ASM disk management features that Oracle Database Appliance supports.

Oracle Database Appliance enables you to manage your Oracle ASM disks.

Bringing Oracle ASM Disk Groups Online Automatically

Oracle Database Appliance periodically checks the status of Oracle ASM disks in disk groups. If any Oracle ASM disk is OFFLINE due to transient disk errors, then Oracle Database Appliance attempts to bring the disk ONLINE.

Optimizing Oracle ASM Disk Group Rebalance Operations

Oracle Database Appliance ensures rebalancing of Oracle ASM disks to complete as quickly as possible, without overloading the system and the disks at the same time. This ensures that the system is in a steady state running with the appropriate redundancy. There are default thresholds for the values for rebalancing operations, and you can also set the custom threshold values. For example:

```
odacli modify-agentconfig-parameters -n ASMRM_CPU_RQ -v 50 -d "CPU RUN QUEUE
THRESHOLD" -u
odacli modify-agentconfig-parameters -n ASMRM_MAX_HDD_DISK_RQ -v 2 -d "HDD
DISK QUEUE THRESHOLD" -u
odacli modify-agentconfig-parameters -n ASMRM_MAX_SSD_DISK_RQ -v 32 -d "SSD
DISK QUEUE THRESHOLD" -u
odacli modify-agentconfig-parameters -n ASMRM_MAX_NVME_DISK_RQ -v 50 -d "NVME
DISK QUEUE THRESHOLD" -u
```

The above command options set custom threshold limits for rebalance monitoring of Oracle ASM disks.

You can monitor rebalance operations using the `odacli describe-schedule -i` *`Schedule ID`* and `odacli list-scheduled-executions` commands.

Managing Storage on Single-Node Systems

Understand the storage options for your Oracle Database Appliance X9-2S and X9-2L systems.

- About Storage on Oracle Database Appliance X9-2S and X9-2L
  Understand the storage for your Oracle Database Appliance single-node system.
- Adding NVMe Storage Disks
  Depending on the available drives, you can expand Oracle Database Appliance X9-2 storage to add NVMe disks or replace existing NVMe disks.
- Replacing NVMe Storage Disks
  Understand how you can replace existing NVMe disks on Oracle Database Appliance.

About Storage on Oracle Database Appliance X9-2S and X9-2L

Understand the storage for your Oracle Database Appliance single-node system.

Oracle Database Appliance X9-2S has two 6.8TB NVMe disks that host DATA and RECO disk groups. There are two partitions, one each for DATA and RECO for Oracle ASM storage information. The storage capacity is fixed and cannot be expanded.

Oracle Database Appliance X9-2L has two 6.8TB NVMe disks that host DATA and RECO disk groups. There are two partitions, one each for DATA and RECO for Oracle ASM storage information. When you first deploy and configure X9-2L in this release, you can set the storage on X9-2L in multiple of 2 packs of NVMe drives, such as 2, 4, 6, 8, and 10 disks, up to a maximum of 12 disks.

The table describes the NVMe storage configurations with expansion memory and storage options for single-node systems.

Table 10-1 Storage Options for Oracle Database Appliance X9-2S and X9-2L

Configuration	Oracle Database Appliance X9-2S	Oracle Database Appliance X9-2L
Base Configuration	2 x 6.8 TB NVMe = 13.6 TB NVMe	2 x 6.8 TB NVMe = 13.6 TB NVMe
Storage addition options	None	10 x 6.8 TB NVMe storage drives for a total storage of 81.6 TB NVMe. Order Qty 1 - 7603394 (Two 6.8 TB 2.5-inch NVMe PCIe 4.0 SSD with coral-d bracket for Oracle Database Appliance X9-2L) Pack of two 6.8 TB NVMe disks to a maximum of 12 disks.

Adding NVMe Storage Disks

Depending on the available drives, you can expand Oracle Database Appliance X9-2 storage to add NVMe disks or replace existing NVMe disks.

Use the ODAADMCLI commands to perform appliance storage maintenance tasks, including perform storage diagnostics and collect diagnostic logs for storage components.

Preparing for a Storage Upgrade

1. Check the disk health of the existing storage disks.

```
# odaadmcli show disk
```

2. Run the the `odaadmcli show disk` and `asmcmd lsdsk -p` commands to view and review the storage disk information in OAKD and Oracle Automatic Storage Management (Oracle ASM).

```
# odaadmcli show disk


# asmcmd lsdsk -p
```

3. Use ORAchk to confirm Oracle ASM and Oracle Clusterware health.

Adding NVMe Storage Disks

The default configuration for Oracle Database Appliance X9-2S or X9-2L includes two (2) NVMe disks. You cannot expand storage for Oracle Database Appliance X9-2S.

For Oracle Database Appliance X9-2L, you can expand storage by adding 2, 4, 6, 8, and 10 disks, up to a maximum of 12 disks. When you expand storage, adding odd numbers of NVMe drives is **not** supported.

> **WARNING:**
>
> **Pulling a drive before powering it off will crash the kernel, which can lead to data corruption. Do not pull the drive when the LED is an amber or green color. When you need to replace an NVMe drive, use the software to power off the drive before pulling the drive from the slot. If you have more than one disk to replace, complete the replacement of one disk before starting replacement of the next disk.**

Follow **all** these steps to add NVMe storage disks:

1. Before adding the NVMe disks, ensure that the current disks are online in `oakd` and Oracle ASM. Otherwise, the prechecks fail. For example, for 2-disks expansion to slots 2 and 3, the disks in slots 0 and 1 must be online in Oracle ASM and `oakd`. For 4-disks expansion from slots 2 to 5 when slots 0 to 5 are filled, then all disks in slots 0 to 1 must be online. For 10-disks expansion from slots 2 to 11, all disks in slots 0 to 1 must be online.
2. Insert each disk one at a time in the appropriate slot and wait for the disk to power ON.
3. The disk automatically powers on when you insert the disk in the slot. Wait for one minute and then check disk status. **If the disk is in the ON state, then you need not power on the disk manually.** If the disk state is OFF as per the disk status, and then power on the disk manually, and then check the status again.

   ```
   # odaadmcli power disk status slot_number


   # odaadmcli power disk on slot_number
   ```

   For example, to add two (2) NVMe disks, insert the disks in slots 2 and 3.

   ```
   # odaadmcli power disk status pd_02
   # odaadmcli power disk on pd_02

   # odaadmcli power disk status pd_03
   # odaadmcli power disk on pd_03
   ```

4. Repeat steps 2 and 3 for each disk to be added.
5. Run the `odaadmcli expand storage` command to add the new storage disks. **Note:** You must run this step to add the storage disk. Otherwise, the newly-added disk is not visible to OAKD and hence does not display when you run the `odaadmcli show disk` or `odaadmcli show storage` commands. The newly-added disk is recognized by OAKD after the `odaadmcli expand storage` command completes running successfully.

   ```
   # odaadmcli expand storage -ndisk number_of_disks
   ```

   For example, to add two (2) NVMe drives:

   ```
   #odaadmcli expand storage -ndisk 2
   Running precheck, it may take a few minutes.
   Precheck passed.
   Check the progress of expansion of storage by executing 'odaadmcli show
   ```

```
disk'
Waiting for expansion to finish. It may take several minutes to complete
depending upon the number of disks being expanded
```

6. Check the status of the new disk in OAKD with the `odaadmcli show disk` command. The disk must have the status `Online` and `Good` in OAKD. Check the status of the new disk in Oracle ASM with the `asmcmd lsdsk -p` command. The disk must be in `CACHED MEMBER ONLINE NORMAL` state.

```
# odaadmcli show disk


# asmcmd lsdsk -p
```

Replacing NVMe Storage Disks

Understand how you can replace existing NVMe disks on Oracle Database Appliance.

Preparing for a Storage Upgrade

1. Update Oracle Database Appliance to the latest Patch Bundle before expanding storage.

```
# odacli describe-component
```

2. Check the disk health of the existing storage disks.

```
# odaadmcli show disk
```

3. Run the the `odaadmcli show disk` and `asmcmd lsdsk -p` commands to view and review the storage disk information in OAKD and Oracle Automatic Storage Management (Oracle ASM).

```
# odaadmcli show disk


# asmcmd lsdsk -p
```

4. Use ORAchk to confirm Oracle ASM and Oracle Clusterware health.

Replacing NVMe Storage Disks

Follow **all** these steps to replace NVMe storage disks:

> **WARNING:**
>
> **Pulling a drive before powering it off will crash the kernel, which can lead to data corruption. Do not pull the drive when the LED is an amber or green color. When you need to replace an NVMe drive, use the software to power off the drive before pulling the drive from the slot. If you have more than one disk to replace, complete the replacement of one disk before starting replacement of the next disk.**

1. Power OFF the NVMe disk before removing it from the slot.
2. Wait for one minute for OAKD to complete the operation for disk removal.
3. Insert the new disk in the slot.
4. Wait for at least 2-3 minutes between inserting each disk for OAKD to complete the operation to add the disk to Oracle ASM and OAK.
5. Check the status of the new disk in OAKD with the `odaadmcli show disk` command. The disk must have the status `Online` and `Good` in OAKD. Check the status of the new disk in Oracle ASM with the `asmcmd lsdsk -p` command. The disk must be in `CACHED MEMBER ONLINE NORMAL` state.

```
# odaadmcli show disk


# asmcmd lsdsk -p
```

Managing Storage on High-Availability Systems

Understand the storage for your Oracle Database Appliance X9-2-HA system.

- About Storage Options for Oracle Database Appliance X9-2-HA
  Oracle Database Appliance High-Availability systems have options for high performance and high capacity storage configurations.
- Adding Solid-State Drives (SSDs) for Data Storage
  Add a pack of solid-state drives (SSDs) for data storage into the existing Oracle Database Appliance X9-2-HA base configuration to fully populate the base storage shelf.
- Adding the Storage Expansion Shelf
  After the base storage shelf is fully populated, you can add the storage expansion shelf to expand your data storage on your multi-node platform.

About Storage Options for Oracle Database Appliance X9-2-HA

Oracle Database Appliance High-Availability systems have options for high performance and high capacity storage configurations.

The base configuration of Oracle Database Appliance X9-2-HA hardware model has six slots (slots 0-5) with 7.68 TB drives of SSD raw storage. If you choose to order and deploy the full storage capacity, then you can fill the remaining 18 slots (slots 6-23) with either SSD or HDD drives. For even more storage, you can add a storage expansion shelf to double the storage capacity of your appliance.

In all configurations, the base storage and the storage expansion shelf each have six SSDs for DATA/RECO in the SSD option or FLASH in the HDD option.

Oracle Database Appliance X9-2-HA does not allocate dedicated SSD drives for REDO disk groups. Instead, the space for REDO logs is allocated on SSD drives as required.

For Oracle ASM storage, the REDO logs are stored in the available disk group space during database creation, based on the database shape selected. For Oracle ACFS storage, the space for REDO logs is allocated during the database storage creation assuming the minimum db shape (odb1s). If you create the database storage without database, then the space allocated for REDO logs is 4 GB, assuming the minimum db shape (odb1s). Subsequently,

when you create a database with your required database shape on the existing database storage, the REDO logs space is extended based on shape of the database.

On Oracle Database Appliance X9-2-HA High Performance configurations, with only SSD drives, the DATA and RECO disk groups use all the SSD drives whether 6, 12, 18, 24, or 48 with storage expansion shelf. REDO logs are stored in the RECO disk group.

On Oracle Database Appliance X9-2-HA High Capacity configurations, with both HDD and SSD drives, the DATA and RECO disk groups use the HDD drives, and the SSD drives store the FLASH disk group. REDO logs are stored in the FLASH disk group.

On both High Performance and High Capacity configurations, REDO logs are always created on SSD drives, similar to earlier Oracle Database Appliance hardware models. REDO logs are always created with high redundancy irrespective of the redundancy level of the disk group, whether RECO or FLASH.

High Performance

A high performance configuration uses solid state drives (SSDs) for DATA and RECO storage. The base configuration has six disks, each with 7.68 TB SSD raw storage for DATA and RECO.

You can add up to three (3) 6-Pack SSDs on the base configuration, for a total of 184.32 TB SSD raw storage. If you need more storage, you can double the capacity by adding an expansion shelf of SSD drives. The expansion shelf provides an additional 24 SSDs, each with 7.68TB raw storage for DATA and RECO, for a total of another 184.32 TB SSD raw storage.

Adding an expansion shelf requires that the base storage shelf and expansion shelf are fully populated with SSD drives. When you expand the storage, there is no downtime.

A system fully configured for high performance has 368.64 TB SSD raw storage for DATA and RECO.

High Capacity

A high capacity configuration uses a combination of SSD and HDD drives.

The base configuration has six disks, each with 7.68 TB SSD raw storage for FLASH.

The following expansion options are available:

- Base shelf: additional 324 TB HDD raw storage for DATA and RECO (18 HDDs, each with 18 TB storage)
- Expansion Storage shelf: additional shelf storage configuration must be identical to the storage configuration of the base shelf.

A system fully configured for high capacity has a total of 740.16 TB raw storage for DATA, RECO, and FLASH, with 92.16 TB SSD and 648 TB HDD.

Table 10-2 Storage Options for Oracle Database Appliance X9-2-HA

Configuration	Oracle Database Appliance X9-2-HA SSD-Only Configuration for High Performance	Oracle Database Appliance X9-2-HA SSD and HDD Configuration for High Capacity
Base configuration	Base storage shelf contains 6 SSDs of 7.68 TB. • 6 x 7.68 TB SSD = 46 TB SSD	Base storage shelf is fully populated with 6-pack SSDs of 7.68 TB and 18-drives of HDDs with 18 TB. • 6 x 7.68 TB SSD = 46 TB SSD • 18 x 18TB HDD = 324 TB HDD • Total storage on the first JBOD = 370 TB, with 46 TB SSD and 324 TB HDD
Storage addition options	Base shelf contains 6 SSDs. Additional 18 SSDs must be added in packs of 6. • Base system: 6 x 7.68 TB SSD = 46 TB SSD • Adding 6 SSDs: 12x 7.68 TB SSD = 92 TB SSD • Adding 12 SSDs: 18 x 7.68 TB SSD = 138 TB SSD • Adding 18 SSDs: 24 x 7.68 TB SSD = 184 TB SSD (full shelf)	Not applicable. Base storage shelf is fully populated.
Storage shelf expansion options	• The optional expansion storage shelf can only be installed after the base storage shelf is fully populated, and it must have the same configuration as the base storage shelf. • Total storage on the base storage shelf = 184 TB SSD • Storage on the expansion shelf = 24 x 7.68 TB SSD = 184 TB SSD • Total storage including both JBODs = 368.6 TB SSD	• The optional expansion storage shelf can only be installed after the base storage shelf is fully populated, and it must have the same configuration as the base storage shelf. • Total storage on the base storage shelf = 370 TB, with 46 TB SSD and 324 TB HDD • Total storage including both JBODs = 740 TB, with 92 TB SSD and 648 TB HDD

Adding Solid-State Drives (SSDs) for Data Storage

Add a pack of solid-state drives (SSDs) for data storage into the existing Oracle Database Appliance X9-2-HA base configuration to fully populate the base storage shelf.

If you need to add storage to the base configuration, you can order one, two, or three 6-pack of SSDs to complete the base configuration on Oracle Database Appliance X9-2-HA.

You must fully populate the base configuration before you can add an expansion shelf to Oracle Database Appliance X9-2-HA. If you add an expansion shelf, the shelf must have the same disk storage configuration as the base configuration.

> **Note:**
>
> For a high-performance configuration, you can add SSDs to the base storage shelf or add a storage expansion shelf. For high-capacity base configuration with 6-SSDs, if you want to expand storage to use HDDs, then you must reimage and deploy the appliance.

Before adding the disks to the system, ensure that Oracle Database Appliance is on the latest release.

1. Insert disks one at a time in the slots.

   To add one 6-pack of SSDs, insert the disks in slots 6 to 11. To add two 6-pack of SSDs, insert the disks in slots 6 to 17. To add three 6-pack of SSDs, insert the disks in slots 6 to 23.

   > **Note:**
   >
   > Allow at least one minute between inserting each disk.

   After all disks are added, go to Step 2.

2. Run the `odaadmcli expand storage` command on any node.

   ```
   # odaadmcli expand storage -ndisk number_of_disks_to_be_added -enclosure
   enclosure_number_of_the_disks_to_be_added
   ```

   The enclosure number is 0 when you add storage disks to the first JBOD.

   For example:

   ```
   # odaadmcli expand storage -ndisk 6 -enclosure 0
   Running precheck, it may take a few minutes.
   Precheck passed.
   Check the progress of expansion of storage by running 'odaadmcli show disk'
   Waiting for expansion to finish. It may take several minutes to complete
   depending upon the number of disks being expanded.

   odaadmcli expand storage command completed.
   Check status of newly expanded disk in OAK by running 'odaadmcli show disk'
   All of them must be ONLINE status. Check status of newly expanded disk in
   ASM
   by running 'asmcmd lsdsk -p', mount status must be CACHED and header status
   should be MEMBER for all the newly added disks. If the status of newly
   added
   disks in OAK and ASM are different then expand storage command failed.
   ```

   It takes 10 to 12 minutes to add all of the disks to the configuration.

3. Run the `odaadmcli show disk` command to ensure that all disks are listed, are online, and are in a good state.

   ```
   # odaadmcli show disk
   ```

4. Verify that the disks in slots 6 to 11 are added to Oracle Automatic Storage Management (Oracle ASM).

a. Run the `asm_script` to verify that the disks in slots 6 to 11 are added to Oracle Automatic Storage Management (Oracle ASM). If the 6 disks are successfully added (CACHED and MEMBER), then go to Step 7.

```
# su gridUser /opt/oracle/oak/bin/stordiag/asm_script.sh 1 6
```

For example:

```
# su grid /opt/oracle/oak/bin/stordiag/asm_script.sh 1 6 | grep CACHED
.......
/dev/mapper/SSD_E0_S06_1399645200p1 SSD_E0_S06_1399645200P1 1 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S06_1399645200p2 SSD_E0_S06_1399645200P2 3 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S07_1399646692p1 SSD_E0_S07_1399646692P1 1 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S07_1399646692p2 SSD_E0_S07_1399646692P2 3 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S08_1399649840p1 SSD_E0_S08_1399649840P1 1 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S08_1399649840p2 SSD_E0_S08_1399649840P2 3 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S09_1399649424p1 SSD_E0_S09_1399649424P1 1 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S09_1399649424p2 SSD_E0_S09_1399649424P2 3 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S10_1399649846p1 SSD_E0_S10_1399649846P1 1 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S10_1399649846p2 SSD_E0_S10_1399649846P2 3 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S11_1399649428p1 SSD_E0_S11_1399649428P1 1 NORMAL
ONLINE CACHED MEMBER
/dev/mapper/SSD_E0_S11_1399649428p2 SSD_E0_S11_1399649428P2 3 NORMAL
ONLINE CACHED MEMBER
```

b. If the disks are not added to Oracle ASM, then add them manually. As `grid` user, execute the `sqlplus '/as sysasm'` command on the first node to add the disks to Oracle ASM.

For a system without Oracle Automatic Storage Management Filter Driver (Oracle ASM Filter Driver) configured, add the Oracle ASM disks as follows:

```
SQL> alter diskgroup /*+ _OAK_AsmCookie */ data add disk
'/dev/mapper/SSD_E0_S06_1399765076p1' name SSD_E0_S06_1399765076p1,
'/dev/mapper/SSD_E0_S07_1399765116p1' name SSD_E0_S07_1399765116p1,
'/dev/mapper/SSD_E0_S08_1399765484p1' name SSD_E0_S08_1399765484p1,
'/dev/mapper/SSD_E0_S09_1399765504p1' name SSD_E0_S09_1399765504p1,
'/dev/mapper/SSD_E0_S10_1399765506p1' name SSD_E0_S10_1399765506p1,
'/dev/mapper/SSD_E0_S11_1399765508p1' name SSD_E0_S11_1399765508p1;

SQL> alter diskgroup /*+ _OAK_AsmCookie */ reco add disk
'/dev/mapper/SSD_E0_S06_1399765076p2' name SSD_E0_S06_1399765076p2,
'/dev/mapper/SSD_E0_S07_1399765116p2' name SSD_E0_S07_1399765116p2,
'/dev/mapper/SSD_E0_S08_1399765484p2' name SSD_E0_S08_1399765484p2,
```

```
'/dev/mapper/SSD_E0_S09_1399765504p2' name SSD_E0_S09_1399765504p2,
'/dev/mapper/SSD_E0_S10_1399765506p2' name SSD_E0_S10_1399765506p2,
'/dev/mapper/SSD_E0_S11_1399765508p2' name SSD_E0_S11_1399765508p2;
```

For a system with Oracle Automatic Storage Management Filter Driver (Oracle ASM Filter Driver) configured, add the Oracle ASM disks as follows:

```
SQL> alter diskgroup /*+ _OAK_AsmCookie */ data add disk
'AFD:SSD_E0_S06_1399765076P1' name SSD_E0_S06_1399765076p1,
'AFD:SSD_E0_S07_1399765116P1' name SSD_E0_S07_1399765116p1,
'AFD:SSD_E0_S08_1399765484P1' name SSD_E0_S08_1399765484p1,
'AFD:SSD_E0_S09_1399765504P1' name SSD_E0_S09_1399765504p1,
'AFD:SSD_E0_S10_1399765506P1' name SSD_E0_S10_1399765506p1,
'AFD:SSD_E0_S11_1399765508P1' name SSD_E0_S11_1399765508p1;

SQL> alter diskgroup /*+ _OAK_AsmCookie */ reco add disk
'AFD:SSD_E0_S06_1399765076P2' name SSD_E0_S06_1399765076p2,
'AFD:SSD_E0_S07_1399765116P2' name SSD_E0_S07_1399765116p2,
'AFD:SSD_E0_S08_1399765484P2' name SSD_E0_S08_1399765484p2,
'AFD:SSD_E0_S09_1399765504P2' name SSD_E0_S09_1399765504p2,
'AFD:SSD_E0_S10_1399765506P2' name SSD_E0_S10_1399765506p2,
'AFD:SSD_E0_S11_1399765508P2' name SSD_E0_S11_1399765508p2;
```

5. Use the `odaadmcli show validation storage errors` command to show hard storage errors.

   Hard errors include having the wrong type of disk inserted into a particular slot, an invalid disk model, or an incorrect disk size.

   ```
   # odaadmcli show validation storage errors
   ```

6. Use the `odaadmcli show validation storage failures` command to show soft validation errors.

   A typical soft disk error would be an invalid version of the disk firmware.

   ```
   # odaadmcli show validation storage failures
   ```

7. Confirm that the `oak_storage_conf.xml` file shows the number of disks added on both nodes. For example, if you added 6 disks to the base configuration, then the `numberOfDisks` option must show 12. If you added 12 disks to the base configuration, then the `numberOfDisks` option must show 18.

   ```
   #cat /opt/oracle/oak/conf/oak_storage_conf.xml
   <!--
   This file is created by the ODA software as part of system provisioning
   based on system provisioning requests. Values of element nodes can be
   changed by OAK in response to storage configuration change operation.
   DO NOT EDIT THIS FILE.
   -->
   <CometConfiguration>
     <OakStorageConfigInfo type="string" dimension="vector" readonly="true"
   required="true" default="">
       <!-- Number of disks part of OAK -->
       <numberOfDisks>12</numberOfDisks>
       <!-- Number of partitions per disk part of DATA diskgroup in single
   partition scheme -->
       <!-- All ASM diskgroups in single partition scheme will have same
   ```

```
number of partition -->
    <numOfDataDiskPartitionInAsm>1</numOfDataDiskPartitionInAsm>
  </OakStorageConfigInfo>
</CometConfiguration>
```

Adding the Storage Expansion Shelf

After the base storage shelf is fully populated, you can add the storage expansion shelf to expand your data storage on your multi-node platform.

The expansion shelf is available on Oracle Database Appliance multi-node platforms, such as Oracle Database Appliance X9-2-HA. The addition of the storage expansion shelf includes checks across both nodes. It is important to confirm that SSH does work across the nodes and all users can connect as expected using their shared password.

You must fully populate the base configuration before you can add an expansion shelf. If you add an expansion shelf, the shelf must have the same disk storage configuration as the base storage shelf.

> **Note:**
>
> Oracle recommends that you add a storage expansion shelf when you have relatively little activity on your databases. When the system discovers the new storage, Oracle Automatic Storage Management (Oracle ASM) automatically rebalances the disk groups. The rebalance operation may degrade database performance until the operation completes.

1. Install and cable the storage expansion shelf, but do **not** power on the expansion shelf.

> **Caution:**
>
> Review cabling instructions carefully to ensure that you have carried out cabling correctly. Incorrect connections can cause data loss when adding a storage expansion shelf to Oracle Database Appliance with existing databases.

2. Verify that both nodes plus the base storage shelf are up and running. Log into each server node and run the `odacli validate-storagetopology` command to confirm that the base configuration cabling is correct.

```
# odacli validate-storagetopology
 ...
      INFO   : Check if JBOD powered on
  SUCCESS    : JBOD : Powered-on
      INFO   : Check for correct number of EBODS(2 or 4)
  SUCCESS    : EBOD found :
2

                INFO       : Check for overall status of cable validation
on Node0
```

```
  SUCCESS   : Overall Cable Validation on Node0
  SUCCESS   : JBOD Nickname set correctly : Oracle Database Appliance - E0
```

Run the command to confirm that the two server nodes are properly cabled to the base storage shelf and all disks are online, with a good status, and added to the existing disk groups on both nodes. If there any failures, then fix the cabling before proceeding to the next step.

> **Note:**
>
> If the output shows that `EBOD found` is 2, then you only have the base storage shelf. If `EBOD found` is 4, then you have a base storage shelf and an expansion shelf.

> **Note:**
>
> If you add a new JBOD fresh from the factory, then the output of the `odacli validate-storagetopology` command is:
>
> ```
> # odacli validate-storagetopology
>  ...
> WARNING : JBOD Nickname is incorrectly set to :
> ```

3. Power on the storage expansion shelf and wait for 20 minutes before issuing the CLI command for storage expansion.
4. Log in to each server node and run the `odacli validate-storagetopology` command to validate the storage cabling and confirm that the new storage shelf is recognized.

```
# odacli validate-storagetopology

  INFO    : Check if JBOD powered on
  SUCCESS : 2JBOD : Powered-
on
  INFO    : Check for correct number of EBODS(2 or 4)
  SUCCESS : EBOD found :
4
   ...
   ...

   INFO    : Check for overall status of cable validation on Node0
   SUCCESS : Overall Cable Validation on Node0
   SUCCESS : JBOD0 Nickname set correctly : Oracle Database Appliance - E0
   SUCCESS : JBOD1 Nickname set correctly : Oracle Database Appliance -
E1
```

If you add a new JBOD fresh from the factory, then the output of the `odacli validate-storagetopology` command is:

```
# odacli validate-storagetopology
 ...
WARNING : JBOD Nickname is incorrectly set to :
```

Look for the following indicators that both storage shelves are recognized:

- When there are two shelves, the JBOD (just a bunch of disks) is numbered. For example:

  ```
  SUCCESS : 2JBOD : Powered-on
  ```

- When both shelves are recognized, the `EBOD found` value is 4.

  ```
  SUCCESS : EBOD found : 4
  ```

- When the expansion shelf is cabled properly, the nickname is `E1`. For example:

  ```
          SUCCESS : JBOD0 Nickname set correctly : Oracle Database
  Appliance - E0
          SUCCESS : JBOD1 Nickname set correctly : Oracle Database
  Appliance - E1
  ```

Fix any errors before proceeding.

5. Run the `odaadmcli show disk` command to ensure that all disks in the expansion shelf are listed, are online, and are in a good state.

   ```
   # odaadmcli show disk
   ```

   When all disks are online and in a good state, proceed to the next step.

6. Run the `odaadmcli show enclosure` command to check the health of components in expansion shelf.

   ```
   # odaadmcli show enclosure
   ```

7. Run the `odaadmcli expand storage` command.

   ```
   # odaadmcli expand storage -ndisk 24 -enclosure 1

   Running precheck, it may take a few minutes.
   Precheck passed.
   Check the progress of expansion of storage by running 'odaadmcli show disk'
   Waiting for expansion to finish. It may take several minutes to complete
   depending upon the number of disks being expanded

   odaadmcli expand storage command completed.
   Check status of newly expanded disk in OAK by running 'odaadmcli show disk'
   All of them must be ONLINE status. Check status of newly expanded disk in
   ASM
   by running 'asmcmd lsdsk -p', mount status must be CACHED and header status
   should be MEMBER for all the newly added disks. If the status of newly
   ```

```
added
disks in OAK and ASM are different then expand storage command failed.
```

It takes approximately 30 to 40 minutes to add all of the disks to the configuration.

8. Use the `odaadmcli show validation storage errors` command to show hard storage errors.

   Hard errors include having the wrong type of disk inserted into a particular slot, an invalid disk model, or an incorrect disk size.

   ```
   # odaadmcli show validation storage errors
   ```

9. Use the `odaadmcli show validation storage failures` command to show soft validation errors.

   A typical soft disk error would be an invalid version of the disk firmware.

   ```
   # odaadmcli show validation storage failures
   ```

10. Run the `odacli describe-component` command to verify that all firmware components in the storage expansion are current.

    ```
    # odacli describe-component
    ```

11. If needed, update the storage shelf and then run the `odacli describe-component` command to confirm that the firmware is current.

    ```
    # odacli update
    # odacli describe-component
    ```

11

Managing Networks

View the public and private networks that are configured on your Oracle Database Appliance.

- About Network Infrastructure and VLANs on Oracle Database Appliance
  Learn about networks and virtual local area networks (VLANs) on the appliance.
- About Oracle Database Appliance X9-2 Network Interfaces
  Learn about onboard network interfaces for Oracle Database Appliance.
- Viewing Configured Networks and Network Interfaces
  Use the Browser User Interface to display a list of configured networks, network details, and interfaces.
- Creating a Network
  Understand how you can use the Browser User Interface to create a network for the appliance.
- Creating a Physical Network
  Understand how you can use the Browser User Interface to create a physical network on the unused network interface.
- Updating a Network
  Understand how you can use the Browser User Interface to update a network and revise the IP address, subnet mask, gateway, or type of network.
- Adding a Network Interface
  Understand how you can add a configured network interface.
- Deleting a Network Interface
  Understand how you can delete a configured network interface.
- Deleting a Network Interface Using Browser User Interface
  Understand how you can delete a configured network interface using the Browser User Interface.
- Deleting a Network Using the Browser User Interface
  Understand how you can delete a configured network using the Browser User Interface.

About Network Infrastructure and VLANs on Oracle Database Appliance

Learn about networks and virtual local area networks (VLANs) on the appliance.

Oracle Database Appliance has two dual-port public network interfaces (either copper or fiber), which are bonded. For dual-port networks cards, the bonded network interface used for primary public network is always `btbond1`. You can, optionally, configure upto six bonded interfaces, `btbond1` to `btbond6`. If you use network cards with four ports, then you can select either `btbond1` or `btbond2` as the public network interface.

You can use the Browser User Interface to display all physical and virtual networks. You can use ODACLI commands or the Browser User Interface to create, update, and deleted networks.

Virtual Local Area Networks (VLANs)

Oracle Database Appliance supports multiple virtual local area networks (VLANs) on the same network port or bond. VLANs are multiple logical networks that are created from a single physical network switch port, providing network security isolation for multiple workloads that share a common network. For example, application, backup, and management networks. Each VLAN acts as an independent logical network operating with other VLANs over the same physical connection. The VLAN tag associated with the data packet and network define the network. You can create a collection of isolated networks to enhance network security and bandwidth and keep data packets separated.

The network interfaces differ, depending on your Oracle Database Appliance hardware. The VLAN is created on `btbond0` in single-node and high-availability platforms. In all cases, connections to user domains are through the selected interfaces. A switch that supports tagged VLANs uses VLAN IDs to identify the packet, including to which network the packet belongs.

> **Note:**
>
> To use VLANs with Oracle Database Appliance, you must configure the VLANs before you deploy the appliance.

The Browser User Interface enables you to create, list, and delete VLANs on the appliance. For high-availability systems, you can use the Browser User Interface to create a VLAN on both nodes of the appliance. To create a VLAN on a specific node, use the command-line interface.

For a bare metal deployment, use the Browser User Interface or `odacli` commands to manage the following types of VLANs:

- Data Guard: For Oracle Data Guard.
- Database: For Oracle Database.
- Backup: For backup operations.
- Management: For management traffic.
- Other: For usage defined by the customer. For example, for applications.

The public VLAN is setup when you configure the first network using the command `odacli configure-firstnet`. You can set up only one public VLAN. Use the command-line interface to create other VLANs. For high-availability systems, the IP addresses for Node 0 and Node 1 cannot be the same. Oracle Database Appliance does not support Dynamic Host Configuration Protocol (DHCP) to provide IP addresses, subnet mask and default gateway.

About Oracle Database Appliance X9-2 Network Interfaces

Learn about onboard network interfaces for Oracle Database Appliance.

The network interfaces differ, depending on your Oracle Database Appliance hardware. In all cases, connections to user domains are through the selected interfaces. Two ports inside a network card are used to create an active-backup mode bonding interface.

The following table lists the default network interfaces for Oracle Database Appliance bare metal systems on Oracle Database Appliance X9-2-HA.

Table 11-1 Network Interfaces for Oracle Database Appliance X9-2-HA Bare Metal Deployment

PCIe Network Card Slot	PCIe Network Port	Network Interfaces	Network Bonds	IP Addresses
PCIe Slot 1 (Interconnect)	2 x 10/25GbE SFP28 ports	p1p1, p1p2	icbond0	Node 0: 192.168.16.24 Node 1: 192.168.16.25
PCIe Slot 6 (required)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p6p1, p6p2 (p6p3, p6p4)	btbond1 (btbond2)	Can be assigned during provisioning or post deployment.
PCIe Slot 4 (optional)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p4p1, p4p2 (p4p3, p4p4)	btbond3 (btbond4)	Can be assigned during provisioning or post deployment.
PCIe Slot 8 (optional)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p8p1, p8p2 (p8p3, p8p4)	btbond5 (btbond6)	Can be assigned during provisioning or post deployment.
On-board	1GBase-T	em1	Not applicable	Assigned post deployment

The following table lists the default network interfaces for Oracle Database Appliance bare metal systems on Oracle Database Appliance X9-2S and X9-2L.

Table 11-2 Network Interfaces for Oracle Database Appliance X9-2S and X9-2L Bare Metal Deployment

PCIe Network Card Slot	PCIe Network Port	Network Interfaces	Network Bonds	IP Addresses
PCIe Slot 6	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p6p1, p6p2 (p6p3, p6p4)	btbond1 (btbond2)	btbond1 and btbond2 can be assigned during deployment.
PCIe Slot 4 (optional for Oracle Database Appliance X9-2L)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p4p1, p4p2 (p4p3, p4p4)	btbond3 (btbond4)	Can be assigned during deployment. If not assigned during deployment, then it can be assigned post deployment.
PCIe Slot 7 (optional for Oracle Database Appliance X9-2S)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p7p1, p7p2 (p7p3, p7p4)	btbond3 (btbond4)	Can be assigned during deployment. If not assigned during deployment, then it can be assigned post deployment.

Table 11-2 (Cont.) Network Interfaces for Oracle Database Appliance X9-2S and X9-2L Bare Metal Deployment

PCIe Network Card Slot	PCIe Network Port	Network Interfaces	Network Bonds	IP Addresses
PCIe Slot 8 (optional for Oracle Database Appliance X9-2S)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p8p1, p8p2 (p8p3, p8p4)	btbond5 (btbond6)	Can be assigned during deployment. If not assigned during deployment, then it can be assigned post deployment.
On-board	1GBase-T	em1	Not applicable	Assigned post deployment

Viewing Configured Networks and Network Interfaces

Use the Browser User Interface to display a list of configured networks, network details, and interfaces.

1. Click the **Appliance** tab in the Browser User Interface.
2. Click the **Network** tab in the left navigation to display a list of configured networks and configured databases.

   When the appliance is a high-availability system, the network details for **Node0** and **Node1** display on the page.
3. Click **Show Interfaces** in the upper left corner of the Network page to display a list of network interfaces.

   If the appliance is a high-availability system, then the network interface details for **Node0** and **Node1** display on the page.
4. Click **Show Networks** to go back to the network view.

Creating a Network

Understand how you can use the Browser User Interface to create a network for the appliance.

When naming a network, you can use alphanumeric characters, dashes (-) and underscores (_). The network name cannot exceed 40 characters and must begin with an alpha character.

1. Click the **Appliance** tab in the Browser User Interface.
2. Click the **Network** tab in the left navigation to display a list of configured networks.
3. If the appliance is a high-availability system, then the Browser User Interface helps you create the network for both nodes at the same time.
4. Click **Create Network** in the upper right corner.
5. Enter the network information in the required fields. The Interface field has a drop-down menu with available choices. The Subnet Mask field has a drop-down menu with available choices and a search box. Optionally, enter a Gateway IP address and select an option from the Network Type list. The network type helps to identify if the network is for backup, database, dataguard, management, private, or other.

If the appliance is a high-availability system, then you must also specify the SCAN (Oracle Single Client Access Name) details such as name and IP address, and VIP (Virtual IP) name and address.

6. Click **Create**.

Creating a Physical Network

Understand how you can use the Browser User Interface to create a physical network on the unused network interface.

1. Click the **Appliance** tab in the Browser User Interface.
2. Click the **Network** tab in the left navigation to display a list of configured networks.
3. If the appliance is a single-node system, then click **Create Network** in the upper right corner. If the appliance is a high-availability system, then go to Step 4.
    a. Enter the network information in the required fields. The Interface field has a drop-down menu with available choices. The Subnet Mask field has a drop-down menu with available choices and a search box. Optionally, enter a Gateway IP address and select an option from the Network Type list. The network type helps to identify if the network is for backup, database, dataguard, management, private, or other.
    b. Click **Create**.
4. If the appliance is a high-availability system, then the Browser User Interface helps you create the network for both nodes at the same time. Click **Create Network** in the upper right corner.
    a. Enter the network information in the required fields. The Interface field has a drop-down menu with available choices. The Subnet Mask field has a drop-down menu with available choices and a search box. Optionally, enter a Gateway IP address and select an option from the Network Type list. The network type helps to identify if the network is for backup, database, dataguard, management, private, or other.
    b. Click **Create**.

Updating a Network

Understand how you can use the Browser User Interface to update a network and revise the IP address, subnet mask, gateway, or type of network.

1. Click the **Appliance** tab in the Browser User Interface.
2. Click the **Network** tab in the left navigation to display a list of configured networks.

    When the appliance is a high-availability system, tabs named **Node0** and **Node1** display in the right corner, below the **Refresh** button. Click a tab to display network details for each node.
3. If the appliance is a high-availability system, click **Node0** or **Node1** to display the node where the network that you want to edit resides.
4. Expand the **Actions** menu, then click **Update** for the network that you want to edit.
5. Update the information in the IP Address, Subnet Mask, and Gateway fields and select a network type from the list, as needed, then click **Update**. For High-Availability deployments, you can also update the Virtual IP Address and Oracle Single Client Access Name (SCAN).

Adding a Network Interface

Understand how you can add a configured network interface.

This procedure explains how to add a network interface card on Oracle Database Appliance X8-2.

1. Power off the appliance.
2. Add the network interface card.
3. Power on the appliance.
4. The network configuration files are automatically updated for network bonding and configuration changes.

Deleting a Network Interface

Understand how you can delete a configured network interface.

This procedure explains how to delete a network interface card on Oracle Database Appliance X8-2. You cannot delete the network interface card in slot 7. You also cannot delete the `btbond1` and `btbond2` networks.

1. Disconnect all networks for the network interface card from all applications on the appliance.
2. Power off the appliance.
3. Remove the network interface card.
4. Power on the appliance.
5. Run the command `odacli delete networkinterface` to clean up the network configuration files.

```
odacli delete networkinterface network_interface_name

[root@oak ~]# odacli delete-networkinterface btbond3
Network btbond3 has been deleted successfully

[root@oak ~]# odacli delete-networkinterface btbond4
Network btbond4 has been deleted successfully
```

Deleting a Network Interface Using Browser User Interface

Understand how you can delete a configured network interface using the Browser User Interface.

1. Click the **Appliance** tab in the Browser User Interface.
2. Click the **Network** tab in the left navigation to display a list of configured networks.
3. Click the **Show Interfaces** tab in the right navigation to display a list of configured network interfaces.

4. Expand the **Actions** menu, then click **Delete** for the network interface that you want to delete.
5. Confirm the action when prompted.

Deleting a Network Using the Browser User Interface

Understand how you can delete a configured network using the Browser User Interface.

1. Click the **Appliance** tab in the Browser User Interface.
2. Click the **Network** tab in the left navigation to display a list of configured networks.

   When the appliance is a high-availability system, tabs named **Node0** and **Node1** display in the right corner, below the **Refresh** button. Click a tab to display network details for each node.
3. If the appliance is a high-availability system, then click **Node0** or **Node1** to display the node where the network that you want to delete resides.
4. Expand the **Actions** menu, then click **Delete** for the network that you want to delete.
5. Confirm the action when prompted.

12

Backup, Restore and Recover Databases

Review backup, restore, and recovery operations for your databases.

- About Database Backup and Recovery Options
  Backup your databases in Oracle Database Appliance and use the restore and recover features to restore a database to the same or another system or to recover database files.
- Configuring NFS Backup and Restore on Oracle Database Appliance
  To backup databases to Network File System (NFS) location or to restore the database from NFS location, you must configure the NFS backup location to ensure that Oracle Database Appliance can perform read and write operations.
- Configuring Agent Proxy Settings for Object Store Access
  If the Object Store IP address is accessible only through proxy setup by the Oracle Database Appliance server, then define the proxy setting for the agent, so that the agent can access the Object Store.
- Creating a Database Backup Policy
  Understand how to create a policy to backup a database to an internal or external Oracle Fast Recovery Area (FRA) location or to the Object Store using the Browser User Interface.
- Updating a Database with a Backup Policy
  Attach a backup policy to a database to define the database backup attributes and destination.
- Updating a Backup Policy
  Use the Browser User Interface to update the recovery window, change the ObjectStore Credential, or change the crosscheck option for a backup policy.
- Backing Up a Database Using the Browser User Interface
  Apply a backup policy to a database, create a manual backup, update a database backup schedule, or update an archive log backup scheduler.
- Viewing Backup Reports
  Understand how you can use the Browser User Interface to view backup reports.
- Recovering a Database Using the Browser User Interface
  Understand the recovery options available for recovering a database.
- Deleting a Backup Using the Browser User Interface
  Use the Browser User Interface to delete database backups.
- Deleting a Backup Policy
  Use the Browser User Interface to delete a database backup policy.
- Using the CLI to Backup, Restore, and Recover
  Use the command-line interface to backup, restore, and recover databases.

About Database Backup and Recovery Options

Backup your databases in Oracle Database Appliance and use the restore and recover features to restore a database to the same or another system or to recover database files.

It is important to create a backup recovery strategy to protect the databases in Oracle Database Appliance from data loss. A user error or a physical problem with a disk can cause a failure of a read or write operation on a disk file required to run the database. You can create a backup policy in the Browser User Interface or from the command-line interface.

The following backup options are available:

- Backup to an Oracle Fast Recovery Area (FRA) disk (Internal FRA)
- Backup to Oracle Cloud Infrastructure Object Storage (Oracle Object Storage)
- Backup to a Network File System (NFS) location (External FRA)

You can use the backup, restore, and recovery features with databases that have backup policy attached to them in the system. You can choose to manage your backups, and set up your backup, restore, and recovery options.

The backup feature enables you to create a backup policy with your backup parameters, such as backup location, recovery window, enable and disable crosscheck, and ObjectStoreSwift Object details. For new and existing databases, you can create and attach a backup policy to a database to perform backup. After a backup policy is attached to a database, the DCS agent automatically schedules the database backup and archive logs backup, for the database. By default, the frequency of the database backup schedule is once a day and the frequency of archivelog backup schedule is 30 minutes. The database backup scheduler performs level 0 backup every Sunday and a level 1 backup Monday through Saturday. However the level 0 backup day can be changed from the default Sunday to any day of the week. The database backup scheduler and archive logs backup schedulers can be disabled or have their frequencies changed.

The DCS agent generates and saves a backup report for each backup. The backup report contains the metadata required to recover or restore a database.

Oracle Database Appliance uses the Oracle Recovery Manager (RMAN) to recover the database. You can recover from a backup report, a point-in-time (PITR), a System Change Number (SCN), or from the latest backup. When a backup report is used for recovery, it performs an SCN recovery, by reading the SCN value from the backup report. For recovery of data block, tablespaces, data files, PDBs, see the *Oracle Database Backup and Recovery Reference Guide*.

Backup Policy

The backup policy defines the backup details. When you create a backup policy, you define the destination for the database backups, either Internal FRA (Disk) or External FRA (NFS location), or Oracle Object Storage, and you define the recovery window.

If you use the Oracle Object Storage for backup and recovery, you must have an ObjectStore Resource ID or Name.

You can assign the backup policy to the database when you create the database. You can also attach a backup policy to an existing database or update a backup policy.

You can specify compression algorithm while creating the backup policy. The supported compression algorithms are BASIC, LOW, MEDIUM, and HIGH. For Objectstore backup policy, you can specify any compression algorithm. By default, if you do not specify the compression algorithm while creating the Objectstore backup policy, then MEDIUM compression algorithm is used. By default, if you do not specify the compression algorithm while creating the Disk or NFS backup policy, then BASIC compression algorithm is used. Refer to the *Oracle Database Licensing Information User Manual* for license considerations when you use LOW, MEDIUM, or HIGH compression algorithms for Disk or NFS backup policy.

Backup Levels

Specify the backup level when you take a backup. The supported back levels are:

- **Level 0**: This backup captures all blocks in the datafile. By default, level 0 backups occur every Sunday. You can customize your backup settings through the Browser User Interface or the command-line. A level 0 backup is a full backup and is used as the parent for a level 1 backup.
- **Level 1**: This backup is an incremental backup that captures only those blocks that change between backups in each datafile. By default, a level 1 backup occurs daily, Monday through Saturday, and captures images of each block in a datafile that changed. Level 1 backups are differential backups. The Level 1 backup includes blocks that were changed from the most recent level 1 or level 0 backup.
- **LongTerm**: This backup is a long-term, or archival backup. Use LongTerm to create a backup that is all-inclusive and exempt from the backup retention policy. The backup is all-inclusive because every file needed to restore and recover the database is backed up to a single location. The backup is exempt from the retention policy that is defined in the recovery window. For long term backups, you must use External FRA (NFS location) or Objectstore backup policy.
- **Archivelog**: This is a backup of all archive logs not yet backed up to the backup destination, such as Internal FRA, External FRA (NFS location), or Object Store.

When a database is modified with a backup configuration, database backup schedule and archive logs backup schedules are created automatically. Automatic database backup schedule creates a level 0 backup on Level 0 Backup Day, by default, on Sunday, and level 1 on the remaining days of the week. Also, automatic archive logs backup schedules creates archive logs backup for the database with a default frequency of 30 minutes. These backup schedules can be edited to enable or disable or change the frequency using cron expression.

Backup to Disk

When you backup to disk, you are using the Oracle Fast Recovery Area (FRA) of the disk. FRA is a storage directory that allows backup and recovery operations on Oracle databases. The directory contains archived redo logs, backup pieces and copies, and the control file.

Advantages to backing up to disk are as follows:

- Quick recovery from backups stored on disk
- Automate management of recovery related files (like archived logs)

> **Note:**
>
> - When you backup to disk, the data for non-TDE database is not encrypted and RMAN backup encryption are not allowed. In addition, backup to disk requires a lot of disk space (up to two to three times the size of the database).
> - If the database is TDE-enabled, then the RMAN backup encryption password is not required, irrespective of the backup destination.

The following is a high-level overview of the workflow for backup and recovery when using a disk:

1. Create a backup policy to define the backup destination (disk) and recovery window for the disk destination.
2. Create a database and attach the backup policy to the database. When the backup policy is associated with a database, archive logs backups and database backups are automatically created.
3. Manage obsolete backups.
4. Recover the database from a backup on the disk.

Backup to External FRA (Network File System)

Specifying a Network File System (NFS) location for backups, enables you to safely and securely store or retrieve a database backup at any time.

The advantages of using NFS backup location are:

- With NFS, you can access remote files, locally, by mounting local file systems on a network.
- Provides a centralized backup location, without requiring manual refresh.
- Supports enhanced security options using firewalls and Kerberos.
- Support for encrypted backups using passwords. You optionally, set the backup password, and provide the same during restore operation.
- The RMAN Backup Encryption password is applicable for databases that do not have TDE enabled. If the database has TDE enabled, then the RMAN backup password is not required, irrespective of the backup destination.

The following is a high-level overview of the workflow for backup and recovery when using an NFS location:

1. Create a mount point for the NFS location.

   The mount point must be accessible from both nodes. The `oracle` user must have read/write permissions to the NFS location. If the request is being made to a remote server, ensure that it is running. See *Creating a Mount Point for NFS Backup Location* for the procedure to create a mount point.
2. Create a backup policy specifying the NFS location, and the recovery window for NFS destination.
3. Associate the database with this backup policy, either during database creation, or by updating the backup policy for the database.
4. Backups are automatically scheduled, and you can also run manual backups. You can specify manual backup options in the Browser User Interface or using ODACLI commands.
5. Manage obsolete backups.
6. Restore from backup. When you save your backups to NFS, you can restore the database to the same system or to a different system.
7. Recover the database from a backup on NFS.

Backup to Oracle Object Storage

Oracle Object Storage is a highly secure, scalable storage platform that enables you to safely and securely store or retrieve a database backup at any time. Data and the associated metadata are stored as an object in a logical container (bucket). Oracle Database Appliance creates a Backup Report and a unique ID to identify each backup.

Oracle Object Storage is tightly integrated with Oracle Cloud Infrastructure Identity and Access Management. The data, user-defined metadata associated with the data, and encryption keys are automatically encrypted using 256-bit Advanced Encryption Standard (AES- 256). The encryption keys are rotated frequently and cannot be disabled. By default, only authenticated users that have explicitly been granted access to specific resources can access data stored in Oracle Object Storage. Data is uploaded and downloaded over SSL endpoints using the HTTPS protocol.

When you backup to Oracle Object Storage, you are using your backup policy and your Oracle ObjectStoreSwift credentials. The credentials are stored in an encrypted Oracle wallet. You can attach the credentials to one or more backup policies.

> **Note:**
>
> Credentials are validated when you create the backup policy that uses Oracle Object Storage. The credentials are not validated against the URL and tenancy.

Advantages to using Oracle Object Storage are as follows:

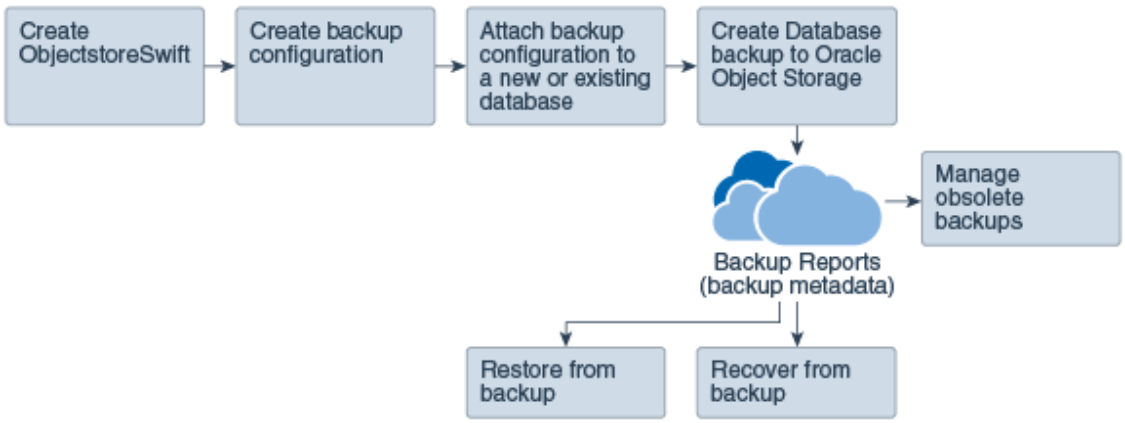
- Backups are encrypted using the password for non-TDE databases. For TDE-enabled databases, backups are encrypted by default and do not require the RMAN backup encryption password to be specified separately.

  > **Caution:**
  >
  > Save the password that you use during backups in a safe location, you must provide the password during recovery operations.

- Quick recovery from backups. With the Oracle Object Storage, you can safely and securely use the Browser User Interface to store or retrieve data directly from within the cloud platform, at any time.
- Automate management of recovery related files (like archived logs)
- Oracle Object Storage is not directly accessed by the operating system, instead it uses APIs to access the storage at the application level.

The following is a high-level overview of the workflow for backup and recovery when using Oracle Object Storage:

1. Create an Object Store object with your credentials.
2. Create a backup policy to define the backup location (Object Store), Object Store Credentials Name, the container (bucket) name where you want to store backups, and the recovery window.
3. Create a database and attach the backup policy either as a part of creating the database, or update the database with backup policy after creating the database. When the backup policy is associated with a database, backups are automatically created.
4. Manage obsolete backups.
5. Restore from backup. When you save your backups to Oracle Object Storage, you can restore the database to the same system or to a different system.
6. Recover the database from a backup on Oracle Object Storage.



> **Note:**
>
> Before you can use this backup option, you must create an Oracle Cloud Infrastructure Object Storage account. When you create the account, you define the account credentials, end point URL, and bucket. Go to the Oracle Store and sign up for Oracle Cloud Infrastructure Object Storage.

Backup Reports

A backup report is generated for each backup and is similar to, but does not replace, a recovery catalog. The report contains the Oracle Database Appliance information and metadata needed to restore or recover a database. The Backup Report is designed to keep track of all of the information required to restore or recover a database.

The following is an example of a level 1 backup report:


```
{
  "id" : "2d82460c-d648-4e75-8c7d-72cc90bc442a",
  "dbResId" : "b5fc646e-01a6-4c8b-8286-7633346c4329",
  "tag" : null,
  "dbId" : "2717054291",
  "dbName" : "HRDbO",
  "dbUniqueName" : "HRDbOu",
  "backupType" : "REGULAR-L1",
  "keepDays" : null,
  "backupLocation" : "https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/dbaasimage/backupbucket",
  "cfBackupHandle" : "c-2717054291-20171108-04",
  "spfBackupHandle" : "c-2717054291-20171108-04",
  "pitrTimeStamp" : "November 08, 2017 12:43:14 PM UTC",
  "pitrSCN" : "1175058",
  "resetLogsTimeStamp" : "November 08, 2017 09:55:34 AM UTC",
  "resetLogsSCN" : "1112268",
  "oraHomeVersion" : "12.2.0.1.170814 (26723265, 26609817)",
  "sqlPatches" : "25811364,26609817",
  "backupLogLoc" : "https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/dbaasimage/backupbucket/scaoda702c1n1/rmanlog/
HRDbOu/2717054291/2017-11-08/rman_backup_2017-11-08_12-42-41.0545.log",
  "tdeWalletLoc" : null,
  "dbConfigLoc" : "https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/dbaasimage/backupbucket/scaoda702c1n1/dbconfig/
HRDbOu/2717054291/2017-11-08/DBCONFIG_TAG20171108T124407_2017-11-08_12-44-07.0533.tar.gz",
  "name" : "Backup_Report_HRDbO",
  "createTime" : "November 08, 2017 12:42:08 PM UTC",
  "state" : {
    "status" : "CONFIGURED"
  },
  "updatedTime" : "November 08, 2017 12:44:12 PM UTC",
  "backupReportLogDetail" : "https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/dbaasimage/backupbucket/scaoda702c1n1/
rmandetaillogreport/HRDbOu/2717054291/2017-11-08/rman_list_backup_detail_2017-11-08_12-44-04.0362.log",
  "dbInfo" : {
    "dbClass" : "OLTP",
    "dbType" : "RAC",
    "dbShape" : "odb1",
    "dbEdition" : "EE",
    "dbStorage" : "ASM"
  },
  "dbDataSize" : "1542M",
```

```
  "dbRedoSize" : "16403M"
}
```

About Backup of TDE Wallet

When the database backup is performed, the TDE wallet backup is also taken. The database backup can be of any type like regular-L0, regular-L1, longterm and archivelog. Use the `odacli list-tdebackupreports` command to view the latest TDE wallet backup in a backup destination. For example, if a TDE-enabled database named `tdedb1` was attached with Disk, and then NFS, and later Oracle Object Storage backup configuration, and the database was backed up to each of these backup destination eventually, then the `odacli list-tdebackupreports` command lists three entries of TDE wallet backup corresponding to Disk, NFS and Oracle Object Storage backup destination, respectively, for the database `tdedb1`. The reason for selecting the latest copy is that the TDE wallet is a cumulative wallet, that is, all previous changes, such as re-key operation is available in the latest TDE wallet itself.

About Recovery Options

The recovery in Oracle Database Appliance always performs a full database restore or recover. The recovery options are LATEST, PITR, SCN, and BackupReport.

The following types of recovery are available:

- LATEST: Performs a complete recovery (valid backups and all required archived logs and online redo logs must be available)
- PITR: Point-in-Time Recovery that performs a recovery to a specified timestamp within the current incarnation of database
- SCN: SCN-based recovery that performs a recovery is an incomplete recovery to a specified SCN within the current incarnation of database.
- BackupReport: Performs a recovery based on the Backup Report SCN. The BackupReport option is similar to SCN, except that the recovery uses the Backup Report SCN instead of a specified SCN.

Point-in-Time Database Recovery

Point-in-Time Recovery (PITR) recovers the database to a specific point-in-time. You must specify the timestamp for this type of recovery.

A point-in-time recovery is often used for the following situations:

- You want to recover the database to an SCN before a user or administrative error.
- Complete database recovery failed because all necessary archived redo logs were not available.
- A database upgrade fails.
- You are creating a test database or a reporting database from production database backups.

System Change Number Recovery

A system change number (SCN) is a stamp that defines a committed version of a database at a point in time. Oracle assigns every committed transaction a unique SCN. You can recover a database to the latest time or to an SCN.

The database is restored from backups created before the target time, and then applies incremental backups and logs to re-create all changes between the time of the data file backups and the end point of recovery. When the end point is specified as an SCN, the

database applies the redo logs and stops after each redo thread or the specified SCN, whichever occurs first. When the end point is specified as a time, the database internally determines a suitable SCN for the specified time and then recovers to this SCN.

SCN is a primary mechanism to maintain data consistency in Oracle database. With each execution, there is an increase in the count of the SCN. The following are some examples of areas that use SCN:

1. Redo records. Every redo record has an SCN version of the redo record in the redo header (redo records can have non-unique SCN). Given redo records from two threads (as in the case of RAC), recovery will order them in SCN order, essentially maintaining a strict sequential order.
2. Data block. Every data block has block SCN (block version). In addition to that, a change vector in a redo record also has expected block SCN. This means that a change vector can be applied to one and only version of the block. Code checks if the target SCN in a change vector is matching with the block SCN before applying the redo record. If there is a mismatch, corruption errors are thrown.
3. Read consistency. Every query has query environment which includes an SCN at the start of the query. A session can see the transactional changes only if that transaction commit SCN is lower then the query environment SCN.
4. Commit. Every commit generates an SCN (commit SCN) that marks a transaction boundary. Group commits are possible too.

Restoring Archive Logs

The database auto backup policies - Regular-L0, and Regular-L1 delete archivelog, which could be essential in later process, such as in configuration or reinstate operations for Oracle Data Guard. Oracle Database Appliance supports restore of archive logs for a database. You can choose to restore archive logs from one SCN to another SCN, or from one timestamp to another timestamp.

Run the command `odacli restore-archivelog` specifying the SCN or timestamp period.

Obsolete Database Backups

It is important to remove older backups once they become obsolete. The type of backup and the recovery window that you define in the database backup policy determines when database backups become obsolete.

For level 0, level 1, and archivelog backups, run the command `odacli delete-backup` for a given Database Resource Identifier to delete obsolete backups and backup reports based on the defined recovery window.

For Longterm backups, provide the corresponding backup report as an input and use the command `odacli delete-backup` to delete the backups from Oracle Object Storage or NFS backup location and the back report you specify in the command.

Typically, older level 0 and level 1 database backups are considered obsolete based on the defined recovery window and those obsolete backups are deleted when you run the `odacli delete-backup` command. The defined recovery window is as follows:

- Disk: 1-14 days
- Object Storage: 1-30 days

Starting with Oracle Database Appliance release 19.15, on attaching a backup policy to a database, the DCS agent creates a scheduler for deleting the database backup. It deletes the obsolete backups and backup reports for the database. By default, the frequency of this delete

backup scheduler is once a week. It runs on every Monday to delete obsolete backups and backup reports of the database. The delete backup scheduler can be edited to disable it or to change its frequency.

Related Topics

- Configuring an NFS Backup Location
  Configure Network File System (NFS) backup location, to set up the NFS backup and mount the NFS client location on Oracle Database Appliance.

Configuring NFS Backup and Restore on Oracle Database Appliance

To backup databases to Network File System (NFS) location or to restore the database from NFS location, you must configure the NFS backup location to ensure that Oracle Database Appliance can perform read and write operations.

Following are the steps to configure the NFS location, backup database to NFS location, and restore database as a standby database from the NFS backup location.

- Configuring an NFS Backup Location
  Configure Network File System (NFS) backup location, to set up the NFS backup and mount the NFS client location on Oracle Database Appliance.
- Backing Up the Database to NFS Location
  To backup a database to NFS location on Oracle Database Appliance, you must create an NFS backup configuration object and then attach the same to the database.
- Restoring the Database from NFS Backup
  Understand the steps to restore a database from NFS location.

Configuring an NFS Backup Location

Configure Network File System (NFS) backup location, to set up the NFS backup and mount the NFS client location on Oracle Database Appliance.

An NFS setup involves an NFS server location and an NFS client location. While performing database backup operation, it is recommended that you configure the NFS server location on a different system, than the system where the source database, that is, the database to be backed up, is running. This ensures the database backups are not lost, even if the system where the source database is running, is corrupted or lost. However, the NFS client location must be configured on the same machine where the database is running, to perform database backup operation. Also, while performing database restore operation, the NFS client location must be configured on the same system where the database must be restored.

> **Note:**

- An NFS backup configuration object must be created and attached to the database. This NFS backup location is a mandatory parameter to create a NFS backup configuration object.
- For TDE-enabled database, Oracle Database Appliance requires the backup path to be different for database and TDE wallet, as per security guidelines. If the source database is TDE-enabled, then configure another NFS client location following the steps in this topic.

Run the following steps on the NFS server:

1. Create an Oracle ADVM volume and format the volume as Oracle ACFS, on any node, for example, `node0`, of a different Oracle Database Appliance system, as the `grid` operating system user.

```
[root@***** ~]# su - grid

[grid@***** ~]# asmcmd

ASMCMD> volcreate -G data -s 100G backup

ASMCMD> volinfo -G data backup
Diskgroup Name: DATA
Volume Name: BACKUP
Volume Device: /dev/asm/backup-322
State: ENABLED
Size (MB): 102400
Resize Unit (MB): 64
Redundancy: HIGH
Stripe Columns: 8
Stripe Width (K): 4096
Usage:
Mountpath:
ASMCMD> exit
[grid@***** ~]$ mkfs -t acfs /dev/asm/backup-322
mkfs.acfs: version = 19.0.0.0.0
mkfs.acfs: on-disk version = 46.0
mkfs.acfs: volume = /dev/asm/backup-322
mkfs.acfs: volume size = 107374182400 ( 100.00 GB )
mkfs.acfs: Format complete.

[grid@***** ~]$ exit
```

2. Create an operating system directory as the `root` user to export Oracle ACFS volume created in the previous step.

```
[root@***** ~]# mkdir /nfs_backup_server
```

3. Register the file system with Oracle Clusterware and start the file system as the `root` user.

```
[root@***** ~]# /u01/app/19.13.0.0/grid/bin/srvctl add filesystem -
d /dev/asm/backup-322 -path /nfs_backup_server -mountowner oracle -
mountgroup asmadmin
[root@***** ~]#/u01/app/19.13.0.0/grid/bin/srvctl start filesystem -
d /dev/asm/backup-322
```

4. Identify the owner and group of the *DBHOME*`/bin/oracle` binary of the source database and set the directory ownership to the same owner and group. In the following example, the owner and group of the *DBHOME*`/bin/oracle` binary are `oracle` and `asmadmin` respectively. Hence, the ownership of NFS server location is set to the same. The permissions of the directory must be set to 750.

```
[root@***** ~]# ls -ltr /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1/bin/oracle
    -rwsr-s--x 1 oracle asmadmin 449234896 Sep 21 11:39 /u01/app/
```

```
odaorahome/oracle/product/19.0.0.0/dbhome_1/bin/oracle

[root@***** ~]# chown oracle:asmadmin /nfs_backup_server
[root@***** ~]# chmod 750 /nfs_backup_server
[root@***** ~]# ls -ld /nfs_backup_server
    drwxr-x--- 2 oracle asmadmin 4096 Sep 21 12:25 /nfs_backup_server
```

5. When NFS server location and NFS client location are on different systems, then there may be cases where the IDs of the `oracle` user and the `asmadmin` group may be different. If the IDs of the `oracle` user and the `asmadmin` group are the same, then do not run step 6. In the following example, the ID of `oracle` user is 1001 on the NFS server system and 1000 on the NFS client system.

```
Output of the system where the NFS server location is configured:

    [root@***** ~]# id oracle
    uid=1001(oracle) gid=1001(oinstall)
groups=1001(oinstall),1006(asmdba),1003(dba),1002(dbaoper)


    Output of the system where the NFS client location is configured:

    [root@***** ~]# id oracle
    uid=1000(oracle) gid=1001(oinstall)
groups=1001(oinstall),1006(asmdba),1003(dba),1002(dbaoper)
```

   In such a case, the ownership of the NFS server location must be set to the same ID as the `oracle` useron the system where the NFS client location is configured. The ownership of the `/nfs_backup_server` location must be set to 1000 as follows:

```
 [root@***** ~]# chown 1000:asmadmin /nfs_backup_server/
```

6. Export the NFS server location to the systems where the NFS client location is created.

```
[root@***** ~]#  cat /etc/exports
    /nfs_backup_server Node0_NFS_Client_Location_IP(rw,sync,no_root_squash)
    /nfs_backup_server Node1_NFS_Client_Location_IP(rw,sync,no_root_squash)
```

   Note that the NFS mount option must be `(rw,sync,no_root_squash)`. Starting with Oracle Database Appliance release 19.17, there is no dependency on NFS mount option `no_root_squash`, It can then be `(rw,sync)`.

7. Restart the NFS server.

```
Command on Oracle Linux 7 systems:
     [root@***** ~]# /bin/systemctl restart nfs.service

Command on Oracle Linux 8 systems:
      [root@***** ~]# /bin/systemctl restart nfs-server
```

8. Verify the mount point using the `showmount -e` command.

```
[root@***** ~]# showmount -e
    Export list for *****:
```

```
    /nfs_backup_server
Node0_NFS_Client_Location_IP,Node1_NFS_Client_Location_IP
```

Follow these steps to mount the NFS location on Oracle Database Appliance. If the source database is on an Oracle Database Appliance high-availability system, then run the following steps on both nodes.

1. Create a NFS client location as the `root` user.

```
[root@***** ~]# mkdir /nfs_backup_client
```

2. Mount the NFS client location on to the NFS server location, using the IP address of the system where the NFS server location is configured.

```
[root@***** ~]# mount -t nfs NFS_Server_Location_IP:/nfs_backup_server /
nfs_backup_client
```

Related Topics

- About Database Backup and Recovery Options
  Backup your databases in Oracle Database Appliance and use the restore and recover features to restore a database to the same or another system or to recover database files.

Related Topics

- Preparing for Database Backup to NFS Location
  Use the command-line interface to create a policy to backup a database to an External FRA (NFS Location).

Backing Up the Database to NFS Location

To backup a database to NFS location on Oracle Database Appliance, you must create an NFS backup configuration object and then attach the same to the database.

The relationship between NFS backup configuration object and database is one-to-many, that is, you can attach one NFS backup configuration object to multiple databases. Use the NFS backup location that you configured in the *Configuring an NFS Backup Location* topic to create an NFS backup configuration object.

Follow these steps to backup a database to an NFS location:

1. Create the NFS backup configuration with the NFS backup location using the `odacli create-backupconfig` command. In the following sample output, the NFS backup location, specified using the `-c` option, is `/nfs_backup_client/`, the name of the NFS backup configuration object, specified using the `-n` option is `nfsbcfg`, the destination, specified using the `-d` option is `nfs` and the retention period, specified using the `-w` option is 1 day.

   If the source database is TDE-enabled, then along with database backup location, specified using the `-c` option, the TDE wallet location must also be specified using the `-f` option. The TDE backup path must be different from the database backup path and can be configured in the same way as the database backup path.

```
#  odacli create-backupconfig -n nfsbcfg -d nfs -c /nfs_backup_client/ -w 1
    {
      "jobId" : "0a86bd50-7ac7-4d6d-af4b-1ee2c4d451d5",
      "status" : "Created",
      "message" : "backup config creation",
      "reports" : [ ],
```

```
    "createTimestamp" : "September 21, 2022 17:02:21 PM UTC",
    "resourceList" : [ {
      "resourceId" : "023eb9ac-a4d0-4071-a358-c904791d15d7",
      "resourceType" : null,
      "resourceNewType" : "BackupConfig",
      "jobId" : "0a86bd50-7ac7-4d6d-af4b-1ee2c4d451d5",
      "updatedTime" : null
    } ],
    "description" : "create backup config:nfsbcfg",
    "updatedTime" : "September 21, 2022 17:02:21 PM UTC"
  }

#  odacli describe-job -i 0a86bd50-7ac7-4d6d-af4b-1ee2c4d451d5

  Job details
  ----------------------------------------------------------------
                       ID:  0a86bd50-7ac7-4d6d-af4b-1ee2c4d451d5
              Description:  create backup config:nfsbcfg
                   Status:  Success
                  Created:  September 21, 2022 5:02:21 PM UTC
                  Message:  backup config creation

  Task Name                                Start
Time                          End Time                          Status
  ----------------------------------------
---------------------------------- ----------------------------------
----------
  Backup config metadata persist           September 21, 2022 5:02:21 PM
UTC   September 21, 2022 5:02:21 PM UTC   Success
```

2. Attach the NFS backup configuration object to the database using the `odacli modify-database` command.

```
#  odacli modify-database -n mydb -bin nfsbcfg
  {
    "jobId" : "4cd995f1-feb4-413e-a863-71e87cc97cff",
    "status" : "Created",
    "message" : "Modify database",
    "reports" : [ ],
    "createTimestamp" : "September 21, 2022 17:37:13 PM UTC",
    "resourceList" : [ {
      "resourceId" : "09b476c2-8f09-4636-9233-d8789c320b25",
      "resourceType" : "DB",
      "resourceNewType" : null,
      "jobId" : "4cd995f1-feb4-413e-a863-71e87cc97cff",
      "updatedTime" : "September 21, 2022 17:37:13 PM UTC"
    } ],
    "description" : "Modify database : mydb",
    "updatedTime" : "September 21, 2022 17:37:13 PM UTC"
  }

 # odacli describe-job -i 4cd995f1-feb4-413e-a863-71e87cc97cff

  Job details
  ----------------------------------------------------------------
                       ID:  4cd995f1-feb4-413e-a863-71e87cc97cff
```

```
                Description:  Modify database : mydb
                     Status:  Success
                    Created:  September 21, 2022 5:37:13 PM UTC
                    Message:  Modify database

    Task Name                                Start
Time                          End Time                          Status
    ----------------------------------------
---------------------------------- ----------------------------------
----------
    Validate OMF parameter values            September 21, 2022 5:37:18 PM
UTC   September 21, 2022 5:37:19 PM UTC   Success
    update db with backupconfig attributes   September 21, 2022 5:37:34 PM
UTC   September 21, 2022 5:37:39 PM UTC   Success
    Enable Database Autobackup               September 21, 2022 5:37:39 PM
UTC   September 21, 2022 5:37:39 PM UTC   Success
    Enable Archivelog Autobackup             September 21, 2022 5:37:39 PM
UTC   September 21, 2022 5:37:39 PM UTC   Success
    Configure Control file Auto Backup Format September 21, 2022 5:37:39
PM UTC   September 21, 2022 5:37:42 PM UTC   Success
    Backup Current Control file              September 21, 2022 5:37:42 PM
UTC   September 21, 2022 5:37:50 PM UTC   Success
    Update metadata for database:mydb        September 21, 2022 5:37:50 PM
UTC   September 21, 2022 5:37:50 PM UTC   Success
```

3. Back up the database using the `odacli create-backup` command. For example, you can take a backup type called Regular-L0 with archivelog using the `-ka` option.

   Note that you must take the backup with the `-ka` option, if you want to restore the backup as a standby database only for Oracle Database Appliance releases earlier than 19.12. From Oracle Database Appliance release 19.12 and later, the `-ka` option is not mandatory for restoring the database as standby.

```
 # odacli create-backup -n mydb -bt regular-l0 -ka
    {
      "jobId" : "6dc7ae65-93e3-4dc2-b4a7-563c8f98b01a",
      "status" : "Created",
      "message" : null,
      "reports" : [ ],
      "createTimestamp" : "September 21, 2022 17:39:03 PM UTC",
      "resourceList" : [ ],
      "description" : "Create regular-l0 Backup[TAG:auto][Db:mydb][NFS:/
nfs_backup_client/orabackups/scaoda823-c/database/3031000803/mydb]",
      "updatedTime" : "September 21, 2022 17:39:03 PM UTC"
    }

 #  odacli describe-job -i 6dc7ae65-93e3-4dc2-b4a7-563c8f98b01a

    Job details
    ----------------------------------------------------------------
                         ID:  6dc7ae65-93e3-4dc2-b4a7-563c8f98b01a
                Description:  Create regular-l0 Backup[TAG:auto][Db:mydb]
[NFS:/nfs_backup_client/orabackups/scaoda823-c/database/3031000803/mydb]
                     Status:  Success
                    Created:  September 21, 2022 5:39:03 PM UTC
                    Message:
```

```
    Task Name                                Start
Time                          End Time                          Status
    ----------------------------------------
---------------------------------- ----------------------------------
----------
    Validate backup config                   September 21, 2022 5:39:05 PM
UTC   September 21, 2022 5:39:05 PM UTC   Success
    NFS location existence validation        September 21, 2022 5:39:05 PM
UTC   September 21, 2022 5:39:05 PM UTC   Success
    Backup Validations                       September 21, 2022 5:39:05 PM
UTC   September 21, 2022 5:39:11 PM UTC   Success
    Recovery Window validation               September 21, 2022 5:39:11 PM
UTC   September 21, 2022 5:39:13 PM UTC   Success
    Archivelog deletion policy configuration September 21, 2022 5:39:13 PM
UTC   September 21, 2022 5:39:16 PM UTC   Success
    Database backup                          September 21, 2022 5:39:16 PM
UTC   September 21, 2022 5:40:27 PM UTC   Success
```

4. Oracle Database Appliance creates a backup report corresponding to each backup. Identify the backup report of the above database backup job using the `odacli list-backupreports` command. Describe the backup using the `odacli describe-backupreport -` *`backupreport_id`* command and save its output to a JSON file, as shown in the following example. After the JSON file for the backup report is ready, copy the file to the target system where you want to restore the database backup.

```
# odacli list-backupreports | grep -i mydb
  496d6e76-d061-4bc2-ae0a-b3325eaa15fa     09b476c2-8f09-4636-9233-
d8789c320b25     3031000803    mydb       mydb           regular-l0
auto         September 21, 2022 5:39:16 PM UTC   September 21, 2022
5:39:16 PM UTC   Configured
# odacli describe-backupreport -i 496d6e76-d061-4bc2-ae0a-b3325eaa15fa >
mydb.json
```

Restoring the Database from NFS Backup

Understand the steps to restore a database from NFS location.

Follow these steps to restore a database from an NFS location:

1. The NFS client location must be created on both the nodes of the Oracle Database Appliance system as the `root` user.

```
[root@***** ~]# mkdir /nfs_backup_client
```

2. At the server system, export the NFS server location to the Oracle Database Appliance system using the IP address of the Oracle Database Appliance system.

```
[root@***** ~]#  cat /etc/exports
    /nfs_backup_server <Node0_target_machine_IP>(rw,sync,no_root_squash)
    /nfs_backup_server <Node1_target_machine_IP>(rw,sync,no_root_squash)
```

For Oracle Database Appliance release 19.16 and earlier, you must specify the `no_root_squash` parameter. Starting with Oracle Database Appliance release 19.17, you need not specify the `no_root_squash` parameter.

3. Restart the NFS server on the system with the NFS server location.

```
Command on Oracle Linux 7 systems:
     [root@***** ~]# /bin/systemctl restart nfs.service

Command on Oracle Linux 8 systems:
      [root@***** ~]# /bin/systemctl restart nfs-server
```

4. Restore the database.

```
# odacli irestore-database -r mydb.json
    Enter SYS user password:
    Retype SYS user password:
    {
      "jobId" : "85a8792a-ca9b-4a50-8202-11d63949d3f2",
      "status" : "Created",
      "message" : null,
      "reports" : [ ],
      "createTimestamp" : "September 21, 2022 17:43:06",
      "resourceList" : [ ],
      "description" : "Database service recovery with db name: mydb",
      "updatedTime" : "September 21, 2022 17:43:06"
    }

 # odacli describe-job -i 85a8792a-ca9b-4a50-8202-11d63949d3f2

    Job details
    ----------------------------------------------------------------
                         ID:  85a8792a-ca9b-4a50-8202-11d63949d3f2
                Description:  Database service recovery with db name: mydb
                     Status:  Success
                    Created:  September 21, 2022 5:43:06 PM UTC
                    Message:

    Task Name                                Start
Time                          End Time                          Status
    ----------------------------------------
---------------------------------- ----------------------------------
----------
    Check if cluster ware is running         September 21, 2022 5:43:30 PM
UTC   September 21, 2022 5:43:30 PM UTC   Success
    Check if cluster ware is running         September 21, 2022 5:43:30 PM
UTC   September 21, 2022 5:43:30 PM UTC   Success
    Creating DbStorage for DbRestore         September 21, 2022 5:43:30 PM
UTC   September 21, 2022 5:48:36 PM UTC   Success
    Validating dbHome available space        September 21, 2022 5:43:30 PM
UTC   September 21, 2022 5:43:30 PM UTC   Success
    Validating dbHome available space        September 21, 2022 5:43:30 PM
UTC   September 21, 2022 5:43:30 PM UTC   Success
    Validating DiskSpace for DATA            September 21, 2022 5:43:30 PM
UTC   September 21, 2022 5:43:31 PM UTC   Success
    Generating SSH key                       September 21, 2022 5:43:31 PM
UTC   September 21, 2022 5:43:31 PM UTC   Success
    .
    .
    .
```

```
    Reset Associated Networks               September 21, 2022 6:30:03 AM
UTC   September 21, 2022 6:30:04 AM UTC   Success
    Set log_archive_dest for Database       September 21, 2022 6:30:04 AM
UTC   September 21, 2022 6:30:11 AM UTC   Success
```

Configuring Agent Proxy Settings for Object Store Access

If the Object Store IP address is accessible only through proxy setup by the Oracle Database Appliance server, then define the proxy setting for the agent, so that the agent can access the Object Store.

To create a backup policy that uses Object Store location, the agent must be able to access the Object Store URL.

1. Define the `HttpProxyHost` and `HttpProxyPort` settings in the `modify-agentconfig-parameters` command.

```
 # odacli modify-agentconfig-parameters -n HttpProxyHost -v www-
proxy.test.com -n HttpProxyPort -v 80 -u

Job details
----------------------------------------------------------------
ID: 0b0cbf9b-b0ab-4523-a096-5da4e48fc825
Description: Update agent configuration parameter values [HttpProxyHost,
HttpProxyPort]
Status: Created
Created: October 23, 2023 4:56:53 PM IST
Message:

Task Name Start Time End Time Status
----------------------------------------
```

   For more information about the `modify-agentconfig-parameters` command usage, see the *Oracle Database Appliance Command-Line Interface*.

2. Verify that the update succeeded:

```
# odacli describe-job -i 0b0cbf9b-b0ab-4523-a096-5da4e48fc825

Job details
----------------------------------------------------------------
ID: 0b0cbf9b-b0ab-4523-a096-5da4e48fc825
Description: Update agent configuration parameter values [HttpProxyHost,
HttpProxyPort]
Status: Success
Created: October 23, 2023 4:56:53 PM IST
Message:

Task Name Start Time End Time Status
```

3. Run the `list-agentconfigParameters` command to view the changes in the proxy settings:

```
# odacli list-agentConfigParameters
```

```
Name Value Description Updated
---------------------------- ------------------------------------
HttpProxyHost www-proxy.test.com October 23, 2023 4:56:53 PM IST
HttpProxyPort 80 October 23, 2018 4:56:53 PM IST
HttpsProxyHost Https proxy server host October 23, 2023 12:23:21 AM IST
HttpsProxyPort Https proxy server port October 23, 2023 12:23:21 AM IST
OSPatchRepos Repo list for OS patching October 23, 2023 12:23:21 AM IST
```

You can now use the Browser User Interface or the command-line interface to create a backup policy to use the ObjectStore location for backup.

Related Topics

- odacli list-agentconfig-parameters
  Use the `odacli list-agentconfig-parameters` command to list configuration variables used by the appliance.

Creating a Database Backup Policy

Understand how to create a policy to backup a database to an internal or external Oracle Fast Recovery Area (FRA) location or to the Object Store using the Browser User Interface.

The backup policy defines the backup, including the location where you want to store the backups and the recovery window. Backup to disk requires a lot of disk space (up to two to three times the size of the database).
Follow these steps to create a backup policy from the Browser User Interface:

1. Click the **Database** tab in the Browser User Interface.
2. Click **Backup Policy** in the left navigation to display a list of backup policies.
3. Click **Create Backup Policy**.
4. Enter a name for the backup policy and select the number of days for the recovery window. Select **Enable Crosscheck** to determine if the files on the disk or in the media management catalog correspond to data in the RMAN repository. The **Enable Compression** checkbox is selected by default. Selecting this option ensures backups are compressed. Select one of the following as the backup destination:
   - To backup to disk, select **Internal FRA** as the backup destination.
   - To backup to the cloud, select **ObjectStore** as the backup destination. If you have more than one Object Store, then select the Object Store Credential Name from the list. Enter a name in the Container Name field.
   - To backup to an NFS location, select **External FRA** as the backup destination, and specify the NFS mount point location.
   - To not define a destination, select **None**.

   The Backup Policy name must begin with an alpha character and contain only alphanumeric characters.
5. You must specify the **TDE Wallet Backup Location** to associate the backup configuration to a TDE-enabled database.
6. Click **Create**. Click **Yes** to confirm that you want to create the backup policy.

   A link to the job appears. When the job completes successfully, the backup configuration is ready.

Update an existing database to attach the backup configuration to the database using the ID or Name. When you attach the backup configuration to a database, the DCS agent schedules

daily automatic backups for the database and automatic backups for archivelog of the database.

Updating a Database with a Backup Policy

Attach a backup policy to a database to define the database backup attributes and destination.

1. In the Browser User Interface, click the **Database** tab.
2. For the database that you want to modify, select **Modify** in the Actions drop down list.
3. In the Modify Database page, select a backup policy from the **Select Backup Policy** drop down list.
4. Specify and confirm the **Backup Encryption Password**.
5. Click **Modify** to submit the job.
6. Click the **Activity** tab to check the job status.

   When the job completes successfully, the backup policy is associated with the database.

Updating a Backup Policy

Use the Browser User Interface to update the recovery window, change the ObjectStore Credential, or change the crosscheck option for a backup policy.

Crosscheck determines if the files on the disk on in the media management catalog correspond to data in the RMAN repository.

1. Click the **Database** tab in the Browser User Interface.
2. Click **Backup Policy** in the left navigation to display a list of backup policies.
3. Expand the **Actions** menu for a backup policy, then click **Update**.
4. To change the recovery window, use the up or down arrow to change the number of days.
5. To enable or disable the crosscheck, select or deselect **Enable Crosscheck**.
6. Click **Update**. Click **Yes** to confirm that you want to update the backup policy.

   A link to the job appears. When the job completes successfully, the backup configuration is ready.

Backing Up a Database Using the Browser User Interface

Apply a backup policy to a database, create a manual backup, update a database backup schedule, or update an archive log backup scheduler.

You can use the Browser User Interface or the command-line interface to create a database backup. In the command-line interface, use the command `odacli create-backup` for a specified Database.

Before creating a database backup, you must have a backup policy. The backup policy defines the recovery window and the backup destination. You must associate a backup policy with the database, otherwise you cannot create backups.
Follow these steps to backup a database from the Browser User Interface:

1. In the Browser User Interface, click the **Database** tab.
2. Click a database name to select a database from the list.

3. Review the database information, including the backup policy name and destination details. To select a backup policy for the database, click **Apply Backup Policy**. Select a policy and specify the **Backup Encryption Password**. For TDE-enabled databases, backups are encrypted by default and do not require the RMAN backup encryption password to be specified separately. For databases that do not have TDE enabled, the **Backup Encryption Password** is mandatory for Objectstore backup destination, optional for NFS backup destination and not required for Disk backup destination.
4. Click **Manual Backup**, **Update Database Backup Schedule** or **Update Archive Log Backup Schedule**.

   If the action is disabled, then apply a backup policy.
5. Click **Yes** to confirm the job.

   A link to the job appears. When the job completes successfully, the backup is ready. A list of backups appears at the bottom of the page.

Viewing Backup Reports

Understand how you can use the Browser User Interface to view backup reports.

A report is generated when a database backup is performed. The backup report lists the type, location, and date and timestamp of the backup.

1. In the Browser User Interface, click the **Database** tab, then click the database name in the list of databases.

   Information about the database appears on the Database Information page.
2. Scroll to the bottom of the page to view a list of Backups.
3. Click the Job ID for a backup to view the Backup Report and more details about the backup.

Recovering a Database Using the Browser User Interface

Understand the recovery options available for recovering a database.

1. In the Browser User Interface, click the **Database** tab, then select a database from the list.
2. On the Database Information page, click **Recover**.
3. On the Recover Database page, select any of the following recovery options:

   Recover Full Database to the specified Backup: Select the existing backup from which you want to recover the database

   Recover Full Database to the Latest: Select this option to recover the database from the last known good state, with the least possible data loss.

   Recover Full Database to the specified Timestamp: Specify the timestamp to recover the database.

   Recover Full Database to the System Change Number (SCN): Specify the SCN of the backup from which you want to recover the database.
4. Specify and confirm the **Backup Encryption Password** for databases without Transparent Database Encryption (TDE) enabled.
5. If the database has TDE enabled, then the **TDE Wallet Backup Location** and **TDE Password** fields are displayed.
6. Click the **Activity** tab to check the job status.

When the job completes successfully, the database is recovered as per the specified recovery options.

Deleting a Backup Using the Browser User Interface

Use the Browser User Interface to delete database backups.

You can delete Level-0, Level-1, and Archive Log backups anytime. To delete long term backups, you must check the Backup Report for the database to determine when the backup is obsolete.

1. Click the **Database** tab in the Browser User Interface.
2. Click the database name, in blue font, to display more details about the database for which you want to delete the backup.
3. Click **Delete Backups** .
4. Select the Backup type you want to delete, whether Level-0, Level-1, and Archive Log backup or long Term backup, then click **Start**. Click **Yes** to confirm that you want to delete the backup.

   A link to the job appears. When the job completes successfully, the backup is deleted.

Deleting a Backup Policy

Use the Browser User Interface to delete a database backup policy.

1. Click the **Database** tab in the Browser User Interface.
2. Click **Backup Policy** in the left navigation to display a list of backup policies.
3. Expand the **Actions** menu for a backup policy, then click **Delete**. Click **Yes** to confirm that you want to delete the backup policy.

   A link to the job appears. When the job completes successfully, the backup policy is deleted.

Using the CLI to Backup, Restore, and Recover

Use the command-line interface to backup, restore, and recover databases.

- Preparing for Database Backup to Disk
  Use the command-line interface to create a policy to backup a database to the Oracle Fast Recovery Area (FRA) of a disk.
- Preparing for Database Backup to NFS Location
  Use the command-line interface to create a policy to backup a database to an External FRA (NFS Location).
- Preparing for Database Backup to Oracle Cloud
  Create a backup configuration to backup a database to Oracle Cloud Infrastructure Object Storage.
- Backing Up a Database with the CLI
  Create a level 0, level 1, LongTerm database backup, or archivelog backup with the command-line interface.

- Recovering a Database with the CLI
  Recover the database to the system using a Backup Report, point-in-time recovery (PITR), a system change number (SCN), or from the latest backup time.
- Restoring a Database with the CLI
  You can restore a database to a system from Oracle Object Storage or NFS backup location using a backup report and the command-line interface.
- Restoring Archivelog with the CLI
  You can restore archivelog from one SCN to another SCN, or from one timestamp to another timestamp using the command-line interface.
- Deleting Backups with the CLI
  Delete older, obsolete level 0, level 1, archive log, and long term backups along with obsolete backup reports.

Preparing for Database Backup to Disk

Use the command-line interface to create a policy to backup a database to the Oracle Fast Recovery Area (FRA) of a disk.

You can create a backup policy using the Browser User Interface or the command-line interface. Most of the backup and recovery commands enable you to provide either a resource name or resource ID to identify the database.

Backup to disk or the **Backup to Internal FRA** option in the Browser User Interface, requires a lot of disk space (up to two to three times the size of the database).

Follow these steps to create a backup policy through the command-line interface:

1. Create a BackupConfig object (backup configuration) using the command `odacli create-backupconfig`.

   The backup configuration defines the backup, including the location where you want to store the backups and the recovery window.

   The following example creates a backup configuration named `mkg1dailydisk` with a backup to `disk` and a 1 day recovery window.

   ```
   # odacli create-backupconfig -d Disk -n mkg1dailydisk -w 1
   ```

2. Check the job status.

   ```
   # odacli describe-job -i job_id_from_step1
   ```

   When the job completes successfully, the backup configuration is ready.

3. Update an existing database to attach the backup configuration to the database using the ID or Name.

   When you attach the backup configuration to a database, the DCS agent schedules daily automatic backups for the database and automatic backups for the archivelog.

   The following example uses the Database Resource ID and Backup Configuration ID:

   ```
   # odacli modify-database -i d3c4d8f6-5eb7-4f9e-ab27-7bdd5013ac90 -bi
   9d942e0a-ba00-4cbc-9bfb-0de83ed279e5 -bp
   ```

The following example uses the Database Resource Name, `mydb` and the Backup Configuration Name, `mydb`:

```
# odacli modify-database -n mydb -bin mybcfg
```

Preparing for Database Backup to NFS Location

Use the command-line interface to create a policy to backup a database to an External FRA (NFS Location).

You can create a backup policy using the Browser User Interface or the command-line interface. Most of the backup and recovery commands enable you to provide either a resource name or resource ID to identify the database.

Backup to NFS location or the **Backup to External FRA** option in the Browser User Interface, requires creation of an NFS mount point.

Follow these steps to create a backup policy through the command-line interface:

1. Create a BackupConfig object (backup configuration) using the command `odacli create-backupconfig`.

   The backup configuration defines the backup, including the location where you want to store the backups and the recovery window.

   The following example creates a backup configuration named `mkg1dailynfs` with a backup to `disk` and a 1 day recovery window.

   ```
   # odacli create-backupconfig -d NFS -n mkg1dailynfs -c absolute-path-to-parent-directory-for-NFS-destination -w 1
   ```

2. Check the job status.

   ```
   # odacli describe-job -i job_id_from_step1
   ```

   When the job completes successfully, the backup configuration is ready.

3. (Optional) Update an existing database to attach the backup configuration to the database using the ID or Name.

   When you attach the backup configuration to a database, the DCS agent schedules daily automatic backups for the database and automatic backups for the archivelog.

   The following example uses the Database Resource ID and Backup Configuration ID:

   ```
   #  odacli modify-database -i d3c4d8f6-5eb7-4f9e-ab27-7bdd5013ac90 -bi 9d942e0a-ba00-4cbc-9bfb-0de83ed279e5 -bp
   ```

   The following example uses the Database Resource Name, `mydb` and the Backup Configuration Name, `mydb`:

   ```
   # odacli modify-database -in mydb -bin mybcfg -bp
   ```

   The `-bp` option to specify the backup password is optional and is not applicable for TDE-enabled databases.

Related Topics

- Configuring an NFS Backup Location
  Configure Network File System (NFS) backup location, to set up the NFS backup and mount the NFS client location on Oracle Database Appliance.

Preparing for Database Backup to Oracle Cloud

Create a backup configuration to backup a database to Oracle Cloud Infrastructure Object Storage.

Before you can backup to Oracle Object Storage, you must sign up for an Oracle Cloud Infrastructure Object Storage account and know your Oracle Object Storage credentials. When you create an Oracle ObjectStoreSwift object, the command creates and stores the credential details in the system and stores the password in an encrypted Oracle wallet. The Oracle wallet contains the Oracle Object Storage password.

Go to the Oracle Store to sign up for Oracle Cloud Infrastructure Object Storage.

1. Create an Oracle ObjectStoreSwift object using the command `create-objectstoreswift`.

   ```
   # odacli  create-objectstoreswift -e swift end point URL -n Object Store Swift name -t Object Store tenant name -u Object Store user name
   ```

   For example:

   ```
   # odacli create-objectstoreswift -e https://
   swiftobjectstorage.r1.oracleiaas.com/v1 -n ossn1 -t mytenant -u
   user.name@example.com
   ```

   The output includes an objectstoreswift Resource ID to identify an ObjectStoreSwift credential.

2. Create a BackupConfig object (backup configuration) using the command `odacli create-backupconfig` .

   The backup configuration defines the backup, including the location (container) where you want to store the backups, the recovery window, and the credentials that are defined in your encrypted Oracle wallet.

   The following example creates a backup configuration named `hr_7d_cloud` with a backup to the `ObjectStore`. The following backup parameters are defined: the container is `hr_bucket`, crosscheck is not enabled, the recovery window is 7 days, and the ObjectStoreSwift Resource ID is provided, which attaches the ObjectStoreSwiftObject to the backup configuration.

   ```
    # odacli  create-backupconfig  -n hr_7d_cloud -d ObjectStore -w 7 -o
   b974f006-5503-4949-ab6c-6f1a56e4ac37 -c hr_bucket -no-cr
   ```

   > **Note:**
   >
   > If the ObjectstoreSwift Object was not created with valid credentials, then creating the backup configuration will fail.

3. Check the job status.

   ```
   odacli describe-job -i job_id_from_step2
   ```

   When the job completes successfully, the backup configuration is ready.

4. Update an existing database to attach the backup configuration to the database.

   When you attach the backup configuration to a database, the DCS agent schedules daily automatic backups for the database, automatic backups for the archivelog, and weekly deletion of obsolete backups and backup reports for the database.

   ```
   # odacli modify-database -i e3c4d8f6-5eb7-4f9e-ab27-7bdd5013ac10 -bi
   7d942e0a-ba00-4cbc-9bfb-0de83ed279e5 -bp
   ```

> **Note:**
>
> When you create a new database, you have the option to associate the database with any type of backup configuration.

Backing Up a Database with the CLI

Create a level 0, level 1, LongTerm database backup, or archivelog backup with the command-line interface.

You can create a database backup in the Browser User Interface or with the command-line interface. When you attach a backup configuration object to a database, the DCS agent automatically schedules daily backups for the database. The day and time are based on the Oracle Database Appliance time zone. The AutoSchedule triggers a level 0 Backup every Sunday, a level 1 the other 6 days of the week, and an archivelog backup every 30 minutes. Use the command `odacli modify-schedule` to disable or change the scheduled backup time.

To create a Long Term database backup, or a level 0 or level 1 backup outside of the automatic backups, use the command `odacli create-backup`.

> **Note:**
>
> To recover a database, there must be a level 0 backup taken, so that the backup report contains information about the base backup. Restoring a database to the same system with a different DB Name and DB Unique Name or to another system does not always require a long term database backup report. You can use archivelog, or level 0, or level 1 backup reports as well. If you use the archivelog backup report , a prior database backup must exist.To restore a database to another system, you must have a long term database backup in Oracle Object Storage.

1. Verify that a backup configuration object is associated with the database.

2. Create a single backup, outside of the automatic backups, using the command `odacli create-backup`.

Use the Database Resource ID or Database Name to identify the database to backup, specify the type of backup, and use a tag to create a name for the backup. You can use up to 30 alphanumeric characters for the backup name tag. For example, create a level 0 backup named `2018Jan02_HRLevel0` for resource ID `20576eb1-bc32-4e34-bf97-fda0b60ca15b`

```
# odacli create-backup -i 20576eb1-bc32-4e34-bf97-fda0b60ca15b -bt Regular-L0 -t 2018Jan02_HRLevel0
```

When the job finishes, a backup report is generated for the backup. The report contains all of the metadata required to recover a database.

3. (Optional) Display a list of backup reports.

```
# odacli list-backupreports
```

Recovering a Database with the CLI

Recover the database to the system using a Backup Report, point-in-time recovery (PITR), a system change number (SCN), or from the latest backup time.

Recovering a database in Oracle Database Appliance is a full RMAN database recovery.

Recovering from a backup report requires JSON input for the backup report.

When recovering a TDE-enabled database, you need not specify the backup password.

1. Display a list of backup reports and locate the latest Backup Report for the database and save the Backup Report ID.

```
# odacli list-backupreports
```

2. Use the Backup Report ID to display the details of the Backup Report. Determine how you want to recover the database: PITR, SCN, or Latest and locate the needed information in the Backup Report.

```
# odacli describe-backupreport -i 7fbb06bf-e3a3-4aed-b617-4d1cd3d33387
{
  "id" : "7fbb06bf-e3a3-4aed-b617-4d1cd3d33387",
  "dbResId" : "4903d18a-6c20-4860-9ed0-d2ea7bd3d7b9",
  "tag" : "auto",
  "dbId" : "3022438697",
  "dbName" : "tdbasm1",
  "dbUniqueName" : "tdbasm1",
  "backupType" : "regular-l0",
  "keepDays" : null,
  "backupLocation" : "https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups",
  "cfBackupHandle" : "c-3022438697-20201022-03",
  "spfBackupHandle" : "c-3022438697-20201022-03",
  "pitrTimeStamp" : "October 22, 2020 07:51:59 AM UTC",
  "pitrSCN" : "2549429",
  "resetLogsTimeStamp" : "October 22, 2020 05:23:56 AM UTC",
  "resetLogsSCN" : "2510142",
  "oraHomeVersion" : "19.8.0.0.200714",
  "sqlPatches" : "31281355",
```

```
  "backupLogLoc" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
database/3022438697/tdbasm1/rmanlog/2020-10-22/
rman_backup_auto_2020-10-22_07-51-06.0546.log",
  "tdeWalletLoc" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/TestLtBackupFailur/scaoda8s002-
c/database/3022438697/tdbasm1/tdewallet/
ewallet_2020102207525186_tdbasm1.p12",
  "dbConfigLoc" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
database/3022438697/tdbasm1/dbconfig/2020-10-22/
DBCONFIG_auto_2020-10-22_07-52-43.0230.tar.gz",
  "name" : "Backup_Report_tdbasm1",
  "createTime" : "October 22, 2020 07:51:03 AM UTC",
  "state" : {
    "status" : "CONFIGURED"
  },
  "updatedTime" : "October 22, 2020 07:52:55 AM UTC",
  "backupReportLogDetail" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
database/3022438697/tdbasm1/rmandetaillogreport/2020-10-22/
rman_list_backup_detail_auto_2020-10-22_07-52-33.0919.log",
  "dbInfo" : {
    "dbClass" : "OLTP",
    "dbType" : "SI",
    "dbShape" : "odb1",
    "dbEdition" : "EE",
    "dbStorage" : "ASM",
    "dbRedundancy" : null,
    "pdbName" : null,
    "isCdb" : false
  },
  "dbDataSize" : "1662M",
  "dbRedoSize" : "12298M",
  "rmanBackupPieces" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
database/3022438697/tdbasm1/backuppieces/2020-10-22/
backupPieces_auto_4903d18a-6c20-4860-9ed0-
d2ea7bd3d7b9_20201022075238.json",
  "compressionAlgo" : "MEDIUM",
  "cpuPool" : null,
  "numberOfCores" : null
}
```


3. To recover from a backup report, based on SCN, identify the database ID and the backup report for recovery.

   In the following example, the JSON input file for the backup report is `backupreport.json`.

   ```
   # odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -br
   backupreport.json
   ```

4. To recover based on PITR, identify the database ID and the date and time stamp for recovery.

   ```
   # odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t PITR -r
   01/08/2018 12:57:33
   ```

5. To recover based on SCN, identify the database ID and the SCN number.

   ```
   # odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t SCN -s
   392375947
   ```

6. To recover based on the latest backup, identify the database ID and use the Latest option.

   ```
   # odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t Latest
   ```

7. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

   ```
   # odacli describe-job -i jobId
   ```

   When the job completes successfully, the recovery is complete.

> **Note:**
>
> Use the `-p` option only if the database backup is encrypted using the RMAN backup password. The `-p` option is not applicable for TDE-enabled databases.

Restoring a Database with the CLI

You can restore a database to a system from Oracle Object Storage or NFS backup location using a backup report and the command-line interface.

When you restore a database, environment and validation checks and setup tasks are performed. Restoring a database requires a number of tasks and configuration details, such as creating database storage, Oracle Home, recreating the control file, registering the database, and establishing RMAN restore and recovery. An Oracle Database Appliance backup report is generated for each backup and contains the metadata required to restore a database. The report is designed to help you to quickly and efficiently restore a database.

You can restore a database from one appliance to another appliance, or to the same appliance when the source database is deleted:

- To restore to a different system, copy the backup report generated by the command `odacli create-backup` to any folder on the other machine and use the command `odacli irestore-database` with the backup report. You must provide the complete location of the backup report when you run the command `odacli irestore-database`.
- To restore to the same system, delete the source database and use the backup report generated by the command `odacli create-backup`.

To restore a database from Object Store backup, the backup report corresponding to that backup must be present in the target system. Also an ObjectStoreSwift object with the credential that can access the Object Store backup must be created and specified while restoring the database.

> **Note:**
>
> For Standard Edition Oracle Database 19c or later, you cannot restore Oracle RAC or Oracle RAC One Node Database.

Follow these steps to restore a database using the command-line (CLI):

1. Display a list of backup reports and locate the latest Backup Report in Oracle Object Storage or NFS for the database and save the Backup Report ID.

```
# odacli list-backupreports
```

2. Use the Backup Report ID to display the details of the latest Backup Report.

```
#  odacli  describe-backupreport -i 7fbb06bf-e3a3-4aed-b617-4d1cd3d33387
{
  "id" : "7fbb06bf-e3a3-4aed-b617-4d1cd3d33387",
  "dbResId" : "4903d18a-6c20-4860-9ed0-d2ea7bd3d7b9",
  "tag" : "auto",
  "dbId" : "3022438697",
  "dbName" : "tdbasm1",
  "dbUniqueName" : "tdbasm1",
  "backupType" : "regular-l0",
  "keepDays" : null,
  "backupLocation" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups",
  "cfBackupHandle" : "c-3022438697-20201022-03",
  "spfBackupHandle" : "c-3022438697-20201022-03",
  "pitrTimeStamp" : "October 22, 2020 07:51:59 AM UTC",
  "pitrSCN" : "2549429",
  "resetLogsTimeStamp" : "October 22, 2020 05:23:56 AM UTC",
  "resetLogsSCN" : "2510142",
  "oraHomeVersion" : "19.8.0.0.200714",
  "sqlPatches" : "31281355",
  "backupLogLoc" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
database/3022438697/tdbasm1/rmanlog/2020-10-22/
rman_backup_auto_2020-10-22_07-51-06.0546.log",
  "tdeWalletLoc" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/TestLtBackupFailur/scaoda8s002-
c/database/3022438697/tdbasm1/tdewallet/
ewallet_2020102207525186_tdbasm1.p12",
  "dbConfigLoc" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
database/3022438697/tdbasm1/dbconfig/2020-10-22/
DBCONFIG_auto_2020-10-22_07-52-43.0230.tar.gz",
  "name" : "Backup_Report_tdbasm1",
  "createTime" : "October 22, 2020 07:51:03 AM UTC",
  "state" : {
    "status" : "CONFIGURED"
  },
  "updatedTime" : "October 22, 2020 07:52:55 AM UTC",
  "backupReportLogDetail" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
```

```
database/3022438697/tdbasm1/rmandetaillogreport/2020-10-22/
rman_list_backup_detail_auto_2020-10-22_07-52-33.0919.log",
  "dbInfo" : {
    "dbClass" : "OLTP",
    "dbType" : "SI",
    "dbShape" : "odb1",
    "dbEdition" : "EE",
    "dbStorage" : "ASM",
    "dbRedundancy" : null,
    "pdbName" : null,
    "isCdb" : false
  },
  "dbDataSize" : "1662M",
  "dbRedoSize" : "12298M",
  "rmanBackupPieces" : "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/paasdevdbcssi/Bha_Backups/scaoda8s002-c/
database/3022438697/tdbasm1/backuppieces/2020-10-22/
backupPieces_auto_4903d18a-6c20-4860-9ed0-
d2ea7bd3d7b9_20201022075238.json",
  "compressionAlgo" : "MEDIUM",
  "cpuPool" : null,
  "numberOfCores" : null
}
```

3. (Optional) If you need to generate a new backup report, use the command `odacli create-backup`.

   ```
   # odacli create-backup
   ```

4. Locate the Swift Object Store credential ID for the database backup.

   ```
   # odacli list-objectstoreswifts
   ```

5. Save the backup report from Step 2, as a JSON file in any directory on the appliance. For example, `backupreport.json`.
6. Restore the database to the system using the JSON file in the command `odacli irestore-database`. You must specify the abolute path to the JSON file.

   Run the command, then enter the SYS user password when prompted.

   ```
   # odacli irestore-database -r backupreport.json -oid Object Store ID
   ```

   To restore a TDE-enabled database, the TDE wallet password must be provided. You must specify the `-t` option in the command `odacli irestore-database`. If source database is TDE enabled, then the target database which was restored will also have TDE enabled.
7. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

   ```
   # odacli describe-job -i jobId
   ```

Restoring Archivelog with the CLI

You can restore archivelog from one SCN to another SCN, or from one timestamp to another timestamp using the command-line interface.

Follow these steps to restore archivelog using the command-line (CLI):

- Restore archivelog from one SCN to another SCN, or from one timestamp to another timestamp:

```
# odacli restore-archivelog -fs from_scn -ts to_scn
# odacli restore-archivelog -ft from_timestamp -tt to_timestamp
```

Deleting Backups with the CLI

Delete older, obsolete level 0, level 1, archive log, and long term backups along with obsolete backup reports.

For level 0, level 1, and archive log backups, the recovery window defined in the Backup Configuration determines when the backup is obsolete. The ranges are as follows:

- Backup to disk: between 1 and 14 days
- Backup to the Oracle Object Storage: between 1 and 31 days

For Long Term backups to the Oracle Object Storage or NFS, view the `keepDays` attribute of the Backup Report for a database to determine when the backup is obsolete. To delete Long Term Backup, the Database Resource ID or Database name and the Backup Report are required.

Follow these steps to delete database backups using `odacli` commands:

1. Display a list of all backup reports.

```
# odacli  list-backupreports
```

2. Obtain the Database ID and type of backup.

```
# odacli  describe-backupreport -i Backup Report ID
```

3. To delete obsolete level 0, level 1, and archivelog backups along with obsolete backup reports, use the Database ID or Database Name to delete the backup.

```
# odacli delete-backup -i 20576eb1-bc32-4e34-bf97-fda0b60ca15b
```

4. To delete Long Term backups, use the Backup Report from the JSON file to delete the backup.

   In the following example, the JSON input file for the backup report is `backupreport.json`.

```
# odacli delete-backup -i 20576eb1-bc32-4e34-bf97-fda0b60ca15b -br
backupreport.json
```

13

Using Oracle Data Guard for Disaster Management and Recovery on Oracle Database Appliance

Understand how you can use Oracle Data Guard for disaster management and recovery on Oracle Database Appliance.

- About Oracle Data Guard on Oracle Database Appliance
  Oracle Database Appliance provides client interface through ODACLI commands for easy configuration and management of Oracle Data Guard for high availability, data protection, and disaster recovery.
- Configuring Oracle Data Guard on Oracle Database Appliance
  Oracle Database Appliance enables you to set up Oracle Data Guard using ODACLI commands.
- Viewing Oracle Data Guard Status on Oracle Database Appliance
  View Oracle Data Guard status using ODACLI commands.
- Performing Oracle Data Guard Operations
  Perform failover, switchover, and reinstate operations on Oracle Data Guard using ODACLI commands.
- Creating an Oracle Data Guard Network
  Create an Oracle Data Guard network and associate it with a database using ODACLI commands.
- Migrating an Oracle Database from One System to Another Using Integrated Oracle Data Guard
  Understand how you can use Integrated Oracle Data Guard to migrate an Oracle Database from one system to another.
- Patching Oracle Data Guard Databases on Oracle Database Appliance
  Understand how you can patch primary and standby databases on Oracle Data Guard using ODACLI commands.
- Upgrading Oracle Data Guard Databases on Oracle Database Appliance
  Understand how you can upgrade primary and standby databases on Oracle Data Guard using ODACLI commands.
- Enabling Oracle Active Data Guard on Existing Configurations
  Understand how you can enable Oracle Active Data Guard on existing configurations.
- Registering Oracle Data Guard on Oracle Database Appliance
  Oracle Database Appliance enables you to migrate Oracle Data Guard configuration that were manually configured, and register the configuration metadata with Oracle Database Appliance using ODACLI commands.
- Deconfiguring Oracle Data Guard on Oracle Database Appliance
  Deconfigure Oracle Data Guard status using ODACLI commands.

About Oracle Data Guard on Oracle Database Appliance

Oracle Database Appliance provides client interface through ODACLI commands for easy configuration and management of Oracle Data Guard for high availability, data protection, and disaster recovery.

Oracle Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions. Oracle Data Guard maintains these standby databases as copies of the production database. Then, if the production database becomes unavailable because of a planned or an unplanned outage, Oracle Data Guard can switch any standby database to the production role, minimizing the downtime associated with the outage. Oracle Data Guard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability. Oracle Data Guard transport services are also used by other Oracle features such as Oracle Streams and Oracle GoldenGate for efficient and reliable transmission of redo from a source database to one or more remote destinations.

With Oracle Data Guard, administrators can optionally improve production database performance by offloading resource-intensive backup and reporting operations to standby systems.

The ability to create clone databases on a standby system enables you to set up test or development environments based on your standby databases. Since standby systems are not production systems, they are a better choice for seeding test or development environments.

> **Note:**
>
> Oracle Database Enterprise Edition includes Oracle Data Guard.

Configuring Oracle Data Guard on Oracle Database Appliance

Oracle Database Appliance enables you to set up Oracle Data Guard using ODACLI commands.

Prerequisites for Oracle Database Guard Configuration

Before you begin:

- Oracle recommends running the primary and the standby databases on separate Oracle Database Appliance hardware, so ensure that you have at least two separate Oracle Database Appliance machines.
- Oracle recommends that the primary and standby systems have the same Oracle Database Appliance configuration. The database must have identical configuration for database version and storage (both must have either Oracle ASM or Oracle ACFS storage) to avoid any unpredictability with the database switch roles.
- The primary and standby systems must be the same Oracle Database Appliance release, and must be on Oracle Database Appliance release 19.8 or later.
- If you have customized the operating system, then ensure that environments on both machines are identical.

- Ensure that your deployment follows Oracle Maximum Availability Architecture (MAA) best practices. See the Oracle Maximum Availability Architecture (MAA) page on Oracle Technology Network.
- If you decide to use Oracle ObjectStore for backup and recovery, then you must configure access for both the primary and standby systems.

Setting up Backup and Recovery for Oracle Database Guard Configuration

You can use either Object Store or External FRA (NFS) to set up the backup for primary and standby databases when configuring Oracle Data Guard on Oracle Database Appliance. A backup is necessary to instantiate a standby database.

To set up Object Store backup (recommended):

1. Configure the agent proxy settings:

   ```
   # odacli modify-agentconfig-parameters -n HttpProxyHost -v www-
   proxy.test.com -n HttpProxyPort -v 80 -u
   ```

   For more information, see the topic *Configuring Agent Proxy Settings for Object Store Access* in this guide.

2. Create the Object Store credentials on both primary and standby systems:

   ```
   # odacli create-objectstoreswift -e swift_end_point_URL -n Object
   Store_Swift_name -t Object_Store_tenant_name -u Object_Store_user_name
   ```

   For example:

   ```
   # odacli create-objectstoreswift -e https://swiftobjectstorage.us-
   ashburn-1.oraclecloud.com/v1  -n aaa -t test_user -u testuser
   ```

3. View the Object Store Swift name:

   ```
   # odacli list-objectstoreswifts
   ```

4. Create the backup configuration:

   ```
   # odacli create-backupconfig -d objectstore -c testuser -on aaa -w 7 -cr -
   n backupConfig7days
   ```

5. Associate the backup configuration with the database:

   ```
   # odacli modify-database -in test_db -bin backupConfig7days -bp
   ```

To set up NFS backup:

1. Create the backup configuration. For High-Availability deployments, run the command on one node only.

   ```
   # odacli create-backupconfig -n nfsbkup -w 10 -d NFS -c /u01/nfs_backup
   ```

2. Associate the backup configuration with the database:

   ```
   # odacli modify-database -in test_db -bin nfsbkup
   ```

Configuring Oracle Data Guard using ODACLI commands

Follow these steps:

1. Take a backup of your primary database to Object Store or External FRA (NFS).

   ```
   # odacli create-backup --backupType Regular-L0 -in primary_DB
   ```

   For example:

   ```
   # odacli create-backup --backupType Regular-L0 -in test_db
   ```

2. Save the backup report:

   ```
   # odacli describe-backupreport -i eba406c7-ae46-4d1d-a1e5-3e985397b171 >
   backup_report_test_db_0620.json
   ```

   The backup report ID is the ID of the backup created in step 1 and can be obtained by running the `odacli list-backupreports` command.

3. Copy the backup report to the standby system and restore the backup to a standby database. Note that to create a database with role as standby in an Oracle Data Guard configuration on Oracle Database Appliance, you do not need to copy the wallet of the primary database when using Oracle Key Vault keystore for TDE-enabled databases. The same wallet can be specified in the `odacli irestore-database` command.

   ```
   # odacli irestore-database -ro STANDBY
   ```

   If you want to use Object Store, then specify the Object Store location:

   ```
   # odacli irestore-database -r backup_report.json -u dbUniqueName -on
   objectStoreName -bp -ro STANDBY
   Rman Backup Encryption Password(s). Multiple passwords needs to be comma
   separated.:
   Password for SYS user:
   ```

   In the above command, specify the `dbUniqueName` of the standby system.

   If your backup option is NFS, then specify the NFS location:

   ```
   odacli irestore-database -r backup_report.json -u dbUniqueName -ro STANDBY
   Password for SYS user:
   ```

4. Configure Oracle Data Guard, by running the `odacli configure-dataguard` command on the primary system. Provide the values for your configuration through the interactive command line interface.
   In the interactive CLI configuration steps, the parameters are as follows:

   - Standby site address is IP address or host name of the standby host. Provide the fully qualified domain name and hostname if the primary and the standby systems are in the same domain and DNS is configured.
   - Select Oracle Data Guard protection modes to meet availability, performance and data protection requirements. Oracle Data Guard Protection Modes are Maximum

Availability, Maximum Performance, and Maximum Protection. The log transport modes are ASYNC, SYNC, and FASTSYNC.
You can select the following combinations of protection modes and transport types:

Table 13-1 Oracle Data Guard protection modes and transport modes

Protection Mode \ Transport Type	ASYNC	FASTSYNC**	SYNC
MAXPERFORMANCE	Y*	Y	Y
MAXAVAILABILITY	N	Y	Y*
MAXPROTECTION	N	N	Y*

* in the table indicates the default supported pair and ** FASTSYNC mode is available only in Oracle Database 12.1 or later.

```
# odacli configure-dataguard
Standby site address: test_domain
root@test_domain's password:
BUI username for Standby site. If Multi-user Access is disabled on Standby
site, enter 'oda-admin';
otherwise, enter the name of the user who has irestored the Standby
database (default: oda-admin):
BUI password for Standby site:
Database name for Data Guard configuration: test_db
Primary database SYS password:
********************************************************************************
***************
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port: 1521
Standby database listener port: 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: test_dg
Active Data Guard: disabled
Do you want to edit this Data Guard configuration? (Y/N, default:N):
As a requirement, Standby database's SYS password will be set to Primary
database's after Data Guard configuration. Are you sure you want to
proceed with Data Guard configuration? (Y/N): y
********************************************************************************
***************
Configure Data Guard test_dg started
********************************************************************************
***************
Step 1: Validate Data Guard configuration request (Primary site)
Description: Validate DG Config Creation for db test_db
Job ID: 523a38ac-e641-46e2-bf38-ecfb641e5dc9
Started December 08, 2020 21:08:59 PM UTC
Validate create Data Guard configuration request
Finished December 08, 2020 21:09:02 PM UTC
********************************************************************************
***************
Step 2: Validate Data Guard configuration request (Standby site)
Description: Validate DG Config Creation for db test_db
```

```
Job ID: 712843f9-65ca-4aea-95eb-4e5c461f3672
Started December 08, 2020 21:09:03 PM UTC
Validate create Data Guard configuration request
Finished December 08, 2020 21:09:05 PM UTC
********************************************************************************
****************
Step 3: Download password file from Primary database (Primary site)
Description: Download orapwd file from Primary database
Started December 08, 2020 21:09:05 PM UTC
Prepare orapwd file for Primary database test_db
Finished December 08, 2020 21:09:10 PM UTC
********************************************************************************
****************
Step 4: Upload password file to Standby database (Standby site)
Description: Upload orapwd file to Standby database
Started December 08, 2020 21:09:10 PM UTC
Write orapwd file to Standby database test_db
Finished December 08, 2020 21:09:19 PM UTC
********************************************************************************
****************
Step 5: Configure Primary database (Primary site)
Description: DG Config service for db test_db - ConfigurePrimary
Job ID: ad8486a2-bea6-47ff-a8ab-330be98987ab
Started December 08, 2020 21:09:19 PM UTC
Configure host DNS on primary env
Configure Data Guard Tns on primary env
Enable Data Guard related Db parameters for primary env
Enable force logging and archivelog mode in primary env
Enable FlashBack
Configure network parameters for local listener on primary env
Restart listener on primary env
Create services for primary db
Finished December 08, 2020 21:09:35 PM UTC
********************************************************************************
****************
Step 6: Configure Standby database (Standby site)
Description: DG Config service for db test_db - ConfigureStandby
Job ID: 47876bb8-8fef-41a6-a0b6-92b184c74900
Started December 08, 2020 21:09:37 PM UTC
Configure Data Guard Tns on standby env
Configure host DNS on standby env
Clear Data Guard related Db parameters for standby env
Enable Data Guard related Db parameters for standby env
Enable force logging and archivelog mode in standby env
Populate standby database metadata
Configure network parameters for local listener on standby env
Reset Db sizing and hidden parameters for ODA best practice
Restart Listener on standby env
Create services for standby db
Finished December 08, 2020 21:11:02 PM UTC
********************************************************************************
****************
Step 7: Configure and enable Data Guard (Primary site)
Description: DG Config service for db test_db - ConfigureDg
Job ID: 64674e50-3eea-48d0-a25e-bc61401a3d71
Started December 08, 2020 21:11:02 PM UTC
```

```
Config and enable Data Guard
Post check Data Guard configuration
Finished December 08, 2020 21:12:21 PM UTC
********************************************************************************
***************
Step 8: Enable Flashback (Standby site)
Description: DG Config service for db test_db - EnableFlashback
Job ID: 6f461b2f-88e1-41bf-9ca8-d91bc7d4bf97
Started December 08, 2020 21:12:22 PM UTC
Enable FlashBack
Finished December 08, 2020 21:12:33 PM UTC
********************************************************************************
***************
Step 9: Re-enable Data Guard (Primary site)
Description: DG Config service for db test_db - ReenableDg
Job ID: 3a4a404d-e1b5-4c4e-9db1-545676eebc77
Started December 08, 2020 21:12:34 PM UTC
Re-enable Data Guard if inconsistent properties found
Post check Data Guard configuration
Finished December 08, 2020 21:12:36 PM UTC
********************************************************************************
***************
Step 10: Create Data Guard status (Primary site)
Description: DG Status service for db test_db - NewDgconfig
Job ID: 572dea8d-5682-4193-a984-0aebb072be45
Started December 08, 2020 21:12:36 PM UTC
Create Data Guard status
Finished December 08, 2020 21:12:39 PM UTC
********************************************************************************
***************
Step 11: Create Data Guard status (Standby site)
Description: DG Status service for db test_db - NewDgconfig
Job ID: 75e55d7f-49c3-4e2d-8907-a7130cc05034
Started December 08, 2020 21:12:40 PM UTC
Create Data Guard status
Finished December 08, 2020 21:12:41 PM UTC
********************************************************************************
***************
Configure Data Guard test_dg completed
********************************************************************************
***************
```

Example with Oracle Data Guard custom configuration:

```
# odacli configure-dataguard
Standby site address: test_domain
root@test_db's password:
BUI username for Standby site. If Multi-user Access is disabled on Standby
site, enter 'oda-admin';
otherwise, enter the name of the user who has irestored the Standby
database (default: oda-admin):
BUI password for Standby site:
Database name for Data Guard configuration: test_db
Primary database SYS password:
********************************************************************************
```

```
****************
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port: 1521
Standby database listener port: 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: test_dg
Active Data Guard: disabled
Do you want to edit this Data Guard configuration? (Y/N, default:N): y
********************************************************************************
****************
Primary site network for Data Guard configuration [Public-network]
(default: Public-network):
Standby site network for Data Guard configuration [Public-network]
(default: Public-network):
Primary database listener port (default: 1521):
Standby database listener port (default: 1521):
Transport type [ASYNC, FASTSYNC, SYNC] (default: ASYNC): SYNC
Protection mode [MAX_PROTECTION, MAX_PERFORMANCE, MAX_AVAILABILITY]
(default: MAX_PERFORMANCE): MAX_PROTECTION
Data Guard configuration name (default: pcpub_scpub): custom_test_dg
Enable Active Data Guard? (Y/N, default:N): y
The TrustStore of the Standby site should be removed unless it is
necessary for some other purpose. Do you
As a requirement, Standby database's SYS password will be set to Primary
database's after Data Guard configuration. Are you sure you want to
proceed with Data Guard configuration? (Y/N): y
********************************************************************************
****************
Configure Data Guard custom_test_dg started
********************************************************************************
****************
Step 1: Validate Data Guard configuration request (Primary site)
Description: Validate DG Config Creation for db test_db
Job ID: 523a38ac-e641-46e2-bf38-ecfb641e5dc9
Started December 08, 2020 21:08:59 PM UTC
Validate create Data Guard configuration request
Finished December 08, 2020 21:09:02 PM UTC
********************************************************************************
****************
Step 2: Validate Data Guard configuration request (Standby site)
Description: Validate DG Config Creation for db test_db
Job ID: 712843f9-65ca-4aea-95eb-4e5c461f3672
Started December 08, 2020 21:09:03 PM UTC
Validate create Data Guard configuration request
Finished December 08, 2020 21:09:05 PM UTC
********************************************************************************
****************
Step 3: Download password file from Primary database (Primary site)
Description: Download orapwd file from Primary database
Started December 08, 2020 21:09:05 PM UTC
Prepare orapwd file for Primary database test_db
Finished December 08, 2020 21:09:10 PM UTC
********************************************************************************
```

```
***************
Step 4: Upload password file to Standby database (Standby site)
Description: Upload orapwd file to Standby database
Started December 08, 2020 21:09:10 PM UTC
Write orapwd file to Standby database test_db
Finished December 08, 2020 21:09:19 PM UTC
********************************************************************************
***************
Step 5: Configure Primary database (Primary site)
Description: DG Config service creation for db test_db - ConfigurePrimary
Job ID: ad8486a2-bea6-47ff-a8ab-330be98987ab
Started December 08, 2020 21:09:19 PM UTC
Configure host DNS on primary env
Configure Data Guard Tns on primary env
Enable Data Guard related Db parameters for primary env
Enable force logging and archivelog mode in primary env
Enable FlashBack
Configure network parameters for local listener on primary env
Restart listener on primary env
Create services for primary db
Finished December 08, 2020 21:09:35 PM UTC
********************************************************************************
***************
Step 6: Configure Standby database (Standby site)
Description: DG Config service creation for db test_db - ConfigureStandby
Job ID: 47876bb8-8fef-41a6-a0b6-92b184c74900
Started December 08, 2020 21:09:37 PM UTC
Configure Data Guard Tns on standby env
Configure host DNS on standby env
Clear Data Guard related Db parameters for standby env
Enable Data Guard related Db parameters for standby env
Enable force logging and archivelog mode in standby env
Populate standby database metadata
Configure network parameters for local listener on standby env
Reset Db sizing and hidden parameters for ODA best practice
Restart Listener on standby env
Create services for standby db
Finished December 08, 2020 21:11:02 PM UTC
********************************************************************************
***************
Step 7: Configure and enable Data Guard (Primary site)
Description: DG Config service creation for db test_db - ConfigureDg
Job ID: 64674e50-3eea-48d0-a25e-bc61401a3d71
Started December 08, 2020 21:11:02 PM UTC
Config and enable Data Guard
Post check Data Guard configuration
Finished December 08, 2020 21:12:21 PM UTC
********************************************************************************
***************
Step 8: Enable Flashback (Standby site)
Description: DG Config service creation for db test_db - EnableFlashback
Job ID: 6f461b2f-88e1-41bf-9ca8-d91bc7d4bf97
Started December 08, 2020 21:12:22 PM UTC
Enable FlashBack
Finished December 08, 2020 21:12:33 PM UTC
********************************************************************************
```

```
****************
Step 9: Re-enable Data Guard (Primary site)
Description: DG Config service creation for db test_db - ReenableDg
Job ID: 3a4a404d-e1b5-4c4e-9db1-545676eebc77
Started December 08, 2020 21:12:34 PM UTC
Re-enable Data Guard if inconsistent properties found
Post check Data Guard configuration
Finished December 08, 2020 21:12:36 PM UTC
********************************************************************************
****************
Step 10: Enable Active Data Guard (Standby site)
Description: DG Config service creation for db test_db - EnableActivedg
Job ID: 3117c046-2d71-453d-9950-c85bed19f5f2
Started December 08, 2020 21:12:38 PM UTC
Started
Open standby db
Restart standby db
Finished December 08, 2020 21:12:45 PM UTC
********************************************************************************
****************
Step 11: Create Data Guard status (Primary site)
Description: DG Status service creation for db test_db - NewDgconfig
Job ID: 572dea8d-5682-4193-a984-0aebb072be45
Started December 08, 2020 21:12:46 PM UTC
Create Data Guard status
Finished December 08, 2020 21:12:49 PM UTC
********************************************************************************
****************
Step 12: Create Data Guard status (Standby site)
Description: DG Status service creation for db test_db - NewDgconfig
Job ID: 75e55d7f-49c3-4e2d-8907-a7130cc05034
Started December 08, 2020 21:12:52 PM UTC
Create Data Guard status
Finished December 08, 2020 21:12:55 PM UTC
********************************************************************************
****************
Configure Data Guard custom_test_dg completed
********************************************************************************
****************
```

You can also specify a JSON file to configure Oracle Data Guard. An example JSON file is described in this topic.

```
# odacli configure-dataguard -r configdg_ODA_HA.json
Standby machine address: test_domain
BUI username for Standby site. If Multi-user Access is disabled on Standby
site, enter 'oda-admin';
otherwise, enter the name of the user who has irestored the Standby
database (default: oda-admin):
BUI password for Standby machine:
Primary database SYS password:
As a requirement, Standby database's SYS password will be set to Primary
database's after Data Guard configuration. Are you sure you want to
proceed? (Y/N): y
Data Guard configuration for database 'test' started
```

```
    ********************************************************************************
    ****************
    Step 1: Validate Data Guard configuration request (Primary machine)
    .....
```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Example JSON for configuring Oracle Data Guard on Oracle Database Appliance

```
{
  "name": "test1_test7",
  "protectionMode": "MAX_PERFORMANCE",
  "enableFlashback": true,
  "enableActiveDg": false,
  "replicationGroups": [
    {
      "sourceEndPoints": [
        {
          "endpointType": "PRIMARY",
          "hostName": test_domain1",
          "listenerPort": 1521,
          "databaseUniqueName": "test1",
          "ipAddress": "test_IPaddress"
        },
    {
      "endpointType": "PRIMARY",
      "hostName": "test_domain1",
      "listenerPort": 1521,
      "databaseUniqueName": "test1",
      "ipAddress": "test_IPaddress2"
    }
      ],
      "targetEndPoints": [
        {
          "endpointType": "STANDBY",
          "hostName": "test_domain2",
          "listenerPort": 1521,
          "databaseUniqueName": "test7",
          "ipAddress": "test_IPaddress3"
        },
        {
          "endpointType": "STANDBY",
          "hostName": "test_domain2",
          "listenerPort": 1521,
          "databaseUniqueName": "test7",
          "ipAddress": "test_IPaddress4"
        }
      ],
      "transportType": "ASYNC"
    }
  ]
}
```

Related Topics

- Configuring Agent Proxy Settings for Object Store Access
  If the Object Store IP address is accessible only through proxy setup by the Oracle Database Appliance server, then define the proxy setting for the agent, so that the agent can access the Object Store.
- https://www.oracle.com/in/database/technologies/high-availability/maa.html

Viewing Oracle Data Guard Status on Oracle Database Appliance

View Oracle Data Guard status using ODACLI commands.

Viewing Oracle Data Guard Status using ODACLI commands

Follow these steps:

- To view the status of primary and standby databases in your Oracle Data Guard configuration, use the following command:

```
# odacli list-dataguardstatus
```

- To view the status of your specific Oracle Data Guard configuration, use the following command:

```
# odacli describe-dataguardstatus -i 5f0ed48a-366d-454c-a389-fe367772dbb7
```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli Oracle Data Guard Commands
  Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.

Performing Oracle Data Guard Operations

Perform failover, switchover, and reinstate operations on Oracle Data Guard using ODACLI commands.

Performing Oracle Data Guard operations using ODACLI commands

Follow these steps:

- To switch roles between the primary database and standby database in your Oracle Data Guard configuration, run the following command on the **primary** system:

```
# odacli switchover-dataguard -i dg_ID -u
standby_db_unique_name_to_switchover
```

For example:

```
# odacli switchover-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u
test_db2
```

Confirm that the switchover of roles completed successfully:

```
# odacli describe-dataguardstatus -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19
```

- A failover provides for transitioning a standby database in your configuration to take over the primary database role, if all instances of an Oracle RAC primary database fail, or are unreachable.
  To failover a standby database to the primary database in your Oracle Data Guard configuration, run the following command on your current **standby** system:

  ```
  # odacli failover-dataguard -i dg_ID -u
  standby_db_unique_name_to_failover_to
  ```

  For example:

  ```
  # odacli failover-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u
  test_db2
  ```

  Confirm that the requested database failover to the new primary database has completed successfully:

  ```
  # odacli describe-dataguardstatus -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19
  ```

- To reinstate a failed primary database after a failover and convert it to a new standby database in your Oracle Data Guard configuration, run the following command on your new primary system:

  ```
  # odacli reinstate-dataguard -i dg_ID -u db_unique_name_to_reinstate
  ```

  For example:

  ```
  # odacli reinstate-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u
  test_db2
  ```

  Confirm that the requested database is reinstated as a new standby database:

  ```
  # odacli describe-dataguardstatus -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19
  ```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli Oracle Data Guard Commands
  Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.

Creating an Oracle Data Guard Network

Create an Oracle Data Guard network and associate it with a database using ODACLI commands.

Configuring an Oracle Data Guard Network using ODACLI commands

You can configure Oracle Data Guard using a network of type Public, Database, or Dataguard.

Follow these steps:

1. Create the network:

```
# odacli create-network -n btbond1 -t VLAN -p 192.209.10.227 -w Dataguard -
no-d -s 255.255.255.240 -v 369 -g 192.209.10.225
```

2. View the new network:

```
# odacli list-networks

ID                                       Name                 NIC
Interface Type  Subnet Mask        Gateway            VLAN ID  Node
Networks
---------------------------------------- -------------------- ------------
--------------- ------------------ ------------------ --------
------------------------
836cfa4b-5d2c-45e3-a62d-099fc7c57984   Private-network      priv0
INTERNAL        255.255.255.240    [ IP Address on node0:  192.168.16.24 ]
1a312239-761d-4321-84d2-8da406f21e5c   Public-network       btbond1.359
VLAN            255.255.255.240    10.209.10.65       359  [ IP Address on
node0:  192.209.10.69 ]
f279ac96-5a4a-4b73-86af-1afbdb6ae5d4   vlan369              btbond1.369
VLAN            255.255.255.240    10.209.10.225      369  [ IP Address on
node0:  192.209.10.227]
```

3. To attach a network of type Dataguard to a database:

```
# odacli modify-database -in test_db -an vlan369
```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli Oracle Data Guard Commands
  Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.

Migrating an Oracle Database from One System to Another Using Integrated Oracle Data Guard

Understand how you can use Integrated Oracle Data Guard to migrate an Oracle Database from one system to another.

Migrating Oracle Data Guard Database using ODACLI commands

If your database is on an old Oracle Database Appliance hardware model, and you want to move your deployment to a new Oracle Database Appliance hardware, then you can use your Oracle Data Guard configuration on Oracle Database Appliance to migrate from one system to another. The same steps can also be applied to migrate an Oracle Database from an Oracle Database Appliance bare Metal system to a DB system in a KVM deployment, or from one DB system in a KVM deployment to another DB system.

Before you begin:

- Oracle recommends running the primary and the standby databases on separate Oracle Database Appliance hardware, so ensure that you have at least two separate Oracle Database Appliance machines.
- The primary and standby systems must be the same Oracle Database Appliance release, and must be on Oracle Database Appliance release 19.8 or later. The instances on the primary and standby databases must have similar configuration for database parameter settings including database version, memory, CPU, networking, and storage (Oracle ASM or Oracle ACFS), thus avoiding any unpredictability when the databases switch roles.
- If you have customized the operating system, then ensure that environments on both machines are identical.
- Ensure that your deployment follows Oracle Maximum Availability Architecture (MAA) best practices.

Follow these steps:

1. Take a backup of your primary database to Object Store or External FRA (NFS).

   ```
   # odacli create-backup primary_DB
   ```

   For example:

   ```
   # odacli create-backup --backupType Regular-L0 -in test_db
   ```

2. Save the backup report:

   ```
   # odacli describe-backupreport -i eba406c7-ae46-4d1d-a1e5-3e985397b171 >
   backup_report_test_db_0620.json
   ```

   The backup report ID is the ID of the backup created in step 1 and can be obtained by running the `odacli list-backupreports` command.

3. Copy the backup report to the standby system and restore the backup to a standby database.

   ```
   # odacli irestore-database -ro STANDBY
   ```

If your backup option is Object Store, then specify the Object Store location:

```
# odacli irestore-database -r backup_report.json -u dbUniqueName -on
objectStoreName -bp -ro STANDBY -dh
Password for SYS user:
Rman Backup Encryption Password(s). Multiple passwords needs to be comma
separated.:
```

If your backup option is NFS, then specify the NFS location:

```
odacli irestore-database -r backup_report.json -u dbUniqueName -ro STANDBY
Password for SYS user:
```

4. Configure Oracle Data Guard, by running the following command on the primary system:

```
# odacli configure-dataguard
Standby site address: test_domain
root@test_domain's password:
BUI username for Standby site. If Multi-user Access is disabled on Standby
site, enter 'oda-admin';
otherwise, enter the name of the user who has irestored the Standby
database (default: oda-admin):
BUI password for Standby site:
Database name for Data Guard configuration: test_db
Primary database syspassword:
********************************************************************************
****************
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port: 1521
Standby database listener port: 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: test_dg
Active Data Guard: disabled
Do you want to edit this Data Guard configuration? (Y/N, default:N):
As a requirement, Standby database's SYS password will be set to Primary
database's after Data Guard configuration. Are you sure you want to
proceed with Data Guard configuration? (Y/N): y
********************************************************************************
****************
Configure Data Guard test_dg started
********************************************************************************
****************
Step 1: Validate Data Guard configuration request (Primary site)
Description: Validate DG Config Creation for db test_db
Job ID: 523a38ac-e641-46e2-bf38-ecfb641e5dc9
Started December 08, 2020 21:08:59 PM UTC
Validate create Data Guard configuration request
Finished December 08, 2020 21:09:02 PM UTC
********************************************************************************
****************
Step 2: Validate Data Guard configuration request (Standby site)
Description: Validate DG Config Creation for db test_db
```

```
Job ID: 712843f9-65ca-4aea-95eb-4e5c461f3672
Started December 08, 2020 21:09:03 PM UTC
Validate create Data Guard configuration request
Finished December 08, 2020 21:09:05 PM UTC
********************************************************************************
****************
Step 3: Download password file from Primary database (Primary site)
Description: Download orapwd file from Primary database
Started December 08, 2020 21:09:05 PM UTC
Prepare orapwd file for Primary database test_db
Finished December 08, 2020 21:09:10 PM UTC
********************************************************************************
****************
Step 4: Upload password file to Standby database (Standby site)
Description: Upload orapwd file to Standby database
Started December 08, 2020 21:09:10 PM UTC
Write orapwd file to Standby database test_db
Finished December 08, 2020 21:09:19 PM UTC
********************************************************************************
****************
Step 5: Configure Primary database (Primary site)
Description: DG Config service creation for db test_db - ConfigurePrimary
Job ID: ad8486a2-bea6-47ff-a8ab-330be98987ab
Started December 08, 2020 21:09:19 PM UTC
Configure host DNS on primary env
Configure Data Guard Tns on primary env
Enable Data Guard related Db parameters for primary env
Enable force logging and archivelog mode in primary env
Enable FlashBack
Configure network parameters for local listener on primary env
Restart listener on primary env
Create services for primary db
Finished December 08, 2020 21:09:35 PM UTC
********************************************************************************
****************
Step 6: Configure Standby database (Standby site)
Description: DG Config service creation for db test_db - ConfigureStandby
Job ID: 47876bb8-8fef-41a6-a0b6-92b184c74900
Started December 08, 2020 21:09:37 PM UTC
Configure Data Guard Tns on standby env
Configure host DNS on standby env
Clear Data Guard related Db parameters for standby env
Enable Data Guard related Db parameters for standby env
Enable force logging and archivelog mode in standby env
Populate standby database metadata
Configure network parameters for local listener on standby env
Reset Db sizing and hidden parameters for ODA best practice
Restart Listener on standby env
Create services for standby db
Finished December 08, 2020 21:11:02 PM UTC
********************************************************************************
****************
Step 7: Configure and enable Data Guard (Primary site)
Description: DG Config service creation for db test_db - ConfigureDg
Job ID: 64674e50-3eea-48d0-a25e-bc61401a3d71
Started December 08, 2020 21:11:02 PM UTC
```

```
Config and enable Data Guard
Post check Data Guard configuration
Finished December 08, 2020 21:12:21 PM UTC
********************************************************************************
***************
Step 8: Enable Flashback (Standby site)
Description: DG Config service creation for db test_db - EnableFlashback
Job ID: 6f461b2f-88e1-41bf-9ca8-d91bc7d4bf97
Started December 08, 2020 21:12:22 PM UTC
Enable FlashBack
Finished December 08, 2020 21:12:33 PM UTC
********************************************************************************
***************
Step 9: Re-enable Data Guard (Primary site)
Description: DG Config service creation for db test_db - ReenableDg
Job ID: 3a4a404d-e1b5-4c4e-9db1-545676eebc77
Started December 08, 2020 21:12:34 PM UTC
Re-enable Data Guard if inconsistent properties found
Post check Data Guard configuration
Finished December 08, 2020 21:12:36 PM UTC
********************************************************************************
***************
Step 10: Create Data Guard status (Primary site)
Description: DG Status service creation for db test_db - NewDgconfig
Job ID: 572dea8d-5682-4193-a984-0aebb072be45
Started December 08, 2020 21:12:36 PM UTC
Create Data Guard status
Finished December 08, 2020 21:12:39 PM UTC
********************************************************************************
***************
Step 11: Create Data Guard status (Standby site)
Description: DG Status service creation for db test_db - NewDgconfig
Job ID: 75e55d7f-49c3-4e2d-8907-a7130cc05034
Started December 08, 2020 21:12:40 PM UTC
Create Data Guard status
Finished December 08, 2020 21:12:41 PM UTC
********************************************************************************
***************
Configure Data Guard test_dg completed
********************************************************************************
***************
```

5. Check the status of Oracle Data Guard and ensure it is working:

```
# odacli describe-dataguardstatus
```

6. Switch over to the standby database:

```
# odacli switchover-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u
houdg02
```

7. Deconfigure Oracle Data Guard. The process also deletes the standby database on the old Oracle Database Appliance hardware machine.

```
# odacli deconfigure-dataguard -i 8396aac4-f7e7-42e4-a2e8-0cbbc89ce31b
Standby site address: test_domain
```

```
root@test_domain's password:
BUI username for Standby site. If Multi-user Access is disabled on Standby
site, enter 'oda-admin';
otherwise, enter the name of the user who has irestored the Standby
database (default: oda-admin):
BUI password for Standby site:
Standby database will be deleted after Data Guard configuration is
removed. Are you sure you want to proceed? (Y/N): y
Deconfigure Dataguard Started
********************************************************************************
****************
Step 1: Deconfigure Data Guard (Primary site)
Description: Deconfigure DG service
Job ID: d181fa4b-5bc9-431f-8f83-75ff44c934ff
Started December 09, 2020 18:59:17 PM UTC
Deconfigure Data Guard service
Cleanup broker resources
Finished December 09, 2020 19:00:14 PM UTC
********************************************************************************
****************
Step 2: Delete Data Guard status (Primary site)
Description: DG Status service creation - UpdateDgconfig
Job ID: 2873f1f2-9355-4cd2-8245-893957ba5d48
Started December 09, 2020 19:00:14 PM UTC
Update Data Guard status
Finished December 09, 2020 19:00:14 PM UTC
********************************************************************************
****************
Step 3: Delete Data Guard status (Standby site)
Description: DG Status service creation - UpdateDgconfig
Job ID: c4bdf567-6525-49f1-b59c-75866e3ce207
Started December 09, 2020 19:00:15 PM UTC
Update Data Guard status
Finished December 09, 2020 19:00:16 PM UTC
********************************************************************************
****************
Step 4: Delete Standby database (Standby site)
Description: Database service deletion with db name: test_db with id :
d8006215-bbd3-41d0-a37e-a4181adef042
Job ID: e207cac9-4f96-46fa-b120-462ae877bdc1
Started December 09, 2020 19:00:19 PM UTC
Validate db d8006215-bbd3-41d0-a37e-a4181adef042 for deletion
Database Deletion By RHP
Unregister Db From Cluster
Kill Pmon Process
Database Files Deletion
Delete File Groups of Database test_db
Finished December 09, 2020 19:01:07 PM UTC
********************************************************************************
****************
Data Guard configuration is removed
```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli Oracle Data Guard Commands
  Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.

Patching Oracle Data Guard Databases on Oracle Database Appliance

Understand how you can patch primary and standby databases on Oracle Data Guard using ODACLI commands.

Prerequisites for patching Oracle Data Guard primary and standby databases using ODACLI commands

Before patching the database homes, upload the Oracle Database clone files for the database version, to the repository. See *Updating Oracle Database Appliance Repository with Database Clone Files Using the CLI* for the procedure to update the repository with the latest Oracle Database clone files. Also ensure that the primary Oracle Database Appliance system and the standby Oracle Database Appliance system are on the same Oracle Database Appliance release. It is recommended that the two systems be on the latest Oracle Database Appliance release. See *Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line* for the procedure to patch your Oracle Database Appliance deployment.

> **Important:**
>
> Patch the standby database on Oracle Database Appliance first, before patching the primary database to ensure that all functions are updated correctly.
> Patching the standby first may affect Oracle Active Data Guard OJVM functionality. Hence, ensure that you patch the standby system first, and then patch the primary system immediately after that.

Patching Oracle Data Guard primary and standby databases using `odacli update-database` command

You can use the `odacli update-database` command to update the databases in your Oracle Data Guard configuration. Follow these steps for patching databases in your Oracle Data Guard configuration:

Patching workflow with primary database A and standby database B:

1. Create prepatch report for database B:

   ```
   # odacli create-prepatchreport -db -dbid databaseId -to dbhomeid
   ```

2. Update standby database B:

   ```
   # odacli update-database -i databaseId -to dbhomeId
   ```

3. Switch roles between the primary database and standby database in your Oracle Data Guard configuration, by running the following command on the primary system:

```
# odacli switchover-dataguard -i dgid -u db_A
```

4. Create prepatch report for database A:

```
# odacli create-prepatchreport -db -dbid databaseId -to dbhomeid
```

5. Update database A:

```
# odacli update-database -i databaseId -to dbhomeid
```

6. Switch roles again between the primary database and standby database in your Oracle Data Guard configuration, by running the following command on the primary system:

```
# odacli switchover-dataguard -i dgid -u db_A
```

7. Create a prepatch report for the datapatch for database A:

```
# odacli create-prepatchreport -dp -dbid databaseId
```

8. Apply datapatch on database A:

```
# odacli update-database -i databaseId -dp
```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Patching Oracle Data Guard primary and standby databases using `odacli update-dbhome` command

Follow these steps to patch the databases in your Oracle Data Guard configuration on Oracle Database Appliance:

1. Run the following command on the standby database:

```
# odacli update-dbhome
```

2. After patching job on the standby database is successful, run the following command on the primary database:

```
# odacli update-dbhome
```

3. Confirm that the patching jobs are successful:

```
# /opt/oracle/dcs/bin/odacli list-jobs
```

4. Confirm the existing Data Guard configuration is working properly by running the following command on both primary and standby systems:

```
odacli describe-dataguardstatus -i dgid
```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Upgrading Oracle Data Guard Databases on Oracle Database Appliance

Understand how you can upgrade primary and standby databases on Oracle Data Guard using ODACLI commands.

Upgrading Oracle Data Guard primary and standby databases using ODACLI commands

Before upgrading the database homes, upload the RDBMS Clone Files for the database version, to the repository. See *Updating Oracle Database Appliance Repository with Database Clone Files Using the CLI* for the procedure to update the repository with the latest RDBMS Clone Files. Ensure that the primary Oracle Database Appliance system and the standby Oracle Database Appliance system are on the same Oracle Database Appliance release, and on Oracle Database Appliance release 19.8 or later. It is recommended that the two systems be on the latest Oracle Database Appliance release. See *Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line* for the procedure to patch your Oracle Database Appliance deployment.

Follow these steps to upgrade the databases in your Oracle Data Guard configuration on Oracle Database Appliance:

1. Run the following command to disable log transport and apply:

   ```
   EDIT DATABASE primary_database SET STATE='TRANSPORT-OFF';
   EDIT DATABASE standby_database SET STATE='APPLY-OFF';
   ```

2. Upgrade the standby database:

   ```
   # odacli upgrade-database -i standby_databaseId -to destination_DbHomeId
   ```

3. Upgrade the primary database:

   ```
   # odacli upgrade-database -i primary_databaseId -to destination_DbHomeId
   ```

4. Run the following command to enable log transport and apply:

   ```
   EDIT DATABASE primary_database SET STATE='TRANSPORT-ON';
   EDIT DATABASE standby_database SET STATE='APPLY-ON';
   ```

5. Confirm the existing Data Guard configuration is working properly by running the following command on both primary and standby systems several times:

   ```
   odacli describe-dataguardstatus -i dgid
   ```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli Oracle Data Guard Commands
  Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.

Enabling Oracle Active Data Guard on Existing Configurations

Understand how you can enable Oracle Active Data Guard on existing configurations.

Oracle Active Data Guard is licensed option to the Oracle Database Enterprise Edition and enables advanced capabilities that extend the basic Oracle Data Guard functionality.

Oracle Active Data Guard function requires database to be open as read-only while enabling the *redo log apply* option. Follow these steps to enable Oracle Active Data Guard on existing configurations:

1. Modify database to have read-only start option:

   ```
   srvctl modify database -d standby_db_unique_name -startoption "read only"
   srvctl stop database -d standby_db_unique_name
   srvctl start database -d standby_db_unique_name
   ```

   To check if Oracle Data Guard has read-only start option set, run the following command:

   ```
   srvctl config database -d standby_db_unique_name | grep "Start options"
   ```

2. Enable redo log apply by running the following PL/SQL commands:

   ```
   ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
   ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
   ```

Registering Oracle Data Guard on Oracle Database Appliance

Oracle Database Appliance enables you to migrate Oracle Data Guard configuration that were manually configured, and register the configuration metadata with Oracle Database Appliance using ODACLI commands.

Prerequisites for Oracle Database Guard Registration

Before you begin:

- Oracle recommends running the primary and the standby databases on separate Oracle Database Appliance hardware, so ensure that you have at least two separate Oracle Database Appliance machines.
- Oracle recommends that the primary and standby systems have the same Oracle Database Appliance configuration. The database must have identical configuration for database version and storage (both must have either Oracle ASM or Oracle ACFS storage) to avoid any unpredictability with the database switch roles.
- The primary and standby systems must be the same Oracle Database Appliance release, and must be on Oracle Database Appliance release 19.15 or later.
- If you have customized the operating system, then ensure that environments on both machines are identical.
- Ensure that your deployment follows Oracle Maximum Availability Architecture (MAA) best practices. See the Oracle Maximum Availability Architecture (MAA) page on Oracle Technology Network.

Registering Oracle Data Guard using ODACLI commands

Follow these steps:

1. Register Oracle Data Guard with Oracle Database Appliance, in one of the following ways: Register Oracle Data Guard using interactive mode:

```
[root@node1 ~]# odacli register-dataguard
Standby site address: test_domain
BUI username for Standby site (default: oda-admin):
BUI password for Standby site:
root@node1's password:
Database name for Data Guard configuration: provdb1
Primary database SYS password:
********************************************************************************
****************
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port (TCP): 1521
Standby database listener port (TCP): 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: provdb1_sprovdb
Does the above Data Guard configuration match your actual configuration?
(Y/N, default:N): y
********************************************************************************
****************
Register Data Guard provdb1_sprovdb started
********************************************************************************
****************
Step 1: Validate register Data Guard configuration request (Primary site)
Description: Validate DG Config Creation for db provdb1
Job ID: 41459fe0-10cd-4575-9a9f-ca29c552e198
Started February 22, 2022 14:15:02 PM PST
Validate if database ID exists
Validate if dg config name exists
Validate database role
Validate if database is configured with Data Guard already
Validate tnsnames.ora
Validate database connection
Validate if data guard in good status
Precheck switchover DataGuard
Validate if input matches DGMGRL output
Validate if flashback enabled
Finished February 22, 2022 14:15:23 PM PST
********************************************************************************
****************
Step 2: Validate register Data Guard configuration request (Standby site)
Description: Validate DG Config Creation for db provdb1
Job ID: a915d750-897c-4af7-9149-6699dd1a1016
Started February 22, 2022 14:15:26 PM PST
Validate if database ID exists
Validate if dg config name exists
Validate database role
Validate if database is configured with Data Guard already
```

```
Validate tnsnames.ora
Validate database connection
Validate if data guard in good status
Validate if input matches DGMGRL output
Validate if flashback enabled
Finished February 22, 2022 14:15:34 PM PST
********************************************************************************
****************
Step 3: Create Data Guard status (Primary site)
Description: DG Status operation for db provdb1 - RegisterDg
Job ID: dee12ea8-8979-44cc-93ce-40a3a35b228a
Started February 22, 2022 14:15:35 PM PST
Create Data Guard status
Finished February 22, 2022 14:15:37 PM PST
********************************************************************************
****************
Step 4: Create Data Guard status (Standby site)
Description: DG Status operation for db provdb1 - RegisterDg
Job ID: 8e8b794f-26d5-44f4-bea6-98e443050e05
Started February 22, 2022 14:15:38 PM PST
Create Data Guard status
Finished February 22, 2022 14:15:40 PM PST
********************************************************************************
****************
Register Data Guard provdb1_sprovdb completed
********************************************************************************
****************
```

Register Oracle Data Guard with dialogue and user input:

```
[root@node1 ~]# odacli register-dataguard
Standby site address: test_domain
BUI username for Standby site (default: oda-admin):
BUI password for Standby site:
root@node1's password:
Database name for Data Guard configuration: provdb1
Primary database SYS password:
********************************************************************************
****************
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port (TCP): 1521
Standby database listener port (TCP): 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: provdb1_sprovdb
Does the above Data Guard configuration match your actual configuration?
(Y/N, default:N): n
********************************************************************************
****************
Primary site network for Data Guard configuration [Public-network]
(default: Public-network):
Standby site network for Data Guard configuration [Public-network]
(default: Public-network):
```

```
Primary database listener port (TCP) (default: 1521):
Standby database listener port (TCP) (default: 1521):
Transport type [ASYNC, FASTSYNC, SYNC] (default: ASYNC): SYNC
Protection mode [MAX_PROTECTION, MAX_PERFORMANCE, MAX_AVAILABILITY]
(default: MAX_PERFORMANCE):
Data Guard configuration name (default: provdb1_sprovdb): test
********************************************************************************
****************
Register Data Guard test started
********************************************************************************
****************
Step 1: Validate register Data Guard configuration request (Primary site)
Description: Validate DG Config Creation for db provdb1
Job ID: 83a90a68-746d-47cd-a75b-d6d758061c9b
Started February 18, 2022 16:34:53 PM PST
Validate if database ID exists
Validate if dg config name exists
Validate database role
Validate tnsnames.ora
Validate database connection
Validate database connection
Validate if data guard in good status
Validate if data guard in good status
Finished February 18, 2022 16:34:58 PM PST
Step 1 failed, the job ID of the failed job on Primary site is
83a90a68-746d-47cd-a75b-d6d758061c9b
[root@scaoda8m014c1 ~]# odacli describe-job -i 83a90a68-746d-47cd-a75b-
d6d758061c9b

Job details
----------------------------------------------------------------
                     ID:  83a90a68-746d-47cd-a75b-d6d758061c9b
            Description:  Validate DG Config Creation for db provdb1
                 Status:  Failure
                Created:  February 18, 2022 4:34:52 PM PST
                Message:  DCS-10001:Internal error encountered:
dgConfigName in request test does not match DGMGRL result provdb1_sprovdb.

Task Name                                Start
Time                          End Time                          Status
----------------------------------------
---------------------------------- ----------------------------------
----------
Validate register Data Guard configuration task February 18, 2022 4:34:53
PM PST    February 18, 2022 4:34:56 PM PST    Failure
Validate if database ID exists           February 18, 2022 4:34:53 PM
PST    February 18, 2022 4:34:53 PM PST    Success
Validate if dg config name exists        February 18, 2022 4:34:53 PM
PST    February 18, 2022 4:34:53 PM PST    Success
Validate database role                   February 18, 2022 4:34:53 PM
PST    February 18, 2022 4:34:53 PM PST    Success
Validate tnsnames.ora                    February 18, 2022 4:34:53 PM
PST    February 18, 2022 4:34:53 PM PST    Success
Validate database connection             February 18, 2022 4:34:53 PM
PST    February 18, 2022 4:34:55 PM PST    Success
Validate if data guard in good status    February 18, 2022 4:34:55 PM
```

```
PST    February 18, 2022 4:34:56 PM PST    Success
Validate if data guard in good status    February 18, 2022 4:34:56 PM
PST    February 18, 2022 4:34:56 PM PST    Failure
```

Register Oracle Data Guard using JSON file:

```
[root@node1 ~]# odacli register-dataguard -r dg.json
Standby site address: test-domain
BUI username for Standby site (default: oda-admin):
BUI password for Standby site:
root@node1's password:
Primary database SYS password:
********************************************************************************
****************
Register Data Guard provdb1_sprovdb started
********************************************************************************
****************
Step 1: Validate register Data Guard configuration request (Primary site)
Description: Validate register dataguard status request for db provdb1
Job ID: 400f0616-41dc-488f-acaa-8ba04ed23757
Started February 11, 2022 13:25:10 PM PST
Validate if database ID exists
Validate if dg config name exists
Validate database role
Validate tnsnames.ora
Validate database connection
Validate if data guard in good status
Validate if flashback enabled
Finished February 11, 2022 13:25:16 PM PST
********************************************************************************
****************
Step 2: Validate register Data Guard configuration request (Standby site)
Description: Validate register dataguard status request for db provdb1
Job ID: 29966afc-49a9-4efc-855e-5c4832d3ae47
Started February 11, 2022 13:25:17 PM PST
Validate if database ID exists
Validate if dg config name exists
Validate database role
Validate tnsnames.ora
Validate database connection
Validate if data guard in good status
Validate if flashback enabled
Finished February 11, 2022 13:25:22 PM PST
********************************************************************************
****************
Step 3: Create Data Guard status (Primary site)
Description: DG Status operation for db provdb1 - NewDgconfig
Job ID: a2457631-77e1-4322-9c3a-798943302e13
Started February 11, 2022 13:25:23 PM PST
Create Data Guard status
Finished February 11, 2022 13:25:25 PM PST
********************************************************************************
****************
Step 4: Create Data Guard status (Standby site)
Description: DG Status operation for db provdb1 - NewDgconfig
```

```
Job ID: 3534febb-b362-4f60-bc50-f2a8b8d42f46
Started February 11, 2022 13:25:27 PM PST
Create Data Guard status
Finished February 11, 2022 13:25:28 PM PST
********************************************************************************
***************
Register Data Guard provdb1_sprovdb completed
********************************************************************************
***************
```

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Example JSON for registering Oracle Data Guard on Oracle Database Appliance

```
[root@node1 ~]# cat dg.json
{
  "name" : "provdb1_sprovdb",
  "protectionMode" : "MAX_PERFORMANCE",
  "replicationGroups" : [ {
    "sourceEndPoints" : [ {
      "endpointType" : "PRIMARY",
      "databaseUniqueName" : "provdb1",
      "ipAddress" : "10.32.65.225",
      "hostName" : "host1.domain.com",
      "listenerPort" : 1521,
      "serviceName" : "provdb.domain.com",
      "sysPassword" : null
    } ],
    "targetEndPoints" : [ {
      "endpointType" : "STANDBY",
      "databaseUniqueName" : "sprovdb",
      "ipAddress" : "10.32.65.230",
      "hostName" : "host2.domain.com",
      "listenerPort" : 1521,
      "serviceName" : "sprovdb.domain.com",
      "sysPassword" : null
    } ],
    "transportType" : "ASYNC"
  } ]
}
```

Related Topics

- https://www.oracle.com/in/database/technologies/high-availability/maa.html

Deconfiguring Oracle Data Guard on Oracle Database Appliance

Deconfigure Oracle Data Guard status using ODACLI commands.

Deconfiguring Oracle Data Guard Status using ODACLI commands

Follow these steps:

- To deconfigure Oracle Data Guard and delete the standby database, run the following command on the primary system:

```
# odacli deconfigure-dataguard -i 8396aac4-f7e7-42e4-a2e8-0cbbc89ce31b
Standby site address: test_domain
root@test_domain's password:
BUI username for Standby site. If Multi-user Access is disabled on Standby
site, enter 'oda-admin';
otherwise, enter the name of the user who has irestored the Standby
database (default: oda-admin):
BUI password for Standby site:
Standby database will be deleted after Data Guard configuration is
removed. Are you sure you want to proceed? (Y/N): y
Deconfigure Dataguard Started
********************************************************************************
****************
Step 1: Deconfigure Data Guard (Primary site)
Description: Deconfigure DG service
Job ID: d181fa4b-5bc9-431f-8f83-75ff44c934ff
Started December 09, 2020 18:59:17 PM UTC
Deconfigure Data Guard service
Cleanup broker resources
Finished December 09, 2020 19:00:14 PM UTC
********************************************************************************
****************
Step 2: Delete Data Guard status (Primary site)
Description: DG Status service creation - UpdateDgconfig
Job ID: 2873f1f2-9355-4cd2-8245-893957ba5d48
Started December 09, 2020 19:00:14 PM UTC
Update Data Guard status
Finished December 09, 2020 19:00:14 PM UTC
********************************************************************************
****************
Step 3: Delete Data Guard status (Standby site)
Description: DG Status service creation - UpdateDgconfig
Job ID: c4bdf567-6525-49f1-b59c-75866e3ce207
Started December 09, 2020 19:00:15 PM UTC
Update Data Guard status
Finished December 09, 2020 19:00:16 PM UTC
********************************************************************************
****************
Step 4: Delete Standby database (Standby site)
Description: Database service deletion with db name: test_db with id :
d8006215-bbd3-41d0-a37e-a4181adef042
Job ID: e207cac9-4f96-46fa-b120-462ae877bdc1
Started December 09, 2020 19:00:19 PM UTC
Validate db d8006215-bbd3-41d0-a37e-a4181adef042 for deletion
Database Deletion By RHP
Unregister Db From Cluster
Kill Pmon Process
Database Files Deletion
Delete File Groups of Database test_db
Finished December 09, 2020 19:01:07 PM UTC
********************************************************************************
```

```
****************
Data Guard configuration is removed
```

For deconfiguring Oracle Data Guard, you must specify the Oracle Data Guard Configuration ID. Run the `odacli list-dataguardstatus` command to find the Oracle Data Guard Configuration ID.

For more information about these command options, see the *Oracle Database Appliance Command Line Reference* chapter in this guide.

Related Topics

- odacli Oracle Data Guard Commands
  Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.

14

Managing CPU Pools on Oracle Database Appliance Bare Metal and KVM Deployments

Use CPU pools to manage CPU resources on Oracle Database Appliance bare metal and KVM deployments.

- About CPU Pools
  CPU pools enable management of CPU resources on Oracle Database Appliance on both bare metal and KVM deployments. There are three types of CPU pools, that is, BM, VM, and DB system.
- Creating a CPU Pool in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a CPU pool in a bare metal or KVM deployment.
- Remapping CPU Pools in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to remap CPU pools in a bare metal or KVM deployment.
- Viewing CPU Pools in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to view all CPU pools or details about a CPU pool in a bare metal or KVM deployment.
- Modifying a CPU Pool in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to modify a CPU pool in a bare metal or KVM deployment.
- Deleting a CPU Pool in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a CPU pool in a bare metal or KVM deployment.
- Viewing CPU Pool Audit Records in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a CPU pool in a bare metal or KVM deployment.
- About Over-Subscribed CPU Pools
  Configure virtual machines so that they do not subscribe to more CPU resources than the resources available in CPU pools.

About CPU Pools

CPU pools enable management of CPU resources on Oracle Database Appliance on both bare metal and KVM deployments. There are three types of CPU pools, that is, BM, VM, and DB system.

Use the BM CPU pool to ensure that the database on a bare metal host runs exclusively on the CPU lists specified in the CPU pool. All Oracle processes are bound to the CPUs in this CPU pool and run only on these CPUs. The bare metal CPU pool is assigned to database by updating the `init.ora` parameter `processor_group_name`.

Use the VM CPU pools to cage and manage CPU resource allocations to workloads for virtual machines. Workloads are isolated by creating CPU pools and assigning or pinning virtual

machines to a specific CPU pool. When you pin a virtual machine to a CPU pool, you ensure that the virtual machine uses CPUs in only that CPU pool.

The DB system CPU pool is similar to the VM CPU pool, but it only applies to a DB system.

Resources in CPU pools cannot overlap, that is, CPU pools cannot share CPUs. But one CPU pool can be assigned to multiple databases (BM CPU pool), or VMs (VM CPU pools), or DB systems (shared DB system CPU pools).

Oracle Database Appliance provides ODACLI commands to create, update, modify, and delete CPU pools.

By default, the first core of the socket is reserved and cannot be used by the VM and DB system CPU pools for optimal performance. This behavior is available from release 19.12 and later when NUMA is enabled. If you want to use the first core of the socket in your CPU pool, then use the `--use-reserved-cores` option to enable use of the first core of socket for VM and DB system CPU pools.

About Bare Metal CPU Pools

For new databases, you can attach the new CPU pool using the `create-database` command. For existing databases, use the `modify-database` command to attach the CPU pool to the database. Restart the database manually for changes to take effect. Clone, irestore and recover database operations will also try to use the same CPU pool if the CPU pool is used by the source database. Specify the `--cpupool` option to use the new CPU pool. The standby database uses the CPU pool configured for the primary database. Specify the `--cpupool` option in the `irestore-database` command when restoring the backup to the standby site if you want to use a separate CPU pool on the standby site. For more details about the command syntax, see the *Oracle Database Appliance Command Line Interface* chapter in this guide.

About VM CPU Pools

For new VMs, you can attach the new CPU pool using the `create-vm` command. For existing VMs, use the `modify-vm` command to attach or update the CPU pool. Use the `--live` option to apply the changes to a running VM, and use the `--config` parameter to persist the change. For more details about the command syntax, see the *Oracle Database Appliance Command Line Interface* chapter in this guide.

About DB System CPU Pools

Creating a DB system automatically creates an internal DB system CPU pool associated with this DB system. The internal DB system CPU pool is created and managed internally. It cannot be created, modified, or deleted manually. Starting with Oracle Database Appliance release 19.11, you can create a new type of shared DB system CPU pool which can be shared among DB systems. For more details, see the topic *Managing Shared CPU Pool with Database System in a KVM Deployment* in this guide.

About Audit Records for CPU Pools

CPU pool operations history are recorded in audit records metadata and system log for audit purposes. The CPU pool changes include create, modify, delete operations, and attach and detach operations from VMs and databases. The CPU pool audit records can be queried using `odacli list-auditrecords` and `odacli describe-auditrecord` commands but they can not be changed or deleted. For more details about the command syntax, see the *Oracle Database Appliance Command Line Interface* chapter in this guide.

Understanding the Effect of Changing the CPU Cores on CPU Pool Allocation

You may have changed the CPU core count on Oracle Database Appliance using the `odacli modify-cpucore` command. You must assign cores in multiples of two, with a minimum of two cores. If you change the CPU core count, then you can subsequently only increase the CPU core count.

If you change the CPU core count in an Oracle Database Appliance environment which has CPU pools, then the following behavior is seen:

- `odacli create-cpupool` and `odacli modify-cpupool` will always select the online CPUs. Therefore, on both bare metal and KVM deployments, any CPU pool created after running the `odacli modify-cpucore` command, will select from the available online CPUs.
- For existing CPU pools on bare metal and KVM deployments, some CPUs can become offline when you run the `odacli modify-cpucore` command. Check the CPU pools for any offline CPUs using the `odacli describe-cpupool` command. You can then update the CPU pool to use available ONLINE CPUs by using the `odacli modify-cpupool` command.
- If the databases use a CPU pool and some CPUs become offline, then use the `odacli modify-cpupool` command to update the CPU pool. Restart the databases to use the updated CPU pool.

Related Topics

- Managing Shared CPU Pool with DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to manage shared CPU pool with DB system in a KVM deployment.
- Oracle Database Appliance Command-Line Interface
  The command-line interface has different classes of tools to manage Oracle Database Appliance.

Creating a CPU Pool in a Bare Metal or KVM Deployment

Use ODACLI commands or the Browser User Interface to create a CPU pool in a bare metal or KVM deployment.

Using ODACLI to Create a CPU Pool

Use the command `odacli create-cpupool` to create a CPU pool.

Create a VM CPU pool named `cpupool1` with 6 cores associated to it.

```
# odacli create-cpupool -n cpupool1 -c 6 -vm
```

Using Browser User Interface to Create a CPU Pool

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **CPU Pool** to display the CPU Pool page.

4. Click **Create CPU Pool**.
5. In the Create CPU Pool page, specify the following:
   - **CPU Pool Name**: Name of the CPU pool
   - **CPU Pool Type**: Specify if the CPU pool is for bare metal or virtual system.
   - **Number of CPU Cores**: Number of CPU cores allocated to the pool
   - **Socket to Use**: Specify the sockets to use for the CPU pool.
   - **Configure on**: For high-availability deployments, select the nodes where you want to configure the CPU pool.
   - **Use Reserved CPU Cores**: Select this option to use reserved CPU cores.
6. Click **Create**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Related Topics

- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- Remapping CPU Pools in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to remap CPU pools in a bare metal or KVM deployment.

Remapping CPU Pools in a Bare Metal or KVM Deployment

Use ODACLI commands or the Browser User Interface to remap CPU pools in a bare metal or KVM deployment.

Using ODACLI to Remap CPU Pools

When you patch the bare metal system to Oracle Database Appliance release 19.12 and later, NUMA is enabled after patching. Then, use the command `odacli remap-cpupools` to remap CPU pools.

See the topic *About Enabling NUMA on Oracle Database Appliance* for more information.

Using Browser User Interface to Remap CPU Pools

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **CPU Pool** to display the CPU Pools page.

4. Click **Remap CPU Pool** on the page. This will remap all CPU pools for NUMA allocation.
5. In the confirmation dialog, select **Force Run** if you want to force the remap operation. Select **Use Reserved CPU Cores** to use reserved CPU cores.
6. Click **Yes** to confirm the action.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Related Topics

- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Viewing CPU Pools in a Bare Metal or KVM Deployment

Use ODACLI commands or the Browser User Interface to view all CPU pools or details about a CPU pool in a bare metal or KVM deployment.

Using ODACLI to View CPU Pools

The command `odacli describe-cpupool` displays details about a CPU pool. Use the command `odacli list-cpupools` to view all CPU pools in the deployment.

```
# odacli list-cpupools


# odacli describe-cpupool -n cpupool1
```

Using Browser User Interface to View CPU Pools

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **CPU Pool** to display the CPU Pool page.
4. You can view the list of all configured CPU pools.
5. To view the details of a CPU pool, click on the CPU pool or click on the Actions drop down and select View Details.

Modifying a CPU Pool in a Bare Metal or KVM Deployment

Use ODACLI commands or the Browser User Interface to modify a CPU pool in a bare metal or KVM deployment.

Using ODACLI to Modify CPU Pool

Use the command `odacli modify-cpupool` to modify a CPU pool.

```
# odacli modify-cpupool -n cpupool1 -c 10
```

Restart the associated databases to use the updated CPU pool.

Using Browser User Interface to Modify CPU Pool

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **CPU Pool** to display the CPU Pools page.
4. From the Actions drop down list for the CPU pool you want to modify, select **Modify**.
5. In the Modify CPU Pool page, specify any changes in the Number of CPU Cores, and select if you want to apply the changes to running VMs or if you want to force the modify operation. Then click **Modify**.
6. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
7. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Deleting a CPU Pool in a Bare Metal or KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a CPU pool in a bare metal or KVM deployment.

Using ODACLI to Delete CPU Pool

Use the command `odacli delete-cpupool` to delete a CPU Pool.

Delete a CPU pool named `cpupool1`.

```
# odacli delete-cpupool -n cpupool1
```

Using Browser User Interface to Delete CPU Pool

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **CPU Pool** to display the CPU Pool page.
4. From the Actions drop down list for the CPU pool you want to delete, select **Delete**.
5. Click Yes to confirm your choice.
6. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
7. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Viewing CPU Pool Audit Records in a Bare Metal or KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a CPU pool in a bare metal or KVM deployment.

Using ODACLI to View Audit Record Details for CPU Pools

Use the `odacli list-auditrecords` command to list the audit records for a CPU Pool:

```
# odacli list-auditrecords
```

Use the `odacli describe-auditrecord` command to describe the audit record for a CPU Pool.

```
# odacli describe-auditrecord
```

Using Browser User Interface to View Audit Record Details for CPU Pools

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Security** tab.
3. Select **Show Advanced Search** and specify the filters for the CPU audit records details such as the From and To dates, Resource type, Resource name, DCS user name, Operation type, and the number of records to display the CPU Pool details.
4. From the list of audit records displayed, click on the audit record ID for which you to drill down for more details.

About Over-Subscribed CPU Pools

Configure virtual machines so that they do not subscribe to more CPU resources than the resources available in CPU pools.

A CPU resource can belong to one, and only one, CPU pool. You can assign multiple virtual machines to a CPU pool. A CPU pool becomes oversubscribed when the virtual machines that are active in the pool require more CPUs than you configured for the pool. For example, if a CPU pool has four CPUs, then you can start two virtual machines that are defined to use four CPUs each. In this case, the CPU pool is oversubscribed, because each of the four CPUs is supporting two virtual machines.

If you stop one of those virtual machines, but start another one that requires two CPUs, then the CPU pool is still oversubscribed. This is because two of the CPUs are supporting both virtual machines. When oversubscribing a CPU pool, assess the performance of the virtual machines in that pool. Reassign one or more virtual machines to a different CPU pool if sharing an oversubscribed pool degrades performance to unacceptable levels.

15

Managing an Oracle Database Appliance KVM Deployment

KVM virtualization uses a kernel-based virtual machine (KVM) to create a virtualized environment for your Linux applications.

Understand the Oracle Database Appliance KVM architecture, and procedures to deploy a guest virtual machine, manage high availability, manage CPU pools, and configure network settings for Oracle Database Appliance KVM deployment.

- About Oracle Database Appliance KVM Deployment
  You can use Oracle KVM to optimize your system resource use for databases and applications.
- About Oracle Database Appliance KVM Virtualization Architecture
  Review this topic to understand how Oracle Database Appliance KVM deployment works with Oracle Database Appliance hardware.
- About Virtual Machines and Oracle Database Appliance KVM Deployments
  Oracle Database Appliance KVM deployment is designed to run and deploy virtual machines to manage system resources.
- About KVM DB Systems on Oracle Database Appliance
  Understand how you can deploy KVM-based DB systems on Oracle Database Appliance to run your Oracle Database environments.
- Deploying Multiple Databases on DB Systems on Oracle Database Appliance
  Understand the guidelines and procedure to deploy multiple databases on a DB system on Oracle Database Appliance.
- Managing DB Systems in KVM Deployment
  Use ODACLI to create, list, describe, start, stop, and delete DB systems in an Oracle Database Appliance KVM deployment.
- Setting Up and Configuring a KVM Deployment for Applications
  Understand the steps to set up and configure KVM deployment for applications.
- Managing VM Storage in KVM Deployment
  Use ODACLI to create, view, modify, and delete VM storage in an Oracle Database Appliance KVM deployment.
- Managing Virtual Networks in KVM Deployment
  Use ODACLI to create, view, start, stop, and delete virtual networks in an Oracle Database Appliance KVM deployment.
- Managing Virtual Disks in KVM Deployment
  Use ODACLI to create, view, clone, modify, and delete virtual disks on Oracle Database Appliance KVM deployment.
- Managing Virtual Machines in KVM Deployment
  Use ODACLI to create, view, clone, modify, start, stop, and delete virtual machines in an Oracle Database Appliance KVM deployment.
- About Overcommitting Memory or CPUs in an Oracle Database Appliance KVM System
  Understand performance and other considerations before overcommiting CPU and memory for application KVM.

- Example JSON File to Create a Single-Node DB System
  Follow the JSON file example to create a JSON file to deploy a single-node DB System, with role separation, with the command `odacli create-dbsystem`.
- Example JSON File to Create a High-Availability DB System
  Follow the JSON file example to create a JSON file to deploy a High-Availability DB system, with role separation, with the command `odacli create-dbsystem`.

About Oracle Database Appliance KVM Deployment

You can use Oracle KVM to optimize your system resource use for databases and applications.

You can deploy a Kernel-based Virtual Machine (KVM) virtual platform on Oracle Database Appliance. With Oracle Database Appliance KVM deployments, you can use the capabilities of Oracle KVM to effectively allocate resources to databases and applications running on the same physical Oracle Database Appliance. Rather than simply disabling unnecessary server cores, you can use the excess capacity to host other workloads. This enables consolidation of both databases and applications, while retaining the ease of deployment and management associated with Oracle Database Appliance.

> **See Also:**
>
> For more information about supported operating systems, see the *Oracle Linux KVM User's Guide*: https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/index.html

The KVM feature provides a set of modules that enable you to use the Oracle Linux kernel as a hypervisor. KVM supports x86_64 processor architecture. By default, KVM is built into the Unbreakable Enterprise Kernel (UEK) release. KVM features are actively developed and may vary depending on platform and kernel release. If you are using Unbreakable Enterprise Kernel, you should refer to the release notes for the kernel release of your Oracle Database Appliance to obtain information about features and any known issues or limitations that may apply. KVM is supported on Oracle Linux 7.

What are the differences between KVM and Oracle VM Virtualization?

Oracle KVM makes it easy to setup and manage the virtualized environment with little virtualization expertise. With the KVM deployment, you can consolidate entire multi-tier environments within a single Oracle Database Appliance system.

The following are some of the advantages of deploying Oracle Database Appliance with the KVM option:

- Deploy database and applications in a single hardware environment.
- Use your CPU cores (and memory) efficiently.
- Use virtual local area networks (VLANs) to provide separate networks to different virtual machines.
- Use VM storage to grow storage for the virtual machine repository.
- Enables easy resource management per VM and maintainence of quality of service (QoS)
- Provides solution-in-a-box with application, middleware, and databases.

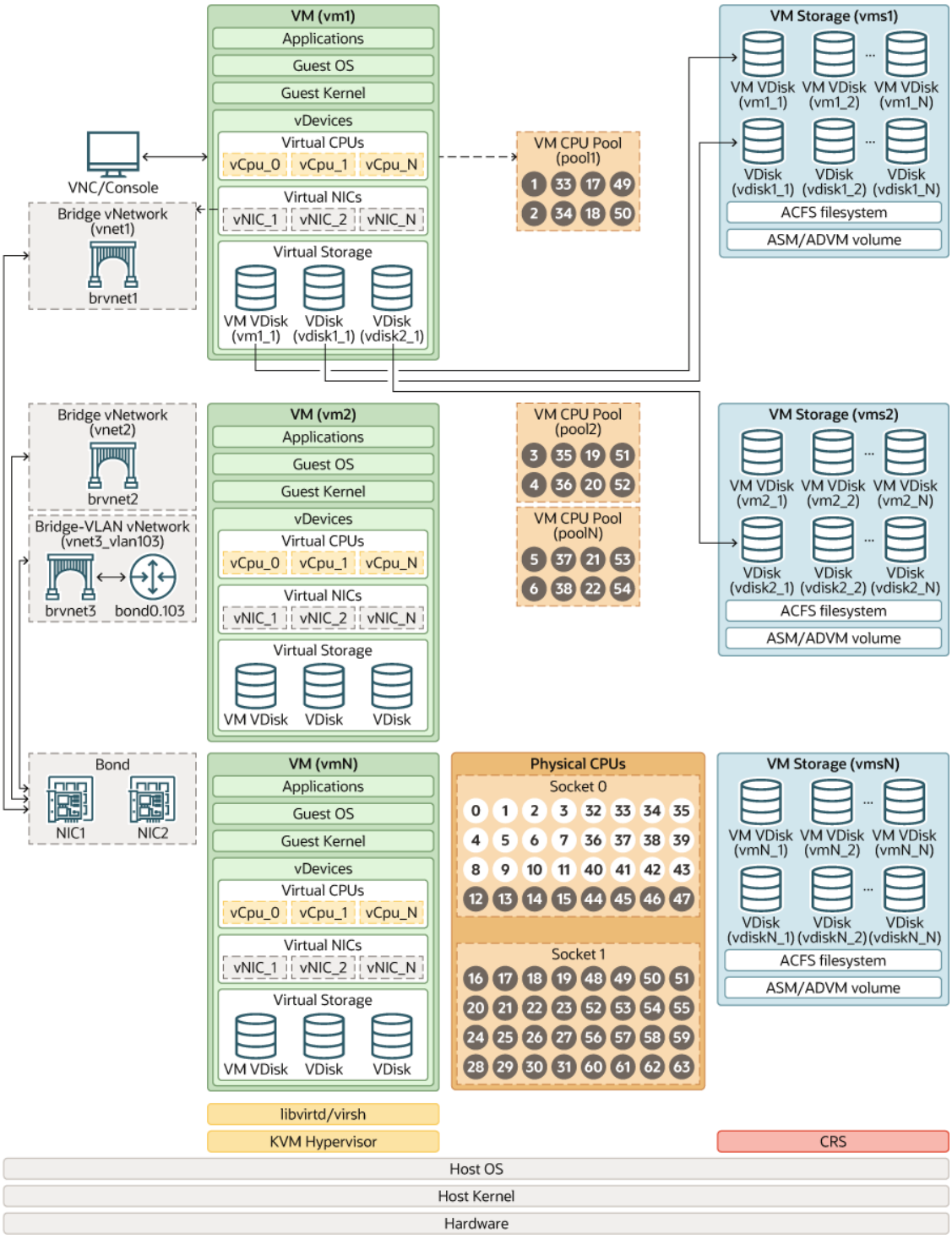
- Improves data center efficiency by increasing space utilization and reducing energy consumption and cooling costs.
- Oracle KVM virtualization is available on bare metal deployments of Oracle Database Appliance, whereas Oracle VM virtualization on Oracle Database Appliance requires Virtualized Platform setup.
- Oracle Database Appliance supports KVM on all hardware models, whereas Virtualized Platform deployments using Oracle VM are supported on only Oracle Database Appliance High-Availability models.
- Oracle Database Appliance KVM deployments use Type 2 host operating system-based hypervisor, whereas Oracle Database Appliance Virtualized Platform deployments use Type 1 bare metal hypervisor.
- KVM is the virtualization technology used in Oracle Cloud Infrastructure (OCI), where as Oracle Database Appliance Virtualized Platform is based on Xen technology.

About Oracle Database Appliance KVM Virtualization Architecture

Review this topic to understand how Oracle Database Appliance KVM deployment works with Oracle Database Appliance hardware.

Oracle Database Appliance KVM deployment provides virtualization technology that enables multiple applications to share the same physical server. The Oracle Database Appliance KVM architecture is engineered specifically to leverage the Oracle Database Appliance hardware capabilities. Oracle Database Appliance KVM stack is integrated with Oracle Clusterware. The KVM resources such as VM storages and Virtual Machines are registered as CRS resources and are automatically managed by CRS for high availability.

The Oracle Database Appliance KVM virtualization architecture uses the virtual machine components shown in the following illustration:



About Virtual Machines and Oracle Database Appliance KVM Deployments

Oracle Database Appliance KVM deployment is designed to run and deploy virtual machines to manage system resources.

Configure virtual machines on Oracle Database Appliance KVM deployment to manage the use of resources, such as the CPU pool, memory, and other shared resources by your application.

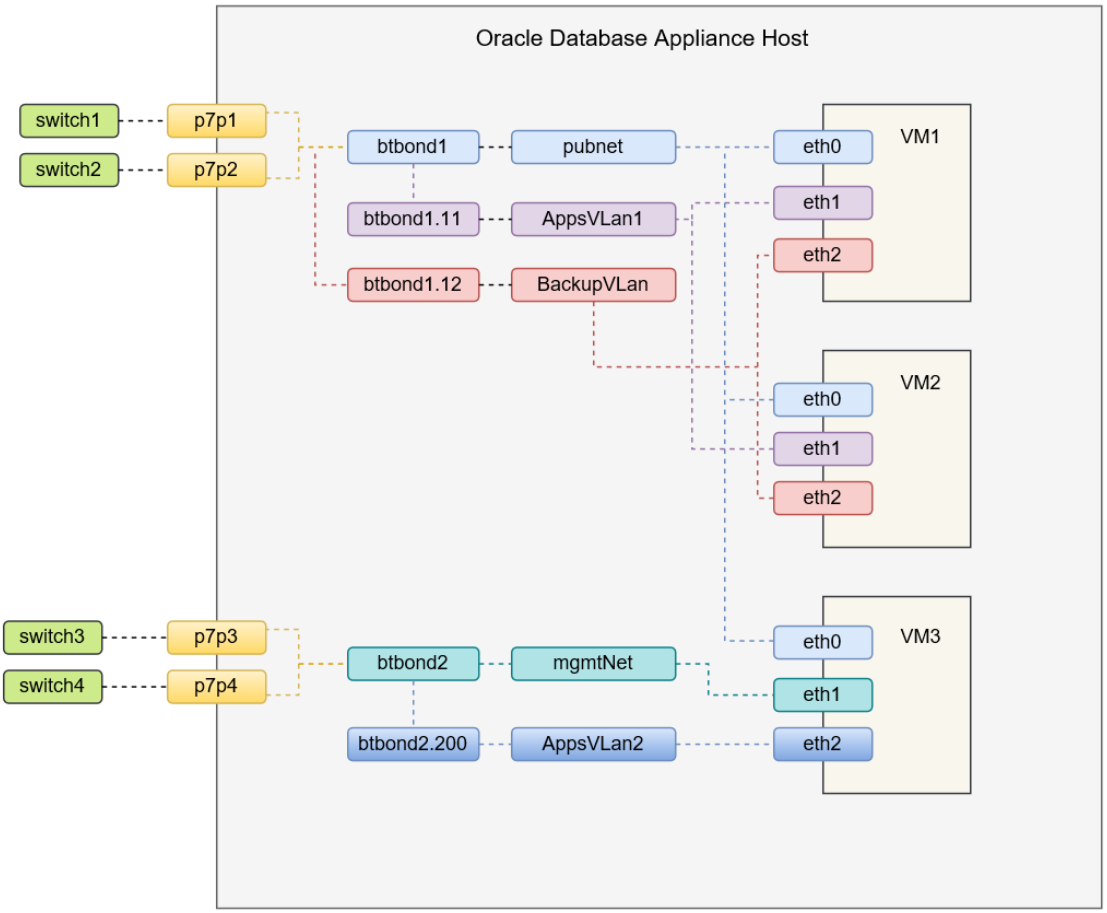
Understand the terminology of the various components you need to set up for an Oracle Database Appliance KVM deployment. The ODACLI tooling handles all the required configuration for your Oracle Database Appliance KVM deployment.

VM Storage

A VM storage is a central location for storing resources that are essential to creating and managing virtual machines. These resources include ISO files (virtual DVD images), virtual machine configuration files, and virtual disks. The VM storage is configured on an Oracle Automatic Storage Management Cluster File System (Oracle ACFS). Oracle Database Appliance KVM uses VM storage to optimize available disk space usage in the environment, and for easy reallocation of virtual machines if a physical server fails. The virtual machine files can be stored on shared disks, providing shared storage for the virtual machines. You can:

- Create one or more virtual machines on the VM storage.
- Use ODACLI commands to create and manage VM storage, virtual disks, and their virtual machines, and the underlying architecture shown in the illustration.

Virtual Networks



Oracle Database Appliance KVM virtual network supports two type of networks, bridged and bridged-vlan. The above figure is an example of the KVM virtual network on Oracle Database Appliance X8-2 hardware models.

In a bridged network, a Linux bridge is created and the network interface or bond interface, is attached to the bridge. A default bridge network named pubnet is created on the public network selected during the appliance deployment. Any VM that wants to access this public network can be attached to this pubnet vnetwork. This pubnet vnetwork cannot be modified or deleted. In the above diagram, btbond1 is used for the public network, and default vnetwork pubnet is created with `btbond1`. `eth0` of VMs are attached to this pubnet vnetwork. Except for the default pubnet bridge, no other bridged network is allowed to be created on this public network interface. Additional bridged vnetwork can be created on other interfaces that are not used for the public network. In the above diagram, the `mgmtNet` bridged network is created with `btbond2`. `eth1` of VM3 is attached to this `mgmtNet` vnetwork.

For example:

```
odacli create-vnetwork --name mgmtNet --bridge mgmtNet --type bridged --
interface btbond2 --ip ip_address --gateway gateway --netmask netmask
```

In a bridged-vlan network, VLAN can be created on all available public interfaces, including the interface where public network is already configured. For example, you can create a VLAN on both `btbond1` and `btbond2`. Follow proper procedures to configure the VLAN on the switch before creating the bridged-vlan network. In the figure above, `backupVLan` and `AppsVlan1` are created from `btbond1` and attached to `eth1` and `eth2` of VM1 and VM2 respectively.

For example:

```
odacli create-vnetwork --name backupvlan --bridge backupvlan --type bridged-
vlan --vlan-id 12 --interface btbond1 --ip ip_address --gateway gateway --
netmask netmask
```

vnetwork is not supported on private interfaces and secondary interfaces of the bond interface.

Virtual Disks

In addition to virtual machines, you can create virtual disks in VM storage. Virtual disks provide additional storage options for virtual machines by enabling you to attach additional block storage to your virtual machines. Similarly, you can detach the disk if you no longer need the additional space. You can use virtual disks to expand existing file system storage inside the virtual machine by adding the new virtual disk to an existing logical volume, or by creating a new file system on a virtual disk. Virtual disks can optionally be attached to multiple virtual machines.

Virtual Machines

A *virtual machine* is granted virtual resources, and can be started, stopped, cloned, and restarted independently. By default, virtual machines are created with autostart and failover enabled for high-availability models. During failover, the VM is automatically started, and there will be attempt to restart once before failing over to a different node on high-availability models. You can enable or disable autostart and failover using the `odacli modify-vm` command options. The option to autostart is also available on single-node Oracle Database Appliance models.

About KVM DB Systems on Oracle Database Appliance

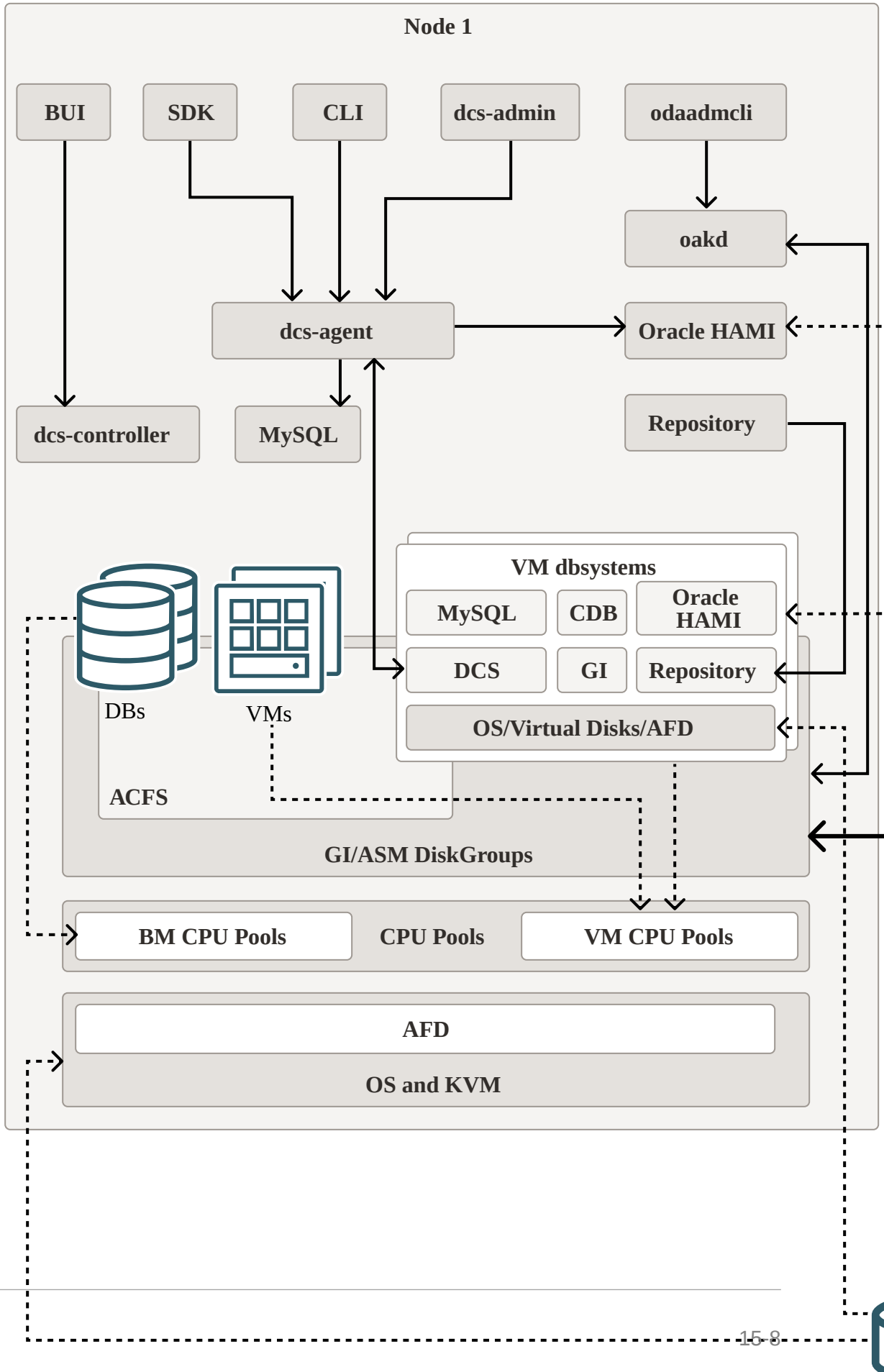
Understand how you can deploy KVM-based DB systems on Oracle Database Appliance to run your Oracle Database environments.

About KVM DB Systems

KVM DB systems enable hard partitioning for Oracle Database licensing, where each KVM DB system has its own CPU pool that is automatically assigned during KVM DB system creation. Oracle Database Appliance simplifies the management of KVM DB systems with the built-in Browser User Interface (BUI) or ODACLI Command Line Interface (ODACLI).

About KVM DB System Architecture

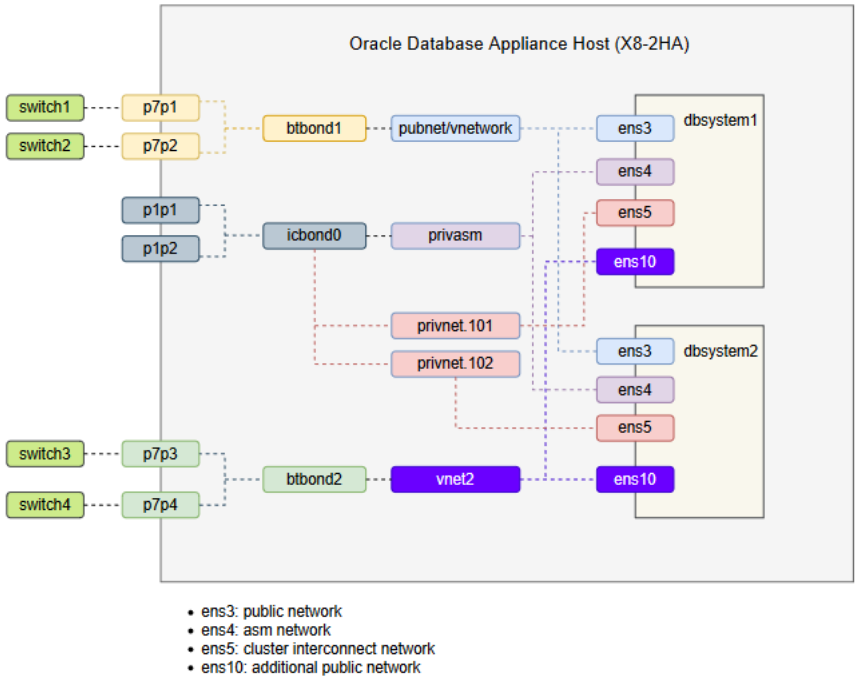
You can provision and configure a DB system on an Oracle KVM. The database can be a single-instance Oracle Database or an Oracle RAC Database with two instances. You can create multiple databases in each DB system.


Node 1
BUI
SDK
CLI
dcs-admin
odaadmcli
oakd
dcs-agent
Oracle HAMI
dcs-controller
MySQL
Repository
VM dbsystems
MySQL
CDB
Oracle HAMI
DCS
GI
Repository
OS/Virtual Disks/AFD
DBs
VMs
ACFS
GI/ASM DiskGroups
BM CPU Pools
CPU Pools
VM CPU Pools
AFD
OS and KVM

About KVM DB System Networks

Each Oracle Database Appliance DB system on KVM has the following networks created by default:

- **Public network:** The default vnetwork pubnet is created on the bare metal system, based on the public interface you select for Oracle Grid Infrastructure provisioning. When you create a DB system, the public network of the DB system is attached to the pubnet vnetwork created on the bare metal system. The pubnet network is selected by default as the public network of the DB system. You can also create a different vnetwork and use it as public network for the dbsystem.
- **Oracle ASM network:** Oracle Database Appliance DB system uses Oracle ASM to manage the database storage. The Oracle ASM instance runs on bare metal systems, but not on the DB system. The database on the DB System uses the Oracle ASM listener running on the bare metal system to communicate with Oracle ASM. The Oracle ASM network for example, privasm in the network diagram below, is created to facilitate this communication between the database on the DB system and the Oracle ASM listener on the bare metal system. This network is also used for communications between the DCS agent on the DB system and DCS agent on the bare metal system, and for enabling sharing of the central repository on bare metal system with the DB systems.
- **DB system cluster interconnect network:** When DB system is created as a two-node cluster on high-availability model, a cluster interconnect network is created automatically and it is deleted when the DB system is deleted. Each cluster owns its cluster interconnect. Different DB system cannots communicate with each other using cluster interconnect network. 192.168.16.0/24 and 192.168.17.0/24 are reserved for the use of these two internal networks. So, ensure that the IP CIDR 192.168.16.0/24 and 192.168.17.0/24 are available for use by Oracle Database Appliance DB system on KVM.
- Starting with Oracle Database Appliance release 19.12, you can create a vnetwork, either a bridged or bridged-vlan vnetwork, such as vnet2 in the image below, and then use the `odacli modify-dbsystem` command to attach the vnetwork to the dbsystem.



Restrictions When Deploying KVM DB System

The following restrictions apply when you deploy KVM DB system on Oracle Database Appliance:

- You cannot use Oracle ACFS for storage within the DB system. The database running inside the dbsystem uses Oracle ASM storage. The VM is created on Oracle ACFS based storage.
- On Oracle Database Appliance DB systems, the Oracle Grid Infrastructure software installed is of the same version as the Oracle Database version you install. The version is specified in the DB system JSON payload, with the attribute `version` for `database`.
- You cannot configure CPU pools or run VM commands within the DB system.

Deploying Multiple Databases on DB Systems on Oracle Database Appliance

Understand the guidelines and procedure to deploy multiple databases on a DB system on Oracle Database Appliance.

Supported DB System Shapes

In earlier releases of Oracle Database Appliance, where the DB system supported creation of only one database, the DB system shape was the same as the database shape. Starting with Oracle Database Appliance release 19.23, you can create multiple databases on the DB system, with a separate DB system shape dbsX, where X is the number of CPU cores on the DB system VM. X is an even number that ranges from 2 to N-2 where N is the maximum number of CPU cores on the bare metal system. You can view the value of N with the odacli list-cpucores command on the bare metal system. The default memory of the DB system is eight times that of the CPU cores of the DB system. You can create the DB system with custom memory size that is larger than the default memory size by specifying the memory size in the JSON file or BUI when you create the DB system.

Restrictions and Guidelines for Deploying Multiple Databases on DB Systems

You can provision and configure a DB system on an Oracle KVM. The database can be a single-instance Oracle Database or an Oracle RAC Database with two instances. Starting with Oracle Database Appliance release 19.23, you can create multiple databases in each DB system. For an existing appliance, after you patch the bare metal system to Oracle Database Appliance release 19.23, you can create new DB systems that support multiple databases. For existing DB systems to support multiple databases, you must patch them to Oracle Database Appliance release 19.23 first. To check whether a DB system supports multiple databases, run the odacli list-dbsystems command and verify that the shape of the DB system is *dbsx*.

- You can create multiple databases on DB systems with Oracle Database release 19c version, but cannot create different major version databases within a DB system. To create a DB system that supports 19c database, specify a 19c starter database during the DB system creation. If no starter database is specified in the DB system creation payload, a DB system that supports 19c database is created.
- You can use the `odacli create-database` command to create the first database if you did not choose to create a starter database, or additional databases in the DB system. The Oracle Grid Infrastructure home and database home are configured on the local file system `/u01` of the DB system.

- When you create a database in DB system with the `odacli create-database` command, the default database shape is `odb1`.
- DB system must be sized properly based on the sizing requirement of databases running in the DB system. Taken together, the shapes of databases running in a DB system must be no greater than the shape of the DB system, for example, databases odbA, odbB, ..., odbN running in the DB system shape dbsX, then A+B+...N must be less than or equal to X. You can change the DB system shapes to meet the sizing requirement of the databases in the DB system.
- When you create a DB system, if you specify the starter database, then Oracle Grid Infrastructure of the same release as the database is installed. If you do not specify a starter database when you create a DB system, then Oracle Grid Infrastructure of the same release as the Oracle Database Appliance is installed.
- When you delete a DB system, and you use the `--force` option with the `odacli delete-dbsystem` command, all database data files in the DB system are deleted.
- You can move databases in the DB system across different database homes using the `odacli move-database` command.
- You can restore databases with the `odacli irestore-database` command to a DB system with support for multiple databases. This may result in multiple databases running in the DB system. Make sure the DB system shape supports the additional databases restored. If not, change the DB system shape to accommodate additional databases.
- Starting with Oracle Database Appliance release 19.23, the `odacli modify-dbsystem --shape` command only changes the shape of the DB system; it does not change the shapes of the database inside the DB system any more. You must modify database shapes accordingly based on the new shape of the DB system. Ensure that you scale up database shapes after the DB system shape is scaled up. Similarly, scale down database shapes before DB system shape is scaled down, otherwise databases could fail to start because of reduced DB system memory.

Creating Multiple Databases on DB Systems

You can create databases on the DB system in one of the following ways:

- When you create the DB system using the `odacli create-dbsystem` command, you can optionally choose to create the starter database. The Oracle Grid Infrastructure software is installed and the started database is created. You specify the starter database details in the JSON file.
- If you do not specify the starter database details in the JSON file when you create the DB system, then only Oracle Grid Infrastructure is installed, but a database is not created. After the `odacli create-dbsystem` command completes successfully, you can then create multiple databases using the `odacli create-database` command.

Example JSON files to create databases on DB systems

Example JSON file to create DB system with starter database:

```
{
    "system": {
         "name": "dbsystem1",
         "diskGroup": "DATA",
         "systemPassword": "password",
          "timeZone": "Pacific/Majuro",
          "enableRoleSeparation": false,
          "shape": "dbs4",
```

```
        "customRoleSeparation": {
            "users": [
                {
                    "name": "oracle",
                    "id": 1618,
                    "role": "oracleUser"
                }
            ],
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1018,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1019,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1020,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1021,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
                    "id": 1022,
                    "role": "asmoper"
                },
                {
                    "name": "asmdba",
                    "id": 1023,
                    "role": "asmdba"
                }
            ]
        }
    },
    "network": {
        "domainName": "us.oracle.com",
        "ntpServers": [
            "xx.xx.xxx.xx"
        ],
        "dnsServers": [
            "xxx.xxx.xx.x",
            "xxx.xx.xxx.xx"
        ],
        "scanName": "dbsystem1-scan",
        "scanIps": [
            "xx.xx.xxx.xxx",
            "xx.xx.xxx.xxx"
        ],
```

```
            "nodes": [
                {
                    "name": "node1",
                    "ipAddress": "xx.xx.xx.xxx",
                    "netmask": "xxx.xxx.xxx.x",
                    "gateway": "xx.xx.xxx.x",
                    "number": 0,
                    "vipName": "node1-vip",
                    "vipAddress": "xx.xx.xxx.xxx"
                },
                {
                    "name": "node2",
                    "ipAddress": "xx.xx.xxx.xxx",
                    "netmask": "xxx.xxx.xxx.0",
                    "gateway": "xx.xx.xxx.x",
                    "number": 1,
                    "vipName": "node2-vip",
                    "vipAddress": "xx.xx.xxx.xxx"
                }
            ]
        },
        "grid": {
            "language": "en",
            "enableAFD": false
         },
        "database": {
            "name": "rZWuZTw4",
            "uniqueName": "rZWuZTw4U",
            "domainName": "test_domain",
            "adminPassword": "password",
            "version": "19.25.0.0.241015",
            "edition": "EE",
            "type": "RAC",
            "dbClass": "IMDB",
            "shape": "odb2",
            "role": "PRIMARY",
            "targetNodeNumber": null,
            "enableDbConsole": false,
            "enableUnifiedAuditing": true,
            "redundancy": null,
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
                "dbLanguage": "AMERICAN"
            },
            "rmanBackupPassword": null,
            "level0BackupDay": null,
            "enableTDE": false,
            "tdePassword": null,
            "isCdb": true,
            "pdbName": "test_pdb",
            "pdbAdminUser": "test_pdb_admin"
        }
}
```

Example JSON file to create DB system without starter database:

```
{
    "system": {
         "name": "dbsystem1",
         "diskGroup": "DATA",
         "systemPassword": "password",
          "timeZone": "Pacific/Majuro",
          "enableRoleSeparation": false,
          "shape": "dbs4",
          "customRoleSeparation": {
               "users": [
                    {
                         "name": "oracle",
                         "id": 1618,
                         "role": "oracleUser"
                    }
               ],
               "groups": [
                    {
                         "name": "oinstall",
                         "id": 1018,
                         "role": "oinstall"
                    },
                    {
                         "name": "dbaoper",
                         "id": 1019,
                         "role": "dbaoper"
                    },
                    {
                         "name": "dba",
                         "id": 1020,
                         "role": "dba"
                    },
                    {
                         "name": "asmadmin",
                         "id": 1021,
                         "role": "asmadmin"
                    },
                    {
                         "name": "asmoper",
                         "id": 1022,
                         "role": "asmoper"
                    },
                    {
                         "name": "asmdba",
                         "id": 1023,
                         "role": "asmdba"
                    }
               ]
          }
     },
     "network": {
          "domainName": "testdomain",
          "ntpServers": [
               "xx.xx.xxx.xx"
```

```
            ],
            "dnsServers": [
                 "xxx.xxx.xx.x",
                 "xx.xx.xxx.xx"
            ],
            "scanName": "dbsystem1-scan",
            "scanIps": [
                 "xx.xx.xxx.xxx",
                 "xx.xx.xxx.xxx"
            ],
            "nodes": [
                 {
                      "name": "node1",
                      "ipAddress": "xx.xx.xx.xx",
                      "netmask": "xxx.xxx.xxx.x",
                      "gateway": "xx.xx.xxx.x",
                      "number": 0,
                      "vipName": "node1-vip",
                      "vipAddress": "xx.xx.xxx.xxx"
                 },
                 {
                      "name": "node2",
                      "ipAddress": "xx.xx.xx.xx",
                      "netmask": "xxx.xxx.xxx.x",
                      "gateway": "xx.xx.xxx.x",
                      "number": 1,
                      "vipName": "node2-vip",
                      "vipAddress": "xx.xx.xxx.xxx"
                 }
            ]
       },
       "grid": {
            "language": "en",
            "enableAFD": false
        },
       "database": null
}
```

Managing DB Systems in KVM Deployment

Use ODACLI to create, list, describe, start, stop, and delete DB systems in an Oracle Database Appliance KVM deployment.

- Creating a DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a DB system in a KVM deployment.
- Listing DB Systems in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to list DB systems in a KVM deployment.
- Describing a DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to describe a DB system in a KVM deployment.

- Modifying a DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to modify a DB system in a KVM deployment.
- Attaching and Detaching a Network for a DB System in a KVM Deployment
  Use ODACLI commands to attach or detach networks for DB system in a KVM deployment.
- Starting a DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to start a DB system in a KVM deployment.
- Stopping a DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to stop a DB system in a KVM deployment.
- Deleting a DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a DB system in a KVM deployment.
- Managing Backup, Restore, and Recovery on a DB System in a KVM Deployment
  Understand the backup, restore, and recovery operations supported on a DB system in a KVM deployment.
- Managing Shared CPU Pool with DB System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to manage shared CPU pool with DB system in a KVM deployment.
- Managing Oracle Data Guard on a DB System in a KVM Deployment
  Understand the Oracle Data Guard operations supported on a DB system in a KVM deployment.

Creating a DB System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to create a DB system in a KVM deployment.

Important:

Oracle Grid Infrastructure of the same version as the database version is installed in the DB system. Use the command `odacli describe-dbsystem-image` to check the supported database versions for the DB system. Oracle Database Appliance release 19.25 supports creation of 23ai DB system that runs Oracle Grid Infrastructure 23ai and and Oracle Database 23ai. To create 23ai DB system, download and update the repository with DBVM 23ai template (patch 36524660), Oracle Grid Infrastructure 23ai clone (patch 36524627), and Oracle Database 23ai clones (patch 36524642). Specify the starter Oracle Database 23ai version in the create-dbsystem JSON file. Use the `odacli describe-dbsystem-image` command to check the exact Oracle Database 23ai version. The 23ai DB system supports running only Oracle Database 23ai database.

NOT_SUPPORTED:

To create the DB system, access control must be enabled in Oracle ASM running on bare metal system. When the bare metal system is newly provisioned or patched to Oracle Database Appliance release 19.25, access control is already enabled, which is a prerequisite for setting appropriate file ownerships for all the files created in DB systems and on the bare metal system. Before you create the DB system, the files on Oracle ASM disk groups that do not have file ownership must be set up with ownership. If ownership is not set, then creation of DB system fails. Use the `odacli modify-dbfileattributes` command to set up the file ownerships.

Using ODACLI to Create a KVM DB System

Use the command `odacli create-dbsystem` to create a KVM DB system.

Follow these steps:

1. Download the Oracle Database Appliance KVM DB System Image from My Oracle Support to a temporary location on an external client. Refer to the release notes for details about the software for the latest release.
   For example, to create a 23ai DB system, use patch number 36524660. For 19c DB system, use patch number 32451228.

   *patch_number*_1925000_Linux-x86-64.zip

2. Unzip the software — it contains README.html and one or more zip files for the patch.

   ```
   unzip patch_number_1925000_Linux-x86-64.zip
   odacli-dcs-19.25.0.0.0-date-ODAVM-19.25.0.0.zip
   ```

3. Update the repository with the image.

   ```
   # /opt/oracle/dcs/bin/odacli update-repository -f /tmp/odacli-
   dcs-19.25.0.0.0-date-ODAVM-19.25.0.0.zip
   ```

4. Run the `odacli describe-dbsystem-image` command to query the supported Oracle Grid Infrastructure and Oracle Database versions for this DB system KVM image. The Oracle Grid Infrastructure version that supports the Oracle Database version is installed.

   ```
   # odacli describe-dbsystem-image
   DB System Image details
   ------------------------------------------------------------------------
   -----
   Component Name Supported Versions Available Versions
   ------------------- ------------------- --------------------
   DBVM 23.5.0.1.0 23.5.0.1.0
   GI 23.5.0.24.07 23.5.0.24.07
      23.5.0.24.07 23.5.0.24.07
   DB 23.5.0.24.07 23.5.0.24.07
      23.5.0.24.07 23.5.0.24.07
   DBVM 19.24.0.0.0 19.24.0.0.0
     GI 19.24.0.0.240716 19.24.0.0.240716
                         19.23.0.0.240416 not-available
                         19.22.0.0.240116 not-available
   ```

```
                         19.21.0.0.231017 not-available
                         19.20.0.0.230718 not-available
                         19.19.0.0.230418 not-available
                         19.18.0.0.230117 not-available
                         19.17.0.0.221018 not-available
                         19.16.0.0.220719 not-available
                         19.15.0.0.220419 not-available
                         19.14.0.0.220118 not-available
                         19.13.0.0.211019 not-available
                         19.12.0.0.210720 not-available
                         19.11.0.0.210420 not-available
                         21.8.0.0.221018 not-available
                         21.7.0.0.220719 not-available
                         21.6.0.0.220419 not-available
                         21.5.0.0.220118 not-available
                         21.4.0.0.211019 not-available
                         21.3.0.0.210720 not-available
   DB 19.24.0.0.240716 19.24.0.0.240716
                         19.23.0.0.240416 not-available
                         19.22.0.0.240116 not-available
                         19.21.0.0.231017 not-available
                         19.20.0.0.230718 not-available
                         19.19.0.0.230418 not-available
                         19.18.0.0.230117 not-available
                         19.17.0.0.221018 not-available
                         19.16.0.0.220719 not-available
                         19.15.0.0.220419 not-available
                         19.14.0.0.220118 not-available
                         19.13.0.0.211019 not-available
                         19.12.0.0.210720 not-available
                         19.11.0.0.210420 not-available
                         21.8.0.0.221018 not-available
                         21.7.0.0.220719 not-available
                         21.6.0.0.220419 not-available
                         21.5.0.0.220118 not-available
                         21.4.0.0.211019 not-available
                         21.3.0.0.210720 not-available
```

5. If the `odacli describe-dbsystem-image` command output shows any component as `not-available` on the system, then download the component with `supported versions` and run the `odacli update-repository` to import the component to the repository.
   To deploy Oracle Database release 19.25, deploy Oracle Grid Infrastructure release 19.25.
   To deploy 23ai DB system, ensure that the DBVM 23.4.x, Oracle Grid Infrastructure 23.4.x, and Oracle Database 23.4.x clones are shown available in the `odacli describe-dbsystem-image` command output.
6. Create the `prov.json` file as per examples provided in this chapter.
   Oracle Database 21c and later supports only Container Databases (CDB). Ensure that you set the parameter `"isCdb": true`, and provide both the `pdbName` and `pdbAdminUser` in the JSON file if provisioning an Oracle Database 21c or later DB system.
7. Run the `odacli create-dbsystem` command with the `prov.json` file on the host.

   ```
   # odacli create-dbsystem -p prov.json
   ```

8. If the `odacli create-dbsystem` command displays the message `ASM ACL setup is not completed, please run 'odacli modify-dbfileattributes' command`, then run the `odacli modify-dbfileattributes` command, and make sure the job finishes successfully before running the `odacli create-dbsystem` command.
9. If the `odacli create-dbsystem` command displays the message `css_critical configuration should be different on both nodes` on a high-availability environment, then follow the instructions below to set css_critical and ensure that the DB system functions properly when interconnect fails.
    a. Run `crsctl set server css_critical yes` on the first bare metal host. You must restart the Oracle Clusterware stack on the node for the values to take effect.
    b. Run `crsctl set server css_critical no` on the second bare metal host. You must restart the Oracle Clusterware stack on the node for the values to take effect.
    c. Run `crsctl get server css_critical` to verify the value.
    Refer to *Oracle Clusterware Administration and Deployment Guide* for more details about setting `css_critical`.

Using Browser User Interface to Create a KVM DB System

Follow these steps:

1. Log into the Browser User Interface:

    `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.
4. In the page, click **Create DB System**.
5. In the Create DB System page, provide the DB system values.
6. If you want to use an existing configuration file, click **Browse** and select the configuration file. The system information from the file is populated in the relevant fields.
7. In the System Information tab, specify the values as follows.
    - **DB System Name**: Enter the DB System name. The name cannot end with a dash (-). Do not exceed 15 characters.
    - **Domain Name**: Enter the domain name.
    - **Region**: Select the region of the world where the Oracle Database Appliance is located.
    - **Time Zone**: Select the time zone where the Oracle Database Appliance is located.
    - Select **Specify Multi-User Access Option** to enable multi-user access on the DB system. You can choose either the **Enable Multi-User Access** or **Enable Passwordless Multi-User Access** option. For more information about these features, see the chapter *Implementing Multi-User Access on Oracle Database Appliance* in this guide.
        – If you choose to enable multi-user access, then specify and confirm the **ODA Administrator User Password**, **Oracle User Password**, and **Grid User Password**. Click **Configure Multi-User Access Settings** and set the **User Password Expiry Duration**, **Session Expiration for CLI**, and **Maximum Failed Login Attempts**.

  - If you choose to enable passwordless multi-user access, then you do not need to specify any of the above passwords.

- **System Password** and **Confirm Password**: Enter the system password in both fields. The system password is the password set for UNIX and `root` users. The password must contain at least two characters each from: uppercase letters, lowercase letters, numbers (0-9), and allowed special characters #, - or _. The password must have a minimum of 9 characters and a maximum of 30 characters. Select the **Assign same password for admin, oracle, and grid users** option if you want to keep the same password for all users.
- **Disk Group for VM Storage**: Select the disk group that is used for hosting the KVM storage. Note that this is different from the database file storage for the database that runs in the KVM, which is displayed automatically from the correct Oracle ASM disk group based on the database file type.
- **CPU Pool Name**: Select the CPU pool to be associated with the DB system.
- Select **Force Run** if you want to allow the DB system to use this CPU pool even if selecting the CPU pool leads to oversubscription of the CPU pool. Select **Use Reserved CPU Cores** to use reserved CPU cores.
- **Memory Size**: Specify the memory to be allocated to the DB system. The memory size can be in KB, MB, GB, or TB. The default is GB.
- **VM Storage Redundancy**: Specify the redundancy for the VM storage, either **Mirror** or **High**.
- (Optional) **DNS Servers**: Enter addresses for one or more DNS servers.
- (Optional) **NTP Servers**: Enter addresses for one or more NTP servers.
- **Public Network**: Select from the existing virtual networks.

8. In the Network Information tab, specify the client access network details.
   - **Node Name**: For Node0, enter the host name for the primary client access network.
   - **IP Address**: Enter the virtual IP address that is shared between the nodes.
   - **Subnet Mask**: Enter the subnet mask address for the primary client access network.
   - **Gateway**: Enter the gateway address for the primary client access network.

   For two-node deployments, provide the above values for both nodes. In addition, you must also specify the following:
   - **VIP Name** and **VIP Address**: Specify the Virtual IP name and address.
   - **SCAN Name** and **SCAN IP Address**: Specify the Single Client Access Name (SCAN) and SCAN IP address.
9. In the User and Group Selection tab, configure your users and groups and specify whether or not you want to allow operating system role separation:
   - Two users with six groups: Customize Users and Groups, select **No**. Allow OS Role Separation, select **Yes**. This is the default configuration.
   - Two customized users with six customized groups: Customize Users and Groups, select **Yes**. Allow OS Role Separation, select **Yes**.
   - Single user with two groups: Customize Users and Groups, select **No**. Allow OS Role Separation, select **No**.
   - Single user with six groups: Customize Users and Groups, select **Yes**. Allow OS Role Separation, select **No**.

- Specify the **GI User**, **DB User**, **Install Group**, **DBA Oper Group**, **DBA Group**, **ASM Admin Group**, **ASM Oper Group**, **ASM DBA Group**, and the User and Group IDs.

10. In the Database Information tab, specify the following information to configure the database:

    - **DB Name**: Enter a name for the database. The name must contain alphanumeric characters and cannot exceed 8 characters.
    - **DB Version**: Specify the database version.
    - (Optional) **DB Unique Name**: Enter a globally unique name for the database. Databases with the same DB Name within the same domain (for example, copies of a database created for reporting or a physical standby) must have a different DB Unique Name that is unique within the enterprise. The name must begin with a lowercase or uppercase letter, and contain only alphanumeric characters and underscores(_). The name must not contain dollar ($), and pound (#) characters. The name cannot exceed 30 characters.
    - **CDB**: Select **Yes** or **No** to specify whether or not you want a Container Database (CDB).
    - **PDB Name**: Enter a name for the pluggable database (PDB).
    - **PDB Admin User**: Enter an Admin user name for the pluggable database (PDB). The name must begin with an alphanumeric character. You can specify alphanumeric characters and underscore (_) in the name.
    - **Database Edition**: Select the Oracle Database edition, either Standard Edition and Enterprise Edition. Your license determines which database edition you are eligible to create in the DB System.
    - **Deployment:** Select the type of Deployment, whether **RAC**, **RAC One**, or **SI**.
      For Standard Edition Oracle Database 19c or later, you can only create single-instance Oracle Database. For Standard Edition Oracle Database 19.6 or later, you can choose to enable high availability for single-instance database.

      For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, you can choose to enable high availability for single-instance databases. For Enterprise Edition Oracle Database 19.15 or later or Oracle Database 21.6 or later, if you choose to create a single-instance database, then the high availability feature is disabled by default. To enable high-availability, set the value in the **Enable High Availability** field to **Yes**.
    - **Sys and PDB Admin User Password** and **Confirm Password**: Provide a password for the database.
    - **Shape**: Select a database shape from the list.
    - **Database Redundancy:** If disk group redundancy is FLEX, then select **HIGH** or **MIRROR** from the drop down list for database redundancy. If the disk group redundancy is not FLEX, then the database redundancy field is not available and is set to the same as the disk group redundancy internally.
    - In the **Database Class** field, select a database class from the drop-down list. If an option is not available in the list, it is not supported for the database edition on the Oracle Database Appliance or the version that you selected. The default is OLTP.
    - **Configure EM Express**: Select **Yes** or **No**. Select **Yes** to configure the Oracle Enterprise Manager Database Express (EM Express) console.
    - **Character set**: Select a character set.
    - **National Characterset**: Select a national characterset.

- **Language**: Select the database language.
- **Territory**: Select a territory or location from the list.
- For Oracle Database Enterprise Edition 19c or later, you can choose to enable Transparent Database Encryption (TDE). Select **Yes** or **No** in the **Enable TDE** option. Specify and confirm the **TDE Password**. By default, the TDE option is disabled.
- **Data Files on Flash Storage**: Select **Yes** or **No**. This option is only available if the high-availability system has HDD storage drives.

11. Click **Create**.
12. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
13. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Related Topics

- Implementing Multi-User Access on Oracle Database Appliance
  Understand how multi-user access can be implemented on Oracle Database Appliance, its advantages and restrictions and the associated life cycle management changes for your appliance.
- Example JSON File to Create a High-Availability DB System
  Follow the JSON file example to create a JSON file to deploy a High-Availability DB system, with role separation, with the command `odacli create-dbsystem`.
- Example JSON File to Create a Single-Node DB System
  Follow the JSON file example to create a JSON file to deploy a single-node DB System, with role separation, with the command `odacli create-dbsystem`.
- odacli describe-dbsystem-image
  Use the `odacli describe-dbsystem-image` command to view the supported Oracle Grid Infrastructure and Oracle Database versions and the availability of the clone files and Database System KVM image in the Oracle Database Appliance system.
- odacli modify-dbfileattributes
  Use the `odacli modify-dbfileattributes` command to properly set the file attributes (permissions and ownership) of files on Oracle ASM disk group on a bare metal host to support a dbsystem. Only run this command if prompted by the `odacli create-dbsystem` command.

Listing DB Systems in a KVM Deployment

Use ODACLI commands or the Browser User Interface to list DB systems in a KVM deployment.

Using ODACLI to List DB Systems

Use the command `odacli list-dbsystems` to list DB systems.

```
# odacli list-dbsystems
```

Using Browser User Interface to List DB Systems

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.
4. In the page, click on the DB system for which you want to view details.

Related Topics

- odacli list-dbsystems
  Use the `odacli list-dbsystems` command to view the database systems on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Describing a DB System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to describe a DB system in a KVM deployment.

Using ODACLI to Describe a DB System

Use the command `odacli describe-dbsystem` to describe a DB system.

```
# odacli describe-dbsystem -n dbsystem_name
```

Using Browser User Interface to Describe a DB System

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.
4. In the page, click on the DB System for which you want to view details or click on the **Actions** drop down and select **View Details** to view the DB System details.
5. Click on the System Information tab for the DB System details, associated CPU Pool, Storage details, and associated VMs.
6. Click on the Database Information tab to view the associated database details.
7. Click on the Network Information tab to view the network details for single node or high-availability deployments.

Related Topics

- odacli describe-dbsystem
  Use the `odacli describe-dbsystem` command to describe a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Modifying a DB System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to modify a DB system in a KVM deployment.

Using ODACLI to Modify a DB System

Use the command `odacli modify-dbsystem --shape` to change shape of the DB system. Starting with Oracle Database Appliance release 19.23, the `odacli modify-dbsystem --shape` command only changes the shape of the DB system; it does not change the shapes of the database inside the DB system any more. You must modify database shapes accordingly based on the new shape of the DB system. Ensure that you scale up database shapes after the DB system shape is scaled up. Similarly, scale down database shapes before DB system shape is scaled down; otherwise databases could fail to start because of reduced DB system memory.

```
# odacli modify-dbsystem -n dbsystem_name -s dbs4 [-dvn v_networks_to_detach]
[-en] [-gw gateway] [-ip ip_address] [-nm netmask] [-sip scan_ip_address] [-
sn scan_name] [-s shape]  [-vips vip_address] [-vt v_network_type] [-m
custom_memory_size]
```

You can attach a CPU pool to the DB system, or remove the CPU pool from the DB system.

Use the command `odacli modify-dbsystem -m` to increase the DB system memory. This does not change the database memory configuration. You must manually update the database memory configuration such as SGA, PGA, and others to utilize the increased memory. The command does not support reducing DB system memory. Changing the shape of the DB system after you change the DB system memory resets the memory of the DB system.

You can increase the DB system memory with the `odacli modify-dbsystem` command using the `-m` option.

```
# odacli modify-dbsystem -n dbsystem_name -m 24G
```

To reduce the DB system memory, use the `--shape` option with the command `odacli modify-dbsystem`. The DB system is automatically restarted for the new shape to take effect.

```
# odacli modify-dbsystem --name dbsystem1 --shape dbs2
```

Use the command `odacli modify-dbsystem` to attach or detach a new network to the DB system. This new network can be used for database backup, Oracle Data Guard setup or other purposes in the DB system.

You can use `odacli modify-dbsystem` to attach or detach the shared DB system CPU pool. Attaching a shared DB system CPU pool removes the internal CPU pool that was created at DB system creation time. When detaching shared CPU pool from the DB system, an internal CPU pool is automatically created and attached to the DB system. One shared DB system CPU pool can be attached to multiple DB systems. Oversubscribing to shared DB system CPU pool is allowed, but there could be potential performance impact if shared CPU pool is oversubscribed.

> **Note:**
>
> The `odacli modify-dbsystem -s shape` command only works on Oracle Database Appliance release 19.13 and later DB systems. The CPU pool commands are available on Oracle Database Appliance release 19.11 and later DB systems.
> In Oracle Database Appliance release 19.23 and later, the `odacli modify-dbsystem -s` shape command changes the DB system shape only; it does not modify the database shape. Run the `odacli modify-database` command in the DB system to change the database shape in the DB system.

Using Browser User Interface to Modify a KVM DB System

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.
4. In the page, click on the DB System for which you want to view details or click on the **Actions** drop down and select **Modify** to change the following:
   - DB System shape
   - Attach or detach CPU pool for the DB system
   - Attach or detach networks
   - Enable or disable NUMA for the DB system
   - Memory size allocated to the DB system
   - VM storage redundancy (either Mirror or High)
5. Select the new shape and CPU pool and click **Modify**.
6. Click **Enable NUMA** to set NUMA capabilities, and click **Modify**.
7. Click the Detach Networks field to view the networks, select the network you want to detach, and click **Modify**.
8. To attach a network, specify the **Name**, **IP Address**, **Subnet Mask**, **Gateway**, **Network Type**, **Default Network**, and click **Modify**.
9. Confirm your action to submit the job to modify the DB System. You can manually run the `odacli modify-database` command inside the DB system to change the database shape.

Related Topics

- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Attaching and Detaching a Network for a DB System in a KVM Deployment

Use ODACLI commands to attach or detach networks for DB system in a KVM deployment.

Using ODACLI to Attach or Detach a Network to a DB System

You can attach a new vnetwork to the DB system and use it for database backup, Oracle Data Guard configuration, and other options. Use the `odacli modify-dbsystem` to attach or detach network for a DB system in a KVM deployment. The vnetwork must exist before you can attach it to the dbsystem.

Follow these steps to attach a vnetwork to a DB system:

1. Identify the name of vnetwork bridge to attach to the DB system using the `odacli list-vnetworks` and `odacli describe-vnetwork -n` *`vnetwork_name`* commands.
2. Use the `odacli modify-dbsystem` command to attach vnetwork to DB system:

   ```
   # odacli modify-dbsystem -n dbsystem1 -avn vnet2 -t Dataguard -ip
   192.168.10.119,192.168.10.120 -nm 255.255.240.0 -gw 192.168.10.1 -sn scan1
   -sip 192.168.10.30,192.168.10.31 -vips
   vip1:0:192.168.15.57,vip2:1:192.168.15.58
   ```

   The IP address, netmask (nm) and gateway (gw) values are used to configure the new interface of the DB system. The SCAN name (sn), SCAN IP address (sip) and Virtual IP address (vip) are used when the vnetwork type is `database` or `dataguard`.

To detach a vnetwork in the DB system, run the following command:

```
# odacli modify-dbsystem -n name -dvn vnetwork_name
```

Using Browser User Interface to Modify a DB System

Follow these steps:

1. Log into the Browser User Interface:

   `https://`*`host-ip-address`*`:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.
4. In the page, click on the DB System for which you want to view details or click on the **Actions** drop down and select **Modify** to attach or detach vnetworks for the DB system.
5. Click the Detach Networks field to view the vnetworks, select the vnetwork you want to detach, and click `Modify`.

6. To attach a network, specify the **Name**, **IP Address**, **Subnet Mask**, **Gateway**, **Network Type**, **Default Network**, and click `Modify`.
7. Confirm your action to submit the job to modify the DB system.

Starting a DB System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to start a DB system in a KVM deployment.

Using ODACLI to start a DB System

Use the command `odacli start-dbsystem` to start a DB system.

```
# odacli start-dbsystem -n dbsystem_name
```

Using Browser User Interface to Start a DB System

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.
4. In the page, click on the DB System which you want to start.
5. Click on the System Information tab and then click **Start**.
6. For high-availability deployments, select the node on which you want to start the DB System. Click **Yes**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.
9. When the DB System starts, the state displays as ONLINE.

Related Topics

- odacli start-dbsystem
  Use the `odacli start-dbsystem` command to start a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Stopping a DB System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to stop a DB system in a KVM deployment.

Using ODACLI to Stop a DB System

Use the command `odacli stop-dbsystem` to stop a DB system.

```
# odacli stop-dbsystem -n dbsystem_name
```

Using Browser User Interface to Stop a DB System

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.
4. In the page, click on the DB System which you want to stop.
5. Click on the System Information tab and then click **Stop**.
6. For high-availability deployments, select the node on which you want to stop the DB System. Click **Yes**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.
9. When the DB system stops, the state displays as OFFLINE.

Related Topics

- odacli stop-dbsystem
  Use the `odacli stop-dbsystem` command to stop a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Deleting a DB System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a DB system in a KVM deployment.

Using ODACLI to Delete a KVM DB System

Use the command `odacli delete-dbsystem` to delete a KVM DB system.

```
# odacli delete-dbsystem -n dbsystem_name -f
```

If you do not specify the `-f` option, then the operation deletes the DB system and Oracle Clusterware files, and retains the database files. With the `-f` option, the operation deletes the DB system, Oracle Clusterware files, and the database files.

Using Browser User Interface to Delete a KVM DB System

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **DB Systems** to display the DB Systems page.

4. In the page, for DB System which you want to delete, click on the **Actions** drop down list and select **Delete**.
5. Confirm that you want to delete the DB System. Select **Force Delete** if you want to delete the DB system, Oracle Clusterware files, and the database files during the delete operation. Click **Yes**.
6. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
7. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Related Topics

- odacli delete-dbsystem
  Use the `odacli delete-dbsystem` command to delete a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Managing Backup, Restore, and Recovery on a DB System in a KVM Deployment

Understand the backup, restore, and recovery operations supported on a DB system in a KVM deployment.

About Backup, Restore, and Recovery on a DB System

Similar to the bare metal systems, you can use ODACLI commands to configure and perform backup, restore, and recovery operations on DB systems. For information about backup, restore, and recovery on Oracle Database Appliance, see the chapter *Backup, Recover and Restore* in this guide.

Related Topics

- Backup, Restore and Recover Databases
  Review backup, restore, and recovery operations for your databases.

Managing Shared CPU Pool with DB System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to manage shared CPU pool with DB system in a KVM deployment.

Using ODACLI to Manage Shared CPU Pool with DB system

Creating a DB system automatically creates an internal DB system CPU pool associated with this DB system. This DB system pool is managed internally and cannot be created, modified, or deleted manually. You can create a new type of shared CPU pool using the `odacli create-cpupool --dbsystem/-dbs -n cores` command. Similarly to VM and BM types, this CPU pool must not overlap with other CPU pools. In high-availability environments, the DB system CPU pool cannot be local, it must exist on both nodes always.

Associating Shared DB System CPU Pool

A shared DB system CPU pool can be associated to a new DB system or existing DB system. To create a new DB system with shared DB system CPU pool, add the "cpuPoolName" parameter in the `create-dbsystem json` file. The associated internal CPU pool is automatically

deleted after a shared CPU pool is attached to a DB system. Use the `odacli modify-dbsystem` command to associate the CPU pool to an existing DB system.

```
# odacli modify-dbsystem -n dbsystem_name -cp dbspool1
```

The cores of the DB System (based on its shape) must fit into the shared DB System CPU pool. Associating a shared DB system CPU pool to an existing DB system removes the internal CPU pool associated to the DB system. A single shared DB System CPU pool can be shared by multiple DB Systems. Oversubscription to the DB system CPU pool is allowed, but performance may be impacted if the CPU pool is oversubscribed. The list of associated DB systems can be queried using `odacli list-cpupool` or `odacli describe-cpupool` commands.

Use `odacli describe-dbsystem` to check the details of the CPU pool associated to the DB system:

```
# odacli describe-dbsystem -n dbs4e912c
 CPU Pool
--------------------------
                     Name:  ce3f42bb28
           Number of cores:  2

                     Host:  n1
         Effective CPU set:  9-10,29-30
               Online CPUs:  9, 10, 29, 30
              Offline CPUs:  NONE

                     Host: n2
         Effective CPU set:  9-10,29-30
               Online CPUs:  9, 10, 29, 30
              Offline CPUs:  NONE
```

Use `odacli list-cpupools` to view the DB system CPU pool type and associated DB systems. The internal DB system CPU pool has type "DB_SYSTEM" and shared DB system CPU pool has type "DB_SYSTEM_SHARED".

```
# odacli list-cpupools
Name                 Type               Configured on              Cores
Associated resources           Created
Updated
-------------------  -----------------  -------------------------  -----
----------------------------  ----------------------
----------------------
bmpool1              BM                 n1,             2
NONE                           2021-03-30 17:54:59 UTC  2021-03-30 17:55:00
UTC

n2

vmpool1              VM                 n1,             2
NONE                           2021-03-30 17:54:48 UTC  2021-03-30 17:54:49
UTC

n2
```

```
dbspool1              DB_SYSTEM_SHARED    n1,             4
NONE                            2021-03-30 17:55:11 UTC  2021-03-30 17:55:11
UTC

n2

ce3f42bb28            DB_SYSTEM           n1,             2
dbs4e912c                       2021-03-30 17:57:43 UTC  2021-03-30 18:01:27
UTC
                                          n2
```

Use `odacli describe-cpupool` to find out more details about DB system CPU pool:

```
# odacli describe-cpupool -n ce3f42bb28
CPU Pool
details
--------------------------------------------------------------------------------
--
                        ID:  8e62933b-b394-4bcf-9c32-6a4cea2e0360
                      Name:  ce3f42bb28
                   Created:  2021-03-30 17:57:43 UTC
                   Updated:  2021-03-30 18:01:27 UTC
                      Type:  DB_SYSTEM
           Number of cores:  2
      Associated resources:  dbs4e912c

 CPU Allocations
-------------------------
                      Node:  n1
         Effective CPU set:  9-10,29-30
               Online CPUs:  9, 10, 29, 30
              Offline CPUs:  NONE

                      Node:  n2
         Effective CPU set:  9-10,29-30
               Online CPUs:  9, 10, 29, 30
              Offline CPUs:  NONE
```

Dissociating Shared DB System CPU Pool

A DB System with a shared DB System CPU pool could have the shared CPU pool detached from it and an internal DB System CPU Pool is created. If the internal DB system CPU pool cannot be created, then the `odacli modify-dbsystem` command fails.

Dissociating shared DB System CPU Pool:

```
# odacli modify-dbsystem -n dbs4e912c -no-cp
```

Modifying Shared DB System CPU Pool

A shared DB system CPU pool could be resized if all the associated DB systems shapes still fit into the cores of the pool (CPU pool size >= individual DB system size) The change is effective immediately on running DB system VMs.

Modifying a DB System Associated with a Shared DB System CPU Pool

If a DB System is associated with a shared DB System CPU Pool and its shape is modified with the `odacli modify-dbsystem` command, then the new shape of the DB system must fit into the cores of the pool.

Deleting Shared DB System CPU Pool

A shared DB system CPU Pool can be deleted using the `odacli delete-cpupool` command only if it has no associated DB systems.

Related Topics

- Oracle Database Appliance KVM Hard Partitioning Compliance
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- Remapping CPU Pools in a Bare Metal or KVM Deployment
  Use ODACLI commands or the Browser User Interface to remap CPU pools in a bare metal or KVM deployment.
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Managing Oracle Data Guard on a DB System in a KVM Deployment

Understand the Oracle Data Guard operations supported on a DB system in a KVM deployment.

About Using Oracle Data Guard on a DB System

Similar to the bare metal systems, you can configure and use Oracle Data Guard on DB systems using ODACLI commands. You can set up Oracle Data Guard between the bare metal system and DB system, or between two DB systems. Integrated Oracle Data Guard can also be used for migrating from bare metal system to DB system, or between two DB systems. For information about Oracle Data Guard on Oracle Database Appliance, see the chapter *Using Oracle Data Guard for Disaster Management and Recovery on Oracle Database Appliance* in this guide.

Related Topics

- Using Oracle Data Guard for Disaster Management and Recovery on Oracle Database Appliance
  Understand how you can use Oracle Data Guard for disaster management and recovery on Oracle Database Appliance.

Setting Up and Configuring a KVM Deployment for Applications

Understand the steps to set up and configure KVM deployment for applications.

Follow these steps to set up a KVM deployment on Oracle Database Appliance

1. Create a VM storage.
2. Create a virtual network.
3. Create virtual disks.
4. Create virtual machines.

The following links provide information about performing these steps:

Related Topics

- Creating a VM Storage in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a VM storage in a KVM deployment.
- Creating a Virtual Network in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a virtual network in a KVM deployment.
- Creating a Virtual Disk in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a virtual disk in a KVM deployment.
- Creating a Virtual Machine in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a virtual machine in a KVM deployment.

Managing VM Storage in KVM Deployment

Use ODACLI to create, view, modify, and delete VM storage in an Oracle Database Appliance KVM deployment.

- Creating a VM Storage in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a VM storage in a KVM deployment.
- Viewing VM Storage in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to view all VM storage or details about a VM storage in a KVM deployment.
- Modifying VM Storage in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to modify VM storage in a KVM deployment.
- Deleting a VM Storage in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a VM storage in a KVM deployment.

Creating a VM Storage in a KVM Deployment

Use ODACLI commands or the Browser User Interface to create a VM storage in a KVM deployment.

Using ODACLI to Create VM Storage

Use the command `odacli create-vmstorage` to create a VM Storage.

Create a VM storage named `share1` of 8 GB.

```
# odacli create-vmstorage -n share1 -s 8G
```

Using Browser User Interface to Create VM Storage

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **Show VM Instances** to display the VM Instances page.
4. In the page, select the **Create VM Storage** option and click **Next**.
5. In the Create VM Storage page, specify the following:
   - **Storage Name:** Name of the VM storage
   - **Storage Size:** Size of the storage to be allocated
   - **ASM Disk group:** Select the Oracle ASM disk group
   - **Redundancy:** Select the VM storage redundancy, either Mirror or High
6. Click **Create**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Viewing VM Storage in a KVM Deployment

Use ODACLI commands or the Browser User Interface to view all VM storage or details about a VM storage in a KVM deployment.

Using ODACLI to View VM Storage

The command `odacli describe-vmstorage` displays details about VM storage. Use the command `odacli list-vmstorages` to view all VM storages in the deployment.

```
# odacli list-vmstorages
```

```
# odacli describe-vmstorage
```

Using Browser User Interface to View VM Storage

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show VM Storage** tab to view the list of all configured VM storages.
5. Click on a VM Storage to view the details of the VM storage.

Modifying VM Storage in a KVM Deployment

Use ODACLI commands or the Browser User Interface to modify VM storage in a KVM deployment.

Using ODACLI to Modify VM Storages

Use the command `odacli modify-vmstorage` to modify VM storage.

Increase the size of a VM storage named `share1` by 10 gigabytes.

```
# odacli modify-vmstorage -n share1 -i 10G
```

Decrease the size of a VM storage named `share1` by 5 gigabytes.

```
# odacli modify-vmstorage -n share1 -s 5G
```

Using Browser User Interface to Modify VM Storage

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show VM Storage** tab.
5. In the page, select a VM Storage, select the **Modify** option and click **Next**.
6. In the Modify VM Storage page, specify **Increment** in size, the VM storage **Redundancy** as either **Mirror** or **High**, and click **Modify**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Deleting a VM Storage in a KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a VM storage in a KVM deployment.

Using ODACLI to Delete VM Storage

Use the command `odacli delete-vmstorage` to delete a VM storage.

Delete a VM storage named `vs1`.

```
# odacli delete-vmstorage -n vs1
```

Using Browser User Interface to Modify VM Storage

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show VM Storage** tab.
5. In the page, select a VM Storage, select the **Delete** option.
6. Click Yes to confirm your choice.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Managing Virtual Networks in KVM Deployment

Use ODACLI to create, view, start, stop, and delete virtual networks in an Oracle Database Appliance KVM deployment.

- Creating a Virtual Network in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a virtual network in a KVM deployment.
- Viewing Virtual Networks in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to view all virtual networks or details about a virtual network in a KVM deployment.
- Starting and Stopping Virtual Networks in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to start or stop virtual networks in a KVM deployment.
- Modifying a Virtual Network in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to modify a virtual network in a KVM deployment.

- Deleting a Virtual Network in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a virtual network in a KVM deployment.

Creating a Virtual Network in a KVM Deployment

Use ODACLI commands or the Browser User Interface to create a virtual network in a KVM deployment.

Using ODACLI to Create Virtual Network

Use the command `odacli create-vnetwork` to create a virtual network.

Create a VM network of type `bridged` using the interface `btbond2`.

```
# odacli create-vnetwork --name mgmtNet --bridge mgmtNet --type bridged --
interface btbond2 --ip 192.168.120.26 --gateway 192.168.120.1 --netmask
255.255.255.0
```

Create a VM network of type `bridged-vlan` using the interface `btbond1`.

```
# odacli create-vnetwork --name backupvlan --bridge backupvlan --type bridged-
vlan --vlan-id 12 --interface btbond1 --ip 192.168.100.10 --gateway
192.168.100.1 --netmask 255.255.255.0
```

Using Browser User Interface to Create Virtual Network

Follow these steps:

1. Log into the Browser User Interface:

   `https://`*`host-ip-address`*`:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **Show VM Instances** to display the VM Instances page.
4. In the page, select the **Create Virtual Network** option and click **Next**.
5. In the Create Virtual Network page, specify the following:
   - Name: Name of the virtual network
   - IP Address: Virtual network IP address
   - Subnet Mask and Gateway: Virtual network subnet mask and gateway
   - Network Type: Virtual network type
   - Interface: Virtual network interface
   - Bridge Name: Name of the network bridge
   - VLAN ID: ID of the VLAN network
6. Click **Create**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.

8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Viewing Virtual Networks in a KVM Deployment

Use ODACLI commands or the Browser User Interface to view all virtual networks or details about a virtual network in a KVM deployment.

Using ODACLI to View Virtual Networks

The command `odacli describe-vnetwork` displays details about a virtual network. Use the command `odacli list-vnetworks` to view all virtual networks in the deployment.

```
# odacli list-vnetworks


# odacli describe-vnetwork -n vnet1
```

Using Browser User Interface to Create Virtual Networks

Follow these steps:

1. Log into the Browser User Interface:

   `https://`*`host-ip-address`*`:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show Virtual Networks** tab to view the list of all configured virtual networks.
5. Click on a Virtual Network to view the details of the virtual network.

Starting and Stopping Virtual Networks in a KVM Deployment

Use ODACLI commands or the Browser User Interface to start or stop virtual networks in a KVM deployment.

Using ODACLI to Start and Stop Virtual Networks

The command `odacli start-vnetwork` starts a virtual network. Use the command `odacli stop-vnetwork` to stop a virtual network in the deployment.

```
# odacli start-vnetwork -n vnet1


# odacli stop-vnetwork -n vnet1
```

Using Browser User Interface to Start and Stop Virtual Networks

Follow these steps:

1. Log into the Browser User Interface:

   `https://`*`host-ip-address`*`:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show Virtual Networks** tab to view the list of all configured virtual networks.
5. To start a virtual network, click on a Virtual Network, and then click **Start**.
6. To stop a virtual network, click on a virtual network, and then click **Stop**.

Modifying a Virtual Network in a KVM Deployment

Use ODACLI commands or the Browser User Interface to modify a virtual network in a KVM deployment.

Using ODACLI to Modify Virtual Network

Use the command `odacli modify-vnetwork` to modify a virtual network.

```
# odacli modify-vnetwork -n vnet1 -g 10.11.44.41
```

Using Browser User Interface to Modify Virtual Network

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show Virtual Networks** tab.
5. In the page, select a Virtual Network, select the **Modify** option and click **Next**.
6. In the Modify Virtual Network page, specify any changes in the Subnet Mask, Gateway, or IP Address, and click **Modify**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Deleting a Virtual Network in a KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a virtual network in a KVM deployment.

Using ODACLI to Delete Virtual Networks

Use the command `odacli delete-vnetwork` to delete a virtual network.

Delete a virtual network named `vnet1`.

```
# odacli delete-vnetwork -n vnet1
```

Using Browser User Interface to Modify Virtual Networks

Follow these steps:

1. Log into the Browser User Interface:

   ```
   https://host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show Virtual Networks** tab.
5. In the page, select a virtual network, select the **Delete** option.
6. Click Yes to confirm your choice.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Managing Virtual Disks in KVM Deployment

Use ODACLI to create, view, clone, modify, and delete virtual disks on Oracle Database Appliance KVM deployment.

- Creating a Virtual Disk in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a virtual disk in a KVM deployment.
- Viewing Virtual Disks in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to view all virtual disks or details about a virtual disk in a KVM deployment.
- Cloning a Virtual Disk in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to clone a virtual disk in a KVM deployment.
- Modifying a Virtual Disk in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to modify a virtual disk in a KVM deployment.
- Deleting a Virtual Disk in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a virtual disk in a KVM deployment.

Creating a Virtual Disk in a KVM Deployment

Use ODACLI commands or the Browser User Interface to create a virtual disk in a KVM deployment.

Ensure that a VM storage exists before you create a virtual disk.

Using ODACLI to Create Virtual Disks

Use the command `odacli create-vdisk` to create a virtual disk.

Specify the name of the virtual disk in the command `odacli create-vdisk -n name`. Identify the storage in which you create the virtual disk by using the `-st storage` option. Use the option `-s size` to specify the virtual disk size. If you want to configure the virtual disk as a shared disk, then use the option `-sh`.

The vdisk name must start with a letter followed by underscores, hyphens, or alphanumeric characters and can have a maximum length of 30 characters.

Create a 2 GB non-sparse and shareable virtual disk named `vdisk1` inside the VM storage `vms1`.

```
odacli create-vdisk -n vdisk1 -st vms1 -s 2G -sh
```

Using Browser User Interface to Create Virtual Disks

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the Virtual Disk page.
4. Click **Create Virtual Disk**.
5. In the page, select the **Create Virtual Disk** option and click **Next**.
6. In the Create Virtual Disk page, specify the following:
   - **Virtual Disk Name:** Name assigned to the virtual disk that is unique within the name repository
   - **VM Storage Name:** Name of the VM storage where the virtual disk will be created
   - **Disk Size:** Size of the virtual disk
   - **Shared:** Specify if you want to share the virtual disk
   - **Sparse or Non-Sparse:** Specify if the virtual disk is sparse
7. Click **Create**.
8. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
9. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Viewing Virtual Disks in a KVM Deployment

Use ODACLI commands or the Browser User Interface to view all virtual disks or details about a virtual disk in a KVM deployment.

Using ODACLI to View Virtual Disks

The command `odacli describe-vdisk` displays details about a virtual disk. Use the command `odacli list-vdisks` to view all virtual disks in the deployment.

```
# odacli list-vdisks
```

```
# odacli describe-vdisk -n vdisk_name
```

Using Browser User Interface to Create Virtual Disks

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show Virtual Disks** tab to view the list of all configured virtual disks.
5. Click on a virtual disk to view the details of the virtual disk.

Cloning a Virtual Disk in a KVM Deployment

Use ODACLI commands or the Browser User Interface to clone a virtual disk in a KVM deployment.

Using ODACLI to Clone Virtual Disks

Use the command `odacli clone-vdisk` to clone an existing virtual disk.

Specify the source vdisk from which you want to clone with the `-n` option, and specify the name of the cloned virtual disk with the `-cn` option.

Create a clone of a virtual disk named `vdisk1`, with the name `vdisk1_clone`. The cloned disk is created on the same storage as `vdisk1`.

```
# odacli clone-vdisk -n vdisk1 -cn vdisk1_clone
```

Using Browser User Interface to Create Virtual Disks

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.

3. Click **VM Instances** to display the Virtual Disk page.
4. Click **Create Virtual Disk**.
5. In the page, select the **Clone Virtual Disk** option and click **Next**.
6. In the Clone Virtual Disk page, specify the following:
   - Select the source virtual disk you want to clone
   - Specify the name of the cloned virtual disk
7. Click **Create**.
8. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
9. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Modifying a Virtual Disk in a KVM Deployment

Use ODACLI commands or the Browser User Interface to modify a virtual disk in a KVM deployment.

Using ODACLI to Modify Virtual Disks

Use the command `odacli modify-vdisk` to increase the size of a virtual disk.

Increase the size of a virtual disk named `vdisk1` by 4 gigabytes.

```
# odacli modify-vdisk -n vdisk1 -i 4G
```

Using Browser User Interface to Modify Virtual Disks

Follow these steps:

1. Log into the Browser User Interface:

   `https://`*`host-ip-address`*`:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show Virtual Disks** tab.
5. In the page, select a Virtual Disk, select the **Modify Virtual Disk** option and click **Next**.
6. In the Modify Virtual Disk page, specify Increment in size and if you want to share the virtual disk.
7. Click **Modify**.
8. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
9. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Deleting a Virtual Disk in a KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a virtual disk in a KVM deployment.

Using ODACLI to Delete Virtual Disks

Use the command `odacli delete-vdisk` to delete a virtual disk.

Delete a virtual disk named `vdisk1`.

```
# odacli delete-vdisk -n vdisk1
```

Using Browser User Interface to Modify Virtual Disks

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show Virtual Disks** tab.
5. In the page, select a Virtual Disk, select the **Delete** option.
6. Click Yes to confirm your choice.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Managing Virtual Machines in KVM Deployment

Use ODACLI to create, view, clone, modify, start, stop, and delete virtual machines in an Oracle Database Appliance KVM deployment.

- Creating a Virtual Machine in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a virtual machine in a KVM deployment.
- Cloning a Virtual Machine in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to clone a virtual machine instance in a KVM deployment.
- Modifying a Virtual Machine in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to modify a virtual machine in a KVM deployment.
- Viewing Virtual Machines in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to view all virtual machines or details about a virtual machine in a KVM deployment.

- Starting and Stopping Virtual Machines in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to start or stop virtual machines in a KVM deployment.
- Deleting a Virtual Machine in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a virtual machine in a KVM deployment.

Creating a Virtual Machine in a KVM Deployment

Use ODACLI commands or the Browser User Interface to create a virtual machine in a KVM deployment.

Ensure that a VM storage exists before you create a VM instance.

Using ODACLI to Create Virtual Machine Instance

Use the command `odacli create-vm` to create a VM instance. Specify the preferred node on which to start the virtual machine after creation (`-pn` option).

Following is an example command to create a VM named `vm1` with 8 vcpus and 8 GB memory. A VM disk of size 10 GB is created automatically as `/dev/vda` inside the VM for the operating system installation. The VM disk is stored in the VM storage `vms1`. The source used for the installation is located on `/u01/OL7.iso`. If you want to attach an additional virtual disk to the VM, you can specify `--vdisks` *`vdisk_name`*, where *`vdisk_name`* is an existing virtual disk. This virtual disk when attached, is shown as `/dev/vdb` inside the VM.

```
# odacli create-vm -n vm1 -vc 8 -m 8G -vms vms1 -src /u01/OL7.iso -size 10G
```

To use a kickstart configuration file to create the VM instance, include the file in the `--extra-args` parameter. Create a kickstart configuration file, including the network configuration for the guest VM. Ensure that you specify the network details in the `--extra-args` parameter so that the VM bootstrap can use it to access the kickstart configuration file in this format `"dns=`*`dns ip`* `ip=`*`ip`*`::`*`gateway`*`:`*`netmask`*`:`*`hostname`*`:`*`interface`*`:none"`. The IP must belong to the vnetwork to be attached to the VM, and the kickstart configuration must be available over this vnetwork.

For example:

```
odacli create-vm --name odaksvm --vnetwork pubnet --memory 4G --
source /u01/kvm/images/OL77_x86_64.iso --vmstorage myvms --size 10G --vcpus 2
--extra-args "dns=dns_IP ip=vm_IP::gateway_IP:netmask:vm_hostname:eth0:none
console=tty0 console=ttyS0,115200 ks=http://Webserver_IP:portnumber/ks.txt"
```

If you do not use a kickstart configuration file to create the VM instance, then complete the VM instance creation as follows:

1. Run the `odacli create-vm` command and check that it completes successfully.
2. Use the `odacli describe-vm` command to check the VNC port.
3. Log in to the VM instance using the VNC port: *`host:vncport`*. See the section *Connecting to the VM instance Using VNC* for the steps to connect to the VM instance using VNC.
4. Complete the installation process by providing your values for the installation screens.
5. Log in again to the VM instance using the VNC port: *`host:vncport`*. See the section *Connecting to the VM instance Using VNC* for the steps to connect to the VM instance using VNC.

6. If you specified `--vnetwork` in the `odacli create-vm` command, then you can run `ifconfig -a` to view the network interface in the VM. Configure the network as you would on a Linux system.
7. After the network is configured, log into the VM using the network and customize the VM instance.
8. If you did not specify `--vnetwork` in the `odacli create-vm` command, then the VM network is created using the default Network Address Translation (NAT) based KVM bridge `virbr0`. eth0 is configured as DHCP inside the VM.

Connecting to the VM instance Using VNC: Method 1

1. Find out the VNC display port of the VM from the command `odacli describe-vm -n` *`vm_name`*`: 127.0.0.1:1`

```
# odacli describe-vm -n vm1
VM details
--------------------------------------------------------------------------------
-----
                ID: c280af13-997c-49b1-97ce-0617610535f1
              Name: vm1
 ...
 Graphics settings: vnc,listen=127.0.0.1
      Display Port: 127.0.0.1:1 <<<<
 ...
```

2. Run vncserver on the Oracle Database Appliance host. Note down the vncserver address `odahost:11`.

```
# vncserver
New 'odahost:11 (root)' desktop is odahost:11
```

3. Launch vncviewer from your desktop, login to the Oracle Database Appliance host using the vncserver address created in step 2: odahost:11. This launches a vncviewer that logs into the Oracle Database Appliance host.
4. From vncviewer launched in step 3, run the `vncviewer` command with the display port from the `odacli describe-vm` output. For example, in the previous step, the `odacli describe-vm` command shows the display port is `127.0.0.1:1`, so type `vncviewer 127.0.0.1:1`.

Connecting to the VM instance Using VNC: Method 2

1. Find out the VNC display port of the VM from the command `odacli describe-vm`. For example, the Display Port is `Display Port: 127.0.0.1:1`, and the current node `myodahost1`.

```
# odacli describe-vm -n vm1
VM details
--------------------------------------------------------------------------------
-----
                ID: c280af13-997c-49b1-97ce-0617610535f1
              Name: vm1
 ...
 Graphics settings: vnc,listen=127.0.0.1
      Display Port: 127.0.0.1:1 <<<<
 ...
```

```
 Status
-------------------------
             Current node:  myodahost1   <<<
            Current state:  ONLINE
             Target state:  ONLINE
```

2. On Linux or macOS operating systems where you want to launch the vncviewer, use the following command to create the SSH tunnel:

   ```
   ssh -L localport:127.0.0.1:vncport root@odahost
   ```

   where localport is an available port number greater than 1024 on your local machine (for example, 12345) where vnc viewer is launched. vncport is the VNC port displayed in the odacli describe-vm command plus 5900. For example, if `odacli describe-vm` returns the display port as `127.0.0.1:1`, then use 5901 as the VNC port. `odahost1` is the host name of the Oracle Database Appliance host where the VM is running. 127.0.0.1 is the address where vnc server is listening at. Do not change the `127.0.0.1` address. For example:

   ```
   ssh -L 12345:127.0.0.1:5901 root@odahost1
   ```

   On Windows, use PuTTY to create the SSH tunnel.

3. Launch vncviewer from your desktop, enter VNC server address as `localhost:localport`, where localport is one used in the SSH tunnel in step 2 (such as 12345). For example:

   ```
   vncserver: localhost:12345
   ```

How to find VM network interface attached to the vnetwork

If you specified `--vnetwork` in the `odacli create-vm` command, then you can run `ifconfig -a` to view the network interface in the VM. Find out the VM network interface name that is attached to the vnetwork by matching the interface MAC address to the MAC address displayed in the `odacli describe-vm` command.

For example:

```
# odacli describe-vm -n vm1
VM details
--------------------------------------------------------------------------------
--
                        ID: c280af13-997c-49b1-97ce-0617610535f1
                      Name: vm1
...
 Parameters
-------------------------
   ...
                              Config Live
                              -------------------------
------------------------
                    Memory: 2.00 GB 2.00 GB
                Max Memory: 2.00 GB 2.00 GB
...
                 vNetworks: pubnet:52:54:00:15:b5:c4 pubnet:52:54:00:15:b5:c4
<<<<
```

Run "ip link show"inside the VM, find the interface name whose MAC address (the link/ether field in the ip command output: link/ether 52:54:00:15:b5:c4) that matches the vnetwork MAC address in the describe-vm output "pubnet:52:54:00:15:b5:c4", in this case it is eth0. So eth0 is the interface attached to the vnetwork "pubnet".

```
# ip link show
...
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
    link/ether 52:54:00:15:b5:c4 brd ff:ff:ff:ff:ff:ff
```

Configure the network interface eth0 as you would on a Linux system.

Using Browser User Interface to Create Virtual Machine Instance

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **Show VM Instances** to display the VM Instances page.
4. In the page, select the **Create VM Instance** option and click **Next**.
5. In the Create Virtual VM page, specify the following:
   - VM Name: Name assigned to the VM instance that is unique within the name repository
   - VM Storage Name: Name of the VM storage where the VM instance will be created
   - Source Installation: The source from which you want to create the VM
   - Preferred Node: Node where you want to run the VM instance
   - Memory Size: Size of the memory to be allocated
   - CPU Pool Name: Select the CPU Pool
   - Number of vCPUs to Use: Number of virtual CPUs to be allocated
6. Click **Create**.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Cloning a Virtual Machine in a KVM Deployment

Use ODACLI commands or the Browser User Interface to clone a virtual machine instance in a KVM deployment.

Using ODACLI to Clone Virtual Machine Instance

Use the command `odacli clone-vm` to clone an existing virtual machine.

The name of the VM you create is defined by the command `odacli clone-vm -cn name`. Specify the source VM from which you want to clone.

Create a clone of a virtual machine named `vm1`, with the name `vm1_clone`. The cloned VM is created on the same storage as `vm1`.

```
# odacli clone-vm -n vm1 -cn vm1_clone
```

Using Browser User Interface to Clone Virtual Machine Instance

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Instance page.
4. Click **Create VM Instance**.
5. In the page, select the **Clone VM Instance** option and click **Next**.
6. In the Clone VM Instance page, specify the following:
   - Select the Source VM you want to clone
   - Specify the name and description of the cloned VM instance
7. Click **Create**.
8. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
9. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Modifying a Virtual Machine in a KVM Deployment

Use ODACLI commands or the Browser User Interface to modify a virtual machine in a KVM deployment.

Using ODACLI to Modify Virtual Machines

Use the command `odacli modify-vm` to modify a virtual machine.

Update the configuration of a VM named `vm1`, setting the CPU count to 6 and the maximum memory to 6 gigabytes on both the running VM and subsequent boots.

```
# odacli modify-vm -n vm1 -c 6 -mm 6G --live --config
```

Modify a VM to attach a vnetwork:

```
# odacli modify-vm -n vm_name -avn vnetwork_name
```

- When an application VM is created, a default network interface is created by default in the VM. This interface is attached to the default kvm bridge (virbr0) at the host. This interface is intended for convenient communication between the host and VM. This interface does not attach to any physical network interface at the host. Do not configure this interface to use for any external communication.

- You must create a vnetwork using the command `odacli create-vnetwork` and attach the vnetwork (`odacli modify-vm -avn`) to the VM. This creates a second interface in the VM. Configure this network interface for your network requirement.
- The network interface name in the VM depends on the operating system.

Using Browser User Interface to Modify Virtual Machines

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show VM Instances** tab.
5. In the page, select a Virtual Machine, select the **Modify** option and click **Next**.
6. In the Modify VM page, specify Increment in size, and if you want to Auto Start, Set Failover, or Enable NUMA.
7. You can also modify the CPU Pool, Number of vCPUs to use, Memory Size, and attach and detach virtual disks and virtual networks.
8. Select if you want to save the configuration or apply the configuration to the running VM.
9. Click **Modify**.
10. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
11. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

Viewing Virtual Machines in a KVM Deployment

Use ODACLI commands or the Browser User Interface to view all virtual machines or details about a virtual machine in a KVM deployment.

Using ODACLI to View Virtual Machines

The command `odacli describe-vm` displays details about a virtual machine. Use the command `odacli list-vms` to view all virtual machines in the deployment.

```
# odacli list-vms
```

```
# odacli describe-vm -n vm_name
```

Using Browser User Interface to Create Virtual Machines

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show VM Instances** tab to view the list of all configured virtual machines.
5. Click on a VM Instance to view the details of the virtual machine.

Starting and Stopping Virtual Machines in a KVM Deployment

Use ODACLI commands or the Browser User Interface to start or stop virtual machines in a KVM deployment.

Using ODACLI to Start and Stop Virtual Machines

The command `odacli start-vm` starts a virtual machine. Use the command `odacli stop-vm` to stop a virtual machine in the deployment.

If `pref-node` is defined for the VM, then the VM starts on the `pref-node`. If `pref-node` is not defined for the VM, then the VM can start on any node, in a high-availability deployment. However, if you specify the node name `-n`, then the VM starts on the specified node, even if the preferred node is defined.

```
# odacli start-vm -n vm1


# odacli stop-vm -n vm1
```

Using Browser User Interface to Start and Stop Virtual Machines

Follow these steps:

1. Log into the Browser User Interface:

   `https://`*`host-ip-address`*`:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show VM Instances** tab to view the list of all configured virtual machines.
5. To start a virtual machine, click on a VM Instance, and then click **Start** and select the node to start the virtual machine.
6. To stop a virtual machine, click on a VM Instance, and then click **Stop** to stop the virtual machine. Click **Force Stop** to close all running processes and stop the virtual machine.

Deleting a Virtual Machine in a KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a virtual machine in a KVM deployment.

Using ODACLI to Delete Virtual Machines

Use the command `odacli delete-vm` to delete a virtual machine.

Delete a virtual machine named `vm1`.

```
# odacli delete-vm -n vm1
```

Using Browser User Interface to Modify Virtual Machines

Follow these steps:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Appliance** tab.
3. Click **VM Instances** to display the VM Details page.
4. Click the **Show VM Instances** tab.
5. In the page, select a VM Instance, select the **Delete** option.
6. Click Yes to confirm your choice.
7. When you submit the job, the job ID and a link to the job appears. Click the link to display the job status and details.
8. Validate that the job completed. You can track the job in the **Activity** tab in the Browser User Interface, or run the command `odacli describe-job` with the job ID.

About Overcommitting Memory or CPUs in an Oracle Database Appliance KVM System

Understand performance and other considerations before overcommiting CPU and memory for application KVM.

Oracle Database Appliance does not restrict overcommiting of CPU and memory for application KVM. However, it is not recommended to overcommit CPU or memory. Overcommitting CPU can lead to slow performance of the VMs and the host. Overcommitting memory can cause the system to run out of memory (OOM), which may lead to the Linux kernel shutting down important system processes. Before you decide to overcommit CPU or memory, ensure that you test your systems. When sizing the application VMs, it is recommended to reserve at least 24 GB memory and 2 CPU cores for bare metal system KVM host when there is no database running on the bare metal system host.

Oracle Database Appliance validates oversubscription of shared CPU pools for DB systems as well as memory associated with DB systems. Starting with Oracle Database Appliance release 19.15, the commands `odacli start-dbsystem`, `odacli modify-dbsystem`, `odacli create-dbsystem`, and `odacli modify-cpupool` display errors if the operations result in overcommiting of the CPU pool or memory for the DB system. You can use the `--force`/`-f` option to allow overcommit of resources for DB systems. This check applies to DB systems only, not for application VMs.

Example JSON File to Create a Single-Node DB System

Follow the JSON file example to create a JSON file to deploy a single-node DB System, with role separation, with the command `odacli create-dbsystem`.

Use the example JSON file to create a file for your environment.

> **Note:**
>
> It is important to review the readme and the examples carefully before creating your JSON file.

Example 15-1 JSON File to Create a Single-Node Oracle KVM Database System with Role Separation

The following is an example of a JSON file that creates a single-node DB 23ai DB system on Oracle Database Appliance bare metal system. To create a DB 23ai DB system, use the example JSON, and update the `version` in the `database` section to DB 23ai version. Use the `odacli describe-dbsystem-image` command to view the supported 23ai version. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "clusterName": "cl-test",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
                    "id": 1005,
                    "role": "asmoper"
                },
                {
                    "name": "asmdba",
```

```
                        "id": 1006,
                        "role": "asmdba"
                    }
                ],
                "users": [
                    {
                        "name": "grid",
                        "id": 1000,
                        "role": "gridUser"
                    },
                    {
                        "name": "oracle",
                        "id": 1001,
                        "role": "oracleUser"
                    }
                ]
            }
        },
        "database": {
            "name": "db19",
            "uniqueName": "db19",
            "domainName": "example.com",
            "version": "23.5.0.24.07",
            "edition": "EE",
            "type": "SI",
            "dbClass": "OLTP",
            "shape": "odb2",
            "role": "PRIMARY",
            "targetNodeNumber": null,
            "enableDbConsole": false,
            "enableFlashStorage": false,
            "enableFlashCache": false,
            "enableUnifiedAuditing": true,
            "enableEEHA": true,
            "enableSEHA": false,
            "redundancy": null,<<< if diskgroup redundancy is FLEX, then database
redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can be null
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
                "dbLanguage": "ENGLISH"
            },
            "enableTDE": false,
             "isCdb": true,
                "pdbName": "pdb1",
                "pdbAdminUser": "pdbadmin"
        },
        "network": {
            "domainName": "test_domain",
            "ntpServers": [
                "xx.xxx.xx.xxx"
            ],
            "dnsServers": [
                "xx.xxx.xx.xxx"
            ],
```

```
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xx.xxx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xx.x",
                "number": 0
            }
        ],
  "publicVNetwork": "vnet1"
    },
    "grid": {
        "language": "en",
        "enableAFD": true
    }
}
```

Example 15-2 JSON File to Create a Single-Node Oracle KVM Database System without Role Separation

The following is an example of a JSON file that creates a single-node Oracle KVM Database System on Oracle Database Appliance bare metal platform, without role separation. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "clusterName": "cl-test",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
```

```
                    },
                    {
                        "name": "asmoper",
                        "id": 1005,
                        "role": "asmoper"
                    },
                    {
                        "name": "asmdba",
                        "id": 1006,
                        "role": "asmdba"
                    }
                ],
                "users": [
                    {
                        "name": "grid",
                        "id": 1000,
                        "role": "gridUser"
                    },
                    {
                        "name": "oracle",
                        "id": 1001,
                        "role": "oracleUser"
                    }
                ]
            }
        },
        "database": {
            "name": "db19",
            "uniqueName": "db19",
            "domainName": "example.com",
            "version": "19.25.0.0.241015",
            "edition": "EE",
            "type": "SI",
            "dbClass": "OLTP",
            "shape": "odb4",
            "role": "PRIMARY",
            "targetNodeNumber": null,
            "enableDbConsole": false,
            "enableUnifiedAuditing": true,
            "redundancy": "HIGH",<<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
                "dbLanguage": "ENGLISH"
            },
            "enableTDE": false,
            "isCdb": true,
                "pdbName": "pdb1",
                "pdbAdminUser": "pdbadmin"
        },
        "network": {
            "domainName": "test_domain",
            "ntpServers": [],
```

```
        "dnsServers": [
            "xx.xxx.xx.xxx"
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xx.xxx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xx.x",
                "number": 0
            }
        ],
        "publicVNetwork": "vnet1"
      },
    "grid": {
        "language": "en",
         "enableAFD": true
            }
}
```

Example 15-3 JSON File to Create a Single-Node Oracle KVM Database System with the `--template/-t` option

The following is an example of a JSON file template sample that creates a single-node Oracle KVM Database System on Oracle Database Appliance bare metal platform. This template sample is generated when you run the `odacli create-dbsystem` command with the `--template/-t` option. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{

"system": {
        "name": "test_dbsystem",
        "shape": "dbs2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "clusterName" : "clus-test",
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                }
            ],
            "users": [
                {
                    "name": "grid",
                    "id": 1000,
                    "role": "gridUser"
                },
                {
                    "name": "oracle",
```

```
                "id": 1001,
                "role": "oracleUser"
            }
        ]
    }
},
"database": {
    "name": "dbtest",
    "uniqueName": "dbtest",
    "domainName": "example.com",
    "version": "19.16.0.0.220719",
    "edition": "EE",
    "type": "SI",
    "dbClass": "OLTP",
    "shape": "odb2",
    "role": "PRIMARY",
    "targetNodeNumber": "0",
    "enableDbConsole": false,
    "enableFlashStorage": false,
    "enableFlashCache": false,
    "enableUnifiedAuditing": true,
    "enableEEHA": true,
    "enableSEHA": false,
    "redundancy": "MIRROR",
    "characterSet": {
        "characterSet": "AL32UTF8",
        "nlsCharacterset": "AL16UTF16",
        "dbTerritory": "AMERICA",
        "dbLanguage": "ENGLISH"
    },
    "enableTDE": false,
    "isCdb": true,
    "pdbName": "pdb1",
    "pdbAdminUser": "pdbadmin"
},
"network": {
    "domainName": "example.com",
    "ntpServers": [
        "xxx.xxx.xxx.xxx"
    ],
    "dnsServers": [
        "xxx.xxx.xxx.xxx"
    ],
    "nodes": [
        {
            "name": "node1",
            "ipAddress": "xxx.xxx.xxx.xxx",
            "netmask": "xxx.xxx.xxx.xxx",
            "gateway": "xxx.xxx.xxx.xxx",
            "number": 0
        }
    ],
    "publicVNetwork": "vnet1"
},
"grid": {
    "language": "en",
```

```
        "enableAFD": true
    }
}
```

Example 15-4 JSON File to Create a Single-Node Oracle KVM Database System with the `--template-annotated/-ta` option

The following is an example of a JSON file that creates a single-node Oracle KVM Database System on Oracle Database Appliance bare metal platform with the `--template-annotated/-ta` option. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system" : {
        "name" : "test_dbsystem",                       ---> The name for the
DB System
        "shape" : "dbs2",                               ---> The shape for the
DB System
        "customMemorySize" : "24G",                     ---> The memory size
for the DB System
        "timeZone" : "America/Los_Angeles",             ---> The timezone for
the DB System
        "diskGroup" : "DATA",                           ---> The ASM disk
group to be used for the DB System volumes. Allowed options: <DATA>, <RECO>,
<REDO>
        "cpuPoolName" : "shared_dbsystem_cpupool",      ---> The CPU Pool for
the DB System
        "enableRoleSeparation" : true,                  ---> Whether or not to
enable Role Separation. If true, need to specify 6 groups and two users.
'groupName' and 'userName' can be customized. If false, need to specify 2
groups and one user. 'groupName' and 'userName' can be customized. Allowed
options: <true>, <false>
        "customRoleSeparation" : {
            "groups" : [
                {
                    "name" : "oinstall",                ---> Name of the group
to create, must be unique
                    "id" : 1001,                        ---> ID of the group
to create, must be unique
                    "role" : "oinstall",                ---> Role of the
group
                }
            ],
            "users" : [
                {
                    "name" : "grid",                    ---> Name of the user
to create, must be unique
                    "id" : 1000,                        ---> ID of the user to
create, must be unique
                    "role" : "gridUser",                ---> Role of the
user
                },
                {
                    "name" : "oracle",                  ---> Name of the user
to create, must be unique
                    "id" : 1001,                        ---> ID of the user to
```

```
create, must be unique
                    "role" : "oracleUser",             ---> Role of the
user
                }
            ]
        }
    },
    "database" : {
        "name" : "dbtest",                             ---> Name for the DB
System database
        "uniqueName" : "dbtest",                       ---> The unique name
for the DB System database
        "domainName" : "example.com",                  ---> The domain name
that the DB System database will have
        "version" : "19.25.0.0.241015",            ---> The version for the
DB System database. Execute 'odacli describe-dbsystem-image' to get the
supported version
        "edition" : "EE",                              ---> Enter 'EE' for
Enterprise Edition, or 'SE' for Standard Edition
        "type" : "SI",                                 ---> Enter a database
deployment type. Allowed options: <RAC>, <RACOne>, <SI> (single instance
database)
        "dbClass" : "OLTP",                            ---> The class for the
DB System database. Standard Edition only supports OLTP. Allowed options:
<OLT>, <DSS>, <IMDB>
        "shape" : "odb2",                              ---> The DB System
database shape, it must be same as the DB System shape
        "role" : "PRIMARY",                            ---> The role for the
DB System database. Allowed options: <PRIMARY>, <STANDBY>
        "targetNodeNumber" : "0",                      ---> Use '0' for Node0
and '1' for Node1. For RAC or RACOne use null. The default is '0' for
Node0
        "enableDbConsole" : false,                     ---> Whether or not to
create dbconsole or EM express. By default, dbconsole is disabled. Allowed
options: <true>, <false>
        "enableFlashStorage" : false,                  ---> Whether or not to
enable the flash storage for the DB System database. Allowed options: <true>,
<false>
        "enableFlashCache" : false,                    ---> Whether or not to
enable the flash cache for the DB System database. Allowed options: <true>,
<false>
        "enableUnifiedAuditing" : true,                ---> Whether or not to
enable unified auditing for the DB System database. Allowed options: <true>,
<false>
        "enableEEHA" : true,                           ---> Whether or not to
enable HA for EE SI database. enableEEHA is only valid when the edition is
EE. Allowed options: <true>, <false>
        "enableSEHA" : false,                          ---> Whether or not to
enable HA for SE SI database. enableSEHA is only valid when the edition is
SE. Allowed options: <true>, <false>
        "redundancy" : "MIRROR",                       ---> If diskgroup
redundancy is FLEX, then database redundancy must be set to one of 'HIGH' or
'MIRROR', otherwise, can be null
        "characterSet" : {
            "characterSet" : "AL32UTF8",               ---> The character set
for the DB System database
```

```
            "nlsCharacterset" : "AL16UTF16",           ---> The NLS character
set for the DB System database
            "dbTerritory" : "AMERICA",                 ---> The territory for
the DB System database
            "dbLanguage" : "ENGLISH",                  ---> The language for
the DB System database
        },
        "enableTDE" : false,                           ---> Whether or not to
enable TDE. Allowed options: <true>, <false>
        "isCdb" : true,                                ---> Enter 'true' if
this database is container DB. For 21c and above, cdb must be true. Enter
'false' if this is non-cdb. Allowed options: <true>, <false>
        "pdbName" : "pdb1",                            ---> Enter pdbName if
isCdb parameter is 'true', use 'null' if isCdb is 'false'
        "pdbAdminUser" : "pdbadmin",                   ---> Enter PDB admin
user name, use 'null' if isCdb parameter is 'false'
    },
    "network" : {
        "domainName" : "example.com",                  ---> The domain name
for the DB System network
        "ntpServers" : [
            "xxx.xxx.xxx.xxx"
        ],
        "dnsServers" : [
            "xxx.xxx.xxx.xxx"
        ],
        "nodes" : [
            {
                "name" : "node1",                      ---> The name used to
configure the hostname
                "ipAddress" : "xxx.xxx.xxx.xxx",       ---> The IP address
for this node
                "netmask" : "xxx.xxx.xxx.xxx",         ---> The mask of the
network for this node
                "gateway" : "xxx.xxx.xxx.xxx",         ---> The gateway
address for this node
                "number" : 0,                          ---> 0 (Use 0 for the
first node of the DB System instance)
            }
        ],
        "publicVNetwork" : "vnet1",                    ---> The name of the
public virtual network. The ipAddress, vipAddress and scanIps must be on this
publicVNetwork
    },
    "grid" : {
        "language" : "en",                             ---> The language used
for GI (Grid Infrastructure) installation
        "enableAFD" : true,                            ---> Whether or not to
enable ASM Filter Driver, enabled by default. Allowed options: <true>,
<false>
    }
}
```

Example JSON File to Create a High-Availability DB System

Follow the JSON file example to create a JSON file to deploy a High-Availability DB system, with role separation, with the command `odacli create-dbsystem`.

Use the example JSON file to create a file for your environment.

Example 15-5 JSON File to Create a High-Availability DB system without Role Separation

The following is an example of a JSON file that creates a High-Availability DB system on Oracle Database Appliance bare metal platform, without role separation. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "clusterName": "cl-test",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
                    "id": 1005,
                    "role": "asmoper"
                },
                {
                    "name": "asmdba",
                    "id": 1006,
```

```
                        "role": "asmdba"
                    }
                ],
                "users": [
                    {
                        "name": "grid",
                        "id": 1000,
                        "role": "gridUser"
                    },
                    {
                        "name": "oracle",
                        "id": 1001,
                        "role": "oracleUser"
                    }
                ]
            }
        },
        "database": {
            "name": "dbtest",
            "uniqueName": "dbtest",
            "domainName": "test_domain",
            "version": "19.25.0.0.241015",
            "edition": "EE",
            "type": "SI",
            "dbClass": "OLTP",
            "shape": "odb2",
            "role": "PRIMARY",
            "targetNodeNumber": "0",
            "enableDbConsole": false,version
            "enableUnifiedAuditing": true,
            "enableEEHA": true,
            "redundancy": "MIRROR", <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
                "dbLanguage": "AMERICAN"
            },
             "enableTDE": false,
             "isCdb": true,
                "pdbName": "pdb1",
                "pdbAdminUser": "pdbadmin"
        },
        "network": {
            "domainName": "test_domain",
            "ntpServers": [],
            "dnsServers": [
                "xx.xxx.xx.xxx"
            ],
            "nodes": [
                {
                    "name": "node1",
                    "ipAddress": "xx.xx.xxx.xx",
                    "netmask": "xxx.xxx.xxx.x",
```

```
                "gateway": "xx.xx.xxx.x",
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xx.xx.xxx.xx"
            },
            {
                "name": "node2",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xx.xx.xxx.xx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "test-scan",
        "scanIps": [
            "xx.xx.xxx.xx",
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en",
         "enableAFD": true
    }
}
```

Example 15-6 JSON File to Create a High-Availability DB system with Role Separation

The following is an example of a JSON file that creates a High-Availability DB system on Oracle Database Appliance bare metal platform. The example uses role separation. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_system",
        "shape": "dbs2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
```

```
                    {
                        "name": "dba",
                        "id": 1003,
                        "role": "dba"
                    },
                    {
                        "name": "asmadmin",
                        "id": 1004,
                        "role": "asmadmin"
                    },
                    {
                        "name": "asmoper",
                        "id": 1005,
                        "role": "asmoper"
                    },
                    {
                        "name": "asmdba",
                        "id": 1006,
                        "role": "asmdba"
                    }
                ],
                "users": [
                    {
                        "name": "grid",
                        "id": 1000,
                        "role": "gridUser"
                    },
                    {
                        "name": "oracle",
                        "id": 1001,
                        "role": "oracleUser"
                    }
                ]
            }
        },
        "database": {
            "name": "dbtest",
            "uniqueName": "dbtest",
            "domainName": "test_domain",
            "version": "23.5.0.24.07",
            "edition": "EE",
            "type": "SI",
            "dbClass": "OLTP",
            "shape": "odb2",
            "role": "PRIMARY",
            "targetNodeNumber": "0",
            "enableDbConsole": false,
            "enableUnifiedAuditing": true,
            "enableEEHA": true,
            "redundancy": null, <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
```

```
            "dbLanguage": "AMERICAN"
        },
        "enableTDE": false,
        "isCdb": true,
            "pdbName": "pdb1",
            "pdbAdminUser": "pdbadmin"
    },
    "network": {
        "domainName": "test_domain",
        "ntpServers": [],
        "dnsServers": [
            "xx.xxx.xx.xxx"
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xx.xx.xxx.xx"
            },
            {
                "name": "node2",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xx.xx.xxx.xx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "test-scan",
        "scanIps": [
            "xx.xx.xxx.xx",
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en",
         "enableAFD": true
    }
}
```

Example 15-7 JSON File to Create a High-Availability DB system with Standard Edition High-Availability Enabled

The following is an example of a JSON file that creates a High-Availability DB system on Oracle Database Appliance bare metal platform. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
```

```
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
                    "id": 1005,
                    "role": "asmoper"
                },
                {
                    "name": "asmdba",
                    "id": 1006,
                    "role": "asmdba"
                }
            ],
            "users": [
                {
                    "name": "grid",
                    "id": 1000,
                    "role": "gridUser"
                },
                {
                    "name": "oracle",
                    "id": 1001,
                    "role": "oracleUser"
                }
            ]
        }
    },
    "database": {
        "name": "dbtest",
```

```
        "uniqueName": "dbtest",
        "domainName": "test_domain",
        "version": "19.25.0.0.241015",
        "edition": "SE",
        "type": "SI",
        "dbClass": "OLTP",
        "shape": "odb2",
        "role": "PRIMARY",
        "targetNodeNumber": "0",
        "enableDbConsole": false,
        "enableUnifiedAuditing": true,
        "enableSEHA": true,
        "redundancy": "MIRROR", <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
        "characterSet": {
            "characterSet": "AL32UTF8",
            "nlsCharacterset": "AL16UTF16",
            "dbTerritory": "AMERICA",
            "dbLanguage": "AMERICAN"
        },
         "enableTDE": false,
         "isCdb": true,
            "pdbName": "pdb1",
            "pdbAdminUser": "pdbadmin"
    },
    "network": {
        "domainName": "test_domain",
        "ntpServers": [],
        "dnsServers": [
            "xx.xxx.xx.xxx"
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xx.xx.xxx.xx"
            },
            {
                "name": "node2",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xx.xx.xxx.xx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "test-scan",
        "scanIps": [
            "xx.xx.xxx.xx",
```

```
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en"
        "enableAFD": true
    }
}
```

Example 15-8 JSON File to Create a High-Availability DB system with Oracle RAC Database

The following is an example of a JSON file that creates a High-Availability DB system on Oracle Database Appliance bare metal platform. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
                    "id": 1005,
                    "role": "asmoper"
                },
                {
                    "name": "asmdba",
                    "id": 1006,
```

```
                        "role": "asmdba"
                    }
                ],
                "users": [
                    {
                        "name": "grid",
                        "id": 1000,
                        "role": "gridUser"
                    },
                    {
                        "name": "oracle",
                        "id": 1001,
                        "role": "oracleUser"
                    }
                ]
            }
        },
        "database": {
            "name": "dbtest",
            "uniqueName": "dbtest",
            "domainName": "test_domain",
            "version": "19.25.0.0.241015",
            "edition": "EE",
            "type": "RAC",
            "dbClass": "OLTP",
            "shape": "odb2",
            "role": "PRIMARY",
            "targetNodeNumber": "0",
            "enableDbConsole": false,
            "enableUnifiedAuditing": true,
            "redundancy": "MIRROR", <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
                "dbLanguage": "AMERICAN"
            },
             "enableTDE": false,
             "isCdb": true,
                "pdbName": "pdb1",
                "pdbAdminUser": "pdbadmin"
        },
        "network": {
            "domainName": "test_domain",
            "ntpServers": [],
            "dnsServers": [
                "xx.xxx.xx.xxx"
            ],
            "nodes": [
                {
                    "name": "node1",
                    "ipAddress": "xx.xx.xxx.xx",
                    "netmask": "xxx.xxx.xxx.x",
                    "gateway": "xx.xx.xxx.x",
```

```
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xx.xx.xxx.xx"
            },
            {
                "name": "node2",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xx.xx.xxx.xx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "test-scan",
        "scanIps": [
            "xx.xx.xxx.xx",
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en"
        "enableAFD": true
    }
}
```

Example 15-9 JSON File to Create a High-Availability DB system with the `--template/-t` option

The following is an example of a JSON file that creates a High-Availability DB system on Oracle Database Appliance bare metal platform. This template sample is generated when you run the `odacli create-dbsystem` command with the `--template/-t` option. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_dbsystem",
        "shape": "dbs2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                }
            ],
            "users": [
                {
                    "name": "grid",
```

```
                "id": 1000,
                "role": "gridUser"
            },
            {
                "name": "oracle",
                "id": 1001,
                "role": "oracleUser"
            }
        ]
    }
},
"database": {
    "name": "dbtest",
    "uniqueName": "dbtest",
    "domainName": "example.com",
    "version": "19.25.0.0.241015",
    "edition": "EE",
    "type": "SI",
    "dbClass": "OLTP",
    "shape": "odb2",
    "role": "PRIMARY",
    "targetNodeNumber": "0",
    "enableDbConsole": false,
    "enableFlashStorage": false,
    "enableFlashCache": false,
    "enableUnifiedAuditing": true,
    "enableEEHA": true,
    "enableSEHA": false,
    "redundancy": "MIRROR",
    "characterSet": {
        "characterSet": "AL32UTF8",
        "nlsCharacterset": "AL16UTF16",
        "dbTerritory": "AMERICA",
        "dbLanguage": "AMERICAN"
    },
    "enableTDE": false,
    "isCdb": true,
    "pdbName": "pdb1",
    "pdbAdminUser": "pdbadmin"
},
"network": {
    "domainName": "example.com",
    "ntpServers": [
        "xxx.xxx.xxx.xxx"
    ],
    "dnsServers": [
        "xxx.xxx.xxx.xxx"
    ],
    "nodes": [
        {
            "name": "node1",
            "ipAddress": "xxx.xxx.xxx.xxx",
            "netmask": "xxx.xxx.xxx.xxx",
            "gateway": "xxx.xxx.xxx.xxx",
            "number": 0,
            "vipName": "node1-vip",
```

```
                "vipAddress": "xxx.xxx.xxx.xxx"
            },
            {
                "name": "node2",
                "ipAddress": "xxx.xxx.xxx.xxx",
                "netmask": "xxx.xxx.xxx.xxx",
                "gateway": "xxx.xxx.xxx.xxx",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xxx.xxx.xxx.xxx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "dbtest-scan",
        "scanIps": [
            "xxx.xxx.xxx.xxx",
            "xxx.xxx.xxx.xxx"
        ]
    },
    "grid": {
        "language": "en",
        "enableAFD": true
    }
}
```

Example 15-10 JSON File to Create a High-Availability DB system with the `--template/-t` option

The following is an example of a JSON file that creates a High-Availability DB system on Oracle Database Appliance bare metal platform. This template sample is generated when you run the `odacli create-dbsystem` command with the `--template/t` option. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_dbsystem",
        "shape": "dbs2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                }
            ],
            "users": [
                {
                    "name": "grid",
                    "id": 1000,
                    "role": "gridUser"
```

```
                },
                {
                    "name": "oracle",
                    "id": 1001,
                    "role": "oracleUser"
                }
            ]
        }
    },
    "database": {
        "name": "dbtest",
        "uniqueName": "dbtest",
        "domainName": "example.com",
        "version": "19.25.0.0.241015",
        "edition": "EE",
        "type": "SI",
        "dbClass": "OLTP",
        "shape": "odb2",
        "role": "PRIMARY",
        "targetNodeNumber": "0",
        "enableDbConsole": false,
        "enableFlashStorage": false,
        "enableFlashCache": false,
        "enableUnifiedAuditing": true,
        "enableEEHA": true,
        "enableSEHA": false,
        "redundancy": "MIRROR",
        "characterSet": {
            "characterSet": "AL32UTF8",
            "nlsCharacterset": "AL16UTF16",
            "dbTerritory": "AMERICA",
            "dbLanguage": "AMERICAN"
        },
        "enableTDE": false,
        "isCdb": true,
        "pdbName": "pdb1",
        "pdbAdminUser": "pdbadmin"
    },
    "network": {
        "domainName": "example.com",
        "ntpServers": [
            "xxx.xxx.xxx.xxx"
        ],
        "dnsServers": [
            "xxx.xxx.xxx.xxx"
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xxx.xxx.xxx.xxx",
                "netmask": "xxx.xxx.xxx.xxx",
                "gateway": "xxx.xxx.xxx.xxx",
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xxx.xxx.xxx.xxx"
            },
```

```
            {
                "name": "node2",
                "ipAddress": "xxx.xxx.xxx.xxx",
                "netmask": "xxx.xxx.xxx.xxx",
                "gateway": "xxx.xxx.xxx.xxx",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xxx.xxx.xxx.xxx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "dbtest-scan",
        "scanIps": [
            "xxx.xxx.xxx.xxx",
            "xxx.xxx.xxx.xxx"
        ]
    },
    "grid": {
        "language": "en",
        "enableAFD": true
    }
}
```

Example 15-11 JSON File to Create a High-Availability DB system with the `--template-annotated/-ta` option

The following is an example of a JSON file that creates a High-Availability DB system on Oracle Database Appliance bare metal platform. This template sample is generated when you run the `odacli create-dbsystem` command with the `--template-annotated/-ta` option. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system" : {
        "name" : "test_dbsystem",                       ---> The name for the
DB System
        "shape" : "dbs2",                               ---> The shape for the
DB System
        "customMemorySize" : "24G",                     ---> The memory size
for the DB System
        "timeZone" : "America/Los_Angeles",             ---> The timezone for
the DB System
        "diskGroup" : "DATA",                           ---> The ASM disk
group to be used for the DB System volumes. Allowed options: <DATA>, <RECO>,
<REDO>
        "cpuPoolName" : "shared_dbsystem_cpupool",      ---> The CPU Pool for
the DB System
        "useReservedCores" : false,                     ---> Whether or not to
use reserved CPU cores. Allowed options: <true>, <false>
        "enableRoleSeparation" : true,                  ---> Whether or not to
enable Role Separation. If true, need to specify 6 groups and two users.
'groupName' and 'userName' can be customized. If false, need to specify 2
groups and one user. 'groupName' and 'userName' can be customized. Allowed
options: <true>, <false>
        "customRoleSeparation" : {
            "groups" : [
```

```
                {
                    "name" : "oinstall",                ---> Name of the group
to create, must be unique
                    "id" : 1001,                        ---> ID of the group
to create, must be unique
                    "role" : "oinstall",                ---> Role of the
group
                }
            ],
            "users" : [
                {
                    "name" : "grid",                    ---> Name of the user
to create, must be unique
                    "id" : 1000,                        ---> ID of the user to
create, must be unique
                    "role" : "gridUser",                ---> Role of the
user
                },
                {
                    "name" : "oracle",                  ---> Name of the user
to create, must be unique
                    "id" : 1001,                        ---> ID of the user to
create, must be unique
                    "role" : "oracleUser",              ---> Role of the
user
                }
            ]
        }
    },
    "database" : {
        "name" : "dbtest",                              ---> Name for the DB
System database
        "uniqueName" : "dbtest",                        ---> The unique name
for the DB System database
        "domainName" : "example.com",                   ---> The domain name
that the DB System database will have
        "version" : "19.25.0.0.241015",                 ---> The version for
the DB System database. Execute 'odacli describe-dbsystem-image' to get the
supported version
        "edition" : "EE",                               ---> Enter 'EE' for
Enterprise Edition, or 'SE' for Standard Edition
        "type" : "SI",                                  ---> Enter a database
deployment type. Allowed options: <RAC>, <RACOne>, <SI> (single instance
database)
        "dbClass" : "OLTP",                             ---> The class for the
DB System database. Standard Edition only supports OLTP. Allowed options:
<OLT>, <DSS>, <IMDB>
        "shape" : "odb2",                               ---> The DB System
database shape, it must be same as the DB System shape
        "role" : "PRIMARY",                             ---> The role for the
DB System database. Allowed options: <PRIMARY>, <STANDBY>
        "dbStorage" : "ASM",                            ---> Storage type of
the DB System database. The default value is ASM. Allowed options: <ASM>,
<PMEM>
        "level0BackupDay" : "wednesday",                ---> Backup day of the
DB System database. Allowed options: <monday>, <tuesday>, <wednesday>,
```

```
<thursday>, <friday>, <saturday>, <sunday>
        "targetNodeNumber" : "0",                       ---> Use '0' for Node0
and '1' for Node1. For RAC or RACOne use null. The default is '0' for
Node0
        "enableDbConsole" : false,                      ---> Whether or not to
create dbconsole or EM express. By default, dbconsole is disabled. Allowed
options: <true>, <false>
        "enableFlashStorage" : false,                   ---> Whether or not to
enable the flash storage for the DB System database. Allowed options: <true>,
<false>
        "enableFlashCache" : false,                     ---> Whether or not to
enable the flash cache for the DB System database. Allowed options: <true>,
<false>
        "enableUnifiedAuditing" : true,                 ---> Whether or not to
enable unified auditing for the DB System database. Allowed options: <true>,
<false>
        "enableEEHA" : true,                            ---> Whether or not to
enable HA for EE SI database. enableEEHA is only valid when the edition is
EE. Allowed options: <true>, <false>
        "enableSEHA" : false,                           ---> Whether or not to
enable HA for SE SI database. enableSEHA is only valid when the edition is
SE. Allowed options: <true>, <false>
        "redundancy" : "MIRROR",                        ---> If diskgroup
redundancy is FLEX, then database redundancy must be set to one of 'HIGH' or
'MIRROR', otherwise, can be null
        "characterSet" : {
            "characterSet" : "AL32UTF8",                ---> The character set
for the DB System database
            "nlsCharacterset" : "AL16UTF16",            ---> The NLS character
set for the DB System database
            "dbTerritory" : "AMERICA",                  ---> The territory for
the DB System database
            "dbLanguage" : "AMERICAN",                  ---> The language for
the DB System database
        },
        "enableTDE" : false,                            ---> Whether or not to
enable TDE. Allowed options: <true>, <false>
        "isCdb" : true,                                 ---> Enter 'true' if
this database is container DB. For 21c and above, cdb must be true. Enter
'false' if this is non-cdb. Allowed options: <true>, <false>
        "pdbName" : "pdb1",                             ---> Enter pdbName if
isCdb parameter is 'true', use 'null' if isCdb is 'false'
        "pdbAdminUser" : "pdbadmin",                    ---> Enter PDB admin
user name, use 'null' if isCdb parameter is 'false'
    },
    "network" : {
        "domainName" : "example.com",                   ---> The domain name
for the DB System network
        "ntpServers" : [
            "xxx.xxx.xxx.xxx"
        ],
        "dnsServers" : [
            "xxx.xxx.xxx.xxx"
        ],
        "nodes" : [
            {
```

```
                "name" : "node1",                       ---> The name used to
configure the hostname
                "ipAddress" : "xxx.xxx.xxx.xxx",        ---> The IP address
for this node
                "netmask" : "xxx.xxx.xxx.xxx",          ---> The mask of the
network for this node
                "gateway" : "xxx.xxx.xxx.xxx",          ---> The gateway
address for this node
                "number" : 0,                           ---> 0 (Use 0 for the
first node of the DB System instance)
                "vipName" : "node1-vip",                ---> The VIP name for
this first node, only for HA case
                "vipAddress" : "xxx.xxx.xxx.xxx",       ---> The VIP address
for this first node, only for HA case
            },
            {
                "name" : "node2",                       ---> The name used to
configure the hostname
                "ipAddress" : "xxx.xxx.xxx.xxx",        ---> The IP address
for this node
                "netmask" : "xxx.xxx.xxx.xxx",          ---> The mask of the
network for this node
                "gateway" : "xxx.xxx.xxx.xxx",          ---> The gateway
address for this node
                "number" : 1,                           ---> 1 (Use 1 for the
second node of the DB System instance)
                "vipName" : "node2-vip",                ---> The VIP name for
this second node, only for HA case
                "vipAddress" : "xxx.xxx.xxx.xxx",       ---> The VIP address
for this second node, only for HA case
            }
        ],
        "publicVNetwork" : "vnet1",                     ---> The name of the
public virtual network. The ipAddress, vipAddress and scanIps must be on this
publicVNetwork
        "scanName" : "dbtest-scan",                     ---> The scan name of
the DB System. It should resolve to the IPs defined in the scanIps. Only for
HA case
        "scanIps" : [
            "xxx.xxx.xxx.xxx",
            "xxx.xxx.xxx.xxx"
        ]
    },
    "grid" : {
        "language" : "en",                              ---> The language used
for GI (Grid Infrastructure) installation
        "enableAFD" : true,                             ---> Whether or not to
enable ASM Filter Driver, enabled by default. Allowed options: <true>,
<false>
    }
}
```

16

Oracle Database Appliance Command-Line Interface

The command-line interface has different classes of tools to manage Oracle Database Appliance.

- About Oracle Database Appliance Command-Line Interface
  Three classes of tools are available to perform deployment, lifecycle management, and system administration on Oracle Database Appliance.
- Managing ODACLI Privileges and Security with SUDO
  Oracle Appliance Manager command-line utility requires `root` system privileges for most administration actions. You may want to use SUDO as part of your system auditing and security policy.
- ODACLI Command Changes with Multi-User Access on Oracle Database Appliance
  Understand the ODACLI command changes when you enable multi-user access.
- odacli Network Commands
  Use the `odacli network` commands to list and describe network interfaces.
- odacli Multi-User Access Commands
  Use `odacli` commands to enable multi-user access for the appliance.
- odacli Apply Patch and Update Commands
  Use the commands `odacli update` and `apply patch` to apply patches and update the appliance.
- odacli Appliance Commands
  Use the `odacli appliance` commands to perform lifecycle activities for the appliance.
- odacli Backup and Recovery Commands
  Use the odacli backup and recover commands to backup to and restore from Oracle Cloud Infrastructure Object Storage or disk or NFS.
- odacli Oracle Data Guard Commands
  Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.
- odacli CPU Core Commands
  Use the CPU Core commands to enable CPU cores and display current and historical CPU core configurations.
- odacli CPU Pool Commands for Bare Metal and KVM
  Use odacli commands to manage resource allocations on Oracle Database Appliance for bare metal and KVM deployments.
- odacli Database Commands
  Use the `odacli database` commands to perform database lifecycle operations.
- odacli DBHome Commands
  Use the `odacli DBHome` commands to manage database Home operations.
- odacli Database Storage Commands
  Use the Database Storage commands to list, describe, create, and delete Oracle database storage.

- odacli Diagnostics Collection Commands
  Use the diagnostics collection commands to display diagnostics details.
- odacli Job Commands
  Use the `odacli list-jobs` and `odacli describe-job` commands to display job details.
- odacli Oracle Key Vault Server Configuration Commands
  Use `odacli` commands to manage Oracle Key Vault server configuration for databases on Oracle Database Appliance.
- Log Commands
  Use the `odacli` log commands to specify the options to collect and delete logs.
- odacli Oracle Auto Service Request Commands
  Use the Oracle Auto Service Request (Oracle ASR) commands to configure, update, test, and delete Oracle ASR on the system.
- Adaptive Classification and Redaction (ACR) Commands
  Understand Adaptive Classification and Redaction (ACR) commands.
- odacli OS Commands
  Use the `odacli OS` commands to list and update operating system (OS) parameters.
- odaadmcli Hardware Monitoring Commands
  Use the `hardware monitoring` commands to display hardware configurations.
- Storage Commands
  Understand the commands to perform storage operations and diagnostics.

About Oracle Database Appliance Command-Line Interface

Three classes of tools are available to perform deployment, lifecycle management, and system administration on Oracle Database Appliance.

The command-line interface (CLI) is an alternative to Oracle Database Appliance Browser User Interface for some appliance configuration and patching tasks, database management, and job activity.

Oracle Database Appliance uses a role-based command-line interface. Use the `odacli` commands to perform lifecycle management tasks and the `odaadmcli` commands to perform storage and hardware monitoring maintenance. Many tasks related to managing Oracle Databases are also required with databases on Oracle Database Appliance. Tasks common to Oracle Database generally are described in the Oracle Database documentation library. However, to simplify tasks, use the Oracle Database Appliance command-line interface. The `odacli` and `odaadmcli`utilities combine the capabilities of the `SYS` database administrator role and the operating system Superuser (`root` user). Always perform administrative tasks using the Oracle Database Appliance Browser User Interface or CLI utilities.

The following classes of commands are available:

- Deployment and Configuration: Use the deployment and configuration commands as part of the initial deployment of the appliance and to configure CPU cores.
- Lifecycle management: Use `odacli` commands to perform database and system administration tasks for the appliance. The `odacli` commands are the primary interface to perform life cycle activities for the appliance.
- Administration: Use `odaadmcli` commands to perform hardware administration tasks for the appliance. The `odaadmcli` commands are the interface for infrequent administration activities needed to manage appliance hardware components, storage, and VLAN.

You can perform the following deployment and configuration tasks:

- Configure the first network in the appliance
- Unzip and copy the Oracle Database Appliance software to the correct locations
- Set the number of CPU Cores for the system

You can perform the following appliance lifecycle tasks with `odacli` commands:

- Create and describe the appliance
- Create, list, describe, and delete databases
- Create, list, describe, and delete Oracle Database Homes
- Create, list, and describe the networks
- List and describe the jobs

You can perform the following appliance administration tasks with `odaadmcli` commands:

- Show storage, disks, disk groups, and controllers
- Display storage diagnostics for disks and NVM Express (NVMes)
- Show server, memory, processor, power, cooling, and network details

Depending on your version of Oracle Appliance Manager and your hardware, some of the `odacli` commands may not be available to you. To see which commands are supported on your version of Oracle Appliance Manager and your hardware, run the `odacli` help command: `odacli -h`.

Command-Line Interface Syntax

The command-line interface commands and parameters are case-sensitive.

An `odacli` or `odaadmcli`command uses the following command syntax:

```
odacli command [options]
```

- *command* is an action you want to perform on the appliance. For example: `odacli list-networks` or `odacli describe-jobs`.
- *options* are optional parts of the `odacli` command. Options can consist of one or more options that extend the use of the `odacli` command carried out on an object. Options include additional information about the action that you want to perform on the object. Option names are preceded with a dash. Some options require the name of an object for the command to perform the action that you want to carry out. If an option is preceded with an * (asterisk), this information is required to submit the command. When appending `-j` to the odacli command, the output is returned in JSON format. The help option (`-h`) is an option that is available with almost all commands. When you include the `-h` option, you can obtain additional information about the command that you want to perform.

Oracle Database Appliance Manager Command-Line Interface Help

Run the `-h` command to see the usage information for all commands available for your Oracle Database Appliance. For example:

```
odacli -h
```

Run `odacli command -h` to see detailed help about a specific command. For example, to see detailed help for the `odacli describe-dbhome` command, use the following:

```
odacli describe-dbhome -h
```

Managing ODACLI Privileges and Security with SUDO

Oracle Appliance Manager command-line utility requires `root` system privileges for most administration actions. You may want to use SUDO as part of your system auditing and security policy.

For most tasks, Oracle recommends that you log in as `root` to use the Oracle Appliance Manager command-line interface on Oracle Database Appliance. If you are not logged in as `root`, then you cannot carry out most actions on the appliance. For example, if you are not logged in as `root`, then you can view storage information, but you cannot modify the storage.

Allowing Root User Access Using SUDO

In environments where system administration is handled by a different group than database administration, or where security is a significant concern, you may want to limit access to the `root` user account and password. SUDO enables system administrators to grant certain users (or groups of users) the ability to run commands as `root`, while logging all commands and arguments as part of your security and compliance protocol.

A SUDO security policy is configured by using the file `/etc/sudoers`. Within the `sudoers` file, you can configure groups of users and sets of commands to simplify and audit server administration with SUDO commands.

> **Caution:**
>
> Configuring SUDO to allow a user to perform any operation is equivalent to giving that user `root` privileges. Consider carefully if this is appropriate for your security needs.

Example 16-1 SUDO Example 1: Allow a User to Perform Any ODACLI Operation

This example shows how to configure SUDO to enable a user to perform any ODACLI operation. You do this by adding lines to the commands section in the `/etc/sudoers` file:

```
## The commands section may have other options added to it.
##
Cmnd_Alias ODACLI_CMDS=/opt/oracle/dcs/bin/odacli *
jdoe ALL = ODACLI_CMDS
```

In this example, the user name is `jdoe`. The file parameter setting `ALL= ODACLI_CMDS` grants the user `jdoe` permission to run all `odacli` commands that are defined by the command alias `ODACLI_CMDS`. After configuration, you can copy one `sudoers` file to multiple hosts. You can also create different rules on each host.

> **Note:**
>
> Before database creation, you must set up user equivalency with SSH for the root user on each server. If you do not set up user equivalency and configure SSH on each server, then you are prompted to provide the root password for each server during database creation.

After you configure the `sudoer` file with the user, the user `jdoe` can run the set of `odacli` commands configured with the command alias `ODACLI_CMDS`. For example:

```
odacli create-database --dbname newdb
Enter new 'SYS, SYSTEM and PDB Admin user' password:
Confirm 'SYS, SYSTEM and PDB Admin user' password:

Job details
----------------------------------------------------------------
                     ID: 1bc31577-f910-4d3f-b6ff-8e3fccd30141
            Description: Database service creation with db name: newdb
                 Status: Created
                Created: November 30, 2018 9:23:57 AM PST
                Message:

Task Name Start Time End Time Status
----------------------------------------------------------
```

Example 16-2 SUDO Example 2: Allow a User to Perform Only Selected ODACLI Operations

To configure SUDO to allow a user to perform only selected ODACLI operations, add lines to the commands section in the `/etc/sudoers` file as follows:

```
## DCS commands for oracle user
Cmnd_Alias DCSCMDS = /opt/oracle/dcs/bin/odacli describe-appliance
oracle  ALL=       DCSCMDS


$ sudo /opt/oracle/dcs/bin/odacli describe-appliance

Appliance Information
----------------------------------------------------------------
                     ID: a977bb04-6cf0-4c07-8e0c-91a8c7e7ebb8
               Platform: OdaliteL
        Data Disk Count: 6
         CPU Core Count: 20
                Created: October 24, 2017 6:51:52 AM HDT

System Information
----------------------------------------------------------------
                   Name: system1001
            Domain Name: example.com
              Time Zone: America/Adak
             DB Edition: EE
            DNS Servers: 10.200.76.198 10.200.76.199 192.0.2.254
```

```
            NTP Servers: 10.200.0.1 10.200.0.2

Disk Group Information
------------------------------------------------------------------
DG Name                   Redundancy                Percentage
------------------------ ------------------------- ------------
Data                      Normal                    90
Reco                      Normal                    10
```

In this example, the user `jdoe2` tries to run the `sudo odacli list-databases` command, which is not part of the set of commands that is configured for that user. SUDO prevents `jdoe2` from running the command.

```
[jdoe2@servernode1 ~]$ sudo /opt/oracle/dcs/bin/odacli list-databases

Sorry, user jdoe2 is not allowed to execute '/opt/oracle/dcs/bin/odacli list-
databases' as root on servernode1.
```

ODACLI Command Changes with Multi-User Access on Oracle Database Appliance

Understand the ODACLI command changes when you enable multi-user access.

About the `odacli-view` command

In a multi-user access-enabled environment, to run automated scripts, that is, to generate a system report based on the output of `odacli` commands, you may need to provide the user credentials when prompted by the system. In such cases, the automation of scripts may fail. In such cases, use the `odacli-view` command instead of the `odacli` command. The `odacli-view` command allows you to run read-only odacli commands as root user without providing user credentials. Read-only `odacli` commands include commands to list, describe, get, read, or ping.

However, the security of the system is not compromised because the user credentials of a securely-generated internal user are used for this purpose. The `odacli-view` command can be used in Oracle Database Appliance deployments with or without multi-user access. In an Oracle Database Appliance environment where multi-user access is not enabled, `odacli-view` has the same functionality as `odacli`. In a multi-user access-enabled environment, `odacli-view` provides the information about all the resources in a system. For example, `odacli-view list-databases` command lists all the databases in the system.

Usage:

```
/opt/oracle/dcs/bin/odacli-view command_name
```

For example:

```
/opt/oracle/dcs/bin/odacli-view list-dbhomes
/opt/oracle/dcs/bin/odacli-view list-databases
/opt/oracle/dcs/bin/odacli-view list-users
```

ODACLI Command Usage and Entitlement Requirements When Multi-User Access is Enabled

> **Note:**
>
> If you have enabled multi-user access, then you can run all supported ODACLI commands on KVM and DB systems, after logging in as `odaadmin`.

> **Note:**
>
> The authentication token support for ODACLI session management is linked to a multi-user access user account. Since `root` is an operating system administrative user and not a multi-user access user, auth token based session management system is not supported when a user logs in as `root`. Therefore, you must provide an Oracle Database Appliance account user name and password to run any ODACLI command.

The following table describes the ODACLI commands that have changes for multi-user access and the entitlements required to run the commands.

ODACLI Command	Roles Required	Changes in command option
odacli configure-firstnet	ODA-ADMINISTRATOR	No change
odacli create-network	ODA-ADMINISTRATOR	No change
odacli delete-network	ODA-ADMINISTRATOR	No change
odacli delete-networkinterface	ODA-ADMINISTRATOR	No change
odacli describe-network	ODA-ADMINISTRATOR, ODA-DB	No change
odacli describe-networkinterface	ODA-ADMINISTRATOR, ODA-DB	No change
odacli list-networks	ODA-ADMINISTRATOR, ODA-DB	No change
odacli list-networkinterfaces	ODA-ADMINISTRATOR, ODA-DB	No change
odacli modify-network	ODA-ADMINISTRATOR	No change
odacli cleanup-patchrepo	ODA-ADMINISTRATOR, ODA-DB	No change
odacli create-prepatchreport	ODA-ADMINISTRATOR, ODA-DB	No change
odacli describe-prepatchreport	ODA-ADMINISTRATOR, ODA-DB	No change
odacli list-prepatchreports	ODA-ADMINISTRATOR, ODA-DB	No change
odacli delete-prepatchreport	ODA-ADMINISTRATOR, ODA-DB	No change
odacli list-availablepatches	ODA-ADMINISTRATOR, ODA-DB	No change
odacli describe-component	ODA-ADMINISTRATOR, ODA-DB	No change
odacli describe-latestpatch	ODA-ADMINISTRATOR, ODA-DB	No change
odacli list-agentconfig-parameters	ODA-ADMINISTRATOR	No change
odacli modify-agentconfig-parameters	ODA-ADMINISTRATOR	No change
odacli delete-agentconfig-parameters	ODA-ADMINISTRATOR	No change

ODACLI Command	Roles Required	Changes in command option
odacli update-dbhome	ODA-ADMINISTRATOR, ODA-DB	User odaadmin with the role of ODA-ADMINISTRATOR can update any database home. A custom user with the role of ODA-DB can only update a database home that they have primary access (as the creator) or shared access (as a user who is granted access by odaadmin).
odacli update-dcsadmin	ODA-ADMINISTRATOR	No change
odacli update-dcsagent	ODA-ADMINISTRATOR	No change
odacli update-dcscomponents	ODA-ADMINISTRATOR	No change
odacli update-registry	User with any role	No change
odacli update-repository	ODA-ADMINISTRATOR, ODA-DB	No change
odacli update-server	ODA-ADMINISTRATOR	No change
odacli update-storage	ODA-ADMINISTRATOR	No change
odaadmcli orachk	ODA-ADMINISTRATOR	No change
odacli set-credential	ODA-ADMINISTRATOR, ODA-DB	Not supported when multi-user access is enabled. When multi-user access is enabled, log in as a custom user and use odacli change-password instead.
odacli create-appliance	ODA-ADMINISTRATOR	Add an attribute called "isMultiUserAccessEnabled" :true in the provision.json file. When you run the command, you are prompted to enter and confirm passwords for odaadmin, oracle and grid users and the provisioning job is submitted.
odacli describe-system	User with any role	No change
odacli list-featuretracking	ODA-ADMINISTRATOR	No change
odacli create-backup	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli create-backupconfig	ODA-ADMINISTRATOR, ODA-DB	If creating backupconfig based on ObjectStoreSwift, user must be the creator of the ObjectStoreSwift.
odacli create-objectstoreswift	ODA-ADMINISTRATOR, ODA-DB	No change
odacli delete-backup	ODA-ADMINISTRATOR, ODA-DB	User must have primary (creator of that database) or shared access (granted by odaadmin) on the database whose backup is being deleted.
odacli delete-backupconfig	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli describe-backupreport	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli delete-objectstoreswift	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access

ODACLI Command	Roles Required	Changes in command option
odacli describe-schedule	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli irestore-database	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli list-backupconfigs	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli list-backupreports	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli list-objectstoreswifts	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli list-schedules	ODA-ADMINISTRATOR, ODA-DB	No changes
odacli list-tdebackupreports	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli recover-database	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli restore-archivelog	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli restore-tdewallet	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli modify-backupconfig	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access
odacli modify-schedule	ODA-ADMINISTRATOR, ODA-DB	No change
odacli configure-dataguard	ODA-ADMINISTRATOR, ODA-DB	The creator of the primary database and the standby database could be the same or different users. The default is oda-admin.
odacli deconfigure-dataguard	ODA-ADMINISTRATOR, ODA-DB	The creator of the primary database and the standby database could be the same or different users. The default is oda-admin.
odacli describe-dataguardstatus	ODA-ADMINISTRATOR, ODA-DB	No change
odacli failover-dataguard	ODA-ADMINISTRATOR, ODA-DB	No change
odacli list-dataguardstatus	ODA-ADMINISTRATOR, ODA-DB	No change
odacli reinstate-dataguard	ODA-ADMINISTRATOR, ODA-DB	No change
odacli switchover-dataguard	ODA-ADMINISTRATOR, ODA-DB	No change
odacli list-cpucores	ODA-ADMINISTRATOR	No change
odacli describe-cpucore	ODA-ADMINISTRATOR	No change
odacli modify-cpucore	ODA-ADMINISTRATOR	No change
odacli create-cpupool	ODA-ADMINISTRATOR	Only odaadmin can create it but a non-admin user with role of ODA-DB can use them.
odacli remap-cpupools	ODA-ADMINISTRATOR	No change
odacli list-cpupools	ODA-ADMINISTRATOR, ODA-DB	A non-admin user with the role of ODA-DB can run the odacli list-cpupools command to view all available CPU pools.

ODACLI Command	Roles Required	Changes in command option
odacli describe-cpupool	ODA-ADMINISTRATOR, ODA-DB	A non-admin user with the role of ODA-DB can run the odacli describe-cpupool command to view details of the CPU pool.
odacli modify-cpupool	ODA-ADMINISTRATOR	No change
odacli delete-cpupool	ODA-ADMINISTRATOR	No change
odacli list-auditrecords	ODA-ADMINISTRATOR	No change
odacli describe-auditrecord	ODA-ADMINISTRATOR	No change
odacli clone-database	ODA-ADMINISTRATOR, ODA-DB	User must have primary or shared access to the database home and the source database being cloned. A user with shared access to source database gets only shared access to the cloned database. Hence, the primary owner of the source database remains the primary owner of the cloned database in this case.
odacli create-database	ODA-ADMINISTRATOR, ODA-DB	For database creation in an existing database home, the user must have primary or shared access to the database home. A user with shared access to the database home gets only shared access to the new database, whose primary owner remains the primary owner of the database home. When using existing database storage for the new database, the user must have primary or shared access to that database storage.
odacli delete-database	ODA-ADMINISTRATOR, ODA-DB	Can access only those databases where the user has primary or shared access
odacli describe-database	ODA-ADMINISTRATOR, ODA-DB	Can access only those databases where the user has primary or shared access
odacli list-databases	ODA-ADMINISTRATOR, ODA-DB	User can only view listing of the databases for which they have primary or shared access. Use the -all option to view all databases and request access.
odacli modify-database	ODA-ADMINISTRATOR, ODA-DB	User should have primary or shared access to the database and the backupconfig being attached. Additionally, the primary owner (creator) of the database and backupconfig should be the same.

ODACLI Command	Roles Required	Changes in command option
odacli move-database	ODA-ADMINISTRATOR, ODA-DB	User must have primary or shared access to the source database that is being moved and the target database home. Additionally, the primary owner of the source and target database home must be the same.
odacli register-database	ODA-ADMINISTRATOR, ODA-DB	User must be the primary owner (creator) of the database home on which this database is to be registered.
odacli upgrade-database	ODA-ADMINISTRATOR, ODA-DB	User must have primary or shared access to the source database (the database being upgraded) and the target database home. Additionally, the owner of the source and target database home must be the same.
odacli update-database	ODA-ADMINISTRATOR, ODA-DB	User must have primary or shared access to the source database (the database being updated) and the target database home. Additionally, the owner of the source and target database home must be the same.
odacli configure-dbhome-storage	ODA-ADMINISTRATOR	No change
odacli list-dbhome-storages	ODA-ADMINISTRATOR	No change
odacli describe-dbhome-storage	ODA-ADMINISTRATOR	No change
odacli modify-dbhome-storage	ODA-ADMINISTRATOR	No change
odacli create-dbhome	ODA-ADMINISTRATOR, ODA-DB	Can access only those database homes where the user has primary or shared access.
odacli delete-dbhome	ODA-ADMINISTRATOR, ODA-DB	Can access only those database homes where the user has primary or shared access.
odacli describe-dbhome	ODA-ADMINISTRATOR, ODA-DB	Can access only those database homes where the user has primary or shared access.
odacli list-dbhomes	ODA-ADMINISTRATOR, ODA-DB	Can access only those database homes where the user has primary or shared access. Use the `-all` option to view all database homes and request access.
odacli create-dbstorage	ODA-ADMINISTRATOR	Can access only those dbstorages where the user has primary or shared access.
odacli delete-dbstorage	ODA-ADMINISTRATOR	Can access only those dbstorages where the user has primary or shared access.
odacli describe-dbstorage	ODA-ADMINISTRATOR	Can access only those dbstorages where the user has primary or shared access.

ODACLI Command	Roles Required	Changes in command option
odacli list-dbstorages	ODA-ADMINISTRATOR	Can access only those dbstorages where the user has primary or shared access.
odacli modify-dbstorage	ODA-ADMINISTRATOR	Can access only those dbstorages where the user has primary or shared access.
odacli modify-dgstorage	ODA-ADMINISTRATOR	No change
odacli create-jobdata-retention-policy	ODA-ADMINISTRATOR	No change
odacli create-purge-jobdata-job	ODA-ADMINISTRATOR	No change
odacli delete-jobdata-retention-policy	ODA-ADMINISTRATOR	No change
odacli describe-job	User with any role	No change
odacli list-jobs	User with any role	No change
odacli list-jobdata-retention-policy	ODA-ADMINISTRATOR	No change
odacli list-purge-jobdata-job	ODA-ADMINISTRATOR	No change
odacli list-scheduled-executions	ODA-ADMINISTRATOR	No change
odaadmcli manage diagcollect	ODA-ADMINISTRATOR	No change
odacli create-auto-logclean-policy	ODA-ADMINISTRATOR	No change
odacli create-logcleanjob	ODA-ADMINISTRATOR	No change
odacli describe-logcleanjob	ODA-ADMINISTRATOR	No change
odacli list-auto-logclean-policy	ODA-ADMINISTRATOR	No change
odacli list-logcleanjobs	ODA-ADMINISTRATOR	No change
odacli list-logspaceusage	ODA-ADMINISTRATOR	No change
odacli configure-asr	ODA-ADMINISTRATOR	No change
odacli delete-asr	ODA-ADMINISTRATOR	No change
odacli describe-asr	ODA-ADMINISTRATOR	No change
odacli test-asr	ODA-ADMINISTRATOR	No change
odacli modify-asr	ODA-ADMINISTRATOR	No change
odacli list-osconfigurations	ODA-ADMINISTRATOR	No change
odacli modify-osconfigurations	ODA-ADMINISTRATOR	No change
odaadmcli show cooling	root	No change
odaadmcli show env_hw	root	No change
odaadmcli show fs	root	No change
odaadmcli show memory	root	No change
odaadmcli show network	root	No change
odaadmcli show power	root	No change
odaadmcli show processor	root	No change
odaadmcli show server	root	No change
odaadmcli asmops diskgroup	root	No change
odaadmcli expand storage	root	No change
odacli list-dgdisks	ODA-ADMINISTRATOR, ODA-DB	No change
odacli validate-storagetopology	ODA-ADMINISTRATOR	No change
odaadmcli power disk	root	No change

ODACLI Command	Roles Required	Changes in command option
odaadmcli show controller	root	No change
odaadmcli show disk	root	No change
odaadmcli show diskgroup	root	No change
odaadmcli show raidsyncstatus	root	No change
odaadmcli show storage	root	No change
odaadmcli stordiag	root	No change

odacli Network Commands

Use the `odacli network` commands to list and describe network interfaces.

- odacli configure-firstnet
  Use the command `odacli configure-firstnet` to configure the first network in the appliance after racking and connecting the power and network cables.
- odacli create-network
  Use the `odacli create-network` command to create a network.
- odacli delete-network
  Use the command `odacli delete-network` to delete a network.
- odacli delete-networkinterface
  Use the command `odacli delete-networkinterface` to delete a network interface on Oracle Database Appliance X8-2 hardware models.
- odacli describe-network
  Use the `odacli describe-network` command to display the details of a specific network.
- odacli describe-networkinterface
  Use the `odacli describe-networkinterface` command to display the details of any network interface.
- odacli list-networks
  Use the `odacli list-networks` command to display networks.
- odacli list-networkinterfaces
  Use the `odacli list-networkinterfaces` command to display network interfaces.
- odacli modify-network
  Use the `odacli modify-network` command to update an existing network configuration.

odacli configure-firstnet

Use the command `odacli configure-firstnet` to configure the first network in the appliance after racking and connecting the power and network cables.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# /opt/oracle/dcs/bin/odacli configure-firstnet
```

Usage Notes

This command ensures that the system is available in the network, enabling you to manage the deployment through the Oracle Appliance Manager Browser User Interface.

- Bonded network configuration: Bonded dual-ported interface. With the bonded configuration, you can only enable one of the interfaces. The bonded configuration supports VLANs. The bonded network configuration uses `btbond1` on Oracle Database Appliance hardware models other than X8-2. On Oracle Database Appliance X8-2, the bonded network configuration uses `btbond1` to `btbond6`. This is the default setting.
- Non-bonded network configuration (on Oracle Database Appliance hardware models other than X8-2): Two separate physical network interfaces, this configuration does not support VLANs. The non-bonded network configuration uses `em2` and `em3` as public interfaces. Note that Oracle Database Appliance X8-2 does not support non-bonded network configuration.
- To change a non-bonded network configuration to a bonded configuration, run the Oracle Database Appliance Cleanup Script and redeploy the appliance.
- You can choose to enable Link Aggregation Control Protocol (LACP) on the bonding interface when you run `odacli configure-firstnet`.
- You can use the `-o` option with the `odacli configure-firstnet` command to configure the network speed for Oracle Dual Port 25 Gb Ethernet Adapter.
- You can also use the `-o` option with the `odacli configure-firstnet` command to enable or disable LACP on a bonding interface after deployment.

  Enabling LACP on a bonding interface after deployment:

  ```
  # odacli configure-firstnet -o
  INFO: Configure network option
  Select the Interface to configure the network on (btbond1 btbond3 btbond5)
  [btbond1]:btbond3
  Use LACP bonding on btbond3 (yes/no) [no]: yes
  INFO: network interface [btbond3] setup in LACP mode
  ```

  Disabling LACP on a bonding interface after deployment:

  ```
  # odacli configure-firstnet -o
  INFO: Configure network option
  Select the Interface to configure the network on (btbond1 btbond3 btbond5)
  [btbond1]:btbond3
  Use LACP bonding on btbond3 (yes/no) [yes]: no
  INFO: network interface [btbond3] setup in active-backup mode
  ```

Example 16-3 Configuring the First Network as a Bonded Network

Configure the first network to use a `btbond1` interface without configuring DHCP. Complete the IP address, netmask address, and gateway address.

Values that you need to provide are shown in *italic font*, with the exception of the net1 gateway address; the program obtains the gateway IP address. The program derives this gateway address using the network information you provided for the other IP addresses. Accept this value, unless your network administrator provides an alternative gateway address that is different from the default that the appliance command-line interface detects.

The following is an example of the command on a single node platform:

```
# /opt/oracle/dcs/bin/odacli configure-firstnet

Select the Interface to configure the network on (btbond1): btbond1
Configure DHCP on btbond1 (yes/no): no
INFO: You have chosen Static configuration
Enter the IP address to configure: 10.1.10.2
Enter the Netmask address to configure: 255.255.255.0
Enter the Gateway address to configure: 10.1.10.1
INFO: Plumbing the IPs now
INFO: Restarting the network
Shutting down interface btbond1: bonding: btbond1: Removing slave em2.
bonding: btbond1: releasing active interface em2
bonding: btbond1: making interface em3 the new active one.
bonding: btbond1: Removing slave em3.
bonding: btbond1: releasing active interface em3
::::::::::
```

Example 16-4 Configuring the First Network as a Non-Bonded Network

The following is an example of a non-bonded configuration. For a non-bonded configuration, answer `no` to using a bonding public interface. Note that Oracle Database Appliance X8-2 does not support non-bonded network configuration.

```
# /opt/oracle/dcs/bin/odacli configure-firstnet
Using bonding public interface (yes/no) [yes]: no
INFO: Breaking the bonding on btbond1
INFO: remove bonding module: rmmod bonding
INFO: remove slave setup in /etc/sysconfig/network-scripts/ifcfg-em2
INFO: remove slave setup in /etc/sysconfig/network-scripts/ifcfg-em3
INFO: Restarting the network
Shutting down interface em1: [ OK ]
Shutting down loopback interface: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface em1: [ OK ]
Bringing up interface em2: [ OK ]
Bringing up interface em3: [ OK ]
INFO: Restarting the DCS agent
initdcsagent stop/waiting
initdcsagent start/running, process 57629
Select the Interface to configure the network on (em2 em3) [em2]:
Configure DHCP on em2 (yes/no) [no]:
INFO: You have chosen Static configuration
Enter the IP address to configure : 10.31.102.101
Enter the Netmask address to configure : 255.255.240.0
Enter the Gateway address to configure[10.31.96.1] :
INFO: Plumbing the IPs now
INFO: Restarting the network
Shutting down interface em1: [ OK ]
```

```
Shutting down interface em2: [ OK ]
Shutting down interface em3: [ OK ]
Shutting down loopback interface: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface em1: [ OK ]
Bringing up interface em2: Determining if ip address 10.31.102.101 is already
in use for device em2...
[ OK ]
Bringing up interface em3: [ OK ]
```

Example 16-5 Configuring the Speed of the Network Interface

The following is an example of setting the speed of the network card manually.

```
# odacli configure-firstnet -o
INFO: Configure network option
Select the Interface to configure the network on (btbond1 btbond2 btbond3)
[btbond1]:btbond3

Use LACP bonding on btbond3 (yes/no) [no]:
Set the network speed on btbond3 to 1) AUTO, 2) 1G, 3) 10G, 4) 25G [AUTO]: 3
INFO: The speed of the network port is set to '10G'
INFO: Please set the speed for the connected ports in the switch
INFO: stop the network 'btbond3'
INFO: start the network 'btbond3'
INFO: Network interface btbond3 (p2p1, p2p2) has been set to using speed '10G'
```

odacli create-network

Use the `odacli create-network` command to create a network.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli create-network [-d|-no-d] -n name [-nn nodeName] [-lp listener_ports]
[-sp scan_ports] [-f interface] [ -p ipaddress]  [-w {Public|Dataguard|
Database|Management|Backup|Other}] [-s subnetmask] [-g gateway] [-h] [-sip] [-
sn] [-vs] [-j]
```

Parameters

Parameter	Description
`--defaultnetwork, -d`	(Optional) Identifies the default network.
`--gateway, -g`	(Optional) Defines the network gateway. The gateway is required for the default network.
`--help, -h`	(Optional) Displays help for using the command.
`--interface, -f`	(Optional) Defines the name of the network interface.
`--interfacetype, -t`	(Optional) Defines the network interface type {BOND|PHYSICAL|VLAN|INTERNAL} (non case-sensitive).

Parameter	Description
`--ipaddress, -p`	(Optional) Defines the network IP address. This can be specified in the format IPAddress or nodeNumber0:IPAddress0,nodeNumber1:IPAddress1,...
`--listener-ports, -lp`	(Optional) Specifies the comma-separated list of listener ports. The default value is 1521.
`--scan-ports, -sp`	(Optional) Specifies the comma-separated list of SCAN listener ports. The default value is 1521.
`--json, -j`	(Optional) Displays JSON output.
`--networktype, -w`	(Optional) Defines the type of network. Options are: {Public|Dataguard|Database|Management|Backup|Other}. The default is Other.
`--name, -n`	(Mandatory) Name of the network.
`--no-defaultnetwork, no-d`	(Optional) Identifies a network as not the default network. Use `--defaultnetwork -d` to identify a default network.
`--subnetmask, -s`	(Optional) Defines the Network Subnet Mask.
`--scanips, -sip`	(Optional) Defines the SCAN IP addresses (in the format scanip0,scanip1,...)
`--scanname, -sn`	(Optional) Defines the SCAN name.
`--nodeName, -nn`	(Optional) Defines the node name for High-Availability deployments.
`--vips, -vs`	(Optional) List of virtual IP addresses. They can be specified in the format vipname0:nodeNumber0:vip0,vipname1:nodenumber1:vip1,...
`--vlanid, -v`	(Optional) Add VLAN as a new network. This option is available for for High-Availability deployments.

Usage Notes

- Network of `public` type can only be defined when deploying the appliance or when running `odacli configure-firstnet` during initial provisioning.
- Use this command to create an additional network not done in `create-appliance`.
- You are only allowed to create a network on the bond interface.
- On Oracle Database Appliance non-High Availability systems, you cannot specify Virtual IP addresses. The Virtual IP address is set to the same value as the IP address.
- Single Client Access Name (SCAN) cannot be set up on Oracle Database Appliance non-High Availability systems.

Example 16-6 Creating a Network

The following example creates a new network, `sfpbond1`, with IP address `192.0.2.15`. The network is an additional network that uses subnet mask `255.255.255.0` and is not a default network.

```
# odacli create-network -n network1 -f sfpbond1 -p 192.0.2.15 -w Backup -s
255.255.255.0 -no-d
```

Example 16-7 Creating a VLAN Network

```
odacli create-network -g 10.31.131.129 -f btbond1 -t VLAN -p
0:10.31.131.132,1:10.31.131.133 -nn vlan150 -w Database -no-d -sip
10.31.131.134,10.31.131.135 -sn scaoda702c1-vlan150-scan -s 255.255.255.128 -
vs n1-vlan150-vip:0:10.31.131.136,n2-vlan150-vip:1:10.31.131.137 -v 150
```

odacli delete-network

Use the command `odacli delete-network` to delete a network.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

To delete a network:

```
odacli  delete-network -i id [-n] [-h] [-j]
```

Parameters

Parameter	Description
`--id, -i`	Defines the network identity.
`--name, -n`	Specifies the network name.
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

You cannot delete the Public-network or Private-network after the system is deployed.

On High-Availability systems, if the node number is not specified in the `odacli delete-network` command, then the network on both nodes is deleted.

Example 16-8 Deleting a Network

The following example deletes a backup network with a network ID of 55db39db-d95c-42c5-abbd-b88eb99b83ec.

```
# odacli delete-network -i 55db39db-d95c-42c5-abbd-b88eb99b83ec



"jobId" : "c26d217e-419b-4a91-8680-7b06bcfe9828",
"status" : "Running",
"message" : null,
"reports" : [ {
     "taskId" : "TaskSequential_137",
     "taskName" : "deleting network",
     "taskResult" : "Running",
     "startTime" : "September 8, 2024 23:14:32 PM EDT",
     "endTime" : "September 8, 2024 23:14:32 PM EDT",
```

```
    "status" : "Running",
    "taskDescription" : null,
    "parentTaskId" : "TaskSequential_135",
    "jobId" : "c26d217e-419b-4a91-8680-7b06bcfe9828",
    "tags" : [ ],
    "reportLevel" : "Info",
    "updatedTime" : "September 8, 2024 23:14:32 PM EDT"
  },{
  "taskId" : "TaskZJsonRpcExt_142",
    "taskName" : "Setting up Network",
    "taskResult" : "Network setup success",
    "startTime" : "September 8, 2024 23:14:32 PM EDT",
    "endTime" : "September 8, 2024 23:14:32 PM EDT",
    "status" : "Success",
    "taskDescription" : null,
    "parentTaskId" : "TaskParallel_141",
    "jobId" : "c26d217e-419b-4a91-8680-7b06bcfe9828",
    "tags" : [ ],
    "reportLevel" : "Info",
    "updatedTime" : "September 8, 2024 23:14:32 PM EDT"   } ],
  "createTimestamp" : "September 8, 2024 23:14:32 PM EDT",
  "description" : "Network service delete",
  "updatedTime" : "September 8, 2024 23:14:32 PM EDT"
}
```


odacli delete-networkinterface

Use the command `odacli delete-networkinterface` to delete a network interface on Oracle Database Appliance X8-2 hardware models.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To delete a network interface:

```
odacli  delete-networkinterface [-i id] [-n name] [-h] [-j]
```

Parameters

Parameter	Description
`--id`, `-i`	Defines the network interface ID.
`--name`, `-n`	Defines the network interface name.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

You cannot delete the network interface card in slot 7.

Example 16-9 Deleting a Network Interface

```
[root@oak ~]/opt/oracle/dcs/bin/odacli delete-networkinterface -n btbond4
{
"jobId" : "a81066e2-417e-4df6-b810-08df24c646c1",
"status" : "Running",
"message" : null,
"reports" : ,
"createTimestamp" : "September 8, 2024 00:01:09 AM UTC",
"resourceList" : ,
"description" : "Network Interface Deletion with name btbond4",
"updatedTime" : "September 8, 2024 00:01:09 AM UTC"
}
```

Example 16-10 Deleting btbond1 Network Interface is Not Allowed

```
[root@oak ~]# /opt/oracle/dcs/bin/odacli delete-networkinterface -n btbond1
DCS-10001:Internal error encountered: Delete operation is not allowed on
network interface : btbond1.
```

Example 16-11 Deleting Network Interface When Network Exists is Not Allowed

```
[root@oak ~]# opt/oracle/dcs/bin/odacli delete-networkinterface -n btbond3
DCS-10001:Internal error encountered: Delete operation is not allowed –
networks {[Public-network]} found on interface btbond3. Pleae delete the
network first.
```

odacli describe-network

Use the `odacli describe-network` command to display the details of a specific network.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To display the details of a specific network:

```
odacli describe-network [-i id] [-nn node_name] [-n name] [-j][-h]
```

Parameters

Parameter	Description
`--id`, `-i`	Identifies the network ID. Use the `odacli list-networks` command to obtain the `id`.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.
`--name`, `-n`	Specifies the network name.
`--node-name`, `-nn`	(Optional) Displays the network interface information for the specified node name.

Example 16-12 Displaying Network Details

Enter the following command to display the details of the public network:

```
# odacli describe-network -n Public-network

Network details
----------------------------------------------------------------
                     ID:  a11573f1-cad4-41fc-9dd0-3ba958b605f8
                   Name:  Public-network
                    NIC:  btbond1
         Interface Type:  BOND
            Subnet Mask:  255.255.255.0
                Gateway:  10.31.116.1
                VLAN ID:
                   Type:  Public
                Default:  true
                Created:  March 30, 2020 8:37:00 AM PDT
                Updated:  March 30, 2023 10:25:27 AM PDT
                 Status:  Configured
          Node Networks:  [IP Address on node0: 10.31.117.60, IP Address on
node1: 10.31.117.62]
              Scan Name:  oda1-scan
               Scan Ips:  10.31.117.64 10.31.117.65
                   Vips:  [oda1-vip on node0: 10.31.116.61, oda2-vip on
node1: 10.31.116.63]
   Associated Databases: odacn db122a db122b db122c db122d
```

Example 16-13 Displaying Network Details

Enter the following command to display the details of network ID 93c07043-3002-494a-8fa0-f3ae932fb4c5:

```
# odacli describe-network -i 93c07043-3002-494a-8fa0-f3ae932fb4c5 -nn oda1

Network details
----------------------------------------------------------------
                     ID:  93c07043-3002-494a-8fa0-f3ae932fb4c5
                   Name:  Private-network
                    NIC:  ibbond0
          InterfaceType:  BOND
             IP Address:  192.168.16.25
            Subnet Mask:  255.255.255.0
                Gateway:
                 VlanId:
                   Type:  Private
                Default:  false
                Created:  November 18, 2023 10:54:16 PM PST
```

odacli describe-networkinterface

Use the `odacli describe-networkinterface` command to display the details of any network interface.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To display the details of a specific network interface:

```
odacli describe-networkinterface -i id -n name [-nn node_name] [-j][-h]
```

Parameters

Parameter	Description
`--id`, `-i`	Identifies the network interface ID. Use the `odacli list-networks` command to obtain the `id`.
`--name`, `-n`	Specifies the network interface name.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.
`--node-name`, `-nn`	(Optional) Displays the network interface information for the specified node name.

Usage Notes

By default, this command always displays the network interface description of the local node. To display the description of another node, specify the node name option (`-nn`).

Example 16-14 Displaying Network Details

Enter the following command to display the details of network interface with ID fe1bf0a7-f56e-44cd-9a84-f374c0aa4b61:

```
# /opt/oracle/dcs/bin/odacli describe-networkinterface
-i fe1bf0a7-f56e-44cd-9a84-f374c0aa4b61 -nn oda1

Network Interface details
----------------------------------------------------------------
                     ID:  fe1bf0a7-f56e-44cd-9a84-f374c0aa4b61
                   Name:  eth2
                    NIC:  eth2
                   Type:  PHYSICAL
                Members:  eth2
        Slave Interface:  true
                Created:  October 16, 2023 1:16:59 PM PDT
```

odacli list-networks

Use the `odacli list-networks` command to display networks.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-networks [-nn node_name] [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.
`--node-name`, `-nn`	(Optional) Displays the network interface information for the specified node name.

Example 16-15 Displaying a List of Networks

Use the `odacli list-networks` command to display a list of networks:

```
# odacli list-networks
ID Name NIC Interface Type Subnet Mask Gateway VLAN ID Node Networks
------------------------------------ -------------------- ------------
-------------- ------------------ ------------------ --------
----------------------
30012eeb-1205-4cb6-9fc0-488972b2e420 Private-network priv0 INTERNAL
255.255.255.240 [IP Address on node0: 192.168.16.24]
75d01d4f-e940-4d0f-a4fe-ee04f5a90c73 Public-network btbond1.370 VLAN
255.255.255.0 10.209.10.241 370 [IP Address on node0: 10.209.10.244]
```

odacli list-networkinterfaces

Use the `odacli list-networkinterfaces` command to display network interfaces.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-networkinterfaces [-nn node_name] [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--node-name`, `-nn`	(Optional) Displays the network interface information for the specified node name.

Example 16-16 Displaying a List of Network Interfaces

Use the `odacli list-networkinterfaces` command to display a list of network interfaces:

```
# odacli list-networkinterfaces -nn oda1
ID                                       Name                 NIC
Type
---------------------------------------- --------------------
d5907a23-1c5d-48c7-8d47-2c188ed43ddd     bond0                bond0
BOND
4f7ea558-9a43-42a9-8e08-6bfdf3a33229     bond1                bond1
BOND
743ced8d-bbe5-4987-b316-5fdf95d5e60b     eth0                 eth0
PHYSICAL
a4121f9e-e694-4852-a521-44efc6ef3fde     eth1                 eth1
PHYSICAL
c62c5f04-aa93-4783-a3a7-275bf9fab2d9     eth2                 eth2
PHYSICAL
bda21dad-5c1c-4073-89e5-798b8fce8533     eth3                 eth3
PHYSICAL
0141f1ac-5c34-4393-8b99-76094b6f795c     ib0                  ib0
PHYSICAL
e8cb138a-087f-4739-bb8d-90b1d903aeb6     ib1                  ib1
PHYSICAL
a31cfd63-fb90-4cbb-a2fb-382c5e33983b     ibbond0              ibbond0
BOND
```

odacli modify-network

Use the `odacli modify-network` command to update an existing network configuration.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To update a network:

```
odacli  modify-network -i id -n name [-p IP address] [-lp listener_ports] [-
sp scan_ports] [-w {Public|Dataguard|Database|Management|Backup|Other} ] [-s
network subnet mask] [-g network gateway] [-sip] [-sn] [-vs] [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	Defines the network identity.
`--name, -n`	Defines the network name.
`--gateway, -g`	(Optional) Defines the network gateway.
`--ipaddress, -p`	(Optional) Defines the network IP address. This can be specified in the format IPAddress or nodeNumber0:IPAddress0,nodeNumber1:IPAddress1,...
`--networktype, -w`	(Optional) Defines the type of network. {Public|Dataguard|Database|Management|Backup|Other}
`--scanips, -sip`	Defines the SCAN IP addresses (in the format scanip0,scanip1,...)
`--scanname, -sn`	Defines the SCAN name.
`--listener-ports, -lp`	(Optional) Specifies the comma-separated list of listener ports. The default value is 1521.
`--scan-ports, -sp`	(Optional) Specifies the comma-separated list of SCAN listener ports. The default value is 1521.
`--subnetmask, -s`	(Optional) Defines the Network Subnet Mask.
`--vips, -vs`	List of virtual IP addresses. They can be specified in the format vipname0:nodeNumber0:vip0,vipname1:nodenumber1:vip1,...
`--help,-h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- You cannot modify the Public and Private-interfaces after the system is deployed.
- On Oracle Database Appliance non-High Availability systems, you cannot update Virtual IP addresses. The Virtual IP address is set to the same value as the IP address.
- Single Client Access Name (SCAN) cannot be set up on Oracle Database Appliance non-High Availability systems.
- On Oracle Database Appliance High Availability systems, you cannot update network of type Database to other types of network (such as backup, management and so on). You can, however, update IP address, Virtual IP address, Gateway, and Subnet mask for all network types.
- The system has both SFP+ and 10GBaseT bonded pairs, which means that one of them is used for the public, and you can configure the other after deployment if you want additional connectivity. For example, if you want a backup network.

Example 16-17 Modifying a Network

The following example updates the network ID `192.0.0.2` and designates the network as a backup network:

```
# odacli modify-network -i 192.0.0.2 -w Backup
```

odacli Multi-User Access Commands

Use `odacli` commands to enable multi-user access for the appliance.

> **Note:**
>
> The multi-user access feature is available for standalone Oracle Database Appliance systems. During provisioning, a single domain and tenancy is created by default and all users are created within the default domain and tenancy.

- odacli list-domains
  Use the `odacli list-domains` command to list all multi-user access domains.
- odacli describe-domain
  Use the `odacli describe-domain` command to describes a multi-user access domain.
- odacli list-tenants
  Use the `odacli list-tenants` command to list all tenants in a multi-user access domain.
- odacli describe-tenant
  Use the `odacli describe-tenant` command to describe a tenant in a multi-user access domain.
- odacli create-user
  Use the `odacli create-user` command to create a user in a multi-user access system.
- odacli activate-user
  Use the `odacli activate-user` command to activate a user in a multi-user access system.
- odacli list-users
  Use the `odacli list-users` command to list all users in a multi-user access system.
- odacli describe-user
  Use the `odacli describe-user` command to describe a user in a multi-user access system.
- odacli authorize-user
  Use the `odacli authorize-user` command in a multi-user access system to unlock a user account that is locked due to multiple failed login attempts or where the password has expired, and authorize the user to reset their password to reactivate their account.
- odacli delete-user
  Use the `odacli delete-user` command in a multi-user access system to delete a user account.
- odacli list-user-roles
  Use the `odacli list-user-roles` command to list user roles in a multi-user access system.
- odacli describe-user-role
  Use the `odacli describe-user-role` command to describe a user role in a multi-user access system.

- odacli list-user-entitlements
  Use the `odacli list-user-entitlements` command to list all entitlements in a multi-user access system.
- odacli describe-user-entitlement
  Use the `odacli describe-user-entitlement` command to describe an entitlement in a multi-user access system.
- odacli list-user-operations
  Use the `odacli list-user-operations` command to list all operations in a multi-user access system.
- odacli describe-user-operation
  Use the `odacli describe-user-operation` command to describe an operation in a multi-user access system.
- odacli list-resources-access
  Use the `odacli list-resources-access` command to list access to DCS resources in a multi-user access system.
- odacli describe-resource-access
  Use the `odacli describe-resource-access` command to describe access to a DCS resource in a multi-user access system.
- odacli grant-resource-access
  Use the `odacli grant-resource-access` command to grant shared access to a DCS resource in a multi-user access system.
- odacli revoke-resource-access
  Use the `odacli revoke-resource-access` command to revoke access to a DCS resource in a multi-user access system.
- odacli reset-password
  Use the `odacli reset-password` command to reset the password for a user, after it is unlocked, in a multi-user access system.
- odacli change-password
  Use the `odacli change-password` command to change the password for a user in a multi-user access system.

odacli list-domains

Use the `odacli list-domains` command to list all multi-user access domains.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli list-domains [-j] [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

- The `odaadmin` user can run this command to view all the multi-user access domains.

Example 16-18

```
[odaadmin@oda1 bin]$ odacli list-domains
ID                                       Name         Description    Is Active
----------------------------------------------------------------------------
811f176a-26c9-4480-afb3-2c735d565d91 odalocal.com Default domain true
```

odacli describe-domain

Use the `odacli describe-domain` command to describes a multi-user access domain.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli describe-domain -dn` *`domain_name`* `[-j] [-h]`

Parameters

Parameter	Description
`--domain-name, -dn`	(Required) Specifies the multi-user access domain.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- The `odaadmin` user can run this command to view details of the multi-user access domain.

Example 16-19

```
# odacli describe-domain -dn odalocal.com
Domain details
----------------------------------------------------------------
                     ID: 811f176a-26c9-4480-afb3-2c735d565d91
                   Name: odalocal.com
            Description: Default domain
              Is Active: true
                Created: November 22, 2021 7:39:50 PM UTC
           Last Updated: November 22, 2021 7:39:50 PM UTC
```

odacli list-tenants

Use the `odacli list-tenants` command to list all tenants in a multi-user access domain.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-tenants -dn domain_name [-j] [-h]
```

Parameters

Parameter	Description
`--domain-name, -dn`	(Optional) Specifies the multi-user access domain name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- The `odaadmin` user can run this command to view the tenants in a multi-user access domain.

Example 16-20

```
[odaadmin@oda1 bin]$ odacli list-tenants
ID Domain Name                        Tenant Name  Tenant   Description    Is
Active
------------------------------------------------------------------------------
---------------------- ---------------------------------------- ----------
1656723b-f614-4539-9634-d9013e611015 odalocal.com odalocal Default tenant true
```

odacli describe-tenant

Use the `odacli describe-tenant` command to describe a tenant in a multi-user access domain.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-tenant -tn tenant_name -dn domain_name [-j] [-h]
```

Parameters

Parameter	Description
`--tenant-name, -tn`	(Required) Specifies the multi-user access tenant name.

Parameter	Description
`--domain-name, -dn`	(Optional) Specifies the multi-user access domain name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- The `odaadmin` user can run this command to view the tenant details in a multi-user access domain.

Example 16-21

```
[odaadmin@oda1 bin]$ odacli describe-tenant -tn odalocal
Tenant details
----------------------------------------------------------------
                     ID: 1656723b-f614-4539-9634-d9013e611015
            Domain Name: odalocal.com
            Tenant Name: odalocal
     Tenant Description: Default tenant
              Is Active: true
                Created: November 22, 2021 7:39:50 PM UTC
           Last Updated: November 22, 2021 7:39:50 PM UTC
```

odacli create-user

Use the `odacli create-user` command to create a user in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli create-user -u` *`user_id`* `-r` *`roles`* `[-gc] [-j] [-h]`

Parameters

Parameter	Description
`--user-id, -u`	(Required) Specifies the user whose details are being created.
`--user-roles, -r`	(Required) Specifies the role to be assigned to the user. Use commas to specify multiple roles.
`--generate-cert, -gc`	(Optional) This option generates and registers a pair of certificates, one for the client and the other for the server (DCS agent). After these certificates are mutually exchanged and authenticated, the client and server can communicate with each other using mTLS (mutual Transport Layer Security). This attribute is optional and must be enabled only when you need to connect to the DCS agent with mTLS in addition to Basic Authentication.
`--help, -h`	(Optional) Displays help for using the command.

Parameter	Description
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to create a user in a multi-user access system. After `odaadmin` creates a user and generates temporary login credentials, the new user must activate their account by connecting to the appliance with the temporary password. Run the `odacli activate-user` command and provide the temporary password and then specify the new password. You can also activate the user from the Browser User Interface.

Example 16-22

```
[odaadmin@oda1 bin]$ odacli create-user -u dcsdb2 -r ODA-DB
Enter ODA account password for user 'dcsdb2':
Retype ODA account password for user 'dcsdb2':

Job details
----------------------------------------------------------------
                     ID: 856b3d79-8ae1-427c-a77a-8893254a451f
            Description: User creation(dcsdb2)
                 Status: Running
                Created: November 29, 2023 7:01:54 AM UTC
                Message:

Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------

[odaadmin@oda1 bin]$ odacli describe-job -i 856b3d79-8ae1-427c-
a77a-8893254a451f
Job details
----------------------------------------------------------------
                     ID: 856b3d79-8ae1-427c-a77a-8893254a451f
            Description: User creation(dcsdb2)
                 Status: Success
                Created: November 29, 2023 7:01:54 AM UTC
                Message:

Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
```

odacli activate-user

Use the `odacli activate-user` command to activate a user in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli activate-user [-j] [-h]
```

Parameters

Parameter	Description
`--domain-name, -d`	(Optional) Specifies the name of the domain in which the user exists.
`--tenant-name, -t`	(Optional) Specifies the tenancy in which the user exists.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to activate a user in a multi-user access system. To activate a user, the user to be activated must connect to the Oracle Database Appliance system through SSH and then run the command.

Example 16-23

```
[dcsdb2@oda1 bin]$ odacli activate-user
Enter your ODA account password:
Enter new password for your ODA account:
Retype new password for your ODA account:
User 'dcsdb2' activated successfully.
```

odacli list-users

Use the `odacli list-users` command to list all users in a multi-user access system.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli list-users [-d domain_name] [-t tenant_name] [-u user_id] [-j] [-h]
```

Parameters

Parameter	Description
`--domain-name, -d`	(Optional) Specifies the name of the domain in which the user exists.
`--tenant-name, -t`	(Optional) Specifies the tenancy in which the user exists.
`--user-id, -u`	(Optional) Specifies the user for whom the list is being generated.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- The `odaadmin` user can run this command to view all the users in a multi-user access system.

Example 16-24

```
[odaadmin@oda1 bin]$ odacli list-users

ID                                   DCS User Name  OS User Name
Role(s)           Account Status  User Type
-------------------------------------------------------------------------------
-------------------------------------------------
------------------------- ---------------
13b64a36-23f8-4ba2-9af3-e0a17a16b376 dcsdb2         dcsdb2       ODA-
DB            Active          Custom
5006da5b-0fc0-4d6f-9115-17d89ad235e4 oakdroot       oakdroot     ODA-
OAKDROOT      Active          System
7af066af-55bd-48a9-8db5-c0d3049a1541 ugrid          ugrid        ODA-
GRID          Active          System
8361ca4b-8747-4687-9459-799c6a17d1b1 uoracle        uoracle      ODA-
DB            Active          System
a205ce98-2675-49a2-81e2-4e1c7540f023 odaadmin       odaadmin     ODA-
ADMINISTRATOR Active          System
```

odacli describe-user

Use the `odacli describe-user` command to describe a user in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-user [-u user_id] [-d domain_name] [-rd] [-t tenant_name] [-
j] [-h]
```

Parameters

Parameter	Description
`--user-id, -u`	(Required) Specifies the user whose details are being viewed.
`--domain-name, -d`	(Optional) Specifies the name of the domain in which the user exists.
`--role-details, -rd`	(Optional) Specifies the entitlements and operations associated with the user role.
`--tenant-name, -t`	(Optional) Specifies the tenancy in which the user exists.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view details about a user in a multi-user access system.

Example 16-25

```
[odaadmin@oda1 bin]$ odacli describe-user -u dcsdb1

User details
----------------------------------------------------------------
                     ID: be3182b4-9933-44a8-89b8-02b883db9d30
            DCS User ID: dcsdb1
             OS User ID: dcsdb1
       Primary OS Group: uoinstall
    Secondary OS Groups: uasmdba,dcsdb1_dba,dcsdb1_dbaoper,dcsdb1,dcsagent
                Role(s): ODA-DB
         Account Status: Active
         Account Status: Custom
                Created: November 25, 2021 12:21:16 PM UTC
           Last Updated: November 25, 2021 12:27:31 PM UTC
```

odacli authorize-user

Use the `odacli authorize-user` command in a multi-user access system to unlock a user account that is locked due to multiple failed login attempts or where the password has expired, and authorize the user to reset their password to reactivate their account.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli authorize-user -u` *`user_id`* `[-d` *`domain_name`*`] [-t` *`tenant_name`*`] [-j] [-h]`

Parameters

Parameter	Description
`--user-id, -u`	(Required) Specifies the user whose account is locked.
`--domain-name, -d`	(Optional) Specifies the name of the domain in which the user exists.
`--tenant-name, -t`	(Optional) Specifies the tenancy in which the user exists.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to authorize a user in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-26

```
[odaadmin@oda1 bin]$ odacli authorize-user -u dcsdb2
Enter your ODA account password:
Enter new password for user 'dcsdb2':
Retype new password for user 'dcsdb2':
User 'dcsdb2' authorized successfully.
User can now login with the temporary password assigned to them by the system
administrator
and reset their password by executing 'odacli reset-password' command or via
BUI.
```

odacli delete-user

Use the `odacli delete-user` command in a multi-user access system to delete a user account.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli delete-user -u` *`user_id`* `[-d` *`domain_name`*`] [-t` *`tenant_name`*`] [-j] [-h]`

Parameters

Parameter	Description
`--user-id, -u`	(Required) Specifies the user account to be deleted.
`--domain-name, -d`	(Optional) Specifies the name of the domain in which the user exists.
`--tenant-name, -t`	(Optional) Specifies the tenancy in which the user exists.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to delete a user in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.
- The user can be deleted only if there are no resources currently being held in their name.

Example 16-27

```
[odaadmin@oda1 bin]$ odacli delete-user -u dcsdb2

Job details
----------------------------------------------------------------
                     ID: a8724a04-3225-43b3-9484-a949bce1c5c0
            Description: User deletion(dcsdb2)
                 Status: Success
                Created: November 19, 2021 11:44:35 AM UTC
                Message:
```

```
Task Name Start Time End Time Status dcs
------------------------------------ --------------------------------
---------------------------------- ----------

[odaadmin@oda1 bin]$ odacli describe-job -i a8724a04-3225-43b3-9484-
a949bce1c5c0

Job details
----------------------------------------------------------------
                     ID: a8724a04-3225-43b3-9484-a949bce1c5c0
            Description: User deletion(dcsdb2)
                 Status: Success
                Created: November 19, 2021 11:44:35 AM UTC
                Message:

Task Name Start Time End Time Status
------------------------------------ --------------------------------
---------------------------------- ----------
```

odacli list-user-roles

Use the `odacli list-user-roles` command to list user roles in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli list-user-roles [-n` *role_name*`] [-j] [-h]`

Parameters

Parameter	Description
`--role-name, -n`	(Optional) Displays the list for the specified role.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view all the roles in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-28

```
[odaadmin@oda1 bin]$ odacli list-user-roles

Role Name         Role Description                                  Is Admin
Role Is Internal Is Active
-------------------------
--------------------------------------------------------------------------------
```

```
------------ -------------- -------------- ----------
ODA-DB            Role with DB provisioning related entitlements
false         false       true
ODA-GRID          Role with GRID provisioning related entitlements
false         true        true
ODA-DBVMINFRA     Role with DBVM user related entitlements false
true          true
ODA-ADMINISTRATOR System Administrator role with all entitlements
true          true        true
ODA-OAKDROOT      Role with OAKD related entitlements
false         true        true
```

odacli describe-user-role

Use the `odacli describe-user-role` command to describe a user role in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli describe-user-role [-n role_name] [-j] [-h]`

Parameters

Parameter	Description
`--role-name, -n`	(Required) Specifies the user role for which you want to list the details.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to describe a user role in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-29

```
[odaadmin@oda1 bin]$ odacli describe-user-role -n ODA-DB
Role details
----------------------------------------------------------------
                    Name: ODA-DB
             Description: Role with DB provisioning related entitlements
          Is Admin Role: false
            Is Internal: false
              Is Active: true
  Mapped Entitlement(s): PROVISIONDB-MGMT,BACKUPDB-MGMT,PATCHDB-
MGMT,DATAGUARD-MGMT,DBSTORAGE-MGMT,OBJSTORESWIFT-MGMT,RMANREPORT-
MGMT,SCHEDULE-MGMT,ORACHK-MGMT,ASM-RDBMS-INSTANCE-MGMT,CPUPOOLVIEW-
MGMT,COMMON-ALL
```

odacli list-user-entitlements

Use the `odacli list-user-entitlements` command to list all entitlements in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli list-user-entitlements [-ao] [-n entitlement_name] [-j] [-h]`

Parameters

Parameter	Description
`--active-only, -ao`	(Optional) Lists the active entitlements.
`--entitlement-name, -n`	(Optional) Displays the list for the specified entitlement.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view all entitlements in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-30

```
[odaadmin@oda1 bin]$ odacli list-user-entitlements

Entitlement Name        Entitlement
Description                                                 Is Active
--------------------------------
--------------------------------------------------------------------
TOKEN-MGMT              Entitlement for performing user token management
operations            true
OSCONFIGS-MGMT          Entitlement for performing os configurations related
operations        true
FEATURETOGGLE-MGMT      Entitlement for performing feature enable/diable
management operations true
SERVER-MGMT             Entitlement for performing server related
operations                 true
HAT-MGMT                Entitlement for performing HAT Administrative
operations                true
ASM-RDBMS-INSTANCE-MGMT Entitlement for performing ASM/RDBMS instance related
operations       true
RMANREPORT-MGMT         Entitlement for performing RMAN report(s) related
operations           true
DBHOME-MGMT             Entitlement for performing DBHomes (on ACFS) related
operations        true
DCSCOMPS-MGMT           Entitlement for performing DCS component related
operations            true
```

```
KVM-MGMT-INTERNAL       Entitlement for performing blackbox KVM related
internal operations    true
...
```

odacli describe-user-entitlement

Use the `odacli describe-user-entitlement` command to describe an entitlement in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-user-entitlement -n entitlement_name [-j] [-h]
```

Parameters

Parameter	Description
`--entitlement-name, -n`	(Required) Displays the details for the specified entitlement.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view an entitlement in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-31

```
[odaadmin@oda1 bin]$ odacli describe-user-entitlement -n PROVISIONDB-MGMT

Entitlement details
----------------------------------------------------------------
                     Name: PROVISIONDB-MGMT
             Description: Entitlement for performing DB provision related
operations
                Is Active: true
    Mapped Operation(s): clone-database,create-database,delete-
database,describe-database,list-databases,modify-database,move-
database,register-database,update-database,upgrade-database,create-
dbhome,delete-dbhome,describe-dbhome,list-dbhomes,describe-netsecurity,update-
netsecurity,list-supported-dbversions,get-db-shapes,restore-
archivelog,restore-tdewallet,modify-dbfileattributes
```

odacli list-user-operations

Use the `odacli list-user-operations` command to list all operations in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli list-user-operations [-ao] [-n operations_name] [-j] [-h]`

Parameters

Parameter	Description
`--active-only, -ao`	(Optional) Lists the active multi-user access operations.
`--operation-name, -n`	(Optional) Displays the list for the specified operation.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view operations in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-32

```
[odaadmin@oda1 bin]$ odacli list-user-operations

Operation Name           Operation Description      Is Active
---------------------------------
--------------------------------------------------------------------------------
---------------- ----------
list-orachkreports       List orachkreports         true
list-osconfigurations    List os configurations     true
describe-database        Describe database          true
retrieveall-resources    Retrieve all resources     true
modify-dbsystem-vm       Modify DB System VMs       true
list-backupconfigs       List backup configurations true
```

odacli describe-user-operation

Use the `odacli describe-user-operation` command to describe an operation in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-user-operation [-n operation_name] [-j] [-h]
```

Parameters

Parameter	Description
`--operation-name, -n`	(Optional) Displays the details for the specified operation.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view an operation in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-33

```
[odaadmin@oda1 bin]$ odacli describe-user-operation -n create-database
Operation details
----------------------------------------------------------------
                  Name: create-database
           Description: Create database
             Is Active: true
```

odacli list-resources-access

Use the `odacli list-resources-access` command to list access to DCS resources in a multi-user access system.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli list-resources-access [-ao] [-rn resource_name] [-rt resource_type] [-
j] [-h]
```

Parameters

Parameter	Description
`--active-only, -ao`	(Optional) Lists the active resources.
`--resource-name, -rn`	(Optional) Displays the list by resource name.
`--resource-type, -rt`	(Optional) Displays the list for the specified resource type, for example, `oda_dbhome`, `oda_db`, `oda_dbstorage`, `oda_objstoreswift`, and `oda_backupconfig`.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view access to resources in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-34

```
[odaadmin@oda1 bin]$ odacli list-resources-access

Resource ID                           Resource Name    Resource
Location                                          Resource Type    Primary
Owner Shared Access Is Active
--------------------------------------
--------------------------------------------------------------------------------
---------------- ---------------------------------------- ----------
cfc65e00-5d9e-4c00-a7a0-f7ea37884e06 asmtde1          53f434fe-598e-46a0-bbdc-
a456404da0c9                        ODA_DB            odadb1
odaadmin      true
1dd795e1-b582-49f1-b1fb-eb3d77eb158b bkpcfg_o1
ObjectStore                                                 ODA_BACKUPCONFIG
odadb1       odaadmin      true
06682f78-8f5f-4dd2-a918-7935e00bf0c4 o1               https://
swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1 ODA_OBJSTORESWIFT
odadb1       odaadmin      true
53f434fe-598e-46a0-bbdc-a456404da0c9 OraDB19000_home1 /u01/app/odaorahome/
odadb1/product/19.0.0.0/dbhome_1       ODA_DBHOME        odadb1
odaadmin      true
abc9a670-493e-4226-b186-6bc91ea48b9a default
Disk                                                        ODA_BACKUPCONFIG
uoracle      odaadmin      true
2360eebc-8899-471c-828f-c008e56fefea asmtde1
DATA,RECO,RECO                                              ODA_DBSTORAGE
odadb1       odaadmin      true
```

odacli describe-resource-access

Use the `odacli describe-resource-access` command to describe access to a DCS resource in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-resource-access [-ri resource_ID] [-j] [-h]
```

Parameters

Parameter	Description
`--resource-ID, -ri`	(Required) Specifies the resource ID.
`--help, -h`	(Optional) Displays help for using the command.

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to view access to a resource in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-35

```
[odaadmin@oda1 bin]$ odacli describe-resource-access -ri cfc65e00-5d9e-4c00-
a7a0-f7ea37884e06

Resource Access details
----------------------------------------------------------------
            Resource ID: cfc65e00-5d9e-4c00-a7a0-f7ea37884e06
          Resource Name: asmtde1
      Resource Location: 53f434fe-598e-46a0-bbdc-a456404da0c9
          Resource Type: ODA_DB
          Primary Owner: odadb1
          Shared Access: odaadmin
              Is Active: true
                Created: November 17, 2021 1:31:55 PM UTC
           Last Updated: November 17, 2021 1:31:55 PM UTC
```

odacli grant-resource-access

Use the `odacli grant-resource-access` command to grant shared access to a DCS resource in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli grant-resource-access [-ri` *`resource_ID`*`] [-u` *`user_ID`*`] [-j] [-h]`

Parameters

Parameter	Description
`--resource-ID`, `-ri`	(Required) Specifies the resource ID.
`--user-ID`, `-u`	(Required) Specifies the user name to be granted shared access.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to grant access to a resource in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-36

```
[odaadmin@oda1 bin]$ odacli grant-resource-access -ri cfc65e00-5d9e-4c00-a7a0-
f7ea37884e06 -u odadb2
Shared access of resource granted to user successfully...
[odaadmin@oda1 bin]$ odacli describe-resource-access -ri cfc65e00-5d9e-4c00-
a7a0-f7ea37884e06
Resource Access details
----------------------------------------------------------------
            Resource ID: cfc65e00-5d9e-4c00-a7a0-f7ea37884e06
          Resource Name: asmtde1
      Resource Location: 53f434fe-598e-46a0-bbdc-a456404da0c9
          Resource Type: ODA_DB
          Primary Owner: odadb1
          Shared Access: odaadmin,odadb2
              Is Active: true
                Created: November 17, 2021 1:31:55 PM UTC
           Last Updated: November 19, 2021 12:46:55 PM UTC
```

odacli revoke-resource-access

Use the `odacli revoke-resource-access` command to revoke access to a DCS resource in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli revoke-resource-access [-ri resource_ID] [-u user_ID] [-j] [-h]
```

Parameters

Parameter	Description
`--resource-ID, -ri`	(Required) Specifies the resource ID.
`--user-ID, -u`	(Required) Specifies the user name whose shared access is to be revoked.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to revoke access to a resource in a multi-user access system. Only the system administrator `odaadmin` has the necessary privileges to run this command.

Example 16-37

```
[odaadmin@oda1 bin]$ odacli revoke-resource-access -ri cfc65e00-5d9e-4c00-
a7a0-f7ea37884e06 -u dcsdb2
Shared access of resource to the user revoked successfully...
[odaadmin@oda1 bin]$ odacli describe-resource-access -ri cfc65e00-5d9e-4c00-
a7a0-f7ea37884e06
```

```
Resource Access details
----------------------------------------------------------------
           Resource ID: cfc65e00-5d9e-4c00-a7a0-f7ea37884e06
         Resource Name: asmtde1
      Resource Location: 53f434fe-598e-46a0-bbdc-a456404da0c9
         Resource Type: ODA_DB
         Primary Owner: odadb1
         Shared Access: odaadmin
             Is Active: true
               Created: November 17, 2021 1:31:55 PM UTC
          Last Updated: November 19, 2021 12:53:43 PM UTC
```

odacli reset-password

Use the `odacli reset-password` command to reset the password for a user, after it is unlocked, in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli reset-password [-d domain_name] [-t tenant_name] [-j] [-h]`

Parameters

Parameter	Description
`--domain-name, -d`	(Optional) Specifies the name of the domain in which the user exists.
`--tenant-name, -t`	(Optional) Specifies the tenancy in which the user exists.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to reset the password for a user in a multi-user access system. The password of a user must be reset either when the user account is locked due to multiple failed login attempts or if the password has expired. The `odaadmin` must first unlock the user account by running the `odacli authorize-user` command. The user can then log in with the temporary password assigned by `odaadmin`, run the `odacli reset-password` command, provide the temporary password, and then specify the new password. You can also reset the password from the Browser User Interface.

Example 16-38

```
[dcsdb2@oda1 bin]$ odacli reset-password
Enter your ODA account password:
Enter new password for your ODA account:
Retype new password for your ODA account:
Password reset for user 'dcsdb2' successful...
```

odacli change-password

Use the `odacli change-password` command to change the password for a user in a multi-user access system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli change-password [-d domain_name] [-t tenant_name] [-j] [-h]`

Parameters

Parameter	Description
`--domain-name, -d`	(Optional) Specifies the name of the domain in which the user exists.
`--tenant-name, -t`	(Optional) Specifies the tenancy in which the user exists.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- This command must be run by the active user who wants to change the password. To reset the password of a locked or expired account, use the `odacli reset-password` command.

Example 16-39

```
[dcsdb2@oda1 bin]$ odacli reset-password
Enter your ODA account password:
Enter new password for your ODA account:
Retype new password for your ODA account:
Password reset for user 'dcsdb2' successful...
```

odacli Apply Patch and Update Commands

Use the commands `odacli update` and `apply patch` to apply patches and update the appliance.

- odacli cleanup-patchrepo
  Use the `odacli cleanup-patchrepo` command to delete obsolete or old patches from the repository.
- odacli create-prepatchreport
  Use the `odacli create-prepatchreport` command to run pre-checks for patching.
- odacli describe-prepatchreport
  Use the `odacli describe-prepatchreport` command to display the pre-check report, with a list of pre-checks run with status and comments.

- odacli list-prepatchreports
  Use the `odacli list-prepatchreports` command to display all pre-check reports, with a list of pre-checks run with status and comments.
- odacli delete-prepatchreport
  Use the `odacli delete-prepatchreport` command to delete any pre-check report.
- odacli list-availablepatches
  Use the `odacli list-availablepatches` command to display all available patches for Oracle Database Appliance.
- odacli describe-component
  Use the `odacli describe-component` command to display the installed version and the latest available version for each component.
- odacli describe-latestpatch
  Use the `odacli describe-latestpatch` command to display a list of the latest supported patch versions for each component.
- odacli list-agentconfig-parameters
  Use the `odacli list-agentconfig-parameters` command to list configuration variables used by the appliance.
- odacli modify-agentconfig-parameters
  Use the `odacli modify-agentconfig-parameters` command to modify configuration variables used by the appliance.
- odacli delete-agentconfig-parameters
  Use the `odacli delete-agentconfig-parameters` command to delete configuration variables used by the appliance.
- odacli update-database
  Use the `odacli update-database` command to update a database to a specific database home with later version, and within the same major release.
- odacli update-dbhome
  Use the `odacli update-dbhome` command to update a specific database home to the latest patch bundle version.
- odacli update-dcsadmin
  Use the `odacli update-dcsadmin` command to update the DCS admin for DCS components upgrade.
- odacli update-dcsagent
  Use the `odacli update-dcsagent` command to update the agent.
- odacli update-dcscomponents
  Use the `odacli update-dcscomponents` command to update the DCS components such as the DCS agent, CLI, and controller.
- odacli update-registry
  Use the `odacli update-registry` command to update the metadata for the components that are already present in the registry. For database home and grid home, this command updates the patch information, when you apply patches manually.
- odacli update-repository
  Use the `odacli update-repository` command to update the repository with the new Oracle Database Appliance software.
- odacli update-server
  Use the `odacli update-server` command to update the operating system, firmware, Oracle Appliance Kit, Oracle Clusterware, and all other infrastructure components.

- odacli update-storage
  Use the `odacli update-storage` command to update the storage.
- odacli update-server-postcheck
  Use the `odacli update-server-postcheck` command to perform deployment health checks after upgrading the oeprating system from Oracle Linux 6 to Oracle Linux 7.
- odaadmcli orachk
  Use the `odaadmcli orachk` command to check configuration settings for Oracle Database Appliance components.

odacli cleanup-patchrepo

Use the `odacli cleanup-patchrepo` command to delete obsolete or old patches from the repository.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli cleanup-patchrepo [-nn node_name] [-cl clones] [-v version [-comp
component] [-l local]
```

Parameters

Parameter	Description
`--clones`, `-cl`	Specifies the option to clean up clone files (DB and GI) from the repository `/opt/oracle/oak/pkgrepos/orapkgs/clones/`.
`--version`, `-v`	Defines the Oracle Database Appliance release you want to delete. You cannot delete the initial or current release files. For example, 12.2.1.4.0.
`--component`, `-comp`	A comma-separated list of components as `{DB,GI}` to cleanup for a specific Oracle Database Appliance patch version. The default is both `{DB,GI}`.
`--local`, `-l`	(Optional) Cleans up the repository on the local node.
`--node-name`, `-nn`	(Optional) Cleans up the repository on the specified node names.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

- Use the command to free up space in the patch repository.
- This command does not delete the current patches or latest installed patch repository.

Example 16-40 Deleting RDBMS and GI components for a specific release

```
# odacli cleanup-patchrepo -cl -comp db,gi -v 12.2.1.4.0
{
"jobId" : "d915ffc0-c7f6-49cf-8ddd-ab5d2ad9072f",
"status" : "Created",
```

```
"message" : null,
"reports" : ,
"createTimestamp" : "April 17, 2024 06:44:28 AM UTC",
"resourceList" : ,
"description" : "Cleanup patchrepos",
"updatedTime" : "April 17, 2024 06:44:28 AM UTC"
```

Example 16-41 Deleting Clone Files

```
# odacli cleanup-patchrepo -cl
{
"jobId" : "5d8549a2-1a5e-4b4f-9867-c1f671c659c4",
"status" : "Created",
"message" : null,
"reports" : ,
"createTimestamp" : "April 17, 2024 06:43:45 AM UTC",
"resourceList" : ,
"description" : "Cleanup patchrepos",
"updatedTime" : "April 17, 2024 06:43:45 AM UTC"
}
```

odacli create-prepatchreport

Use the `odacli create-prepatchreport` command to run pre-checks for patching.

Patching pre-checks help identify and remedy any problems before patching is attempted, and ensure all components are ready for updates.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli create-prepatchreport [-db] [-dbid database_id] [-dp] [-d][-i
dbhome_id] [-f files] [-l] [-nn node_name] [-os] [-r] [-s] [-sko] [-st] [-to
dest_db_home_id] [-v version] [-j] [-h]
```

Parameters

Parameter	Description
`--database`, `-db`	(Optional) Specifies the database component for running the pre-checks. This option cannot be used with the `-dp` and `-d` options.
`--databaseid`, `-dbid`	(Optional) Specifies the ID of the database to be updated and for which you want to run the pre-checks.
`--datapatch`, `-dp`	(Optional) Specifies the datapatch component for running the pre-checks. This option cannot be used with the `-db` and `-d` options.
`--dbhome`, `-d`	(Optional) Specifies the database home component for running the pre-checks. This option cannot be used with the `-dp` and `-db` options.
`--dbhomeid`, `-i`	(Optional) Specifies the IDs of the database homes to be updated and for which you want to run the pre-checks.

Parameter	Description
--files, -f	(Optional) Specifies files required to update repository, comma-separated if multiple files are required.
--operating-system, -os	(Optional) Specifies the operating system components. Use this option to run operating system upgrade. If you specify -c, then the only option is os. If you specify -c os, then you can also specify -local (optional).
--node-name, -nn	(Optional) Runs patch pre-checks on specified node names.
--local, -l	(Optional) Runs patch pre-checks on local node only.
--operating-system, -os	(Optional) Specify this option to run the patch pre-checks for the operating system components.
--repo, -r	(Optional) Specify this option to run the patch pre-checks for the repository components.
--server, -s	(Optional) Specify this option to run the patch pre-checks for the server components.
--storage, -st	(Optional) Specify this option to run the patch pre-checks for the storage components.
--version, -v	(Optional) Specifies the version for running the pre-checks.
--to-home, -to	(Optional) Specifies the destination Oracle Database Home on the target database for the update. Note that you must use this option with the -db option.
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output. The default is false.

Usage Notes

Use the odacli create-prepatchreport command to generate a pre-check report. Use the --node or --local option to run the patch pre-checks on specific nodes or the local node.

Example 16-42 Creating Pre-Check Report

```
# odacli create-prepatchreport -v 19.25.0.0.0 -s

Job details
----------------------------------------------------------------
                     ID: e07993d5-0a06-44e7-819a-36d708c52bb6
            Description: Patch pre-checks for [OS, ILOM, GI, ORACHKSERVER]
                 Status: Created
                Created: September 20, 2023 1:01:59 AM PDT
                Message: Use 'odacli describe-prepatchreport -i
e07993d5-0a06-44e7-819a-36d708c52bb6' to check details of results

Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
----------------------------
```

Example 16-43 Creating Pre-Check Report for Updating Database and Applying Datapatch

```
# odacli describe-prepatchreport -i efa93519-a365-4e9a-9f25-ad3c653cccef

Patch pre-check report
------------------------------------------------------------------------
                 Job ID: efa93519-a365-4e9a-9f25-ad3c653cccef
            Description: Patch pre-checks for [SINGLEDB, ORACHKSINGLEDB]:
Target DB is db10181, Destination DbHome is OraDB19000_home8
                  Status: SUCCESS
                 Created: October 18, 2023 11:19:03 AM PDT
                  Result: All pre-checks succeeded

Node Name
---------------
node1

Pre-Check Status Comments
------------------------------ -------- --------------------------------------
__SINGLEDB__
Is system provisioned Success Verified system is provisioned
Validate dbHomesOnACFS Success User has configured disk group for
configured Database homes on ACFS
Validate Oracle base Success Successfully validated Oracle Base
Evaluate DB clone availability Success Successfully validated clone file
                                          exists
Evaluate DB patching with RHP Success Successfully validated patching DB
                                          with RHP.
Validate command execution Success Validated command execution

__ORACHK__
Running orachk Success Successfully ran Orachk
Validate command execution Success Validated command execution
wandli: Node Name
---------------
node2

Pre-Check Status Comments
------------------------------ -------- --------------------------------------
__SINGLEDB__
Is system provisioned Success Verified system is provisioned
Validate dbHomesOnACFS Success User has configured disk group for
configured Database homes on ACFS
Validate Oracle base Success Successfully validated Oracle Base
Evaluate DB clone availability Success Successfully validated clone file
                                          exists
Evaluate DB patching with RHP Success Successfully validated patching DB
                                          with RHP.
Validate command execution Success Validated command execution

__ORACHK__
Running orachk Success Successfully ran Orachk
Validate command execution Success Validated command execution
wandli: example for applying datapatch,
```

```
[root@oda1 ~]# odacli describe-prepatchreport -i 2cdb0b6e-
ce24-4857-86ab-91480a2ac4e8

Patch pre-check report
------------------------------------------------------------------------
                 Job ID: 2cdb0b6e-ce24-4857-86ab-91480a2ac4e8
            Description: Patch pre-checks for [SINGLEDB, ORACHKSINGLEDB]:
Target DB is season, Destination DbHome is OraDB19000_home6
                  Status: SUCCESS
                 Created: September 16, 2023 10:38:58 PM UTC
                  Result: All pre-checks succeeded

Node Name
---------------
oda1

Pre-Check Status Comments
------------------------------ -------- --------------------------------------
__SINGLEDB__
Is system provisioned Success Verified system is provisioned
Validate dbHomesOnACFS Success User has configured disk group for
configured Database homes on ACFS
Validate Oracle base Success Successfully validated Oracle Base
Evaluate DB clone availability Success Successfully validated clone file
                                          exists
Evaluate DB patching with RHP Success Successfully validated patching DB
                                          with RHP.
Validate command execution Success Validated command execution

__ORACHK__
Running orachk Success Successfully ran Orachk
Validate command execution Success Validated command execution
```

odacli describe-prepatchreport

Use the `odacli describe-prepatchreport` command to display the pre-check report, with a list of pre-checks run with status and comments.

Patching pre-checks help identify and remedy any problems before attempting to patch, and ensure all components are ready for updates.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-prepatchreport -i ID [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	Specifies the Job ID for the pre-check report.

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Use the `odacli describe-prepatchreport` command to display the pre-check report.

The pre-checks report indicates whether the server patching can be rolling or not.

Example 16-44 Patch pre-checks report with rolling mode patching supported when Expander or Controller upgrade is not required

```
[root@node1 bin]# odacli describe-prepatchreport -i 83e41cc6-bd0e-4c53-a9fe-
c3f6b7ecd47b

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  83e41cc6-bd0e-4c53-a9fe-c3f6b7ecd47b
            Description:  Patch pre-checks for [STORAGE]
                 Status:  SUCCESS
                Created:  September 3, 2021 2:51:29 PM UTC
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__STORAGE__
Validate patching tag          Success   Validated patching tag:
19.12.0.0.0.
Patch location validation      Success   Verified patch
location
Patch tag validation           Success   Verified patch
tag
Storage patch tag validation   Success   Verified storage patch
location
Verify ASM disks status        Success   ASM disks are
online
Validate rolling patch         Success   Rolling mode patching allowed
as
                                         there is no expander and
controller

upgrade.
Validate command execution     Success   Validated command
execution

Node Name
---------------
```

```
node2

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__STORAGE__
Validate patching tag          Success   Validated patching tag:
19.12.0.0.0.
Patch location validation      Success   Verified patch
location
Patch tag validation           Success   Verified patch
tag
Storage patch tag validation   Success   Verified storage patch
location
Verify ASM disks status        Success   ASM disks are
online
Validate rolling patch         Success   Rolling mode patching allowed
as
                                         there is no expander and
controller

upgrade.
Validate command execution     Success   Validated command execution
```

Example 16-45 Patch pre-checks report with rolling mode patching not supported when Expander or Controller upgrade is required

```
[root@node1 bin]# odacli describe-prepatchreport -i 4a3bef2b-687e-41f2-b3d7-
cbda79e63877

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  4a3bef2b-687e-41f2-b3d7-cbda79e63877
            Description:  Patch pre-checks for [STORAGE]
                 Status:  SUCCESS
                Created:  September 7, 2021 12:30:57 PM UTC
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__STORAGE__
Validate patching tag          Success   Validated patching tag:
19.12.0.0.0.
Patch location validation      Success   Verified patch
location
Patch tag validation           Success   Verified patch
tag
Storage patch tag validation   Success   Verified storage patch
location
```

```
Verify ASM disks status         Success   ASM disks are
online
Validate rolling patch          Alert     Rolling mode patching not allowed
as
                                          there is expander and controller

upgrade.
Validate command execution      Success   Validated command
execution

Node Name
---------------
node2

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__STORAGE__
Validate patching tag           Success   Validated patching tag:
19.12.0.0.0.
Patch location validation       Success   Verified patch
location
Patch tag validation            Success   Verified patch
tag
Storage patch tag validation    Success   Verified storage patch
location
Verify ASM disks status         Success   ASM disks are
online
Validate rolling patch          Alert     Rolling mode patching not allowed
as
                                          there is expander and controller

upgrade.
Validate command execution      Success   Validated command
execution
```

Example 16-46 Displaying the Patch Pre-Checks Report for Operating System, Oracle Grid Infrastructure, and ORAchk

```
[root@oda1 opt]# odacli describe-prepatchreport -i 20d59005-f7f1-421d-bbb1-
bfc4b5a26293
Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  20d59005-f7f1-421d-bbb1-bfc4b5a26293
            Description:  Patch pre-checks for [OS, ILOM, GI, ORACHKSERVER,
SERVER]
                 Status:  SUCCESS
                Created:  May 13, 2022 7:31:24 AM HKT
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status   Comments
```

```
------------------------------ --------
-------------------------------------
__OS__
Validate supported versions     Success   Validated minimum supported
versions.
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Is patch location available     Success   Patch location is
available.
Verify OS patch                 Success   Verified OS
patch
Validate command execution      Success   Validated command
execution

__ILOM__
Validate ILOM server reachable  Success   Successfully connected with
ILOM
                                          server using public IP and
USB

interconnect
Validate supported versions     Success   Validated minimum supported
versions.
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Is patch location available     Success   Patch location is
available.
Checking Ilom patch Version     Success   Successfully verified the
versions
Patch location validation       Success   Successfully validated
location
Validate command execution      Success   Validated command
execution

__GI__
Validate GI metadata            Success   Successfully validated GI
metadata
Validate supported GI versions  Success   Validated minimum supported
versions.
Validate available space        Success   Validated free space
under /u01
Is clusterware running          Success   Clusterware is
running
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Is system provisioned           Success   Verified system is
provisioned
Validate ASM in online          Success   ASM is
online
Validate kernel log level       Success   Successfully validated the OS
log

level
Validate minimum agent version  Success   GI patching enabled in
current
                                          DCSAGENT
```

```
version
Validate Central Inventory      Success   oraInventory validation
passed
Validate patching locks         Success   Validated patching
locks
Validate clones location exist  Success   Validated clones
location
Validate DB start dependencies  Success   DBs START dependency check
passed
Validate DB stop dependencies   Success   DBs STOP dependency check
passed
Evaluate GI patching            Success   Successfully validated GI
patching
Validate command execution      Success   Validated command
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution

__SERVER__
Validate local patching         Alert     Rolling patching is not possible:
in
                                          IB enabled
environments
Validate command execution      Success   Validated command
execution

Node Name
---------------
node2

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__OS__
Validate supported versions     Success   Validated minimum supported
versions.
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Is patch location available     Success   Patch location is
available.
Verify OS patch                 Success   Verified OS
patch
Validate command execution      Success   Validated command
execution

__ILOM__
Validate ILOM server reachable  Success   Successfully connected with
ILOM
                                          server using public IP and
USB

interconnect
```

```
Validate supported versions     Success   Validated minimum supported
versions.
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Is patch location available     Success   Patch location is
available.
Checking Ilom patch Version     Success   Successfully verified the
versions
Patch location validation       Success   Successfully validated
location
Validate command execution      Success   Validated command
execution

__GI__
Validate GI metadata            Success   Successfully validated GI
metadata
Validate supported GI versions  Success   Validated minimum supported
versions.
Validate available space        Success   Validated free space
under /u01
Is clusterware running          Success   Clusterware is
running
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Is system provisioned           Success   Verified system is
provisioned
Validate ASM in online          Success   ASM is
online
Validate kernel log level       Success   Successfully validated the OS
log

level
Validate minimum agent version  Success   GI patching enabled in
current

                                          DCSAGENT

version
Validate Central Inventory      Success   oraInventory validation
passed
Validate patching locks         Success   Validated patching
locks
Validate clones location exist  Success   Validated clones
location
Validate DB start dependencies  Success   DBs START dependency check
passed
Validate DB stop dependencies   Success   DBs STOP dependency check
passed
Evaluate GI patching            Success   Successfully validated GI
patching
Validate command execution      Success   Validated command
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution
```

```
__SERVER__
Validate local patching         Alert     Rolling patching is not possible:
in
                                          IB enabled
environments
Validate command execution      Success   Validated command execution
```

Example 16-47 Pre-checks report for storage

```
# odacli describe-prepatchreport -i 95887f92-7be7-4865-a311-54318ab385f2

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  95887f92-7be7-4865-a311-54318ab385f2
            Description:  Patch pre-checks for [STORAGE]
                 Status:  SUCCESS
                Created:  May 13, 2022 12:52:37 PM HKT
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__STORAGE__
Validate patching tag          Success   Validated patching tag:
19.15.0.0.0.
Patch location validation      Success   Verified patch
location
Patch tag validation           Success   Verified patch
tag
Storage patch tag validation   Success   Verified storage patch
location
Verify ASM disks status        Success   ASM disks are
online
Validate rolling patch         Success   Rolling mode patching allowed
as
                                         there is no expander and
controller

upgrade.
Validate command execution     Success   Validated command
execution

Node Name
---------------
node2

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__STORAGE__
```

```
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Patch location validation       Success   Verified patch
location
Patch tag validation            Success   Verified patch
tag
Storage patch tag validation    Success   Verified storage patch
location
Verify ASM disks status         Success   ASM disks are
online
Validate rolling patch          Success   Rolling mode patching allowed
as
                                          there is no expander and
controller

upgrade.
Validate command execution      Success   Validated command execution
```

Example 16-48 Pre-checks report for updating database home

```
# odacli describe-prepatchreport -i 6f040374-d271-4711-a7db-e6c04597ad69

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  6f040374-d271-4711-a7db-e6c04597ad69
            Description:  Patch pre-checks for [DB, ORACHKDB]: DbHome is
OraDB19000_home1
                 Status:  SUCCESS
                Created:  May 13, 2022 8:59:48 AM HKT
                 Result:  All pre-checks succeeded

Node Name
---------------
node1

Pre-Check                      Status   Comments
------------------------------ --------
-------------------------------------
__DB__
Validate DB Home ID            Success   Validated DB Home
ID:

d03d5187-1c20-470a-9d51-49a0d6f60772
Validate patching tag          Success   Validated patching tag:
19.15.0.0.0.
Is system provisioned          Success   Verified system is
provisioned
Validate minimum agent version Success   Validated minimum agent
version
Is GI upgraded                 Success   Validated GI is
upgraded
Validate available space for   Success   Validated free space required
under
db                                       /u01/app/
odaorahome
```

```
Validate dbHomesOnACFS          Success   User has configured diskgroup
for
configured                                Database homes on
ACFS
Validate Oracle base            Success   Successfully validated Oracle
Base
Is DB clone available           Success   Successfully validated clone
file

exists
Evaluate DBHome patching with   Success   Successfully validated
updating
RHP                                       dbhome with RHP.  and local
patching
                                          is

possible
Validate command execution      Success   Validated command
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution

Node Name
---------------
node2

Pre-Check                      Status   Comments
------------------------------ --------
--------------------------------------
__DB__
Validate DB Home ID             Success   Validated DB Home
ID:

d03d5187-1c20-470a-9d51-49a0d6f60772
Validate patching tag           Success   Validated patching tag:
19.15.0.0.0.
Is system provisioned           Success   Verified system is
provisioned
Validate minimum agent version  Success   Validated minimum agent
version
Is GI upgraded                  Success   Validated GI is
upgraded
Validate available space for    Success   Validated free space required
under
db                                        /u01/app/
odaorahome
Validate dbHomesOnACFS          Success   User has configured diskgroup
for
configured                                Database homes on
ACFS
Validate Oracle base            Success   Successfully validated Oracle
Base
Is DB clone available           Success   Successfully validated clone
```

```
file

exists
Evaluate DBHome patching with   Success   Successfully validated
updating
RHP                                       dbhome with
RHP.
Validate command execution      Success   Validated command
execution

__ORACHK__
Running orachk                  Success   Successfully ran
Orachk
Validate command execution      Success   Validated command
execution
```

odacli list-prepatchreports

Use the `odacli list-prepatchreports` command to display all pre-check reports, with a list of pre-checks run with status and comments.

Patching pre-checks help identify and remedy any problems before attempting to patch, and ensure all components are ready for updates.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli list-prepatchreports [-h] [-j]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Use the `odacli list-prepatchreports` command to display all pre-check reports.

Example 16-49 Displaying All Patch Pre-Checks Reports

```
# odacli list-prepatchreports
ID   Description  Created                             Status

------------ ---------------------------------- ----------

8a8a14b5-1b5e-4eeb-8ba9-c8136fb4eea1     Patch pre-checks for [DB]: DbHome is
OraDB12102_home1
May 16, 2018 7:00:56 PM PDT         Running

2c9a747d-a452-4e48-bcab-9c7cd9f5f35b     Patch pre-checks for [STORAGE]
```

```
May 16, 2018 6:53:23 PM PDT         Success

f3b9c7c1-3061-4577-848d-645669d71f72     Patch pre-checks for [OS, ILOM, GI]
May 16, 2018 7:00:56 PM PDT         Running
```

odacli delete-prepatchreport

Use the `odacli delete-prepatchreport` command to delete any pre-check report.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli delete-prepatchreport -i Report_ID
```

Parameters

Parameter	Description
`--id, -i`	Describes the ID of the report to be deleted.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

Use the `odacli delete-prepatchreport` command to delete pre-check reports.

Example 16-50 Deleting A Patch Pre-Checks Report

```
# odacli delete-prepatchreport -i 741f635b-7c75-4832-8813-782367f1e6fd
```

odacli list-availablepatches

Use the `odacli list-availablepatches` command to display all available patches for Oracle Database Appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-availablepatches [-h] [-j]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Use the `odacli list-availablepatches` command to display all available patches for Oracle Database Appliance.

Example 16-51 Displaying All Available Patches on Bare Metal System

```
# odacli list-availablepatches
------------------- ------------------------ ------------------------
-----------------------------
ODA Release Version  Supported DB Versions     Available DB Versions
Supported Platforms
------------------- ------------------------ ------------------------
-----------------------------
19.14.0.0.0          21.5.0.0.220118          21.5.0.0.220118          DB
System
                     19.14.0.0.220118         19.14.0.0.220118         Bare
Metal, DB System
                     12.2.0.1.220118          12.2.0.1.220118          Bare
Metal
                     12.1.0.2.220118          12.1.0.2.220118          Bare
Metal

19.11.0.0.0          19.11.0.0.210420         19.11.0.0.210420         Bare
Metal, DB System
                     18.14.0.0.210420         18.14.0.0.210420         Bare
Metal
                     12.2.0.1.210420          12.2.0.1.210420          Bare
Metal
                     12.1.0.2.210420          12.1.0.2.210420          Bare
Metal
```

Example 16-52 Displaying All Available Patches on DB System

```
# odacli list-availablepatches
------------------- ------------------------ ------------------------
-----------------------------
ODA Release Version  Supported DB Versions     Available DB Versions
Supported Platforms
------------------- ------------------------ ------------------------
-----------------------------
19.14.0.0.0          21.5.0.0.220118          21.5.0.0.220118          DB
System
                     19.14.0.0.220118         19.14.0.0.220118         Bare
Metal, DB System
                     12.2.0.1.220118          12.2.0.1.220118          Bare
Metal
                     12.1.0.2.220118          12.1.0.2.220118          Bare
Metal

19.11.0.0.0          19.11.0.0.210420         19.11.0.0.210420         Bare
Metal, DB System
                     18.14.0.0.210420         18.14.0.0.210420         Bare
Metal
                     12.2.0.1.210420          12.2.0.1.210420          Bare
```

```
Metal
                        12.1.0.2.210420           12.1.0.2.210420           Bare
Metal
```

odacli describe-component

Use the `odacli describe-component` command to display the installed version and the latest available version for each component.

If there are multiple DB homes installed on Oracle Database Appliance, then the output displays the version details for each of the installed homes and the databases in each database home. You can use this command to check the component version after applying a patch.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli describe-component [-d][-j][-h][-s][-v][-l][-nn` *node_name*`]`

Parameters

Parameter	Description
`--dbhomes`, `-d`	(Optional) Lists the database home versions and available versions.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.
`--local`, `-l`	(Optional) Describes the components for the local node. Use to display details on the local node of multi-node high availability (HA) systems. This option is not needed to display component details on single-node systems.
`--node-name`, `-nn`	(Optional) Describes the components for a specific node name on high availability (HA) systems. This option is not needed to display component details on single node systems.
`--server`, `-s`	(Optional) Lists the server components and versions and the available versions to which you can patch them.
`--systemversion`, `-v`	(Optional) Displays only the system version. This option is available only on high-availability systems.

Usage Notes

Use the `odacli describe-component` command to get component details. On a multi-node environment, the command provides details across all nodes. Use the `--node` or `--local` option to get component details for a specific node.

The `odacli describe-component` command displays the local and shared storage components and the updates required for them separately.

Example 16-53 Displaying patch details for components for High-Availability Oracle Database Appliance Systems for a release

```
# odacli describe-component
System Version
--------------
23.5.0.0.0

System Node Name
----------------
oak0

Local System Version
--------------------
23.5.0.0.0

Component Installed Version Available Version
---------------------------------------- --------------------
--------------------
OAK 23.5.0.0.0 up-to-date

DCSCONTROLLER 23.5.0.0.0 up-to-date

DCSCLI 23.5.0.0.0 up-to-date

DCSAGENT 23.5.0.0.0 up-to-date

DCSADMIN 23.5.0.0.0 up-to-date

OS 8.10 up-to-date

ILOM 5.1.4.20.r157857 up-to-date

BIOS 84060600 up-to-date

LOCAL CONTROLLER FIRMWARE {
     [c0, c1] E2MU200 up-to-date

     [c4, c5, c6] 26.33.1048 26.37.1014
}

SHARED CONTROLLER FIRMWARE {
     [c2, c3] 23.00.01.00 up-to-date
}

FIRMWAREEXPANDER {
     [c2x0, c3x0] 0310 up-to-date
}
SHARED DISK FIRMWARE {
     [c2d0, c2d1, c2d2, RXA0 up-to-date
     c2d3, c2d4, c2d5,
     c3d0, c3d1, c3d2,
     c3d3, c3d4, c3d5]

     [c2d6, c2d7, c2d8, A7B0 up-to-date
     c2d9, c2d10,
```

```
     c2d11, c2d12,
     c2d13, c2d14,
     c2d15, c2d16,
     c2d17, c2d18,
     c2d19, c2d20,
     c2d21, c2d22,
     c2d23, c3d6, c3d7,
     c3d8, c3d9, c3d10,
     c3d11, c3d12,
     c3d13, c3d14,
     c3d15, c3d16,
     c3d17, c3d18,
     c3d19, c3d20,
     c3d21, c3d22, c3d23]
}

HMP 2.4.9.2.600 up-to-date

System Node Name
----------------
oak1

Local System Version
--------------------
23.5.0.0.0
Component Installed Version Available Version
---------------------------------------- --------------------
--------------------
OAK 23.5.0.0.0 up-to-date

DCSCONTROLLER 23.5.0.0.0 up-to-date

DCSCLI 23.5.0.0.0 up-to-date

DCSAGENT 23.5.0.0.0 up-to-date

DCSADMIN 23.5.0.0.0 up-to-date

OS 8.10 up-to-date

ILOM 5.1.4.20.r157857 up-to-date

BIOS 84060600 up-to-date

LOCAL CONTROLLER FIRMWARE {
     [c0, c1] E2MU200 up-to-date

     [c4, c5, c6] 26.33.1048 26.37.1014
}

SHARED CONTROLLER FIRMWARE {
     [c2, c3] 23.00.01.00 up-to-date
}

FIRMWAREEXPANDER {
     [c2x0, c3x0] 0310 up-to-date
```

```
}
SHARED DISK FIRMWARE {
     [c2d0, c2d1, c2d2, RXA0 up-to-date
     c2d3, c2d4, c2d5,
     c3d0, c3d1, c3d2,
     c3d3, c3d4, c3d5]

     [c2d6, c2d7, c2d8, A7B0 up-to-date
     c2d9, c2d10,
     c2d11, c2d12,
     c2d13, c2d14,
     c2d15, c2d16,
     c2d17, c2d18,
     c2d19, c2d20,
     c2d21, c2d22,
     c2d23, c3d6, c3d7,
     c3d8, c3d9, c3d10,
     c3d11, c3d12,
     c3d13, c3d14,
     c3d15, c3d16,
     c3d17, c3d18,
     c3d19, c3d20,
     c3d21, c3d22, c3d23]
}

HMP 2.4.9.2.600 up-to-date
```

Example 16-54 List DB Home Details

```
# odacli describe-component -d

System Version
--------------
23.5.0.0.0

System Node Name
----------------
oak0

Local System Version
--------------------
23.5.0.0.0

System Node Name
----------------
oak1

Local System Version
--------------------
23.5.0.0.0
```

Example 16-55 List the Server Components and Versions

```
# odacli describe-component -s
[root@oak0 ~]# odacli describe-component -s
```

```
System Version
--------------
23.5.0.0.0

System Node Name
----------------
oak0

Local System Version
--------------------
23.5.0.0.0

Component Installed Version Available Version
---------------------------------------- --------------------
--------------------
OAK 23.5.0.0.0 up-to-date

DCSCONTROLLER 23.5.0.0.0 up-to-date

OS 8.10 up-to-date

DCSCLI 23.5.0.0.0 up-to-date

DCSADMIN 23.5.0.0.0 up-to-date

ILOM 5.1.4.20.r157857 up-to-date

BIOS 84060600 up-to-date
LOCAL CONTROLLER FIRMWARE {
     [c0, c1] E2MU200 up-to-date

     [c4, c5, c6] 26.33.1048 26.37.1014
}

SHARED CONTROLLER FIRMWARE {
     [c2, c3] 23.00.01.00 up-to-date
}

FIRMWAREEXPANDER {
     [c2x0, c3x0] 0310 up-to-date
}

SHARED DISK FIRMWARE {
     [c2d0, c2d1, c2d2, RXA0 up-to-date
     c2d3, c2d4, c2d5,
     c3d0, c3d1, c3d2,
     c3d3, c3d4, c3d5]

     [c2d6, c2d7, c2d8, A7B0 up-to-date
     c2d9, c2d10,
     c2d11, c2d12,
     c2d13, c2d14,
     c2d15, c2d16,
     c2d17, c2d18,
     c2d19, c2d20,
     c2d21, c2d22,
```

```
     c2d23, c3d6, c3d7,
     c3d8, c3d9, c3d10,
     c3d11, c3d12,
     c3d13, c3d14,
     c3d15, c3d16,
     c3d17, c3d18,
     c3d19, c3d20,
     c3d21, c3d22, c3d23]
}
System Node Name
---------------
oak1

Local System Version
--------------------
23.5.0.0.0

Component Installed Version Available Version
---------------------------------------- --------------------
--------------------
OAK 23.5.0.0.0 up-to-date

DCSCONTROLLER 23.5.0.0.0 up-to-date

OS 8.10 up-to-date

DCSCLI 23.5.0.0.0 up-to-date

DCSADMIN 23.5.0.0.0 up-to-date

ILOM 5.1.4.20.r157857 up-to-date

BIOS 84060600 up-to-date

LOCAL CONTROLLER FIRMWARE {
     [c0, c1] E2MU200 up-to-date

     [c4, c5, c6] 26.33.1048 26.37.1014
}

SHARED CONTROLLER FIRMWARE {
     [c2, c3] 23.00.01.00 up-to-date
}

FIRMWAREEXPANDER {
     [c2x0, c3x0] 0310 up-to-date
}
SHARED DISK FIRMWARE {
     [c2d0, c2d1, c2d2, RXA0 up-to-date
     c2d3, c2d4, c2d5,
     c3d0, c3d1, c3d2,
     c3d3, c3d4, c3d5]

     [c2d6, c2d7, c2d8, A7B0 up-to-date
     c2d9, c2d10,
     c2d11, c2d12,
```

```
      c2d13, c2d14,
      c2d15, c2d16,
      c2d17, c2d18,
      c2d19, c2d20,
      c2d21, c2d22,
      c2d23, c3d6, c3d7,
      c3d8, c3d9, c3d10,
      c3d11, c3d12,
      c3d13, c3d14,
      c3d15, c3d16,
      c3d17, c3d18,
      c3d19, c3d20,
      c3d21, c3d22, c3d23]
}
```

odacli describe-latestpatch

Use the `odacli describe-latestpatch` command to display a list of the latest supported patch versions for each component.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli describe-latestpatch [-h]
```

Parameters

Parameter	Description
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output. The default is false.

Example 16-56 Listing the Latest Supported Versions

```
#  odacli describe-latestpatch

ComponentType   availableVersion
--------------- --------------------
gi              18.7.0.0.190716
db              18.7.0.0.190716
db              12.2.0.1.190716
db              12.1.0.2.190716
oak             18.7.0.0.0
asr             18.3.1
ilom            3.0.16.22.f.r100119
ilom            4.0.2.31.r126282
ilom            4.0.4.41.r130359
ilom            4.0.4.37.r130617
ilom            4.0.0.24.r121140
ilom            4.0.4.40.r130348
ilom            4.0.4.38.r130206
```

```
os              6.10
bios            17140300
bios            30150300
bios            25080100
bios            41060300
bios            38130200
bios            39090000
bios            52010400
firmwareexpander 0342
firmwareexpander 001e
firmwareexpander 0306
firmwaredisk    a901
firmwaredisk    0r3q
firmwaredisk    a29a
firmwarecontroller 11.05.03.00
firmwarecontroller qdv1rf30
firmwarecontroller vdv1ry03
firmwarecontroller 20.08.01.14
firmwarecontroller 2.11.1280
firmwarecontroller 13.00.00.00
firmwarecontroller 4.650.00-7176
firmwarecontroller kpyair3q
dcsagent        18.7.0.0.0
firmwaredisk    a7e0
firmwaredisk    a4c0
firmwaredisk    a72a
firmwaredisk    a880
firmwaredisk    a122
firmwaredisk    a38k
firmwaredisk    c122
firmwaredisk    944a
firmwaredisk    9440
firmwaredisk    e12b
firmwaredisk    5g08
firmwaredisk    a902
firmwaredisk    a140
firmwaredisk    sa03
firmwaredisk    0b25
firmwaredisk    sf04
firmwaredisk    c38k
firmwaredisk    a3a0
firmwaredisk    pd51
firmwaredisk    m554
firmwaredisk    0121
firmwaredisk    xc311102
```

odacli list-agentconfig-parameters

Use the `odacli list-agentconfig-parameters` command to list configuration variables used by the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-agentconfig-parameters [-h] [-j] [-n] [-t]
```

Parameters

Parameter	Description
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output. The default is false.
--name, -n	Specifies the name of the parameter.
--type, -t	(Optional) Specifies the configuration value for the given type. The default value is PUBLIC.

Usage Notes

Use the `odacli list-agentconfig-parameters` command to list configuration variables used by the appliance.

Example 16-57 Displaying the HttpProxyPort

```
./odacli list-agentconfig-parameters -n HttpProxyPort

Name      Value      Description      Updated
----------------------------------------------------
HttpProxyPort  Http proxy server port    September 8, 2024 4:14:10 AM UTC
```

Example 16-58 Displaying the agent configuration

```
# odacli list-agentconfig-parameters

Name          Value     Description   Updated
------------------------------
-----------------------------------------------------------------
HttpProxyHost   Http proxy server host  September 8, 2024 4:14:10 AM
UTC
HttpProxyPort   Http proxy server port   September 8, 2024 4:14:10 AM
UTC
HttpsProxyHost   Https proxy server host  September 8, 2024 4:14:10 AM
UTC
HttpsProxyPort  Https proxy server port  September 8, 2024 4:14:10 AM
UTC
```

Example 16-59 Displaying Oracle ASM configuration

```
# odacli list-agentconfig-parameters -t ASM
Name                Value  Description  Updated
------------------------------
```

```
----------------------------------------------------------------
------------------------------------------------------------------------
----------------------------------
asm_power_limit        4  June 22, 2021 12:13:10 AM UTC
```

Related Topics

- Configuring Agent Proxy Settings for Object Store Access
  If the Object Store IP address is accessible only through proxy setup by the Oracle Database Appliance server, then define the proxy setting for the agent, so that the agent can access the Object Store.

odacli modify-agentconfig-parameters

Use the `odacli modify-agentconfig-parameters` command to modify configuration variables used by the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli modify-agentconfig-parameters -n name_param [-adbs] [-a] [-bm] [-c
comment_param] [-dbs dbsys_param] [-d description_param] [-r] [-t type_param]
[-u] [-v value_param] [-j] [-h]
```

Parameters

Parameter	Description
`--append`, `-a`	(Optional) Appends the parameter values. For example, `-n p1 -v v1 -n p2 -v v2 -a`.
`--all-dbsystems`, `-adbs`	(Optional) Dynamic parameter update for all DB systems.
`--bm-only`, `-bm`	(Optional) Dynamic parameter update for bare metal systems only.
`--comment`, `-c`	(Optional) Specifies the comment for the parameter.
`--dbsystem`, `-dbs`	(Optional) DB system name for dynamic parameter update.
`--description`, `-d`	(Optional) Specifies the description of the parameter.
`--name`, `-n`	Specifies the name of the parameter. Provide multiple parameter values in the format `-n p1 -v v1 -n p2 -v v2`.
`--reset`, `-r`	Resets the parameter to the default value. For example: `-n p1 -n p2 -r`.
`--update`, `-u`	Replaces the parameter with the specified value. For example: `-n p1 -v v1 -n p2 -v v2 -u`.

Parameter	Description
`--value, -v`	Specifies the value of the parameter. Provide multiple parameter values in the format `-n p1 -v v1 -n p2 -v v2`.
`--type, -t`	(Optional) Specifies the type of the parameter.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

Use the `odacli modify-agentconfig-parameters` command to modify configuration variables used by the appliance. The supported configuration parameters are `HttpsProxyHost` and `HttpsProxyPort`.

Multiple values can be specified. For High-Availability models, the command sets the parameter values on both nodes in the cluster.

For a DB system, dynamic parameter value specified using the `-dbs` flag has more priority than the parameter value specified using the `-adbs` flag, which in turn has a higher priority than the case where the parameter was updated without using any flag from the bare metal system.

Example 16-60 Setting Multiple Parameters

```
# odacli modify-agentconfig-parameters -n HttpsProxyPort -v 80 -d "Http proxy
port"
-n HttpsProxyPort -v 80 -d "Https proxy port" -u

modify-agentconfig-parameters -n HttpsProxyPort -n HttpProxyPort -r

modify-agentconfig-parameters -n HttpsProxyHost -v 90 -a

modify-agentconfig-parameters -n OSPatchRepos -v
```

Example 16-61 Setting Oracle RDBMS Parameters

```
# odacli modify-agentconfig-parameters -t RDBMS -n processes -v 300 -u
The value of processes will be set to 300 in BM and all the DBVMs.

#  odacli list-agentconfig-parameters -t RDBMS -n processes

Name               Value     Description
Updated
--------------------------------------------------------------------------------
-
processes          300                       September 8, 2024 5:24:45 PM UTC
```

Example 16-62 Updating the RDBMS Parameter for all DB systems in the Bare Metal System

```
# odacli modify-agentconfig-parameters -t RDBMS -n processes -v 400 -u -adbs
```

In the bare metal system, the value of the `processes` parameter is not changed:

```
# odacli list-agentconfig-parameters -t RDBMS -n processes

Name                Value        Description
Updated
------------------------------
------------------------------------------------------
processes           300                          September 8, 2024 12:08:15
AM UTC
```

In all DB systems, the value of `processes` is changed:

```
# odacli list-agentconfig-parameters -t RDBMS -n processes

Name                Value        Description
Updated
------------------------------
------------------------------------------------------
processes           400                          September 8, 2024 12:08:15
AM UTC
```

You can set the value for a specific DB system:

```
# odacli modify-agentconfig-parameters -t RDBMS -n processes -v 700 -u -dbs
DBSYSTEM1
```

Value is changed on `DBSYSTEM1`:

```
# odacli list-agentconfig-parameters -t RDBMS -n processes

Name                Value        Description
Updated
------------------------------
------------------------------------------------------
processes           700                          September 8, 2024 12:08:15
AM UTC
```

odacli delete-agentconfig-parameters

Use the `odacli delete-agentconfig-parameters` command to delete configuration variables used by the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli delete-agentconfig-parameters -n name_param [-adbs] [-bm] [-dbs
dbs_param] [-t type_param] [-j] [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--name, -n`	(Mandatory) Specifies the name of the parameter.
`--type, -t`	(Optional) Deletes the configuration value for the given type.
`--all-dbsystems, -adbs`	(Optional) Deletes the dynamic parameter updated for DB systems, using the `-ad` option in the `odacli modify-agentconfig-parameters` command.
`--bm-only, -bm`	(Optional) Deletes the dynamic parameter only from the bare metal systems.
`--dbsystem, -dbs`	(Optional) Deletes the dynamic parameter for a specific DB system.

Usage Notes

Use the `odacli delete-agentconfig-parameters` command to delete configuration values. For High-Availability models, the command deletes the parameter values on both nodes in the cluster.

Example 16-63 Deleting configuration parameters

```
# odacli delete-agentconfig-parameters -n HttpProxyHost

Job details
----------------------------------------------------------------
                     ID:  c6e123c6-1b76-48f8-973d-63c8dab65545
            Description:  delete configuration parameter HttpProxyHost
                 Status:  Running
                Created:  September 8, 2024 9:45:39 AM PDT
                Message:

Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
```

Example 16-64 Deleting processes configuration parameter

```
# odacli delete-agentconfig-parameters -t rdbms -n processes
odacli list-agentconfig-parameters -t RDBMS -n processes
No result found

Note: Parameters updated using -adbs, -dbs flags should be deleted using the
same corresponding flags.

# odacli delete-agentconfig-parameters -t rdbms -n processes -dbs DBSYSTEM1
```

```
# odacli delete-agentconfig-parameters -t rdbms -n processes -adbs
```

odacli update-database

Use the `odacli update-database` command to update a database to a specific database home with later version, and within the same major release.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-database [-a] [-dp] [-f] [-i db_id] [-imp] [-l] [-n db_name] [-
nn node_name] [-r] [-to db_home_id] [-j] [-h]
```

Parameters

Parameter	Description
`--to-home, -to`	(Optional) Specifies the destination Oracle Database Home ID. The `-to` option is not a mandatory option if you want to apply datapatch using the `odacli update-database` command. The `-to` option is only required when you want to use the `odacli update-database` command to patch a specific database to a specific database home.
`--abort, -a`	(Optional) Aborts the unfinished or failed patching operation.
`--name, -n`	(Optional) Specifies the database name the patch.
`--ignore-missing-patch, -imp`	(Optional) Ignores all missing patches and forces the operation.
`--force, -f`	(Optional) Ignores precheck error and forces the operation.
`--local, -l`	(Optional) Updates the database locally.
`--node-name, -nn`	(Optional) Specifies the node name to be updated.
`--id, -i`	(Optional) Specifies the database ID to be patched.
`--datapatch, -dp`	(Optional) Applies only the datapatch.
`--revert, -r`	(Optional) Reverts the unfinished or failed patching operation.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

- You must run the patching pre-checks at least once before running the `odacli update-database` command. Analyze any failed pre-checks, fix them, and rerun pre-checks. If any

of the failed pre-checks can be ignored, only then run the `odacli update-database` command with the `-f` option.

- The `update-database` command patches a database to a specific database home with later version, and within the same major release.
- To patch a database to a specific database home, the database must be running.
- To patch a database, it must have the status `Configured`. Use the `odacli list-database` command to see a list of configured databases. Patch requests for target databases in any other status are rejected during patching.
- You can patch only the database homes in the `Configured` status. Use the `odacli list-dbhomes` command to see a list of configured database homes. Patch requests for target database homes in any other status are rejected during patching.
- The `--datapatch` option applies the datapatch to the specific database. This option is helpful for databases that have moved to a destination database home without the datapatch. Examples of using the `--datapatch` option are as follows:
  - When you patch the database as a standby database and then switch the database as the primary database.
  - When you must apply additional or one-off data patches after patching your Oracle Database Appliance deployment.

Example 16-65 Updating an Oracle Database

To patch a specific database to a specific database home:

```
# odacli update-database -i e5b60db5-2ee9-4cfc-93ee-179302ee6ebc -to
dc98eaa9-289c-4710-a898-ca6287a07a9c
```

When the database home version is earlier than the target database version, an error is encountered:

```
# odacli update-database -i c05458c7-0f1d-4260-8e62-003b3ec26d74 -to
7ef2449d-3583-47fd-a57a-6ba3fd224119
DCS-10045:Validation error encountered: Target database's version is higher
than destination dbhome, no need to patch.
```

odacli update-dbhome

Use the `odacli update-dbhome` command to update a specific database home to the latest patch bundle version.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli update-dbhome -i dbhome_id -v version [-f] [-imp] [-p] [-l] [-nn
node_name] [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	Defines the Oracle Database Home to update.
`--help-h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.
`--precheck, -p`	(Optional) Analyzes the patch.
`--force, -f`	(Optional) Ignores precheck error and forces the operation.
`--local, -l`	(Optional) Updates the Database Home locally.
`--node-name, -n`	(Optional) Specifies the node name to be updated.
`--version, -v`	Defines the Oracle Database Appliance version to update. For example, 19.23.0.0.0.
`--ignore-missing-patch, -imp`	(Optional) Ignores missing patch error and forces the operation.

Usage Notes

- You must run the patching pre-checks at least once before running the `odacli update-dbhome` command. Analyze any failed pre-checks, fix them, and rerun pre-checks. If any of the pre-checks fail, then run the `odacli update-dbhome` command with the `-f` option.
- The `update-dbhome` command applies the latest release update (RU) for Oracle Database home.
- To be updated with the latest patches, the database must be running.
- Only databases in the `Configured` status are updated. Use the `odacli list-database` command to see a list of configured databases. Databases in any other status are skipped during the update.
- When you use the `--local` option, you can patch in the following way. Note that you can patch the nodes in any order, for example, you can patch the instance on `node1` first, and then the instance on `node0`.
  - Run `--local/--node 0` on `node0`, then `--node 1` on `node0`.
  - Or, run `--local/--node 0` on `node0`, then `-local/--node 1` on `node1`.
  - Or, run `--local/--node` on either of the nodes, and then update the database home without using the `--local/--node` on either of the nodes.

Example 16-66 Updating an Oracle Database Home

To apply the latest patch bundle to update an Oracle Database:

```
# odacli update-dbhome -i ad6c7326-e460-411e-94df-230dedbef743 -v 19.25.0.0.0
```

odacli update-dcsadmin

Use the `odacli update-dcsadmin` command to update the DCS admin for DCS components upgrade.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-dcsadmin -v version  [-j] [-h]
```

Parameters

Parameter	Description
`--version, -v`	Defines the Oracle Database Appliance version after update.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

The `update-dcsadmin` command sets up the `dcsadmin` and other components. Run the `update-dcsadmin` command only after you update the DCS agent.

Example 16-67 Updating DCS Admin

```
# odacli update-dcsadmin -v 19.25.0.0.0
{
  "jobId" : "4c238b93-e641-4f9d-9f86-93d0574dd234",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "September 8, 2024 03:11:53 AM UTC",
  "resourceList" : [ ],
  "description" : "DcsAdmin patching",
  "updatedTime" : "September 8, 2024 03:11:53 AM UTC"
}
```

odacli update-dcsagent

Use the `odacli update-dcsagent` command to update the agent.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-dcsagent -v version  [-j] [-h]
```

Parameters

Parameter	Description
`--version`, `-v`	Defines the Oracle Database Appliance version after update.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

The `update dcsagent` command updates the `dcs-agent` to the RPM package that is located in the *location* and restarts the `dcs-agent`. After the update is applied, the agent automatically restarts. It will take a few minutes to reconnect to the `dcs-agent`. Wait until the agent shuts down and completes restarting before performing any tasks.

> **Note:**
>
> Before updating the agent, ensure that you do not have any jobs running or pending during the update window.

Example 16-68 Updating the Agent

To update the `dcs-agent` to version 19.25:

```
# odacli update-dcsagent -v 19.25
{
  "jobId" : "77e454d3-eb68-4130-a247-7633f8d6192b",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "September 8, 2024 14:09:24 PM CST",
  "description" : "DcsAgent patching",
  "updatedTime" : "September 8, 2024 14:09:24 PM CST"
}
```

odacli update-dcscomponents

Use the `odacli update-dcscomponents` command to update the DCS components such as the DCS agent, CLI, and controller.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-dcscomponents  -v version  [-j] [-h]
```

Parameters

Parameter	Description
`--version`, `-v`	Defines the Oracle Database Appliance version after update.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

The `update-dcscomponents` command sets up the `dcscomponents` such as the DCS agent, CLI, and controller.

Example 16-69 Updating DCS Components

```
# odacli update-dcscomponents -v 19.25.0.0.0
{
  "jobId" : "4c238b93-e641-4f9d-9f86-93d0574dd234",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "September 8, 2024 03:11:53 AM UTC",
  "resourceList" : [ ],
  "description" : "DcsComponents patching",
  "updatedTime" : "September 8, 2024 03:11:53 AM UTC"
}
```

odacli update-registry

Use the `odacli update-registry` command to update the metadata for the components that are already present in the registry. For database home and grid home, this command updates the patch information, when you apply patches manually.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-registry  -n component [-u db_unique_name] [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--component`, `-n`	Defines the Oracle Database Appliance component for the registry update. When you specify `system`, then the components discovered include `gihome, dbnode, sysinstance`. When you specify `all`, then the components discovered include `dbhome, db, dbstorage, asr`. You can also specify the individual component to be refreshed. The individual options include `gihome, dbnode, sysinstance, dbhome, DB, dbstorage, ASR`.
`--db-unique-name`, `-u`	(Optional) Specifies the global unique name for the database. This option is only applicable for components `DB` or `dbstorage`.
`--force`, `-f`	This option is deprecated.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

- Starting with Oracle Database Appliance release 19.18, the `odacli update-registry` command does not require the `-force` option for metadata refresh of a component that already exists in the registry metadata. If the metadata for the component to be updated already exists in the appliance registry metadata, then the command rediscovers the attributes from the environment and updates the system registry.
- The following attribute cannot be discovered with the `odacli update-registry` command:
  - **`DB` component:** `rmanBkupPassword`, `enableEEHA`, `isBcfgInSync`, `dbConsoleEnable`, `cpupool`, `dbClass`, `backupConfigId`, `cloudStorageContainer`, `isAutoBackupDisabled`, `registerOnly`, `entitySource`, `instanceOnly`, `dbShape`, `enableSEHA`.
- Use the `--db-unique-name`, `-u` option to update the appliance registry metadata for a specific database or dbstorage. The `odacli update-registry` command then modifies only the registry metadata for the database or dbstorage associated with the database unique name. For example:

  ```
  odacli update-registry -n db -u db_unique_name
  odacli update-registry -n dbstorage -u db_unique_name
  ```

- The `odacli update-registry` command discovers and updates the metadata in the registry only for databases created using ODACLI commands or the Browser User Interace (BUI). An error is displayed when you run the `odacli update-registry` command to update the metadata for databases that were not created using ODACLI commands or BUI. Hence, for database migration cases, do not use the `odacli update-registry` command. Instead, use the `odacli register-database` command. For example:

  ```
  # odacli update-registry -n db -u provdb
  DCS-10131: Database 'provdb' cannot discover because metadata does not
  exist

  # dcserr 10131
  ```

```
10131, Discover_New_Database_Not_Supported, "Database ''{0}'' cannot
discover because metadata does not exist."
// *Cause: The database did not support discovery of new entities.
// *Action: Use the 'odacli register-database' command.
```

- Similarly, if you run the `odacli update-registry -n dbstorage` command to update the metadata for databases or dbstorage that were not created using ODACLI commands or BUI, then an error is displayed. While migrating a database, if the database runs on the appliance and dbstorage also exists, but the corresponding appliance registry metadata does not exist, then you cannot run the `odacli update-registry -n dbstorage` command to discover the registry metadata. Instead, use the `odacli create-dbstorage` command.

  ```
  # odacli update-registry -n dbstorage -u newstorage
  DCS-10132: Dbstorage 'newstorage' cannot discover because metadata does
  not exist

  # dcserr 10132
  10132, Discover_New_Dbstorage_Not_Supported, "Dbstorage ''{0}'' cannot
  discover because metadata does not exist."
  // *Cause: The database storage did not support discovery of new entities.
  // *Action: Use the 'odacli create-dbstorage' command.
  ```

- The `odacli update-registry` command rediscovers only the following DCS components:
  - System resources : Oracle Grid Infrastructure (`gihome`), Oracle Database node (`dbnode`), System instance (`sysinstance`)
  - Database Home (`dbhome`)
  - Database (`db`)
  - Database storage (`dbstorage`)
  - Oracle ASR (`asr`)

  The `odacli update-registry` command does not rediscover any other components not mentioned in the above list.
- If a TDE database is created without using Oracle Database Appliance tooling and the registry is updated for database components, then this database is added to the DCS metadata. If the TDE Wallet Management of the database is `ODA`, then the TDE password of the database must be updated in the DCS metadata by modifying the database with the same option you use to change the TDE password. You can specify the same password for both the current and new TDE password to ensure that the TDE password is updated in the DCS metadata.
- A report is generated after the `odacli update-registry` command completes running. This report displays the values present before the command was run and the updated values. The report is saved in the location `/opt/oracle/dcs/logs/discoveryReport.txt`. If the command is run again, then the new report is appended to the same file along with the timestamp of the latest run of the command.

Example 16-70 Update Registry Report

```
# odacli update-registry -n system
-----------Begin Discovery at 2023-01-24_12-06-28.0666-------------
**********************
Database Storage Discovery
```

```
**********************
**********************
DbStorage Details : Vredf
**********************
           ================
==========================================
==========================================
           Parameter Name                  Old
Value                                                              New
Value
           ================
==========================================
==========================================
           Database Unique Name
Vredf
Vredf
           Name
Vredf
Vredf
           State
CONFIGURED
CONFIGURED
           Entity Source
null
null
           Volume Name1
datvredf
datvredf
           Volume Type1
Data
Data
           Volume Disk Group1
data
data
           Volume File System1             /u02/app/oracle/oradata/
Vredf                                           /u02/app/oracle/oradata/
Vredf
           Volume Initial Size1
102400
102400
           Volume Name2
orahome_sh
orahome_sh
           Volume Type2
Redo
Redo
           Volume Disk Group2
data
data
           Volume File System2             /u01/app/
odaorahome                                                         /u01/app/
odaorahome
           Volume Initial Size2
204800
204800
           Volume Name3
```

```
reco
reco
          Volume Type3
Reco
Reco
          Volume Disk Group3
reco
reco
          Volume File System3             /u03/app/
oracle/                                                    /u03/app/
oracle/
          Volume Initial Size3
3050496
3050496

================================================================================
============================================================

**********************
DbStorage Location : Vredf
**********************
          ================
====================================
====================================
          Parameter Name                  Old
Value                                              New
Value
          ================
====================================
====================================
          DATA Destination                /u02/app/oracle/oradata/
Vredf                                /u02/app/oracle/oradata/
Vredf
          Database Unique Name
Vredf
Vredf
          Database Id                     6909f458-9bc4-5d7d-7606-
c765cd20b823                         6909f458-9bc4-5d7d-7606-
c765cd20b823
          Database Storage Type
Acfs
Acfs
          Flash Cache Destination
null
null
          Reco Destination                /u03/app/oracle/
fast_recovery_area/                     /u03/app/oracle/
fast_recovery_area/
          Redo Destination                /u04/app/oracle/
redo/                                     /u04/app/oracle/
redo/
          Name
Vredf
Vredf
          State
CONFIGURED
```

```
CONFIGURED

================================================================================
============================================================
================================================================================
============================================================

-----------Begin Discovery at 2023-01-24_12-43-48.0876-------------
**********************
Database Home : OraDB19000_home1
**********************
       ================
========================================
========================================
       Parameter Name                  Old
Value                                                   New Value
       ================
========================================
========================================
       Database Edition
EE                                                      EE
       Database Home Location          /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_1          /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_1
       Database Version
19.18.0.0.230118                                        19.18.0.0.230118



       Entity Source
null                                                    null
       Id                              976643xf-6fa9-7466-
x7ws-391b6agb4d4c                          5bb324e4-361e-4f76-
a40d-080f24d68faf
       Name
OraDB19000_home1                                        OraDB19000_home1
       State
CONFIGURED                                              CONFIGURED
       Updated Time                    2023-01-24
03:33:31.65                                     Tue Jan 24 12:43:41 UTC
2023
       Unified Auditing
TRUE                                                    TRUE

================================================================================
============================================================
**********************
Database Home : OraDB19000_home2
**********************
       ================
========================================
========================================
       Parameter Name                  Old
Value                                                   New Value
       ================
========================================
```

```
========================================
       Database Edition
EE                                                              EE
       Database Home Location          /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_2          /u01/app/odaorahome/oracle/product/19.0.0.0/
dbhome_2
       Database Version
19.18.0.0.230118                                                19.18.0.0.230118
       Entity Source
null                                                            null
       Id
5f3595fa-3509-49a1-8507-36441e12ba46
b790a5f9-9792-421b-931a-fea341384f58
       Name
OraDB19000_home2                                                OraDB19000_home2
       State
CONFIGURED                                                      CONFIGURED
       Updated Time                     2023-01-24
08:21:30.65                                        Tue Jan 24 12:44:10 UTC
2023
       Unified Auditing
TRUE                                                            TRUE

================================================================================
============================================================

**********************
Database : oneDb
**********************
       ================
========================================
=======================================
       Parameter Name                   Old
Value                                                   New Value
       ================
========================================
=======================================
       Id                               24d62501-0b3e-44e5-
bd09-0e49f57560c6                           e3c522ae-7d89-49cc-
bd3c-291cef8fb606
       dbName
oneDb                                                       oneDb
       State
CONFIGURED                                                   CONFIGURED
       BackupConfigId
null                                                         null
       CloudStorageContainer
null                                                         null
       CpuPool
null                                                         null
       dbUniqueName
oneDb0                                                      oneDb0
       Db Character Set
AL32UTF8                                                     AL32UTF8
       DbLanguage
AMERICAN                                                     AMERICAN
```

```
       DbTerritory
AMERICA                                                   AMERICA
       NlsCharacterSet
AL16UTF16                                                 AL16UTF16
       DbClass
Oltp                                                      Oltp
       DbConsoleEnable
false                                                     false
       DbDomainName
null                                                      us.oracle.com
       DbEdition
EE                                                        EE
       DbHomeId                           976643xf-6fa9-7466-
x7ws-391b6agb4d4c                             976643xf-6fa9-7466-
x7ws-391b6agb4d4c
       DBId
3409102472                                                3409102472
       isCdb
true                                                      true
       DbOnFlashStorage
null                                                      false
       DbRedundancy
null                                                      null
       DbRole
PRIMARY                                                   PRIMARY
       DbShape
Odb1                                                      Odb1
       DbStorage
ACFS                                                      ACFS
       DbTargetNodeName()
null                                                      null
       DbTargetNodeNumber()
0                                                         0
       pdbName
PDB1                                                      PDB1
       pdbUserName
null                                                      PDBADMIN
       Database Type
Si                                                        Si
       DbVersion
19.18.0.0.230118                                          19.18.0.0.230118
       enableEEHA
false                                                     false
       enableSEHA
false                                                     false
       enableTDE
false                                                     false
       InstanceOnly
false                                                     false
       isAutoBackupDisabled
true                                                      true
       IsBcfgInSync
null                                                      null
       IsFileConversionComplete
null                                                      null
       Level0BackupDay
```

```
null                                                                    null
       RegisterOnly
null                                                                    null
       RmanBkupPassword
null                                                                    null
       TdePassword
null                                                                    null
       TdeWalletManagement
null                                                                    null

================================================================================
============================================================
```

odacli update-repository

Use the `odacli update-repository` command to update the repository with the new Oracle Database Appliance software.

Prerequisites

Before updating the repository, you must upload the Oracle Database Appliance software to the appliance. If the patch contains more than one zip file, then extract and concatenate the zip files before updating the repository.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To unpack and copy the patch bundle to the correct locations in the file system:

```
odacli update-repository  -f filename [-j] [-h]
```

Parameters

Parameter	Description
`--filename`, `-f`	Defines the zip filename of patch bundle or RDBMS clones downloaded from My Oracle Support. Provide a comma-separated list of absolute file paths of the end user and patch bundles.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.
`--help`,`-h`	(Optional) Displays help for using the command.

Usage Notes

- Before updating the repository, ensure that you do not have any jobs running or pending during the update window.
- Upload the patches to `dom0` on Node0. On multi-node systems, the DCS-Agent copies the patch to Node1 and updates the patch repository on both nodes.

- Starting with Oracle Database Appliance release 19.12, a new Oracle ACFS file system is created to store the Oracle Database and Oracle Grid Infrastructure clone files. The DATA disk group provides a space of 150 GB for the file system. When provisioning a system with Oracle Database Appliance release 19.12, the file system is created after Oracle Grid Infrastructure software is configured. When patching a system with Oracle Database Appliance release 19.12, the file system is created when you run the `odacli update-server` command. If you run the command `odacli update-repository` after this file system is configured during provisioning or patching, then the `odacli update-repository` command fails if Oracle Clusterware is not running. The following error is displayed:

  ```
  DCS-10626:Cannot update the repository as the Clusterware is down. Some
  files need to be stored on an ACFS file system.
  ```

Example 16-71 Updating the Repository

The server patch zip file is located in the `/u01/tmpdir` directory. The following example updates the repository with the latest server patch and then lists the job to verify that the update is successful:

```
# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/oda-sm-19.25.0.0.0-
date-server.zip,/tmp/oda-sm-19.25.0.0.0-date-server.zip
{
  "jobId" : "d3510276-da05-447b-990e-6d30964f8f79",
  "status" : "Created",
  "message" : "/u01/tmpdir/oda-sm-12.2.1.1.0-171031-server.zip",
  "reports" : [ ],
  "createTimestamp" : "September 8, 2024 14:13:45 AM CST",
  "description" : "Repository Update",
  "updatedTime" : "September 8, 2024 14:13:45 AM CST"
}
[root@oak1 tmpdir]# ./odacli list-jobs
# odacli list-jobs
ID                                       Description
Created                                  Status
----------------------------             ----------------
----------------------------             -------
6f27a29a-959f-44e1-b984-7473e3c918ad  Server Patching   September 8, 2024
14:19:05 AM CST    Success
```

odacli update-server

Use the `odacli update-server` command to update the operating system, firmware, Oracle Appliance Kit, Oracle Clusterware, and all other infrastructure components.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-server [-c] -v version [-f][-ng] [-l] [-n node_name] [-j] [-h]
```

Parameters

Parameter	Description
`--version, -v`	(Mandatory) Defines the version to update.
`--component,-c`	(Optional) Component that is to be upgraded. Supported value is OS.
`--force, -f`	(Optional) Ignores precheck error and forces the patching operation.
`--local, -l`	(Optional) Updates the server on the local node.
`--node-name, -nn`	(Optional) Specifies the node name to be updated.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- The `update-server` command applies the patches to various infrastructure components and Oracle Clusterware.

  After the update is applied, the agent automatically restarts. It will take a few minutes to reconnect to the server. Wait until the agent shuts down and completes restarting before performing any tasks. Allow at least two (2) minutes before running the next command.
- You must run the patching pre-checks at least once before running the `odacli update-server` command. Analyze any failed pre-checks, fix them, and rerun pre-checks. If any of the pre-checks fail, then run the `odacli update-server` command with the `-f` option.
- When you use the `--local` option, you can patch in the following way. Note that you can patch the nodes in any order, for example, you can patch the instance on `node1` first, and then the instance on `node0`.
  - Run `--local/--node 0` on `node0`, then `--node 1` on `node0`.
  - Or, run `--local/--node 0` on `node0`, then `-local/--node 1` on `node1`.
  - Or, run `--local/--node` on either of the nodes, and then update the server without using the `--local/--node` on either of the nodes.

> **Note:**
>
> Before updating the server, ensure that you do not have any jobs running or pending during the update window. You must run the `odacli create-prepatchreport` command before you patch your server; otherwise, the `odacli update-server` command fails with an error message prompting you to run the patching pre-checks. You can use the `-f` option to override precheck failures and complete the server patching.

Example 16-72 Updating the Server

Run the `odacli update-server` command to update the server to 19.25.0.0.0. On a high-availability system, run the command on one node. The patches are applied automatically to both nodes.

```
# odacli update-server -v 19.25.0.0.0
{
  "jobId" : "6f27a29a-959f-44e1-b984-7473e3c918ad",
  "status" : "Created",
  "message" : "Success of Server Update may trigger reboot of node after 4-5
minutes.
Please wait till node restart",
  "reports" : [ ],
  "createTimestamp" : "September 8, 2024 14:13:45 PM CST",
  "resourceList" : [ ],
  "description" : "Server Patching",
  "updatedTime" : "September 8, 2024 14:13:45 PM CST"
}
```

odacli update-storage

Use the `odacli update-storage` command to update the storage.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-storage -v version [-h] [-j] [-r]
```

Parameters

Parameter	Description
`--version, -v`	Defines the version to update.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--help, -h`	(Optional) Displays help for using the command.
`--rolling, -r`	(Optional) Enables to patch shared disks in rolling fashion without stopping Oracle Clusterware. The shared disks must be online.

Usage Notes

- If the FIRMWAREEXPANDER Installed Version is lower than or equal to the Available Version, then you cannot use rolling mode to update the storage.
- If the FIRMWARECONTROLLER Installed Version is lower than or equal to the Available Version, then you cannot use rolling mode to update the storage.

Example 16-73 Updating the Storage

```
# odacli update-storage -v 19.25.0.0.0
```

odacli update-server-postcheck

Use the `odacli update-server-postcheck` command to perform deployment health checks after upgrading the oeprating system from Oracle Linux 6 to Oracle Linux 7.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-server-postcheck -v version [-j] [-h]
```

Parameters

Parameter	Description
`--version, -v`	(Required) Specifies the version to update.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--help, -h`	(Optional) Displays help for using the command.

odaadmcli orachk

Use the `odaadmcli orachk` command to check configuration settings for Oracle Database Appliance components.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To perform diagnostic checks of components:

```
odaadmcli orachk [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
Components such as `-a, -acchk, -applypatch, -autostop`	(Optional) Specifies the comma-separated list of components for which you want to collect data. Specify the component to be checked in the command. For example: `# odaadmcli orachk -a`
`--verbose`	Displays detailed message.

Usage Notes

The `odaadmcli orachk` command invokes `orachk` from the directory `opt/oracle.SupportTools/orachk`.

Example 16-74 Running odaadmcli orachk

```
[root@oak bin] # ./odaadmcli orachk
INFO: 2019-09-04 16:41:26: Running orachk under /opt/oracle.SupportTools/
orachk
```

Example 16-75 Running odaadmcli orachk for a component

```
[root@oak bin] # ./odaadmcli orachk -a
INFO: 2019-09-19 10:45:16: Running orachk under /usr/bin/orachk
Checking ssh user equivalency settings on all nodes in cluster for root
```

odacli Appliance Commands

Use the `odacli appliance` commands to perform lifecycle activities for the appliance.

- odacli modify-asmport
  Use the `odacli modify-asmport` command to modify the Oracle ASM listener port number.
- odacli set-credential
  Use the `odacli set-credential` command to change the `oda-admin` or `oda-cliadmin` user credentials on a deployment where multi user access is not enabled.
- odacli create-appliance
  Use the `odacli create-appliance` command in a JSON file format to provision Oracle Database Appliance.
- odacli describe-system
  Use the `odacli describe-system` command to display details about components installed on the system.
- odacli ping-agent
  Use the `odacli ping-agent` command to ping the DCS agent resource.
- odacli list-nodes
  Use the `odacli list-nodes` command to list all nodes in the deployment.
- odacli list-featuretracking
  Use the `odacli list-featuretracking` command to display the latest feature tracking report.

odacli modify-asmport

Use the `odacli modify-asmport` command to modify the Oracle ASM listener port number.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli modify-asmport -p [-j] [-h]
```

Parameters

Parameter	Description
--p, -p	(Mandatorty) Specifies the new port number for the Oracle ASM listener.
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output.

Usage Notes

- You can modify the Oracle ASM listener port number on the bare metal system. All DB systems that you provision after changing the Oracle ASM listener port number use the new Oracle ASM port number.

Example 16-76 Command output details on bare metal system and DB system

```
[root@node1 tmp]# odacli modify-asmport -p 1528

[root@node1 tmp]# odacli describe-job -i 3f182c48-7ff8-4642-af13-e330b668c29e

Job details
----------------------------------------------------------------
                     ID:  3f182c48-7ff8-4642-af13-e330b668c29e
            Description:  Modify ASM listener port
                 Status:  Success
                Created:  July 18, 2024 4:03:05 PM CST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Stop CRS on DB System(s)                 July 18, 2024 4:03:05 PM
CST             July 18, 2024 4:14:44 PM CST             Success
Modify the ASM listener port on BM       July 18, 2024 4:14:44 PM
CST             July 18, 2024 4:14:48 PM CST             Success
Change discovery address for DB Systems  July 18, 2024 4:14:48 PM
CST             July 18, 2024 4:34:17 PM CST             Success
```

After running the command on the bare metal system, the DB system job details are as follows:

```
# Stopping Oracle Clusterware

[root@node1 ~]# odacli describe-job -i 72ed400c-c8ff-4d82-9d8c-b7561d3f3616
```

```
Job details
----------------------------------------------------------------
                     ID:  72ed400c-c8ff-4d82-9d8c-b7561d3f3616
            Description:  Stop Clusterware
                 Status:  Success
                Created:  July 18, 2024 10:14:12 AM CEST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Stop crs with force option               July 18, 2024 10:14:13 AM
CEST           July 18, 2024 10:14:23 AM CEST           Success



# Updating the port

[root@node1 ~]# odacli describe-job -i b728b936-7426-463f-b6d4-b7ab9ff9f399

Job details
----------------------------------------------------------------
                     ID:  b728b936-7426-463f-b6d4-b7ab9ff9f399
            Description:  Modify ASM listener port
                 Status:  Success
                Created:  July 18, 2024 11:32:42 AM CEST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Updating the ASM discovery address in    July 18, 2024 11:32:43 AM
CEST           July 18, 2024 11:32:43 AM CEST           Success
OLR

Starting the clusterware                 July 18, 2024 11:32:43 AM
CEST           July 18, 2024 11:33:44 AM CEST           Success
Updating the ASM discovery address in    July 18, 2024 11:33:44 AM
CEST           July 18, 2024 11:34:44 AM CEST           Success
OCR

Restarting Clusterware                   July 18, 2024 11:34:44 AM
CEST           July 18, 2024 11:37:24 AM CEST           Success
```

Related Topics

- odacli modify-asmport
  Use the `odacli modify-asmport` command to modify the Oracle ASM listener port number.

odacli set-credential

Use the `odacli set-credential` command to change the `oda-admin` or `oda-cliadmin` user credentials on a deployment where multi user access is not enabled.

Syntax

To reset the `oda-admin` or `oda-cliadmin` user credentials in interactive mode:

```
odacli set-credential --username username [-j] [-h]
```

Parameters

Parameter	Description
`--username`, `-u`	User name credentials that you want to reset. The user name can be `oda-admin` or `oda-cliadmin`. The `oda-admin` user credentials allow access to the BUI and the `oda-cliadmin` user credentials allow ODACLI to communicate with the DCS agent internally.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

Only the `root` user can reset the `oda-admin` or `oda-cliadmin` user credentials.

Example 16-77 Resetting the oda-admin Password in Interactive Mode

To reset the `oda-admin` user password to a new password in interactive mode:

```
# odacli set-credential --username oda-admin
Enter 'user' password:
Retype 'user' password:
```

odacli create-appliance

Use the `odacli create-appliance` command in a JSON file format to provision Oracle Database Appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To view help for the `odacli create-appliance` command:

```
odacli create-appliance  -r requestjson [-j] [-h]
```

> **Note:**
>
> The `odacli create-appliance` command only supports a JavaScript Object Notation (JSON) file format. An example JSON files and a readme are available in an appendix in this document and in the `/opt/oracle/dcs/sample` directory.

Parameters

Parameter	Description
`--requestjson, -r`	JSON input for appliance creation.
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

For example JSON files, see the chapter *Creating an Appliance Using a JSON File* in this publication.

odacli describe-system

Use the `odacli describe-system` command to display details about components installed on the system.

File Path

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

To list jobs and view job details and status:

```
odacli describe-system [-h] [-b] [-d] [-j]
```

Parameters

Parameter	Description
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.
`--bom, -b`	(Optional) Displays the bill of materials for the installed components on the appliance.
`--details, -d`	(Optional) Display on the command-line, the details of all installed components on the appliance.

Usage Notes

Do not provide both options `-b` and `-d` at the same time, in the command.

Example 16-78 Example Command to View the Bill of Materials from the Command Line for Bare Metal Deployments

```
# odacli describe-system -b
ODA Components Information
------------------------------
Component Name                Component
Details
--------------
--------------------------------------------------------------------------------
-----------------
NODE                          Name : oda1
                              Domain Name :
                              Time Stamp : April 29, 2020 7:00:12 PM UTC


RPMS                          Installed RPMS : acl-2.2.49-7.el6_9.1.x86_64,
                                               aide-0.14-11.el6.x86_64,
                                               alsa-lib-1.1.0-4.el6.x86_64,
                                               at-3.1.10-49.el6.x86_64,
                                               atk-1.30.0-1.el6.x86_64,
                                               attr-2.4.44-7.el6.x86_64,
                                               audit-2.4.5-6.el6.x86_64,
                                               audit-libs-2.4.5-6.el6.x86_64,
                                               audit-libs-
python-2.4.5-6.el6.x86_64,
                                               augeas-
libs-1.0.0-10.el6.x86_64,

authconfig-6.1.12-23.el6.x86_64,
                                               avahi-
libs-0.6.25-17.el6.x86_64,
                                               b43-openfwwf-5.2-10.el6.noarch,

basesystem-10.0-4.0.1.el6.noarch,
                                               bash-4.1.2-48.el6.x86_64,
                                               bc-1.06.95-1.el6.x86_64,
                                               bind-
libs-9.8.2-0.62.rc1.el6_9.5.x86_64,
                                               bind-
utils-9.8.2-0.62.rc1.el6_9.5.x86_64,

binutils-2.20.51.0.2-5.47.el6_9.1.x86_64,
                                               biosdevname-0.7.2-1.el6.x86_64,
                                               bridge-utils-1.2-10.el6.x86_64,
                                               busybox-1.15.1-21.el6_6.x86_64,
                                               bzip2-1.0.5-7.el6_0.x86_64,
                                               bzip2-
libs-1.0.5-7.el6_0.x86_64,
                                               ca-
certificates-2017.2.14-65.0.1.el6_9.noarch,
                                               cairo-1.8.8-6.el6_6.x86_64,
                                               celt051-0.5.1.3-0.el6.x86_64,

checkpolicy-2.0.22-1.el6.x86_64,
```

```
chkconfig-1.3.49.5-1.el6.x86_64,
                                                    cloog-
ppl-0.15.7-1.2.el6.x86_64,
                                                    compat-libcap1-1.10-1.x86_64,
                                                    compat-libstdc+
+-33-3.2.3-69.el6.x86_64,
                                                    compat-
readline5-5.2-17.1.el6.x86_64,
                                                    compat-sap-c+
+-4.8.2-16.el6.x86_64,
                                                    ConsoleKit-0.4.1-6.el6.x86_64,
                                                    ConsoleKit-
libs-0.4.1-6.el6.x86_64,
                                                    ConsoleKit-
x11-0.4.1-6.el6.x86_64,

coreutils-8.4-46.0.1.el6.x86_64,
                                                    coreutils-
libs-8.4-46.0.1.el6.x86_64,
                                                    cpio-2.10-13.el6.x86_64,
                                                    cpp-4.4.7-18.el6.x86_64,
                                                    cpupowerutils-1.3-2.el6.x86_64,
                                                    cpuspeed-1.5-22.0.1.el6.x86_64,
                                                    cracklib-2.8.16-4.el6.x86_64,
                                                    cracklib-
dicts-2.8.16-4.el6.x86_64,
                                                    crash-7.1.4-1.0.1.el6_7.x86_64,

crda-3.13_2015.10.22-3.el6.x86_64,

createrepo-0.9.9-27.el6_9.noarch,
                                                    cronie-1.4.4-16.el6_8.2.x86_64,
                                                    cronie-
anacron-1.4.4-16.el6_8.2.x86_64,
                                                    crontabs-1.10-33.el6.noarch,
                                                    cryptsetup-
luks-1.2.0-11.el6.x86_64,
                                                    cryptsetup-luks-
libs-1.2.0-11.el6.x86_64,
                                                    cups-
libs-1.4.2-78.el6_9.x86_64,

....
....
....
```

odacli ping-agent

Use the `odacli ping-agent` command to ping the DCS agent resource.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli ping-agent [-j] [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-79

```
# odacli ping-agent
```

odacli list-nodes

Use the `odacli list-nodes` command to list all nodes in the deployment.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli list-nodes [-j] [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-80

```
# odacli list-nodes
```

odacli list-featuretracking

Use the `odacli list-featuretracking` command to display the latest feature tracking report.

File Path

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

Use the `odacli list-featuretracking` command to display the latest feature tracking report.

```
odacli list-featuretracking [-h] [-j]
```

Parameters

Parameter	Description
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-81 Example Command to View the Features Report from the Command Line

```
# odacli list-featuretracking
                                    DCS Feature Tracking Report

Last collection date:   10/26/18 15:35
Report version:         1.0

----------------------------------------------------------------
DCS Feature Usage Summary

                                                          Currently
Detected    Total     Last Usage
Feature Name                                              Used        Usages
Samples           Time
--------------------------------------------------------- ---------
Object Store                                              TRUE
12       12 10/26/18 15:35
Automatic Service Request (ASR)                           FALSE
0       12
Storage Expansion Shelf                                   FALSE
0       12


----------------------------------------------------------------
DCS High Water Mark Statistics

Measured Feature Name                                       High Water
Mark       Category
-----------------------------------------------------------
Maximum usage of CPU                                        8.12
%                DCS Agent
Maximum usage of Memory                                     437.12
MB             DCS Agent
Maximum number of threads
150                   DCS Agent
Maximum number of open file descriptors
257                   DCS Agent
```

```
Maximum number of Databases
1                     Database
Maximum number of EE Databases
1                     Database
Maximum number of SE Databases
0                     Database
Maximum number of CDBs
1                     Database
Maximum number of non-CDBs
0                     Database
Maximum number of RAC Databases
1                     Database
Maximum number of RAC One Databases
0                     Database
Maximum number of Single Instance Databases
0                     Database
Maximum number of PDBs
0                     Database
Maximum number of Datafiles
7                     Database
Maximum number of Database Homes
1                     Database
Maximum number of Flashback Databases
0                     Database
Maximum size of a Database                                  6.40
GB                Database
Minimum Database version
18.0.0.0.0            Database
Maximum Database version
18.0.0.0.0            Database
Maximum number of Backups
12                    Database
Maximum number of NFS Backups
0                     Database
Maximum number of Object Store Backups
12                    Database
Maximum number of Disk Backups
0                     Database
Maximum number of Regular-L0 Backups
0                     Database
Maximum number of Regular-L1 Backups
1                     Database
Maximum number of ArchiveLog Backups
9                     Database
Maximum number of Longterm Backups
2                     Database
Maximum interval between Database Backups                   0 Day(s),
01:01:06    Database
Maximum number of Object Store tenants
1                     Object Store
Maximum number of enabled CPU cores
12                    Server
Maximum number of disks
24                    Storage
Maximum number of ASM Disks Groups
3                     Storage
```

```
Maximum size of ASM Disk Groups                                 11.46
TB              Storage
Maximum usage of ASM Disk Groups                                1.70
%                Storage
Maximum number of ASM Disks Groups with redundancy 'NORMAL'
0                    Storage
Maximum number of ASM Disks Groups with redundancy 'HIGH'
3                    Storage
Maximum number of ASM Disks Groups with redundancy 'EXTERN'
0                    Storage
Maximum number of ASM Disks Groups with redundancy 'FLEX'
0                    Storage
Maximum ASM Disk Groups compatibility
18.0.0.0.0           Storage
Minimum ASM Disk Groups compatibility
18.0.0.0.0           Storage
Maximum number of non-ACFS file systems
4                    Storage
Maximum size of non-ACFS file systems                           98.31
GB              Storage
Maximum usage of non-ACFS file systems                          82.71
%               Storage
Maximum number of ACFS file systems
1                    Storage
Maximum size of ACFS file systems                               5.00
GB               Storage
Maximum usage of ACFS file systems                              12.19
%               Storage
```

odacli Backup and Recovery Commands

Use the odacli backup and recover commands to backup to and restore from Oracle Cloud Infrastructure Object Storage or disk or NFS.

- odacli create-backup
  Use the command `odacli create-backup` to create a Level 0, Level 1, archivelog, or Longterm backup.
- odacli create-backupconfig
  Use the command `odacli create-backupconfig` to create a backup configuration.
- odacli create-objectstoreswift
  Use the command `odacli create-objectstoreswift` to create and store the Oracle credential details required to backup to Oracle Object Store.
- odacli describe-backupconfig
  Use the command `odacli describe-backupconfig` to describe a backup configuration.
- odacli delete-backup
  Use the command `odacli delete-backup` to delete obsolete backups and backup reports.
- odacli delete-backupconfig
  Use the command `odacli delete-backupconfig` to delete a backup configuration.

- odacli describe-backupreport
  Use the command `odacli describe-backupreport` to display details of a specific backup report.
- odacli describe-objectstoreswift
  Use the command `describe-objectstoreswift` to describe a Swift ObjectStore.
- odacli delete-objectstoreswift
  Use the command `odacli delete-objectstoreswift` to delete the credentials for the ObjectStore account.
- odacli describe-schedule
  Use the command `odacli describe-schedule` to display details for a specific schedule.
- odacli irestore-database
  Use the command `odacli irestore-database` to restore a database from one system to other system from an external FRA (NFS) or Object Store location based on a backup report.
- odacli list-backupconfigs
  Use the command `odacli list-backupconfigs` to list all backup configurations.
- odacli list-backupreports
  Use the command `odacli list-backupreports` to display a list of all backup reports.
- odacli list-objectstoreswifts
  Use the command `odacli list-objectstoreswifts` to display a list of credentials for the ObjectStore account.
- odacli list-schedules
  Use the command `odacli list-schedules` to display all of the schedules in the system, including database backup schedules and some internal maintenance schedules.
- odacli list-tdebackupreports
  Use the `odacli list-tdebackupreports` command to view TDE Wallet backup reports.
- odacli recover-database
  Use the `odacli recover-database` command to recover or restore a database from backup.
- odacli restore-archivelog
  Use the `odacli restore-archivelog` command to restore archivelog from one SCN to another SCN, or from one timestamp to another timestamp.
- odacli restore-tdewallet
  Use the `odacli restore-tdewallet` command to restore TDE wallet from the backup location.
- odacli modify-backupconfig
  Use the command `odacli modify-backupconfig` to update a backup configuration.
- odacli modify-objectstoreswift
  Use the command `odacli modify-objectstoreswift` to change the credentials for the ObjectStore account.
- odacli modify-schedule
  Use the command `odacli modify-schedule` to update the schedule for a database, or to disable the database backup schedule.

odacli create-backup

Use the command `odacli create-backup` to create a Level 0, Level 1, archivelog, or Longterm backup.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
  odacli  create-backup [-i Database Resource ID | -n Database Resource Name]
[-h] [-j]
```

Parameters

Parameter	Description
`--backupType, -bt {Regular-L0|Regular-L1|Longterm|archivelog}`	Defines the type of backup. The options are not case sensitive.
`--component, -c {Database|TDEWallet}`	(Optional) Defines the component. The supported options are Database and TDE Wallet. The options are not case-sensitive. The default value is Database. If you specify the value as Database, then a backup of both the database and TDE Wallet is taken.
`--name, -n`	Defines the database name.
`--id, -i`	Defines the Database Resource ID.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.
`--keep-archivelog, -ka`	Keeps archive logs.
`--keepDays, -k`	Defines the Keep Days. For Longterm Backup Type only.
`--tag, -t`	Defines the name of the backup. A tag is alphanumeric, up to 30 characters. Required for Longterm Backup Type. The tag must be wrapped in single quotes when special characters are used.

Usage Notes

- Use the command `odacli create-backup` for a specified Database Resource ID and provide a tag for the backup name. Use up to 30 alphanumeric characters for the backup name tag. Three types of backups are available:
  - Level 0: An RMAN incremental backup that backs up all data blocks in the data files being backed up. An incremental backup at level 0 is identical in content to a full backup, but unlike a full backup, the level 0 backup is part of an incremental backup strategy.
  - Level 1: An RMAN incremental backup that includes only those blocks that have been changed since the "parent" backup was taken. A parent backup can be either a level 0 or a level 1 backup.

  - Longterm: Longterm backups are available when backing up to NFS (External FRA) or Oracle Cloud Infrastructure Object Storage (Oracle Object Storage).
  - Archivelog: This option is used to perform backup of all archivelogs not yet backed up to the backup destination, such as Internal FRA, External FRA (NFS location), or Object Store.
- For TDE-enabled databases with TDE keys stored in Oracle Key Vault, the TDE wallet of the database is not backed up because the wallet is stored outside Oracle Database Appliance.
- The Backup Type must be specified in most cases, when you run the `odacli create-backup` command. If the Backup Component is TDE Wallet, then you need not specify the Backup Type.
- This command creates a Backup Report with a Resource ID. To get the ID, use the command `odacli describe-job -i job_id -j` and look for the `resourceId` attribute in `resourceList`.

Example 16-82 Create a Manual Database Backup

Create a long term backup that is kept for 90 days and named q12018HR.

```
  # odacli create-backup -i Database Resource ID -bt Longterm -c database -k
90 -t q12018HR
```

Example 16-83 Create a Level 0 Database Backup

Create a Level 0 backup named `2017Dec22ProductionLevel0` for resource ID `20576eb1-bc32-4e34-bf97-fda0b60ca15b`

```
# odacli create-backup -i 20576eb1-bc32-4e34-bf97-fda0b60ca15b -bt Regular-L0
-t 2017Dec22ProductionLevel0
```

Example 16-84 Create a Backup of only the TDE Wallet

```
# odacli create-backup -n testdb -c TDEWallet
```

Example 16-85 Create a Backup of the database and TDE Wallet

```
# odacli create-backup -n testdb -bt regular-l0
```

odacli create-backupconfig

Use the command `odacli create-backupconfig` to create a backup configuration.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
  odacli  create-backupconfig  -n backup configuration name -d backup
destination{Disk|ObjectStore|NFS|None} [-c] [-cr] [-h] [-j] [-no-cr] [-o] [-
w] [-f] [-ca] [-dc] [-ec] [-on]
```

Parameters

Parameter	Description
`--backupdestination, -d {Disk|ObjectStore|NFS|None}`	Defines the backup destination. The options are not case sensitive.
`--container, -c`	(Optional) Defines the object store container.
`--crosscheck, -cr`	(Optional) Enable crosscheck.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--name, -n`	Defines the backup configuration name.
`--no-crosscheck, -no-cr`	(Optional) Disable crosscheck.
`--objectstoreswiftId, -o`	(Optional) Defines the swift object store credential ID.
`--tde-container, -f`	(Optional) Defines the container or backup location to backup the TDE Wallet. Must be the absolute path to the parent directory when the backup target is NFS, and must be the container name when the backup target is Objectstore.
`--recoverywindow, -wDisk: {1-14} | ObjectStore: {1-31}`	(Optional) Defines the Recovery Window in days. {1-14} days for Disk and {1-31} days for Object Storage in the cloud.
`--compression-algo, -ca`	(Optional) Defines the compression algorithm `{basic|low|medium|high}` that RMAN must use. `(default: basic(disk/NFS), medium(ObjectStore))`
`--disable-compression, -dc`	(Optional) Disables compression for the backups. Compression is enabled by default.
`--enable-compression, -ec`	(Optional) Enables compression for the backups. Compression is enabled by default.
`--objectstoreswiftName, -on`	(Optional) Defines the Objectstore Swift resource name. Specify the Objectstore Swift resource name or Objectstore Swift ID.

Usage Notes

- To backup a TDE-enabled database, backup configuration with separate TDE Wallet backup location and Database backup location is mandatory.
- To backup a non-TDE database, backup configuration with either single database backup location or backup configuration with both database backup location and TDE Wallet backup location can be used.
- For Disk backup configuration, TDE Wallet backup location is by default, Fast Recovery Area (FRA), but in a separate path from the Database backup path within FRA. Hence, TDE Wallet backup location need not be specified when creating the Disk backup configuration.
- The recovery window that is defined in the backup configuration determines when backups are considered obsolete. The following are guidelines:
  - Disk: 1-14 days
  - Object Storage: 1-31 days

Example 16-86 Create a Backup Configuration for disk

Create a backup configuration named `production` that backs up to disk with a 14 day recovery window.

```
# odacli create-backupconfig -d Disk -n production -w 14
```

Example 16-87 Create a Backup Configuration for NFS location

Create a backup configuration that backs up to NFS with a 2 day recovery window.

```
# odacli create-backupconfig -d NFS -n NfsPolicy1 -c /tmp/Nfsbackps -w 2
```

Example 16-88 Create a Backup Configuration with separate backup locations for database backup and TDE Wallet backup

```
# odacli create-backupconfig -n bcfg1 -d Objectstore -w 1 -c location1-on
oss1 -f location2
```

Example 16-89 Create a Backup Configuration with single backup location for database backup

```
# odacli create-backupconfig -n bcfg2 -d Objectstore -w 1 -c testb2 -on oss2
```

odacli create-objectstoreswift

Use the command `odacli create-objectstoreswift` to create and store the Oracle credential details required to backup to Oracle Object Store.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli create-objectstoreswift -e swift end point URL [-h] [-j] -n Object
Store Swift name -t Object Store tenant name -u Object Store user name
```

Parameters

Parameter	Description
`--endpointurl, -e`	Defines the swift end point URL.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--name, -n`	Defines the Object Store Swift name.
`--tenantname, -t`	Defines the Object Store Swift tenant name.
`--username, -u`	Defines the Object Store Swift user name.

Usage Notes

The command creates and stores the Oracle Cloud Infrastructure Object Storage credential details in the system and stores the password in an encrypted Oracle wallet. You can attach the credentials to one or more backup configurations.

The credentials are validated during Object Store backup configuration creation against the endpoint URL and tenancy.

odacli describe-backupconfig

Use the command `odacli describe-backupconfig` to describe a backup configuration.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
  odacli  describe-backupconfig  [-i backup_configuration_id] [-n name] [-h]
[-j]
```

Parameters

Parameter	Description
`--id, -i`	Specifies the backup configuration identifier (ID).
`--name, -n`	Specifies the backup configuration name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Example 16-90 Describing a Backup Configuration

Describe a backup configuration named `production`

```
# odacli describe-backupconfig -n production
```

odacli delete-backup

Use the command `odacli delete-backup` to delete obsolete backups and backup reports.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
  odacli  delete-backup -i Database Resource ID -n name [-br] [-h] [-j]
```

Parameters

Parameter	Description
`--backupreport, -br`	(Optional) Defines the backup report. To delete a Long Term backup, use a JSON input file for the backupreport.
`--id, -i`	Defines the Database Resource Identifier (ID). To delete obsolete level 0, level 1 and archivelog backups and backup reports, use the database resource ID.
`--name, -n`	Defines the Database resource name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

- Delete older, obsolete level 0 and level 1, archive log backups and backup reports with the Database Resource ID. The recovery window that is defined in the backup configuration determines when backups are considered obsolete.
- Delete long term backups and backup report from Oracle Object Storage or NFS backup location using a JSON file with the `--backupreport` option.
- To locate the database ID, view the databases in the Browser User Interface or use the command `odacli list-databases`.

Example 16-91 Delete a Level 0 or Level 1 Backup

```
# odacli delete-backup -i 20576eb1-bc32-4e34-bf97-fda0b60ca15b
```

Example 16-92 Delete a Long Term Backup Report

Delete a Long Term backup using a JSON input file for the Backup Report. In this example, backupreport.json is the JSON input for the backupreport.

```
# odacli delete-backup -i 20576eb1-bc32-4e34-bf97-fda0b60ca15b -br
backupreport.json
```

odacli delete-backupconfig

Use the command `odacli delete-backupconfig` to delete a backup configuration.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli  delete-backupconfig -n name -i backup configuration id [-h] [-j]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--id, -i`	Defines the Backup Config identifier (ID).
`--name, -n`	Defines the Backup Config name.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

You cannot delete a backup configuration if it is attached to a database.

Example 16-93 Deleting a Backup Configuration

Delete a backup configuration named `production`

```
# odacli delete-backupconfig -n production
```

odacli describe-backupreport

Use the command `odacli describe-backupreport` to display details of a specific backup report.

File Path

`$ORACLE_HOME``/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli  describe-backupreport  [-h] [-j] [-i]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--id, -i`	Defines the backup report ID.

Example 16-94 Backup report for a TDE-enabled database (contains a keystoreType attribute)

```
odacli describe-backupreport -i d2326683-9766-4933-bee1-a2b59c9d7e9e
{
  "id" : "d2326683-9766-4933-bee1-a2b59c9d7e9e",
  "dbResId" : "96a98156-b657-403e-a87c-357c2069c285",
  "tag" : "auto",
  "dbId" : "1851419729",
  "dbName" : "db15",
  "dbUniqueName" : "db15",
```

```
  "backupType" : "Regular-L0",
  "keepDays" : null,
  "backupLocation" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/db",
  "cfBackupHandle" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/db/c-1851419729-20240326-03",
  "spfBackupHandle" : "/nfs_backup/orabackups/dbsfb225975e/database/
1851419729/db15/db/c-1851419729-20240326-03",
  "pitrTimeStamp" : "March 26, 2024 04:15:06 AM PDT",
  "pitrSCN" : "1675444",
  "resetLogsTimeStamp" : "March 26, 2024 03:01:39 AM PDT",
  "resetLogsSCN" : "1575481",
  "oraHomeVersion" : "19.22.0.0.240116",
  "sqlPatches" : "35926646,34774667,35943157",
  "backupLogLoc" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/rmanlog/2024-03-26/rman_backup_auto_2024-03-26_04-14-29.0478.log",
  "tdeWalletLoc" : null,
  "dbConfigLoc" : "/nfs_backup/orabackups/dbsfb225975e/database/1851419729/
db15/dbconfig/2024-03-26/DBCONFIG_auto_2024-03-26_04-15-19.0472.tar.gz",
  "name" : "Backup_Report_db15",
  "createTime" : "March 26, 2024 04:14:26 AM PDT",
  "state" : {
    "status" : "CONFIGURED"
  },
  "updatedTime" : "March 26, 2024 04:14:26 AM PDT",
  "backupReportLogDetail" : "/nfs_backup/orabackups/dbsfb225975e/database/
1851419729/db15/rmandetaillogreport/2024-03-26/
rman_list_backup_detail_auto_2024-03-26_04-15-16.0129.log",
  "dbInfo" : {
    "dbClass" : "OLTP",
    "dbType" : "RAC",
    "dbShape" : "odb2",
    "dbEdition" : "EE",
    "dbStorage" : "ASM",
    "dbRedundancy" : null,
    "pdbName" : null,
    "isCdb" : false
  },
  "dbDataSize" : "2441M",
  "dbRedoSize" : "16403M",
  "rmanBackupPieces" : "/nfs_backup/orabackups/dbsfb225975e/database/
1851419729/db15/backuppieces/2024-03-26/backupPieces_auto_96a98156-b657-403e-
a87c-357c2069c285_20240326041518.json",
  "compressionAlgo" : "BASIC",
  "cpuPool" : null,
  "numberOfCores" : null,
  "keystoreType" : "OKV" <<<< HERE
}
```

Example 16-95 Display Details of a Specific Backup Report

```
#  odacli  describe-backupreport -i 2d82460c-d648-4e75-8c7d-72cc90bc442a
{
  "id" : "2d82460c-d648-4e75-8c7d-72cc90bc442a",
  "dbResId" : "b5fc646e-01a6-4c8b-8286-7633346c4329",
```

```
  "tag" : null,
  "dbId" : "2717054291",
  "dbName" : "ExampleDB",
  "dbUniqueName" : "ExampleDBu",
  "backupType" : "REGULAR-L1",
  "keepDays" : null,
  "backupLocation" : "https://swiftobjectstorage.example.com/v1/dbaasimage/
backupbucket",
  "cfBackupHandle" : "c-2717054291-20180108-04",
  "spfBackupHandle" : "c-2717054291-20180108-04",
  "pitrTimeStamp" : "September 08, 2024 12:43:14 PM UTC",
  "pitrSCN" : "1175058",
  "resetLogsTimeStamp" : "September 08, 2024 09:55:34 AM UTC",
  "resetLogsSCN" : "1112268",
  "oraHomeVersion" : "12.2.0.1.170814 (26723265, 26609817)",
  "sqlPatches" : "25811364,26609817",
  "backupLogLoc" : "https://swiftobjectstorage.example.com/v1/dbaasimage/
backupbucket/scaoda702c1n1/rmanlog/ExampleDBu/2717054291/2018-01-08/
rman_backup_2018-01-08_12-42-41.0545.log",
  "tdeWalletLoc" : null,
  "dbConfigLoc" : "https://swiftobjectstorage.example.com/v1/dbaasimage/
backupbucket/scaoda702c1n1/dbconfig/ExampleDBu/2717054291/2018-01-08/
DBCONFIG_TAG20180108T124407_2018-01-08_12-44-07.0533.tar.gz",
  "name" : "Backup_Report_ExampleDB",
  "createTime" : "January 08, 2018 12:42:08 PM UTC",
  "state" : {
    "status" : "CONFIGURED"
  },
  "updatedTime" : "January 08, 2018 12:44:12 PM UTC",
  "backupReportLogDetail" : "https://swiftobjectstorage.example.com/v1/
dbaasimage/backupbucket/scaoda702c1n1/rmandetaillogreport/ExampleDBu/
2717054291/2018-01-08/rman_list_backup_detail_2018-01-08_12-44-04.0362.log",
  "dbInfo" : {
    "dbClass" : "OLTP",
    "dbType" : "RAC",
    "dbShape" : "odb1",
    "dbEdition" : "EE",
    "dbStorage" : "ASM"
  },
  "dbDataSize" : "1542M",
  "dbRedoSize" : "16403M"
}
```


odacli describe-objectstoreswift

Use the command `describe-objectstoreswift` to describe a Swift ObjectStore.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli  describe-objectstoreswift -i Object Store_Swift_id -n Object
Store_Swift_name  [-h] [-j]
```

Parameters

Parameter	Description
`--id, -i`	Specifies the Object Store Swift identifier (ID).
`--name, -n`	Specifies the Object Store Swift name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Example 16-96 Describing the Swift Object Store

```
# odacli  describe-objectstoreswift -i Object Store Swift id
```

odacli delete-objectstoreswift

Use the command `odacli delete-objectstoreswift` to delete the credentials for the ObjectStore account.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
# odacli  delete-objectstoreswift -i Object Store Swift id -n Object Store
Swift name  [-h] [-j]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--id, -i`	Defines the Object Store Swift identifier (ID).
`--name, -n`	Defines the Object Store Swift name.

Usage Notes

You cannot delete the Object Store credentials if they are attached to a backup configuration.

Example 16-97 Deleting the Oracle Object Store Credentials

```
# odacli  delete-objectstoreswift -i Object Store Swift id
```

odacli describe-schedule

Use the command `odacli describe-schedule` to display details for a specific schedule.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
#  odacli  describe-schedule  [-h] [-j] [-i]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--id, -i`	Defines the schedule with an identifier (ID).

Example 16-98 Display Schedule Details

```
#  odacli  describe-schedule -i scheduleid
```

odacli irestore-database

Use the command `odacli irestore-database` to restore a database from one system to other system from an external FRA (NFS) or Object Store location based on a backup report.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
 odacli irestore-database -r [-tn] [-n] [-bp] [-cl] [-nn] [-co] [-s] [-ro] [-
dr] [-y] [-c] [-oid] [-tf] [-bl] [-t] [-kt] [-ocp] [-tl] [-cp] [-dh] [-dn] [-
de] [-f] [-rd] [-on] [-ha|-no-ha] [-own] [-osc] [-h] [-j]
```

Parameters

Parameter	Description
`--backupPassword(s), -bp`	(Optional) Defines the RMAN password for recovery. You can provide more than one password, but the passwords must be separated by a comma(,).
`--backupReport, -r`	JSON input for a backup report.
`--enable-tde, -t`	Specifies whether to enable Transparent Data Encryption (TDE) which uses software keystore.

Parameter	Description
--keystore-type, -kt	Specifies the type of keystore which stores the TDE master encryption key. The value can be software or okv. The default is software.
--okvclient-path, -ocp	Specifies the absolute path to the okvclient.jar file. Multiple values must be comma-separated. Cannot be used with the --okv-server-config,-osc option.
--dbClass, -cl {EE: OLTP/DSS/IMDB, SE: OLTP}	(Optional) Defines the Database class.
--dbConsoleEnable, -co	(Optional) Enables the Database Console.
--dbShape, -s {odb1,odb2, and so on}	(Optional) Defines the database shape.
--dbStorage, -dr {ACFS|ASM}	(Optional) Defines the database storage. Database Storage {ACFS|ASM} (non case-sensitive). The default is ASM.
--dbType, -y	(Optional) Defines the type of database. For high-availability deployments, the values can be RAC|RACOne|SI, with the default as RAC. For single-node deployments, the value is SI.
--dbEdition, -de	(Optional) Defines the Database Edition (SE or EE).
--databaseHome, -dh	(Optional) Defines the database home.
--dbName, -n	(Optional) Defines the database name.
--dbUniqueName, -u	(Optional) Defines the database unique name.
--dbDomainName, -dn	Defines the database domain name.
--dbrole, -ro	Defines the database role for Oracle Data Guard operations. The options are PRIMARY|STANDBY.
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output. The default is false.
--node-name, -tn	(Optional) Specifies the target host name (applicable only for Oracle RAC One Node Database and single-instance Oracle Database, where supported). This option is available for Oracle Database Appliance high-availability deployments.
--enable-tde, -t	(Optional) Specifies whether you want to enable TDE on the target database.
--tde-backup-location, -tl	(Optional) Specifies the backup location for TDE wallet.
--keystore-type, -kt	Specifies the type of keystore which stores the TDE master encryption key. The value can be software or okv. The default is software.
--okv-server-config, -osc	Specifies the name of the Oracle Key Vault server configuration object which serves as the metadata of the Oracle Key Vault user credentials used by the target database.
--okv-wallet-name, -own	Specifies the name of the wallet of the target database on the Oracle Key Vault server. Use this option only if the wallet is manually created on the Oracle Key Vault server.
--noOfRmanChannels, -c	(Optional) Defines the number of RMAN channels (parallelism) Default: SE Edition: 1, Other Enterprise Editions: 5
--objectStoreId, -oid	Defines the Swift Object Store credential ID.
--objectStoreName, -on	Defines the Swift Object Store resource name.

Parameter	Description
`--backuplocation, -bl`	(Optional) Specifies the NFS or local directory path where backups are available.
`--dbRedundancy, -rd`	Specifies the database redundancy value, that is, `{HIGH|MIRROR}`. To specify the `dbRedundancy` option for an Oracle ASM or ACFS storage-based database, at least one disk group of FLEX redundancy must exist. To specify the `dbRedundancy` option for an Oracle ASM storage database, the `dbVersion` or `dbHomeVersion` must be 12.2 or later. The precedence logic for setting the `dbRedundancy` is: 1. `dbRedundancy` that is specified in the `irestore-database` command. 2. `dbRedundancy` in the backup report. 3. If the disk group redundancy is FLEX, then the default `dbRedundancy` value is MIRROR.
`--associated-networks, -nn`	Specifies the associated network names (in the format networkName1,networkName2,...).
`--dbOnFlashStorage, -f`	(Optional) Specifies whether to enable data on Flash storage.
`--no-dbOnFlashStorage, -no-f`	(Optional) Specifies whether to disable data on Flash storage.
`--disable-ha, -no-ha`	(Optional) Disables High Availability for single-instance Oracle Database 19c and later.
`--enable-ha, -ha`	(Optional) Enables High Availability for single-instance Oracle Database 19c and later.

Usage Notes

> **Caution:**
>
> When you create a database with Transparent Data Encryption (TDE) enabled, with TDE keys stored in software keystore, you are prompted to set a password for the TDE wallet. Provide a strong password for security compliance. Set the password carefully, and ensure that this password is available to you at all times for database management operations. Failure to provide the TDE wallet password when prompted, will cause an error in accessing encrypted data.

- The command `odacli irestore-database` restores a database to a system using the Backup Report that is in the Oracle Object Store or External FRA (NFS) location. You can restore a database from one appliance to another appliance, or to the same appliance after the source database is deleted.
- An Oracle wallet (ObjectStoreSwift credentials) must be created to access the backups in Oracle Object Store. This command performs the environment checks, validation checks, and tasks needed to restore a database to an Oracle Database Appliance system.
- If `-cp` parameter is provided as part of the command, the database is associated with the particular CPU pool.

- If `-cp` parameter is not provided as part of the command, and if the backup report has information about the associated CPU pool, then the command validates the CPU pool specified in the backup report. If the validation fails, then an error message is displayed.
- If the backup report does not have information about any associated CPU pool, then the database is created without being associated with any CPU pool.
- To restore a TDE-enabled database, the TDE wallet password must be provided.
- If source database is TDE enabled, then the target database which was restored will also have TDE enabled.
- Specify the TDE wallet location if the backup report does not have TDE wallet location or if the backup report is inaccessible or unavailable.
- To create a database with role as standby in an Oracle Data Guard configuration on Oracle Database Appliance, you do not need to copy the wallet of the primary database when using Oracle Key Vault keystore for TDE-enabled databases. The same wallet can be specified in the `odacli irestore-database` command.

Example 16-99 Restoring a Database to the Same System

Run the command `odacli irestore-database` with the backup report. Enter the SYS user password when prompted.

```
# odacli irestore-database -r backupreport.json -oid Object Store ID
```

Example 16-100 Restoring a Database to a Different System

To restore to a different system, copy the backup report to the other machine's `\bin` folder (or provide the complete path to the backup report), then run the command `odacli irestore-database` with the backup report. Enter the SYS user password when prompted.

```
# odacli irestore-database -r backupreport.json -oid Object Store ID
```

Example 16-101 Restoring a Database with TDE Wallet backup from a backup report

```
# odacli irestore-database -r backupreport.json -on oss1 -t
Enter SYS user password:
Retype SYS user password:
Enter TDE wallet password:
```

Example 16-102 Restoring a Database with TDE Wallet backup specified on the CLI

```
# odacli irestore-database -r backupreport.json -tl https://
swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paas_test/tde_Backups/
test2-c/database/91276171/tdbacf2/tdewallet/ewallet_2020021714442745.p12 -on
oss2 -p -m
Enter SYS user password:
Retype SYS user password:
Enter TDE wallet password:
```

Example 16-103 Restoring to a standby database when Oracle Data Guard is configured

```
# odacli irestore-database -r backup_report.json -u dbUniqueName -ro STANDBY
```

Example 16-104 Recovering a Database that was associated with a CPU Pool

```
# odacli irestore-database -n irmendb3 -r dbcpupllong.json -on ostore -bp -dh
45156cc5-e768-4817-8b52-2a0a482be18f
Enter SYS user password:
Retype SYS user password:
Enter RMAN backup encryption password:
Do you want to provide another RMAN backup encryption password? [y/n]
(default 'n'): n
{
  "jobId" : "31af6cc8-5e04-459b-8017-271c55b8c99b",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "August 10, 2024 11:20:08 AM UTC",
  "resourceList" : [ ],
  "description" : "Database service recovery with db name: irmendb3",
  "updatedTime" : "August 10, 2024 11:20:08 AM UTC"
}
```

Example 16-105 Recovering a Database that was associated with a CPU Pool

Restore an Oracle single-instance database okvdb without using the Oracle Key Vault server configuration object as follows:

```
odacli irestore-database -r /tmp/bkpreport.json -n okvdb8 -y SI -t -kt OKV -
ocp /tmp/okvclient.jar
Enter SYS and SYSTEM user password:
Retype SYS and SYSTEM user password:
Enter TDE wallet password:

Job details
----------------------------------------------------------------
                     ID:  dce7bde6-7fc4-48ec-891b-649525f1ce4f
            Description:  Database service recovery with DB name: okvdb8
                 Status:  Success
                Created:  August 28, 2024 2:50:46 PM CST
                Message:

Task Name                                Node Name                 Start
Time                               End Time
Status
---------------------------------------- -------------------------
----------------------------------------
---------------------------------------- ----------------
Restore Database                         n1          August 28, 2024 2:51:09
PM CST           August 28, 2024 2:58:11 PM CST           Failure
Restore Database                         n1          August 28, 2024 2:51:10
PM CST           August 28, 2024 2:58:10 PM CST           Failure
Check if cluster ware is running         n1          August 28, 2024 2:51:14
PM CST           August 28, 2024 2:51:14 PM CST           Success
Check if cluster ware is running         n2          August 28, 2024 2:51:14
PM CST           August 28, 2024 2:51:14 PM CST           Success
Creating DbStorage for DbRestore         n1          August 28, 2024 2:51:14
PM CST           August 28, 2024 2:51:21 PM CST           Success
Validating DiskSpace for DATA            n1          August 28, 2024 2:51:14
```

```
PM CST           August 28, 2024 2:51:15 PM CST          Success
Setting up SSH equivalence                n1          August 28, 2024 2:51:15
PM CST           August 28, 2024 2:51:19 PM CST          Success
Configuring user access to ACFS           n1          August 28, 2024 2:51:20
PM CST           August 28, 2024 2:51:20 PM CST          Success
filesystems for
okvdb8st

Audit directory creation                  n1          August 28, 2024 2:51:20
PM CST           August 28, 2024 2:51:21 PM CST          Success
Audit directory creation                  n2          August 28, 2024 2:51:21
PM CST           August 28, 2024 2:51:21 PM CST          Success
Configure TDE for RestoreDb TaskFlow      n1          August 28, 2024 2:51:22
PM CST           August 28, 2024 2:51:39 PM CST          Success
Auxiliary Instance Creation               n2          August 28, 2024 2:51:24
PM CST           August 28, 2024 2:51:36 PM CST          Success
TDE Wallet directory creation             n1          August 28, 2024 2:51:36
PM CST           August 28, 2024 2:51:36 PM CST          Success
Installing OKV Client Software            n2          August 28, 2024 2:51:36
PM CST           August 28, 2024 2:51:37 PM CST          Success
Auto login TDE Wallet creation            n1          August 28, 2024 2:51:37
PM CST           August 28, 2024 2:51:38 PM CST          Success
Password based TDE Wallet open            n1          August 28, 2024 2:51:38
PM CST           August 28, 2024 2:51:39 PM CST          Success
Install Object Store Swift module         n2          August 28, 2024 2:51:58
PM CST           August 28, 2024 2:52:16 PM CST          Success
Restoring Spfile From Casper              n2          August 28, 2024 2:52:16
PM CST           August 28, 2024 2:52:38 PM CST          Success
Customize Db Parameters                   n2          August 28, 2024 2:52:38
PM CST           August 28, 2024 2:52:47 PM CST          Success
Create spfile for restore db              n2          August 28, 2024 2:52:47
PM CST           August 28, 2024 2:52:48 PM CST          Success
Restoring control file                    n2          August 28, 2024 2:52:48
PM CST           August 28, 2024 2:53:11 PM CST          Success
Mounting db                               n2          August 28, 2024 2:53:11
PM CST           August 28, 2024 2:53:35 PM CST          Success
Validating backup for RestoreDB           n2          August 28, 2024 2:53:35
PM CST           August 28, 2024 2:53:41 PM CST          Success
Restoring DB for migration                n2          August 28, 2024 2:53:41
PM CST           August 28, 2024 2:54:26 PM CST          Success
Change DBID and/or DBName                 n2          August 28, 2024 2:54:27
PM CST           August 28, 2024 2:54:28 PM CST          Success
Register Database taskflow                n1          August 28, 2024 2:54:31
PM CST           August 28, 2024 2:57:08 PM CST          Success
Create SPFile in shared loc               n2          August 28, 2024 2:54:31
PM CST           August 28, 2024 2:54:37 PM CST          Success
Delete Local Spfile                       n2          August 28, 2024 2:54:37
PM CST           August 28, 2024 2:54:37 PM CST          Success
Register DB with clusterware              n2          August 28, 2024 2:54:37
PM CST           August 28, 2024 2:55:23 PM CST          Success
Set SysPassword and Create PwFile         n1          August 28, 2024 2:55:23
PM CST           August 28, 2024 2:55:25 PM CST          Success
Enable block change tracking              n2          August 28, 2024 2:55:26
PM CST           August 28, 2024 2:55:28 PM CST          Success
Creating pfile                            n2          August 28, 2024 2:55:28
PM CST           August 28, 2024 2:55:29 PM CST          Success
```

```
Updating db env                           n2           August 28, 2024 2:55:29
PM CST           August 28, 2024 2:55:30 PM CST           Success
Enable DbSizing Template                  n2           August 28, 2024 2:55:30
PM CST           August 28, 2024 2:56:20 PM CST           Success
Create tns entry                          n1           August 28, 2024 2:56:20
PM CST           August 28, 2024 2:56:21 PM CST           Success
Create tns entry                          n2           August 28, 2024 2:56:21
PM CST           August 28, 2024 2:56:22 PM CST           Success
Running datapatch                         n2           August 28, 2024 2:56:22
PM CST           August 28, 2024 2:56:23 PM CST           Success
Set CPU pool                              n1           August 28, 2024 2:56:23
PM CST           August 28, 2024 2:56:23 PM CST           Success
Reset Associated Networks for Database    n2           August 28, 2024 2:57:10
PM CST           August 28, 2024 2:57:13 PM CST           Success
Reset Associated Networks                 n1           August 28, 2024 2:57:13
PM CST           August 28, 2024 2:57:14 PM CST           Success
Set log_archive_dest for Database         n2           August 28, 2024 2:57:14
PM CST           August 28, 2024 2:57:18 PM CST           Success
Setting Database parameter for OKV        n1           August 28, 2024 2:57:18
PM CST           August 28, 2024 2:58:05 PM CST           Success
Enable New Tablespace Encryption          n2           August 28, 2024 2:58:07
PM CST           August 28, 2024 2:58:08 PM CST           Success
Deleting OKV Client Software Jar          n2           August 28, 2024 2:58:09
PM CST           August 28, 2024 2:58:09 PM CST           Success
Copy Pwfile to Shared Storage             n1           August 28, 2024 2:58:09
PM CST           August 28, 2024 2:58:10 PM CST           Success
```

odacli list-backupconfigs

Use the command `odacli list-backupconfigs` to list all backup configurations.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli  list-backupconfigs  [-h] [-j]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Example 16-106 Displaying a List of Backup Configuration

Display a list of backup configurations.

```
# odacli list-backupconfigs
```

odacli list-backupreports

Use the command `odacli list-backupreports` to display a list of all backup reports.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
#  odacli  list-backupreports  [-h] [-j]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Displays a list of all database backup reports generated from the command `odacli create-backup`.

Example 16-107 Display a List of all Backup Reports

```
#  odacli  list-backupreports
```

odacli list-objectstoreswifts

Use the command `odacli list-objectstoreswifts` to display a list of credentials for the ObjectStore account.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
# odacli list-objectstoreswifts [-h] [-j]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.

Example 16-108 Displaying a List of ObjectStore Swift Credentials

```
# odacli list-objectstoreswifts
```

odacli list-schedules

Use the command `odacli list-schedules` to display all of the schedules in the system, including database backup schedules and some internal maintenance schedules.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
#  odacli  list-schedules  [-h] [-j]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.

Usage Notes

The command lists all of the schedules in the system, including database backup schedules and some internal maintenance schedules.

Starting with Oracle Database Appliance release 19.16, DCS software provides a cron job for cleaning up SSH equivalence that was set up by DCS. The SSH cleaner job runs every 30 minutes. If the SSH cleaner job finds that there are currently no DCS jobs which use SSH, then it cleans up SSH keys, and removes passwordless SSH equivalence across the Oracle Database Appliance nodes on high-availability systems. It also deletes any SSH keys (`id_rsa` and `id_rsa.pub`) present in the `/root/.ssh` and `/home/operating_system_user/.ssh` directories. The SSH cleaner job ensures that DCS jobs are thread safe because jobs (threads) do not interleave while operating on shared resources, such as SSH. To retain the SSH equivalence, or the SSH keys, you can disable the SSH cleaner job job using the `odacli modify-schedule` command.

Example 16-109 Display a List of Scheduled Database Backups

Display a list of all scheduled database backups and details.

```
# odacli list-schedules

ID                                       Name
Description
CronExpression                 Disabled
---------------------------------------- -------------------------
-------------------------------------------------
------------------------------ --------
072dceea-9f09-4c1c-a155-6aa4e3d64669     AgentState metastore cleanup
internal agentstateentry metastore maintenance     0 0 0 1/1 * ?
*                false
76e530bf-50db-4598-8b9b-6aa89e58b7d7     bom maintenance              bom
reports generation                            0 0 1 ? * SUN *
false
```

```
7b2a8981-fb88-480f-8280-90d463fcbff8     feature_tracking_job         Feature
tracking job                              0 0 20 ? * WED *
false
af5bc954-c867-4ba4-b587-c9d527cbe07b     Log files Cleanup            Auto
log file purge bases on policy                 0 0 3 1/1 * ? *
false
c4b14c87-e975-43ba-adf3-cb54c72c9cf0     Rebalance Monitoring
Rebalance poll and Power modification              0 0/5 * 1/1 * ?
*              true
d21ccf0d-3341-4030-b204-f3eed892e443     SSHCleanerJob                SSH
cleaner job to clean up stale SSH keys         0 0/30 * 1/1 * ? *
false
```

odacli list-tdebackupreports

Use the `odacli list-tdebackupreports` command to view TDE Wallet backup reports.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-tdebackupreports [-h] [-j]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

The TDE backup report saves the latest TDE Wallet backup location of a TDE-enabled database, when a database backup or a TDE Wallet backup is taken for that database.

Example 16-110 Viewing the TDE backup report

```
# odacli list-tdebackupreports
DbResID                                  OraDbId
BackupLocation
---------------------------------------- --------------------
----------------------------------------
49772c1b-e4e4-401b-a5f0-16c7992d6766     1080131389           https://
swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paastest/
TestLtBackupFailur/test2-c/database/1080131389/tddb3/tdewallet/
ewallet_2020053120350968.p12
f1043fa4-b1ee-461b-84d6-8541e4f6f053     1052183392           /u01/NFS_TDE/
orabackups/test2-c/database/1052183392/tddb4/tdewallet/
ewallet_2020060102471941.p12
944a5613-2a1f-41fb-b7b1-91d35e3e8530     1002460890           /u03/app/oracle/
fast_recovery_area/TDDB5/tdewallet/ewallet_2020060103125231.
```

odacli recover-database

Use the `odacli recover-database` command to recover or restore a database from backup.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
  odacli  recover-database [-i] [-br] [-h] [-j] [-r] [-t] [-p] [-s] [-n] [-
tp] [-tl] [-tz] [-cp]
```

Parameters

Parameter	Description
`--backupReport`, `-br`{Regular-L0|Regular-L1|Longterm}	(Optional) JSON input for a backup report.
`--name`, `-n`	Defines the database name.
`--id`, `-i`	Defines the Database Resource ID.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.
`--tde-backup-location`, `-tl`	(Optional) Specifies the backup location for TDE wallet. If not specified, the latest known TDE backup location is used.
`--recoveryTimeStamp`, `-r`	(Optional) Defines the date and time of the backup. The Recovery Timestamp (in format mm/dd/yyyy hh:mm:ss) is required when the recovery is a point in time recovery (PITR).
`--recoverytype`, `-t` {Latest|PITR|SCN}	(Optional) Defines the recovery type. Do not provide the recovery type if you define the Backup Report.
`--rmanrecoverypassword(s)`, `-p`	(Optional) Defines the password for recovery. You can provide more than one password, but the passwords must be within single quote separated by comma(,).
`--scn`, `-s`	(Optional) Defines the SCN recovery type. Required when the RecoveryType is SCN.
`--cpupool`, `-cp`	(Optional) Specifies the CPU pool name.
`--timeZone`, `-tz`	(Optional) Specifies the recovery time zone, in a format similar to `{area}/{city}` such as `america/los_angeles`. This option is required when the recovery type is a point in time recovery (PITR).

Usage Notes

Recovers a database to the latest, a point in time recovery (PITR), or System Change Number (SCN) as input. You can also recover a database from a Backup Report provided as JSON input file.

This command performs various environment and validation checks in order to attempt to ensure that recovery of database is successful. If non-TDE database backups are in ObjectStore, the RMAN recovery passwords are needed to recover the database. This includes RMAN restore and recovery.

This command always performs a full RMAN database restore and recovery. This command is most useful when there is a complete database loss or when the majority of the database files are lost. If you do not require a full RMAN restore, you can perform a manual recovery. For example, a single data file loss or control file loss.

- If `-cp` parameter is provided as part of the command, recovered database is associated with the particular CPU pool.
- If `-cp` parameter is not provided as part of the command, and if the backup report has information about the associated CPU pool, then the recovered database is also associated with the same CPU pool.
- If the original database does not have any associated CPU pool, then the recovered database is also not associated with any CPU pool.

Example 16-111 Recovering a Database to a Point-in-Time

```
# odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t PITR -r
11/08/2017 12:57:33 -p
```

Example 16-112 Recovering a Database to the latest

```
# odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t Latest -p
```

Example 16-113 Recovering a Database to an SCN

```
# odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t SCN -s
392375947
```

Example 16-114 Restoring a TDE wallet when performing database recovery is not supported for TDE-enabled databases with TDE keys stored in Oracle Key Vault

The following error is displayed:

```
# odacli recover-database -n db15 -tl /nfs_backup/ewallet.p12 -t latest
 DCS-10045:Validation error encountered: Usage of TDE wallet related
parameters is not allowed for a TDE database with OKV keystore type.
```

Example 16-115 Restoring the TDE Wallet while performing database recovery, if TDE wallets are lost

```
# odacli recover-database -n db1 -t latest -tp -tl /u01/NFS_TDE/testbackups/
test2-c/database/91276171/tdbacf2/tdewallet/ewallet_2020021714442745.p12
Enter TDE wallet password:
```

Example 16-116 Recovering a Database that was associated with a CPU Pool

```
# odacli recover-database -t latest -n irmendb4  -cp testcpupool3
{
  "jobId" : "b09f2d3f-2b01-4dbb-9797-122ba786e0e4",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 10, 2023 14:56:57 PM UTC",
  "resourceList" : [ ],
```

```
  "description" : "Create recovery-latest for db : irmendb4",
  "updatedTime" : "May 10, 2023 14:56:57 PM UTC"
}
```

odacli restore-archivelog

Use the `odacli restore-archivelog` command to restore archivelog from one SCN to another SCN, or from one timestamp to another timestamp.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli restore-archivelog -n database_name -i DB_ID -fs from_scn -ts to_scn
-ft from_timestamp -tt to_timestamp [-h] [-j]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the Database Name.
`--id, -i`	(Optional) Specifies the Database resource ID.
`--from-scn, -fs`	(Optional) Specifies the start SCN of archivelog restore.
`--from-timestamp, -ft`	(Optional) Specifies the start time of archivelog restore (in the format mm/dd/yyyy hh:mi:ss).
`--to-scn, -ts`	(Optional) Specifies the end SCN of archivelog restore.
`--to-timestamp, -tt`	(Optional) Specifies the end time of archivelog restore (in the format mm/dd/yyyy hh:mi:ss).
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

You can restore archivelog for all supported database versions and database types for an Oracle Database Appliance release.

Example 16-117 Restoring archivelog

```
# odacli restore-archivelog -fs from_scn -ts to_scn
# odacli restore-archivelog -ft from_timestamp -tt to_timestamp
```

odacli restore-tdewallet

Use the `odacli restore-tdewallet` command to restore TDE wallet from the backup location.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli  restore-tdewallet [-i] [-n] [-tl] [-h] [-j]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the Database Name.
`--id, -i`	Specifies the Database ID.
`--tde-backup-location, -tl`	(Optional) Specifies the backup location for TDE wallet.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

> **Caution:**
>
> When you create a database with Transparent Data Encryption (TDE) enabled, you are prompted to set a password for the TDE wallet. Provide a strong password for security compliance. Set the password carefully, and ensure that this password is available to you at all times for database management operations. Failure to provide the TDE wallet password when prompted, will cause an error in accessing encrypted data.

- This command prompts for TDE wallet password. When prompted, enter the password of the TDE wallet. This password is the one you created or last updated using the modify-database command.
- TDE Wallet backup location is optional. If no TDE Wallet backup location is provided, then the latest TDE Wallet backup of the database known to the system is used for restoring the TDE Wallet.
- This command is not supported for the TDE-enabled databases with TDE keys stored in Oracle Key Vault. The following error is displayed:

```
# odacli restore-tdewallet -n db15
Enter TDE wallet password:
DCS-10144:Restore TDE wallet operation is not supported for TDE databases
with OKV keystore type.
```

Example 16-118 Restoring TDE wallet for a database from a backup location

```
# odacli restore-tdewallet -n testdb -tl /u01/NFS_TDE/testbackups/test2-c/
database/91276171/tdbacf2/tdewallet/ewallet_2020021714442745.p12
Enter TDE wallet password:
```

odacli modify-backupconfig

Use the command `odacli modify-backupconfig` to update a backup configuration.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
  odacli  modify-backupconfig [-i] [-n] [-ca] -d backup destination{Disk|
ObjectStore|None} [-c] [-cr][-no-cr] [-dc] [-ec] [-o][-w]  [-h] [-j]
```

Parameters

Parameter	Description
`--id`, `-i`	Defines the Backup Config ID.
`--name`, `-n`	Defines the Backup Config name.
`--backupdestination`, `-d`{Disk|ObjectStore|None}	Defines the backup destination. The options are not case sensitive. Applicable to System type backup config.
`--compression-algo`, `-ca`{BASIC|LOW|MEDIUM|HIGH}	Defines the Compression Algorithm that RMAN must use. The default is BASIC for Disk and NFS backup and MEDIUM for ObjectStore backup.
`--container`, `-c`	(Optional) Defines the object store container or absolute path to the parent directory of NFS destination for backups.
`--crosscheck`, `-cr`	(Optional) Enable Crosscheck. Applicable to Database type backup config.
`--name`, `-n`	Defines the backup configuration name.
`--no-crosscheck`, `-no-cr`	(Optional) Disable crosscheck.
`--objectstoreswiftid`, `-o`	(Optional) Defines the swift object store credential ID.
`--recoverywindow`, `-w` Disk: {1-14} ObjectStore: {1-30}	(Optional) Defines the Recovery Window in days. {1-14} days for Disk and {1-30} days for Object store. Applicable to Database type backup config.
`--disable-compression`, `-dc`	(Optional) Disables compression for the backups.
`--enable-compression`, `-ec`	(Optional) Enables compression for the backups.
`--objectstoreswiftname`, `-on`	(Optional) Describes the Swift Object Store credential name.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.

Usage Notes

- The recovery window that is defined in the backup configuration determines when backups are considered obsolete. The following are guidelines:
  - Disk: 1-14 days

- ObjectStore in Casper: 1-30 days

Example 16-119 Revise a Backup Configuration for Disk

Create a backup configuration named `production` that backs up to disk with a 14 day recovery window.

```
# odacli modify-backupconfig -d Disk -n production -w 14
```

odacli modify-objectstoreswift

Use the command `odacli modify-objectstoreswift` to change the credentials for the ObjectStore account.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
# odacli  modify-objectstoreswift [-n name] [-i id] [-p swiftpassword] [-u
Object Store user name] [-h] [-j]
```

Parameters

Parameter	Description
`--id`, `-i`	Defines the Object Store Swift identifier (ID).
`--name`, `-n`	Defines the Object Store Swift name.
`--swiftpassword`, `-p`	(Optional) Defines the Object Store Swift password.
`--username`, `-u`	(Optional) Defines the Object Store Swift user name.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Use this command to update the password when it is changed for an ObjectStore account. The command updates the Oracle ObjectStore credential details in the system and stores the password in an encrypted Oracle wallet.

The credentials are validated during the command `odacli modify-backupconfig` with `objectstore` as the destination. The credentials are not validated against endpoint URL and tenancy.

Example 16-120 Changing the Oracle Casper ObjectStore Password

```
# odacli  modify-objectstoreswift -i Object Store Swift id -p
```

Example 16-121 Changing the Oracle ObjectStore User Name

```
# odacli  modify-objectstoreswift -i Object Store Swift id -u Object Store
user name
```

odacli modify-schedule

Use the command `odacli modify-schedule` to update the schedule for a database, or to disable the database backup schedule.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
#  odacli modify-schedule  [-i] [-x cronExpression] [-t description] [-d] [-
e] [-h] [-j]
```

Parameters

Parameter	Description
`--scheduleid, -i`	Defines the schedule with an identifier (ID).
`--cronExpression, -x`	(Optional) Specifies the cron expression. Use 5 fields or 6 fields cron expression only.
`--description, -t`	(Optional) Provides a description for the update schedule.
`--disable, -d`	(Optional) Disables the schedule.
`--enable, -e`	(Optional) Enables a disabled schedule.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Backups incur overhead on the system. When possible, do not schedule backups to run when users are trying to access data.

Starting with Oracle Database Appliance release 19.20, the DCS agent supports only 5 fields cron expressions or 6 fields cron expressions by default, in job schedules. When you patch your deployment from Oracle Database Appliance release 19.19 or earlier, the DCS agent automatically converts all 7 fields cron expression from existing job schedules into an equivalent 6 fields cron expression.

Example 16-122 Modify the Time the Backup Occurs

Edit the cron expression to change the time of scheduled backups for a given schedule ID.

```
#  odacli  modify-schedule -i scheduleid  -x "0 0 13 1/1 * *" -e
```

Example 16-123 Disable Scheduled Database Backups

```
#  odacli  modify-schedule -i scheduleid  -d
```

odacli Oracle Data Guard Commands

Use ODACLI commands to configure and manage Oracle Data Guard in your Oracle Database Appliance deployment.

- odacli configure-dataguard
  Use the command `odacli configure-dataguard` to configure Oracle Data Guard on Oracle Database Appliance.
- odacli deconfigure-dataguard
  Use the command `odacli deconfigure-dataguard` to deconfigure Oracle Data Guard on Oracle Database Appliance.
- odacli describe-dataguardstatus
  Use the command `odacli describe-dataguardstatus` to view Oracle Data Guard status on Oracle Database Appliance.
- odacli failover-dataguard
  Use the command `odacli failover-dataguard` on the standby system to failover the Oracle Data Guard primary database to the standby database.
- odacli list-dataguardstatus
  Use the command `odacli list-dataguardstatus` to verify Oracle Data Guard status on both primary and standby databases.
- odacli reinstate-dataguard
  Use the command `odacli reinstate-dataguard` on the new primary system to reinstate a failed primary database after a failover and convert it to a new standby database.
- odacli register-dataguard
  Use the command `odacli register-dataguard` to register Oracle Data Guard with Oracle Database Appliance.
- odacli switchover-dataguard
  Use the command `odacli switchover-dataguard` on the primary system to reverse the Oracle Data Guard primary and standby database roles on Oracle Database Appliance.

odacli configure-dataguard

Use the command `odacli configure-dataguard` to configure Oracle Data Guard on Oracle Database Appliance.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
# odacli configure-dataguard [-h] [-j] [-i] [-r] [-s]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--requestjson, -r`	(Optional) Specifies the JSON file for creating the appliance.
`--skip-password-copy, -s`	(Optional) Specifies that step to copy the password from primary database to standby database can be skipped.

Usage Notes

If the database orapwd file is not in the typical location, then an error may occur in the configuration process. In such a case, manually copy the orapwd file from the primary database to the standby database in advance, and configure Oracle Data Guard with the `--skip-password-copy` option.

Example 16-124 Configuring Oracle Data Guard interactively on the command line

```
# odacli configure-dataguard
Standby site address: test_domain
BUI username for Standby site (default: oda-admin):
BUI password for Standby site:
TrustStore path for Standby DCS server: /cacerts.a
TrustStore password for Standby DCS server: <TrustStore_Password>
Database name for Data Guard configuration: cpub
Primary database syspassword:
********************************************************************************
*****
Data Guard default settings
Primary site network for Data Guard configuration: Public-network
Standby site network for Data Guard configuration: Public-network
Primary database listener port: 1521
Standby database listener port: 1521
Transport type: ASYNC
Protection mode: MAX_PERFORMANCE
Data Guard configuration name: pcpub_scpub
Active Data Guard: disabled
Do you want to edit this Data Guard configuration? (Y/N, default:N):
The TrustStore of the Standby site should be removed unless it is necessary
for some other purpose. Do you want to delete it? (Y/N, default:Y): n
As a requirement, Standby database's SYS password will be set to Primary
database's after Data Guard configuration. Are you sure you want to proceed
with Data Guard configuration? (Y/N): y
********************************************************************************
*************
Configure Data Guard pcpub_scpub started
********************************************************************************
*************
Step 1: Validate Data Guard configuration request (Primary site)
Description: Validate DG Config Creation
Job ID: bafb8ef4-d3c9-4421-84a1-4c75d73e71a0
```

```
Started August 25, 2020 05:51:20 UTC
Validate create Data Guard configuration request
Finished August 25, 2020 05:51:27 UTC
********************************************************************************
*************
Step 2: Validate Data Guard configuration request (Standby site)
Description: Validate DG Config Creation
Job ID: a7b80880-40ab-4dbf-b8be-22717ae87711
Started August 25, 2020 05:51:27 UTC
Validate create Data Guard configuration request
Finished August 25, 2020 05:51:33 UTC
********************************************************************************
*************
Step 3: Download password file from Primary database (Primary site)
Description: Download orapwd file from Primary database
Started August 25, 2020 05:51:33 UTC
Prepare orapwd file for Primary database cpub
Finished August 25, 2020 05:51:38 UTC
********************************************************************************
*************
Step 4: Upload password file to Standby database (Standby site)
Description: Upload orapwd file to Standby database
Started August 25, 2020 05:51:38 UTC
Write orapwd file to Standby database cpub
Finished August 25, 2020 05:51:47 UTC
********************************************************************************
*************
Step 5: Configure Primary database (Primary site)
Description: DG Config service creation - ConfigurePrimary
Job ID: b6726f76-0a3a-4bb1-8fec-a3eb311531cf
Started August 25, 2020 05:51:47 UTC
Configure host DNS on primary env
Configure Data Guard Tns on primary env
Enable Data Guard related Db parameters for primary env
Enable force logging and archivelog mode in primary env
Enable FlashBac
Configure network parameters for local listener on primary env
Restart listener on primary env
Create services for primary db
Finished August 25, 2020 05:52:10 UTC
********************************************************************************
*************
Step 6: Configure Standby database (Standby site)
Description: DG Config service creation - ConfigureStandby
Job ID: e6b552be-fef0-4cd6-a61d-81292f957a6e
Started August 25, 2020 05:52:10 UTC
Configure Data Guard Tns on standby env
Configure host DNS on standby env
Clear Data Guard related Db parameters for standby env
Enable Data Guard related Db parameters for standby env
Enable force logging and archivelog mode in standby env
Populate standby database metadata
Configure network parameters for local listener on standby env
Reset Db sizing and hidden parameters for ODA best practice
Restart Listener on standby env
Create services for standby db
```

```
Finished August 25, 2020 05:53:40 UTC
********************************************************************************
*************
Step 7: Configure and enable Data Guard (Primary site)
Description: DG Config service creation - ConfigureDg
Job ID: 0b79511f-794a-4bc2-936d-5c651110467f
Started August 25, 2020 05:53:40 UTC
Config and enable Data Guard
Post check Data Guard configuration
Finished August 25, 2020 05:55:06 UTC
********************************************************************************
*************
Step 8: Enable Flashback (Standby site)
Description: DG Config service creation - EnableFlashback
Job ID: b526445e-67df-4c35-a7ca-8420298b8bd2
Started August 25, 2020 05:55:06 UTC
Enable FlashBack
Finished August 25, 2020 05:55:17 UTC
********************************************************************************
*************
Step 9: Re-enable Data Guard (Primary site)
Description: DG Config service creation - ReenableDg
Job ID: 4833133f-da65-44a9-beb7-2c4d347c9120
Started August 25, 2020 05:55:23 UTC
Re-enable Data Guard if inconsistent properties found
Post check Data Guard configuration
Finished August 25, 2020 05:55:29 UTC
********************************************************************************
*************
Step 10: Create Data Guard status (Primary site)
Description: DG Status service creation - NewDgconfig
Job ID: 430d0613-bcd3-4e34-9418-c5f4dfddc052
Started August 25, 2020 05:55:33 UTC
Create Data Guard status
Finished August 25, 2020 05:55:37 UTC
********************************************************************************
*************
Step 11: Create Data Guard status (Standby site)
Description: DG Status service creation - NewDgconfig
Job ID: 93bcdd39-657f-4d7c-b893-c50dc5e708c4
Started August 25, 2020 05:55:40 UTC
Create Data Guard status
Finished August 25, 2020 05:55:43 UTC
********************************************************************************
*************
Configure Data Guard pcpub_scpub completed
********************************************************************************
*************
```

Example 16-125 Configuring Oracle Data Guard using JSON file

```
# odacli configure-dataguard -r configdg_ODA_HA.json
Standby machine address: 192.214.108.19
BUI username for Standby machine (default: oda-admin):
BUI password for Standby machine:
```

```
TrustStore path for Standby DCS server (copy from /opt/oracle/dcs/dcscli/
cacerts on Standby machine to this machine and set file permissions to 400): /
cacerts.a
TrustStore password for Standby DCS server: (can be found in /opt/oracle/dcs/
dcscli/dcscli.conf)
Primary database syspassword:
Standby database syspassword (default: same as primary):
The TrustStore of the Standby machine should be removed unless it is
necessary for some other purpose. Do you want to delete it? (Y/N, default:Y):
n
As a requirement, Standby database's SYS password will be set to Primary
database's after Data Guard configuration. Are you sure you want to proceed?
(Y/N): y
Data Guard configuration for database 'test' started
********************************************************************************
*************
Step 1: Validate Data Guard configuration request (Primary machine)
.....
```

Example 16-126 Sample JSON for configuring Oracle Data Guard on Oracle Database Appliance High-Availability Deployments

```
{
  "name": "pink1_pink7",
  "protectionMode": "MAX_PERFORMANCE",
  "enableFlashback": true,
  "enableActiveDg": false,
  "replicationGroups": [
    {
      "sourceEndPoints": [
        {
          "endpointType": "PRIMARY",
          "hostName": "test_domain1",
          "listenerPort": 1521,
          "databaseUniqueName": "pink1",
          "ipAddress": "192.214.96.214"
        },
    {
      "endpointType": "PRIMARY",
      "hostName": "test_domain2",
      "listenerPort": 1521,
      "databaseUniqueName": "pink1",
      "ipAddress": "192.214.96.215"
    }
      ],
      "targetEndPoints": [
        {
          "endpointType": "STANDBY",
          "hostName": "test_domain1",
          "listenerPort": 1521,
          "databaseUniqueName": "pink7",
          "ipAddress": "192.214.108.19"
        },
        {
          "endpointType": "STANDBY",
```

```
            "hostName": "test_domain2",
            "listenerPort": 1521,
            "databaseUniqueName": "pink7",
            "ipAddress": "192.214.108.20"
          }
        ],
        "transportType": "ASYNC"
      }
    ]
}
```

odacli deconfigure-dataguard

Use the command `odacli deconfigure-dataguard` to deconfigure Oracle Data Guard on Oracle Database Appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli deconfigure-dataguard [-h] [-j] -i [-f]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--id, -i`	Specifies the Oracle Data Guard Configuration ID.
`--force, -f`	Forces the deconfiguration of Oracle Data Guard.

Usage Notes

For deconfiguring Oracle Data Guard, you must specify the Oracle Data Guard Configuration ID. Run the `odacli list-dataguardstatus` command to find the Oracle Data Guard Configuration ID.

Example 16-127 Deconfiguring Oracle Data Guard on the Primary Machine

```
# odacli deconfigure-dataguard -i 8396aac4-f7e7-42e4-a2e8-0cbbc89ce31b
Standby site address: test_domain
BUI username for Standby site (default: oda-admin):
BUI password for Standby site:
TrustStore path for Standby DCS server : /cacerts.a
TrustStore password for Standby DCS server: TrustStore_Password
The TrustStore of the Standby site should be removed unless it is necessary
for some other purpose. Do you want to delete it? (Y/N, default:Y): n
Standby database will be deleted after Data Guard configuration is removed.
Are you sure you want to proceed? (Y/N): y
Deconfigure Dataguard Started
********************************************************************************
```

```
*************
Step 1: Deconfigure Data Guard (Primary site)
Description: Deconfigure DG service
Job ID: 8deb3e4c-09cd-4593-8184-f7706dd2ec8a
Started
Deconfigure Data Guard service
Cleanup broker resources
Finished
********************************************************************************
*************
Step 2: Delete Data Guard status (Primary site)
Description: DG Status service creation - UpdateDgconfig
Job ID: 59c61629-6c7d-474c-8c29-c0a9565d91a3
Started
Update Data Guard status
Finished
********************************************************************************
*************
Step 3: Delete Data Guard status (Standby site)
Description: DG Status service creation - UpdateDgconfig
Job ID: 93dcf4cd-1019-4bcd-984a-d3efeb3fae6e
Started
Update Data Guard status
Finished
********************************************************************************
*************
Step 4: Delete Standby database (Standby site)
Description: Database service deletion with db name: panda with id : 30dd49da-
a54f-4a9a-abbf-d3a4e39abee5
Job ID: 0b1cb921-fd41-4e01-86df-beff3adb6de0
Started
Validate db 30dd49da-a54f-4a9a-abbf-d3a4e39abee5 for deletion
Database Deletion
Unregister Db From Cluster
Kill Pmon Process
Database Files Deletion
Deleting Volume
Delete File Groups of Database panda
Finished
********************************************************************************
*************
Data Guard configuration is removed
```

odacli describe-dataguardstatus

Use the command `odacli describe-dataguardstatus` to view Oracle Data Guard status on Oracle Database Appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli describe-dataguardstatus -i [-h] [-j] [-s]
```

Parameters

Parameter	Description
`--id, -i`	(Mandatory) Specifies the Oracle Data Guard Status ID.
`--synchronous, -s`	(Optional) Obtains the latest Oracle Data Guard status synchronously.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Example 16-128 Viewing Oracle Data Guard configuration status

```
# odacli describe-dataguardstatus -i 5f0ed48a-366d-454c-a389-fe367772dbb7
```

odacli failover-dataguard

Use the command `odacli failover-dataguard` on the standby system to failover the Oracle Data Guard primary database to the standby database.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
# odacli failover-dataguard [-h] [-j] [-i] [-u]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--id, -i`	Specifies the Oracle Data Guard Status ID.
`--databaseUniqueName, -u`	Specifies the unique name of the database to failover to.

Usage Notes

After performing the failover, view the status with the `odacli list-dataguardstatus` command.

Example 16-129 Performing Failover Oracle Data Guard

```
# odacli failover-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u houdg02
```

odacli list-dataguardstatus

Use the command `odacli list-dataguardstatus` to verify Oracle Data Guard status on both primary and standby databases.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli list-dataguardstatus [-s] [-h] [-j]
```

Parameters

Parameter	Description
`--synchronous, -s`	(Optional) Obtains the latest Oracle Data Guard status synchronously.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Example 16-130 Viewing Oracle Data Guard status on primary and standby databases

```
# odacli list-dataguardstatus
```

odacli reinstate-dataguard

Use the command `odacli reinstate-dataguard` on the new primary system to reinstate a failed primary database after a failover and convert it to a new standby database.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli reinstate-dataguard [-h] [-j] [-i] [-u]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--id, -i`	Specifies the Oracle Data Guard Status ID.

Parameter	Description
`--databaseUniqueName, -u`	(Optional) Specifies the unique name of the database to reinstate.

Usage Notes

After completing the reinstate operation, view the status with the `odacli list-dataguardstatus` command.

Example 16-131 Performing reinstate operation on Oracle Data Guard

```
# odacli reinstate-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u houdg02
```

odacli register-dataguard

Use the command `odacli register-dataguard` to register Oracle Data Guard with Oracle Database Appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli register-dataguard [-h] [-j] [-r json_input_file]
```

Parameters

Parameter	Description
`--requestjson, -r`	Specifies the JSON input for creation of Oracle Data Guard configuration in the advanced option.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

odacli switchover-dataguard

Use the command `odacli switchover-dataguard` on the primary system to reverse the Oracle Data Guard primary and standby database roles on Oracle Database Appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli switchover-dataguard [-h] [-j] [-i] [-u]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.
`--id, -i`	Specifies the Oracle Data Guard Status ID.
`--databaseUniqueName, -u`	(Optional) Specifies the unique name of the database to switchover to.

Usage Notes

After performing the switchover, view the status with the `odacli list-dataguardstatus` command.

Example 16-132 Switching over Oracle Data Guard

```
# odacli switchover-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u
houdg02
```

odacli CPU Core Commands

Use the CPU Core commands to enable CPU cores and display current and historical CPU core configurations.

- odacli list-cpucores
  Use the `odacli list-cpucores` command lists the history of core configuration changes in the system.
- odacli describe-cpucore
  Use the `odacli describe-cpucore` command to display the current core configuration and the modification date and time.
- odacli modify-cpucore
  Use the `odacli modify-cpucore` command to enable the number of CPU cores in the system.

odacli list-cpucores

Use the `odacli list-cpucores` command lists the history of core configuration changes in the system.

File Path

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-cpucores  [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Example 16-133 Displaying a List of Cores

```
# odacli list-cpucores

Node  Cores  Modified                       Job Status
----- ------ ------------------------------ ---------------
0     10     July 22, 2016 12:06:08 PM SGT  Configured
0     8      July 25, 2016 9:39:59 AM SGT   Configured
```

odacli describe-cpucore

Use the `odacli describe-cpucore` command to display the current core configuration and the modification date and time.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
 odacli describe-cpucore  [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Example 16-134 Displaying the Current Core Configuration

```
# odacli describe-cpucore

Node  Cores  Modified                       Job Status
----- ------ ------------------------------ ---------------
0      8     July 25, 2016 9:39:59 AM SGT   Configured
```

odacli modify-cpucore

Use the `odacli modify-cpucore` command to enable the number of CPU cores in the system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli modify-cpucore  -c cores [-j] [-h]
```

Parameters

Parameter	Description
`--cores, -c`	Defines the number of cores to be enabled in the system.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- The number of cores you enable must be a multiple of 2.
- After the initial configuration, you cannot reduce the number of cores. You can only increase the number of cores.
- When you run the command, you are prompted to confirm the operation with the following message:

  ```
  Modifying the enabled number of CPU requires a reboot of all nodes in the
  ODA system. Are you sure you want to proceed with this operation? (Y/N):
  ```

- Nodes are restarted after modifying the cores in the BIOS. Note this results in database outage as all the nodes in the cluster are restarted at the same time. Once the system is restarted, the system is updated with the correct number of cores.

Example 16-135 Enabling CPU Cores

The following command enables 8 CPU cores.

```
# odacli modify-cpucore -c 32
Modifying the enabled number of CPU requires a reboot of all nodes in the ODA
system. Are you sure you want to proceed with this operation? (Y/N): y
{
  "jobId" : "1a59d182-79f0-4847-81e6-ab3ebf779096",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "September 8, 2024 10:29:37 AM UTC",
  "resourceList" : [ ],
  "description" : "CPU cores service update",
  "updatedTime" : "September 8, 2024 10:29:37 AM UTC"
}
WARNING: Reducing the number of enabled cores may set offline any of the CPUs
used by a CPU Pool. Thus, please review the status of all the CPU Pools after
this operation using the 'describe-cpupool' command. In case of any CPU from
a CPU Pool was offline please run 'modify-cpupool' to update the list of CPUs
and use only online CPUs.
WARNING: All the nodes in this ODA system will reboot after this operation
```

odacli CPU Pool Commands for Bare Metal and KVM

Use odacli commands to manage resource allocations on Oracle Database Appliance for bare metal and KVM deployments.

CPU pools enable management of CPU resources in both bare metal and KVM environments. There are two types of CPU pools, that is, BM and VM.

Use the BM CPU pool to ensure that the database on a bare metal host runs exclusively on the CPUs specified in the CPU pool.

Use the VM CPU pools to cage and manage CPU resource allocations to workloads for virtual machines. Workloads are isolated by creating CPU pools and assigning or pinning virtual machines to a specific CPU pool. When you pin a virtual machine to a CPU pool, you ensure that the virtual machine uses CPUs in only that CPU pool.

Resources in CPU pools cannot overlap, that is, CPU pools cannot share CPUs.

- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli list-cpupools
  Use the `odacli list-cpupools` command to list all CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli describe-cpupool
  Use the `odacli describe-cpupool` command to describe CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-cpupool
  Use the `odacli modify-cpupool` command to modify a CPU Pool configuration on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli delete-cpupool
  Use the `odacli delete-cpupool` command to delete a CPU Pool configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli list-auditrecords
  Use the `odacli list-auditrecords` command to list the audit records for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli describe-auditrecord
  Use the `odacli describe-auditrecord` command to describe the audit record for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

odacli create-cpupool

Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli create-cpupool -n name -c number_of_cores {-bm|-vm} [-nn node_name] [-s socket]
[-dbs]  [-urc] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the CPU Pool name.
`--cores`, `-c`	Specifies the CPU Pool cores.
`--bm`, `-bm`	(Optional) Specifies the CPU Pool as Bare Metal.
`--vm`, `-vm`	(Optional) Specifies the CPU Pool as Virtual Machine.
`--dbsystem`, `-dbs`	(Optional) Specifies the CPU Pool type as DB System.
`--socket`, `-s`	(Optional) Specifies the socket to use. If not specified, then use all available sockets.
`--use-reserved-cores`, `-urc`	(Optional) Specifies to use reserved CPU cores.
`--node-name`, `-nn`	(Optional) Specifies the node where the CPU Pool will be created. On High-Availability deployment, if this option is not specified, then the CPU pool is created on both nodes.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

- Attach the BM CPU pool to databases. Bare metal CPU pools can be assigned to multiple databases.
- Attach VM CPU pool VM to guest machines. VM CPU pools can be assigned to multiple VMs.
- Attach dbsystem CPU pool to DB systems. dbsystem CPU pool can be assigned to multiple DB systems. After you create a dbsystem CPU pool, you can create a new DB system with this CPU pool by adding the new CPU pool parameter in the create-dbsystem json file, or modify-dbsystem to attach CPU pool to an existing DB system.
- Resources in CPU pools cannot overlap, that is, CPU pools cannot share CPUs.
- After you create a BM CPU pool, you can associate this CPU pool to a database using the `create-database` or `modify-database` command.
- After you create a VM CPU pool, you can associate this CPU pool to a VM using the `create-vm` or `modify-vm` command.
- Number of cores must be an even number between 2 and the total number of available cores on the system.
- By default, a CPU in the CPU pool is allocated across two sockets in release 19.12 and later. Optionally, you can choose to allocate the CPU based on the socket number (0 or 1).

This is especially useful to support Standard Edition licensing. It is recommended that you use the default allocation and do not mix two types of allocation.

- By default, the first core of the socket is reserved and cannot be used by the VM and DB system CPU pools for optimal performance. Use the `--use-reserved-cores` option to enable use of the first core of socket for VM and DB system CPU pools.

Example

Create a BM CPU pool named `cpupool1` with 6 cores associated to it.

```
# odacli create-cpupool -n cpupool1 -c 6 -bm
```

Create a VM CPU pool named `cpupool2` with 6 cores associated to it.

```
# odacli create-cpupool -n cpupool2 -c 6 -vm
```

Create a DB System CPU pool named `cpupool3` with 6 cores associated to it.

```
# odacli create-cpupool -n cpupool3 -c 6 -dbs
```

Create a DB System CPU pool named `cpupool4` with 4 cores of CPUs allocated from socket 0.

```
# odacli create-cpupool -n cpupool4 -dbs -s 0 -c 4
```

odacli remap-cpupools

Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli remap-cpupools -f -urc [-j] [-h]
```

Parameters

Parameter	Description
`--force`, `-f`	(Optional) Specifies to force the operation.
`--use-reserved-cores`, `-j`	(Optional) Specifies to use reserved CPU cores.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

If the `odacli remap-cpupools` command fails with the error `DCS-10001:Internal error encountered: Not enough physical CPUs available for CPU Pool '`*`cpupool_name`*`' on node '`*`node_name`*`'`, run the `odacli remap-cpupools` command again with the `--use_reserved-cores` option.

Use the `--force` option to remap CPU pools even if the CPU pool is marked as NUMA-enabled. You must use `odacli remap-cpupools` command, after updating the bare metal CPU cores with the command `odacli modify-cpucores` for CPU core number changes.

By default, the first core of the socket is reserved and cannot be used by the VM and DB system CPU pools for optimal performance. Use the `--use-reserved-cores` option to enable use of the first core of socket for VM and DB system CPU pools.

Example

```
# odacli remap-cpupools
```

Related Topics

- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

odacli list-cpupools

Use the `odacli list-cpupools` command to list all CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli list-cpupools [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
# odacli list-cpupools
```

odacli describe-cpupool

Use the `odacli describe-cpupool` command to describe CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli describe-cpupool -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the name of the CPU Pool.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example: Describing a CPU pool named `cpupool1`

```
# odacli describe-cpupool -n cpupool1
```

odacli modify-cpupool

Use the `odacli modify-cpupool` command to modify a CPU Pool configuration on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli modify-cpupool -n name -c cores -f -l [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the CPU Pool name
`--cores, -c`	Specifies the new number of CPU cores
`--force, -f`	(Optional) Forces the operation
`--live, -l`	(Optional) Applies the changes to running VMs
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

For VM CPU pool, if `--live` is not specified, then the changes are applied to the VM configuration file and will be activated next time the VM is started. If `--live` is specified, then the changes will be applied to both VM configuration file and to the running VM.

Example

Set the size of a CPU Pool `cpupool1` to 10. If --live is not specified, the change is applied to the VM configuration file and will be activated next time the VM is started.

```
# odacli modify-cpupool -n cpupool1 -c 10
```

odacli delete-cpupool

Use the `odacli delete-cpupool` command to delete a CPU Pool configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli delete-cpupool -n name -f [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the CPU Pool name to be deleted
`--force`, `-f`	(Optional) Forces deletion.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Delete a CPU pool named `cpupool1`.

```
# odacli delete-cpupool -n cpupool1
```

odacli list-auditrecords

Use the `odacli list-auditrecords` command to list the audit records for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli list-auditrecords -du dcs-username -f from_date-range -l latest -nn node_name -
ot operation_type -ou OS_username -rn resource_name -rt resource_type -t to_date-range [-
j] [-h]
```

Parameters

Parameter	Description
`--dcs-user`, `-du`	(Optional) Specifies the DCS user name.
`--from`, `-f`	(Optional) Specfies the *from* date in the date range for the CPU pool audit records.
`--latest`, `-l`	(Optional) Specifies the latest number of CPU pool audit records.
`--node-name`, `-nn`	(Optional) Specifies the node name.
`--op-type`, `-ot`	(Optional) Specifies the type of operation.
`--os-user`, `-ou`	(Optional) Specifies the operating system user.
`--rsc-name`, `-rn`	(Optional) Specifies the resource name.
`--rsc-type`, `-rt`	(Optional) Specifies the resource type.
`--to`, `-t`	(Optional) Specfies the *to* date in the date range for the CPU pool audit records.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
# odacli list-auditrecords
ID                                       Resource type         Resource name
Operation type   DCS username     OS username      Timestamp
---------------------------------------  --------------------  ----------------------
---------------  ---------------  ---------------  ------------------------
41295d99-f263-4284-b6d9-2aff1ae0f850  BM_CPU_POOL           testbmpool
```

```
CREATE          oda-admin       root            2020-10-08 13:27:05 UTC
0f07a013-5bb4-4cd8-bd49-04dc61fcd454  BM_CPU_POOL         pool_9f74a8e9-
DELETE          oda-admin       root            2020-10-08 12:26:30 UTC
d858269b-6c28-4bc2-af71-79004f3561e6  BM_CPU_POOL         pool_9f74a8e9-
CREATE          oda-admin       root            2020-10-08 12:24:36 UTC
9d7852c1-93fb-410e-9e0c-8c0a95ed3f0c  VM_CPU_POOL         pool_d48e81e4-
DELETE          oda-admin       root            2020-10-08 12:06:45 UTC
0235cc84-9345-40bc-ac4f-7de3f63202e2  VM_CPU_POOL         pool_4fe0b4f4-
DELETE          oda-admin       root            2020-10-08 12:06:45 UTC
762f7263-7bd9-4b12-b891-48305165bf36  VM_CPU_POOL         pool_4fe0b4f4-
MODIFY          oda-admin       root            2020-10-08 12:06:22 UTC
0bd2a25d-2327-4918-bd47-8c00f7406f92  VM_CPU_POOL         pool_4fe0b4f4-
MODIFY
```

odacli describe-auditrecord

Use the `odacli describe-auditrecord` command to describe the audit record for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli describe-auditrecord -i [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	Describes the ID of the audit record.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

```
# odacli describe-auditrecord -i 41295d99-f263-4284-b6d9-2aff1ae0f850
Audit Record details
--------------------------------------------------------------------------------
 Event details
--------------------------
                 Record ID:  41295d99-f263-4284-b6d9-2aff1ae0f850
          Record timestamp:  2020-10-08 13:27:05 UTC
              DCS username:  oda-admin
               OS username:  root
                Cluster ID:  test_cluster-c
                   Node ID:  test_cluster

 Resource/Operation
--------------------------
            Resource type:  BM_CPU_POOL
              Resource ID:  b17cb623-bf79-491f-86d8-d8502987e26c
            Resource name:  testbmpool
           Operation type:  CREATE
    Operation description:  Created with '2' cores and allocation
'test_cluster:1-2,17-18'
```

odacli Database Commands

Use the `odacli database` commands to perform database lifecycle operations.

- odacli clone-database
  Use the `odacli clone-database` command to clone a new database from a source database.
- odacli create-database
  Use the `odacli create-database` command to create a new database.
- odacli delete-database
  Use the `odacli delete-database` command to delete a database.
- odacli describe-database
  Use the `odacli describe-database` command to display database details.
- odacli list-databases
  Use the `odacli list-databases` command to list all databases on the appliance.
- odacli modify-database
  Use the `odacli modify-database` command to modify the configuration of a database, such as backup configuration, database class, and database type, and to change TDE wallet management from `EXTERNAL` to `ODA`.
- odacli move-database
  Use the command `odacli move-database` to move a database from one Oracle home to another home of the same database version.
- odacli register-database
  Use the `odacli register-database` command to register a migrated database with the appliance.
- odacli upgrade-database
  You can use the `odacli upgrade-database` command to upgrade a database from a supported release.

odacli clone-database

Use the `odacli clone-database` command to clone a new database from a source database.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli clone-database [-u] [-n] [-dh] [-nh] [-ro] [-s] [-t] [-j] [-f] [-nn] [-p] [-h] [-sh|-no-sh] [-ha|-no-ha] [-cp]
```

Parameters

Parameter	Description
`--databaseUniqueName, -u`	Defines a unique name for the database.
`--dbname, -n`	Defines the name given to the new database (`dbname`.)
`--dbhomeid, -dh`	Specifies the database home ID of the new database.
`--newhome, -nh`	Specifies that a new database home must be created.
`--readonly, -ro`	(Optional) Specifies to open the database in read-only mode.

Parameter	Description
--dbshape, -s	(Optional) Identifies the database shape (template) and determines the total memory allocated to the database. For example, odb1 and odb2. The default is odb1.
--dbtype, -t [RAC|RACOne|SI]	(Optional) Defines the database type. The default database type is the source database type, if the option is not specified. For high-availability deployments, the values can be RAC|RACOne|SI, with the default as RAC. For single-node deployments, the value is SI.
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output.
--sourcedbname, -f	Specifies the name of the source database
--associated-networks, -nn	(Optional) Specifies the associated network names (in the format networkName1,networkName2,...).
--disable-seha, -no-sh	(Optional) Specifies whether you want to disable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.
--enable-seha, -sh	(Optional) Specifies whether you want to enable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.
--disable-ha, -no-ha	(Optional) Disables High Availability for single-instance Oracle Database 19c and later.
--enable-ha, -ha	(Optional) Enables High Availability for single-instance Oracle Database 19c and later.
--cpupool, -cp	(Optional) Specifies the CPU pool name.
--tdepassword, -p	(Optional) Specifies the TDE password of the source database.

Usage Notes

- The source database must be up and running.
- The source database must be opened in either read-write mode, or read-only mode if it is a primary database. A standby database can be in mounted or read-only open mode.
- The source database must use Oracle ACFS storage.
- The source database must not be a multitenant container database (CDB)
- If -cp parameter is provided as part of the command, cloned database is associated with the particular CPU pool.
- If -cp parameter is not provided as part of the command, and if the parent database has an associated CPU pool, then the cloned database is also associated with the same CPU pool as that of parent database.

- If `-cp` parameter is not provided as part of the command, and if the parent database does not have any associated CPU pool, then the cloned database is also not associated with any CPU pool.
- All data files of the source database must be in the same Oracle ACFS directory.

Example 16-136 Cloning a Database in a new database home

```
# odacli list-databases

ID                                       DB Name    DB Type  DB
Version           CDB        Class    Shape    Storage    Status
DbHomeID
---------------------------------------- ---------- --------
------------------- ---------- -------- -------- ---------- ------------
----------------------------------------
bb06935b-384d-482f-b494-45d368d2122a     mydb       SI
19.16.0.0.220719     true       OLTP     odb1     ACFS       CONFIGURED
542032e5-a76c-4d66-92e1-bb297ec6192d
a03b5971-bee5-4d9a-b003-009df6f018f6     primary    SI
19.16.0.0.220719     false      OLTP     odb1     ACFS       CONFIGURED
618b26d7-a49b-452f-a14b-cfc2c79d0555
a5173f68-6b5c-491f-b8b4-05bce0e7ec20     chome1     SI
19.16.0.0.220719     false      OLTP     odb1     ACFS       CONFIGURED
6339e2fc-7ae7-4de2-ac3c-9f5b2eaec131

# odacli list-dbhomes

ID                                       Name                 DB
Version                                  Home
Location                                 Status
---------------------------------------- --------------------
----------------------------------------
---------------------------------------------- ----------
542032e5-a76c-4d66-92e1-bb297ec6192d     OraDB19000_home1
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_1 CONFIGURED
6339e2fc-7ae7-4de2-ac3c-9f5b2eaec131     OraDB19000_home2
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_2 CONFIGURED
618b26d7-a49b-452f-a14b-cfc2c79d0555     OraDB19000_home3
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_3 CONFIGURED

# odacli clone-database -n chome4  -u chome4 -f primary -nh
Enter SYS user password:
Retype SYS user password:

Job details
----------------------------------------------------------------
                     ID:  f3052b47-e9b6-4825-bef3-91635db4f255
            Description:  Snap Database service creation with db name: chome4
                 Status:  Created
                Created:  August 4, 2022 12:23:41 PM UTC
                Message:

Task Name                                Start Time
```

```
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------

# odacli describe-job -i f3052b47-e9b6-4825-bef3-91635db4f255

Job details
----------------------------------------------------------------
                     ID:  f3052b47-e9b6-4825-bef3-91635db4f255
            Description:  Snap Database service creation with db name: chome4
                 Status:  Success
                Created:  August 4, 2022 12:23:41 PM UTC
                Message:

Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
Validate HugeSpaces                      August 4, 2022 12:24:00 PM UTC
August 4, 2022 12:24:00 PM UTC      Success
Validate source database                 August 4, 2022 12:24:00 PM UTC
August 4, 2022 12:24:11 PM UTC      Success
Setting up ssh equivalance               August 4, 2022 12:24:11 PM UTC
August 4, 2022 12:24:12 PM UTC      Success
Creating volume rdochome4                August 4, 2022 12:24:12 PM UTC
August 4, 2022 12:24:28 PM UTC      Success
Creating ACFS filesystem for RECO        August 4, 2022 12:24:28 PM UTC
August 4, 2022 12:24:46 PM UTC      Success
Database home creation                   August 4, 2022 12:24:46 PM UTC
August 4, 2022 12:28:06 PM UTC      Success <========== New task
Creating ACFS database home              August 4, 2022 12:24:46 PM UTC
August 4, 2022 12:24:46 PM UTC      Success <========== New task
Validating dbHome available space        August 4, 2022 12:24:46 PM UTC
August 4, 2022 12:24:46 PM UTC      Success <========== New task
Creating DbHome Directory                August 4, 2022 12:24:46 PM UTC
August 4, 2022 12:24:46 PM UTC      Success <========== New task
Create required directories              August 4, 2022 12:24:46 PM UTC
August 4, 2022 12:24:46 PM UTC      Success <========== New task
Extract DB clone                         August 4, 2022 12:24:46 PM UTC
August 4, 2022 12:26:14 PM UTC      Success <========== New task
ProvDbHome by using RHP                  August 4, 2022 12:26:14 PM UTC
August 4, 2022 12:27:43 PM UTC      Success <========== New task
Enable DB options                        August 4, 2022 12:27:43 PM UTC
August 4, 2022 12:28:02 PM UTC      Success <========== New task
Creating wallet for DB Client            August 4, 2022 12:28:06 PM UTC
August 4, 2022 12:28:06 PM UTC      Success <========== New task
Audit directory creation                 August 4, 2022 12:28:06 PM UTC
August 4, 2022 12:28:06 PM UTC      Success <========== New task
Creation of snapshot database            August 4, 2022 12:28:06 PM UTC
August 4, 2022 12:32:46 PM UTC      Success
Removing Disabled Redo Threads           August 4, 2022 12:32:46 PM UTC
August 4, 2022 12:32:47 PM UTC      Success
Creation of SPFile for Shared location   August 4, 2022 12:32:50 PM UTC
August 4, 2022 12:32:56 PM UTC      Success
Delete SPFile from local location        August 4, 2022 12:32:56 PM UTC
August 4, 2022 12:32:56 PM UTC      Success
```

```
Register DB with clusterware             August 4, 2022 12:32:56 PM UTC
August 4, 2022 12:34:23 PM UTC      Success
Set SysPassword and Create PwFile        August 4, 2022 12:34:23 PM UTC
August 4, 2022 12:34:27 PM UTC      Success
Creating pfile                           August 4, 2022 12:34:27 PM UTC
August 4, 2022 12:34:28 PM UTC      Success
Disabling external references            August 4, 2022 12:34:28 PM UTC
August 4, 2022 12:34:29 PM UTC      Success
Set CPU pool                             August 4, 2022 12:34:29 PM UTC
August 4, 2022 12:34:29 PM UTC      Success
Enabling supplemental log data           August 4, 2022 12:35:45 PM UTC
August 4, 2022 12:35:46 PM UTC      Success
Enabling blockchange tracking            August 4, 2022 12:35:46 PM UTC
August 4, 2022 12:35:47 PM UTC      Success
Enabling flashback                       August 4, 2022 12:35:47 PM UTC
August 4, 2022 12:35:48 PM UTC      Success
Copy Pwfile to Shared Storage            August 4, 2022 12:35:48 PM UTC
August 4, 2022 12:35:51 PM UTC      Success
Create tns entry                         August 4, 2022 12:35:51 PM UTC
August 4, 2022 12:35:52 PM UTC      Success
Reset Associated Networks                August 4, 2022 12:35:52 PM UTC
August 4, 2022 12:35:55 PM UTC      Success
Set log_archive_dest for Database        August 4, 2022 12:35:55 PM UTC
August 4, 2022 12:35:58 PM UTC      Success

# odacli list-databases

ID                                       DB Name    DB Type  DB
Version           CDB        Class    Shape    Storage    Status
DbHomeID
---------------------------------------- ---------- --------
-------------------- ---------- -------- -------- ---------- ------------
----------------------------------------
bb06935b-384d-482f-b494-45d368d2122a     mydb       SI
19.16.0.0.220719     true       OLTP     odb1     ACFS       CONFIGURED
542032e5-a76c-4d66-92e1-bb297ec6192d
a03b5971-bee5-4d9a-b003-009df6f018f6     primary    SI
19.16.0.0.220719     false      OLTP     odb1     ACFS       CONFIGURED
618b26d7-a49b-452f-a14b-cfc2c79d0555
a5173f68-6b5c-491f-b8b4-05bce0e7ec20     chome1     SI
19.16.0.0.220719     false      OLTP     odb1     ACFS       CONFIGURED
6339e2fc-7ae7-4de2-ac3c-9f5b2eaec131
6bf06951-d89f-4acb-bb93-39ebdfcec4d7     chome4     SI
19.16.0.0.220719     false      OLTP     odb1     ACFS       CONFIGURED
31693275-85f5-4932-8ad6-8abe8c93ce46

# odacli list-dbhomes

ID                                       Name                 DB
Version                                Home
Location                                 Status
---------------------------------------- --------------------
----------------------------------------
---------------------------------------- ----------
542032e5-a76c-4d66-92e1-bb297ec6192d     OraDB19000_home1
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
```

```
19.0.0.0/dbhome_1 CONFIGURED
6339e2fc-7ae7-4de2-ac3c-9f5b2eaec131     OraDB19000_home2
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_2 CONFIGURED
618b26d7-a49b-452f-a14b-cfc2c79d0555     OraDB19000_home3
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_3 CONFIGURED
31693275-85f5-4932-8ad6-8abe8c93ce46     OraDB19000_home4
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_4 CONFIGURED <=== New DB Home


# odacli describe-database -n chome4
Database details
----------------------------------------------------------------
                     ID: 6bf06951-d89f-4acb-bb93-39ebdfcec4d7
            Description: chome4
                DB Name: chome4
             DB Version: 19.16.0.0.220719
                DB Type: SI
                DB Role: PRIMARY
    DB Target Node Name: scaoda8m002
             DB Edition: EE
                   DBID: 2169245064
 Instance Only Database: false
                    CDB: false
               PDB Name:
    PDB Admin User Name:
      High Availability: false
                  Class: OLTP
                  Shape: odb1
                Storage: ACFS
          DB Redundancy:
           CharacterSet: AL32UTF8
  National CharacterSet: AL16UTF16
               Language: AMERICAN
              Territory: AMERICA
                Home ID: 31693275-85f5-4932-8ad6-8abe8c93ce46
<====================== OraDB19000_home4
        Console Enabled: false
  TDE Wallet Management:
           TDE Enabled: false
     Level 0 Backup Day:
     AutoBackup Enabled: false
                Created: August 4, 2022 12:23:41 PM UTC
         DB Domain Name: example.com
    Associated Networks: Public-network
          CPU Pool Name:
```

Example 16-137 Cloning a Database in an existing database home

```
# odacli list-dbhomes

ID                                       Name                 DB
Version                                  Home
```

```
Location                                  Status
----------------------------------------- --------------------
----------------------------------------
---------------------------------------- ----------
542032e5-a76c-4d66-92e1-bb297ec6192d     OraDB19000_home1
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_1 CONFIGURED

# odacli create-dbhome -v 19.16.0.0.220719

# odacli describe-job -i e2e73a4a-cb54-4bef-90ba-e6d6aeff594a

 Job details
----------------------------------------------------------------
                     ID:  e2e73a4a-cb54-4bef-90ba-e6d6aeff594a
            Description:  Database Home OraDB19000_home2 creation with
version :19.16.0.0.220719
                 Status:  Success
                Created:  August 4, 2022 11:00:48 AM UTC
                Message:  Create Database Home

Task Name                                Start Time
End Time                            Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
Setting up ssh equivalance               August 4, 2022 11:01:04 AM UTC
August 4, 2022 11:01:05 AM UTC      Success
Setting up ssh equivalance               August 4, 2022 11:01:05 AM UTC
August 4, 2022 11:01:06 AM UTC      Success
Creating ACFS database home              August 4, 2022 11:01:06 AM UTC
August 4, 2022 11:01:06 AM UTC      Success
Validating dbHome available space        August 4, 2022 11:01:06 AM UTC
August 4, 2022 11:01:06 AM UTC      Success
Creating DbHome Directory                August 4, 2022 11:01:06 AM UTC
August 4, 2022 11:01:06 AM UTC      Success
Create required directories              August 4, 2022 11:01:06 AM UTC
August 4, 2022 11:01:06 AM UTC      Success
Extract DB clone                         August 4, 2022 11:01:06 AM UTC
August 4, 2022 11:02:35 AM UTC      Success
ProvDbHome by using RHP                  August 4, 2022 11:02:35 AM UTC
August 4, 2022 11:04:06 AM UTC      Success
Enable DB options                        August 4, 2022 11:04:06 AM UTC
August 4, 2022 11:04:25 AM UTC      Success
Creating wallet for DB Client            August 4, 2022 11:04:29 AM UTC
August 4, 2022 11:04:29 AM UTC      Success

# odacli list-dbhomes

ID                                       Name                 DB
Version                                Home
Location                                 Status
---------------------------------------- --------------------
----------------------------------------
---------------------------------------- ----------
542032e5-a76c-4d66-92e1-bb297ec6192d     OraDB19000_home1
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
```

```
19.0.0.0/dbhome_1 CONFIGURED
6339e2fc-7ae7-4de2-ac3c-9f5b2eaec131     OraDB19000_home2
19.16.0.0.220719                         /u01/app/odaorahome/oracle/product/
19.0.0.0/dbhome_2 CONFIGURED


# odacli clone-database -n chome1 -u chome1 -f primary -dh 6339e2fc-7ae7-4de2-
ac3c-9f5b2eaec131 <========== New DB Home OraDB19000_home2  [root@scaoda8m002
bin]# odacli describe-job -i ed76ad28-aff6-47c7-9233-6ef2f7c3e8cd

Job details
----------------------------------------------------------------
                     ID:  ed76ad28-aff6-47c7-9233-6ef2f7c3e8cd
            Description:  Snap Database service creation with db name: chome1
                 Status:  Success
                Created:  August 4, 2022 12:02:54 PM UTC
                Message:

Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
Validate HugeSpaces                      August 4, 2022 12:02:59 PM UTC
August 4, 2022 12:03:00 PM UTC      Success
Validate source database                 August 4, 2022 12:03:00 PM UTC
August 4, 2022 12:03:11 PM UTC      Success
Setting up ssh equivalance               August 4, 2022 12:03:11 PM UTC
August 4, 2022 12:03:11 PM UTC      Success
Creating volume rdochome1                August 4, 2022 12:03:11 PM UTC
August 4, 2022 12:03:27 PM UTC      Success
Creating ACFS filesystem for RECO        August 4, 2022 12:03:27 PM UTC
August 4, 2022 12:03:45 PM UTC      Success
Audit directory creation                 August 4, 2022 12:03:45 PM UTC
August 4, 2022 12:03:45 PM UTC      Success
Creation of snapshot database            August 4, 2022 12:03:45 PM UTC
August 4, 2022 12:08:25 PM UTC      Success
Removing Disabled Redo Threads           August 4, 2022 12:08:25 PM UTC
August 4, 2022 12:08:26 PM UTC      Success
Creation of SPFile for Shared location   August 4, 2022 12:08:28 PM UTC
August 4, 2022 12:08:35 PM UTC      Success
Delete SPFile from local location        August 4, 2022 12:08:35 PM UTC
August 4, 2022 12:08:35 PM UTC      Success
Register DB with clusterware             August 4, 2022 12:08:35 PM UTC
August 4, 2022 12:10:00 PM UTC      Success
Set SysPassword and Create PwFile        August 4, 2022 12:10:00 PM UTC
August 4, 2022 12:10:03 PM UTC      Success
Creating pfile                           August 4, 2022 12:10:03 PM UTC
August 4, 2022 12:10:04 PM UTC      Success
Disabling external references            August 4, 2022 12:10:04 PM UTC
August 4, 2022 12:10:06 PM UTC      Success
Set CPU pool                             August 4, 2022 12:10:06 PM UTC
August 4, 2022 12:10:06 PM UTC      Success
Enabling supplemental log data           August 4, 2022 12:11:18 PM UTC
August 4, 2022 12:11:20 PM UTC      Success
Enabling blockchange tracking            August 4, 2022 12:11:20 PM UTC
August 4, 2022 12:11:21 PM UTC      Success
```

```
Enabling flashback                        August 4, 2022 12:11:21 PM UTC
August 4, 2022 12:11:22 PM UTC      Success
Copy Pwfile to Shared Storage             August 4, 2022 12:11:22 PM UTC
August 4, 2022 12:11:25 PM UTC      Success
Create tns entry                          August 4, 2022 12:11:25 PM UTC
August 4, 2022 12:11:26 PM UTC      Success
Reset Associated Networks                 August 4, 2022 12:11:26 PM UTC
August 4, 2022 12:11:29 PM UTC      Success
Set log_archive_dest for Database         August 4, 2022 12:11:29 PM UTC
August 4, 2022 12:11:32 PM UTC      Success

# odacli list-databases

ID                                       DB Name    DB Type  DB
Version           CDB        Class    Shape    Storage    Status
DbHomeID
---------------------------------------- ---------- --------
------------------- ---------- -------- -------- ---------- ------------
----------------------------------------
bb06935b-384d-482f-b494-45d368d2122a     mydb       SI
19.16.0.0.220719     true       OLTP     odb1     ACFS       CONFIGURED
542032e5-a76c-4d66-92e1-bb297ec6192d
a03b5971-bee5-4d9a-b003-009df6f018f6     primary    SI
19.16.0.0.220719     false      OLTP     odb1     ACFS       CONFIGURED
618b26d7-a49b-452f-a14b-cfc2c79d0555
a5173f68-6b5c-491f-b8b4-05bce0e7ec20     chome1     SI
19.16.0.0.220719     false      OLTP     odb1     ACFS       CONFIGURED
6339e2fc-7ae7-4de2-ac3c-9f5b2eaec131 <============== CLONE DB CREATED
SUCCESSFULLY

# odacli describe-database -n chome1
Database details
----------------------------------------------------------------
                     ID: a5173f68-6b5c-491f-b8b4-05bce0e7ec20
            Description: chome1
                DB Name: chome1
             DB Version: 19.16.0.0.220719
                DB Type: SI
                DB Role: PRIMARY
    DB Target Node Name: scaoda8m002
            DB Edition: EE
                   DBID: 2639447957
 Instance Only Database: false
                    CDB: false
               PDB Name:
    PDB Admin User Name:
      High Availability: false
                  Class: OLTP
                  Shape: odb1
                Storage: ACFS
          DB Redundancy:
           CharacterSet: AL32UTF8
  National CharacterSet: AL16UTF16
               Language: AMERICAN
              Territory: AMERICA
                Home ID: 6339e2fc-7ae7-4de2-ac3c-9f5b2eaec131
```

```
       Console Enabled: false
  TDE Wallet Management:
           TDE Enabled: false
    Level 0 Backup Day:
    AutoBackup Enabled: false
               Created: August 4, 2022 12:02:54 PM UTC
        DB Domain Name: example.com
   Associated Networks: Public-network
         CPU Pool Name:
```

Example 16-138 Cloning a Database and associating it with the same CPU Pool as the parent database

```
# odacli clone-database -f dbcpupin -n dbclone -u dbcloneUnique
Enter SYS user password:
Retype SYS user password:

Job details
----------------------------------------------------------------
                     ID:  74a80586-3523-4367-93a0-38401647c5a0
            Description:  Snap Database service creation with db name: dbclone
                 Status:  Created
                Created:  August 09, 2022 10:25:40 AM UTC
                Message:

Task Name                                Start Time
End Time                            Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
```

Example 16-139 Cloning a Database and associating it with a different CPU pool

```
# odacli clone-database -f dbcpupin -n db1clone -u db1cloneUnique -cp
testdbcpupool2
Enter SYS user password:
Retype SYS user password:

Job details
----------------------------------------------------------------
                     ID:  2fe285b5-2e3b-4e80-9121-e7c8380952b3
            Description:  Snap Database service creation with db name:
db1clone
                 Status:  Created
                Created:  August 09, 2022 10:36:55 AM UTC
                Message:

Task Name                                Start Time
End Time                            Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
```

Example 16-140 Cloning a TDE-enabled database

```
# odacli clone-database --sourcedbname tdbacf2 --dbname tdbacf2c --
databaseUniqueName tdacf2cu -p
Enter SYS user password:
Retype SYS user password:
Enter TDE wallet password of source database:
```

Example 16-141 Cloning a database and opening the clone in read only mode

```
# odacli clone-database -n clonedb -u clonedb -f sourcedb -ro
Enter SYS user password:
Retype SYS user password:
```

The task `Open Clone DB in READ ONLY mode` is displayed as the last task in the description of the above job.

Related Topics

- Cloning an Oracle ACFS Database
  Create a database from an existing Oracle ACFS database using ODACLI commands or BUI.

odacli create-database

Use the `odacli create-database` command to create a new database.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli create-database -n database_name -cs characterset -cl {OLTP|DSS|IMDB}
-l dblanguage -s dbshape -r {ACFS|ASM} -dt dbterritory
-y dbtype -ns nationalscharacterset -d pdbadmin -p pdbname -v version
[-u databaseUniqueName] [-dh Database Home ID] [-c|-no-c] [-co|-no-co]
[-bi backupconfigid] [-io] [-dn] [-j] [-nn] [-tn node_name] [-h] [-rd] [-sh|-
no-sh][-ha|-no-ha] [-t] [-cp] [-osc] [-kt] [-ocp]
```

Parameters

Parameter	Description
`--backupconfigid, -bi`	(Optional) Defines the backup configuration identifier for future use.
`--cdb, -c`	(Optional) Creates the database as a container database. Use the `-c` flag to create a container database and use the `-no-c` flag to create a non-CDB database. The default is `-no-c`.
`--characterset, -cs`	Defines the character set. The default is AL32UTF8.
`--databaseUniqueName, -u`	(Optional) Defines a unique name for the database.
`--dbdomainname, -dn`	Defines the database domain name.

Parameter	Description
--cpupool, -cp	Defines the CPU pool name. The CPU pool must be of type bare metal (BM).
--dbEdition, -de	Defines the Database Edition.
--dbclass, -cl {OLTP|DSS|IMDB}	Defines the database class. The default is OLTP. The options are as follows: • Enterprise Edition: OLTP, DSS, or IMDB. • Standard Edition: OLTP
--dbconsole, -co	(Optional) Enables the Database Console. Use the -no-co flag to disable the Database Console. If not selected, the default is no database console.
--dbhomeid, -dh	(Optional) Identifies the existing Database Home ID.
--dblanguage, -l	Defines the database language. The default language is AMERICAN.
--dbname, -n	Defines the name given to the new database (dbname.)
--dbshape, -s	Identifies the database shape (template) and determines the total memory allocated to the database. For example, odb1 and odb2. The default is odb1. You cannot specify the database shape when you create a instance-only database.
--dbstorage, -r{ACFS|ASM}	Defines the Database Storage, either Oracle ACFS or Oracle ASM. The default value is Oracle ASM.
--dbterritory, -dt	Defines the database territory. The default territory is AMERICA.
--dbtype, -y [RAC|RACOne|SI]	Defines the database type. For high-availability deployments, the values can be RAC|RACOne|SI, with the default as RAC. For single-node deployments, the value is SI.
--help, -h	(Optional) Displays help for using the command.
--instanceonly, -io	(Optional) Creates a database instance, password file and also the underlying Oracle ACFS mount point. You can use the instance as an auxiliary instance for RMAN duplicate.
--json, -j	(Optional) Displays JSON output.
--nationalscharacterset, -ns	Defines the NLS National Character Set. The default is AL16UTF16.
--no-cdb, -no-c	(Optional) Creates a database that is *not* a container database. Use this flag when you want to create a non-CDB database. Use the -c flag to create a container database.
--no-dbconsole, -no-co	(Optional) Disables Database Console. Use the -co flag to enable Database Console.
--pdbadmin, -d	Defines the Pluggable Database (PDB) Admin User.
--pdbname, -p	Defines the Pluggable Database (PDB) name. The default value is pdb1.
--version, -v	Defines the database bundle patch number. To install the latest bundle patch for a release, specify the release version. To specify a specific supported bundle, use the 5 digit format.

Parameter	Description
--dbRedundancy, -rd	Specifies the database redundancy value, that is, {HIGH|MIRROR}. To specify the dbRedundancy option for an Oracle ASM or ACFS storage-based database, all disk groups must have FLEX redundancy. To specify the dbRedundancy option for an Oracle ASM storage database, the dbVersion or dbHomeVersion must be 12.2 or later.
--associated-networks, -nn	Specifies the associated network names (in the format networkName1,networkName2,...).
--enable-tde, -t	Specifies whether to enable Transparent Data Encryption (TDE) which uses software keystore.
--keystore-type, -kt	Specifies the type of keystore which stores the TDE master encryption key. The value can be software or okv. The default is software.
--okv-server-config, -osc	Specifies the name of the Oracle Key Vault server configuration which serves as the metadata of the Oracle Key Vault user credentials. This option must be provided only when the --keystore-type,-kt option is specified..
--okvclient-path, -ocp	Specifies the absolute path to the okvclient.jar file. Multiple values must be comma-separated. Cannot be used with the --okv-server-config,-osc option.
--disable-seha, -no-sh	Specifies whether you want to disable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.
--enable-seha, -sh	Specifies whether you want to enable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.
--target-node, -tn	(Optional) Specifies the target node name (applicable only for Oracle RAC One Node Database and single-instance Oracle Database, where supported).
--levelzerobackupday, -lb	(Optional) Specifies the level zero backup day. The values can be Monday|Tuesday|Wednesday...|Sunday. The default is Sunday.
--backupconfigname, -bn	(Optional) Specifies the name of the backup configuration.
--bkuppassword, -bp	(Optional) Specifies the RMAN backup encryption password.
--enableFlashCache, -fc	(Optional) Specifies whether to enable Flash Cache.
--no-enableFlashCache, -no-fc	(Optional) Specifies whether to disable Flash Cache.
--dbOnFlashStorage, -f	(Optional) Specifies whether to enable data on Flash storage.
--no-dbOnFlashStorage, -no-f	(Optional) Specifies whether to disable data on Flash storage.
--disable-ha, -no-ha	(Optional) Disables High Availability for single-instance Oracle Database 19c and later.
--enable-ha, -ha	(Optional) Enables High Availability for single-instance Oracle Database 19c and later. The --enable-ha,-ha option is the default for single-instance databases on high availability deployments.

Usage Notes

- If the disk group redundancy is FLEX, then the default `dbRedundancy` value is MIRROR.
- Flash cache is disabled by default.
- You cannot mix Oracle Database Standard Edition and Enterprise Edition databases on the same appliance.
- For Standard Edition Oracle Database 19c or later, you cannot create Oracle RAC Or Oracle RAC One Node Database. For Standard Edition, you can only deploy single-instance Oracle Database with or without high-availability enabled.
- Transparent Data Encryption (TDE) is an Advanced Security Option (ASO) available with Oracle Database Enterprise Edition.
- TDE is not configured by default during database creation.
- The `dbname` and `databaseUniqueName` can contain underscores (-), lowercase and uppercase alphabet (a-zA-Z, and numeric characters (0-9), must start with an alphabet, and must not exceed 8 characters in length. Underscores are not permitted only if the `--dbname` is set to RACOne, as it is used as a delimiter to identify the database instance. The `dbname` and `databaseUniqueName` must not also be an Oracle reserved key word.
- Use the `--cdb` or `--no-cdb` flag to indicate whether or not the database is a container database. When neither flag is specified, the default database created is a non-CDB database.
- When `--dbhomeid` is not provided, the `create-database` command creates a new Oracle Database Home.
- When `--dbhomeid` is provided, the `create-database` command creates the database using the existing Oracle Home. Use the `odacli list-dbhomes` command to obtain the `dbhomeid`.
- When you use the command to create an instance-only database, then you cannot specify the database shape (template).
- When you specify both the `--version` and the`--dbhomeid`, the version is ignored and the database is created against the existing database home.
- Oracle Database 19c and later is supported on both Oracle Automatic Storage Management (Oracle ASM) and Oracle ASM Cluster file system (ACFS). The default is Oracle ASM.
- When databases are created on Oracle ACFS, each database is configured with its own Oracle ACFS file system for the datafiles and uses the following naming convention: `/u02/app/`*`db user`*`/oradata/`*`db unique name`*. The default size of this mount point is 100 GB.

  > **Note:**
  >
  > Oracle recommends not sharing the mount point across different databases.

- Online logs are stored in the `/u03/app/`*`db user`*`/redo/` directory.
- Oracle Fast Recovery Area (FRA) is located in the `/u03/app/`*`db user`*`/fast_recovery_area` directory.

- For the version, you can specify the database version, for example, 19.24.0.0, or you can use a 5 digit format to specify a specific patch bundle version. If you use the database version without specifying the bundle patch number, the latest bundle patch is used.
- You must specify either the Oracle Key Vault server configuration or the absolute path of the `okvclient.jar` file.
- You cannot configure Oracle Key Vault on Oracle RAC One database and single-instance database with high-availability enabled.
- For Oracle Key Vault configuration, you must specify one `okvclient.jar` file path for single-instance database. For Oracle RAC database, you must specify multiple `okvclient.jar` file path.
- For TDE-enabled database configuration, if the keystore type is not specified, then, by default, the keystore is software keystore.

> **Caution:**
>
> When you create a database with Transparent Data Encryption (TDE) enabled, you are prompted to set a password for the TDE wallet. Provide a strong password for security compliance. Set the password carefully, and ensure that this password is available to you at all times for database management operations. Failure to provide the TDE wallet password when prompted, will cause an error in accessing encrypted data.

Example 16-142 Creating a TDE-Enabled Database with Software Keystore

```
# odacli create-database -n test -t
Enter new 'SYS, SYSTEM and PDB Admin user' password:
Confirm 'SYS, SYSTEM and PDB Admin user' password:
Enter new 'TDE wallet' password:
Confirm 'TDE wallet' password:
```

Example 16-143 Creating a Database in Interactive Mode

This example creates a 19.25.0.0.0 OLTP container database named `hrdb` with shape `odb2` and enables you to provide the password interactively.

```
# odacli create-database -n hrdb -c -cl OLTP -s odb2 -p pdb1

Enter new 'SYS, SYSTEM and PDB Admin user' password:
Confirm 'SYS, SYSTEM and PDB Admin user' password:
{
   "jobId" : "f12485f2-dcbe-4ddf-aee1-de24d37037b6",
   "status" : "Created",
   "message" : null,
   "reports" : [ ],
   "createTimestamp" : "September 8, 2024 03:54:03 AM EDT",
   "description" : "Database service creation with db name: hrdb",
   "updatedTime" : "September 8, 2024 03:54:03 AM EDT"
}
```

Example 16-144 Creating a Database Against a Different Version

Either of the following statements creates a database against a home with Oracle Database Bundle applied:

```
# odacli create-database -n hrmsdb1 -v 19.25.0.0.0
# odacli create-database -n hrmsdb2 -v 19.25.0.0.241015
```

The following statement creates a new database against a home with a specific Oracle Database Bundle:

```
# odacli create-database -n hrmsdb3 -v 19.25.0.0.241015
```

Example 16-145 Creating a Database associated with a CPU Pool

The following command creates a new database associated with a CPU pool:

```
# odacli create-database -j -n dbcpupl -r ACFS -cp testdbcpupool
```

Example 16-146 Creating a TDE-Enabled Database with Credentials of a User on Oracle Key Vault

An example to create a database `dbokv1` which uses Oracle Key Vault credentials represented by Oracle Key Vault server configuration `okvobj1` is as follows:

```
odacli create-database -n dbokv1 -t -osc okvobj1 -kt OKV
Enter SYS and SYSTEM user password:
Enter SYS and SYSTEM user password:
```

Example 16-147 Creating a TDE-Enabled Database with keys stored in Oracle Key Vault

Create an Oracle RAC database okvdbrac without using the Oracle Key Vault server configuration object. Specify the path to the okvclient.jar file for the local database instance first, followed by the path to okvclient.jar file for the remote database instance in the request.

```
odacli create-database -n okvdbrac -y RAC -t -kt OKV -ocp /tmp/okvracdb1/
okvclient.jar,/tmp/okvracdb2/okvclient.jar
Enter SYS and SYSTEM user password:
Retype SYS and SYSTEM user password:
Enter TDE wallet password:
Retype TDE wallet password:

Job details
----------------------------------------------------------------
                     ID:  cb611ec2-ff87-44f9-a79b-3ab1a290644f
            Description:  Database service creation with DB name: okvdbrac
                 Status:  Created
                Created:  October 20, 2024 6:13:28 AM PDT
                Message:

Task Name                                Start
Time                                 End Time
Status
```

```
---------------------------------------
---------------------------------------
---------------------------------------- ----------------


odacli describe-job -i cb611ec2-ff87-44f9-a79b-3ab1a290644f

Job details
----------------------------------------------------------------
                     ID:  cb611ec2-ff87-44f9-a79b-3ab1a290644f
            Description:  Database service creation with DB name: okvdbrac
                 Status:  Success
                Created:  October 20, 2024 6:13:28 AM PDT
                Message:

Task Name                                Node Name                 Start
Time                               End Time
Status
---------------------------------------- ------------------------
---------------------------------------
---------------------------------------- ----------------
Validating dbHome available space        n1             October 20, 2024
6:13:49 AM PDT          October 20, 2024 6:13:49 AM PDT
Success
Validating dbHome available space        n2             October 20, 2024
6:13:49 AM PDT          October 20, 2024 6:13:49 AM PDT
Success
Setting up SSH equivalence               n1             October 20, 2024
6:13:51 AM PDT          October 20, 2024 6:13:54 AM PDT
Success
Setting up SSH equivalence               n1             October 20, 2024
6:13:54 AM PDT          October 20, 2024 6:13:57 AM PDT
Success
Creating ACFS database home              n1             October 20, 2024
6:13:58 AM PDT          October 20, 2024 6:13:58 AM PDT
Success
Validating dbHome available space        n1             October 20, 2024
6:13:59 AM PDT          October 20, 2024 6:13:59 AM PDT          Success
.
.
.
Setting TDE Master Encryption Key        n1             October 20, 2024
6:34:15 AM PDT          October 20, 2024 6:34:56 AM PDT
Success
Create Users tablespace                  n1             October 20, 2024
6:34:57 AM PDT          October 20, 2024 6:34:59 AM PDT
Success
Clear all listeners from Database        n1             October 20, 2024
6:35:00 AM PDT          October 20, 2024 6:35:01 AM PDT
Success
Deleting OKV Client Software Jar         n1             October 20, 2024
6:35:03 AM PDT          October 20, 2024 6:35:04 AM PDT
Success
Deleting OKV Client Software Jar         n2             October 20, 2024
6:35:04 AM PDT          October 20, 2024 6:35:04 AM PDT
Success
```

```
Encrypt all Tablespaces                  n1             October 20, 2024
6:35:04 AM PDT          October 20, 2024 6:36:15 AM PDT          Success
```

odacli delete-database

Use the `odacli delete-database` command to delete a database.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To delete a database:

`odacli delete-database -i dbid -n name [-h] [-j] [-fd]`

Parameters

Parameter	Description
`--id`, `-i`	Identifies the database home identifier (ID) to display. Use the `odacli list-databases` command to obtain the `--id`.
`--name`, `-n`	Defines the database name.
`--force`, `-fd`	(Optional) Forces the delete operation.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage

- When you run the `odacli delete-database` command on a TDE-enabled database that uses software keystore to store TDE keys, the TDE wallet is also deleted.
- When you run the `odacli delete-database` command on a TDE-enabled database that uses Oracle Key Vault to store TDE keys and was created using Oracle Key Vault user credentials, the TDE wallet and the corresponding endpoints in the Oracle Key Vault are also deleted.
- However, when you run the `odacli delete-database` command on a TDE-enabled database that uses Oracle Key Vault to store TDE keys and was created using endpoints in Oracle Key Vault server, the TDE wallet and the corresponding endpoints in the Oracle Key Vault are not deleted automatically. You must manually delete the TDE wallet and the corresponding endpoints in the Oracle Key Vault.
- The `odacli delete-database` command deletes the database, the file system, and the underlying `advm` volumes assigned to the database. For example, deleting a database named `hrdb` also deletes the file system `/u02/app/oracle/oradata/hrdb`. Do not keep any other files on this database file location. If you have manually added static entries for the database you want to delete in your `listener.ora` file, then remove these entries before running the `odacli delete-database` command.
- The `odacli delete-database` command does not delete the database files. This is expected behavior so that you can archive the database files, if required, or manually restore the database with these database files. Use the `--force` option to delete the database files completely.

Example 16-148 Deleting a Database Named `hrmsdb`

In this example we want to delete a database named `hrmsdb`. Before we can delete the database, we need to know the database home identifier (*dbid*). This example shows how to run the `odacli list-databases` command to list the databases and their associated *dbid*, then how to delete database `hrmsdb`.

```
# odacli list-databases

ID                                       DB Name  DB Version   CDB  Class
---------------------------------------  -------    ----------   ----
-----
a3f4a6c0-a0c9-4c79-bad7-898afcf9de46   hrmsdb  19.22.0.0      true   OLTP
7e28bf52-1a09-49fd-9391-841838d2c42f   crmdb   19.22.0.0.      false  OLTP

(continued)
Shape     Storage  Status
------  ------   ----------
odb1     ACFS      Configured
odb1     ACFS      Configured

# odacli delete-database -i a3f4a6c0-a0c9-4c79-bad7-898afcf9de46
```

odacli describe-database

Use the `odacli describe-database` command to display database details.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

To display database details:

```
odacli describe-database -i dbid -n name [-h] [-j]
```

Parameters

Parameter	Description
`--id`, `-i`	Identifies the database home identifier (ID) to display. Use the `odacli list-databases` command to obtain the `--id`.
`--name`, `-n`	Defines the database name.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-149

Display information for a database.

```
# odacli describe-database -n odacn

Database details
```

```
------------------------------------------------------------
                     ID: 896238139
            Description: odacn
                DB Name: odacn
             DB Version: 19.6.0.0.200114
                DB Type: Si
                DB Role: PRIMARY
    DB Target Node Name: node1
             DB Edition: EE
                   DBID: 896238139
 Instance Only Database: false
                    CDB: false
               PDB Name:
    PDB Admin User Name:
High Availability Enabled: false
                  Class: Oltp
                  Shape: Odb1
                Storage: Asm
          DB Redundancy: MIRROR
           CharacterSet: AL32UTF8
  National CharacterSet: AL16UTF16
               Language: AMERICAN
              Territory: AMERICA
                Home ID: bcc3991a-0ffc-4d29-86ae-40f2b8721dc4
        Console Enabled: false
     Level 0 Backup Day: Sunday
     AutoBackup Enabled: true
                Created: March 30, 2020 9:33:09 AM PDT
         DB Domain Name: domain.com
    Associated Networks: Public-network
```

Example 16-150 Displaying Database Details

Display information for database named `ac48e0d2-a7b0-4ffd-a27e-f8e42b028c5f` :

```
# odacli describe-database -i ac48e0d2-a7b0-4ffd-a27e-f8e42b028c5f

Database details
------------------------------------------------------------
ID: ac48e0d2-a7b0-4ffd-a27e-f8e42b028c5f
Description: rdb1
DB Name: rdb1
DB Version: 19.25.0.0.0
DBID: 1339792271
CDB: true
PDB Name: r1pdb1
PDB Admin User Name: pdbadmin
Class: OLTP
Shape: odb2
Storage: ASM
CharacterSet: DbCharacterSet(characterSet=AL32UTF8,
nlsCharacterset=AL16UTF16, dbTerritory=AMERICA, dbLanguage=AMERICAN)
Home ID: fe87f30c-b810-45d1-8b96-13996ad7a255
Console Enabled: true
Created: September 8, 2024, 2016 6:21:14 PM
```

odacli list-databases

Use the `odacli list-databases` command to list all databases on the appliance.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To display a list of all databases:

```
odacli list-databases [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`-all`	(Optional) This option is available only when multi-user access is enabled. Provides a read-only view of all the databases in the system.

Example 16-151 Displaying a List of Databases

Display a list of databases:

```
# odacli  list-databases

ID                                       DB Name     DB Version  CDB
---------------------------------------  ----------  ----------  ----
ad6c7326-e460-411e-94df-230dedbef743   rdb121a     19.25.0.0.0    true
fb4d02f3-2413-47ca-8584-a768e23ec2e7   ee12db      19.25.0.0.0    false

(Continued)
Class    Shape  Storage  Status
------ -------  ----------
OLTP    odb1   ACFS     Configured
IMDB    odb1   ASM      Configured
```

odacli modify-database

Use the `odacli modify-database` command to modify the configuration of a database, such as backup configuration, database class, and database type, and to change TDE wallet management from `EXTERNAL` to `ODA`.

File Path

/opt/oracle/dcs/bin/odacli

Syntax

```
odacli modify-database [-i] [-dn] [-no-sh] [-sh] [-no-ha] [ha] [-lb] [-no-
bkp] [-tn node_name] [-g] [-cl] [-rd] [-s] [-an] [-ctm] [-bi] [-bin] [-bp] [-
id] [-n] [-y] [-cp] [-no-cpupool] [-rkt] [-ctp] [-h] [-j]
```

Parameters

Parameter	Description
`--name`, `-n`	Defines the database name to be updated. Specify either `--name` or `--id`.
`--id`, `-i`	Defines the database ID to be updated. Specify either `--name` or `--id`.
`--dbRedundancy`, `-rd`	(Optional) Specifies the database redundancy value, that is, `{HIGH|MIRROR}`. To specify the `dbRedundancy` option for an Oracle ASM or ACFS storage-based database, at least one disk group of FLEX redundancy must exist. To specify the `dbRedundancy` option for an Oracle ASM storage database, the `dbVersion` or `dbHomeVersion` must be 12.2 or later.
`--levelzerobackupday`, `-lb`	(Optional) Specifies the Level zero Backup Day. For example, Monday|Tuesday|Wednesday...|Sunday
`--backupconfigid`, `-bi`	(Optional) Defines the backup configuration identifier for future use.
`--backupconfigname`, `-bin`	(Optional) Specifies the name of the backup configuration.
`--bkuppassword`, `-bp`	(Optional) Specifies the RMAN backup encryption password.
`--no-backup`, `-no-bkp`	(Optional) Disables database backups. Specify this option to also remove the database backup and archivelog backup schedulers. To enable database backups again, you must update the database with a backupconfig object.
`--dbclass`, `-cl`	(Optional) Defines the database class. The default is OLTP. The options are OLTP, DSS, or IMDB.
`--dbshape`, `-s`	(Optional) Identifies the database shape (template) and determines the total memory allocated to the database. For example, odb1 and odb2. The default is odb1. You cannot specify the database shape for an instance-only database.
`--attachnetworks`, `-an`	(Optional) Specifies the name of the networks to be attached. Specify the networks in the format *network_name1*, *network_name1*, *network_name1*.
`--detachnetworks`, `-dn`	(Optional) Specifies the name of the networks to be detached. Specify the networks in the format *network_name1*, *network_name1*, *network_name1*.
`--dbtype`, `-y`	Defines new database type. The only option is SI. The conversion to single-instance database is available only for Oracle Database Standard Edition.

Parameter	Description
`--disable-seha, -no-sh`	Disables Standard Edition High Availability for single-instance Oracle Database 19c and later. **Note:** This option is deprecated and will be desupported in a future release.
`--enable-seha, -sh`	Enables Standard Edition High Availability for single-instance Oracle Database 19c and later. **Note:** This option is deprecated and will be desupported in a future release.
`--disable-ha, -no-ha`	(Optional) Disables High Availability for single-instance Oracle Database 19c and later.
`--enable-ha, -ha`	(Optional) Enables High Availability for single-instance Oracle Database 19c and later.
`--target-node, -tn`	(Optional) Specifies the target node name (applicable only for Oracle RAC One Node Database and single-instance Oracle Database, where supported).
`--cpupool, -cp`	(Optional) Specifies the CPU pool name. Use this option to associate a CPU pool to a database that does not use any CPU pool or to change the existing association of a CPU pool to a new CPU pool for a database.
`--no-cpupool`	(Optional) Removes the attached CPU pool.
`--change-tde-password,-ctp`	(Optional) When specified, you are prompted for the current and new TDE Wallet password. Note that in this release, you cannot modify the TDE configuration to use Oracle Key Vault.
`--change-tdewallet-mgmt,-ctm`	(Optional) When specified, the TDE wallet is changed from `external` to `ODA`.
`--rekey-tde,-rkt`	(Optional) When specified, current TDE Wallet password is prompted to rekey the TDE master encryption key.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

> **Caution:**
>
> When you change the TDE wallet password for a database with Transparent Data Encryption (TDE) enabled, you are prompted to set a password for the TDE wallet. Provide a strong password for security compliance. Set the password carefully, and ensure that this password is available to you at all times for database management operations. Failure to provide the TDE wallet password when prompted, will cause an error in accessing encrypted data.

Example 16-152 Modifying the Database Shape (Template) and Database Class

```
# odacli modify-database -s database_shape  -cl database_class -i dbid
odacli modify-database -i 1941d594-c777-4eca-9fce-18b778d5c153 -s odb2 -cl DSS
{
  "jobId" : "833d43a7-bcc6-48a7-9f98-b42ffdab3fe1",
```

```
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 16, 2023 06:48:58 AM UTC",
  "resourceList" : [ ],
  "description" : "modify-database service with db ids: 1941d594-
c777-4eca-9fce-18b778d5c153",
  "updatedTime" : "May 16, 2023 06:48:58 AM UTC"
}
```

Example 16-153 Converting Oracle RAC or Oracle RAC One Node Database to Single-instance Oracle Database

```
# odacli modify-database -n db18 -y SI
For moving a database between homes please use the command 'move-database'.
{
  "jobId" : "4b548365-dac5-4557-a6a8-2f65c2b725e7",
  "status" : "Created",
  "message" : "Modify database",
  "reports" : [ ],
  "createTimestamp" : "May 21, 2023 00:32:55 AM UTC",
  "resourceList" : [ {
    "resourceId" : "de2a7959-e31b-4dee-b600-6b16dfb78c34",
    "resourceType" : "DB",
    "resourceNewType" : null,
    "jobId" : "4b548365-dac5-4557-a6a8-2f65c2b725e7",
    "updatedTime" : "May 21, 2023 00:32:55 AM UTC"
  } ],
  "description" : "Modify database : db18",
  "updatedTime" : "May 21, 2023 00:32:55 AM UTC"
}
```

Example 16-154 Specifying the target node when converting database

```
# odacli modify-database -n db18 -y SI -g node_number
# odacli modify-database -n db18 -y SI -th node_name
```

Example 16-155 Changing the TDE Wallet Password

```
# odacli modify-database -n testdb -ctp
Enter current 'TDE wallet' password:
Enter new 'TDE wallet' password:
Confirm 'TDE wallet' password:
```

Example 16-156 Re-keying the TDE Master Encryption Password

```
# odacli modify-database -n testdb -rkt
Enter current 'TDE wallet' password:
```

Example 16-157 Removing a CPU pool

```
# odacli modify-database -n dbcpupin  -cp testdbcpupool2
DB will be restarted as part of CPU Pool operation. Do you want to continue
```

```
[y/n]:y
For moving a database between homes please use the command 'move-database'.
{
  "jobId" : "c1d05fb3-828a-4644-af9a-9d82b4ef5a89",
  "status" : "Created",
  "message" : "Modify database",
  "reports" : [ ],
  "createTimestamp" : "May 09, 2023 16:18:54 PM UTC",
  "resourceList" : [ {
    "resourceId" : "ff9c138b-a003-40cf-b1fc-0b33cd955ebf",
    "resourceType" : "DB",
    "resourceNewType" : null,
    "jobId" : "c1d05fb3-828a-4644-af9a-9d82b4ef5a89",
    "updatedTime" : "May 09, 2023 16:18:54 PM UTC"
  } ],
  "description" : "Modify database : dbcpupin",
  "updatedTime" : "May 09, 2023 16:18:54 PM UTC"
}
```

odacli move-database

Use the command `odacli move-database` to move a database from one Oracle home to another home of the same database version.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
  odacli move-database [-i dbid] [-imp] [-nr] [-n dbname] [-dh dest-dbhome-id]
[-dhn dest-dbhome-name][-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	(Optional) Defines the database ID.
`--name, -n`	(Optional) Defines the database name.
`--non-rolling, -nr`	(Optional) Specifies whether you want to move the target database in non-rolling manner. By default, the value is false and the database is moved in a rolling manner.
`--ignore-missing-patch, -imp`	(Optional) Ignores all missing patches and forces the operation.
`--dest-db-home-id, -dh`	(Optional) Defines the destination database home ID.
`--dest-dbhome-name, -dhn`	(Optional) Defines the destination database home name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Starting with Oracle Database Appliance release 19.19, when you run the `odacli move-database`, the database is moved in a rolling manner, unless you specify the `--non-rolling,-nr` option.

Example 16-158 Moving a Database from one Oracle home to another

```
# ./odacli move-database -i database_ID -dh destination_database_home_ID
```

odacli register-database

Use the `odacli register-database` command to register a migrated database with the appliance.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli register-database  -c {OLTP|DSS|IMDB} -s dbshape -t dbtype SI|RACOne|
RAC -o hostname -sn servicename [-bi backupconfigid] [-co|-no-co] [-nn] [-tp]
[-ha|-no-ha] [-h] [-j] [-sh|-no-sh]
```

Parameters

Parameter	Description
`--backupconfigid, -bi`	(Optional) Defines the backup configuration identifier for future use.
`--dbclass, -c` {OLTP|DSS|IMDB}	Defines the database class. The database class setting determines the database SGA memory and instance PGA memory configuration. The options are as follows: • Enterprise Edition: OLTP, DSS, or IMDB. • Standard Edition: OLTP
`--dbconsole, -co`	(Optional) Enables the Database Console. Use the `-no-co`flag to disable the Database Console. If not selected, the default is no Database Console.
`--dbshape, -s`	Identifies the database shape (template) and determines the total memory allocated to the database. For example, odb1 and odb2. The default is `odb1`.
`--dbtype, -t` [SI|RACOne|RAC]	Defines the type of database. For high-availability deployments, the values can be RAC|RACOne|SI, with the default as RAC. For single-node deployments, the value is SI.
`--help, -h`	(Optional) Displays help for using the command.
`--hostname, -o`	Defines the host name. Default: local host name
`--json, -j`	(Optional) Displays JSON output.

Parameter	Description
--no-dbconsole, -no-co	(Optional) Disables Database Console. Use the -co flag to enable Database Console.
--servicename, -sn	Defines the Database Service Name. Using this service name, the EZCONNECT String is derived for connecting to the database. For example, hostname:port/servicename. The Port number is the port configured for the listener, as part of the deployment.
--associated-networks, -nn	Specifies the associated network names (in the format networkName1,networkName2,...).
--disable-seha, -no-sh	Specifies whether you want to disable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.
--enable-seha, -sh	Specifies whether you want to enable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.
--disable-ha, -no-ha	(Optional) Disables High Availability for single-instance Oracle Database 19c and later.
--enable-ha, -ha	(Optional) Enables High Availability for single-instance Oracle Database 19c and later.
--tde-wallet-passwd, -tp	Specifies the password for the TDE wallet.

Usage Notes

> **Note:**
>
> It is a good practice to use Easy Connect (EZCONNECT) to test the database connectivity before registering the database.
> Log in as the sys user and enter the following command:
>
> ```
> sqlplus sys/password@//hostname:1521/service name
> ```

- The odacli register-database command is supported on the primary and the standby database. For the standby database, only the database is registered with Oracle Database Appliance. Oracle Data Guard is not registered with Oracle Database Appliance.
- The migrated database is registered with the listener configured during the provisioning of the appliance. The migrated database must be in read-write or read-only mode for the registration to succeed.
- The register-database command validates the datafile and log file locations and moves the controlfile and spfile to the correct locations.

- Some `init.ora` parameters are set, or reset, as part of the registration. Review the parameter changes before and after registration.

  The following are examples of changes implemented as part of registration:

  - The `memory_target` is reset.
  - The `sga_target/pga_aggregate_target/log_buffer/inmemory_size` is configured based on the database class and database shape settings used during registration.
  - The registration process sets or resets the recommended appliance-specific parameters. The database is restarted as part of the register database operation.
- The database being registered must use Oracle Managed Files and the file location must match the DATA Location, REDO Location and RECO Location of the `odacli describe-dbstorage` command.
- As part of the registration process, the database is registered with Oracle Clusterware. Only the primary database is registered with Oracle Clusterware.
- TDE-enabled database can also be registered. If TDE is configured using software keystore and the database version is 18c and higher, then TDE wallet management is also supported similar to a TDE database created using the Oracle Database Appliance framework.
- In this release, you cannot register a TDE-enabled database that uses Oracle Key Vault to store TDE keys.

Example 16-159 Registering a Migrated Database

The following is the syntax to register a single instance OLTP database that is using shape `odb1`.

```
odacli register-database -c OLTP -s odb1 -sn crmdb.example.com
Enter new 'SYS user' password:
Confirm 'SYS user' password:
{
  "jobId" : "317b430f-ad5f-42ae-bb07-13f053d266e2",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "September 8, 2024 05:55:49 AM EDT",
  "description" : "Database service registration with db service name:
crmdb.example.com",
  "updatedTime" : "September 8, 2024 05:55:49 AM EDT"
}
rpandrap: ]# odacli describe-job -i "317b430f-ad5f-42ae-bb07-13f053d266e2"

Job details
----------------------------------------------------------------
                     ID: 317b430f-ad5f-42ae-bb07-13f053d266e2
            Description: Database service registration with db service name:
crmdb.example.com
                  Status: Success
                 Created: September 8, 2024 5:55:49 AM EDT
                 Message:

Task Name                           Start Time
---------------------------- ------------------------------
restore control file           September 8, 2024 5:55:49 AM EDT
```

```
move spfile to right location  September 8, 2024 5:56:08 AM EDT
register DB with clusterware   September 8, 2024 5:56:13 AM EDT
reset db parameters            September 8, 2024 5:57:05 AM EDT
Running DataPatch              September 8, 2024 5:57:36 AM EDT

(Continued)
  End Time                     Status
------------------------------ --------
September 8, 2024 5:56:08 AM EDT   Success
September 8, 2024 5:56:13 AM EDT   Success
September 8, 2024 5:57:05 AM EDT   Success
September 8, 2024 5:57:36 AM EDT   Success
September 8, 2024 5:57:49 AM EDT   Success
```

odacli upgrade-database

You can use the `odacli upgrade-database` command to upgrade a database from a supported release.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To upgrade an Oracle Database:

```
 odacli upgrade-database -i Comma separated list of database ids -n Comma
separated list of database names -from source dbhome id -to destination
dbhome id | -fn source dbhome name -tn destination dbhome name [-ctm] [-j] [-
h] [-sh|-no-sh] [-ha|-no-ha]
```

Parameters

Parameter	Description
`--ids, -i`	(Optional) Defines the Database IDs to upgrade. You can use a comma separated list of database IDs.
`--names, -n`	(Optional) Defines the names of the databases to upgrade.
`--dest-dbhome-id, -to`	The DB HOME ID of the destination database home.
`--dest-dbhome-name, -tn`	The DB HOME name of the destination database home.
`--source-dbhome-id, -from`	The DB HOME ID of the source database home.
`--source-dbhome-name, -fn`	The DB HOME name of the source database home.
`--change-tdewallet-mgmt, -ctm`	(Optional) When specified, the `TDE wallet management` attribute is changed from `External` to `ODA`.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--disable-seha, -no-sh`	(Optional) Specifies whether you want to disable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.

Parameter	Description
`--enable-seha`, `-sh`	(Optional) Specifies whether you want to enable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases. **Note:** This option is deprecated and will be desupported in a future release.
`--disable-ha`, `-no-ha`	(Optional) Disables High Availability for single-instance Oracle Database 19c and later.
`--enable-ha`, `-ha`	(Optional) Enables High Availability for single-instance Oracle Database 19c and later.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

You cannot upgrade Oracle RAC or Oracle RAC One Node Database to a destination database home of Standard Edition 19c or later. You must first convert Oracle RAC or Oracle RAC One Node Database to single-instance Oracle Database using the `odacli modify-database` command and then upgrade the single-instance Oracle Database to a destination database home of Standard Edition 19c or later.

You have the following options for defining the databases to upgrade:

- When you define a single database ID, only the database of the source database home is upgraded.
- When you define more than one database ID, only those databases of the source database home are upgraded.
- When you do not define a database ID, all of the databases under the source database home are upgraded.

You can upgrade databases of the following releases:

- Oracle Database 12.2 to 19c
- Oracle Database 12.1 to 19c
- Oracle Database 12.1 to 12.2

Example 16-160 Upgrading an Oracle Database

In this example, a single database is upgraded. Use the command `odacli list-databases` to display the database ID, then use the command `odacli upgrade-databases` to upgrade the database. After the upgrade is complete, you can run the command `odacli list-databases` again to verify the upgrade.

```
# odacli  list-databases

ID                                       DB Name     DB Version  CDB
---------------------------------------  ----------  ----------  ----
fb4d02f3-2413-47ca-8584-a768e23ec2e7     ee12db      12.1.0.2    false

(Continued)
Class    Shape    Storage  Status
------ -------  --------  ----------
OLTP      odb1      ACFS     Configured
IMDB      odb1      ASM      Configured


# odacli upgrade-database -i ad6c7326-e460-411e-94df-230dedbef743 -from fa4321f5-0543-477d-bb54-
```

```
a429dcc8ee8d -to d752df28-ecdd-4af4-9454-38085ea17f8b
{
   "jobId" : "1bbe8boe-acb0-4296--9c8b-473b69da0c18",
   "status" : "Created",
   "message" : null,
   "reports" : [ ],
   "createTimestamp" : "October 24, 2017 03:54:03 AM EDT",
   "description" : "Database service upgrade with db ids: [ad6c7326-e460-411e-94df-230dedbef743]",
   "updatedTime" : "October 24, 2017  03:54:03 AM EDT"
}

# odacli  list-databases

ID                                       DB Name     DB Version  CDB
---------------------------------------  ----------  ----------  ----
ad6c7326-e460-411e-94df-230dedbef743   rdb121a     12.1.0.2    true
fb4d02f3-2413-47ca-8584-a768e23ec2e7   ee12db      12.1.0.2    false

(Continued)
Class    Shape    Storage  Status
------ -------  --------  ----------
OLTP      odb1      ACFS     Configured
IMDB      odb1      ASM      Configured
```

odacli DBHome Commands

Use the `odacli DBHome` commands to manage database Home operations.

- odacli configure-dbhome-storage
  Use the `odacli configure-dbhome-storage` command to configure the database home storage.
- odacli list-dbhome-storages
  Use the `odacli list-dbhome-storages` command to view the database home storage resources after the volumes for ORACLE_HOME and ORACLE_BASE are created.
- odacli describe-dbhome-storage
  Use the `odacli describe-dbhome-storage` command to view information such as how much space is available (at a volume or diskgroup level).
- odacli modify-dbhome-storage
  Use the `odacli modify-dbhome-storage` command to modify database home storage and add additional disk space.
- odacli create-dbhome
  Use the `odacli create-dbhome` command to create an Oracle Database Home.
- odacli delete-dbhome
  Use the `odacli delete-dbhome` command to delete database home that is not associated with a database.
- odacli describe-dbhome
  Use the `odacli describe-dbhome` command to display Oracle Database Home details.
- odacli list-dbhomes
  Use the `odacli list-dbhomes` command to display a list of Oracle Home directories.

odacli configure-dbhome-storage

Use the `odacli configure-dbhome-storage` command to configure the database home storage.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli configure-dbhome-storage -dg` *`disk_group`* `[-s` *`volume_size`*`] [-j] [-h]`

Parameters

Parameter	Description
`--diskgroup, -dg`	(Mandatory) Defines the name of the disk group.
`--size, -s`	(Optional) Defines the size, in gigabytes (GB), of the ORACLE_HOME volume. The default is: 80 GB. The minimum value that can be specified is 25GB and the maximum value is 1024GB.
`--json, -j`	(Optional) Displays the output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- Valid values for the `--diskgroup` option vary depending on the hardware. For the X5-2, X6-2 and X7-2 HA platforms, you can select between the DATA or RECO diskgroup. For all other platforms such as X8-2, you must select only the DATA diskgroup. With later hardware, you must still run the `odacli configure-dbhome-storage` command.
- When you run the command to create a new database home, it also creates the volumes to hold the database homes. Till the volumes are created, you can run the `odacli configure-dbhome-storage` command to change the diskgroup or the inital size of the volume. Once the volumes for this feature are created, running this command results in an error:

```
# odacli configure-dbhome-storage -dg RECO
DCS-10606:Database homes storage has already been created, no changes are
allowed.
```

Example 16-161 Creating Database Home Storage

```
# odacli configure-dbhome-storage -dg DATA
{
  "jobId" : "553bfeb9-0092-4d73-bbd4-43fcb590886f",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 11, 2021 06:48:12 AM GMT",
  "resourceList" : [ ],
  "description" : "Configure database home storage",
```

```
  "updatedTime" : "May 11, 2021 06:48:12 AM GMT"
}
```

odacli list-dbhome-storages

Use the `odacli list-dbhome-storages` command to view the database home storage resources after the volumes for ORACLE_HOME and ORACLE_BASE are created.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-dbhome-storages [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays the output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

- Starting with Oracle Database Appliance release 19.25, Oracle Database homes are no longer created on the local disk. New Oracle Database homes are created on an Oracle ACFS-managed file system. To create an Oracle ACFS file system, ADVM volumes are created.
- The command lists the resource ID, the node on which the volume is created, and the diskgroup on which the volume is created.

Example 16-162 Listing Database Home Storage on a Single-Node System

```
# odacli list-dbhome-storages

================================================================================
================================
ID                                   Node Description          Disk Group
Volume      Size(GB)    Status
------------------------------------ ---- -------------------- ----------
---------- ---------- ----------
6801cabb-4e45-41d7-9f20-c75dc7cf30c2   0    ORACLE_HOME          DATA
orahome_sh  80         CONFIGURED
f5ad1258-a4a1-4cc1-980c-e2eeca738d99   0    ORACLE_BASE          DATA
odabase_n0  80         CONFIGURED
================================================================================
================================
```

Example 16-163 Listing Database Home Storage on a High-Availability System

```
# odacli list-dbhome-storages

================================================================================
```

```
==============================
ID                                       Node Description          Disk Group
Volume      Size(GB)    Status
---------------------------------------- ---- -------------------- ----------
---------- ---------- ----------
67622ce6-0a00-4fec-b948-7a0ba7922311   0,1  SHARED ORACLE_HOME   DATA
orahome_sh  200        CONFIGURED
ac813348-2935-4903-b9a6-eb4f0d9637b3   0    ORACLE_BASE          DATA
odabase_n0  80         CONFIGURED
c8153e7b-6ab7-4a55-90ad-20d4196d2ac4   1    ORACLE_BASE          DATA
odabase_n1  80         CONFIGURED
================================================================================
==============================
```

odacli describe-dbhome-storage

Use the `odacli describe-dbhome-storage` command to view information such as how much space is available (at a volume or diskgroup level).

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-dbhome-storage -i db_home_id [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	(Mandatory) Specifies the Oracle home storage ID. This is the ID listed in the odacli list-dbhome-storages command output.
`--json, -j`	(Optional) Displays the output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-164 Describing Database Home Storage

```
odacli describe-dbhome-storage -i 67622ce6-0a00-4fec-b948-7a0ba7922311
ORACLE_HOME storage details:
================================================================
                     ID: 67622ce6-0a00-4fec-b948-7a0ba7922311
                  State: CONFIGURED
                Created: February 11, 2021 5:30:10 AM UTC
                Updated: February 18, 2021 4:33:30 PM UTC

Volume Details
--------------
                   Name: ORAHOME_SH
             Disk Group: DATA
            Description: SHARED ORACLE_HOME
            Node Number: 0,1
        Allocated Space: 200.0 GB
```

```
              Used Space: 12.10 GB
              Free Space: 187.89 GB
                Location: /u01/app/odaorahome

DiskGroup Details
-----------------
                    Name: DATA
           Allocated Space: 15.52 TB
                Used Space: 2.69 TB
                Free Space: 12.83 TB
================================================================
```

odacli modify-dbhome-storage

Use the `odacli modify-dbhome-storage` command to modify database home storage and add additional disk space.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli modify-dbhome-storage -i` *`db_home_id`* `-s` *`size`*`[-j] [-h]`

Parameters

Parameter	Description
`--ohomeid, -i`	(Mandatory) Specifies the Oracle home storage ID. This is the ID listed in the `odacli list-dbhome-storages` command output.
`--size, -s`	(Mandatory) Specifies the new size in GB. The minimum value that can be specified is 25GB and the maximum value is 1024GB.
`--json, -j`	(Optional) Displays the output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-165 Modifying Database Home Storage

```
odacli modify-dbhome-storage -i 9b9c9e0f-db04-4985-be48-3f1e263cd73a -s 30
{
  "jobId" : "badfad81-cfc1-4c82-a181-69ebdfb11c22",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "May 11, 2021 14:04:09 PM CST",
  "resourceList" : [ ],
  "description" : "Modify database home storage",
  "updatedTime" : "May 11, 2021 14:04:09 PM CST"
}
```

odacli create-dbhome

Use the `odacli create-dbhome` command to create an Oracle Database Home.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To create an Oracle Database Home:

```
odacli create-dbhome -v version [-j] [-h]
```

Parameters

Parameter	Description
`-v` *version number*	Defines the database bundle patch number.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

For the version number, you can specify the database version, such as 18.7.0.0, 18.5.0.0, 12.2.0.1, or 12.1.0.2, or you can use a 5 digit format to specify a specific patch bundle version. For example, 19.25.0.0.241015. If you use the database version without specifying the bundle patch number, then the latest bundle patch is used.

Example 16-166 Creating an Oracle Database Home

The following example creates an Oracle Database Home version 19.25.0.0.241015.

```
# odacli  create-dbhome -v 19.25.0.0.241015
```

odacli delete-dbhome

Use the `odacli delete-dbhome` command to delete database home that is not associated with a database.

File Path

/opt/oracle/dcs/bin/odacli

Syntax

```
odacli delete-dbhome  -i db_home_ID | -n db_home_name [-h]
```

Parameters

Parameter	Description
`--id, -i`	Identifies the database home using a database identifier (ID).
`--name, -n`	Identifies the database home using a database name.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- Use the `odacli list-dbhomes` command to locate the identifier.
- To delete, or uninstall, a database home (dbhome), there must not be any associated databases in the dbhome.
- Use the `odacli delete-database` command to delete an existing database.

Example 16-167 Deleting an Empty Database Home

```
# odacli delete-dbhome -i 0ce547ca-3df2-4178-a7e6-eefa613aeab4
```

odacli describe-dbhome

Use the `odacli describe-dbhome` command to display Oracle Database Home details.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To display details about Oracle Database Home:

```
odacli describe-dbhome -i dbhomeid -n name [-h] [-j] [-v]
```

Parameters

Parameter	Description
`--id, -i`	Identifies the database home ID. Use the `odacli list-dbhomes` command to get the *id*.
`--name, -n`	Identifies the database home name. Use the `odacli list-dbhomes` command to get the database name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--dbversion, -v`	(Optional) Identifies the Database Home Version. Use the `odacli list-dbhomes –v` command to get the *dbversion*.

Example 16-168 Displaying Oracle Database Home Details

The following output is an example of using the display Oracle Database Home details command:

```
# odacli  describe-dbhome -i b727bf80-c99e-4846-ac1f-28a81a725df6

DB Home details
----------------------------------------------------------------
                            ID: b727bf80-c99e-4846-ac1f-28a81a725df6
                          Name: OraDB19_home1
                       Version: 19.19.0.0.0
                 Home Location: /u01/app/orauser/product/19/dbhome_1
                       Created: Jun 2, 2023 10:19:23 AM
```

odacli list-dbhomes

Use the `odacli list-dbhomes` command to display a list of Oracle Home directories.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To display a list of Oracle Home directories:

```
odacli  list-dbhomes  [-h] [-j]
```

Parameters

Parameter	Description
--help,-h	(Optional) Displays help for using the command.
--json,-j	(Optional) Displays JSON output.
-all	(Optional) This option is available only when multi-user access is enabled. Provides a read-only view of all the database homes in the system.

Example 16-169 Displaying a List of Oracle Home Directories

Run the following command to display a list of Oracle Home directories:

```
# odacli list-dbhomes


ID                                       Name                 DB Version
---------------------------------------- -------------------  ----------
b727bf80-c99e-4846-ac1f-28a81a725df6 OraDB12102_home1  12.1.0.2

(continued)
Home Location
----------------------------------------
/u01/app/orauser/product/12.1.0.2/dbhome_1
```

odacli Database Storage Commands

Use the Database Storage commands to list, describe, create, and delete Oracle database storage.

- odacli create-dbstorage
  Use the `odacli create-dbstorage` command to create the file system for database migrations.
- odacli delete-dbstorage
  Use the `odacli delete-dbstorage` command to delete database storage that is not associated with a database.
- odacli describe-dbstorage
  Use the `odacli describe-dbstorage` command to display storage configuration details.
- odacli describe-dgstorage
  Use the `odacli describe-dgstorages` command to display the disk group information.
- odacli list-dbstorages
  Use the `odacli list-dbstorages` command to display a list of all of the database storage configured in the appliance.
- odacli list-dgstorages
  Use the `odacli list-dgstorages` command to display the disk group information.
- odacli modify-dbstorage
  Use the `odacli modify-dbstorage` command to resize the database storage.
- odacli modify-dgstorage
  Use the `odacli modify-dgstorage` command to release disks.

odacli create-dbstorage

Use the `odacli create-dbstorage` command to create the file system for database migrations.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli create-dbstorage  -n dbname -s dataSize -u databaseUniqueName -r [ASM|
ACFS] -rd [high|mirror] [-h]
```

Parameters

Parameter	Description
`--dbname`, `-n`	Defines the name of the database.

Parameter	Description
`--dataSize, -s`	(Optional) Defines the size, in gigabytes (GB), of the filesystem for storing database files and temp files. The default is: 100 GB The minimum size is 10 GB. When entering the size, do not include GB. For example, for 50 GB, use 50.
`--databaseUniqueName, -u`	(Optional) Defines a unique name for the database. Specify only if the database unique name is different than the database name. The command creates the following mount point: `/u02/app/oracle/oradata/`*`db unique name`*
`--dbstorage, -r [ASM|ACFS]`	(Optional) Defines the type of database storage, either ASM or ACFS. When you select ASM, the command only creates the supporting directory structure for storing non-database files. The default is ASM.
`--db-redundancy, -rd`	(Optional) Defines the database redundancy. The values can be high or mirror. The database redundancy is not applicable to database storages of Oracle ASM type, or database storages on normal or high redundancy disk groups. For database storages of Oracle ACFS type, specifying this option creates volumes of the specified redundancy on Oracle ASM disk groups of flex redundancy.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- The `odacli create-dbstorage` command registers the storage metadata with the Appliance Manager.
- Oracle Database is supported on both Oracle Automatic Storage Management (Oracle ASM) and Oracle ASM Cluster file system (ACFS). The default is Oracle ASM.
- When you create ACFS database storage, the command creates a separate ACFS file system and creates the directory structure for other database files, such as archives and online logs.
- When you create ASM database storage, the command only creates the corresponding directories for non-database files.

Example 16-170 Creating Database Storage

The following statement creates 50 GB ACFS database storage for the APPSDB database.

```
# odacli create-dbstorage -n APPSDB -s 50 -r ACFS
{
   "jobId" : "fc6bf8fd-60c2-44f3-b8b7-efd0e9a2149f",
   "status" : "Created",
   "message" : null,
   "reports" : [ ],
   "createTimestamp" : "August 09, 2016 06:19:35 AM WSST",
   "description" : "Database storage service creation with db name: APPSDB",
   "updatedTime" : "August 09, 2016 06:19:35 AM WSST"
}
```

odacli delete-dbstorage

Use the `odacli delete-dbstorage` command to delete database storage that is not associated with a database.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli delete-dbstorage -i ID -n name [-h] [-j]
```

Parameters

Parameter	Description
`--id, -i`	Identifies the database storage using a database identifier (ID).
`--name, -n`	Specifies the database storage name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- You can only use the `delete-dbstorage` when the storage is not associated with any databases.
- Use the `delete-database` command to delete an existing database.
- Use the `list-dbstorages` command to locate the identifier.

Example 16-171 Deleting Empty Database Storage

```
# odacli delete-dbstorage  -i 9fe39332-cc1a-4b4b-8393-165524a6ef6b
```

odacli describe-dbstorage

Use the `odacli describe-dbstorage` command to display storage configuration details.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-dbstorage  -i ID -n name [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	Specifies the database storage ID.

Parameter	Description
`--name, -n`	Specifies the database storage name.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- Use the `odacli list-dbstorages` command to get the database storage identifier (ID).
- The DATA Location corresponds to the `init.ora` parameter `db_create_file_dest`.
- RECO Location corresponds to the `init.ora` parameter `db_recovery_file_dest`
- REDO Location corresponds to the `init.ora` parameter `db_create_online_log_dest_1`

Example 16-172 Displaying Database Oracle ACFS Storage Details

The following example displays Oracle ASM Cluster file system (ACFS) storage details:

```
# odacli describe-dbstorage -i 9fe39332-cc1a-4b4b-8393-165524a6ef6b

DBStorage details
----------------------------------------------------------------
ID: 9fe39332-cc1a-4b4b-8393-165524a6ef6b
DB Name: rdb121a
DBUnique Name: rdb121a
DB Resource ID: ad6c7326-e460-411e-94df-230dedbef743
Storage Type: Acfs
DATA Location: /u02/app/oracle/oradata/rdb121a
RECO Location: /u03/app/oracle/fast_recovery_area/
REDO Location: /u03/app/oracle/redo/
State: ResourceState(status=Configured)
Created: July 22, 2016 12:07:12 PM SGT
UpdatedTime: July 22, 2016 12:26:39 PM SGT
```

Example 16-173 Displaying Database Oracle ASM Storage Details

The following example displays Oracle Automatic Storage Management (Oracle ASM) storage details:

```
# odacli describe-dbstorage -i 4f2a1b59-ca66-4d80-951c-425ab7b0acae

DBStorage details
----------------------------------------------------------------
ID: 4f2a1b59-ca66-4d80-951c-425ab7b0acae
DB Name: ee12db
DBUnique Name: ee12db
DB Resource ID: fb4d02f3-2413-47ca-8584-a768e23ec2e7
Storage Type: Asm
DATA Location: DATA
RECO Location: RECO
REDO Location: RECO
State: ResourceState(status=Configured)
```

```
Created: July 22, 2016 1:13:51 PM SGT
UpdatedTime: July 22, 2016 1:13:52 PM SGT
```

odacli describe-dgstorage

Use the `odacli describe-dgstorages` command to display the disk group information.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli describe-dgstorage  -d diskgroup -h
```

Parameters

Parameter	Description
--diskgroup, -d	Specifies the disk group for which information is displayed.
--help, -h	(Optional) Displays help for using the command.

Usage Notes

This command displays the storage information for all the databases backed by a disk group, for each disk group present in the system.

Example 16-174 Displaying Disk Group Storage

```
# odacli list-dgstorages -r  redundancy
Diskgroup storage details

--------------------------------------------------------------------------------
-----

Diskgroup: DATA

                  Redundancy: extern|normal|high|flex

                  Physical Free Space: wGB

                  Physical Total Space: xGB

                  Database Unique Name: rdbx



                                      Location: /u02/app/test/test1/rdbx

                                      Used Space: dGB  (acfsutil.total -
acfsutil.freespace)

                                      Free Space: dGB  (acfsutil.freespace)
```

```
                    Database Unique Name: rdby

                                        Location: +DATA/rdby

                                        Used Space: zGB  (v$asm_file.bytes)

                                        Free Space: xGB   (PhyFree/redundancy)

                                        Physical Space:
vGB(v$asm_diskgroup.total_mb)

                                        Physical Used : wGB (v$asm_file.space)

                                        Physical Free :
uGB(v$asm_diskgroup.free_mb)

                                        Physical Reserved : uGB(Lookup Table)

Diskgroup: RECO

                     Redundancy: normal|high|flex

                     Physical Free Space: wGB

                     Physical Total Space: xGB

                     Database Unique Name: rdbx

                                        Location: /u02/app/test/fast_recovery_area



                                        Used Space: dGB  (acfsutil.total -
acfsutil.freespace)

                                        Free Space: dGB  (acfsutil.freespace)

                    Database Unique Name: rdby

                                        Location: +DATA/rdby

                                        Used Space: zGB  (v$asm_file.bytes)

                                        Free Space: xGB   (PhyFree/redundancy)

                                        Physical Space:
vGB(v$asm_diskgroup.total_mb)

                                        Physical Used : wGB (v$asm_file.space)

                                        Physical Free :
uGB(v$asm_diskgroup.free_mb)

                                        Physical Reserved : uGB(Lookup Table)
```

odacli list-dbstorages

Use the `odacli list-dbstorages` command to display a list of all of the database storage configured in the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli list-dbstorages  [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

This command displays a list of all of the filesystems that are configured with the `create-database` command and the `create-dbstorage` command.

Example 16-175 Displaying a List of all Database Storage

```
# odacli list-dbstorages

ID Type                                  DBUnique  Name         Status
---------------------------------------  --------  -----------  ----------
9fe39332-cc1a-4b4b-8393-165524a6ef6b  Acfs       rdb121a     Configured
4f2a1b59-ca66-4d80-951c-425ab7b0acae  Asm        ee12db      Configured
0266edac-c729-4539-861f-3f3d543be9e4  Acfs       db12SE      Configured
```

odacli list-dgstorages

Use the `odacli list-dgstorages` command to display the disk group information.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli list-dgstorages  -r` *`redundancy`* `-h`

Parameters

Parameter	Description
`--redundancy`, `-r`	Specifies the disk group redundancy for FLEX disk group.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

- The command displays the storage-related information for all the disk groups present in the system.
- The redundancy option is valid only for FLEX disk groups. For all disk groups that are not FLEX, logical free space is calculated using the redundancy obtained from v$asm_diskgroup rather than the CLI option.
- Redundancy option can take the values MIRROR | HIGH. Use one of these options to specify the database redundancy for FLEX disk groups.
- In the example below, Physical Total Space is the total space in the disk group. (V$ASM_DISKGROUP.total_mb) Physical Reserved Space is the reserved space required to handle disk failures. Physical Free Space is the free space in the disk group. (V$ASM_DISKGROUP.free_mb) Logical Free Space is the logical free space in the diskgroup. (V$ASM_DISKGROUP.free_mb / redundancy)

This command displays the storage information for all the databases backed by a disk group, for each disk group present in the system.

Example 16-176 Displaying Disk Group Storage

```
# odacli list-dgstorages -r  redundancy
Diskgroup  Redundancy   Physical Total Space  Physical Reserved Space
Physical Free Space  Logical Free
 Space

--------------   -----------------    ----------------------------
---------------------------------
DATA           FLEX                xGB
yGB
        zGB                                    uGB

REDO          HIGH                xGB
yGB
        zGB                                    uGB

RECO          NORMAL          xGB
yGB
  zGB                                  uGB
```

odacli modify-dbstorage

Use the `odacli modify-dbstorage` command to resize the database storage.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli modify-dbstorage [-ds] [-in] [-fs] [-rcs] [-rds] [-h] [-j]
```

Parameters

Parameter	Description
`--datasize, -ds`	(Optional) Specifies the extendable size in GB for the DATA volume. Specify at least one size from FLASH, RECO, REDO, or DATA volumes.
`--dbname, -in`	Defines the database name on which you want to extend the volumes.
`--flashsize, -fs`	(Optional) Specifies the extendable size in GB for the FLASH volume.
`--recosize, -rcs`	(Optional) Specifies the extendable size in GB for the RECO volume.
`--redosize, -rds`	(Optional) Specifies the extendable size in GB for the REDO volume.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage Notes

- The command can resize DATA, RECO, REDO, and FLASH volumes.
- Database name is a mandatory parameter. All other parameters are optional, however, one of the volume sizes must be specified. If you only provide the DATA size, then the DCS agent calculates the recommended RECO size for the given DATA size is internally calculated. The storage is then extended with the recommended values.
- The mounted Oracle ACFS and its underlying Oracle ADVM storage is extended by the specified value.
- The maximum resize value that can be specified in the command at a time is 100 GB. To resize with a value greater than 100 GB, run the command multiple times with a maximum value of 100 GB each time. For further resize, ensure that the command completes successfully before you run it again.
- To check the result of the resize operation, use the `odacli list-dbstorages` command.
- After database resize operation is completed, use the `odacli describe-dbstorage` command to display storage configuration details.

Example 16-177 Modifying the Database Storage

```
# odacli modify-dbstorage -ds 10 -rcs 15 -n amdb1
{
 "jobId" : "e6d54155-ed4c-4fd6-a3f6-454120fa5c18",
 "status" : "Created",
 "message" : null,
 "reports" : [ ],
 "createTimestamp" : "June 03, 2019 06:06:42 AM UTC",
 "resourceList" : [ ],
 "description" : "Database storage modify operation",
 "updatedTime" : "June 03, 2019 06:06:42 AM UTC"
}
```

odacli modify-dgstorage

Use the `odacli modify-dgstorage` command to release disks.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To release disks:

```
odacli modify-dgstorage [-f] [-rd list] [-j] [-h]
```

Parameters

Parameter	Description
`--force`, `-f`	(Optional) Forces release of specified disks. When this option is specified, the check to verify that all disk groups are mounted on the current node is skipped.
`--release-disk`, `-rd`	Specifies the disks to release. Generally, this specifies a single storage device with partitions on different disk groups. You can also specify a comma-separated list of disks to release.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

- Run the `odacli list-dgdisks` to view details of the disk, before running the `odacli modify-dgstorage` command.
- The `odacli modify-dgstorage` command runs a precheck to ensure all disk groups are mounted on the node. You can use the `-f` option to override the precheck.

Example 16-178 Releasing a physical disk HDD_E0_S10_577135552 which has two logical partitions on two different disk groups

```
# odacli modify-dgstorage -rd HDD_E0_S10_577135552
Job details
----------------------------------------------------------------
                     ID:  46bc736c-417b-4ac2-826e-aa857db50e29
            Description:  Release disks: [HDD_E0_S10_577135552]
                 Status:  Running
                Created:  September 8, 2024 6:24:27 PM PDT
                Message:  Release ASM disks

Verifying the result of the Release Disks operation:
# odacli describe-job -i 46bc736c-417b-4ac2-826e-aa857db50e29
Job details
----------------------------------------------------------------
                     ID:  46bc736c-417b-4ac2-826e-aa857db50e29
            Description:  Release disks: [HDD_E0_S10_577135552]
```

```
 Status:  Success
Created:  September 8, 2024 6:24:27 PM PDT
Message:  Release ASM disks
```

odacli Diagnostics Collection Commands

Use the diagnostics collection commands to display diagnostics details.

- odacli create-diagcollection
  Use the `odacli create-diagcollection` command to create diagnostics collection data from the appliance.
- odacli describe-diagcollection
  Use the `odacli describe-diagcollection` command to describe diagnostics collection data from the appliance.
- odacli list-diagcollections
  Use the `odacli list-diagcollections` command to list the diagnostics collections.
- odacli delete-diagcollection
  Use the `odacli delete-diagcollection` command to delete diagnostics collection data from the appliance.

odacli create-diagcollection

Use the `odacli create-diagcollection` command to create diagnostics collection data from the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To create diagnostics collection:

```
odacli create-diagcollection -i id [-d] [-t] [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	(Mandatory) Specifies the DCS job ID for the collection.
`--description, -d`	(Optional) Description of the diagnostic collection.
`--tag, -t`	(Optional) Specifies tag name to identify the collection.
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-179 Creating a diagnostics collection

```
# odacli create-diagcollection -i ID
```

```
Job details
----------------------------------------------------------------
                     ID:  f840feb8-7712-49a1-a525-1ee911176c50
            Description:  Diagnostic collection creation
                 Status:  Created
                Created:  April 11, 2023 7:15:30 PM UTC
                Message:
```

odacli describe-diagcollection

Use the `odacli describe-diagcollection` command to describe diagnostics collection data from the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To describe diagnostics collection:

```
odacli describe-diagcollection -i id [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	(Mandatory) Specifies the diagnostics collection ID.
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-180 Describing a diagnostics collection

```
# odacli describe-diagcollection -i ID

Diagnostic collection
details
--------------------------------------------------------------------------------
--
                        ID: f5c07af8-2635-466d-b487-0f04d15fba79
         Requested job ID: 016e74cb-01e0-4812-96ef-94602b5ea2c5
              Description: Diagnostic collection for job 'Agent State Update
Job'
     Collection file path: /opt/oracle/dcs/commonstore/diag/2023/04/
f5c07af8-2635-466d-b487-0f04d15fba79.tar.gz
          Collection size: 91.88 KB
               Start time: February 27, 2023 7:58:41 PM UTC
                 End time: February 27, 2023 7:59:39 PM UTC
```

odacli list-diagcollections

Use the `odacli list-diagcollections` command to list the diagnostics collections.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To list all diagnostics collections:

```
odacli list-diagcollections [-j] [-h]
```

Parameters

Parameter	Description
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-181 Listing all diagnostics collections

```
ID
Description                                                     Collection
size  Start time                      End time
------------------------------------
------------------------------------------------------------
---------------- ---------------------------------- ------------------------------
3c65bcd3-cd30-4332-88b7-fc25b256db75     Diagnostic collection for job
'DcsAdmin patching'                  8.37 MB          2023-03-08 15:37:02
UTC        2023-03-08 15:38:01 UTC
7dd1d951-056c-47b2-ad78-8614d0b3bbd0     Diagnostic collection for job 'VM
winvm1 start'                     6.59 MB          2023-03-08 15:37:02
UTC        2023-03-08 15:38:01 UTC
```

odacli delete-diagcollection

Use the `odacli delete-diagcollection` command to delete diagnostics collection data from the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To delete diagnostics collection:

```
odacli delete-diagcollection -i id [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	(Mandatory) Specifies the diagnostics collection ID to be deleted.
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-182 Deleting a diagnostics collection

```
# odacli delete-diagcollection -i ID
```

odacli Job Commands

Use the `odacli list-jobs` and `odacli describe-job` commands to display job details.

- odacli create-jobdata-retention-policy
  Use the `odacli create-jobdata-retention-policy` command to create job data retention policy for default DCS jobs and specific auto scheduled jobs.
- odacli create-purge-jobdata-job
  Use the `odacli create-purge-jobdata-job` command to create a job to delete data for default DCS jobs and specific auto scheduled jobs.
- odacli delete-jobdata-retention-policy
  Use the `odacli delete-jobdata-retention-policy` command to delete job data retention for default DCS jobs and specific auto scheduled jobs.
- odacli describe-job
  Use the `odacli describe-job` command to display details about a specific job, including the job ID, status, tasks, and the job created date and time stamp.
- odacli list-jobs
  Use the `odacli list-jobs` command to display a list of jobs, including the job IDs, status, and the job created date and time stamp. The default number of jobs listed is 50 from the earliest to the latest.
- odacli list-pendingjobs
  Use the `odacli list-pendingjobs` command to list all pending jobs.
- odacli list-jobdata-retention-policy
  Use the `odacli list-jobdata-retention-policy` command to list job data retention policy for default DCS jobs and specific auto scheduled jobs.
- odacli list-purge-jobdata-jobs
  Use the `odacli list-purge-jobdata-jobs` command to list purge job data for default DCS jobs and specific auto scheduled jobs.
- odacli list-scheduled-executions

odacli create-jobdata-retention-policy

Use the `odacli create-jobdata-retention-policy` command to create job data retention policy for default DCS jobs and specific auto scheduled jobs.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To create job data retention policy:

```
odacli create-jobdata-retention-policy [-j] [-h] [-w] [-s]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.
`--retentionWindow`, `-w`	(Optional) Describes the time interval to retain the job data. Default is 30 days if not specified.
`--retentionWindowUnit`, `-u`	(Optional) Describes the unit for the `retentionWindow`. Possible values are `Day|Hour|Minute`. Default is `Day` if no value is specified.
`--scheduleIds`, `-s`	(Optional) Specifies the IDs, delimited by commas, of the scheduled jobs similar to the `odacli list-schedules` result. Use `Default` for all DCS jobs.

Usage Notes

- Once a policy is created, an auto scheduled job is created to start purging DCS job records regularly, for example, every day at 4 AM, based on the retention policies you define.

Example 16-183 Creating a default policy to purge all the jobs and scheduled executions older than 30 days

```
# odacli create-jobdata-retention-policy

Job details
----------------------------------------------------------------
ID: 36d00489-39b1-425f-81bf-6d6e3bfede8b
Description: set job data retention policy
Status: Created
Created: April 23, 2020 3:32:09 PM PDT
Message:

Task Name Start Time End Time Status
---------------------------------------- --------------------------------
---------------------------------- ----------
[root@oda1]# ./odacli describe-job -i c40c17f0-4185-40fe-a126-30fe1f0e5cdd
```

```
Job details
----------------------------------------------------------------
ID: c40c17f0-4185-40fe-a126-30fe1f0e5cdd
Description: set job data retention policy
Status: Success
Created: April 23, 2020 5:42:28 PM PDT

Message:
Task Name Start Time End Time Status
------------------------------------ ---------------------------------
---------------------------------- ----------
set job data retention policy   April 23, 2020 5:42:28 PM PDT   April 23,
2020 5:42:28 PM PDT   Success
create auto job data schedule   April 23, 2020 5:42:28 PM PDT   April 23,
2020 5:42:28 PM PDT   Success
```

Example 16-184 Set up a retention policy of 10 days for the schedule 'archive_log_backup'

```
# odacli create-jobdata-retention-policy -s 284a22b5-41e6-4be1-
a916-048eac74e076 -w 10 -u Day

Job details
----------------------------------------------------------------
ID: ea68a90e-a8e2-4fc4-a4f1-75461e12c387
Description: set job data retention policy
Status: Created
Created: April 23, 2020 6:22:14 PM PDT

Message:
Task Name Start Time End Time Status
------------------------------------ ---------------------------------
---------------------------------- ----------
[root@oda1]# ./odacli list-jobdata-retention-policy

Schedule Id     Retention Window    Unit    Modified
------------------- ------------------- --------------------
------------------------
Default     20      Day     April 23, 2020 6:16:59 PM PDT
284a22b5-41e6-4be1-a916-048eac74e076    10  Day     April 23, 2020 6:22:14 PM
PDT
```

odacli create-purge-jobdata-job

Use the `odacli create-purge-jobdata-job` command to create a job to delete data for default DCS jobs and specific auto scheduled jobs.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To create job data purge:

```
odacli create-purge-jobdata-job [-j] [-h] [-i] [-o] [-u] [-s]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.
`--jobIds`, `-i`	(Optional) Purges data for given DCS job IDs, delimited by commas.
`--olderThan`, `-o`	(Optional) Purges job data older than the specified time. If not specified, use the existing `retentionWindow` value.
`--olderThanUnit`, `-u`	(Optional) Describes the unit for the `OlderThan` option. Possible values are `Day|Hour|Minute`. If not specified, use the existing `retentionWindow` value.
`--scheduleIds`, `-s`	(Optional) Specifies the IDs, delimited by commas, of the scheduled jobs similar to the `odacli list-schedules` result.

Example 16-185 Creating a job to purge job data

```
# odacli create-purge-jobdata-job

Job details
----------------------------------------------------------------
ID: e43dbd56-dc7c-4812-b653-0fc3208e0929
Description: purge job data
Status: Created
Created: April 23, 2020 6:27:40 PM PDT

Message:
Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
[root@oda1]# odacli describe-job -i e43dbd56-dc7c-4812-b653-0fc3208e0929
Job details
----------------------------------------------------------------
ID: e43dbd56-dc7c-4812-b653-0fc3208e0929
Description: purge job data

Status: Success
Created: April 23, 2020 6:27:40 PM PDT
Message: Purged 21 metadata records and 10 files. Purge policies are:
Purge all jobs and scheduled executions older than 30 Days
Purge scheduled executions for ID:284a22b5-41e6-4be1-a916-048eac74e076 older
than 10 Day
```

```
Task Name    Start Time  End Time    Status
------------------------------------ ----------------------------------
---------------------------------- ----------
purge job data  April 23, 2020 6:27:40 PM PDT   April 23, 2020 6:27:40 PM
PDT   Success
```

Example 16-186 Purging job data older than 5 hours, including all jobs and scheduled executions related records

```
# odacli create-purge-jobdata-job -o 5 -u Hour
Job details
----------------------------------------------------------------
ID: 4d065c96-a8eb-4974-a446-573b5ad3cb95
Description: purge job data
Status: Created
Created: April 23, 2020 1:55:59 PM PDT
Message:
Task Name Start Time End Time Status

------------------------------------ ----------------------------------
---------------------------------- ----------
[root@oda1]# odacli describe-job -i 4d065c96-a8eb-4974-a446-573b5ad3cb95

Job details
----------------------------------------------------------------
ID: 4d065c96-a8eb-4974-a446-573b5ad3cb95
Description: purge job data
Status: Success
Created: April 23, 2020 1:55:59 PM PDT
Message: Purged 242 metadata records and 0 files. Purge policies are:
Purge all jobs and scheduled executions older than 5 Hour

Task Name Start Time End Time Status
------------------------------------ ----------------------------------
---------------------------------- ----------
purge job data      April 23, 2020 1:55:59 PM PDT       April 23, 2020
1:55:59 PM PDT       Success
```

Example 16-187 Purging job data for multiple jobs

```
# odacli create-purge-jobdata-job -i 87a926cd-a243-49e0-
ba02-9d8968e1de1e,0e2a38c0-041a-41d8-91ad-7f9f7c646f12

Job details
----------------------------------------------------------------
ID: 217cbdbf-15a5-47d3-9582-3f26d6b5eebf
Description: purge job data
Status: Created
Created: April 23, 2020 2:01:25 PM PDT
Message: Task Name Start Time End Time Status
------------------------------------ ----------------------------------
---------------------------------- ----------

[root@oda1]# odacli describe-job -i 217cbdbf-15a5-47d3-9582-3f26d6b5eebf
Job details
```

```
------------------------------------------------------------
ID: 217cbdbf-15a5-47d3-9582-3f26d6b5eebf
Description: purge job data
Status: Success
Created: April 23, 2020 2:01:25 PM PDT
Message: Purged 11 metadata records and 0 files. Purge policies are:
Purge job with ID:87a926cd-a243-49e0-ba02-9d8968e1de1e
Purge job with ID:0e2a38c0-041a-41d8-91ad-7f9f7c646f12

Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
purge job data      April 23, 2020 2:01:25 PM PDT       April 23, 2020
2:01:25 PM PDT       Success
```

Example 16-188 Purging job data for specific job data schedules older than 5 days

```
# odacli create-purge-jobdata-job -s 39cc86d6-30c2-4744-bfbc-
e62535e2aa67,31b3a83b-76ca-44d3-9268-b4dac948e217 -o 5 -u Day

Job details
------------------------------------------------------------
ID: be839a81-a79b-43ce-bf70-c8aba1a9cc91
Description: purge job data
Status: Created
Created: April 23, 2020 2:44:54 PM PDT
Message:

Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
[root@oda1]# ./odacli describe-job -i be839a81-a79b-43ce-bf70-c8aba1a9cc91

Job details
------------------------------------------------------------
ID: be839a81-a79b-43ce-bf70-c8aba1a9cc91
Description: purge job data
Status: Success
Created: April 23, 2020 2:44:54 PM PDT
Message: Purged 0 metadata records and 0 files. Purge policies are:
Purge scheduled executions for ID:39cc86d6-30c2-4744-bfbc-e62535e2aa67 older
than 5 Day
Purge scheduled executions for ID:31b3a83b-76ca-44d3-9268-b4dac948e217 older
than 5 Day
Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
purge job data April 23, 2020 2:44:54 PM PDT April 23, 2020 2:44:54 PM PDT
Success
```

odacli delete-jobdata-retention-policy

Use the `odacli delete-jobdata-retention-policy` command to delete job data retention for default DCS jobs and specific auto scheduled jobs.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To delete job data retention:

```
odacli delete-jobdata-retention-policy [-j] [-h][-s]
```

Parameters

Parameter	Description
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.
`--scheduleIds, -s`	(Optional) Specifies the IDs applied by the job data retention policy. If not supplied, delete all the existing policies.

Example 16-189 Deleting the job data retention policies

```
# odacli delete-jobdata-retention-policy

Job details
----------------------------------------------------------------
ID: ab819d57-9e35-4f55-bd5e-d26fc6372448
Description: delete job data retention policy
Status: Running
Created: April 23, 2020 7:34:05 PM PDT

Message:
Task Name Start Time End Time Status
---------------------------------------- --------------------------------
---------------------------------- ----------
remove job data retention policy entries        April 23, 2020 7:34:05 PM
PDT       April 23, 2020 7:34:05 PM PDT       Running

[root@oda1]# odacli describe-job -i ab819d57-9e35-4f55-bd5e-d26fc6372448

Job details
----------------------------------------------------------------
ID: ab819d57-9e35-4f55-bd5e-d26fc6372448
Description: delete job data retention policy
Status: Success
Created: April 23, 2020 7:34:05 PM PDT
Message:

Task Name Start Time End Time Status
```

```
-------------------------------------- ----------------------------------
---------------------------------- ----------
remove job data retention policy entries        April 23, 2020 7:34:05 PM
PDT       April 23, 2020 7:34:05 PM PDT       Success
remove auto job data schedule                   April 23, 2020 7:34:05 PM
PDT       April 23, 2020 7:34:05 PM PDT       Success

[root@oda1]# odacli list-jobdata-retention-policy

DCS-10032:Resource JobDataRetentionPolicy is not found.
```

odacli describe-job

Use the `odacli describe-job` command to display details about a specific job, including the job ID, status, tasks, and the job created date and time stamp.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To view a specific job, status, and tasks:

```
   odacli describe-job -i id [-e] [-n parent_task] [-sd sorted_by_duration] [-
s status] [-j] [-h]
```

Parameters

Parameter	Description
`--id`, `-i`	(Mandatory) Identifies the job. To get the job identifier (jobid), run the `list-jobs` command.
`--ecr`, `-e`	(Optional) Specifies to display the error correlation report.
`--number`, `-n`	(Optional) Specifies the number of parent tasks.
`--sort-duration`, `-sd`	(Optional) Sorts tasks by duration, either ascending (`asc`) or descending (`desc`).
`--status`, `-s`	(Optional) Specifies the status of the task, such as `running`, `failure`, `success`.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-190 Displaying Details for a Job

To display details of a specific job with job ID `02df22c8-c21f-4162-8265-97f7826c243a`:

```
 # odacli describe-job -i 02df22c8-c21f-4162-8265-97f7826c243a
```

Example 16-191 Viewing Error Correlation Report

To view error correlation report using ODACLI command:

```
# odacli describe-job -i cb6faa85-d2b4-441a-864a-17a8532cf78d --ecr
ODA Assistant - Error Correlation report
----------------------------------------
          Failed job ID: cb6faa85-d2b4-441a-864a-17a8532cf78d
            Description: Database service deletion with DB name: zdjGIEn with
id : 4ab1da3e-3918-4acf-bb7f-5529e750849f
             Start Time: 2023-04-14 07:24:37
               End Time: 2023-04-14 07:26:59
         EC report path:
/opt/oracle/dcs/da/da_repo/0ae47ffd-5641-47ed-b66f-f674a6c1fb1d.json

Failed Task Messages
--------------------
[Database service deletion with DB name: zdjGIEn with id : 4ab1da3e-3918-4acf-
bb7f-5529e750849f] - DCS-10001:Internal error encountered: Failed to delete:
RECO/ZDJGIENU for the database: zdjGIEn.

oda1 Log Messages
-----------------------------
  DCS Agent
  ~~~~~~~~~
    Error Logs
    ==========
    [Kill Pmon Process] - No process running for database : zdjGIEnU
    [Database Files Deletion] - Failed to delete directory RECO/ZDJGIENU due
to error: ORA-29261: bad argument
    [Database Files Deletion] - ORA-15178: directory 'ZDJGIENU' is not empty;
cannot drop this directory
    [Database Files Deletion] - ORA-15028: ASM file '+RECO.431.1134101713'
not dropped; currently being accessed
    [Database Files Deletion] - ORA-06512: at "SYS.X$DBMS_DISKGROUP", line
666
    [Database Files Deletion] - ORA-06512: at line 2 (DBD ERROR:
OCIStmtExecute)
    [Database Files Deletion] - Failed to delete: RECO/ZDJGIENU for the
database: zdjGIEn
    [Database Service deletion for 4ab1da3e-3918-4acf-bb7f-5529e750849f] -
DCS-10001:Internal error encountered: Failed to delete: RECO/ZDJGIENU for the
database: zdjGIEn.
      Error code - DCS-10001
      Cause: An internal error occurred.
      Action: Contact Oracle Support for assistance.
RHP Error Logs
    ==========
    [jobid-cb6faa85-d2b4-441a-864a-17a8532cf78d] -
[BatchMoveOpImpl.getMoveDBStateInfo:9195] attempt to fetch metadata from
repos failed with EntityNotExistsException : PRGR-110 : Repository object
"dbsa03b123e2;;;/u01/app/oracle/product/19.0.0.0/dbhome_1" of type "MOVE"
does not exist.
      Error code - PRGR-110
      Cause: The specified repository object does not exist in the repository
database.
```

```
      Action: Specify the correct repository object name.
    [jobid-cb6faa85-d2b4-441a-864a-17a8532cf78d] -
[GridHomeFactoryImpl.getTempLocation:1437] Exception node : PRCR-1001 :
Resource ora.rhpserver does not exist
      Error code - PRCR-1001
      Cause: An attempt was made to operate on a resource that does not
exist.
      Action: Check the resource using the command 'srvctl config <object>'
and if the resource does not exist, then create it using the command 'srvctl
add <object>'.

Release Notes
-------------
  No matching results were found.

Documentation
-------------
  1. Deleting a Database
      Abstract - Use the Oracle Appliance Manager Browser User Interface to
delete an Oracle database.
      Link -
https://domainname:7093/docs/cmtxn/managing-databases-oda.html#GUID-
D85D987A-1D9E-4615-B9DB-65B4B393B1D1
```

odacli list-jobs

Use the `odacli list-jobs` command to display a list of jobs, including the job IDs, status, and the job created date and time stamp. The default number of jobs listed is 50 from the earliest to the latest.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

To list jobs and view job details and status:

```
odacli list-jobs [-all] [-d] [-desc description_pattern] [-ec] [-o day]
    [-f from] [-hd head] [-l last] [-msg message_pattern]
    [-ndesc not_match_description_pattern]
    [-nmsg not_match_message_pattern] [-sd sorted_by_duration]
    [-s status] [-tl tail] [-t to] [-j] [-h]
```

Parameters

Parameter	Description
`--all-jobs, -all`	(Optional) Displays jobs created by both DCS and error correlation framework.
`--ec-jobs, -ec`	(Optional) Displays jobs created by the error correlation framework.

Parameter	Description
`--for, -o`	(Optional) Describes a job created on a specific date. In the format `YYYY-MM-DD`, such as `2018-01-01` . Example: `# odacli list-jobs -o 2018-08-09`

`--from, -f`	(Optional) Describes a job created since a specific date. In the format `YYYY-MM-DD` or `YYYY-MM-DD HH:mm:ss`, such as `2018-01-01` or `2018-01-01 01:00:00`.
	Examples:
	`# odacli list-jobs -f 2018-08-27`
	`# odacli list-jobs -f "2018-08-27 03:00:00"`

`--head, -hd`	(Optional) Describes the maximum number of older jobs to be displayed, values can be 1 to 200. Example: `odacli list-jobs -hd 10`
`--to, -t`	(Optional) Describes a job created ahead of a specific date. In the format `YYYY-MM-DD` or `YYYY-MM-DD HH:mm:ss`, such as `2018-01-01` or `2018-01-01 01:00:00`. Example: `# odacli list-jobs -t 2018-08-07`
`--last, -l`	(Optional) Describes jobs created from last 'n' time unit - where n can be any integer and time unit can be d for days, h for hours or m for minutes.
`--tail, -tl`	(Optional) Describes the maximum number of latest jobs to be displayed, values can be 1 to 200. Example: `# odacli list-jobs -tl 10`
`--description, -desc`	(Optional) Specifies the pattern to filter jobs based on their description. For example, `.*a`, `b.*`, `.*a|b.*`
`--no-desc, -ndesc`	(Optional) Specifies the pattern to filter jobs that do not match their description. For example, `.*a`, `b.*`, `.*a|b.*`

Parameter	Description
`--message, -msg`	(Optional) Specifies the pattern to filter jobs based on their message. For example, `.*a`, `b.*`, `.*a|b.*`
`--no-message, -nmsg`	(Optional) Specifies the pattern to filter jobs that do not match their message. For example, `.*a`, `b.*`, `.*a|b.*`
`--sort-duration, -sd`	(Optional) Specifies to sort jobs by duration, either `asc` or `desc`.
`--status, -s`	(Optional) Specifies the status of the job, such as created, scheduled, running, failure, success, internalerror. Example: `# odacli list-jobs -s success`
`--describe-job, -d`	(Optional) Specifies the description of the filtered job, if the filter returns a single job.
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-192 Displaying a List of Jobs

To display a list of jobs:

```
# odacli list-jobs

ID                           Description            Created
---------------------------- ---------------------
----------------------------
a6084067-72a1-4625-bea7-efd  Provisioning service   creation Jun 2, 2018
10:19:23 AM

(Continued)
Status
-------
Success
```

Example 16-193 Displaying Specified Number of Older Jobs

To display a specified number of older jobs:

```
# odacli list-jobs -hd 5
ID Description Created Status
--- --- --
643c6186-a667-43d7-847f-d756a96fa072 Create detailed Backup Report
August 5, 2018 12:36:26 AM UTC Success
f29cb82c-6150-487d-aaff-52d65f6ab972 Create recovery-pitr : time '08/05/2018
07:16:12'
for db : igIyoz August 5, 2018 1:01:39 AM UTC Success
cfe19337-ab2f-4ef3-85b5-24601ba31be2 Create detailed Backup Report
```

```
August 5, 2018 1:17:46 AM UTC Success
3749b5e2-240e-4df5-b3c1-32d345243b25 Create detailed Backup Report
August 5, 2018 2:01:07 AM UTC Success
64828852-bb16-4f06-aca9-4bdf9fe7b6f2 Create regular-5 Backup with TAG-
iD5UNjBZ8HN2FYL
for Db:igIyoz August 5, 2018 2:14:19 AM UTC Success
```

Example 16-194 Displaying Jobs in a Date Range

To display jobs created within a time frame:

```
# ./odacli list-jobs -f 2018-08-07 -t 2018-08-08
ID Description Created Status
--- --- --
1fc5ef97-a9db-49b2-9664-7c551bd005f4 Authentication key update for test_user
August 7, 2018 12:56:58 AM UTC Success
e24f86cf-bb2e-4ebe-84e0-da5cedb27ad4 Provisioning service creation August 7,
2018 1:00:54 AM UTC Success
7f67efd6-cce5-47b4-8dc4-fdb732491f99 CPU cores service update August 7, 2018
1:41:05 AM UTC Success
0ae9a312-bac8-43be-8a64-5a12c24c2a02 SSH keys update August 7, 2018 1:42:59
AM UTC Success
af41fdc1-87ff-46cd-bc41-26615e115ae7 SSH key delete August 7, 2018 1:44:56 AM
UTC Success
79e4cbb0-b474-48ab-9e04-d3d602bc0ed2 DcsCli patching August 7, 2018 1:50:18
AM UTC Success
edd55e7d-f81f-4a10-942d-1121aef0bff3 DcsCli patching August 7, 2018 1:50:32
AM UTC Success
acc22c60-3476-4566-8faa-4d36b116eded create backup
config:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa August 7, 2018 2:01:46 AM UTC Success
d89efedf-9110-429a-a3b2-ccd6a53f8564 Database Home OraDB12201_home2 creation
with version :12.2.0.1 August 7, 2018 2:36:05 AM UTC Success
2a5532ae-41fa-47c2-bc90-010a5f26d6b2 Database service creation with db name:
my122 August 7, 2018 3:52:38 AM UTC Success
c4e533e9-c596-478f-92db-2f11f1384075 Discover Components : all August 7, 2018
4:07:09 AM UTC Success
149e89b5-27d3-4c7c-9c03-f029ca3dd495 Discover Components : all August 7, 2018
4:08:45 AM UTC Success
21d4c37b-49ad-48dd-a1db-1f25d9288312 Database Home OraDB11204_home1 creation
with version :11.2.0.4 August 7, 2018 4:19:32 AM UTC Success
b61dbbca-edc1-4a4a-9db2-82b0bdcb64e6 Database service deletion with db name:
myTestDb with id : 6400c81d-5837-480c-b4a1-7c01591ee144 August 7, 2018
4:24:50 AM UTC Success
d4a104df-5796-4e37-9173-82f0e263d642 create backup config:bkfgd August 7,
2018 4:28:54 AM UTC Success
a121d40b-f33b-47a4-8fc6-4e3b84173f44 Database service creation with db name:
my112 August 7, 2018 6:51:13 PM UTC Success
42dd661f-9ba0-4877-ace9-39d3f212c071 Discover Components : all August 7, 2018
7:12:56 PM UTC Success
2f648f5a-c9c5-42ec-adb0-98cf6497c89e Discover System Components : system
August 8, 2018 3:28:12 AM UTC Success
a5755f43-509d-4d4c-b7ef-9f99660c4de7 DB Home Patching: Home Id is
97df26b3-42f2-4189-805b-82d1b38737d0 August 8, 2018 3:52:08 AM UTC Success
79b0e697-065f-4630-a524-8d072a4e139a Database Home OraDB12102_home1 creation
with version :12.1.0.2 August 8, 2018 4:34:30 AM UTC Success
112b75bc-3512-4c28-a479-3e0317eb0dc4 Database service creation with db name:
```

```
mynew121 August 8, 2018 8:02:51 AM UTC Failure
09f56fb0-1e91-4b02-a9b8-5add11a8da32 Database service creation with db name:
my121a August 8, 2018 9:13:01 AM UTC Success
5633ded9-07ea-4bf9-9533-31eb65789fe7 Database service deletion with db name:
mynew121 with id : 24be4362-16c8-476f-903a-f6f2ef59f5e4 August 8, 2018
9:24:31 AM UTC Success
458d1c45-02dc-456c-ae88-5da613faaa66 Database service creation with db name:
mynew121 August 8, 2018 9:27:31 AM UTC Success
f178f378-a9d7-4d5c-b6f5-6f62ea4e05bb Database service deletion with db name:
myTestDb with id : ef50387b-0c62-4446-b210-f8d070e2a043 August 8, 2018
9:40:43 AM UTC Success
7fab740f-a711-466a-ba6d-dd5643374c28 Database service deletion with db name:
myTestDb with id : 6632a083-558a-4eb4-8c2b-af0710179980 August 8, 2018
9:41:04 AM UTC Success
3121529d-2b9d-4bbe-bf58-996c2cf46996 Database service creation with db name:
mydss August 8, 2018 9:42:06 AM UTC Success
3d0a9261-19d7-42bb-8b92-00fcc4f8c41e Discover Components : db August 8, 2018
10:17:22 AM UTC Success
```

odacli list-pendingjobs

Use the `odacli list-pendingjobs` command to list all pending jobs.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-pendingjobs [-j] [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-195

```
# odacli list-pendingjobs
```

odacli list-jobdata-retention-policy

Use the `odacli list-jobdata-retention-policy` command to list job data retention policy for default DCS jobs and specific auto scheduled jobs.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To list jobs and view job details and status:

```
odacli list-jobdata-retention-policy [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-196 Listing all job retention policies

```
# odacli list-jobdata-retention-policy

Schedule Id Retention Window Unit Modified
------------------------------------- --------------------
------------------- ------------------------
Default     30  Days     April 23, 2020 6:16:59 PM PDT
284a22b5-41e6-4be1-a916-048eac74e076    10  Days     April 23, 2020 6:22:14
PM PDT
```

odacli list-purge-jobdata-jobs

Use the `odacli list-purge-jobdata-jobs` command to list purge job data for default DCS jobs and specific auto scheduled jobs.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To list job data purge policy:

```
odacli list-purge-jobdata-jobs [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

The command describes the files that are purged, the purged count, the purged file size, and the time of purge.

Example 16-197 Listing history, statistics, and rules for job data purge jobs

```
# odacli list-purge-jobdata-jobs

Purge By Job                             Purged Object             Count
Size       Type            Purged Time
---------------------------------------- -------------------- ----------
---------- --------------- --------------------
4d065c96-a8eb-4974-a446-573b5ad3cb95        TaskReport
102                 Metadata        April 23, 2020 1:55:59 PM PDT
4d065c96-a8eb-4974-a446-573b5ad3cb95        ServiceJobReport
24                  Metadata        April 23, 2020 1:55:59 PM PDT
4d065c96-a8eb-4974-a446-573b5ad3cb95        LogCleanupSummary
2                   Metadata        April 23, 2020 1:55:59 PM PDT
4d065c96-a8eb-4974-a446-573b5ad3cb95        JobExecutionResource
93                  Metadata        April 23, 2020 1:55:59 PM PDT
4d065c96-a8eb-4974-a446-573b5ad3cb95        JobDataPurgeResult
21                  Metadata        April 23, 2020 1:55:59 PM PDT
```

odacli list-scheduled-executions

Use the command `odacli list-scheduled-executions` to display a list of jobs that were executed.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli list-scheduled-executions [-e] [-1] [-j] [-h]
```

Parameters

Parameter	Description
--executionid, -e	(Optional) Displays the execution ID.
--scheduleid, -i	(Optional) Displays the Schedule ID.
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output. The default is false.

Example 16-198 Displaying a List of Scheduled and Executed Jobs

Display a list of scheduled tasks, the status, and the date and time that the task was executed. In the following example, three (3) scheduled tasks were executed. One task, Schedule ID 04b5750c-0d13-42cf-aba3-85416e8edafb, was executed twice.

```
# odacli list-scheduled-executions

ID                                       ScheduledId
JobId
---------------------------------------- ----------------------------------------
----------------------------------------
9890508c-ff6d-4307-972f-7962ec390871     04b5750c-0d13-42cf-
aba3-85416e8edafb
```

```
10a33e08-695c-4d13-b970-7dc7215f3cdd
7ee1e2aa-80a8-4297-97f7-365b84bcce4d
4a9e9796-4bed-4df7-af1e-de49362dda97   04b5750c-0d13-42cf-aba3-85416e8edafb

(Continued)
Status    Executed Time
--------- ------------------------------
Executed  February 4, 2018 12:00:00 AM UTC
Executed  February 4, 2018 12:00:00 AM UTC
Executed  February 5, 2018 12:00:00 AM UTC
```

To learn more about a scheduled backup, run the command `odacli list-schedules`. The output displays the ID, name, description, and schedule for each task.

```
 # odacli list-schedules

ID                                       Name
Description
---------------------------------------- ----------------------------
---------------------------------------------
d136cc72-4a80-4ab8-9f1e-120b995b2794    metastore maintenance
internal metastore maintenance
04b5750c-0d13-42cf-aba3-85416e8edafb    AgentState metastore cleanup
internal agentstateentry metastore maintenance
7ee1e2aa-80a8-4297-97f7-365b84bcce4d    backupreport maintenance      backup
reports deletion

(Continued)
CronExpression                 Disabled
------------------------------ --------
0 0 0 1/1 * ? *                true
0 0 0 1/1 * ? *                false
0 0 0 1/3 * ? *                false
```

odacli Oracle Key Vault Server Configuration Commands

Use `odacli` commands to manage Oracle Key Vault server configuration for databases on Oracle Database Appliance.

- odacli create-okvserverconfig
  Use the command `odacli create-okvserverconfig` to create an Oracle Key Vault server configuration.
- odacli copy-okvtdewallet
  Use the command `odacli copy-okvtdewallet` to copy an Oracle Key Vault TDE wallet.
- odacli delete-okvserverconfig
  Use the command `odacli delete-okvserverconfig` to delete an Oracle Key Vault server configuration.
- odacli describe-okvserverconfig
  Use the command `odacli describe-okvserverconfig` to view the Oracle Key Vault server configuration details.
- odacli list-okvserverconfigs
  Use the command `odacli list-okvserverconfigs` to view all Oracle Key Vault server configurations.

odacli create-okvserverconfig

Use the command `odacli create-okvserverconfig` to create an Oracle Key Vault server configuration.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli create-okvserverconfig -n name -u okv_user_name [-d okv_description]
[-host okv_host_name] [-ip okv_ip_address] [-h] [-j]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Specifies the name of the Oracle Key Vault server configuration object.
`--user-name, -u`	(Mandatory) Specifies the name of the Oracle Key Vault user on the Oracle Key Vault server.
`--description, -d`	(Optional) Describes the Oracle Key Vault server configuration object.
`--hostname, -host`	(Optional) Specifies the name of the Oracle Key Vault server where the Oracle Key Vault user has an account.
`--server-ip, -ip`	(Optional) Specifies the IP address of the Oracle Key Vault server where the Oracle Key Vault user has an account.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

To create an Oracle Key Vault server configuration `okvobj1` for the Oracle Key Vault server user `epokvusr1` on the specified server:

Example 16-199 Creating an Oracle Key Vault server configuration

```
odacli create-okvserverconfig -n okvobj1 -u epokvusr1 -ip xxx.xx.xxx.xxx
Enter OKV user password :
Retype OKV user password :
```

odacli copy-okvtdewallet

Use the command `odacli copy-okvtdewallet` to copy an Oracle Key Vault TDE wallet.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli copy-okvtdewallet -swn source_wallet_name -tosc
target_okv_server_config -twn target_okv_wallet_name [-shn hostname] [-sip
ip_address] [-su user_name] [-j] [-h]
```

Parameters

Parameter	Description
--source-wallet-name, -swn	(Mandatory) Specifies the name of the wallet from which TDE keys must be copied to the target wallet in the Oracle Key Vault server.
--target-okvserverconfig, -tosc	(Mandatory) Specifies the name of the Oracle Key Vault server configuration object where the target wallet must be created.
--target-wallet-name, -twn	(Mandatory) Specifies the name of the wallet to which the TDE keys must be copied from the source wallet in the Oracle Key Vault server.
--source-hostname, -shn	(Optional) Specifies the fully-qualified domain name (FQDN) of the Oracle Key Vault server where the source wallet is created. Specify this value only if the source wallet and the target wallet use different Oracle Key Vault servers.
--source-ip, -sip	(Optional) Specifies the IP address of the Oracle Key Vault server where the source wallet is created. Specify this value only if the source wallet and the target wallet use different Oracle Key Vault servers.
--source-username, -su	(Optional) Specifies the name of the Oracle Key Vault user who owns the source wallet on the Oracle Key Vault server. Specify this value only if the source wallet and the target wallet use different Oracle Key Vault servers.
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output. The default is false.

Example 16-200 Copying the TDE wallet in the same Oracle Key Vault server and owned by the same Oracle Key Vault user

The source TDE wallet and the target TDE wallet are on the same Oracle Key Vault server, and both are owned by the same user on the Oracle Key Vault server.

```
odacli copy-okvtdewallet -swn tdeokv_on_scaoda8m013-c -tosc okvobj1 -twn
tdeokvss
Job details
----------------------------------------------------------------
                     ID:  50df7dcc-8e17-48db-9206-879c44e57c20
            Description:  Copy wallet in OKV server
                 Status:  Success
                Created:  June 20, 2024 2:56:02 AM CST
                Message:
```

```
Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Creating a TDE wallet copy in the OKV    June 20, 2024 2:56:03 AM
CST             June 20, 2024 2:57:17 AM CST             Success
server
```

Example 16-201 Copying the TDE wallet on the same Oracle Key Vault server but using a different Oracle Key Vault user

The source and the target wallet are on the same Oracle Key Vault server but are owned by different users on the Oracle Key Vault server. In this scenario, the target wallet user must copy the source wallet with the following command. Before running the ODACLI command to copy the wallet, the source wallet Oracle Key Vault user must provide read and manage access permissions on the wallet, to the target wallet Oracle Key Vault user. For more details, see the *Restoring TDE-Enabled Database with TDE Keys Stored on Oracle Key Vault*topic.

```
odacli copy-okvtdewallet -swn tdeokv_on_scaoda8m013-c -tosc okvobj2 -twn
tdeokvdu
Job details
----------------------------------------------------------------
                     ID:  50df7dcc-8e17-48db-9206-879c44e57c20
            Description:  Copy wallet in OKV server
                 Status:  Success
                Created:  June 20, 2024 2:56:02 AM CST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Creating a TDE wallet copy in the OKV    June 20, 2024 2:56:03 AM
CST             June 20, 2024 2:57:17 AM CST             Success
server
```

Example 16-202 Copying the TDE wallet to a different Oracle Key Vault server

The source wallet and the target wallet are on different Oracle Key Vault servers. Specify the source wallet Oracle Key Vault server details also must be provided in the ODACLI command. Specify the source wallet Oracle Key Vault user password, when prompted.

```
odacli copy-okvtdewallet -swn tdeokv_on_scaoda8m013-c -sip xxx.xx.x.xxx -su
epokvusr1 -tosc okvobj3 -twn tdeokvds
Enter OKV user password:
Retype OKV user password:

Job details
----------------------------------------------------------------
                     ID:  50df7dcc-8e17-48db-9206-879c44e57c20
            Description:  Copy wallet in OKV server
```

```
                 Status:  Success
                Created:  June 20, 2024 2:56:02 AM CST
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Creating a TDE wallet copy in the OKV    June 20, 2024 2:56:03 AM
CST             June 20, 2024 2:57:17 AM CST             Success
server
```

odacli delete-okvserverconfig

Use the command `odacli delete-okvserverconfig` to delete an Oracle Key Vault server configuration.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli delete-okvserverconfig [-n name] [-h] [-j]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Specifies the Oracle Key Vault server configuration to delete.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Use the `odacli delete-okvserverconfig` command to delete Oracle Key Vault server configuration. You can delete the Oracle Key Vault server configuration only if the databases that used the Oracle Key Vault server configuration are already deleted.

Example 16-203 Deleting Oracle Key Vault server configuration

```
odacli delete-okvserverconfig -n okvobj1
```

odacli describe-okvserverconfig

Use the command `odacli describe-okvserverconfig` to view the Oracle Key Vault server configuration details.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli describe-okvserverconfig [-n name] [-h] [-j]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Specifies the Oracle Key Vault server configuration.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

For example, the `odacli describe-okvserverconfig` command in a multi-user access enabled system displays the following:

Example 16-204 Viewing Oracle Key Vault server configuration details in a multi-user access enabled system

```
odacli describe-okvserverconfig -n okvobj1

OKV Server Config details
------------------------------------------------------------------
                     ID: 441fb120-35f8-4c9a-9a5c-76aebaba1c36
                   Name: okvobj1
          OKV User Name: epadmin
          OKV Server IP: xxx.xx.xx.xxx
          Primary owner: okvusr1
       Secondary owners: [odaadmin,dbusr1, dbusr2]



odacli describe-okvserverconfig -n okvobj1 -j

{
    "ID" : "441fb120-35f8-4c9a-9a5c-76aebaba1c36",
    "Name" : "okvobj1"
    "Primary owner" : "okvusr1"
    "Secondary Owners : [ "odaadmin", "dbusr1", "dbusr2" ]
    "createTime" : "January 04, 2024 11:25:04 AM UTC",
    "updatedTime" : "January 04, 2024 11:37:48 AM UTC"
}
```

Example 16-205 Viewing Oracle Key Vault server configuration details in a non-multi user access enabled system

In a non-multi user access enabled system, the primary and secondary owner details are not displayed:

```
odacli describe-okvserverconfig -n okvobj1

OKV Server Config details
----------------------------------------------------------------
                     ID: 441fb120-35f8-4c9a-9a5c-76aebaba1c36
                   Name: okvobj1
          OKV User Name: epadmin
          OKV Server IP: xxx.xxx.xx.xxx

odacli describe-okvserverconfig -n okvobj1 -j

{
    "ID" : "441fb120-35f8-4c9a-9a5c-76aebaba1c36",
    "Name" : "okvobj1",
    "OKV User Name" : "epadmin",
    "OKV Server IP": "xxx.xxx.xx.xxx",
    "createTime" : "January 04, 2024 11:25:04 AM UTC",
    "updatedTime" : "January 04, 2024 11:37:48 AM UTC"
}
```

Example 16-206 Viewing Oracle Key Vault server configuration details with the `-j` option

```
odacli describe-okvserverconfig -n dokvobj1 -j
{
  "id" : "2348516e-37f4-41a6-b482-c70ce0e53bc8",
  "name" : "dokvobj1",
  "userName" : "user1",
  "hostname" : null,
  "ipAddress" : "xxx.xxx.xx.xxx",
  "description" : null,
  "createTime" : "October 20, 2024 05:49:22 AM PDT",
  "updatedTime" : "October 20, 2024 05:49:47 AM PDT",
  "state" : {
    "status" : "CONFIGURED"
  }
}
```

odacli list-okvserverconfigs

Use the command `odacli list-okvserverconfigs` to view all Oracle Key Vault server configurations.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli list-okvserverconfigs [-h] [-j]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

Usage Notes

Example 16-207 Viewing all Oracle Key Vault server configurations

```
odacli list-okvserverconfigs

ID                                       Name                 User
Name         IP address          Host Name           Description
---------------------------------------- --------------------
---------------- ------------------- -------------------- -------------------
d017d35a-0757-460a-98f0-9f8d0944c872     okvobj1
epadmin           xxx.xx.xx.xxx       Null                 Brief
description
b6336173-666b-4d81-88bd-5ae384419114     okvobj2
epadmin2          xxx.xx.xx.xxx       Null                 Brief description
```

Log Commands

Use the `odacli` log commands to specify the options to collect and delete logs.

- odaadmcli manage diagcollect
  Use the `odaadmcli manage diagcollect` command to collect diagnostic logs for Oracle Database Appliance components.
- odacli create-auto-logclean-policy
  Use the `odacli create-auto-logclean-policy` command to create a policy for a job to automatically purge log files for a specified time period when certain criteria are met.
- odacli delete-auto-logclean-policy
  Use the `odacli delete-auto-logclean-policy` command to delete a policy for a job to automatically purge log files for a specified time period when certain criteria are met.
- odacli create-logcleanjob
  Use the `odacli create-logcleanjob` command to create a job to purge log files for a specified time period.
- odacli describe-logcleanjob
  Use the `odacli describe-logcleanjobcommand`to describe a log cleanup job.
- odacli list-auto-logclean-policy
  Use the `odacli list-auto-logclean-policy` command to list the jobs to purge log files.

- odacli list-logcleanjobs
  Use the `odacli list-logcleanjobs` command to list the jobs to purge log files.
- odacli list-logspaceusage
  Use the `odacli list-logspaceusage` command to display the log file usage.

odaadmcli manage diagcollect

Use the `odaadmcli manage diagcollect` command to collect diagnostic logs for Oracle Database Appliance components.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To collect diagnostic logs for components:

```
   odaadmcli manage diagcollect [-h] [--dataMask] [--dataSanitize] [--
fromTime] [--toTime] [--Components]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--dataMask`	(Optional) Masks sensitive data. The default is No.
`--dataSanitize`	(Optional) Sanitizes (redacts) sensitive data. The default is No.
`--fromTime`	(Optional) Specifies the time from when you want to collect data. The timestamp formats can be of the type: `mon/dd/yyyy hh:mm:ss`, or `yyyy-mm-dd hh:mm:ss`, or `yyyy-mm-ddThh:mm:ss`, or `yyyy-mm-dd`.
`--toTime`	(Optional) Specifies the time till when you want to collect data. The timestamp formats can be of the type: `mon/dd/yyyy hh:mm:ss`, or `yyyy-mm-dd hh:mm:ss`, or `yyyy-mm-ddThh:mm:ss`, or `yyyy-mm-dd`.
`--Components`	(Optional) Specifies the comma-separated list of components for which you want to collect data. The supported values are: `all, ips, oda, odalite, dcs, odabackup, odapatching, odadataguard, odaprovisioning, odaconfg, odasystem, odastorage, database, asm, crsclient, dbclient, dbwlm, tns, rhp, procinfo, afd, crs, cha, wls, emagent, oms, ocm, emplugins, em, acfs, install, cfgtools, os, ashhtml, ashtext, awrhtml, awrtext`
`--verbose`	Displays detailed message.

Usage Notes

The following types of sensitive information can be redacted using the `--dataMask` or the `--dataSanitize` option:

- Host names
- IP addresses
- Database names
- Tablespace names
- Service names
- Ports
- Operating System user names

For example, when the `--dataMask` option is used, all instances of a sensitive name such as a database name called "payrolldb" are replaced with "*********" in the TFA collection.

For example, when the `--dataSanitize` option is used, all instances of a sensitive name such as a database name called "payrolldb" are replaced with another string, such as "oCjlN7F8P", in the TFA collection.

Example 16-208 Masking Sensitive Data in Log Collection

```
# /opt/oracle/oak/bin/odaadmcli manage diagcollect --dataMask
DataMask is set as true
TFACTL command is: /opt/oracle/tfa/tfa_home/bin/tfactl
Data mask is set.
Collect data within last 12 hours as default
Please wait for several minutes for the collection to complete.
```

odacli create-auto-logclean-policy

Use the `odacli create-auto-logclean-policy` command to create a policy for a job to automatically purge log files for a specified time period when certain criteria are met.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli create-auto-logclean-policy [-h] [-c] [-o] [-u] [-uMB] [-f] [-uPer]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--components`, `-c`	(Optional) Specifies the list of components, separated by comma. The values are `gi|database|dcs` For example, `gi|dcs`.

Parameter	Description
`--olderthan, -o`	(Optional) Cleans logs older than specified time interval. Default is 30 if it is not specified.
`--olderThanUnit, -u`	(Optional) Unit for the `--olderthan` parameter. Default is `Day` if it is not specified.
`--freeSpaceBelowPercentage, -f`	Starts purge when free disk space is below the specified number of percentage of the total partition size. Valid range is 20-50. Default is 20.
`--usageOverMB, -uMB`	Starts purge when log usage is over the specified number of MB. Valid range is 10%-50% of partition size.
`--usageOverPercentage, -uPer`	Starts purge when log usage is over the specified number of percentage of total partition space. Valid range is 10-50.

Example 16-209 Creating Automatic Jobs to Purge Logs

To create a policy for automatic jobs to purge logs:

```
# odacli create-auto-logclean-policy -c 'dcs' -o 30 -uPer 30
Component  UsageOverPercentage  freeSpaceBelowPercentage UsageOverMB
OlderThan   OlderThanUnit

------------------- ------------- -------------------- --------------------
-------------
dcs                  30                   20                   0
30           Day
```

odacli delete-auto-logclean-policy

Use the `odacli delete-auto-logclean-policy` command to delete a policy for a job to automatically purge log files for a specified time period when certain criteria are met.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli odacli delete-auto-logclean-policy [-c] [-h] [-j]
```

Parameters

Parameter	Description
`--components, -c`	(Optional) Specifies the list of components, separated by comma. The values are `gi|database|dcs` For example, `gi|dcs`.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.

odacli create-logcleanjob

Use the `odacli create-logcleanjob` command to create a job to purge log files for a specified time period.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To list jobs and view job details and status:

```
odacli create-logcleanjob [-h] [-c] [-o] [-u]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--components`, `-c`	(Optional) Specifies the list of components, separated by comma. The values are `gi|database|dcs` For example, `gi,dcs`.
`--olderthan`, `-o`	(Optional) Cleans logs older than specified time interval. Default is 30 if it is not specified.
`--unit`, `-u`	(Optional) Unit for the `--olderthan` parameter. Default is `Day` if it is not specified.

Example 16-210 Creating Jobs to Purge Logs

To create jobs to purge logs:

```
# odacli create-logcleanjob
Job details
----------------------------------------------------------------

                     ID:  e03d90b5-41dd-45e0-8b7a-1480d6d7f86f

            Description:  log file cleanup

                 Status:  Created

                Created:  July 25, 2018 8:06:56 PM UTC

                Message:



Task Name   Start Time End Time  Status

---------------------------------------- ----------------------------------
-------------
# ./odacli describe-job -i e03d90b5-41dd-45e0-8b7a-1480d6d7f86f
```

```
Job details

----------------------------------------------------------------

                     ID:  e03d90b5-41dd-45e0-8b7a-1480d6d7f86f

            Description:  log file cleanup

                 Status:  Success

                Created:  July 25, 2018 8:06:56 PM UTC

                Message:


Task Name    Start Time    End Time      Status
------------------------------------ --------------------------------- -
Clean TFA logs     July 25, 2018 8:06:56 PM UTC July 25, 2018 8:06:59 PM UTC
Success

Clean DCS logs   July 25, 2018 8:06:56 PM UTC  July 25, 2018 8:06:56 PM
UTC        Success
```

odacli describe-logcleanjob

Use the `odacli describe-logcleanjobcommand`to describe a log cleanup job.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To list jobs and view job details and status:

```
odacli describe-logcleanjob -i [-h] [-j]
```

Parameters

Parameter	Description
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output.
--id, -i	Displays the summary of the log cleanup with the specified job ID.

Example 16-211 Displaying Log Cleanup Jobs

To display log cleanup jobs:

```
# odacli describe-logcleanjob  -i e03d90b5-41dd-45e0-8b7a-1480d6d7f86f
JobId    Files Deleted  Size Freed   Component  Top Path
```

```
TimeStamp

-------------------- -------------------- --------------------
-------------------- -----
e03d90b5-41dd-45e0-8b7a-1480d6d7f86f 0 0 bytes dcs /opt/oracle/dcs/log  July
25, 2018 8:06:56 PM UTC

e03d90b5-41dd-45e0-8b7a-1480d6d7f86f 0 0 bytes  gi /u01/app/12.2.0.1/grid
July 25, 2018 8:06:59 PM UTC
```

odacli list-auto-logclean-policy

Use the `odacli list-auto-logclean-policy` command to list the jobs to purge log files.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To list jobs and view job details and status:

```
odacli list-auto-logclean-policy [-h] [-c]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--components`, `-c`	(Optional) Specifies the list of components, separated by comma. The values are `gi|database|dcs` For example, `gi,dcs`.

Example 16-212 Listing Jobs to Purge Logs

To list jobs to purge logs:

```
# odacli list-auto-logclean-policy
Component  UsageOverPercentage  freeSpaceBelowPercentage UsageOverMB
OlderThan OlderThanUnit
--- -------------------- --------------------
-----------------------------------------------
gi                   40                   20
0                    60        Day

database             40                   20
0                    60      Day

dcs                  30                   20
0                    30     Day
```

odacli list-logcleanjobs

Use the `odacli list-logcleanjobs` command to list the jobs to purge log files.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To list jobs and view job details and status:

```
odacli list-logcleanjobs [-h]
```

Parameters

Parameter	Description
--help, -h	(Optional) Displays help for using the command.

Example 16-213 Listing Jobs to Purge Logs

To list jobs to purge logs:

```
# odacli list-logcleanjobs
JobId   Files Deleted    Size Freed   Component    Top Path  TimeStamp

-------------------- -------------------- --------------------
--------------------a8ef8315-02e8-4979-ab1f-30251818c6eb
0  0 bytes gi /u01/app/12.2.0.1/grid July 25, 2018 7:45:10 PM UTC

e03d90b5-41dd-45e0-8b7a-1480d6d7f86f 0 0 bytes  gi   /u01/app/12.2.0.1/grid
July 25, 2018 8:06:59 PM UTC

e03d90b5-41dd-45e0-8b7a-1480d6d7f86f 0  0 bytes dcs  /opt/oracle/dcs/log
July 25, 2018 8:06:56 PM UTC

---------- ------------------------------------------------------------
```

odacli list-logspaceusage

Use the `odacli list-logspaceusage` command to display the log file usage.

File Path

/opt/oracle/dcs/bin/odacli

Syntax

To list jobs and view job details and status:

```
odacli list-logspaceusage [-c] [-h]
```

Parameters

Parameter	Description
`--components, -c`	Describes the list of components delimited by comma. The values can be `gi|database|dcs`. For example,`gi,dcs`.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-214 Displaying a List of Logs

To display log file space usage for a list of components:

```
root@node1 ~]# odacli list-logspaceusage

              Timestamp: January 13, 2022 8:38:32 AM UTC
-------------------------------------------------------------


Node : node1
--------------------------------
Component Size(MB) Partition                           LogUsagePercentage
PartitionFreePercentage
-------------------------------------------------------------------------------
-----------------

dcs       685.00   /dev/mapper/VolGroupSys-LogVolOpt  0.76               84.00
gi        2293.76  /dev/mapper/VolGroupSys-LogVolU01  1.63               79.00
database  744.14   /dev/asm/odabase_n0-159            7.27               83.00

Node : node2
--------------------------------
Component Size(MB) Partition                         LogUsagePercentage
PartitionFreePercentage
-------------------------------------------------------------------------------
----------------------

dcs       174.00   /dev/mapper/VolGroupSys-LogVolOpt 0.19               85.00
gi        1863.68  /dev/mapper/VolGroupSys-LogVolU01 1.32               79.00
database  299.36   /dev/asm/odabase_n1-159           2.92               90.00
```

odacli Oracle Auto Service Request Commands

Use the Oracle Auto Service Request (Oracle ASR) commands to configure, update, test, and delete Oracle ASR on the system.

- odacli configure-asr
  Use the `odacli configure-asr` command to configure Oracle Auto Service Request (Oracle ASR) after deploying the appliance.
- odacli describe-asr
- odacli delete-asr
- odacli export-asrconfig
- odacli modify-asr

- odacli test-asr

odacli configure-asr

Use the `odacli configure-asr` command to configure Oracle Auto Service Request (Oracle ASR) after deploying the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli configure-asr -u asr_usr [-i asr_mgr_ip] [-e asr_type] [-p] [-t
proxy_port] [-r proxy_server] [-y proxy_usr] [-hp HTTPS_port] [-j] [-h]
```

Parameters

Parameter	Description
`--username`, `-u`	Specifies the Oracle ASR user name. The user name is the My Oracle Support user name under which the server is registered.
`--asrtype`, `-e` {internal | external}	Specifies the Oracle ASR Configuration Type. The default is internal.
`--https-port`, `-hp`	(Optional) Specifies the HTTPS port for Oracle ASR Manager. The default port is 16162.
`--proxyport`, `-t`	(Optional) Specifies the port on which Oracle ASR must be configured.
`--proxyserver`, `-r`	(Optional) Specifies the proxy server address that can enable connect to My Oracle Support.
`--proxyuser`, `-y`	(Optional) Specifies the proxy user name needed to authenticate the proxy server.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

- All log files for Oracle ASR are located in the `/var/opt/asrmanager/log/` directory.
- To configure an external Oracle ASR Manager, you must define the Oracle ASR Configuration Type as external (`-e external`). For example, `odacli configure-asr -e external`.

Example 16-215 Configuring Oracle ASR with a Proxy Server

This example configures Oracle ASR for user name user@example.com. An HTTPS port 16165 is defined.

```
# odacli configure-asr -u user@example.com -hp 16165
Enter ASR user's password:
Retype ASR user's password:

Job details
```

```
------------------------------------------------------------
                     ID:  2e6244e5-41e3-46cf-bd7a-43098dd071d8
            Description:  Configure ASR
                 Status:  Success
                Created:  September 5, 2023 2:08:56 PM CST
                Message:

Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------------
Modify JavaExec Path                     September 5, 2023 2:08:57 PM CST
September 5, 2023 2:08:57 PM CST    Success
Install ASR                              September 5, 2023 2:08:57 PM CST
September 5, 2023 2:10:13 PM CST    Success
Registering ASR Manager                  September 5, 2023 2:10:14 PM CST
September 5, 2023 2:10:22 PM CST    Success
Creating ASR configuration folders       September 5, 2023 2:10:22 PM CST
September 5, 2023 2:10:22 PM CST    Success
Enable HTTPS for ASR                     September 5, 2023 2:10:22 PM CST
September 5, 2023 2:10:46 PM CST    Success
Importing ASR certificate into DCS       September 5, 2023 2:10:46 PM CST
September 5, 2023 2:10:48 PM CST    Success
truststore

Registering Asset: ODA Host              September 5, 2023 2:10:48 PM CST
September 5, 2023 2:10:48 PM CST    Success
ASR assets activation                    September 5, 2023 2:10:48 PM CST
September 5, 2023 2:10:49 PM CST    Success
Remove AsrConfBackup File                September 5, 2023 2:10:55 PM CST
September 5, 2023 2:10:55 PM CST    Success
```

Example 16-216 Configuring an External Oracle ASR

This example configures Oracle Database Appliance to use an external Oracle ASR instance.

```
# odacli configure-asr -e External

Enter ASR user's password:
Retype ASR user's password:

Job details
----------------------------------------------------------------
                     ID: cdd8445f-aefb-4f43-881a-145eaa1a1181
            Description: Configure ASR
                 Status: Success
                Created: September 5, 2023 1:41:33 PM CST
                Message:

Task Name                                 Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
Creating ASR configuration folders        September 5, 2023 1:41:37 PM CST
September 5, 2023 1:41:38 PM CST    Success
Importing ASR certificate into DCS        September 5, 2023 1:41:38 PM CST
```

```
September 5, 2023 1:41:39 PM CST    Success
truststore
Registering Asset: ODA Host              September 5, 2023 1:41:39 PM CST
September 5, 2023 1:41:39 PM CST    Success
ASR assets activation                    September 5, 2023 1:41:39 PM CST
September 5, 2023 1:41:40 PM CST    Success
```

odacli describe-asr

Use the `odacli describe-asr` command to display Oracle Auto Service Request (Oracle ASR) configuration details.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
describe-asr [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-217 Displaying Oracle ASR Details for ASR Type Internal

```
# odacli describe-asr

ASR details
----------------------------------------------------------------
                     ID: e0e31043-c312-4371-8301-e62642ef8a4a
                   Name: ASR
               ASR Type: Internal
        ASR Manager IP : 10.32.202.13
   ASR Manager username: username@example.com
      HTTPS port number: 16164
                  State: CONFIGURED
                Created: September 3, 2023 4:02:38 AM UTC
                Updated: September 3, 2023 4:03:32 AM UTC
```

odacli delete-asr

Use the `odacli delete-asr` command to remove the Oracle Auto Service Request (Oracle ASR) configuration from the system.

File Path

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli delete-asr [-h]
```

Parameters

Parameter	Description
--help, -h	(Optional) Displays help for using the command.

Example 16-218 Deleting Oracle ASR From the System

```
# odacli delete-asr
{
"jobId" : "5d70bd17-ec4a-48da-8196-1364105db99d",
"status" : "Running",
"message" : null,
"reports" : [ ],
"createTimestamp" : 1469409622451,
"description" : "Delete ASR",
"updatedTime" : 1469409622458
}
```

odacli export-asrconfig

Use the `odacli export-asrconfig` command to export the internal Oracle ASR Manager configuration files into a zip. The generated zip file can be used to configure the external Oracle ASR Manager on another Oracle Database Appliance.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli export-asrconfig [-j] [-h]
```

Parameters

Parameter	Description
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output.

Example 16-219 Exporting the Internal Oracle ASR Configuration

```
# odacli export-asrconfig
Job details
----------------------------------------------------------------
                     ID: 7adb6cdc-31b8-42a4-83ac-a866307329fc
            Description: Export ASR Config
                 Status: Created
```

```
                Created: September 4, 2023 6:11:14 PM UTC
                Message: Run Update repository with the zip /opt/oracle/asr/
asrConfiguration/ASRConfiguration.zip in the External ASR host once the
current job is successful

Task Name Node Name Start Time End Time Status
---------------------------------------- ------------------------
---------------------------------- --------------------------------
```

odacli modify-asr

Use the `odacli modify-asr` command to make changes to Oracle Auto Service Request (Oracle ASR) configuration details after deploying the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli modify-asr -u username [-r proxyserver] [-t proxyport] [-y proxyuser] -
e [internal|external][-hp HTTPS_port] [-j] [-h]
```

Parameters

Parameter	Description
`--username`, `-u`	Defines the Oracle ASR user name. The user name is the My Oracle Support user name under which the server is registered.
`--asrtype`, `-e` [internal | external]	Defines the Oracle ASR configuration type. The default is internal.
`--https-port`, `-hp`	(Optional) Specifies the HTTPS port for Oracle ASR Manager. The default port is 16162.
`--proxyport`, `-t`	(Optional) Defines the proxy server port.
`--proxyserver`, `-r`	(Optional) Defines the proxy server address.
`--proxyuser`, `-y`	(Optional) Defines the proxy user name needed to authenticate the proxy server.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

- Enter the ASR user password during Oracle ASR configuration.
- All log files for Oracle ASR are located in the `/var/opt/asrmanager/log/` directory.
- You cannot use the `modify-asr` command to change the Oracle ASR type. For example, from internal to external. To change the Oracle ASR type, delete the existing configuration using the `odacli delete-asr` and then re-configure Oracle ASR using the `odacli configure-asr` command.
- You cannot modify an external Oracle ASR Manager using the `odacli modify-asr` command. To modify external ASR, you must delete the existing Oracle ASR configuration

and reconfigure Oracle ASR using the `odacli configure-asr` command. Before running the `odacli configure-asr` command, you must update the repository with the modified Oracle ASR Manager configuration zip file.

Example 16-220 Modifying Oracle ASR to use a New Proxy Server

This example updates Oracle ASR for user name scott.tiger@example.com. The password is not defined in the command-line. You are prompted to enter the password during configuration. The proxy server is updated to www-proxy2.example.com.

```
# odacli modify-asr -u scott.tiger@example.com --proxyserver www-
proxy2.example.com --proxyport 80
Enter ASR user's password:
Retype ASR user's password:

Job details
----------------------------------------------------------
ID: 79cb2baa-1644-45c5-a004-a303e3111807
Description: Update ASR
Status: Created
Updated: September 8, 2024 9:53:54 AM PST
Message:

Task Name          Start Time          End Time        Status
---------------- ---------------- ----------------- ----------
```

odacli test-asr

Use the `odacli test-asr` command to test the Oracle Auto Service Request (Oracle ASR) configuration. When the command is run, an email alert is sent to each host and Oracle ILOM registered with Oracle ASR Manager.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
# odacli test-asr [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-221 Testing the Oracle ASR Configuration

```
# odacli test-asr

Job details
----------------------------------------------------------------
                     ID: 51522c07-1b32-4874-9bc7-bbee5885f1e5
            Description: Test ASR
                 Status: Success
```

```
                Created: October 20, 2022 1:02:49 PM UTC
                Message:

Task Name Start Time                          End Time
Status
--------- ------------------------------ ------------------------------
--------
Test ASR  October 20, 2022 1:02:50 PM UTC October 20, 2022 1:02:50 PM UTC
Success
```

Adaptive Classification and Redaction (ACR) Commands

Understand Adaptive Classification and Redaction (ACR) commands.

- odacli describe-acr
  Use the `odacli describe-acr` command to check the status of ACR.
- odacli enable-acr
  Use the `odacli enable-acr` command to enable ACR across all nodes, causing redaction of files during Oracle Trace File Analyzer diagnostic collections to occur.
- odacli redact-acr
  Use the `odacli redact-acr` command to enable ACR across all nodes, causing redaction of files during Oracle Trace File Analyzer diagnostic collections to occur.
- odacli list-acr
  Use the `odacli list-acr` command to print reverse map of sanitized entities.
- odacli disable-acr
  Use the `odacli disable-acr` command to disable ACR across all nodes. Redaction of files during Oracle Trace File Analyzer diagnostic collections, then, does not occur.

odacli describe-acr

Use the `odacli describe-acr` command to check the status of ACR.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-acr [-j] [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-222

```
# odacli describe-acr
```

odacli enable-acr

Use the `odacli enable-acr` command to enable ACR across all nodes, causing redaction of files during Oracle Trace File Analyzer diagnostic collections to occur.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli enable-acr [-j] [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-223

```
# odacli enable-acr
```

odacli redact-acr

Use the `odacli redact-acr` command to enable ACR across all nodes, causing redaction of files during Oracle Trace File Analyzer diagnostic collections to occur.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli redact-acr -i path_name -m mode [-j] [-h]
```

Parameters

Parameter	Description
`--inpath`, `-i`	(Required) Specifies the path of input file to be redacted.
`--mode`, `-m`	(Required) Specifies the redaction mode. The values can be `mask` or `sanitize`.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-224

```
# odacli redact-acr -i path_name -m sanitize
```

odacli list-acr

Use the `odacli list-acr` command to print reverse map of sanitized entities.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-acr -l list [-j] [-h]
```

Parameters

Parameter	Description
`--list`, `-l`	(Required) Specifies the list of sanitized entities.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-225

```
# odacli list-acr -l list
```

odacli disable-acr

Use the `odacli disable-acr` command to disable ACR across all nodes. Redaction of files during Oracle Trace File Analyzer diagnostic collections, then, does not occur.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli disable-acr [-j] [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Example 16-226

```
# odacli disable-acr
```

odacli OS Commands

Use the `odacli OS` commands to list and update operating system (OS) parameters.

- odacli list-osconfigurations
  Use the command `odacli list-osconfigurations` to display the current HugePage and memlock values and view suggested values based on the total available space.
- odacli modify-osconfigurations
  Use the command `odacli modify-osconfigurations` to update the HugePage and memlock values.

odacli list-osconfigurations

Use the command `odacli list-osconfigurations` to display the current HugePage and memlock values and view suggested values based on the total available space.

File Path

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli list-osconfigurations [-h] [-j]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.

Usage Notes

The command displays the configured values for HugePage and memlock from the `/etc/sysctl.conf` and `/etc/security/limits.conf`files. Based on the total available space, suggested values are calculated for the parameters.

Example 16-227 Displaying a List of Configured and Suggested Memlock and HugePage Configurations

```
# odacli list-osconfigurations

Parameter       User            ConfiguredValue                 SuggestedValue
--------------- --------------- ------------------------------- ---------------
Memlock         grid            295971180KB                     289034355KB

Memlock         oracle          295971180KB                     289034355KB

HugeSpace       default         101430MB                        101161MB
```

odacli modify-osconfigurations

Use the command `odacli modify-osconfigurations` to update the HugePage and memlock values.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

`odacli modify-osconfigurations [-hs` *`hugepage`*`] [-m` *`memlock`*`] [-nhp` *`number_of_hugepages`*`] [-j] [-h]`

Parameters

Parameter	Description
`--memlock`, `-m`	(Optional) Updates only the memlock value.
`--hugepage`, `-hs`	(Optional) Updates only the HugePage value.
`--number-hugepages`, `-nhp`	(Optional) Updates HugePages value.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

The command updates memlock in the `/etc/security/limits.conf`file and HugePage in the`/etc/sysctl.conf` file with the suggested values.

You can modify memlock or hugepage. If no option is provided, then both parameters are updated to the suggested values.

Example 16-228 Updating the HugePage and Memlock Parameters to the Suggested Values

```
odacli modify-osconfigurations
{
"jobId" : "954cf7a5-9cad-451c-8820-3140a716af26",
"status" : "Created",
"message" : "Successfully submitted a request to configure OS
parameters.",
"reports" : [ ],
"createTimestamp" : "September 8, 2024 00:03:51 AM MST",
"resourceList" : [ ],
"description" : "Configuring OS Parameter",
"updatedTime" : "September 8, 2024 00:03:51 AM MST"
[root@rwsoda6s002 ~]# odacli describe-job -i
"954cf7a5-9cad-451c-8820-3140a716af26"
Job details
----------------------------------------------------------------
ID: 954cf7a5-9cad-451c-8820-3140a716af26
Description: Configuring OS Parameter
Status: Success
Created: September 8, 2024 12:03:51 AM MST
Message:
Task Name             Start Time                          End Time                            Status
--------------------  ----------------------------------  ----------------------------------  -------
```

```
Setting up memlock.   September 8, 2024 12:03:51 AM MST   September 8, 2024 12:03:51 AM MST
Success
Setting up HugeSpace  September 8, 2024 12:03:51 AM MST   September 8, 2024 12:03:51 AM MST
Success
```

odaadmcli Hardware Monitoring Commands

Use the `hardware monitoring` commands to display hardware configurations.

- odaadmcli show localdisk
  Use the `odaadmcli show localdisk` command to show cooling details.
- odaadmcli show cooling
  Use the `odaadmcli show cooling` command to show cooling details.
- odaadmcli show env_hw
  Use the `odaadmcli show env_hw` command to display information about the environment and hardware.
- odaadmcli show fs
  Use the `odaadmcli show fs` command to display filesystem details.
- odaadmcli show memory
  Use the `odaadmcli show memory` command to display memory details.
- odaadmcli show network
  Use the `odaadmcli show network` command to show network details.
- odaadmcli show power
  Use the `odaadmcli show power` command to display power supply details.
- odaadmcli show processor
  Use the `odaadmcli show processor` command to display processor details.
- odaadmcli show server
  Use the `odaadmcli show server` command to display server details.

odaadmcli show localdisk

Use the `odaadmcli show localdisk` command to show cooling details.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To show cooling details:

```
odaadmcli show localdisk [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
# odaadmcli show localdisk
        NAME            PATH            TYPE
STATUS                  STATE_IN_ILOM

        lpd_0           /dev/sda        SSD
PredictiveFailure       OK
        lpd_1           /dev/sdb        SSD
OK                      OK
```

odaadmcli show cooling

Use the `odaadmcli show cooling` command to show cooling details.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To show cooling details:

```
odaadmcli show cooling [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

odaadmcli show env_hw

Use the `odaadmcli show env_hw` command to display information about the environment and hardware.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To display environment and hardware details:

```
odaadmcli show env_hw [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-229 Displaying Environment and Hardware Details

To display the hardware details, enter the command `odaadmcli show env_hw`. The results show a bare metal Oracle Database Appliance system.

```
# odaadmcli show env_hw

BM X7-2 Medium
```

odaadmcli show fs

Use the `odaadmcli show fs` command to display filesystem details.

File Path

```
/opt/oracle/oak/bin/odaadmcli
```

Syntax

To display filesystem details:

```
   odaadmcli show fs [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-230 Displaying Filesystem Details

```
# odaadmcli show fs

Type     Total Space   Free Space      Total DG Space  Free DG Space
ext3    30237M         7763M           -               -
ext3    484M            416M           -               -
ext3    60475M        38149M           -               -
ext3    100793M       22060M           -               -

acfs    102400M       102158M         4894016M        2418668M
acfs    102400M       100501M         4894016M        2418668M
acfs    102400M       100601M         4894016M        2418668M

(Continued)
Diskgroup Mount Point
          /
          /boot
          /opt
          /u01
DATA      /u02/app/oracle/oradata/ACFSDB1
DATA      /u02/app/oracle/oradata/ACFSDB2
DATA      /u02/app/oracle/oradata/EE12NCDB
```

odaadmcli show memory

Use the `odaadmcli show memory` command to display memory details.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To show memory details:

```
odaadmcli show memory [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Example 16-231 Display Memory Details

```
# odaadmcli show memory

NAME     HEALTH  HEALTH_DETAILS   PART_NO.          SERIAL_NO.
DIMM_0   OK       -               3A4K40BB1-CRC        00CE01154602EADA96
DIMM_11  OK       -               3A4K40BB1-CRC        00CE01154602EADADA
DIMM_3   OK       -               3A4K40BB1-CRC        00CE01154602EADBC7
DIMM_8   OK       -               3A4K40BB1-CRC        00CE01154602EADBA0

(Continued)
LOCATION MANUFACTURER  MEMORY_SIZE  CURR_CLK_SPEED  ECC_Errors
P0/D0    Samsung        32 GB       2400 MHz        0
P0/D1    Samsung        32 GB       2400 MHz        0
P0/D3    Samsung        32 GB       2400 MHz        0
P0/D8    Samsung        32 GB       2400 MHz        0
```

odaadmcli show network

Use the `odaadmcli show network` command to show network details.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To show network details:

```
odaadmcli show network [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Example 16-232 Showing Network Details

```
# odaadmcli show network
NAME            HEALTH HEALTH_DETAILS LOCATION PART_NO MANUFACTURER
MAC_ADDRESS LINK_DETECTED DIE_TEMP

Ethernet_NIC_0     OK     -              NET0     i210    INTEL
00:10:E0:DD:9D:14 no (em1) N/A
Ethernet_NIC_1     OK     -              NET1     X710/X557-AT Intel
3C:FD:FE:78:93:92 no (p2p3) N/A
Ethernet_NIC_2     OK     -              NET2     X710/X557-AT Intel
3C:FD:FE:78:93:91 yes (p2p2) N/A
Ethernet_NIC_3     OK     -              NET3     X710/X557-AT Intel
3C:FD:FE:78:93:90 yes (p2p1) N/A
Ethernet_NIC_4     OK     -              NET4     BCM57414 Broadcom
B0:26:28:3F:D8:B8 yes (p7p2) N/A
Ethernet_NIC_5     OK     -              NET5     X710/X557-AT Intel
3C:FD:FE:78:93:93 no (p2p4) N/A
Ethernet_NIC_6     OK     -              NET6     BCM57414 Broadcom
B0:26:28:3F:D8:B0 yes (p7p1) N/A
```

odaadmcli show power

Use the `odaadmcli show power` command to display power supply details.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

Run this command on the first node, to display power supply details on Oracle Database Appliance High-Availability (HA) models:

```
odaadmcli show power [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Example 16-233 Displaying Power Supply Details

```
# odaadmcli show power

NAME            HEALTH  HEALTH_DETAILS   PART_NO.      SERIAL_NO.
Power_Supply_0  OK            -          7079395     476856Z+1514CE056G
```

```
(Continued)
LOCATION    INPUT_POWER   OUTPUT_POWER   INLET_TEMP          EXHAUST_TEMP
PS0         Present       112 watts      28.000 degree C     34.938 degree C
```

odaadmcli show processor

Use the `odaadmcli show processor` command to display processor details.

File Path

```
/opt/oracle/oak/bin/odaadmcli
```

Syntax

To show processor details:

```
odaadmcli show processor  [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Example 16-234 Displaying Processor Details

```
# odaadmcli show processor

NAME     HEALTH HEALTH_DETAILS  PART_NO.  LOCATION   MODEL
CPU_0     OK      -             060F P0   (CPU 0)    Intel(R) Xeon(R) CPU
E5-2630

(Continued)
MAX_CLK_SPEED   TOTAL_CORES     ENABLED_CORES
2.200 GHz        10              10
```

odaadmcli show server

Use the `odaadmcli show server` command to display server details.

File Path

```
/opt/oracle/oak/bin/odaadmcli
```

Syntax

To show server details:

```
   odaadmcli show server [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Example 16-235 Displaying Server Details

```
# odaadmcli show server

Power State : On
Open Problems : 0
Model : ORACLE SERVER X7-2
Type : Rack Mount
Part Number : 7320190
Serial Number : 1605NM10JJ
Primary OS : Not Available
ILOM Address : 10.209.8.215
ILOM MAC Address : 00:10:E0:95:98:F0
Description : Oracle Database Appliance X7-2 Small 1606NM1s02
Locator Light : Off
Actual Power Consumption : 135 watts
Ambient Temperature : 24.250 degree C
Open Problems Report : System is healthy
```

Storage Commands

Understand the commands to perform storage operations and diagnostics.

- odaadmcli asmops diskgroup
  Use the `odaadmcli asmops diskgroup` command to manage Oracle ASM disk group.
- odaadmcli expand storage
  Use the `odaadmcli expand storage` command to expand storage.
- odacli list-dgdisks
  Use the `odacli list-dgdisks` command to display details about the available disks.
- odacli validate-storagetopology
  Use the `odacli validate-storagetopology` command to check the cable connections between the system controllers and the storage shelf, as well as the cable connection to the storage expansion shelf (if one is installed).
- odaadmcli power disk
  Use the `odaadmcli power disk` command to power a disk on or off for NVMe drives only.
- odaadmcli show controller
  Use the `odaadmcli show controller` command to display details of the controller.
- odaadmcli show disk
  Use the `odaadmcli show disk` command to display the status of a single disk or of all disks on the system.
- odaadmcli show diskgroup
  Use the `odaadmcli show diskgroup` command to list configured diskgroups or display a specific diskgroup configuration.

- odaadmcli show raidsyncstatus
  Use the `odaadmcli show raidsyncstatus` command to display the RAID SYNC status.
- odaadmcli show storage
  Use the `odaadmcli show storage` command to show the storage controllers, expanders, and disks.
- odaadmcli stordiag
  Use the `odaadmcli stordiag` command to collect detailed information for each disk or NVMe.

odaadmcli asmops diskgroup

Use the `odaadmcli asmops diskgroup` command to manage Oracle ASM disk group.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

```
  odaadmcli asmops diskgroup -G diskgroup_name  -o {add|remove|offline|
online} -p path -D disk_name -f -w -t timeout [-h]
```

Parameters

Parameter	Description
`-G`	(Mandatory) Specifies the disk group name.
`-o`	(Mandatory) Specifies the operation to be performed on Oracle ASM disk. Add disk to disk group, remove from disk group, offline the disk or online the disk.
`-p`	(Mandatory for addition of disks) Specify the disk path during addition of disk to Oracle disk group.
`-D`	(Mandatory) Specify Oracle ASM disk name or alias.
`--help`, `-h`	(Optional) Displays help for using the command.
`-f`	(Optional) Specifies if you want to force the operation.
`-w`	(Optional) Specifies if you want to wait for the operation to complete. This option is valid for the disk online operation.
`-t`	(Optional) Specifies the duration after which the disks are dropped by Oracle ASM after a disk offline operation. The time out values can be specified in the format 35m, 2h, and so on. This option is valid for the disk offline operation.

Usage Notes

Run the command on the Oracle Database Appliance system to manage Oracle ASM disk group.

odaadmcli expand storage

Use the `odaadmcli expand storage` command to expand storage.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

```
odaadmcli expand storage -ndisk number of disks to be added -enclosure {0|1} [-h]
```

Parameters

Parameter	Description
`--enclosure {0|1}`	Defines the JBOD or shelf number. The base storage shelf is 0. The storage expansion shelf is 1. This option is only available for multi-node platforms. It is not supported on Oracle Database Appliance single node platforms.
`--help`, `-h`	(Optional) Displays help for using the command.
`-ndisk`	Defines the number of disks to be added in the enclosure.

Usage Notes

On high-availability (HA) Oracle Database Appliance hardware models, run the command only on the first node.

Various storage options are available. On multi-node platforms, the base storage is SSD and you can choose to add either SSD or HDD disks. When adding a storage expansion shelf, all disks must be the same type (you cannot combine SSD and HDD disks).

> **Note:**
>
> In addition to reviewing the `expand storage` command, it is important to follow the procedures for adding storage. Cabling validation and disk placement is important. In some cases, you must move disks to different slots to successfully expand storage.

odacli list-dgdisks

Use the `odacli list-dgdisks` command to display details about the available disks.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To list available disk details:

```
odacli list-dgdisks [-h] [-dg] [-dp] [-j]
```

Parameters

Parameter	Description
--json, -j	(Optional) Displays JSON output.
--help, -h	(Optional) Displays help for using the command.
--diskgroup, -dg	(Optional) Displays all disks in the specified disk group. The possible values are: ALL, DATA, RECO, REDO, FLASH. If no value is specified, then ALL disks are displayed.
--diskpath, -dp	(Optional) Displays all disks that contains the specified value in the path. If no value is specified, then all disks are displayed.

Usage Notes

Example 16-236 Listing All Available Disks on the System

```
# odacli list-dgdisks -dg ALL

Name                      Group Mount St Group Num Mode St State
Redundancy Fail Grp                  Path                              Health Inc
------------------------ ----- -------- --------- ------- --------
---------- ------------------------- -------------------------------- ------ ---
HDD_E0_S19_576667504P1    DATA  CACHED   1         ONLINE  NORMAL
UNKNOWN    HDD_E0_S19_576667504P1    AFD:HDD_E0_S19_576667504P1     0      0
HDD_E0_S08_577087696P1    DATA  CACHED   1         ONLINE  NORMAL
UNKNOWN    HDD_E0_S08_577087696P1    AFD:HDD_E0_S08_577087696P1     0      0
HDD_E0_S11_576021940P1    DATA  CACHED   1         ONLINE  NORMAL
UNKNOWN    HDD_E0_S11_576021940P1    AFD:HDD_E0_S11_576021940P1     0      0
HDD_E0_S13_576660012P1    DATA  CACHED   1         ONLINE  NORMAL
UNKNOWN    HDD_E0_S13_576660012P1    AFD:HDD_E0_S13_576660012P1     0      0
HDD_E0_S15_577131668P1    DATA  CACHED   1         ONLINE  NORMAL
UNKNOWN    HDD_E0_S15_577131668P1    AFD:HDD_E0_S15_577131668P1     0      0
HDD_E0_S09_576662448P1    DATA  CACHED   1         ONLINE  NORMAL
UNKNOWN    HDD_E0_S09_576662448P1    AFD:HDD_E0_S09_576662448P1     0      0
HDD_E0_S18_577136300P1    DATA  CACHED   1         ONLINE  NORMAL
UNKNOWN    HDD_E0_S18_577136300P1    AFD:HDD_E0_S18_577136300P1     0      0
```

Example 16-237 Listing Disks with the Specified Disk Path on the System

```
# odacli list-dgdisks -dp HDD_E0_S10_577135552

Name                    Group Mount St Group Num Mode St State    Redundancy
Fail Grp                Path                          Health Inc
----------------------- ----- -------- --------- ------- -------- ----------
--------------------- -------------------------- ------ ---
```

```
HDD_E0_S10_577135552P1 DATA  CACHED   1         ONLINE  NORMAL   UNKNOWN
HDD_E0_S10_577135552P1 AFD:HDD_E0_S10_577135552P1  0      0
HDD_E0_S10_577135552P2 RECO  CACHED   2         ONLINE  NORMAL   UNKNOWN
HDD_E0_S10_577135552P2 AFD:HDD_E0_S10_577135552P2  0      0
```

odacli validate-storagetopology

Use the `odacli validate-storagetopology` command to check the cable connections between the system controllers and the storage shelf, as well as the cable connection to the storage expansion shelf (if one is installed).

Oracle recommends that you run the `odacli validate-storagetopology` command before deploying the system to ensure that the cabling is correct. This will avoid and prevent problems during deployment due to incorrect or missing cable connections. If the cabling is not correct, you will see errors in your output.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli validate-storagetopology [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

odaadmcli power disk

Use the `odaadmcli power disk` command to power a disk on or off for NVMe drives only.

File Path

```
/opt/oracle/oak/bin/odaadmcli
```

Syntax

To power an NVMe disk on or off:

```
odaadmcli power disk {on|off|status} disk_name [-h]
```

Parameters

Parameter	Description
disk_name	Defines the disk resource name. The resource name format is pd_[0..3].
`{on|off|status}`	Power on a disk, power off a disk, display status.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-238 Powering a Disk Off

This example shows how to power off a disk.

```
# odaadmcli power disk off pd_00
Disk 'pd_00' is already part of ASM

Are you sure you want to power OFF the disk?: 'pd_00'?
[yes/no]:
yes
Powered OFF pd_00
```

Example 16-239 Checking the Disk Status

This example shows how to get the status of disk pd_00.

```
# odaadmcli power disk status pd_00
The disk is powered ON
```

odaadmcli show controller

Use the `odaadmcli show controller` command to display details of the controller.

File Path

```
/opt/oracle/oak/bin/odaadmcli
```

Syntax

To display details of the controller:

```
odaadmcli show controller id [-h]
```

Parameters

Parameter	Description
`controller_id`, *id*	Defines the controller.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-240 Showing Controller Details

```
 # odaadmcli show controller 1
Controller [1] information:
        sun-controller-id = nvme:1b:00.00
        sun-id = nvme:1b:00.00
        sun-controller-manufacturer = Samsung
        pci-vendor-id = 0x0000144d
        sun-controller-model = 0xa821
        pci-device-id = 0x0000a821
        sun-controller-type = NVMe
        sun-card-manufacturer = Sun Microsystems
        pci-subvendor-id = 0x0000108e
        sun-card-model = 0xa803
```

```
pci-subdevice-id = 0x0000a803
pci-address = 1b:00.0
sun-version-firmware = KPYA7R3Q
sun-serial-number = S2LHNAAH101008
sun-product-name = MS1PC2DD3ORA3.2T
pci-slot-number = 11
nvme-power-control = 1
sun-nac-name = /SYS/DBP/NVME1
```

odaadmcli show disk

Use the `odaadmcli show disk` command to display the status of a single disk or of all disks on the system.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To display the status of all disks on the system:

```
odaadmcli show  disk [-h]
```

To display the status of a single disk:

```
odaadmcli show disk disk_name [-h]
```

Parameters

Parameter	Description
disk_name	(Optional) Define the disk resource name. The resource name format is `pd_[0..3]`.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-241 Displaying the Status of All Disks

To display the status of all the disks on the system:

```
# odaadmcli show disk
NAME        PATH             TYPE     STATE     STATE_DETAILS
pd_00     /dev/nvme0n1 NVD         ONLINE     Good
pd_01     /dev/nvme1n1 NVD         ONLINE     Good
```

Example 16-242 Displaying the Status of a Single Disk

To display the status of a disk named `pd_00`:

```
# odaadmcli show disk pd_00

The Resource is : pd_00
ActionTimeout : 1500
ActivePath : /dev/nvme0n1
```

```
 AsmDiskList : |data_00||reco_00|
AutoDiscovery : 1
AutoDiscoveryHi : |data:80:NVD||reco:20:NVD|
CheckInterval : 300
ColNum : 0
CriticalWarning : 0
DependListOpr : add
 Dependency : |0|
DiskId : 360025380144d5332
DiskType : NVD
Enabled : 1
ExpNum : 19
HbaPortNum : 10
IState : 0
Initialized : 0
IsConfigDepende : false
 ModelNum : MS1PC2DD3ORA3.2T
 MonitorFlag : 1
MultiPathList : |/dev/nvme0n1|
Name : pd_00
NewPartAddr : 0
OSUserType : |userType:Multiuser|
PlatformName : X7_1_LITE_S
PrevState : Invalid
PrevUsrDevName :
SectorSize : 512
SerialNum : S2LHNAAH000001
Size : 3200631791616
SlotNum : 0
SmartDiskWarnin : 0
SmartTemperatur : 37
State : Online
StateChangeTs : 1465263789
StateDetails : Good
TotalSectors : 6251233968
TypeName : 0
UsrDevName : NVD_S00_S2LHNAAH101026
VendorName : Samsung
gid : 0
 mode : 660
uid : 0
```

odaadmcli show diskgroup

Use the `odaadmcli show diskgroup` command to list configured diskgroups or display a specific diskgroup configuration.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To list configured diskgroups:

```
odaadmcli show diskgroup [-h]
```

To display DATA configurations:

```
odaadmcli show diskgroup [DATA] [-h]
```

To display RECO configurations:

```
odaadmcli show diskgroup [RECO] [-h]
```

Parameters

Parameter	Description
DATA	(Optional) Displays the DATA diskgroup configurations.
RECO	(Optional) Displays the RECO diskgroup configurations.
`--help, -h`	(Optional) Displays help for using the command.

Example 16-243 Listing All Diskgroups

```
# odaadmcli show diskgroup

DiskGroups
----------
DATA
RECO
```

Example 16-244 Displaying DATA Configurations

```
# odaadmcli show diskgroup DATA

ASM_DISK  PATH                           DISK   STATE   STATE_DETAILS

data_00   /dev/NVD_S00_S2LHNAAH101026p1  pd_00  ONLINE   Good
data_01   /dev/NVD_S01_S2LHNAAH101008p1  pd_01  ONLINE   Good
```

odaadmcli show raidsyncstatus

Use the `odaadmcli show raidsyncstatus` command to display the RAID SYNC status.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To display the status of RAID SYNC:

```
odaadmcli show raidsyncstatus [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-245 Displaying the RAID SYNC Status

To display the RAID SYNC details and status:

```
# odaadmcli show raidsyncstatus

Raid Type  Raid Device  Raid Status  maintainPdFailHistory  Rebuildrate
H/W Raid    /dev/sda     Optimal       ON                     30%
```

odaadmcli show storage

Use the `odaadmcli show storage` command to show the storage controllers, expanders, and disks.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To display the storage controllers, expanders, and disks:

```
odaadmcli show storage [-h]
```

To show storage errors:

```
odaadmcli show storage -errors [-h]
```

Parameters

Parameter	Description
`--errors`	(Optional) Shows storage errors.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-246 Displaying Storage Devices

The following example displays details about the storage devices on an appliance.

```
# odaadmcli show storage
```

```
==== BEGIN STORAGE DUMP ========
Host Description: Oracle Corporation:ORACLE SERVER X6-2
Total number of controllers: 2
    Id = 0
    Pci Slot = 10
    Serial Num = S2LHNAAH101026
     Vendor = Samsung
   Model = MS1PC2DD3ORA3.2T
   FwVers = KPYA7R3Q
   strId = nvme:19:00.00
      Pci Address = 19:00.0

      Id = 1
      Pci Slot = 11
      Serial Num = S2LHNAAH101008
      Vendor = Samsung
    Model = MS1PC2DD3ORA3.2T
    FwVers = KPYA7R3Q
     strId = nvme:1b:00.00
  Pci Address = 1b:00.0


Total number of expanders: 0
Total number of PDs: 2
   /dev/nvme0n1 Samsung NVD 3200gb slot: 0 pci : 19
   /dev/nvme1n1 Samsung NVD 3200gb slot: 1 pci : 1

==== END STORAGE DUMP =========
```

odaadmcli stordiag

Use the `odaadmcli stordiag` command to collect detailed information for each disk or NVMe.

File Path

`/opt/oracle/oak/bin/odaadmcli`

Syntax

To collect storage diagnostics for disks and NVM Express (NVMe):

```
odaadmcli stordiag [-n] [-h]
```

Parameters

Parameter	Description
`-n`*disk_name*	Defines the disk resource name. The resource name format is `pd_[0..3]`.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-247 Displaying NVMe Details

To display detailed information for NVMe `pd_00`:

```
# odaadmcli stordiag pd_00
```

17

Oracle Database Appliance Command-Line Interface for KVM

Use `odacli` commands to manage Oracle Linux Kernel-based Virtual Machine (KVM) on Oracle Database Appliance.

> **Note:**
>
> If you have enabled multi-user access, then you can run all supported ODACLI commands on KVM and DB systems, after logging in as `odaadmin`.

- About Command-Line Interface for Kernel-Based Virtual Machine (KVM)
  The `odacli` commands are the primary method for performing database and system administration on Oracle Database Appliance Kernel-Based Virtual Machine (KVM).
- About ODACLI Commands Supported on Oracle KVM Database System
  Understand the `odacli` commands supported on Oracle Database Appliance Kernel-Based Virtual Machine (KVM) Database System in this release.
- odacli DB System Commands for KVM
  Use the odacli DB system commands to manage databases on Oracle Database Appliance for Oracle KVM.
- odacli VM Storage Commands for KVM
  Use the odacli virtual machine storage commands to manage storage on Oracle Database Appliance for Oracle KVM.
- odacli Virtual Disk Commands for Oracle KVM
  Use the odacli virtual disk commands to manage virtual disks on Oracle Database Appliance for Oracle KVM.
- odacli CPU Pool Commands for Bare Metal and KVM
  Use odacli commands to manage resource allocations on Oracle Database Appliance for bare metal and KVM deployments.
- odacli VM Network Commands for KVM
  Use the odacli virtual machine network commands to manage network on Oracle Database Appliance for Oracle KVM.
- odacli Virtual Machine Commands for Oracle KVM
  Use the odacli virtual machine commands to manage virtual machines on Oracle Database Appliance for Oracle KVM.

About Command-Line Interface for Kernel-Based Virtual Machine (KVM)

The `odacli` commands are the primary method for performing database and system administration on Oracle Database Appliance Kernel-Based Virtual Machine (KVM).

You can perform the following tasks with ODACLI commands on Kernel-Based Virtual Machine (KVM):

- Manage VM instances
- Manage VM storage
- Manage virtual disks
- Manage virtual network
- Manage CPU pools

ODACLI Command Location and Path Configuration

The `odacli` command-line interface is in the following directory:

```
/opt/oracle/dcs/bin/odacli
```

Configure the `root` user account on your Oracle Database Appliance servers to have the `PATH` variable defined to search for odacli commands in the path `/opt/oracle/dcs/bin/odacli`.

ODACLI Syntax

Oracle Appliance Manager command-line interface commands and parameters are case-sensitive.

An `odacli` command uses the following command syntax:

```
odacli command object [options]
```

- *`command`* is an action you want to perform on the appliance. For example: `show`, `locate`, `apply`.
- *`object`* is the target or object on which the `odacli` command performs the operation. For example, you issue a command to create or configure a virtual disk, or a network, or perform other system or software administration. You can also use object abbreviations.
- *`options`* are optional parts of the `odacli` command. Options can consist of one or more options that extend the use of the `odacli` command carried out on an object. Options include additional information about the action that you want to perform on the object. Option names are preceded with a dash. Many options require the name of an object for the command to perform the action that you want to carry out. The help option (`-h`) is an option that is available with every command. When you include the `-h` option, you can obtain additional information about the command that you want to perform.

Oracle Appliance Manager Command-Line Interface Help

Run the following command to see the usage information for all `odacli` commands available for your Oracle Database Appliance:

```
odacli -h
```

Run the following command to see detailed help about a specific `odacli` command:

```
odacli command -h
```

Run the following command to see detailed help about a specific `odacli` command carried out on an object, and the options that are available for you to use with that object.

```
odacli command object -h
```

About ODACLI Commands Supported on Oracle KVM Database System

Understand the `odacli` commands supported on Oracle Database Appliance Kernel-Based Virtual Machine (KVM) Database System in this release.

This release of Oracle Database Appliance supports the provisioning of Oracle KVM database system through `odacli create-dbsystem` or the Browser User Interface. The following ODACLI commands are supported on Oracle KVM Database System on Oracle Database Appliance:

- `odacli ping-agent`
- `odacli delete-agentconfig-parameters`
- `odacli list-agentconfig-parameters`
- `odacli modify-agentconfig-parameters`
- `odacli restore-archivelog`
- `odacli create-auto-logclean-policy`
- `odacli delete-auto-logclean-policy`
- `odacli list-auto-logclean-policy`
- `odacli list-availablepatches`
- `odacli create-backup`
- `odacli delete-backup`
- `odacli create-backupconfig`
- `odacli delete-backupconfig`
- `odacli describe-backupconfig`
- `odacli list-backupconfigs`
- `odacli modify-backupconfig`
- `odacli describe-backupreport`
- `odacli describe-component`
- `odacli set-credential`
- `odacli create-database`
- `odacli delete-database`
- `odacli describe-database`
- `odacli irestore-database`
- `odacli list-databases`
- `odacli modify-database`

- odacli recover-database
- odacli register-database
- odacli configure-dataguard
- odacli deconfigure-dataguard
- odacli failover-dataguard
- odacli reinstate-dataguard
- odacli switchover-dataguard
- odacli describe-dataguardstatus
- odacli list-dataguardstatus
- odacli create-dbhome
- odacli delete-dbhome
- odacli describe-dbhome
- odacli list-dbhomes
- odacli create-dbstorage
- odacli delete-dbstorage
- odacli describe-dbstorage
- odacli list-dbstorages
- odacli describe-job
- odacli list-jobs
- odacli create-jobdata-retention-policy
- odacli delete-jobdata-retention-policy
- odacli list-jobdata-retention-policy
- odacli describe-latestpatch
- odacli create-logcleanjob
- odacli describe-logcleanjob
- odacli list-logcleanjobs
- odacli list-logspaceusage
- odacli describe-network
- odacli list-networks
- odacli describe-networkinterface
- odacli list-networkinterfaces
- odacli list-nodes
- odacli create-objectstoreswift
- odacli delete-objectstoreswift
- odacli describe-objectstoreswift
- odacli list-objectstoreswifts

- `odacli modify-objectstoreswift`
- `odacli list-osconfigurations`
- `odacli modify-osconfigurations`
- `odacli list-pendingjobs`
- `odacli create-prepatchreport`
- `odacli delete-prepatchreport`
- `odacli describe-prepatchreport`
- `odacli list-prepatchreports`
- `odacli create-purge-jobdata-job`
- `odacli list-purge-jobdata-jobs`
- `odacli describe-schedule`
- `odacli list-schedules`
- `odacli modify-schedule`
- `odacli list-scheduled-executions`
- `odacli describe-systemcomponent`
- `odacli list-tdebackupreports`
- `odacli restore-tdewallet`

> **Note:**
>
> No other ODACLI commands than the ones above are supported on Oracle Database Appliance KVM Database System, in this release.

odacli DB System Commands for KVM

Use the odacli DB system commands to manage databases on Oracle Database Appliance for Oracle KVM.

- odacli create-dbsystem
  Use the `odacli create-dbsystem` command to create a new database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-dbfileattributes
  Use the `odacli modify-dbfileattributes` command to properly set the file attributes (permissions and ownership) of files on Oracle ASM disk group on a bare metal host to support a dbsystem. Only run this command if prompted by the `odacli create-dbsystem` command.
- odacli list-dbsystems
  Use the `odacli list-dbsystems` command to view the database systems on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

- odacli describe-dbsystem
  Use the `odacli describe-dbsystem` command to describe a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli describe-dbsystem-image
  Use the `odacli describe-dbsystem-image` command to view the supported Oracle Grid Infrastructure and Oracle Database versions and the availability of the clone files and Database System KVM image in the Oracle Database Appliance system.
- odacli start-dbsystem
  Use the `odacli start-dbsystem` command to start a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli stop-dbsystem
  Use the `odacli stop-dbsystem` command to stop a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli delete-dbsystem
  Use the `odacli delete-dbsystem` command to delete a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

odacli create-dbsystem

Use the `odacli create-dbsystem` command to create a new database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli create-dbsystem [-f] [-p json_payload_path] [-t template] [-ta] [-j] [-h]
```

Parameters

Parameter	Description
`--params`, `-p`	(Optional) Specifies the input parameters file path for DB system creation.
`--force`, `-f`	(Optional) Forces the operation with the specified parameters.
`--template`, `-t`	(Optional) Specifies the path where the template with sample values will be created.
`--template-annotated`, `-ta`	(Optional) Displays the DB system template with field descriptions.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

For example JSON files that you can use to create the DB system, see the *Example JSON File to Create a High-Availability DB System* and *Example JSON File to Create a Single-Node DB System* topics in this guide.

Example

Create a KVM database system using a JSON file for input parameters.

```
# odacli create-dbsystem -p example_system.json

Job details
```

```
------------------------------------------------------------
                     ID:  fcfe305c-3e75-4af4-8702-992aa15619d0
            Description:  DB System example_system creation
                 Status:  Created
                Created:  September 8, 2024 9:28:03 AM CST
                Message:

Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
```

Related Topics

- Example JSON File to Create a High-Availability DB System
  Follow the JSON file example to create a JSON file to deploy a High-Availability DB system, with role separation, with the command `odacli create-dbsystem`.
- Example JSON File to Create a Single-Node DB System
  Follow the JSON file example to create a JSON file to deploy a single-node DB System, with role separation, with the command `odacli create-dbsystem`.
- odacli describe-dbsystem-image
  Use the `odacli describe-dbsystem-image` command to view the supported Oracle Grid Infrastructure and Oracle Database versions and the availability of the clone files and Database System KVM image in the Oracle Database Appliance system.

odacli modify-dbsystem

Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
 odacli modify-dbsystem -n name [-cp cpu_pool_name] [-no-cp] [-r]
 [-avd v_disk_to_attach] [-avn v_network]
 [-dvd v_disk_to_detach] [-dvn v_networks_to_detach] [-en] [-gw gateway]
 [-ip ip_address] [-nm netmask] [-tvm vm_name]
[-sip scan_ip_address] [-sn scan_name] [-s shape]
 [-vips vip_address] [-vt v_network_type] [-m] [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Describes the name of the DB system.
`--redundancy, -r`	(Optional) Specifies the VM storage redundancy for a DB system. The value can be HIGH or MIRROR. This option is not supported for DB systems on non-FLEX disk group. The options can be MIRROR or HIGH. The default option is MIRROR.
`--attach-vdisk, -avd`	(Optional) Specifies the vdisk to attach.
`--attach-vnetwork, -avn`	(Optional) Specifies the name of the vnetwork to attach.
`--cpupool, -cp`	(Optional) Specifies the shared DB System CPU Pool name.
`--detach-vdisk, -dvd`	(Optional) Specifies the vdisk to detach.
`--detach-vnetwork, -dvn`	(Optional) Specifies the name of the vnetworks to detach.

Parameter	Description
`--enable-numa, -en`	(Optional) Enables NUMA configuration for the DB System.
`--gateway, -gw`	(Optional) Specifies the network gateway.
`--ip, -ip`	(Optional) Specifies the network IP address.
`--netmask, -nm`	(Optional) Specifies the netmask of the network.
`--no-cpupool, -no-cp`	(Optional) Specifies to dissociate the current DB System CPU Pool name.
`--target-vm, -tvm`	(Optional) Specifies the DB system host name to attach or detach a vdisk.
`--scan-ips, -sip`	(Optional) Specifies the SCAN IP of the network.
`--scan-name, -sn`	(Optional) Specifies the SCAN of the network.
`--shape, -s`	(Optional) Specifies the new shape. You can specify a higher or lower value than the current shape. This option only changes the shape of the DB system. It does not change the shape of databases in the DB system.
`--vips, -vips`	(Optional) Specifies the virtual IP of the network in the format `vip_name_0:0:vip_ip_0,vip_name_1:1:vip_ip_1`.
`--vnetwork-type, -vt`	(Optional) Specifies the vnetwork type to attach.
`--memory, -m`	(Optional) Defines the memory size. The format is M|G.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- The `odacli modify-dbsystem` command supports the following use cases:
  - attach or detach the shared DB system cpupools.
    Attaching a CPU pool:

    ```
    odacli modify-dbsystem --name dbsystem_name -cp cpupool
    ```

    Detaching a CPU pool:

    ```
    odacli modify-dbsystem --name dbsystem_name --no-cp
    ```

  - Change DB system shape, that is, scale up (increase) or scale down (decrease):

    ```
    odacli modify-dbsystem -n dbsystem name -s new_shape
    ```

  - Enable NUMA for the existing DB system. NUMA is enabled on Oracle Database Appliance release 19.12 and later. For existing DB sytems created before release 19.12, use the `odacli modify-dbsystem --enable_numa` command to enable the NUMA for the DB system. This command includes enabling the NUMA for the VM configuration, operation system, and the database inside the DB system.

    ```
    odacli modify-dbsystem --name dbsystem name --enable_numa
    ```

  - Attach or detach new network to the DB system. Before attaching a vnetwork, ensure that the vnetwork exists. You must specify the IP address and netmask to configure the

new network interface inside the DB system. If the network type is `database` or `dataguard`, specify the VIPs, SCAN name, and SCAN IPs.

```
odacli modify-dbsystem -n dbsystem1 -avn vnet2 -t Dataguard -ip
192.168.10.119,192.168.10.120 -nm 255.255.240.0 -gw 192.168.10.1 -sn
scan1 -sip 192.168.10.30,192.168.10.31 -vips
vip1:0:192.168.15.57,vip2:1:192.168.15.58
```

To detach a network from DB system:

```
odacli modify-dbsystem -n dbsystem1 -dvn vnet2
```

- Starting with Oracle Database Appliance release 19.23, the `odacli modify-dbsystem --shape` command only changes the shape of the DB system; it does not change the shapes of the database inside the DB system any more. You must modify database shapes accordingly based on the new shape of the DB system. Ensure that you scale up database shapes after the DB system shape is scaled up. Similarly, scale down database shapes before DB system shape is scaled down, otherwise databases could fail to start because of reduced DB system memory.
- To attach a vdisk to a DB system, note the following:
  - The vdisk must already exist.
  - When you attach a vdisk, the vdisk attaches to the DB system and displays in the DBVM operating system as a block device, for example, `/dev/vdc`. You must format and create a file system with the block device and maintain the device in the DB system.
  - For high-availability systems, you must specify the DB system to attach the vdisk, using the `--target-vm` option. Specify the VM host name without the domain name in the `--target-vm` option. You can view the VM host name with the `odacli describe-dbsystem` command.
  - On high-availability systems, the vdisk attached to the two nodes on the DB system is local to the DBVM it is attached to. The vdisk is not shared and cannot be shared between two DBVMs in the DB system.
- To detach a vdisk from a DB system, note the following:
  - Detaching a vdisk from the DB system removes the vdisk from the DBVM. The block device, for example, `/dev/vdc` is removed from the DBVM operating system level. You must ensure that sufficient care is exercised at the operating system level, for example, properly back up the content of the file system, remove the block device from logical volume manager, or remove the block device from `/etc/fstab`, before you detach the vdisk. Without proper handling, the DBVM could fail to restart after running the detach vdisk operation.
  - For high-availability systems, you must specify the DB system from which to detach the vdisk, using the `--target-vm` option. Specify the VM host name without the domain name in the `--target-vm` option. You can view the VM host name with the `odacli describe-dbsystem` command.
  - If the detach vdisk operation fails, then you can retry the detach operation with the `--force, -f` parameter to ensure that the vdisk is detached from the device block list and the metadata as well.
- The attach and detach vdisk operations restart the DB system.

Example: Modify a KVM DB system

```
# odacli modify-dbsystem -n example_system1 -cp example_pool -s new_shape
```

Example: Increase the DB System memory to 24 GB

The command supports increasing the DB system memory. To reduce the DB system memory, use the `--shape` option with the command `odacli modify-dbsystem`.

```
# odacli modify-dbsystem -n dbsystem_name -m 24G
```

Related Topics

- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

odacli modify-dbfileattributes

Use the `odacli modify-dbfileattributes` command to properly set the file attributes (permissions and ownership) of files on Oracle ASM disk group on a bare metal host to support a dbsystem. Only run this command if prompted by the `odacli create-dbsystem` command.

Syntax

```
odacli modify-dbfileattributes [-in db_name] [-j] [-h]
```

Parameters

Parameter	Description
`--dbname`, `-in`	(Optional) Describes the name of the database for which you want to modify the file attributes.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

- To create the database system, access control must be enabled in Oracle ASM running on bare metal system. When the bare metal system is newly provisioned or patched to Oracle Database Appliance release 19.25, access control is already enabled. This helps you set appropriate file ownerships for all the files created in DB systems and on bare metal.
- Before you create the database system, the files on Oracle ASM disk groups that do not have file ownership must be set up with ownership. The creation of dbsystem fails, and you are prompted to run the `odacli modify-dbfileattributes` command to set up the file ownership and permission. Since there can be many files on Oracle ASM disk group, the user can run this step at their own convenience. Use the `odacli modify-dbfileattributes` command to set up the file ownerships.

- This command sets the file attributes (permissions and ownership) of all files on Oracle ASM disk group. The files include database files and also Oracle Clusterware files such as OCR (Oracle Clusterware Registry). Before creating the first DB system, set up attributes for files owned by both Oracle Grid Infrastructure and Oracle Database by running the `odacli modify-dbfileattributes` command with no arguments.
- If the overall operation was successful for many files, but failed for some files, then you can run the command again by specifying the database names.

Example

Modify the file attributes of the databases.

```
# odacli modify-dbfileattributes -in example_db_name
```

odacli list-dbsystems

Use the `odacli list-dbsystems` command to view the database systems on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli list-dbsystems [-j] [-h]
```

Parameters

Parameter	Description
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

List the KVM database systems.

```
# odacli list-dbsystems
Name                 Shape       Cores  Memory    Status
Created                 Updated
-------------------  ----------  -----  --------- -------
----------------------  ----------------------
dbsystem1            odb4        4      32.00 GB  CONFIGURED  2020-12-08
17:39:00 CST  2020-12-08 17:39:00 CST
dbsystem2            odb2        2      16.00 GB  CREATING    2020-12-16
09:28:03 CST  2020-12-16 09:28:03 CST
```

odacli describe-dbsystem

Use the `odacli describe-dbsystem` command to describe a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli describe-dbsystem -n db_system_name [-p payload] [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Describes the name of the DB system.
`--payload, -p`	(Optional) Displays the payload describing the specified DB system. The option that can be specified is `COPY`.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

Image version is the version which the DB System was created from. Current version is the current version of the DB system. For instance, if the DB System was created on 19.25 and it was patched to 19.25, the output is as follows:

```
           Image version: 19.24.0.0.0
         Current version: 19.25.0.0.0
```

Example

View details of a KVM database system on a single-node system.

```
# odacli describe-dbsystem -n example_system1

DB System details
----------------------------------------------------------------------------
--
                        ID: 04f718f8-ddc9-4b45-8e8e-07d14165eb9c
                      Name: example_system2
             Image version: 19.25.0.0.0
           Current version: 19.25.0.0.0
                     Shape: odb2
              Cluster name: dbsa4b75be52
              Grid version: 19.25.0.0.241015
                    Memory: 16.00 GB
              NUMA enabled: YES
                    Status: CONFIGURED
                   Created: 2023-04-19 10:00:08 UTC
                   Updated: 2023-04-19 10:04:02 UTC

CPU Pool
-------------------------
                      Name: c4bfa26faa
        Number of cores: 2
                      Host: n1
      Effective CPU set: 0-1,20-21
            Online CPUs: 0, 1, 20, 21
           Offline CPUs: NONE


                      Host: n2
      Effective CPU set: 0-1,20-21
            Online CPUs: 0, 1, 20, 21
```

```
             Offline CPUs: NONE

 VM Storage
--------------------------
               Disk group: DATA
              Volume name: S4BFA26FAA
            Volume device: /dev/asm/s4bfa26faa-497
                     Size: 400.00 GB
              Mount Point: /u05/app/sharedrepo/example_system

 VMs
--------------------------
                     Host: host1
                  VM Name: x4bfa26faa
             VM Host Name: example_domain1
            VM image path: /vm_image_path1
             Target State: ONLINE
            Current State: ONLINE


                     Host: host2
                  VM Name: y4bfa26faa
             VM Host Name: example_domain2
            VM image path: /vm_image_path2
             Target State: ONLINE
            Current State: ONLINE

 VNetworks
--------------------------
                     Host: example_domain1
                  VM Name: x4bfa26faa
                   Public: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ens3 /
BRIDGE(pubnet)
                     ASM: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ens4 /
BRIDGE(privasm) VLAN(icbond0.100)
              Interconnect: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ens5 /


                     Host: example_domain2
                  VM Name: y4bfa26faa
                   Public: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ens3 /
BRIDGE(pubnet)
                     ASM: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ens4 /
BRIDGE(privasm) VLAN(icbond0.100)
              Interconnect: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ens5 /

Databases
--------------------------
                     Name: flashvm
              Resource ID: 350b72f7-2351-49e7-ada3-4051af9594f0
              Unique name: flashvmU
              Database ID:
              Domain name: domain_name
               DB Home ID: 617664f6-0a16-4922-8346-9eeb3d7256ee
                    Shape: odb2
                  Version: 19.0.0.0
```

```
                    Edition: EE
                       Type: SI
                       Role: PRIMARY
                      Class: OLTP
                    Storage: ASM
                 Redundancy:
           Target node name:
              Character set: AL32UTF8
          NLS character set:
                   Language: AMERICAN
                  Territory: AMERICA
            Console enabled: false
               SEHA enabled: false
        Associated networks: NONE
           Backup config ID:
         Level 0 Backup Day: sunday
         Autobackup enabled: true
                TDE enabled: true
                   CDB type: false
                   PDB name:
              PDB admin user:
```

View details of a KVM database system on a High Availability system, with InfiniBand.

```
# odacli describe-dbsystem -n example_system3
DB System details
----------------------------------------------------------------------------------
--
                         ID: 0c5280a1-3ee1-438e-ad11-1cee23de17fd
                       Name: example_system3
              Image version: 19.16.0.0.0
            Current version: 19.16.0.0.0
                      Shape: odb2
               Cluster name: dbsa4b75be52
               Grid version: 19.25.0.0.241015
                     Memory: 16.00 GB
               NUMA enabled: YES
                     Status: CONFIGURED
                    Created: 2020-12-10 02:11:11 UTC
                    Updated: 2020-12-10 02:11:11 UTC

 CPU Pool
-------------------------
                       Name: ce2ab55255
            Number of cores: 2
                       Host: node1
          Effective CPU set: 4-5,24-25
                Online CPUs: 4, 5, 24, 25
               Offline CPUs: NONE


                       Host: node2
          Effective CPU set: 4-5,24-25
                Online CPUs: 4, 5, 24, 25
               Offline CPUs: NONE
```

```
 VM Storage
--------------------------
               Disk group: DATA
              Volume name: SE2AB55255
            Volume device: /dev/asm/se2ab55255-187
                     Size: 400.00 GB
              Mount Point: /u05/app/sharedrepo/example_system

 VMs
--------------------------
                     Host: host1
                  VM Name: xe2ab55255
             VM Host Name: host_name1
            VM image path: /vm_image_path1
             Target State: ONLINE
            Current State: ONLINE


                     Host: host2
                  VM Name: ye2ab55255
             VM Host Name: host_name2
            VM image path: /vm_image_path2
             Target State: ONLINE
            Current State: ONLINE

VNetworks
--------------------------
                     Host: node1
                  VM Name: xe2ab55255
                   Public: xxx.xxx.xxx.xxx /xxx.xxx.xxx.xxx / ens3 /
BRIDGE(pubnet)
                      ASM: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ibbond0 /
VF(90:03.5)
              Interconnect: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ibbond1 /
VF(90:01.7)

                     Host: node2
                  VM Name: ye2ab55255
                   Public: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ens3 /
BRIDGE(pubnet)
                      ASM: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ibbond0 /
VF(90:01.0)
              Interconnect: xxx.xxx.xxx.xxx / xxx.xxx.xxx.xxx / ibbond1 /
VF(90:01.6)

 Databases
--------------------------
                     Name: datavm
              Resource ID: 6aac341a-3c73-46eb-af2a-1eada9b57661
              Unique name: datavmU
              Database ID: 3806144946
              Domain name: domain_name
               DB Home ID: dcc928f1-5eb3-477b-875d-050640b7f68c
                    Shape: odb2
                  Version: 19.25.0.0.241015
```

```
                 Edition: EE
                    Type: RAC
                    Role: PRIMARY
                   Class: OLTP
                 Storage: ASM
              Redundancy:
         Target node name:
            Character set: AL32UTF8
        NLS character set:
                Language: AMERICAN
               Territory: AMERICA
          Console enabled: false
             SEHA enabled: false
       Associated networks: NONE
          Backup config ID:
        Level 0 Backup Day: sunday
        Autobackup enabled: true
               TDE enabled: true
                  CDB type: false
                  PDB name:
              PDB admin user:
```

View details of a DB system in the JSON file format

```
# odacli describe-dbsystem -n scaoda815c2 -p
{
  "forced" : null,
  "system" : {
    "name" : "scaoda815c2",
    "shape" : "odb4",
    "customMemorySize" : "32.00 GB",
    "cpuPoolName" : "c519cd5314",
    "useReservedCores" : false,
    "diskGroup" : "DATA",
    "systemPassword" : null,
    "timeZone" : "UTC",
    "enableRoleSeparation" : true,
    "singleNode" : null,
    "failOver" : false,
    "customRoleSeparation" : {
      "groups" : [ {
        "id" : 1001,
        "name" : "oinstall",
        "role" : "oinstall"
      }, {
        "id" : 1002,
        "name" : "dbaoper",
        "role" : "dbaoper"
      }, {
        "id" : 1003,
        "name" : "dba",
        "role" : "dba"
      }, {
        "id" : 1004,
        "name" : "asmadmin",
```

```
        "role" : "asmadmin"
      }, {
        "id" : 1005,
        "name" : "asmoper",
        "role" : "asmoper"
      }, {
        "id" : 1006,
        "name" : "asmdba",
        "role" : "asmdba"
      } ],
      "users" : [ {
        "id" : 1000,
        "name" : "grid",
        "role" : "gridUser"
      }, {
        "id" : 1001,
        "name" : "oracle",
        "role" : "oracleUser"
      } ]
    },
    "redundancy" : "MIRROR"
  },
  "network" : {
    "domainName" : "us.oracle.com",
    "ntpServers" : [ ],
    "dnsServers" : [ "10.209.76.198" ],
    "scanName" : "scaoda815c2-scan",
    "scanIps" : [ "10.32.68.17", "10.32.70.94" ],
    "nodes" : [ {
      "number" : 1,
      "name" : "scaoda815c2n2",
      "ipAddress" : "10.32.68.20",
      "netmask" : "255.255.248.0",
      "gateway" : "10.32.64.1",
      "vipName" : "scaoda815c2n2-vip",
      "vipAddress" : "10.32.68.21"
    }, {
      "number" : 0,
      "name" : "n1",
      "ipAddress" : "10.32.68.18",
      "netmask" : "255.255.248.0",
      "gateway" : "10.32.64.1",
      "vipName" : "n1-vip",
      "vipAddress" : "10.32.68.19"
    } ],
    "publicVNetwork" : null,
    "clusterInterconnect" : null
  },
  "grid" : {
    "language" : "en",
    "enableAFD" : false
  },
  "database" : {
    "name" : "mydb2",
    "uniqueName" : "mydb2u",
    "domainName" : "domainname",
```

```
    "adminPassword" : null,
    "version" : "19.17.0.0.221018",
    "edition" : "EE",
    "type" : "RAC",
    "dbClass" : "OLTP",
    "shape" : "odb4",
    "role" : "PRIMARY",
    "redundancy" : "MIRROR",
    "characterSet" : {
      "characterSet" : "AL32UTF8",
      "nlsCharacterset" : "AL16UTF16",
      "dbTerritory" : "AMERICA",
      "dbLanguage" : "AMERICAN"
    },
    "enableDbConsole" : false,
    "enableSEHA" : false,
    "enableEEHA" : false,
    "enableFlashStorage" : false,
    "rmanBackupPassword" : null,
    "level0BackupDay" : "sunday",
    "enableTDE" : false,
    "tdePassword" : null,
    "enableUnifiedAuditing" : false,
    "isCdb" : true,
    "pdbName" : "pdb1",
    "pdbAdminUser" : "pdbadmin",
    "targetNodeNumber" : 0,
    "dbStorage" : "ASM"
  }
}
```

odacli describe-dbsystem-image

Use the `odacli describe-dbsystem-image` command to view the supported Oracle Grid Infrastructure and Oracle Database versions and the availability of the clone files and Database System KVM image in the Oracle Database Appliance system.

Syntax

```
odacli describe-dbsystem-image [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

View details of a KVM database system image

```
# odacli describe-dbsystem-image
DB System Image details
--------------------------------------------------------------------------------
```

```
--
Component Name Supported Versions Available Versions
------------------- ------------------- -------------------
DBVM            19.10.0.0.0         19.10.0.0.0
GI              19.10.0.0.210119    19.10.0.0.210119
DB              19.10.0.0.210119    19.10.0.0.210119
```

View details of a system where KVM database system image is not available

```
# odacli describe-dbsystem-image
DB System Image details
--------------------------------------------------------------------------------
--
Component Name Supported Versions Available Versions
------------------- ------------------- -------------------
DBVM            19.10.0.0.0         not-available
GI              19.10.0.0.210119    19.10.0.0.210119
DB              19.10.0.0.210119    19.10.0.0.210119
```

The output shows that the KVM Database System image is not yet available on the system. Download the image and run the `odacli update-repository` to import the image to the repository. The output shows that the Oracle Grid Infrastructure and Oracle Database clones are already available on the system.

odacli start-dbsystem

Use the `odacli start-dbsystem` command to start a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli start-dbsystem -n dbsystem_name [-nn node_name] [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Describes the name of the DB system.
`--node-name, -nn`	(Optional) Specifies the bare metal system node name on which to start the DB system VM.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

Start a KVM database system.

```
# odacli start-dbsystem -n example_system

Job details
----------------------------------------------------------------
                     ID:  68044afa-d3cc-4024-9dc8-50fcf847642c
            Description:  DB System scaoda8s008c4 start
                 Status:  Success
```

```
                Created:  December 16, 2020 9:12:51 AM UTC
                Message:
Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
Start DB System                          December 16, 2020 9:12:51 AM UTC
December 16, 2020 9:12:52 AM UTC    Success
```

odacli stop-dbsystem

Use the `odacli stop-dbsystem` command to stop a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli stop-dbsystem -n dbsystem_name [-nn node_name] [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Describes the name of the DB system.
`--force, -f`	(Optional) Forces the operation to stop the DB system.
`--node-name, -nn`	(Optional) Specifies the bare metal system node name on which to stop the DB system VM.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

Stop a KVM database system.

```
# odacli stop-dbsystem -n example_system

Job details
----------------------------------------------------------------
                     ID:  68044afa-d3cc-4024-9dc8-50fcf847642c
            Description:  DB System scaoda8s008c4 start
                 Status:  Success
                Created:  December 16, 2020 9:12:51 AM UTC
                Message:
Task Name                                Start Time
End Time                                 Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
Stop DB System                           December 16, 2020 9:12:51 AM UTC
December 16, 2020 9:12:52 AM UTC    Success
```

odacli delete-dbsystem

Use the `odacli delete-dbsystem` command to delete a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli delete-dbsystem -n dbsystem_name [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	(Mandatory) Describes the name of the DB system.
`--force`, `-f`	(Optional) If specified, then the operation deletes the database system, Oracle Clusterware files, and the database files.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

- If you do not specify the `-f` option, then the delete DB system operation validates that the DB system does not contain any database. If there are any databases within the DB system, then the following error message is displayed:

  ```
  DB System 'name' must not have any database in order to be deleted, found
  following databases: list_of_databases
  ```

- Specify the `-f` option to delete the DB system, the Oracle Clusterware files, and the database files.
- The `odacli delete-dbsystem` command does not delete the database files. This is expected behavior so that you can archive the database files, if required, or manually restore the database with these database files. Use the `--force` option to delete the database files completely.

Example

Delete a DB system.

```
[root@node1ha17c1n1 log]# odacli delete-dbsystem -n dbsfe0bd8
WARNING: Use --force option to delete the database and datafiles.
Do you confirm delete DB System 'dbsfe0bd8'? (Options: y/n, default: n): y
DCS-10045:Validation error encountered: DB System 'dbsfe0bd8' must not have
any database in order to be deleted, found following databases: dbb34f97.


[root@node1ha17c1n1 log]# odacli describe-job -i 41de9e46-e1d4-44c0-b0ab-
c6049579e9f8

Job details
----------------------------------------------------------------
                     ID:  41de9e46-e1d4-44c0-b0ab-c6049579e9f8
            Description:  DB System dbs7283a3 deletion [Forced]
```

```
                 Status:  Success
                Created:  June 13, 2024 8:14:08 PM UTC
                Message:

Task Name                                Node Name                 Start
Time                                End Time
Status
---------------------------------------- -------------------------
----------------------------------------
---------------------------------------- ----------------
Delete database                          node1ha17c1n1          June 13, 2024
8:14:08 PM UTC             June 13, 2024 8:14:09 PM UTC
Success
Remove DB System from Clusterware        node1ha17c1n1          June 13, 2024
8:14:09 PM UTC             June 13, 2024 8:14:09 PM UTC
Success
Delete ASM client cluster config         node1ha17c1n1          June 13, 2024
8:14:09 PM UTC             June 13, 2024 8:14:09 PM UTC
Success
Deprovision DB System VM(s)              node1ha17c1n1          June 13, 2024
8:14:09 PM UTC             June 13, 2024 8:14:10 PM UTC
Success
Delete DB System ACFS filesystem         node1ha17c1n1          June 13, 2024
8:14:10 PM UTC             June 13, 2024 8:14:11 PM UTC
Success
Delete DB System ACFS mount point        node1ha17c1n1          June 13, 2024
8:14:11 PM UTC             June 13, 2024 8:14:11 PM UTC
Success
Delete DB System ASM volume              node1ha17c1n1          June 13, 2024
8:14:11 PM UTC             June 13, 2024 8:14:12 PM UTC
Success
Delete DB System Networks                node1ha17c1n1          June 13, 2024
8:14:12 PM UTC             June 13, 2024 8:14:12 PM UTC
Success
Remove unused network interfaces         node1ha17c1n1          June 13, 2024
8:14:12 PM UTC             June 13, 2024 8:14:12 PM UTC
Success
Delete imported certificates             node1ha17c1n1          June 13, 2024
8:14:12 PM UTC             June 13, 2024 8:14:13 PM UTC
Success
Delete DB System metadata                node1ha17c1n1          June 13, 2024
8:14:13 PM UTC             June 13, 2024 8:14:13 PM UTC              Success
```

odacli VM Storage Commands for KVM

Use the odacli virtual machine storage commands to manage storage on Oracle Database Appliance for Oracle KVM.

- odacli create-vmstorage
  Use the `odacli create-vmstorage` command to create a new VM storage on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli list-vmstorages
  Use the `odacli list-vmstorages` command to list all VM storages on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

- odacli describe-vmstorage
  Use the `odacli describe-vmstorage` command to describe VM storage on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-vmstorage
  Use the `odacli modify-vmstorage` command to modify VM storage configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli delete-vmstorage
  Use the `odacli delete-vmstorage` command to delete a vmstorage configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

odacli create-vmstorage

Use the `odacli create-vmstorage` command to create a new VM storage on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli create-vmstorage -n name -s size -dg disk_group [-r {MIRROR|HIGH}] [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Name assigned to the VM storage.
`--size, -s`	Specifies the storage size. Format is K|M|G|T. Minimum Size is 1024M (1G)
`--diskgroup, -dg`	(Optional) Defines the ASM disk group of the VM storage (Default: DATA).
`--redundancy, -r`	(Optional) Specifies the VM storage redundancy for a DB system. The value can be HIGH or MIRROR. This option is not supported for DB systems on non-FLEX disk group. The options can be MIRROR or HIGH. The default option is MIRROR.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

Create a VM storage named `share1` of 8 GB.

```
# odacli create-vmstorage -n share1 -s 8G
```

odacli list-vmstorages

Use the `odacli list-vmstorages` command to list all VM storages on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli list-vmstorages [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
# odacli list-vmstorages
```

odacli describe-vmstorage

Use the `odacli describe-vmstorage` command to describe VM storage on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli describe-vmstorage -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the name of the VM storage.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example: Describing a VM Storage named `share1`

```
# odacli describe-vmstorage -n share1
```

odacli modify-vmstorage

Use the `odacli modify-vmstorage` command to modify VM storage configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli modify-vmstorage -n name -i increment [-r {MIRROR|HIGH}] [-s decrement] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the VM storage name
`--increment`, `-i`	(Optional) Defines the size to grow the VM storage. Format is K|M|G|T
`--shrink`, `-s`	(Optional) Defines the size to shrink the VM storage. Format is K|M|G|T
`--redundancy`, `-r`	(Optional) Specifies the VM storage redundancy for a DB system. The value can be HIGH or MIRROR. This option is not supported for DB systems on non-FLEX disk group. The options can be MIRROR or HIGH. The default option is MIRROR.
`--json`, `-j`	(Optional) Displays output in JSON format.

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Increase the size of a VM storage named `share1` by 10 gigabytes.

```
# odacli modify-vmstorage -n share1 -i 10G
```

odacli delete-vmstorage

Use the `odacli delete-vmstorage` command to delete a vmstorage configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli delete-vmstorage -n name [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the vmstorage name to be deleted
`--force`, `-f`	(Optional) Specifies the force option
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Delete a VM storage named `share1`.

```
odacli delete-vmstorage -n share1
```

odacli Virtual Disk Commands for Oracle KVM

Use the odacli virtual disk commands to manage virtual disks on Oracle Database Appliance for Oracle KVM.

- odacli create-vdisk
  Use the `odacli create-vdisk` command to create a new virtual disk in a VM storage on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli list-vdisks
  Use the `odacli list-vdisks` command to list all virtual disks on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli describe-vdisk
  Use the `odacli describe-vdisk` command to describe a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-vdisk
  Use the `odacli modify-vdisk` command to modify a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

- odacli clone-vdisk
  Use the `odacli clone-vdisk` command to clone a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli delete-vdisk
  Use the `odacli delete-vdisk` command to delete a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

odacli create-vdisk

Use the `odacli create-vdisk` command to create a new virtual disk in a VM storage on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli create-vdisk -n name -vms vmstorage_name -s size -sh -sp [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Name assigned to the virtual disk that is unique within the name repository
`--vmstorage`, `-vms`	Name of the VM storage where the virtual disk will be created
`--size`, `-s`	Amount of storage to be assigned from the shared repository to the shared disk. The default unit is G (for gigabytes) and the minimum size is 500 M (for megabytes). Format is K|M|G|T
`--shareable`, `-sh`	(Optional) Sets the option of allowing the virtual disk to be shared by more than one virtual machine (`shared`)
`--sparse`, `-sp`	(Optional) Creates a sparse vdisk. Use the default option, non-sparse, to pre-allocate the full vdisk size in the vdisk image. Specify sparse vdisk with the `--sparse, -sp` option when you do not need to pre-allocate the vdisk size in the vdisk image. The image size can increase till it reaches the total current vdisk size.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Create a 2 GB non-sparse and shareable Virtual Disk named `vdisk1` inside the VM storage `vms1`.

```
odacli create-vdisk -n vdisk1 -vms vms1 -s 2G -sh
```

odacli list-vdisks

Use the `odacli list-vdisks` command to list all virtual disks on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli list-vdisks [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
odacli list-vdisks
```

odacli describe-vdisk

Use the `odacli describe-vdisk` command to describe a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli describe-vdisk -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the name of the virtual disk.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example: Describing a Virtual Disk named `vdisk1`

```
odacli describe-vdisk -n vdisk1
```

odacli modify-vdisk

Use the `odacli modify-vdisk` command to modify a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli modify-vdisk -n name -i [-no-s|-s] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual disk name
`--increment`, `-i`	Defines the size to increase the virtual disk. Format is *value units*, units are K|M|G|T
`-no-shareable`, `--no-s`	(Optional) Set the Virtual Disk as non-shareable
`--shareable`, `-s`	(Optional) Set the Virtual Disk as shareable
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Increase the size of a virtual disk named `vdisk1` by 4 gigabytes.

```
# odacli modify-vdisk -n vdisk1 -i 4G
```

odacli clone-vdisk

Use the `odacli clone-vdisk` command to clone a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli clone-vdisk -n name -cn clone_name [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual disk name to be cloned
`--clone-name`, `-cn`	Specifies the name of the cloned virtual disk
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Create a clone of a virtual disk named `vdisk1`, with the name `vdisk1_clone`. The cloned disk is created on the same storage as `vdisk1`.

```
# odacli clone-vdisk -n vdisk1 -cn vdisk1_clone
```

odacli delete-vdisk

Use the `odacli delete-vdisk` command to delete a virtual disk on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli delete-vdisk -n name [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual disk name to be deleted
`--force`, `-f`	(Optional) Specifies the force option
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Delete a virtual disk named `vdisk1`.

```
# odacli delete-vdisk -n vdisk1
```

odacli CPU Pool Commands for Bare Metal and KVM

Use odacli commands to manage resource allocations on Oracle Database Appliance for bare metal and KVM deployments.

CPU pools enable management of CPU resources in both bare metal and KVM environments. There are two types of CPU pools, that is, BM and VM.

Use the BM CPU pool to ensure that the database on a bare metal host runs exclusively on the CPUs specified in the CPU pool.

Use the VM CPU pools to cage and manage CPU resource allocations to workloads for virtual machines. Workloads are isolated by creating CPU pools and assigning or pinning virtual machines to a specific CPU pool. When you pin a virtual machine to a CPU pool, you ensure that the virtual machine uses CPUs in only that CPU pool.

Resources in CPU pools cannot overlap, that is, CPU pools cannot share CPUs.

- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli list-cpupools
  Use the `odacli list-cpupools` command to list all CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli describe-cpupool
  Use the `odacli describe-cpupool` command to describe CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-cpupool
  Use the `odacli modify-cpupool` command to modify a CPU Pool configuration on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli delete-cpupool
  Use the `odacli delete-cpupool` command to delete a CPU Pool configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli list-auditrecords
  Use the `odacli list-auditrecords` command to list the audit records for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli describe-auditrecord
  Use the `odacli describe-auditrecord` command to describe the audit record for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

odacli create-cpupool

Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli create-cpupool -n name -c number_of_cores {-bm|-vm} [-nn node_name] [-s socket]
[-dbs]  [-urc] [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the CPU Pool name.
`--cores, -c`	Specifies the CPU Pool cores.
`--bm, -bm`	(Optional) Specifies the CPU Pool as Bare Metal.
`--vm, -vm`	(Optional) Specifies the CPU Pool as Virtual Machine.
`--dbsystem, -dbs`	(Optional) Specifies the CPU Pool type as DB System.
`--socket, -s`	(Optional) Specifies the socket to use. If not specified, then use all available sockets.
`--use-reserved-cores, -urc`	(Optional) Specifies to use reserved CPU cores.
`--node-name, -nn`	(Optional) Specifies the node where the CPU Pool will be created. On High-Availability deployment, if this option is not specified, then the CPU pool is created on both nodes.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- Attach the BM CPU pool to databases. Bare metal CPU pools can be assigned to multiple databases.
- Attach VM CPU pool VM to guest machines. VM CPU pools can be assigned to multiple VMs.
- Attach dbsystem CPU pool to DB systems. dbsystem CPU pool can be assigned to multiple DB systems. After you create a dbsystem CPU pool, you can create a new DB system with this CPU pool by adding the new CPU pool parameter in the create-dbsystem json file, or modify-dbsystem to attach CPU pool to an existing DB system.
- Resources in CPU pools cannot overlap, that is, CPU pools cannot share CPUs.
- After you create a BM CPU pool, you can associate this CPU pool to a database using the `create-database` or `modify-database` command.
- After you create a VM CPU pool, you can associate this CPU pool to a VM using the `create-vm` or `modify-vm` command.
- Number of cores must be an even number between 2 and the total number of available cores on the system.
- By default, a CPU in the CPU pool is allocated across two sockets in release 19.12 and later. Optionally, you can choose to allocate the CPU based on the socket number (0 or 1).

This is especially useful to support Standard Edition licensing. It is recommended that you use the default allocation and do not mix two types of allocation.

- By default, the first core of the socket is reserved and cannot be used by the VM and DB system CPU pools for optimal performance. Use the `--use-reserved-cores` option to enable use of the first core of socket for VM and DB system CPU pools.

Example

Create a BM CPU pool named `cpupool1` with 6 cores associated to it.

```
# odacli create-cpupool -n cpupool1 -c 6 -bm
```

Create a VM CPU pool named `cpupool2` with 6 cores associated to it.

```
# odacli create-cpupool -n cpupool2 -c 6 -vm
```

Create a DB System CPU pool named `cpupool3` with 6 cores associated to it.

```
# odacli create-cpupool -n cpupool3 -c 6 -dbs
```

Create a DB System CPU pool named `cpupool4` with 4 cores of CPUs allocated from socket 0.

```
# odacli create-cpupool -n cpupool4 -dbs -s 0 -c 4
```

odacli remap-cpupools

Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli remap-cpupools -f -urc [-j] [-h]
```

Parameters

Parameter	Description
`--force`, `-f`	(Optional) Specifies to force the operation.
`--use-reserved-cores`, `-j`	(Optional) Specifies to use reserved CPU cores.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

If the `odacli remap-cpupools` command fails with the error `DCS-10001:Internal error encountered: Not enough physical CPUs available for CPU Pool '`*`cpupool_name`*`' on node '`*`node_name`*`'`, run the `odacli remap-cpupools` command again with the `--use_reserved-cores` option.

Use the `--force` option to remap CPU pools even if the CPU pool is marked as NUMA-enabled. You must use `odacli remap-cpupools` command, after updating the bare metal CPU cores with the command `odacli modify-cpucores` for CPU core number changes.

By default, the first core of the socket is reserved and cannot be used by the VM and DB system CPU pools for optimal performance. Use the `--use-reserved-cores` option to enable use of the first core of socket for VM and DB system CPU pools.

Example

```
# odacli remap-cpupools
```

Related Topics

- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

odacli list-cpupools

Use the `odacli list-cpupools` command to list all CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli list-cpupools [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
# odacli list-cpupools
```

odacli describe-cpupool

Use the `odacli describe-cpupool` command to describe CPU Pools on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli describe-cpupool -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the name of the CPU Pool.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example: Describing a CPU pool named `cpupool1`

```
# odacli describe-cpupool -n cpupool1
```

odacli modify-cpupool

Use the `odacli modify-cpupool` command to modify a CPU Pool configuration on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli modify-cpupool -n name -c cores -f -l [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the CPU Pool name
`--cores, -c`	Specifies the new number of CPU cores
`--force, -f`	(Optional) Forces the operation
`--live, -l`	(Optional) Applies the changes to running VMs
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

For VM CPU pool, if `--live` is not specified, then the changes are applied to the VM configuration file and will be activated next time the VM is started. If `--live` is specified, then the changes will be applied to both VM configuration file and to the running VM.

Example

Set the size of a CPU Pool `cpupool1` to 10. If --live is not specified, the change is applied to the VM configuration file and will be activated next time the VM is started.

```
# odacli modify-cpupool -n cpupool1 -c 10
```

odacli delete-cpupool

Use the `odacli delete-cpupool` command to delete a CPU Pool configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli delete-cpupool -n name -f [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the CPU Pool name to be deleted
`--force, -f`	(Optional) Forces deletion.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

Delete a CPU pool named `cpupool1`.

```
# odacli delete-cpupool -n cpupool1
```

odacli list-auditrecords

Use the `odacli list-auditrecords` command to list the audit records for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli list-auditrecords -du dcs-username -f from_date-range -l latest -nn node_name -
ot operation_type -ou OS_username -rn resource_name -rt resource_type -t to_date-range [-
j] [-h]
```

Parameters

Parameter	Description
`--dcs-user, -du`	(Optional) Specifies the DCS user name.
`--from, -f`	(Optional) Specfies the *from* date in the date range for the CPU pool audit records.
`--latest, -l`	(Optional) Specifies the latest number of CPU pool audit records.
`--node-name, -nn`	(Optional) Specifies the node name.
`--op-type, -ot`	(Optional) Specifies the type of operation.
`--os-user, -ou`	(Optional) Specifies the operating system user.
`--rsc-name, -rn`	(Optional) Specifies the resource name.
`--rsc-type, -rt`	(Optional) Specifies the resource type.
`--to, -t`	(Optional) Specfies the *to* date in the date range for the CPU pool audit records.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

```
# odacli list-auditrecords
ID                                       Resource type         Resource name
Operation type   DCS username     OS username      Timestamp
---------------------------------------  --------------------  ----------------------
---------------  ---------------  ---------------  ------------------------
41295d99-f263-4284-b6d9-2aff1ae0f850  BM_CPU_POOL           testbmpool
```

```
CREATE          oda-admin       root            2020-10-08 13:27:05 UTC
0f07a013-5bb4-4cd8-bd49-04dc61fcd454  BM_CPU_POOL         pool_9f74a8e9-
DELETE          oda-admin       root            2020-10-08 12:26:30 UTC
d858269b-6c28-4bc2-af71-79004f3561e6  BM_CPU_POOL         pool_9f74a8e9-
CREATE          oda-admin       root            2020-10-08 12:24:36 UTC
9d7852c1-93fb-410e-9e0c-8c0a95ed3f0c  VM_CPU_POOL         pool_d48e81e4-
DELETE          oda-admin       root            2020-10-08 12:06:45 UTC
0235cc84-9345-40bc-ac4f-7de3f63202e2  VM_CPU_POOL         pool_4fe0b4f4-
DELETE          oda-admin       root            2020-10-08 12:06:45 UTC
762f7263-7bd9-4b12-b891-48305165bf36  VM_CPU_POOL         pool_4fe0b4f4-
MODIFY          oda-admin       root            2020-10-08 12:06:22 UTC
0bd2a25d-2327-4918-bd47-8c00f7406f92  VM_CPU_POOL         pool_4fe0b4f4-
MODIFY
```

odacli describe-auditrecord

Use the `odacli describe-auditrecord` command to describe the audit record for a CPU Pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

Syntax

```
odacli describe-auditrecord -i [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	Describes the ID of the audit record.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

```
# odacli describe-auditrecord -i 41295d99-f263-4284-b6d9-2aff1ae0f850
Audit Record details
--------------------------------------------------------------------------------
 Event details
--------------------------
                 Record ID:  41295d99-f263-4284-b6d9-2aff1ae0f850
          Record timestamp:  2020-10-08 13:27:05 UTC
              DCS username:  oda-admin
               OS username:  root
                Cluster ID:  test_cluster-c
                   Node ID:  test_cluster

 Resource/Operation
--------------------------
            Resource type:  BM_CPU_POOL
              Resource ID:  b17cb623-bf79-491f-86d8-d8502987e26c
            Resource name:  testbmpool
           Operation type:  CREATE
    Operation description:  Created with '2' cores and allocation
'test_cluster:1-2,17-18'
```

odacli VM Network Commands for KVM

Use the odacli virtual machine network commands to manage network on Oracle Database Appliance for Oracle KVM.

- odacli create-vnetwork
  Use the `odacli create-vnetwork` command to create a new network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli list-vnetworks
  Use the `odacli list-vnetworks` command to list all virtual networks on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli describe-vnetwork
  Use the `odacli describe-vnetwork` command to describe virtual networks on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli start-vnetwork
  Use the `odacli start-vnetwork` command to start a virtual network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli stop-vnetwork
  Use the `odacli stop-vnetwork` command to stop a virtual network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-vnetwork
  Use the `odacli modify-vnetwork` command to modify a network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli delete-vnetwork
  Use the `odacli delete-vnetwork` command to delete a virtual network configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

odacli create-vnetwork

Use the `odacli create-vnetwork` command to create a new network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli create-vnetwork -n name -if interface-name -t type -br bridge -g gateway -ip IP-address -nm netmask -u uniform -vlan VLAN-network ID [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual network name.
`--interface-name`, `-if`	Specifies the the network interface on the host. This can be physical interface or bonded interface.
`--type`, `-t`	Specifies the virtual network type. Values are bridged or bridged-vlan.
`--bridge`, `-br`	(Optional) Specifies the name of the bridge.
`--gateway`, `-gw`	(Optional) Specifies the virtual network gateway
`--ip`, `-ip`	(Optional) Specifies the virtual network addresses. For more than one address, use the format "*node0_ip,node1_ip,...*"
`--netmask`, `-nm`	(Optional) Specifies the virtual network netmask
`--uniform`, `-u`	(Optional) Specifies whether the virtual network is created on both nodes in a high-availability deployment
`--vlan-id`, `-vlan`	(Optional) Specifies the VLAN network ID
`--json`, `-j`	(Optional) Displays output in JSON format.

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

If you did not specify the IP address, or netmask, or gateway during vnetwork creation, then you can add them later using the `odacli modify-vnetwork` command.

Example

Create a VM network of type `bridged` using the interface `btbond2`.

```
# odacli create-vnetwork -n vnet1 -if btbond2 -type bridged
```

odacli list-vnetworks

Use the `odacli list-vnetworks` command to list all virtual networks on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli list-vnetworks [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
# odacli list-vnetworks
```

odacli describe-vnetwork

Use the `odacli describe-vnetwork` command to describe virtual networks on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli describe-vnetwork -n [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the name of the virtual network.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example: Describing a VM Network named `vnet1`

```
# odacli describe-vnetwork -n vnet1
```

odacli start-vnetwork

Use the `odacli start-vnetwork` command to start a virtual network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli start-vnetwork -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the virtual machine name to be stopped
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

Start a virtual network named `vnet1`.

```
# odacli start-vnetwork -n vnet1
```

odacli stop-vnetwork

Use the `odacli stop-vnetwork` command to stop a virtual network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli stop-vnetwork -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the virtual network to be stopped
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Example

Stop a virtual network named `vnet1`.

```
# odacli stop-vnetwork -n vnet1
```

odacli modify-vnetwork

Use the `odacli modify-vnetwork` command to modify a network on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli modify-vnetwork -n name -if -t -br -gw -ip -nm -u -vlan [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual network name.
`--gateway`, `-gw`	(Optional) Specifies the virtual network gateway
`--ip`, `-ip`	(Optional) Specifies the virtual network addresses. For more than one address, use the format "*node0_ip,node1_ip,...*". Specify the value as `none` to remove the IP address.
`--netmask`, `-nm`	(Optional) Specifies the virtual network netmask
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Modify a VM network to specify the gateway.

```
# odacli modify-vnetwork -n vnet1 -g 10.11.44.41
```

odacli delete-vnetwork

Use the `odacli delete-vnetwork` command to delete a virtual network configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli delete-vnetwork -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the vnetwork name to be deleted
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Delete a VM network named `vnet1`.

```
# odacli delete-vnetwork -n vnet1
```

odacli Virtual Machine Commands for Oracle KVM

Use the odacli virtual machine commands to manage virtual machines on Oracle Database Appliance for Oracle KVM.

- odacli create-vm
  Use the `odacli create-vm` command to create a new virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli list-vms
  Use the `odacli list-vms` command to list all virtual machines on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli describe-vm
  Use the `odacli describe-vm` command to describe a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli modify-vm
  Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli start-vm
  Use the `odacli start-vm` command to start a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli stop-vm
  Use the `odacli stop-vm` command to stop a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli migrate-vm
  Use the `odacli migrate-vm` command to migrate a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli clone-vm
  Use the `odacli clone-vm` command to clone a virtual machine configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli delete-vm
  Use the `odacli delete-vm` command to delete a virtual machine configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

odacli create-vm

Use the `odacli create-vm` command to create a new virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
 odacli create-vm -n name -m memory -bo boot_options -src source -vc vcpus -vms
vmstorage -cp cpupool -d description -g graphics
-mm max-mem-size -mvc max-vcpu-count -osv os_variant -pn preferred_node -s disk_size -vd
vdisks
-vn virtual_network_names -ea extra_args -esrc extra_srcs [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the VM name
`--memory, -st`	Specifies the size of the VM memory. The format is "value_unit". Unit is M|G.
`--source, -src`	Specifies the source installation. If local, it can be an ISO or a bootable disk image. If remote, it can be an URI pointing to an install tree hosted remotely over HTTP, HTTPS, or FTP.
`--vcpus, -vc`	Specifies the number of vCPUs to use.
`--vmstorage, -vms`	Specifies the VM storage name.
`--boot-options, -bo`	Specifies the VM boot options.
`--cpupool, -cp`	(Optional) Specifies the CPU pool name
`--extra-srcs, -esrc`	(Optional) Specifies the extra sources for the guest VM that are mounted as CDROM devices
`--description, -d`	(Optional) Specifies the VM description
`--extra-args, -ea`	(Optional) Specifies the extra arguments for the VM installer.
`--graphics, -g`	(Optional) Specifies the graphics settings for the VM
`--max-memory, -mm`	(Optional) Specifies the maximum size of the VM memory. The format is "value_unit". Unit is M|G.
`--max-vcpus, -mvc`	(Optional) Specifies the maximum number of VCPUs
`--os-variant, -osv`	(Optional) Specifies the operating system variant. Use the command `osinfo-query os` to get the list of the accepted OS variants.
`--pref-node, -pn`	(Optional) Specifies the preferred node to run the VM
`--size, -s`	(Optional) Specifies the size of the VM. The format is "value_unit". Unit is K|M|G|T.
`--vdisks, -vd`	(Optional) Specifies the names of the vdisks to use
`--vnetworks, -vn`	(Optional) Specifies the names of the virtual networks to use
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- Use the `--extra-args` parameter to add kickstart configuration file, kernel parameters for example, serial console. For example: `--extra-args "console=tty0 console=ttyS0,115200 ks=http://192.168.122.100:80/ks.txt"`

Example

Create a VM named `vm1` with 8 vcpus and 8 GB memory. The VM is stored in the VM Storage `vms1`. The source used for the installation is located on `/u01/OL7.iso`.

```
# odacli create-vm -n vm1 -vc 8 -m 8G -vms vs1 -src /u01/OL7.iso
```

Example

If the installation source is located through network, the network configuration of the VM must be provided by `--extra-args` so that the network can be setup at the bootstrap of the VM, to reach the installation source over the network. For example, to create a VM from source

```
https://example.com/iso/OracleLinux/OL7/u9/x86_64/OracleLinux-R7-U9-Server-
x86_64-dvd/:

# odacli create-vm -vn pubnet -m 4G -vc 8 -src https://example.com/iso/
OracleLinux/OL7/u9/x86_64/OracleLinux-R7-U9-Server-x86_64-dvd/ -vd vmstorage_vdisk2 -vms
vmstore -g vnc,listen=0.0.0.0,port=5949 -n vm1 -s 20G --extra-args
"dns=<dns_server_ip_addr>
ip=<vm1_ip_addr>::<gateway_ip>:<netmask>:<vm1_hostname>:ens3:none"
```

odacli list-vms

Use the `odacli list-vms` command to list all virtual machines on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli list-vms [-j] [-h]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

```
# odacli list-vms
```

odacli describe-vm

Use the `odacli describe-vm` command to describe a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli describe-vm -n name [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the name of the virtual machine.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example: Describing a Virtual Machine named `vm1`

```
# odacli describe-vm -n vm1
```

odacli modify-vm

Use the `odacli modify-vm` command to modify a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli modify-vm -n name -as -en -bo boot_options -cfg -avd attach_vdisks -avn
attach_vnetworks -dev device -d description -dn -dvd detach-vdisk -dvn detach-vnetworks -
mm max-memory -mvc max-vcpuc -m memory -no-cp -pn preferred-node -vc vcpu_count -di disk-
increment -en -fo -g graphics -l -no-as [-f] [-g] [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the VM name
`--autostart, -as`	(Optional) Specifies if the auto-start option must be set for the VM
`--no-autostart, -no-as`	(Optional) Disables the auto-start option for the VM
`--boot-options, -bo`	(Optional) Specifies the the boot options to use
`--config, -cfg`	(Optional) Specifies whether the changes persist.
`--live, -l`	(Optional) Indicates whether the specified changes take place live (true) or after the next boot (false). Default is false
`--enable-numa, -en`	(Optional) Enables NUMA configuration for the VM.
`--disable-numa, -dn`	(Optional) Disables NUMA configuration for the VM.
`--cpu-pool, -cp`	(Optional) Specifies the CPU pool name
`--description, -d`	(Optional) Specifies the VM description
`--failover, -fo`	(Optional) Specifies the configuration for the failover
`--no-failover, -no-fo`	(Optional) Specifies the configuration for the failover
`--vcpus, -vc`	(Optional) Specifies the number of vCPUs.
`--memory, -m`	(Optional) Specifies the memory size of the VM. The format is "value_unit". Unit is M|G.
`--max-memory, -mm`	(Optional) Specifies the maximum memory size of the VM. The format is "value_unit". Unit is M|G.
`--max-vcpus, -mvc`	(Optional) Specifies the maximum number of vCPUs.
`--no-cpupool, -no-cp`	(Optional) Dissasociates the associated CPU Pool
`--detach-vdisk, -dvd`	(Optional) Specifies the names of the virtual disks to detach
`--detach-vnetworks, -dvn`	(Optional) Specifies the names of the virtual networks to detach
`--pref-node, -pn`	(Optional) Specifies the preferred node to run the VM
`--disk-increment, -di`	Specifies the change in size of the VM. The format is "value_unit". Unit is K|M|G|T.
`--attach-vdisks, -avd`	(Optional) Specifies the names of the vdisks to attach
`--attach-vnetworks, -avn`	(Optional) Specifies the names of the virtual networks to attach

Parameter	Description
`--device`, `-dev`	(Optional) Modifies the device as specified in the expression: Where device_expression = `[vdisk=vdisk_name,][bus=bus_type,][target=target_device]` `vdisk_name`: If specified, must refer to an attached `vDisk` name. Otherwise it refers to the VM image. `bus_type`: The value can be `ide`, `virtio`, `scsci`, `fdc`, `xen`, `usb`, `sata` or `sd`. `target_device`: Must be a device name compatible with the `bus_type`. For example: `vda`, `hda`, `sda`.
`--force`, `-f`	(Optional) Specifies to force the operation.
`--graphics`, `-g`	(Optional) Defines the graphics settings for the VM. The supported graphics setting is VNC. For example, `vnc,listen=127.0.0.1,port=5901`.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Tips

- `--live`: The change is applied to the running VM instance, but does not persist after the VM stops.
- `--enable-numa`: Use this option to enable NUMA for an existing VM.
- `--config`: The change persists in the configuration of VM instance and takes effect on subsequent reboots.
- If neither `--live` or `--config` is specified, then the default behavior is `--config`.
- You can specify both the `--live` or `--config` options if you want the modification to take effect immediately and on subsequent boots.
- When an application VM is created, a default network interface is created by default in the VM. This interface is attached to the default kvm bridge (virbr0) at the host. This interface is intended for convenient communication between the host and VM. This interface does not attach to any physical network interface at the host. Do not configure this interface to use for any external communication.
- You must create a vnetwork using the command `odacli create-vnetwork` and attach the vnetwork (`odacli modify-vm -avn`) to the VM. This creates a second interface in the VM. Configure this network interface for your network requirement.
- The network interface name in the VM depends on the operating system.

Modifying CPU Count and Memory for a VM

Update the configuration of a VM named `vm1`, setting the CPU count to 6 and the maximum memory to 6 gigabytes on both the running VM and subsequent boots.

```
# odacli modify-vm -n vm1 -vc 6 -mm 6G --live --config
```

Enabling NUMA

Use `--enable-numa` to enable NUMA for the VM configuration. The command `odacli modify-dbsystem --enable-numa` is only supported on application VMs with even number of CPU cores so that CPU cores can be evenly distributed to NUMA nodes. In Oracle Database

Appliance, one CPUcore is equal to 2 vCPU. For more information, see *About Enabling NUMA on Oracle Database Appliance*.

```
# odacli modify-vm --name name --enable-numa
```

Modifying a VM to attach a vnetwork

Modify a VM to attach a vnetwork:

```
# odacli modify-vm -n vm_name -avn vnetwork_name
```

Modifying a VM to change the bus type for the VM image

Modify the bus type for the VM image from virtio to sata. This is useful when the VM image does not support the virtio driver, for example, a Windows VM without virtio driver installed:

```
# odacli modify-vm -dev "bus=sata" -n winvm
```

Modifying a VM to change the bus type for a specific vdisk attached to the VM

To modify the bus type for a specific vdisk attached to the VM, run the following command:

```
# odacli modify-vm -dev "vdisk=vdisk_name, bus=sata" -n winvm
```

The command changes the `winvm` bus type of *`vdisk_name`* to sata. If you do not specify the vdisk parameter, the command changes the bus type to the vdisk that holds the VM image.

Related Topics

- odacli remap-cpupools
  Use the `odacli remap-cpupools` command to remap CPU pools to NUMA allocation on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.
- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a DB system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- odacli create-cpupool
  Use the `odacli create-cpupool` command to create a new CPU pool on Oracle Database Appliance Bare Metal or Kernel-based Virtual Machine (KVM) deployment.

odacli start-vm

Use the `odacli start-vm` command to start a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli start-vm -n name -nn [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual machine name to be stopped

Parameter	Description
`--node-name`, `-nn`	(Optional) Specifies the node name where the virtual machine will be started
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

- If `pref-node` is defined for the VM, then the VM starts on the `pref-node`. If `pref-node` is not defined for the VM, then the VM can start on any node, in a high-availability deployment.
- If you specify the node name `-n`, then the VM starts on the specified node, even if the preferred node is defined.

Example

Start a virtual machine named `vm1`.

```
# odacli start-vm -n vm1
```

odacli stop-vm

Use the `odacli stop-vm` command to stop a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli stop-vm -n name [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual machine name to be stopped
`--force`, `-f`	(Optional) Specifies the force option
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Stop a virtual machine named `vm1`.

```
# odacli stop-vm -n vm1
```

odacli migrate-vm

Use the `odacli migrate-vm` command to migrate a virtual machine on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli migrate-vm -n name -th node_name [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual machine name to be migrated.
`--target-node`, `-th`	Specifies the destination node name.
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Migrate a virtual machine named `vm1` to node named `oda1`.

```
odacli migrate-vm -n vm1 -th oda1
```

odacli clone-vm

Use the `odacli clone-vm` command to clone a virtual machine configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
# odacli clone-vm -n name_of_vm_to_clone_from -cn clone_name [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual machine name to be cloned
`--clone-name`, `-cn`	Specifies the name of the cloned virtual machine
`--description`, `-d`	(Optional) Specifies the description of the virtual machine
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Create a clone of a virtual machine named `vm1`, with the name `vm1_clone`. The cloned VM is created on the same storage as `vm1`.

```
# odacli clone-vm -n vm1 -cn vm1_clone
```

odacli delete-vm

Use the `odacli delete-vm` command to delete a virtual machine configuration on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli delete-vm -n name [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--name`, `-n`	Specifies the virtual machine name to be deleted
`--force`, `-f`	(Optional) Specifies the force option
`--json`, `-j`	(Optional) Displays output in JSON format.
`--help`, `-h`	(Optional) Displays help for using the command.

Example

Delete a virtual machine named `vm1`.

```
odacli delete-vm -n vm1
```

18

Configuring and Using Oracle Auto Service Request (Oracle ASR)

Configure and use Oracle ASR to automatically generate a service request for specific hardware faults.

- About Configuring Oracle ASR
  Configure Oracle Auto Service Request (Oracle ASR) to automatically generate service requests for specific hardware faults.
- Configuring Internal Oracle ASR
  Use ODACLI commands or the Browser User Interface (BUI) to configure Oracle Auto Service Request (Oracle ASR) in Oracle Database Appliance.
- Registering with External Oracle ASR Manager
  Use ODACLI commands or the Browser User Interface to register Oracle Database Appliance with an external Oracle Auto Service Request (Oracle ASR) Manager.
- Configuring External Oracle ASR When Oracle ASR Manager Does Not Run on an Oracle Database Appliance
  Understand how you can configure Oracle Database Appliance with an external Oracle Auto Service Request (Oracle ASR) Manager.
- Testing Oracle ASR
  Use the `odacli test-asr` command to confirm that your Oracle ASR configuration is working.
- Managing an Oracle ASR Configuration
  Use the command-line to review an existing configuration, make changes to a configuration, or delete a configuration.
- Additional Resources for Oracle ASR
  Refer to these sections for more information about Oracle ASR.

About Configuring Oracle ASR

Configure Oracle Auto Service Request (Oracle ASR) to automatically generate service requests for specific hardware faults.

Oracle ASR is a secure support feature that can improve system availability through expedited diagnostics and priority service request handling. You can configure Oracle ASR on Oracle Database Appliance to use its own Oracle ASR Manager (**internal Oracle ASR**) or use Oracle ASR Manager configured on another server in the same network as your appliance (**external Oracle ASR**).

Oracle Database Appliance supports two types of Oracle ASR configurations - Internal and External. You can use the `odacli configure-asr` command to configure both types of Oracle ASR.

- **Internal Oracle ASR:** For this type of configuration, Oracle ASR Manager is installed on the Oracle Database Appliance, where assets need to be monitored. For this type of configuration, Oracle ASR Manager is installed on the Oracle Database Appliance, where

assets need to be monitored. The assets Oracle ILOM and the host are registered with Oracle ASR using an XML payload that is sent to the HTTPS endpoint of Oracle ASR Manager. After successful registration, the alerts are also sent from the assets to the Oracle ASR Manager by composing an XML payload. Once the XML file is sent, an SR is generated based on the alert information present in the XML file.

- **External Oracle ASR:** For this type of configuration, Oracle ASR Manager is not installed on one Oracle Database Appliance. For example, if Oracle ASR manager is not installed on `oda1`, but Oracle ASR Manager is installed on another Oracle Database Appliance, `oda2`, then the assets that belong to `oda2` assets are registered with Oracle ASR Manager installed on `oda1`. To register Oracle Database Appliance assets with an external Oracle ASR Manager, you must first update the repository with Oracle ASR Manager configuration files. After that, configure Oracle ASR as external to register the assets with Oracle ASR Manager.

> **Note:**
>
> With an internal Oracle ASR Manager, an alert is not sent when the server is not running because the Oracle ASR Manager runs on the same server. However, if a critical event occurs on Oracle Database Appliance with an external Oracle ASR Manager, then an alert can still be generated since the Oracle ASR Manager is not running on the same host.

You can configure Oracle ASR as part of deploying the appliance, or you can use the Browser User Interface (BUI) or command-line interface to configure or edit an Oracle ASR after deployment.

Starting with Oracle Database Appliance release 19.21, when you configure Oracle ASR, you specify HTTPS settings and not the SNMP settings. Following are the requirements to configure Oracle ASR:

- Oracle Database Appliance hardware must be associated with a Support Identifier (SI) in My Oracle Support.
- If a proxy server is required for Internet access to Oracle, then you must also provide the name of the proxy server.
- To configure an internal Oracle ASR Manager, you need a My Oracle Support account user name (e-mail address) and password associated with the My Oracle Support account under which the server is registered.
- To register with an external Oracle ASR Manager, use zip file generated by the `odacli export-asrconfig` command when it is run on the appliance where Oracle ASR Manager runs.

Related Topics

- Configuring Internal Oracle ASR
  Use ODACLI commands or the Browser User Interface (BUI) to configure Oracle Auto Service Request (Oracle ASR) in Oracle Database Appliance.
- Configuring External Oracle ASR
  Use ODACLI commands or the Browser User Interface to register Oracle Database Appliance with an external Oracle Auto Service Request (Oracle ASR) Manager.
- odacli configure-asr
  Use the `odacli configure-asr` command to configure Oracle Auto Service Request (Oracle ASR) after deploying the appliance.

Configuring Internal Oracle ASR

Use ODACLI commands or the Browser User Interface (BUI) to configure Oracle Auto Service Request (Oracle ASR) in Oracle Database Appliance.

You can register with an internal Oracle ASR Manager at any time, or modify an existing registration. Before configuring, verify that the Oracle Database Appliance hardware is associated with a Support Identifier (SI) in My Oracle Support.

You need the following information to configure Oracle ASR:

- The e-mail address associated with the My Oracle Support account under which the server is registered
- The password associated with the My Oracle Support account under which the server is registered
- If a proxy server is required for Internet access to Oracle, then you must also provide the name of the proxy server

Configuring Internal ASR Using ODACLI Commands

Use the `odacli configure-asr` command to configure Oracle ASR after deploying the appliance.

```
# odacli configure-asr -u user@example.com -hp 16165

Enter ASR user's password:
Retype ASR user's password:

Job details
----------------------------------------------------------------
                     ID:  1c9f126e-f5d2-4178-9d59-1343314b86ee
            Description:  Configure ASR
                 Status:  Success
                Created:  November 29, 2023 4:42:07 PM UTC
                Message:

Task Name                                Start
Time                                End Time
Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Setting up SSH equivalence               November 29, 2023 4:42:07 PM
UTC         November 29, 2023 4:42:07 PM UTC         Success
Modify JavaExec Path                     November 29, 2023 4:42:08 PM
UTC         November 29, 2023 4:42:08 PM UTC         Success
Install ASR                              November 29, 2023 4:42:08 PM
UTC         November 29, 2023 4:44:53 PM UTC         Success
Registering ASR Manager                  November 29, 2023 4:45:01 PM
UTC         November 29, 2023 4:45:09 PM UTC         Success
Configure ILOM                           November 29, 2023 4:45:09 PM
UTC         November 29, 2023 4:45:13 PM UTC         Success
Creating ASR configuration folders       November 29, 2023 4:45:13 PM
UTC         November 29, 2023 4:45:13 PM UTC         Success
Setup ASR XML templates                  November 29, 2023 4:45:13 PM
```

```
UTC         November 29, 2023 4:45:13 PM UTC         Success
Enable HTTPS for ASR                     November 29, 2023 4:45:13 PM
UTC         November 29, 2023 4:45:36 PM UTC         Success
Importing ASR certificate into DCS       November 29, 2023 4:45:36 PM
UTC         November 29, 2023 4:45:38 PM UTC         Success
truststore

Registering Asset: ODA Host              November 29, 2023 4:45:38 PM
UTC         November 29, 2023 4:45:38 PM UTC         Success
ASR assets activation                    November 29, 2023 4:45:39 PM
UTC         November 29, 2023 4:45:40 PM UTC         Success
Remove AsrConfBackup File                November 29, 2023 4:45:41 PM
UTC         November 29, 2023 4:45:41 PM UTC         Success
```

After configuring Oracle ASR, check the the asset status for both the host and the Oracle ILOM asset as follows:

```
# /opt/asrmanager/bin/asr list_asset
IP_ADDRESS HOST_NAME SERIAL_NUMBER PARENT_SERIAL ASR ASR_STATUS PROTOCOL
SOURCE LAST_HEARTBEAT PRODUCT_NAME
---------- --------- ------------- ------------- --- ---------- --------
------ -------------- ------------
2213XLA00E oda1-ilom 2213XLA00E Y Active HTTP ILOM_XML 2023-11-30
06:36:11.472 ORACLE SERVER X8-2L
10.133.48.239 oda1 2213XLA00E Y Active HTTP ESSW_XML NA ORACLE SERVER X8-2L

Please use My Oracle Support 'http://support.oracle.com' to view the
activation status.
To view the latest available asset status in ASR Manager please execute the
'send_asset_status' command.
```

Configuirng Internal ASR Using the BUI

1. Open a browser and enter the following URL to access BUI:

   `https://ip-address:7093/mgmt/index.html`

2. Login as `oda-admin`.
3. Click the **Appliance** tab.
4. Click the Oracle ASR menu.
5. If Oracle ASR is already configured, then in the Configure ASR page, provide the Oracle ASR configuration values.
6. Select **Internal** in the **ASR Type** field.
7. In the **ASR User Name** field, enter e-mail address associated with the My Oracle Support account under which the server is registered.
8. In the **ASR Password** field, enter the password associated with the My Oracle Support account under which the server is registered.
9. (Optional) Specify the HTTPS port for the Oracle ASR Manager. The default HTTPS port is 16162.
10. (Optional) If you are using an HTTP Proxy for upload to Oracle ASR, complete the following:
    a. **HTTP Proxy used for Upload to ASR**: Select **Yes**.

b. **Proxy User Name**: If you are using a proxy for upload, enter the proxy user name.

c. **Proxy Port**: If you are using a proxy for upload, enter the proxy port.

d. **(Optional) HTTP Proxy Requires Authentication**: If you are using a proxy for upload, select **Yes** if you require authentication. If you do not require authentication, select **No**.

e. **(Optional) Proxy Password**: If you are using a proxy for upload and require authentication, enter the proxy password.

11. Click **Create**.

Registering with External Oracle ASR Manager

Use ODACLI commands or the Browser User Interface to register Oracle Database Appliance with an external Oracle Auto Service Request (Oracle ASR) Manager.

You can register with an external Oracle ASR Manager at any time.

Before configuring, verify that the Oracle Database Appliance hardware is associated with a Support Identifier (SI) in My Oracle Support.

Before configuring external Oracle ASR on an appliance, ensure that Oracle ASR Manager is configured on another appliance.

Configuring External ASR Using ODACLI Commands

Run these steps on the appliance where Oracle ASR Manager is configured:

1. Export the internal Oracle ASR Manager configuration files into a zip file:

```
[root@node1 bin]# odacli export-asrconfig

Job details
----------------------------------------------------------------
                     ID: 9c9e74d4-d3d8-4f77-9394-42c8eb7bfe1e
            Description: Export ASR Config
                 Status: Success
                Created: November 30, 2023 7:02:51 AM UTC
                Message: Run update repository with the zip /opt/
oracle/asr/asrConfiguration/ASRConfiguration.zip in the External ASR host
once the current job is successful

Task Name                                Start Time
End Time                          Status
----------------------------------------
---------------------------------------- --------------------------------
Creating a zip of ASR config files       November 30, 2023 7:02:52 AM UTC
November 30, 2023 7:02:52 AM UTC Success
```

2. Copy the zip file from the Oracle ASR Manager appliance to the system where the external Oracle ASR Manager must be configured. The zip file path is displayed in the job description in step 1.

```
$ scp zipFilePath user@system_name/IP:destination_path_for_ the_zip
```

After copying the Oracle ASR Manager configuration zip file to the external Oracle ASR appliance, run the following steps on the external Oracle ASR appliance:

1. Run the `odacli update-repository` command to extract the zip file on the external Oracle ASR Manager system.

```
[root@node1 bin]# odacli update-repository -f zip_file_path
Job details
----------------------------------------------------------------
                     ID: c659b6ee-9f02-4cb4-aba4-4ed8de883e3a
            Description: Repository Update
                 Status: Success
                Created: November 30, 2023 3:44:28 PM CST
                Message: /opt/oracle/dcs/bin/ASRConfiguration.zip

Task Name Start Time End Time Status
----------------------------------------
----------------------------------------
---------------------------------------- ----------------
Unzip bundle November 30, 2023 3:44:28 PM CST November 30, 2023 3:44:28 PM
CST Success
```

2. Configure Oracle ASR Manager as External.

```
[root@node1 bin]# odacli configure-asr -e External

Enter ASR user's password:
Retype ASR user's password:

[root@node1 bin]# odacli describe-job -i cdd8445f-
aefb-4f43-881a-145eaa1a1181

root@scaoda6f001 ~]# odacli describe-job -i 1099eac8-2879-4341-b391-
d56237696395

Job details
----------------------------------------------------------------
                     ID: 1099eac8-2879-4341-b391-d56237696395
            Description: Configure ASR
                 Status: Success
                Created: November 30, 2023 3:46:50 PM CST
                Message:

Task Name                                Start Time
End Time                          Status
----------------------------------------
---------------------------------------- --------------------------------
Setting up SSH equivalence               November 30, 2023 3:46:52 PM CST
November 30, 2023 3:46:54 PM CST Success
Creating ASR configuration folders       November 30, 2023 3:46:54 PM CST
November 30, 2023 3:46:54 PM CST Success
Setup ASR XML templates                  November 30, 2023 3:46:54 PM CST
November 30, 2023 3:46:54 PM CST Success
Importing ASR certificate into DCS       November 30, 2023 3:46:54 PM CST
November 30, 2023 3:46:56 PM CST Success
truststore
Registering Asset: ODA Host              November 30, 2023 3:46:56 PM CST
November 30, 2023 3:46:57 PM CST Success
ASR assets activation                    November 30, 2023 3:46:57 PM CST
```

```
November 30, 2023 3:46:58 PM CST Success
Configure ILOM                          November 30, 2023 3:46:58 PM CST
November 30, 2023 3:47:07 PM CST Success
```

3. To verify the successful activation of the assets with the external Oracle ASR Manager, run the command to list assets in the Oracle ASR Manager host as follows:

```
[root@node1 bin]# /opt/asrmanager/bin/asr list_asset
IP_ADDRESS   HOST_NAME     SERIAL_NUMBER PARENT_SERIAL ASR ASR_STATUS
PROTOCOL SOURCE LAST_HEARTBEAT PRODUCT_NAME
----------   ---------     ------------- ------------- --- ----------
-------- ------ -------------- ------------
IP_address_1 oda1        1642NM10D1                    Y   Active
HTTP     ESSW_XML    NA         ORACLE SERVER X7-2
IP_address_2 oda2        1921XLA0AN                    Y   Active
HTTP     ILOM_XML    NA         ORACLE SERVER X8-2L
IP_address_3 oda3        1921XLA0AN                    Y   Active
HTTP     ESSW_XML    NA         ORACLE SERVER X8-2L x86/x64 System
IP_address_4 oda4        1921XLA0AW                    Y   Active
HTTP     ILOM_XML    NA         ORACLE SERVER X8-2L
IP_address_5 oda5        1921XLA0AW                    Y   Active
HTTP     ESSW_XML    NA         ORACLE SERVER X8-2L x86/x64 System
```

Use *My Oracle Support* to view the activation status. To view the latest available asset status in Oracle ASR Manager, run the `send_asset_status` command.

Configuring External Oracle ASR Using the BUI

Ensure that the zip file of Oracle ASR configured externally is imported to the repository using the `odacli update-repository` command, and then run the following steps:

1. Open a browser and enter the following URL to access BUI:

   `https://ip-address:7093/mgmt/index.html`

2. Login as `oda-admin`.
3. Click the **Appliance** tab.
4. Click the Oracle ASR menu.
5. Select **External** from the ASR Type menu.
6. Click **Create**.

Configuring External Oracle ASR When Oracle ASR Manager Does Not Run on an Oracle Database Appliance

Understand how you can configure Oracle Database Appliance with an external Oracle Auto Service Request (Oracle ASR) Manager.

Prerequisites:

1. Ensure that the ASR manager is installed with version 23.2.0 or above. Run the `/opt/asrmanager/bin/asr show_version` command to view the Oracle ASR version:

   ```
   /opt/asrmanager/bin/asr show_version
   ```

```
ASR Manager version: 23.2.0

Rules definitions version: 23.2.0.0

Auto Update Statistics
=========================
Last Run Time: 2024-02-08 12:19:27.777
Last Run Status: Oracle Automated Service Manager not yet registered.
Next Run Time: 2024-02-09 12:19:27.777

Skipping ASR connectivity tests as ASR Auto Update isn't enabled

Auto Update Status
=========================
Auto Update functionality is disabled.
Please refer to the My Oracle Support Doc Id: 1503107.1 for instructions
on Auto Update of ASR Manager software.

ASR Manager Services
--------------------
ASR Notification Trap is enabled.
Remote Request feature is enabled.
```

2. Ensure that Oracle ASR is enabled to run with HTTPS by running the following commands:

```
/opt/asrmanager/bin/asr https_config
/opt/asrmanager/bin/asr https_enable
```

Ensure that the required setting is present by running the `/opt/asrmanager/bin/asr show_http_receiver` command as follows:

```
/opt/asrmanager/bin/asr show_http_receiver
    HTTP Receiver configuration:

    HTTP Receiver Status: Enabled
    Host Name: xx.xxx.x.xxx
    SFB forward: true
    HTTP Port: 16161
    HTTPS/SSL Port: 16162
    HTTPS/SSL: Enabled
    Jetty HTTP Compliance: Medium - RFC2616
    Enable Jetty Older Ciphers: true
```

The output of the command must display `HTTPS/SSL` as `Enabled`.

Note that when you run the `/opt/asrmanager/bin/asr https_config` command, and the "What is your first and last name?" question is prompted for the certificate, you must enter the ASR manager IP address and keep the rest of the answers default.

Configuring External ASR Using ODACLI Commands

Run these steps on the appliance where Oracle ASR Manager is configured:

1. Create a directory called `asr` under /`tmp`.

   ```
   mkdir /tmp/asr
   ```

2. Go to the directory created in step 1.

   ```
   cd /tmp/asr
   ```

3. Generate the public key with the name `asrHTTPSCertificate.crt` under the directory created in step 1 for Oracle ASR Manager, using its keystore present at `/var/opt/asrmanager/configuration/asrmKeyStore.jks`. The following command creates a certificate `/tmp/asr/asrHTTPSCertificate.crt` for the keystore:

   ```
   jdkbinpath/keytool -exportcert -alias alias set for the keystore -
   keystore /var/opt/asrmanager/configuration/asrmKeyStore.jks -file /tmp/asr/
   asrHTTPSCertificate.crt -rfc
   Enter keystore password: Enter the certificate password provided in "asr
   https_config" command
   ```

4. Create a new JSON file with the name `asrdetails.json` with the following details under `/tmp/asr` and set the values for the following:

   - `asrManagerIp`: The IP address of the host where Oracle ASR Manager runs.
   - `userName`: The email ID with which Oracle ASR Manager is registered.
   - `httpsPortNumber`: The HTTPS port number set for Oracle ASR Manager in the `asr https_config` command.
   - `httpsCertAlias`: The certificate ALIAS set when prompted by the `asr https_config` command.

   Example of `/tmp/asr/asrdetails.json`:

   ```
   {
     "asrManagerIp" : "xx.xxx.x.xxx",
     "userName" : "test",
     "httpsPortNumber" : 16162,
     "httpsCertAlias" : "oda-asr-client2024-01-31t10:15:52"
   }
   ```

5. Generate a zip file with the files created in step 3 and step 4 `asrHTTPSCertificate.crt` and `asrdetails.json` from the `asr` directory present under `/tmp` directory.

   ```
   cd /tmp
   zip ASRConfiguration.zip asr/asrHTTPSCertificate.crt asr/asrdetails.json
   ```

6. Copy the zip file created in step 5 to the Oracle Database Appliance machine whose assets must be registered with the external Oracle ASR.

   ```
   $ scp /tmp/ASRConfiguration.zip user@machine name/IP:destination path for
   the zip
   ```

After copying the external Oracle ASR Manager configuration zip file to Oracle Database Appliance, run the following steps on the Oracle Database Appliance:

1. Run the `odacli update-repository` command to unzip the zip file created on the external Oracle ASR machine and copied to the appliance.

```
$odacli update-repository -f <zip file path>

odacli describe-job -i c659b6ee-9f02-4cb4-aba4-4ed8de883e3a

Job details
----------------------------------------------------------------
                     ID:  c659b6ee-9f02-4cb4-aba4-4ed8de883e3a
            Description:  Repository Update
                 Status:  Success
                Created:  November 30, 2023 3:44:28 PM CST
                Message:  /opt/oracle/dcs/bin/ASRConfiguration.zip

Task Name                                Start
Time                               End
Time                                 Status
---------------------------------------
---------------------------------------
--------------------------------------- ----------------
Unzip bundle                             November 30, 2023 3:44:28 PM
CST         November 30, 2023 3:44:28 PM CST         Success
```

2. Configure Oracle ASR Manager as External.

```
$odacli configure-asr -e External

Enter ASR user's password: <SSO password>
Retype ASR user's password:

odacli describe-job -i 1099eac8-2879-4341-b391-d56237696395

Job details
----------------------------------------------------------------
                     ID:  1099eac8-2879-4341-b391-d56237696395
            Description:  Configure ASR
                 Status:  Success
                Created:  November 30, 2023 3:46:50 PM CST
                Message:

Task Name                                Start
Time                               End
Time                                 Status
---------------------------------------
---------------------------------------
--------------------------------------- ----------------
Setting up SSH equivalence               November 30, 2023 3:46:52 PM
CST         November 30, 2023 3:46:54 PM CST         Success
Creating ASR configuration folders       November 30, 2023 3:46:54 PM
CST         November 30, 2023 3:46:54 PM CST         Success
Setup ASR XML templates                  November 30, 2023 3:46:54 PM
CST         November 30, 2023 3:46:54 PM CST         Success
Importing ASR certificate into DCS       November 30, 2023 3:46:54 PM
CST         November 30, 2023 3:46:56 PM CST         Success
truststore
```

```
Registering Asset: ODA Host              November 30, 2023 3:46:56 PM
CST         November 30, 2023 3:46:57 PM CST         Success
ASR assets activation                    November 30, 2023 3:46:57 PM
CST         November 30, 2023 3:46:58 PM CST         Success
Configure ILOM                           November 30, 2023 3:46:58 PM
CST         November 30, 2023 3:47:07 PM CST         Success
```

Configuring External ASR Using Browser User Interface (BUI)

After copying the external Oracle ASR Manager configuration zip file to Oracle Database Appliance, follow these steps:

1. Open a browser and enter the following URL to access BUI:

   `https://`*`ip-address`*`:7093/mgmt/index.html`

2. Login as `oda-admin`.
3. Click the **Appliance** tab.
4. Click the Oracle ASR menu, and then click **Export** on the ASR Information page. **Note:** You must run the export operation from the Oracle Database Appliance on which internal Oracle ASR Manager is configured.
5. After the job completes successfully, a zip file is generated in the location `/opt/oracle/asr/asrConfiguration`.
6. Manually copy the zip file from the appliance with the internal Oracle ASR Manager to the system where the external Oracle ASR Manager must be configured.
7. In the Configure ASR page, select **ASR Type** as **External**.
8. Specify the **Zip File Location** copied in step 6.
9. Click **Update Repository**.
10. After the repository is updated with the zip file, click **Next**.
11. Specify and confirm the **ASR Password**.
12. Click **Configure ASR**.
13. After the job completes sccessfully, verify the Oracle ASR configuration on the ASR Information page.

Testing Oracle ASR

Use the `odacli test-asr` command to confirm that your Oracle ASR configuration is working.

When you run the command, a test alert is sent to Oracle ASR manager from both Oracle ILOM and the host registered with Oracle ASR. The registered email ID receives an email about the test alert for the registered Oracle ILOM and host.

Attempting to test the configuration outside of the `odacli test-asr` command might not provide the expected test result.

1. Log in as `root` user.

2. Run the `odacli test-asr` command.

```
# odacli test-asr

Job details
----------------------------------------------------------------
                     ID: 51522c07-1b32-4874-9bc7-bbee5885f1e5
            Description: Test ASR
                 Status: Success
                Created: October 20, 2023 1:02:49 PM UTC
                Message:

Task Name Start Time                          End Time
Status
--------- ------------------------------ ------------------------------
--------
Test ASR October 20, 2023 1:02:50 PM UTC October 20, 2023 1:02:50 PM UTC
Success
```

Managing an Oracle ASR Configuration

Use the command-line to review an existing configuration, make changes to a configuration, or delete a configuration.

You can view and edit the configuration details, but you cannot change the Oracle ASR type. For example, from internal to external. To change the Oracle ASR type, delete the existing configuration and then re-configure Oracle ASR.

All log files for Oracle ASR are located in the `/var/opt/asrmanager/log/` directory.

1. To display Oracle Auto Service Request (Oracle ASR) configuration details, use the `odacli describe-asr` command.

```
# odacli describe-asr

ASR details
----------------------------------------------------------------
                     ID: e0e31043-c312-4371-8301-e62642ef8a4a
                   Name: ASR
               ASR Type: Internal
        ASR Manager IP : 10.32.202.13
   ASR Manager username: scott.tiger@example.com
      HTTPS port number: 16164
                  State: CONFIGURED
                Created: September 3, 2023 4:02:38 AM UTC
                Updated: September 3, 2023 4:03:32 AM UTC
```

2. (Optional) To make changes to Oracle ASR configuration details after deploying the appliance, use the `odacli modify-asr` command.
3. (Optional) To remove the Oracle ASR configuration from the system, use the `odacli delete-asr`command.

```
# odacli delete-asr
{
```

```
"jobId" : "5d70bd17-ec4a-48da-8196-1364105db99d",
"status" : "Running",
"message" : null,
"reports" : [ ],
"createTimestamp" : 1469409622451,
"description" : "Delete ASR",
"updatedTime" : 1469409622458
}
```

Related Topics

- odacli Oracle Auto Service Request Commands
  Use the Oracle Auto Service Request (Oracle ASR) commands to configure, update, test, and delete Oracle ASR on the system.

Additional Resources for Oracle ASR

Refer to these sections for more information about Oracle ASR.

- Oracle ASR product page:

  http://www.oracle.com/asr

- Oracle Database Appliance products qualified for Oracle ASR:
  - Oracle ASR Qualified Oracle Database Appliance Products
  - **IMPORTANT**: Oracle ASR Qualifed Product and Fault Rule Coverage Supplement (Doc ID 2142295.1)
- Oracle ASR user documentation:

  http://docs.oracle.com/cd/E37710_01/index.htm

19

Troubleshooting Oracle Database Appliance

Understand tools you can use to validate changes and troubleshoot Oracle Database Appliance problems.

- About Setting Up and Removing SSH Configuration
  Understand how you can set up and remove SSH configuration.
- Viewing Oracle Database Appliance Error Correlation Reports
  Understand how to view Error Correlation Report and how to interpret the report to troubleshoot your appliance.
- About Enabling Linux Kernel Core Extractor for Troubleshooting
  Understand how to manage Linux Kernel Core Extractor to troubleshoot your appliance.
- Viewing Details About DCS Error Messages
  Understand how to view details about DCS errors for troubleshooting them.
- Collecting Diagnostics Data Using the BUI
  Understand how to collect diagnostics data to troubleshoot errors.
- Resolving Errors When Updating DCS Components During Patching
  Understand how to troubleshoot errors when updating DCS components during patching.
- Viewing Component Information on the Appliance
  View details of all the components installed on the appliance, and the RPM drift information.
- Errors When Logging into the Browser User Interface
  If you have problems logging into the Browser User Interface, then it may be due to your browser or credentials.
- Errors when re-imaging Oracle Database Appliance
  Understand how to troubleshoot errors that occur when re-imaging Oracle Database Appliance.
- Using Oracle Autonomous Health Framework for Running Diagnostics
  Oracle Autonomous Health Framework collects and analyzes diagnostic data collected, and proactively identifies issues before they affect the health of your system.
- Running the Disk Diagnostic Tool
  Use the Disk Diagnostic Tool to help identify the cause of disk problems.
- Running the Oracle Database Appliance Hardware Monitoring Tool
  The Oracle Database Appliance Hardware Monitoring Tool displays the status of different hardware components in Oracle Database Appliance server.
- Disabling the Browser User Interface
  You can also disable the Browser User Interface. Disabling the Browser User Interface means you can only manage your appliance through the command-line interface.
- Preparing Log Files for Oracle Support Services
  If you have a system fault that requires help from Oracle Support Services, then you may need to provide log records to help Oracle support diagnose your issue.

About Setting Up and Removing SSH Configuration

Understand how you can set up and remove SSH configuration.

In earlier Oracle Database Appliance releases, the DCS agent used a scheduled SSH cleaner job that ran every 30 minutes, to clean up SSH connections configured in the system. Starting with Oracle Database Appliance release 19.24, the DCS agent does not remove SSH configurations.

The DCS agent has a new `agentconfig` parameter called `ODA_MANAGED_SSH` that you can use to deactivate the `Setup SSH` job on Oracle Database Appliance. You must set up SSH before you run any job that requires SSH configuration between nodes. If you run a job that requires SSH configuration, then the job uses the pre-configured SSH setup. If the SSH keys are present but not functioning or SSH configuration does not exist, the job displays an error. You must set up a working SSH equivalence between nodes before you rerun the job.

Viewing Oracle Database Appliance Error Correlation Reports

Understand how to view Error Correlation Report and how to interpret the report to troubleshoot your appliance.

About Error Correlation Reports

If a DCS job fails, an Error Correlation job is created automatically to generate an Error Correlation report. You can access and review the generated Error Correlation report from the BUI to explore possible ways of error resolution.

The Error Correlation Report contains the following:

- **Log Messages:** Errors, exceptions and warnings from various log files.
- **Failed Task Messages:** Error message displayed when the DCS job failed.
- **Release Notes:** Relevant Known Issues from *Oracle Database Appliance Release Notes* to help resolve the issue.
- **Documentation:** Relevant topics from the Oracle Database Appliance Documentation Library to help resolve the error.

The Error Correlation Report is generated for every failed DCS job and can be accessed from the BUI. On Oracle Database Appliance high-availability deployments, the Error Correlation report contains the error information derived from log files of both the nodes.

Viewing Error Correlation Reports using ODACLI Commands

You can view the Error Correlation report of a failed DCS job by running the `odacli describe-job -i failed_dcs_job_id --ecr` command. For an example output, see the topic *odacli describe-job* in this guide.

Viewing Error Correlation Reports from the BUI

To view the Error Correlation Report from the Activities page in the BUI:

1. Log into the Browser User Interface:

   `https://host-ip-address:7093/mgmt/index.html`

2. Click the **Activity** tab.

3. In the Activities page, click the **Failure** or **InternalError** link in the failed DCS job for which you want to view the Error Correlation report. Note that only failed DCS jobs have associated Error Correlation Reports.
4. You can also view the Error Correlation Report for the failed DCS job when you click the **Actions** menu, and select **View Error Correlation Report**.
5. The Error Correlation Report contains the following tabs:
   - **Log Messages:** Displays the logs for DCS agent, DCS admin, Oracle HAMI, MySQL, and Oracle FPP. You can expand each section to view the details. Only components that have logs are displayed. If no errors are found, then the message `No errors or exceptions found in logs` is displayed in the Log Messages section.
   - **Failed Task Messages:** Displays the specific error message displayed when the task failed.
   - **Release Notes:** Displays relevant Known Issues from *Oracle Database Appliance Release Notes* to help resolve the issue. You can click each of these links to view the Release Notes entry. If no relevant Known issues are found, then the message `No matching results were found.` is displayed.
   - **Documentation:** Displays relevant topics from the Oracle Database Appliance Documentation Library to help resolve the error. You can click each of these links to view the documentation topic from the Oracle Database Appliance documentation.

To view the Error Correlation Report from the Diagnostics page in the BUI:

1. In the BUI, click the **Diagnostics** tab.
2. In the Diagnostics page, click **Collect Diagnostic Data** for a failed job.
3. The Collect Diagnostics page displays the Error Correlation Report and Job Details in separate tabs for the failed DCS job. Click the **Report File Name** link to download the Error Correlation Report to your local system.
4. The Job details tab displays the steps in the job and the Error Correlation Report contains the **Log Messages**, **Failed Task Messages**, **Release Notes**, and **Documentation** tabs.

Related Topics

- odacli describe-job
  Use the `odacli describe-job` command to display details about a specific job, including the job ID, status, tasks, and the job created date and time stamp.

About Enabling Linux Kernel Core Extractor for Troubleshooting

Understand how to manage Linux Kernel Core Extractor to troubleshoot your appliance.

About Linux Kernel Core Extractor

A Linux kernel panic can occur due to various reasons such as faulty hardware, driver crashes, or software bugs. To identify the cause of kernel panic, it is essential to collect and analyze the vmcore of the crashed kernel. The kdump service is used to collect the vmcore after the first kernel crash. This process is slow for systems with high memory and often fails to generate vmcore when the available space is not sufficient. When Linux Kernel Core Extractor is enabled on Oracle Database Appliance bare metal systems, the crash utility in the kdump kernel collects useful information for troubleshooting without generating vmcore.

Linux Kernel Core Extractor Commands

List generated crash reports:

```
# /usr/sbin/oled lkce list
Followings are the crash*out found in /var/oled/lkce dir:
/var/oled/lkce/crash_20220307-154542.out
```

Purge existing all but last three crash reports:

```
# /usr/sbin/oled lkce clean
lkce deletes all but last three /var/oled/lkce/crash*out files. do you want
to proceed(yes/no)? [no]:
```

Purge all crash reports:

```
# /usr/sbin/oled lkce clean --all
lkce removes all the files in /var/oled/lkce dir. do you want to proceed(yes/
no)? [no]:
```

By default, the crash report contains output for the following crash commands. You can add other crash commands to the `/etc/oled/lkce/crash_cmds_file`.

```
#
# This is the input file for crash utility. You can edit this manually
# Add your own list of crash commands one per line.
#
bt
bt -a
bt -FF
dev
kmem -s
foreach bt
log
mod
mount
net
ps -m
ps -S
runq
quit
```

By default, vmcore generation is disabled. You can enable vmcore generation as follows:

```
# oled lkce configure --vmcore=yes
Restarting kdump service... done!
lkce: set vmcore to yes
```

For additional Linux Kernel Core Extractor commands, refer to the Linux Kernel Core Extractor help.

```
# oled lkce help
Usage: lkce options
options:
    report report-options -- Generate a report from vmcore
    report-options:
        --vmcore=/path/to/vmcore        - path to vmcore
        [--vmlinux=/path/to/vmlinux]        - path to vmlinux
        [--crash_cmds=cmd1,cmd2,cmd3,..]    - crash commands to include
        [--outfile=/path/to/outfile]        - write output to a file

    configure [--default]   -- configure lkce with default values
    configure [--show]  -- show lkce configuration -- default
    configure [config-options]
    config-options:
        [--vmlinux_path=/path/to/vmlinux]   - set vmlinux_path
        [--crash_cmds_file=/path/to/file]   - set crash_cmds_file
        [--kdump_report=yes/no]         - set crash report in kdump kernel
        [--vmcore=yes/no]           - set vmcore generation in kdump kernel
        [--max_out_files=<number>]        - set max_out_files

    enable  -- enable lkce in kdump kernel
    disable -- disable lkce in kdump kernel
    status  -- status of lkce

    clean [--all]   -- clear crash report files
    list        -- list crash report files
```

Viewing Details About DCS Error Messages

Understand how to view details about DCS errors for troubleshooting them.

About Viewing Information About DCS Errors

To view more details about any errors during DCS operations, use the command `dcserr` *error_code*.

```
# /opt/oracle/dcs/bin/dcserr
dcserr error_code

# dcserr 10001
10001, Internal_Error, "Internal error encountered: {0}."
// *Cause: An internal error occurred.
// *Action: Contact Oracle Support Services for assistance.
/
# dcserr 1001
Unknown error code
```

To view more details about DCS errors in the Browser User Interface (BUI), you can provide the DCS error code in the Search box in the BUI. The Search results display the Cause and Action of DCS error codes.

Collecting Diagnostics Data Using the BUI

Understand how to collect diagnostics data to troubleshoot errors.

About Collecting Diagnostics Data

Use the **Diagnostics** tab in the Browser User Interface to view diagnostic information about your deployment and the installed components.

In the Diagnostic Collection page, you can view the available diagnostics collections. Click **Collect Diagnostic Data** to start diagnostics collection. Once the data is collected, click on the collection file path to download the file.

In the Collect Diagnostics page, specify the Job ID for the diagnostics data collection. Optionally, specify a tag and a description for the collection. The details of the Job ID are displayed. Click **Collect** to start the diagnostics data collection.

You can also collect diagnostics from the Activity page, by selecting **Collect Diagnostics** from the **Actions** drop down for a specific job. Click **Collect** to start the diagnostics data collection.

To delete a diagnostic collection, from the Diagnostic Collection page, select the specific collection, and click **Delete**.

This diagnostic collection feature does not replace the `odaadmcli manage diagcollect` command. You can use the `odaadmcli manage diagcollect` command also to enable diagnostics collections, independently of this new feature from the BUI. The `odaadmcli manage diagcollect` command and the diagnostics collection from BUI use the `tfactl` command internally. The diagnostics collection from BUI is aimed to collect other data from DCS metadata that is not collected through `tfactl` and provide greater context for root cause analysis of related DCS jobs failures.

Resolving Errors When Updating DCS Components During Patching

Understand how to troubleshoot errors when updating DCS components during patching.

.

About DCS Components

When you run the `odacli update-dcscomponents` command during patching, pre-checks for MySQL installation are automatically verified before update of Oracle HAMI, MySQL, and DCS components. If any of the pre-checks fail, then the command errors out with a reference to the pre-check report log file location `/opt/oracle/dcs/log/jobfiles/`*`jobId`*`/dcscomponentsPreCheckReport.log`. Review the pre-check report and take corrective actions and then rerun the `odacli update-dcscomponents` command. If there are no pre-check errors, then the patching process proceeds with updating Oracle HAMI, MySQL, and DCS components such as the DCS Agent, DCS CLI, and DCS Controller.

> **Note:**
>
> Run the `odacli update-dcsadmin` command prior to running the `odacli update-dcscomponents` command.

When the `odacli update-dcscomponents` command completes successfully:

The command output is as follows:

```
# ./odacli update-dcscomponents -v 19.25.0.0.0
{
  "jobId" : "3ac3667a-fa22-40b6-a832-504a56aa3fdc",
  "status" : "Success",
  "message" : "Update-dcscomponents is successful on all the node(s):DCS-Agent
shutdown is successful. MySQL upgrade is done before. Metadata migration is
successful. Agent rpm upgrade is successful. DCS-CLI rpm upgrade is
successful.
DCS-Controller rpm upgrade is succ",
  "reports" : null,
  "createTimestamp" : "September 8, 2024 02:37:37 AM CST",
  "description" : "Update-dcscomponents job completed and is not part of Agent
job list",
  "updatedTime" : "September 8, 2024 02:39:10 AM CST"
}
```

The pre-check report log file at the location `/opt/oracle/dcs/log/jobfiles/jobId/dcscomponentsPreCheckReport.log` contains the following:

```
Pre-check Name: Space check
Status: Success
Comments: Required space 3 GB is available in /opt

Pre-check Name: Port check
Status: Success
Comments: Port 3306 is available for running ODA MySQL

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Success
Comments: ODA MySQL rpm dry-run passed

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Success
Comments: ODA MySQL connector/J library found

Pre-check Name: Check for the existence of Metadata migration utility
Status: Success
Comments: Metadata migration utility found

dcs-admin version:
Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.25.0.0.0

Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist
```

```
Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist

Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Success
Comments: Scheduler cron expressions for existing job schedules are up to date
```

When the `odacli update-dcscomponents` command fails:

On Oracle Database Appliance single-node systems, the command output is as follows:

```
# ./odacli update-dcscomponents -v 19.25.0.0.0

DCS-10008:Failed to update DCScomponents: 19.24.0.0.0
Internal error while patching the DCS components :
DCS-10231:Cannot proceed. Pre-checks for update-dcscomponents failed. Refer to
/opt/oracle/dcs/log/jobfiles/jobId/dcscomponentsPreCheckReport.log
on node 0 for details.
```

On Oracle Database Appliance high-availability systems, the command output is as follows:

```
# ./odacli update-dcscomponents -v 19.25.0.0.0

Internal error while patching the DCS components :
DCS-10231:Cannot proceed. Pre-checks for update-dcscomponents failed. Refer to
/opt/oracle/dcs/log/jobfiles/jobId/dcscomponentsPreCheckReport.log
on node 0 and /opt/oracle/dcs/log/jobfiles/jobId/
dcscomponentsPreCheckReport.log
on node 1 for details.
```

The command runs all pre-checks one by one, and errors out at the end if any of the pre-checks is marked as Failed. When a pre-check fails, the error message is displayed on to the console along with the reference to pre-check report log location. The pre-check report log file is at the location `/opt/oracle/dcs/log/jobfiles/jobId/dcscomponentsPreCheckReport.log`.

```
Pre-check Name: Space check
Status: Failed
Comments: Available space in /opt is 2 GB but minimum required space in /opt
is 3 GB

Pre-check Name: Port check
Status: Success
Comments: Port 3306 is available for running ODA MySQL

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Success
Comments: ODA MySQL rpm dry-run passed

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Success
```

```
Comments: ODA MySQL connector/J library found

Pre-check Name: Check for the existence of Metadata migration utility
Status: Success
Comments: Metadata migration utility found

Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.20.0.0.0

Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist

Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist

Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Success
Comments: Scheduler cron expressions for existing job schedules are up to date
```

When the `odacli update-dcscomponents` command fails due to space check error:

The pre-check report log contains the following:

```
Pre-check Name: Space check
Status: Failed
Comments: Available space in /opt is 2 GB but minimum required space in /opt
is 3 GB

Pre-check Name: Port check
Status: Success
Comments: Port 3306 is available for running ODA MySQL

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Success
Comments: ODA MySQL rpm dry-run passed

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Success
Comments: ODA MySQL connector/J library found

Pre-check Name: Check for the existence of Metadata migration utility
Status: Success
Comments: Metadata migration utility found

Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.20.0.0.0
```

```
Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist

Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist

Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Success
Comments: Scheduler cron expressions for existing job schedules are up to date
```

When the `odacli update-dcscomponents` command fails due to port check error:

The pre-check report log contains the following:

```
Pre-check Name: Space check
Status: Success
Comments: Required space 3 GB is available in /opt

Pre-check Name: Port check
Status: Failed
Comments: No port found in the range ( 3306 to 65535 )

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Success
Comments: ODA MySQL rpm dry-run passed

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Success
Comments: ODA MySQL connector/J library found

Pre-check Name: Check for the existence of Metadata migration utility
Status: Success
Comments: Metadata migration utility found

Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.20.0.0.0

Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist

Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist
```

```
Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Success
Comments: Scheduler cron expressions for existing job schedules are up to date
```

When the `odacli update-dcscomponents` command fails due to MySQL RPM installation dry-run check error:

The pre-check report log contains the following:

```
Pre-check Name: Space check
Status: Success
Comments: Required space 3 GB is available in /opt

Pre-check Name: Port check
Status: Success
Comments: Port 3306 is available for running ODA MySQL

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Failed
Comments: ODA MySQL rpm dry-run failed. Failed due to the following error :
Exception details are displayed below

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Success
Comments: ODA MySQL connector/J library found

Pre-check Name: Check for the existence of Metadata migration utility
Status: Success
Comments: Metadata migration utility found

Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.20.0.0.0

Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist

Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist

Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Success
Comments: Scheduler cron expressions for existing job schedules are up to date
```

When the `odacli update-dcscomponents` command fails due to MySQL connector/J library check error:

The pre-check report log contains the following:

```
Pre-check Name: Space check
Status: Success
Comments: Required space 3 GB is available in /opt

Pre-check Name: Port check
Status: Success
Comments: Port 3306 is available for running ODA MySQL

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Success
Comments: ODA MySQL rpm dry-run passed

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Failed
Comments: MySQL connector/J library does not exist. Ensure update-repository
with latest serverzip bundles ran first without any issues prior to running
update-dcscomponents

Pre-check Name: Check for the existence of Metadata migration utility
Status: Success
Comments: Metadata migration utility found

dcs-admin version:
Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.20.0.0.0

Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist

Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist

Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Success
Comments: Scheduler cron expressions for existing job schedules are up to date
```

When the `odacli update-dcscomponents` command fails due to Metadata migration utility check error:

The pre-check report log contains the following:

```
Pre-check Name: Space check
Status: Success
Comments: Required space 3 GB is available in /opt

Pre-check Name: Port check
Status: Success
Comments: Port 3306 is available for running ODA MySQL

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Success
Comments: ODA MySQL rpm dry-run passed

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Success
Comments: ODA MySQL connector/J library found

Pre-check Name: Check for the existence of Metadata migration utility
Status: Failed
Comments: Metadata migration utility does not exist. Ensure update-repository
with latest serverzip bundles ran first without any issues prior to running
update-dcscomponents.

dcs-admin version:
Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.20.0.0.0

Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist

Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist

Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Success
Comments: Scheduler cron expressions for existing job schedules are up to date
```

When the `odacli update-dcscomponents` command displays a warning due to scheduler cron expression:

When patching from Oracle Database Appliance release 19.19 or earlier to the latest release, and you run the `odacli update-dcscomponents` command, there may be a warning in the

precheck report log file if the default cron expressions from the existing list of job schedules are modified. The pre-check report log contains the following:

```
Pre-check Name: Space check
Status: Success
Comments: Required space 3 GB is available in /opt

Pre-check Name: Port check
Status: Success
Comments: Port 3306 is available for running ODA MySQL

Pre-check Name: ODA MySQL rpm installation dry-run check
Status: Success
Comments: ODA MySQL rpm dry-run passed

Pre-check Name: Check for the existence of MySQL connector/J library
Status: Success
Comments: ODA MySQL connector/J library found

Pre-check Name: Check for the existence of Metadata migration utility
Status: Success
Comments: Metadata migration utility found

dcs-admin version:
Pre-check Name: dcs-admin version validation
Status: Success
Comments: dcs-admin is already updated :19.20.0.0.0

Config File Exist dcscontroller:
Pre-check Name: Check DCS config files exists for dcscontroller
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-controller.yml and /opt/oracle/dcs/
conf/dcs-controller-logback.xml exist

Config File Exist dcsagent:
Pre-check Name: Check DCS config files exists for dcsagent
Status: Success
Comments: Files /opt/oracle/dcs/conf/dcs-agent.yml and /opt/oracle/dcs/conf/
dcs-agent-logback.xml exist

Validate scheduler cron expressions:
Pre-check Name: Validate scheduler cron expressions
Status: Warning
Comments: Following cron expressions in the scheduler were modified from
their default values. Starting 19.20, DCS Agent converts 7 fields cron
expression into 6 fields cron expression. No further action needed.
Schedule ID : 3f671ee7-1a03-43fd-b98b-ce33eb09de08 , Custom cron expression :
10
25 * 1/1 * ? 2023
```

Note that the `Status: Warning` means the `update-dcscomponents` pre-check has detected custom cron expressions from the existing list of job schedules. The DCS agent automatically converts from the 7 fields custom cron expression to the equivalent 6 fields cron expression after patching DCS components.

Related Topics

- Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line
  Follow these steps to apply patches to your Oracle Database Appliance bare metal deployment and existing Oracle Database homes, using CLI commands.

Viewing Component Information on the Appliance

View details of all the components installed on the appliance, and the RPM drift information.

Collecting and Viewing the Bill of Materials in the Browser User Interface

Use the Appliance tab in the Browser User Interface to collect and view information about your deployment and the installed components. The Advanced Information tab displays information about the following components:

- Grid Infrastructure Version, and the home directory
- Database Version, Home location, and Edition
- Location and details about the databases configured
- All patches applied to the appliance
- Firmware Controller and Disks
- ILOM information
- BIOS version
- List of RPMs

In the List of RPMs section, click **Show** and then click **RPM Drift** to view the differences between the RPMs installed on the appliance, and the RPMs shipped in the latest Oracle Database Appliance Patch Bundle Update release.

Click **Collect Bill of Materials** to initiate a collection and submit the job. The job ID is displayed. After the collection is complete, click **Refresh** to refresh the information.

Click **Download** to save the components report. You can use this report to help diagnose any deployment issues.

Viewing the Bill of Materials from the Command Line

The bill of materials is also available through the command line for bare metal and virtualized platforms deployments. The information about the installed components is collected according to a set schedule, and stored in the location `/opt/oracle/dcs/Inventory/` for bare metal deployments and in the `/opt/oracle/oak/Inventory/` directory for virtualized platforms. The file is stored in the format `oda_bom_TimeStamp.json`. Use the command `describe-system` to view the bill of materials on the command line. See the *Oracle Database Command-Line Interface* chapter for command options and usage notes.

Example 19-1 Example Command to View the Bill of Materials from the Command Line for Bare Metal Deployments

```
# odacli describe-system -b
ODA Components Information
------------------------------
Component Name                Component
Details
---------------
--------------------------------------------------------------------------------
```

```
-----------------
NODE                            Name : oda1
                                Domain Name : testdomain.com
                                Time Stamp : April 21, 2020 6:21:15 AM UTC


RPMS                            Installed RPMS : abrt-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-addon-
ccpp-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-addon-
kerneloops-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-addon-
pstoreoops-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-addon-
python-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-addon-
vmcore-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-addon-
xorg-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-
cli-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-console-
notification-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-
dbus-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-
libs-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-
python-2.1.11-55.0.1.el7.x86_64,
                                                 abrt-
tui-2.1.11-55.0.1.el7.x86_64,
                                                 acl-2.2.51-14.el7.x86_64,
                                                 adwaita-cursor-
theme-3.28.0-1.el7.noarch,
                                                 adwaita-icon-
theme-3.28.0-1.el7.noarch,
                                                 aic94xx-
firmware-30-6.el7.noarch,
                                                 aide-0.15.1-13.0.1.el7.x86_64,
                                                 alsa-
firmware-1.0.28-2.el7.noarch,
                                                 alsa-lib-1.1.8-1.el7.x86_64,
                                                 alsa-tools-
firmware-1.1.0-1.el7.x86_64,
                                                 at-3.1.13-24.el7.x86_64,
                                                 at-spi2-
atk-2.26.2-1.el7.x86_64,
                                                 at-spi2-
core-2.28.0-1.el7.x86_64,
                                                 atk-2.28.1-1.el7.x86_64,
                                                 attr-2.4.46-13.el7.x86_64,
                                                 audit-2.8.5-4.el7.x86_64,
                                                 audit-libs-2.8.5-4.el7.x86_64,
                                                 audit-libs-
python-2.8.5-4.el7.x86_64,
                                                 augeas-libs-1.4.0-9.el7.x86_64,
```

```
                                                    authconfig-6.2.8-30.el7.x86_64,
                                                    autogen-
libopts-5.18-5.el7.x86_64,
                                                    avahi-
libs-0.6.31-19.el7.x86_64,

basesystem-10.0-7.0.1.el7.noarch,
                                                    bash-4.2.46-33.el7.x86_64,
                                                    bash-
completion-2.1-6.el7.noarch,
                                                    bc-1.06.95-13.el7.x86_64,
                                                    bind-export-
libs-9.11.4-9.P2.el7.x86_64,
                                                    bind-
libs-9.11.4-9.P2.el7.x86_64,
                                                    bind-libs-
lite-9.11.4-9.P2.el7.x86_64,
                                                    bind-
license-9.11.4-9.P2.el7.noarch,
                                                    bind-
utils-9.11.4-9.P2.el7.x86_64,

binutils-2.27-41.base.0.7.el7_7.2.x86_64,
                                                    biosdevname-0.7.3-2.el7.x86_64,
                                                    blktrace-1.0.5-9.el7.x86_64,
                                                    bnxtnvm-1.40.10-1.x86_64,
                                                    boost-date-
time-1.53.0-27.el7.x86_64,
                                                    boost-
filesystem-1.53.0-27.el7.x86_64,
                                                    boost-
iostreams-1.53.0-27.el7.x86_64,
....
....
....
```

Example 19-2 Example Command to View the Bill of Materials from the Command Line for Virtualized Platforms

```
# oakcli describe-system -b
```

Example 19-3 Example Command to View the Bill of Materials Report from the Stored Location

```
# ls -la /opt/oracle/dcs/Inventory/
total 264
-rw-r--r-- 1 root root 83550 Apr 26 05:41 oda_bom_2018-04-26_05-41-36.json
```

Errors When Logging into the Browser User Interface

If you have problems logging into the Browser User Interface, then it may be due to your browser or credentials.

> **Note:**
>
> Oracle Database Appliance uses self-signed certificates. Your browser determines how you log into the Browser User Interface. Depending on the browser and browser version, you may receive a warning or error that the certificate is invalid or not trusted because it is self-signed, or that the connection is not private. Ensure that you accept the self-signed certificate for the agent and Browser User Interface.

Follow these steps to log into the Browser User Interface:

1. Open a browser window.
2. Go to the following URL: `https://ODA-host-ip-address:7093/mgmt/index.html`
3. Get the security certificate (or certificate), confirm the security exception, and add an exception.
4. Log in with your Oracle Database Appliance credentials.

   If you have not already set the `oda-admin` password, then a message is displayed, advising you to change the default password to comply with your system security requirements.
5. If you have not added an exception for the agent security certificate, then a message about accepting agent certificate is displayed.
6. Using a different tab in your browser, go to the following URL: `https://ODA-host-ip-address:7070/login`
7. Get the security certificate (or certificate), confirm the security exception, and add an exception.
8. Refresh the Browser User Interface URL : `https://ODA-host-ip-address:7093/mgmt/index.html`

> **Note:**
>
> If you have any issues logging into the Oracle Database Appliance Browser User Interface on browsers such as macOS Catalina and Google Chrome, then you may need to use any workaround as described on the official site for the product.

Related Topics

- Creating the Appliance
  Create the appliance using the Browser User Interface or ODACLI commands.
- http://www.oracle.com/technetwork/indexes/products/browser-policy-2859268.html

Errors when re-imaging Oracle Database Appliance

Understand how to troubleshoot errors that occur when re-imaging Oracle Database Appliance.

If re-imaging Oracle Database Appliance fails, with old header issues such as errors in storage discovery, or in running GI root scripts, or disk group RECO creation, then use the force mode with `cleanup.pl`.

```
# cleanup.pl -f
```

To ensure that re-imaging is successful, remove the old headers from the storage disks by running the secure erase tool. Verify that the OAK/ASM headers are removed.

```
# cleanup.pl -erasedata
# cleanup.pl -checkHeader
```

Retry the re-imaging operation.

Related Topics

- Re-imaging Oracle Database Appliance
  Bare metal operating system re-imaging installs Oracle Database Appliance operating system software on the local (boot) drive.

Using Oracle Autonomous Health Framework for Running Diagnostics

Oracle Autonomous Health Framework collects and analyzes diagnostic data collected, and proactively identifies issues before they affect the health of your system.

- About Installing Oracle Autonomous Health Framework
  Oracle Autonomous Health Framework is installed automatically when you provision or patch to Oracle Database Appliance release 19.25.
- Using the Oracle ORAchk Health Check Tool
  Run Oracle ORAchk to audit configuration settings and check system health.
- Generating and Viewing Oracle ORAchk Health Check Tool Reports in the Browser User Interface
  Generate Oracle ORAchk Health Check Tool reports using the Browser User Interface.
- Generating and Viewing Database Security Assessment Reports in the Browser User Interface
  Generate and view Database Security Assessment Reports using the Browser User Interface.
- Running Oracle Trace File Analyzer (TFA) Collector Commands
  Understand the installed location of `tfactl` and the options for the command.
- Sanitizing Sensitive Information in Diagnostic Collections
  Oracle Autonomous Health Framework uses Adaptive Classification and Redaction (ACR) to sanitize sensitive data.

- Enabling Adaptive Classification and Redaction (ACR)
  Oracle Database Appliance supports Adaptive Classification and Redaction (ACR) to sanitize sensitive data.
- Sanitizing Sensitive Information in Oracle Trace File Analyzer Collections
  You can redact (sanitize or mask) Oracle Trace File Analyzer diagnostic collections.
- Redacting and Sanitizing Entities in the BUI
  Enable and disable trace file redaction, redact files, and show or hide sanitized entities using the Browser User Interface.
- Sanitizing Sensitive Information in Oracle ORAchk Output
  You can sanitize Oracle ORAchk output.

About Installing Oracle Autonomous Health Framework

Oracle Autonomous Health Framework is installed automatically when you provision or patch to Oracle Database Appliance release 19.25.

When you provision or patch your appliance to Oracle Database Appliance release 19.25, Oracle Autonomous Health Framework is installed in the path `/opt/oracle/dcs/oracle.ahf`.

You can verify that Oracle Autonomous Health Framework is installed by running the following command:

```
[root@oak ~]# rpm -q oracle-ahf
oracle-ahf-193000-########.x86_64
```

> **Note:**
>
> When you provision or patch to Oracle Database Appliance release 19.25, Oracle Autonomous Health Framework automatically provides Oracle ORAchk Health Check Tool and Oracle Trace File Analyzer Collector.

Oracle ORAchk Health Check Tool performs proactive health checks for the Oracle software stack and scans for known problems. Oracle ORAchk Health Check Tool audits important configuration settings for Oracle RAC deployments in the following categories:

- Operating system kernel parameters and packages
- Oracle Database Database parameters, and other database configuration settings
- Oracle Grid Infrastructure, which includes Oracle Clusterware and Oracle Automatic Storage Management

Oracle ORAchk is aware of the entire system. It checks the configuration to indicate if best practices are being followed.

Oracle Trace File Collector provides the following key benefits and options:

- Encapsulation of diagnostic data collection for all Oracle Grid Infrastructure and Oracle RAC components on all cluster nodes into a single command, which you run from a single node
- Option to "trim" diagnostic files during data collection to reduce data upload size
- Options to isolate diagnostic data collection to a given time period, and to a particular product component, such as Oracle ASM, Oracle Database, or Oracle Clusterware

- Centralization of collected diagnostic output to a single node in Oracle Database Appliance, if desired
- On-Demand Scans of all log and trace files for conditions indicating a problem
- Real-Time Scan Alert Logs for conditions indicating a problem (for example, Database Alert Logs, Oracle ASM Alert Logs, and Oracle Clusterware Alert Logs)

Using the Oracle ORAchk Health Check Tool

Run Oracle ORAchk to audit configuration settings and check system health.

> **Note:**
>
> Before running ORAchk, check for the latest version of Oracle Autonomous Health Framework, and download and install it. See My Oracle Support Note 2550798.1 for more information about downloading and installing the latest verion of Oracle Autonomous Health Framework.

Running ORAchk on Oracle Database Appliance 19.25 Baremetal Systems for New Installation

When you provision or upgrade to Oracle Database Appliance 19.25, ORAchk is installed using Oracle Autonomous Framework in the directory `/opt/oracle/dcs/oracle.ahf`.

To run orachk, use the following command:

```
[root@oak bin]# orachk
```

When all checks are finished, a detailed report is available. The output displays the location of the report in an HTML format and the location of a zip file if you want to upload the report. For example, you can choose the filter to show failed checks only, show checks with a Fail, Warning, Info, or Pass status, or any combination.

Review the Oracle Database Appliance Assessment Report and system health and troubleshoot any issues that are identified. The report includes a summary and filters that enable you to focus on specific areas.

Running ORAchk on Oracle Database Appliance 19.25 Virtualized Platform

When you provision or upgrade to Oracle Database Appliance 19.25, ORAchk is installed using Oracle Autonomous Framework in the directory `/opt/oracle.ahf`.

To run `orachk`, use the following command:

```
[root@oak bin]# oakcli orachk
```

Related Topics

- https://support.oracle.com/rs?type=doc&id=2550798.1

Generating and Viewing Oracle ORAchk Health Check Tool Reports in the Browser User Interface

Generate Oracle ORAchk Health Check Tool reports using the Browser User Interface.

1. Log into the Browser User Interface with the `oda-admin` username and password.

   ```
   https://Node0-host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Monitoring** tab.
3. In the Monitoring page, on the left navigation pane, click **ORAchk Report**.

   On the ORAchk Reports page, a list of all the generated ORAchk reports is displayed.
4. In the Actions menu for the ORAchk report you want to view, click **View**.

   The Oracle Database Appliance Assessment Report is displayed. It contains details of the health of your deployment, and lists current risks, recommendations for action, and links for additional information.
5. **To create an on-demand ORAchk report:** On the ORAchk Reports page, click **Create** and then click **Yes** in the confirmation box.

   The job to create an ORAchk report is submitted.
6. Click the link to view the status of the job. Once the job completes successfully, you can view the Oracle Database Appliance Assessment Report on the ORAchk Reports page.
7. **To delete an ORAchk report:** In the Actions menu for the ORAchk report you want to delete, click **Delete**.

Generating and Viewing Database Security Assessment Reports in the Browser User Interface

Generate and view Database Security Assessment Reports using the Browser User Interface.

1. Log into the Browser User Interface with the `oda-admin` username and password.

   ```
   https://Node0-host-ip-address:7093/mgmt/index.html
   ```

2. Click the **Security** tab.
3. In the Security page, on the left navigation pane, click **DBSAT Reports**.

   On the Database Security Assessment Reports page, a list of all the generated DBSAT reports is displayed.
4. In the Actions menu for the ORAchk report you want to view, click **View**.

   The Oracle Database Security Assessment Report is displayed. It contains details of the health of your deployment, and lists current risks, recommendations for action, and links for additional information.
5. **To create a DBSAT report:** On the DBSAT Reports page, click **Create** and then click **Yes** in the confirmation box.

   The job to create a DBSAT report is submitted.

6. Click the link to view the status of the job. Once the job completes successfully, you can view the Oracle Database Appliance Assessment Report on the DBSAT Reports page.
7. **To delete a DBSAT report:** In the Actions menu for the DBSAT report you want to delete, click **Delete**.

Running Oracle Trace File Analyzer (TFA) Collector Commands

Understand the installed location of `tfactl` and the options for the command.

About Using tfactl to Collect Diagnostic Information

When you provision or upgrade to Oracle Database Appliance 19.25, Oracle Trace File Analyzer (TFA) Collector is installed in the directory `/opt/oracle.ahf/bin/tfactl`. You can invoke the command line utility for TFA, `tfactl` from the directory `/opt/oracle.ahf/bin/tfactl`, or simply type `tfactl`.

You can use the following command options to run `tfactl`:

```
 /opt/oracle.ahf/bin/tfactl diagcollect -ips|-oda|-odalite|-dcs|-odabackup|
-odapatching|-odadataguard|-odaprovisioning|-odaconfig|-odasystem|-
odastorage|-database|
-asm|-crsclient|-dbclient|-dbwlm|-tns|-rhp|-procinfo|-afd|-crs|-cha|-wls|
-emagent|-oms|-ocm|-emplugins|-em|-acfs|-install|-cfgtools|-os|-ashhtml|-
ashtext|
-awrhtml|-awrtext -mask -sanitize
```

Table 19-1 Command Options for tfactl Tool

Option	Description
`-h`	(Optional) Describes all the options for this command.
`-ips`	(Optional) Use this option to view the diagnostic logs for the specified component.
`-oda`	(Optional) Use this option to view the logs for the entire Appliance.
`-odalite`	(Optional) Use this option to view the diagnostic logs for the odalite component.
`-dcs`	(Optional) Use this option to view the DCS log files.
`-odabackup`	(Optional) Use this option to view the diagnostic logs for the backup components for the Appliance.
`-odapatching`	(Optional) Use this option to view the diagnostic logs for patching components of the Appliance.
`-odadataguard`	(Optional) Use this option to view the diagnostic logs for Oracle Data Guard component of the Appliance.
`-odaprovisioning`	(Optional) Use this option to view provisioning logs for the Appliance.
`-odaconfig`	(Optional) Use this option to view configuration-related diagnostic logs.
`-odasystem`	(Optional) Use this option to view system information.

Table 19-1 (Cont.) Command Options for tfactl Tool

Option	Description
-odastorage	(Optional) Use this option to view the diagnostic logs for the Appliance storage.
-database	(Optional) Use this option to view database-related log files.
-asm	(Optional) Use this option to view the diagnostic logs for the Appliance.
-crsclient	(Optional) Use this option to view the diagnostic logs for the Appliance.
-dbclient	(Optional) Use this option to view the diagnostic logs for the Appliance.
-dbwlm	(Optional) Use this option to view the diagnostic logs for the specified component.
-tns	(Optional) Use this option to view the diagnostic logs for TNS.
-rhp	(Optional) Use this option to view the diagnostic logs for Rapid Home Provisioning.
-afd	(Optional) Use this option to view the diagnostic logs for Oracle ASM Filter Driver.
-crs	(Optional) Use this option to view the diagnostic logs for Oracle Clusterware.
-cha	(Optional) Use this option to view the diagnostic logs for the Cluster Health Monitor.
-wls	(Optional) Use this option to view the diagnostic logs for Oracle WebLogic Server.
-emagent	(Optional) Use this option to view the diagnostic logs for the Oracle Enterprise Manager agent.
-oms	(Optional) Use this option to view the diagnostic logs for the Oracle Enterprise Manager Management Service.
-ocm	(Optional) Use this option to view the diagnostic logs for the specified component.
-emplugins	(Optional) Use this option to view the diagnostic logs for Oracle Enterprise Manager plug-ins.
-em	(Optional) Use this option to view the diagnostic logs for Oracle Enterprise Manager deployment.
-acfs	(Optional) Use this option to view the diagnostic logs for Oracle ACFS storage.
-install	(Optional) Use this option to view the diagnostic logs for installation.
-cfgtools	(Optional) Use this option to view the diagnostic logs for the configuration tools.
-os	(Optional) Use this option to view the diagnostic logs for the operating system.
-ashhtml	(Optional) Use this option to view the diagnostic logs for the specified component.

Table 19-1 (Cont.) Command Options for tfactl Tool

Option	Description
`-ashtext`	(Optional) Use this option to view the diagnostic logs for the Appliance.
`-awrhtml`	(Optional) Use this option to view the diagnostic logs for the Appliance.
`-awrtext`	(Optional) Use this option to view the diagnostic logs for the specified component.
`-mask`	(Optional) Use this option to choose to mask sensitive data in the log collection.
`-sanitize`	(Optional) Use this option to choose to sanitize (redact) sensitive data in the log collection.

Usage Notes

You can use Trace File Collector (the `tfactl` command) to collect all log files for the Oracle Database Appliance components.

You can also use the command `odaadmcli manage diagcollect`, with similar command options, to collect the same diagnostic information.

For more information about using the `-mask` and `-sanitize` options, see the next topic.

Sanitizing Sensitive Information in Diagnostic Collections

Oracle Autonomous Health Framework uses Adaptive Classification and Redaction (ACR) to sanitize sensitive data.

After collecting copies of diagnostic data, Oracle Trace File Analyzer and Oracle ORAchk use Adaptive Classification and Redaction (ACR) to sanitize sensitive data in the collections. ACR uses a Machine Learning based engine to redact a pre-defined set of entity types in a given set of files. ACR also sanitizes or masks entities that occur in files and directory names. Sanitization replaces a sensitive value with random characters. Masking replaces a sensitive value with a series of asterisks ("*").

ACR currently sanitizes the following entity types:

- Host names
- IP addresses
- MAC addresses
- Oracle Database names
- Tablespace names
- Service names
- Ports
- Operating system user names

ACR also masks user data from the database appearing in block and redo dumps.

Example 19-4 Block dumps before redaction

```
14A533F40 00000000 00000000 00000000 002C0000 [..............,.]
14A533F50 35360C02 30352E30 31322E37 380C3938 [..650.507.2189.8]
14A533F60 31203433 37203332 2C303133 360C0200 [34 123 7310,...6]
```

Example 19-5 Block dumps after redaction

```
14A533F40 ******** ******** ******** ******** [****************]
14A533F50 ******** ******** ******** ******** [****************]
14A533F60 ******** ******** ******** ******** [****************]
```

Example 19-6 Redo dumps before redaction

```
col 74: [ 1] 80
col 75: [ 5] c4 0b 19 01 1f
col 76: [ 7] 78 77 06 16 0c 2f 26
```

Example 19-7 Redo dumps after redaction

```
col 74: [ 1] **
col 75: [ 5] ** ** ** ** **
col 76: [ 7] ** ** ** ** ** ** **
```

Redaction of Literal Values in SQL Statements in AWR, ASH and ADDM Reports

Automatic Workload Repository (AWR), Active Session History (ASH), and Automatic Database Diagnostic Monitor (ADDM) reports are HTML files, which contain sensitive entities such as hostnames, database names, and service names in the form of HTML tables. In addition to these sensitive entities, they also contain SQL statements, that can contain bind variables or literal values from tables. These literal values can be sensitive personal information (PI) stored in databases. ACR processes such reports to identify and redact both usual sensitive entities and literal values present in the SQL statements.

Sanitizing Sensitive Information Using odaadmcli Command

Use the `odaadmcli manage diagcollect` command to collect diagnostic logs for Oracle Database Appliance components. During collection, ACR can be used to redact (sanitize or mask) the diagnostic logs.

```
odaadmcli manage diagcollect [--dataMask|--dataSanitize]
```

In the command, the `--dataMask` option blocks out the sensitive data in all collections, for example, replaces `myhost1` with `*******`. The default is None. The `--dataSanitize` option replaces the sensitive data in all collections with random characters, for example, replaces `myhost1` with `orzhmv1`. The default is None.

Enabling Adaptive Classification and Redaction (ACR)

Oracle Database Appliance supports Adaptive Classification and Redaction (ACR) to sanitize sensitive data.

After collecting copies of diagnostic data, Oracle Database Appliance use Adaptive Classification and Redaction (ACR) to sanitize sensitive data in the collections. You can use the commands `odacli enable-acr` and `odacli disable-acr` to enable or disable ACR across both nodes, not just on the local node.

> **See Also:**
>
> For more information about setting up the staging server for Adaptive Classification and Redaction (ACR), see My Oracle Support note 2882798.1.

Example 19-8 Describing current status of ACR

```
bash-4.2# odacli describe-acr
Trace File Redaction: Enabled
```

Example 19-9 Enabling ACR:

```
bash-4.2# odacli enable-acr

Job details
----------------------------------------------------------------
                 ID:  12bbf784-610a-40a8-b409-e74c58bc35aa
               Description:  Enable ACR job
                Status:  Created
                Created:  April 8, 2021 3:04:13 AM PDT
```

Example 19-10 Disabling ACR

```
bash-4.2# odacli disable-acr

Job details
----------------------------------------------------------------
                 ID:  1d69f8b3-3989-4192-bbb9-6518e425061a
               Description:  Disable ACR job
                Status:  Created
                Created:  April 8, 2021 3:04:13 AM PDT
```

Example 19-11 Enabling ACR during provisioning of the appliance

You can enable ACR during provisioning of the appliance by adding the `acr` option to the JSON file used for provisioning. Specify `true` or `false` for the field `acrEnable` in the JSON file. If the `acr` option is not specified, then ACR is disabled.

```
"acr": {
    "acrEnable": true
}
```

Related Topics

- https://support.oracle.com/rs?type=doc&id=2882798.1

Sanitizing Sensitive Information in Oracle Trace File Analyzer Collections

You can redact (sanitize or mask) Oracle Trace File Analyzer diagnostic collections.

Enabling Automatic Redaction

To enable automatic redaction, use the command:

```
tfactl set redact=[mask|sanitize|none]
```

In the command, the `-mask` option blocks out the sensitive data in all collections, for example, replaces `myhost1` with `*******`. The `-sanitize` option replaces the sensitive data in all collections with random characters, for example, replaces `myhost1` with `orzhmv1`. The `none` option does not mask or sanitize sensitive data in collections. The default is `none`.

Enabling On-Demand Redaction

You can redact collections on-demand, for example, `tfactl diagcollect -srdc ORA-00600 -mask or tfactl diagcollect -srdc ORA-00600 -sanitize`.

1. To mask sensitive data in all collections:

   ```
   tfactl set redact=mask
   ```

2. To sanitize sensitive data in all collections:

   ```
   tfactl set redact=sanitize
   ```

Example 19-12 Masking or Sanitizing Sensitive Data in a Specific Collection

```
tfactl diagcollect -srdc ORA-00600 -mask
tfactl diagcollect -srdc ORA-00600 -sanitize
```

Redacting and Sanitizing Entities in the BUI

Enable and disable trace file redaction, redact files, and show or hide sanitized entities using the Browser User Interface.

1. Log into the Browser User Interface with the `oda-admin` username and password.

   `https://Node0-host-ip-address:7093/mgmt/index.html`

2. Click the **Security** tab.
3. In the Security page, on the left navigation pane, click **Trace File Redaction**.
4. Click the **Trace File Redaction Status** tab.

   The current ACR status is displayed.
5. You can enable or disable ACR status based on the current ACR staus. For example, if the ACR status is disabled, then click **Enable** to enable ACR. The job to change the ACR status is submitted.
6. Click **Refresh Status** to refresh the ACR status display.
7. Click the **Redact Files** tab.
8. Specify the **Input File Path** of the file to be redacted. The file must be in the `.tar`, or `.gz`, or `.zip` file format.
9. Select either **Sanitize** or **Mask** for the **Redaction Mode**.
10. Click **Redact**. The job to redact files is submitted.
11. Click the **Show Sanitized Entities** tab.
12. Specify the **List of sanitized entities** and click **Show**. The list of sanitized entities are displayed.

Sanitizing Sensitive Information in Oracle ORAchk Output

You can sanitize Oracle ORAchk output.

To sanitize Oracle ORAchk output, include the `-sanitize` option, for example, `orachk -profile asm -sanitize`. You can also sanitize post process by passing in an existing log, HTML report, or a zip file, for example, `orachk -sanitize file_name`.

Example 19-13 Sanitizing Sensitive Information in Specific Collection IDs

```
orachk -sanitize comma_delimited_list_of_collection_IDs
```

Example 19-14 Sanitizing a File with Relative Path

```
orachk -sanitize new/orachk_node061919_053119_001343.zip
orachk is sanitizing
/scratch/testuser/may31/new/orachk_node061919_053119_001343.zip. Please
wait...

Sanitized collection is:
/scratch/testuser/may31/orachk_aydv061919_053119_001343.zip
orachk -sanitize ../orachk_node061919_053119_001343.zip
orachk is sanitizing
/scratch/testuser/may31/../orachk_node061919_053119_001343.zip. Please wait...

Sanitized collection is:
/scratch/testuser/may31/orachk_aydv061919_053119_001343.zip
```

Example 19-15 Sanitizing Oracle Autonomous Health Framework Debug Log

```
orachk -sanitize new/orachk_debug_053119_023653.log
orachk is sanitizing /scratch/testuser/may31/new/
orachk_debug_053119_023653.log.
Please wait...

Sanitized collection is: /scratch/testuser/may31/
orachk_debug_053119_023653.log
```

Example 19-16 Running Full Sanity Check

```
orachk -localonly -profile asm -sanitize -silentforce

Detailed report (html) -
/scratch/testuser/may31/orachk_node061919_053119_04448/
orachk_node061919_053119_04448.html

orachk is sanitizing /scratch/testuser/may31/orachk_node061919_053119_04448.
Please wait...

Sanitized collection is: /scratch/testuser/may31/
orachk_aydv061919_053119_04448

UPLOAD [if required] - /scratch/testuser/may31/
orachk_node061919_053119_04448.zip
```

To reverse lookup a sanitized value, use the command:

```
orachk -rmap all|comma_delimited_list_of_element_IDs
```

You can also use `orachk -rmap` to lookup a value sanitized by Oracle Trace File Analyzer.

Example 19-17 Printing the Reverse Map of Sanitized Elements

```
orachk -rmap MF_NK1,fcb63u2

_____________________________________________________________________________
__
| Entity Type | Substituted Entity Name | Original Entity Name |
_____________________________________________________________________________
__
| dbname      | MF_NK1                  | HR_DB1               |
| dbname      | fcb63u2                 | rac12c2              |
_____________________________________________________________________________
__
orachk -rmap all
```

Running the Disk Diagnostic Tool

Use the Disk Diagnostic Tool to help identify the cause of disk problems.

The tool produces a list of 14 disk checks for each node. To display details, where n represents the disk resource name, enter the following command:

```
# odaadmcli stordiag n
```

For example, to display detailed information for NVMe pd_00:

```
# odaadmcli stordiag pd_00
```

Running the Oracle Database Appliance Hardware Monitoring Tool

The Oracle Database Appliance Hardware Monitoring Tool displays the status of different hardware components in Oracle Database Appliance server.

The tool is implemented with the Trace File Analyzer collector. Use the tool both on bare-metal and on virtualized systems. The Oracle Database Appliance Hardware Monitoring Tool reports information only for the node on which you run the command. The information it displays in the output depend on the component that you select to review.

Bare Metal Platform

You can see the list of monitored components by running the command `odaadmcli show -h`

To see information about specific components, use the command syntax `odaadmcli show component`, where `component` is the hardware component that you want to query. For example, the command `odaadmcli show power` shows information specifically about the Oracle Database Appliance power supply:

```
# odaadmcli show power

NAME            HEALTH  HEALTH_DETAILS   PART_NO.      SERIAL_NO.
Power_Supply_0  OK            -          7079395     476856Z+1514CE056G

(Continued)
LOCATION    INPUT_POWER   OUTPUT_POWER   INLET_TEMP          EXHAUST_TEMP
PS0         Present       112 watts      28.000 degree C     34.938 degree C
```

Virtualized Platform

You can see the list of monitored components by running the command `oakcli show -h`

To see information about specific components, use the command syntax `oakcli show component`, where `component` is the hardware component that you want to query. For example,

the command `oakcli show power` shows information specifically about the Oracle Database Appliance power supply:

```
# oakcli show power

NAME            HEALTH HEALTH_DETAILS PART_NO. SERIAL_NO.
Power Supply_0  OK      -             7047410   476856F+1242CE0020
Power Supply_1  OK     -              7047410   476856F+1242CE004J

(Continued)

LOCATION  INPUT_POWER OUTPUT_POWER INLET_TEMP         EXHAUST_TEMP
PS0       Present     88 watts     31.250 degree C    34.188 degree C
PS1       Present     66 watts     31.250 degree C    34.188 degree C
```

> **Note:**
>
> Oracle Database Appliance Server Hardware Monitoring Tool is enabled during initial startup of ODA_BASE on Oracle Database Appliance Virtualized Platform. When it starts, the tool collects base statistics for about 5 minutes. During this time, the tool displays the message "Gathering Statistics…" message.

Disabling the Browser User Interface

You can also disable the Browser User Interface. Disabling the Browser User Interface means you can only manage your appliance through the command-line interface.

1. Log in to the appliance:

   ```
   ssh -l root oda-host-name
   ```

2. Stop the DCS controller. For High-Availability systems, run the command on both nodes.

   ```
   systemctl stop initdcscontroller
   ```

Preparing Log Files for Oracle Support Services

If you have a system fault that requires help from Oracle Support Services, then you may need to provide log records to help Oracle support diagnose your issue.

You can collect diagnostic information for your appliance in the following ways:

- Use the Bill Of Materials report saved in the `/opt/oracle/dcs/Inventory/` directory, to enable Oracle Support to help troubleshoot errors, if necessary.
- You can use Trace File Collector (the `tfactl` command) to collect all log files for the Oracle Database Appliance components.
- Use the command `odaadmcli manage diagcollect` to collect diagnostic files to send to Oracle Support Services.
- Use the Error Correlation report available in the `/opt/oracle/dcs/da/da_repo` directory.

The `odaadmcli manage diagcollect` command consolidates information from log files stored on Oracle Database Appliance into a single log file for use by Oracle Support Services. The location of the file is specified in the command output.

Example 19-18 Collecting log file information for a time period, masking sensitive data

```
# odaadmcli manage diagcollect --dataMask --fromTime 2019-08-12 --toTime
2019-08-25
DataMask is set as true
FromTime is set as: 2019-08-12
ToTime is set as: 2019-08-25
TFACTL command is: /opt/oracle/tfa/tfa_home/bin/tfactl
Data mask is set.
Collect data from 2019-08-12
Collect data to 2019-08-25
```

A

Oracle Database Appliance Software Configuration Defaults

Oracle Database Appliance software configuration defaults.

- Directory Paths for Oracle Database Appliance
  Understand the storage and mount configuration file paths for Oracle Database Appliance.
- Port Numbers and Protocols of Oracle Database Appliance Components
  Understand the port numbers and protocols of Oracle Database Appliance components.
- Location of Log Files
  Log files are available for actions performed in the command-line interface and Browser User Interface and are useful when you need to track and debug jobs.
- Oracle Groups and User Configurations for Oracle Database Appliance
  Review the groups and default users created when you deploy the appliance. All passwords are set to the password that you define during deployment.

Directory Paths for Oracle Database Appliance

Understand the storage and mount configuration file paths for Oracle Database Appliance.

Oracle homes on Oracle Database Appliance follow Optimal Flexible Architecture guidelines.

Directory Paths for Oracle Database Appliance

Item	Directory Path
Grid home	`/u01/app/release-specific_name/gi owner`
Grid base	`/u01/app/gi owner`
Oracle home	`/u01/app/odaorahome/rdbms owner/product/rdbms_version/dbhome_home_sequence_number`
Oracle base	`/u01/app/odaorabase/rdbms owner`
Oracle Inventory	`/u01/app/oraInventory`
Java Runtime Environment (JRE)	`/opt/oracle/dcs/java/release_specific_JRE_version/`
Java SDK	`/opt/oracle/dcs/odajavasdk/`
Oracle ACFS file system	`/opt/oracle/oak/pkgrepos/orapkgs/clones` Starting with Oracle Database Appliance release 19.12, a new Oracle ACFS file system is created to store the Oracle Database and Oracle Grid Infrastructure clone files. The DATA disk group provides a space of 150 GB for the file system.

Port Numbers and Protocols of Oracle Database Appliance Components

Understand the port numbers and protocols of Oracle Database Appliance components.

Table A-1 Port Numbers and Protocols of Oracle Database Appliance Components

Component	Description	Default Port Number	Protocol	Network
BUI	A web-based application to administer and manage the Oracle Database Appliance.	7093	TCP	public
dcs-agent	The DCS agent is a REST application that performs various operations on the appliance.	7070, 7071	TCP	public and private
dcs-admin	This application performs lifecycle management operations on the DCS agent, DCS controller, ODACLI, and other components.	7060	TCP	public and private
MySQL	The metadata store used by DCS agent.	3306	TCP	private
Oracle HAMI	Oracle High Availability Metadata Infrastructure service providing distributed services required by DCS including locking and synchronizing configuration details in the cluster.	22001	TCP	private
oakd	Managing the storage components of Oracle Database Appliance.	Dynamic	TCP	private

Table A-1 (Cont.) Port Numbers and Protocols of Oracle Database Appliance Components

Component	Description	Default Port Number	Protocol	Network
Oracle ASR Manager	Oracle Auto Service Request. Refer to Oracle Auto Service Request documentation for other ports used by ASR Manager.	16161	TCP	public
Oracle ILOM/IPMI	Oracle ILOM for Oracle Database Appliance servers.	623	UDP	public
Oracle Database listeners	Database listeners, including local listener and SCAN listener.	1521	TCP	public
Oracle ASM Listener	Oracle ASM listener	1525	TCP	private
EM Express	Oracle Enterprise Manager Database Express (EM Express).	5500 and increment one per database	TCP	public
ssh	SSH connection to the Oracle Database Appliance servers.	22	SSH over TCP	public and private
VNC Server	The VNC server created by KVM for KVM virtual guest machines	5900+	TCP	localhost (127.0.0.1)
NFS	The central repository on bare metal system is shared by the DB Systems using NFS. Refer to the Oracle Linux documentation to identify ports used by NFS.	2049 - nfs	TCP	private

Note: IP addresses of the private network that Oracle Database Appliance uses are 192.168.16.0/24 and 192.168.17.0/24.

Related Topics

- Port Numbers and Protocols of Oracle Components

> **See Also:**
>
> *Oracle Real Application Clusters Installation Guide for Linux and UNIX* for port numbers and protocols used by Oracle Database components

Location of Log Files

Log files are available for actions performed in the command-line interface and Browser User Interface and are useful when you need to track and debug jobs.

You can also use the Oracle Appliance Manager Browser User Interface to view job activity, including the tasks that make up the job. The status of each task appears in the Browser User Interface and you can drill down to get greater details.

If you log a Service Request, upload all of the logs in the `/opt/oracle/dcs/log`, `/opt/oracle/dcs/da/da_repo`, and `/opt/oracle/dcs/dcscli/log` directories.

Patching Log Files

All patching-related information is logged in the `dcs-agent` log in the directory `/opt/oracle/dcs/log/`.

Error Correlation Reports

If a DCS job fails, an Error Correlation report is generated and is available in the `/opt/oracle/dcs/da/da_repo` directory.

DCS Agent Log Directories

Agent-specific activities are logged in the `dcs-agent` log.

The DCS Agent, controller, and command-line interface output are stored in the directory `/opt/oracle/dcs/log/`.

If an error occurs in the command-line interface layer, then the output will show in the `/opt/oracle/dcs/dcscli/log/dcscli.log` file first.

DCS Components Log Directories

The Oracle HAMI logs are stored in the directory `/opt/oracle/hami/base`, `/opt/oracle/hami/obase`, and `/opt/oracle/dcs/hami/log`. The Oracle HAMI logs are `hamiws_`*`pid_n`*`.trc`, `hamictl_`*`timestamp_pid`*`.log`, and `hamiwsmkwallets_`*`pid`*`.trc`.

The MySQL logs are stored in the log `/opt/oracle/dcs/mysql/log/mysqldb.log`. The log contains information about MySQL such as startup, shutdown, connection transaction details, and errors encountered when running MySQL.

Storage Logs

Storage-related activity is logged in the `oakd` log file.

`/opt/oracle/oak/log/`*`hostname`*`/oak/oakd.log`

For example, `/opt/oracle/oak/log/myhost/oak/oakd.log`

Use the `odaadmcli manage diagcollect` command to collect diagnostic logs for storage components. The files are saved in the `oakdiag` directory.

`/opt/oracle/oak/log/hostname/oakdiag/file name.tar.gz`

For example, `/opt/oracle/oak/log/myhost/oakdiag/oakStorage-myhost-20161120_2217.tar.gz`

Upgrade using Data Preserving Reprovisioning Logs

The `odaupgradeutil` utility logs are stored at `opt/oracle/oak/restore/log`.

Database Logs

Online logs are stored in the `/u03/app/db_user/redo/` directory.

Oracle Fast Recovery Area (FRA) is located in the `/u03/app/db_user/fast_recovery_area` directory.

Database alert logs are stored in the location `$ORACLE_BASE/diag/rdbms/database_unique_name`.

Oracle Auto Service Request (Oracle ASR) Log Files

All log files for Oracle ASR are located in the `/var/opt/asrmanager/log/` directory.

Oracle Groups and User Configurations for Oracle Database Appliance

Review the groups and default users created when you deploy the appliance. All passwords are set to the password that you define during deployment.

Default Operating System Groups and User Configurations

Table A-2 Default Operating System Groups and Users Configuration for Oracle Database Appliance

Groups and Users	Default Value
Oracle Grid Infrastructure installation owner	`grid`, UID 1001
Oracle Database installation owner	`oracle`, UID 1000
Oracle Database system administrator	`sys`
Oracle Database generic administrator	`system`
Oracle Inventory system privileges group	`oinstall`, GID 1001
Oracle ASM Administrators system privileges	`asmadmin`, GID 1004
Oracle ASM Users system privileges	`asmdba`, GID 1006
Oracle ASM Operator system privileges	`asmoper`, GID 1005
Oracle Database Administrators system privileges	`dba`, GID 1003
Oracle Database Operator system privileges	`dbaoper`, GID 1002

Oracle Groups and User Configurations

You can use the Browser User Interface or the `odacli create-appliance` command and a JSON file to deploy the appliance. The following configurations are supported:

- 2 Users with 6 groups (operating system role separation)
- Single User with 6 groups (no operating system role separation)
- Single user with 2 groups (no operating system role separation)

You can customize `groupname`, `username`, and `UID`.

B

Oracle Database Appliance Storage Defaults

Review this section to understand Oracle Database Appliance storage architecture and options and how to determine usable storage.

- About Database File Storage
  Use Oracle Automatic Storage Management (Oracle ASM) or Oracle Automatic Storage Management Cluster File System (Oracle ACFS) for database files storage.
- Oracle ACFS Mount Points and Storage Space
  Review Oracle ASM Cluster file system (ACFS) mount points for Oracle Database Appliance.
- Displaying Mounted Disk Details
  Use the Oracle Automatic Storage Management `lsdg` command to display mounted disk groups and their information for Oracle Database Appliance.
- Usable Space on Oracle Database Appliance X9-2S and X9-2L
  Review the table for the approximate amount of usable space for Oracle Database Appliance X9-2S and X9-2L.
- Usable Space on Oracle Database Appliance X9-2-HA
  Review the table for the approximate amount of usable space for high performance and high capacity storage on Oracle Database Appliance X9-2-HA.
- Usable Free Space with Oracle ASM
  When Oracle ASM calculates usable Free Space, it determines the amount of space to reserve in the case of a disk failure.

About Database File Storage

Use Oracle Automatic Storage Management (Oracle ASM) or Oracle Automatic Storage Management Cluster File System (Oracle ACFS) for database files storage.

Database file systems are used exclusively for storing database files, and they include separate file systems for data files and archive files or backups. You determine the type of database storage when you create the database.

About Oracle ASM Database Storage

With Oracle ASM, database datafiles are stored in DATA disk group. On X8-2, X7-2S, X7-2M, and X6-2 S/M/L systems, which do not have REDO disk group, redo and archive files are stored in RECO disk group. X7-2-HA and other older High-Availability systems have dedicated disks for REDO disk group; redo logs are stored in the REDO disk group.

Reserved storage is the amount of Oracle Automatic Storage Management (Oracle ASM) storage required to restore redundancy in the event of a disk failure. If you use the reserve storage capacity, then the system continues to run, and it is protected through Oracle ASM mirroring. However, in the event of a disk failure, the system is then running in a non-protected and degraded mode. In this event, you must replace disks immediately. If there is no reserved space available, then rebalance cannot restore redundancy after a disk failure.

If you specify mirroring for a file, then Oracle ASM automatically stores redundant copies of the file extents in separate failure groups. Failure groups apply to normal, high, and flex

redundancy disk groups. You can define the failure groups for each disk group when you create or alter the disk group.

> **See Also:**
>
> *Oracle Automatic Storage Management Administrator's Guide* in the Oracle Database documentation library for information about Oracle ASM Mirroring and Disk Group Redundancy.

About Oracle ACFS Database Storage

With Oracle ACFS, an Oracle ACFS file system is created from DATA disk group for each database to store datafiles, and an Oracle ACFS file system is created from RECO disk group for redo and fast recovery area for all databases.

Each database created under ACFS has it own mount points, `/u02/app/oracle/oradata/dbid`. When you delete a database, you should clean up the files.

Storage Configuration Options

When Oracle Database Appliance is deployed, you can configure how the storage capacity is shared between DATA disk group and RECO disk group. You can choose anywhere from 10% to 90% for DATA and the remainder for RECO. The amount of usable storage is determined by the percentage configured for DATA. You can run the command `asmcmd lsdg` to determine the usable storage on the DATA disk group. You can also run the command `odacli list-dbstorages` command to determine the storage available.

Oracle ACFS Mount Points and Storage Space

Review Oracle ASM Cluster file system (ACFS) mount points for Oracle Database Appliance.

If you select Oracle Automatic Storage Management (Oracle ASM) for database storage when you create a database, then an Oracle ASM Cluster file system (ACFS) is not created. All files are in an Oracle ASM disk group.

If you select Oracle ACFS for database storage, then each database has its own Oracle ACFS mount point:

- DATA disk group: `/u02/app/oracleuser/oradata/db_name`
- RECO disk group: `/u03/app/oracleuser`.
- REDO disk group: `/u04/app/oracleuser/redo/db_name`

With Oracle ACFS, the following are created:

- A 100G ACFS is created from +DATA diskgroup for each database. This Oracle ACFS automatically extends the space on demand.
- A common Oracle ACFS with 25% of +RECO diskgroup is created with auto extension on. This file system is used for fast recovery area.
- For Oracle Database Appliance single-node and High-Availability High Performance systems, the REDO volume is created on the RECO disk group. For Oracle Database Appliance High-Availability High Capacity system, the REDO volume is created on FLASH disk group. The REDO volume stores redo logs.

Table B-1 Oracle ACFS Mount Points and Related Oracle ASM Disk Groups and Volume Information

File System	Oracle ASM Disk Group	Oracle ASM Dynamic Volume	Mount Point
DATA	+DATA	/dev/asm/dat*dbname-nnn* For example: /dev/asm/datodacn-123	/u02/app/*oracleuser*/oradata/*dbname* For example: /u02/app/example/oradata/odacn
RECO	+RECO	/dev/asm/reco-*nn*	/u03/app/*oracleuser* This mount point is shared by all databases for fast_recovery_area and redo logs. For fast_recovery_area, the path is: /u03/app/*oracleuser*/fast_recovery_area/*db_name*
REDO	+RECO/ +FLASH depending on environment type (High Capacity or High Performance)	/dev/asm/rdo/dbName-*nnn*	/u04/app/*oracleuser*/redo/*db_name* For example: /u04/app/oracle/redo/mydb

Example B-1 Oracle ACFS Storage Space

When the Oracle ACFS file systems are created, they do not initially consume all of the storage in the appliance. Space is preserved for additional repositories, or in some cases, database files stored directly in Oracle ASM. You can check for available storage space in your file systems by running the operating system command `df -k` as shown in the following example.

```
# df -k
Filesystem                              1K-blocks   Used         Available  Use%   Mounted
on
/dev/mapper/VolGroupSys-LogVolRoot      30963708    14203568     15187276   49%    /
tmpfs                                   65952292      647800     65304492
1%     /dev/shm
/dev/sda1                                 495844       43872       426372   10%    /boot
/dev/mapper/VolGroupSys-LogVolOpt       61927420    18594420     40187272   32%    /opt
/dev/mapper/VolGroupSys-LogVolU01      103212320    49621560     48347880   51%    /u01
/dev/asm/reco-62                        76546048     1469676     75076372
2%     /u03/app/oracle
/dev/asm/datrdb2-268                   104857600     3872368    100985232
4%     /u02/app/oracle/oradata/rdb2
/dev/asm/datndb11-268                  104857600      247160    104610440
1%     /u02/app/oracle/oradata/ndb11
/dev/asm/datndb12-268                  104857600      247160    104610440
1%     /u02/app/oracle/oradata/ndb12
```

Displaying Mounted Disk Details

Use the Oracle Automatic Storage Management `lsdg` command to display mounted disk groups and their information for Oracle Database Appliance.

To display information about a specific disk group, specify the disk group in the command.

1. Log in as a `grid` user.
2. Run the Oracle Automatic Storage Management `lsdg` command.

Example B-2 Determining Storage on the DATA Disk Group

```
ASMCMD [+] > lsdg data

State    Type    Rebal  Sector  Block       AU  Total_MB  Free_MB
Req_mir_free_MB  Usable_file_MB
MOUNTED  NORMAL  N         512   4096  4194304     12288     8835
1117            3859

(continued)
Offline_disks  Voting_files  Name
            0             N  DATA
```

Usable Space on Oracle Database Appliance X9-2S and X9-2L

Review the table for the approximate amount of usable space for Oracle Database Appliance X9-2S and X9-2L.

Oracle Database Appliance X9-2S and X9-2L use 6.4TB NVMe. The storage capacity is the aggregate for DATA and RECO disk groups. The raw storage capacity in the table is based on disk hardware terabytes (based on 1 kilobyte equals 1000 bytes). Approximate usable storage capacity is shown in software storage terabytes (based on 1 kilobyte equals 1024 bytes). Approximate usable storage also accounts for 15% reserved space required to restore full redundancy in case of disk failure (not applicable to the two-drive configuration). The high redundancy values in the table describe the space needed to recover from one failure only.

> **Note:**
>
> The usable storage space described in this topic are approximate values, and may vary as per your deployment.

Table B-2 Usable Disk Capacity on Oracle Database Appliance X9-2S and X9-2L

Number of Drives	Raw Storage	Usable Storage with Normal Redundancy (Double Mirroring)	Usable Storage with High Redundancy (Triple Mirroring)
X9-2S/X9-2L (2 drives)	13.6 TB	6.2 TB	Not Applicable
X8-2L (4 drives)	27.2 TB	10.5 TB	7.0 TB
X8-2L (6 drives)	40.8 TB	15.8 TB	10.5 TB
X8-2L (8 drives)	54.4 TB	21.0 TB	14.0 TB
X8-2L (10 drives)	68 TB	26.3 TB	17.5 TB
X8-2L (12 drives)	81.6 TB	31.5 TB	21.0 TB

Usable Space on Oracle Database Appliance X9-2-HA

Review the table for the approximate amount of usable space for high performance and high capacity storage on Oracle Database Appliance X9-2-HA.

Usable Space for High Performance

> **Note:**
>
> The usable storage space described in this topic are approximate values, and may vary as per your deployment.

Oracle Database Appliance X9-2-HA with high performance configuration uses 7.68TB solid state drives (SSDs). The storage capacity is the aggregate for DATA and RECO disk groups. The raw storage capacity in the table is based on disk hardware terabytes (based on 1 kilobyte equals 1000 bytes). Approximate usable storage capacity is shown in software storage terabytes (based on 1 kilobyte equals 1024 bytes). Approximate usable storage also accounts for 15% reserved space required to restore full redundancy in case of disk failure. The high redundancy values in the table describe the space needed to recover from one failure only.

Table B-3 Usable Disk Capacity for High Performance on Oracle Database Appliance X9-2-HA

Number of Drives	Raw Storage	Usable Storage with Normal Redundancy (Double Mirroring)	Usable Storage with High Redundancy (Triple Mirroring)
6 SSD drives	46 TB	17.8 TB	11.9 TB
12 SSD drives	92 TB	35.6 TB	23.7 TB
18 SSD drives	138 TB	53.4 TB	35.6 TB
24 SSD drives	184 TB	71.2 TB	47.5 TB
48 SSD drives	369 TB	142.5 TB	95.0 TB

Usable Space for High Capacity

Oracle Database Appliance X9-2-HA with high capacity configuration uses 7.68TB solid state drives (SSDs) and 14TB hard disk drives (HDDs). SSD storage capacity is allocated to the FLASH disk group. HDD storage capacity is the aggregate for DATA and RECO disk groups. The raw storage capacity in the table is based on disk hardware terabytes (based on 1 kilobyte equals 1000 bytes). Approximate usable storage capacity is shown in software storage terabytes (based on 1 kilobyte equals 1024 bytes). Approximate usable storage also accounts for 15% reserved space required to restore full redundancy in case of disk failure. The high redundancy values in the table describe the space needed to recover from one failure only.

> **Note:**
>
> The usable storage space described in this topic are approximate values, and may vary as per your deployment.

Table B-4 Usable Disk Capacity for High Capacity on Oracle Database Appliance X9-2-HA

Number of Drives	Raw Storage	Usable Storage with Normal Redundancy (Double Mirroring)	Usable Storage with High Redundancy (Triple Mirroring)
6 SSD drives and 18 HDD drives	SSDs: 46 TB HDDs: 324 TB	SSDs: 17.8 TB HDDs: 125.2 TB	SSDs: 11.9 TB HDDs: 83.5 TB
12 SSD drives and 36 HDD drives	SSDs: 92 TB HDDs: 648 TB	SSDs: 35.6 TB HDDs: 250.5 TB	SSDs: 23.7 TB HDDs: 167 TB

Usable Free Space with Oracle ASM

When Oracle ASM calculates usable Free Space, it determines the amount of space to reserve in the case of a disk failure.

For Oracle Database Appliance with 2 or 4 NVMe drives, this reserved space is not required. When you query Oracle ASM or Oracle Database Appliance commands to view the amount of storage available, the `usable_File_MB` value may report a negative number.

Table B-5 Oracle ASM Calculations

Number of Drives	Redundancy	Total_MB	Free_MB	Req_mir_free_ MB	Usable_file_MB	Name
2	NORMAL	4894016	4893372	0	1220644	RECO/
4	NORMAL	1231176	1230996	610468	305150	RECO/

> **Note:**
>
> Note: 1TB = MB divided by 1024^2

How Oracle ASM and Oracle Database Appliance Define Capacity

The following table describes how capacity terms are defined by Oracle ASM and Oracle Database Appliance.

Table B-6 Definition of Terminology

Term	Oracle ASM Definition	Oracle Database Appliance Definition
`Total_MB`	Size of the disk group in MB	Total usable storage. For example, for 2 NVMe drives, total usable storage is 5.8TB.
`Free_MB`	Free space in the disk group in MB, without regard to redundancy.	Total usable storage after formatting to Oracle ASM disk groups. For example, for 2 NVMe drives, total usable storage is 5.8TB.

Table B-6 (Cont.) Definition of Terminology

Term	Oracle ASM Definition	Oracle Database Appliance Definition
`Req_mir_free_MB`	Amount of space that must be available in the disk group to restore full redundancy after the worst failure that can be tolerated by the disk group.	Amount of space that must be available in the disk group to restore full redundancy after the most severe failure that can be tolerated by the disk group. This is the `REQUIRED_MIRROR_FREE_MB` column from the `V$ASM_DISKGROUP` view.
`Usable_file_MB`	Amount of free space, adjusted for mirroring, that is available for new files.	Total usable space taking into consideration the mirroring level. Oracle ASM also calculates the amount of space required.

C

Creating an Appliance Using JSON File

Understand the process to create an appliance using the command `odacli create-appliance` and view an example JSON file.

- Loading and Saving Appliance Configurations
  You can load a saved JSON file and use it to configure your appliance from the Browser User Interface.
- Readme for the Command odacli create-appliance
  If you want to create the appliance outside of the Browser User Interface, then review the readme file for details on how to create a JSON file to use with the command-line interface.
- Example JSON Files to Create a Single Node Appliance with the CLI
  Follow these JSON file examples to create a JSON file to deploy a single node appliance with the command `odacli create-appliance`.
- Example JSON File to Create a High-Availability Appliance with the CLI
  Follow the JSON file example to create a JSON file to deploy a high-availability appliance with the command `odacli create-appliance`.

Loading and Saving Appliance Configurations

You can load a saved JSON file and use it to configure your appliance from the Browser User Interface.

Using a Saved Configuration to Create a New Appliance in the Browser User Interface

You can load a saved configuration from your client machine, and create a new appliance.

1. Log into the Browser User Interface.
2. In the Create Appliance page, click **Browse** next to Load Configuration.
3. Select the JSON file from the local machine, being used to access the Browser User Interface.
4. The appliance configuration values from the JSON file are populated in the fields on the Create Appliance page.
5. Make any changes required, and submit the request to create the appliance

   Note that for security requirements, you must enter passwords manually.

Readme for the Command odacli create-appliance

If you want to create the appliance outside of the Browser User Interface, then review the readme file for details on how to create a JSON file to use with the command-line interface.

You can use the command `odacli create-appliance` and a JSON file that includes your appliance configuration settings to deploy the appliance instead of using the Browser User Interface.

> **Note:**
>
> It is important to review the readme and the examples carefully before creating your JSON file. If you do not enter your network and Oracle ILOM information correctly based on your setup, you will lose network access to both the host and Oracle ILOM.

Readme

The readme is located in the `/opt/oracle/dcs/sample` directory. Review the readme carefully along with the provided JSON example files. Create a JSON file with the necessary changes based on your environment and requirements.

Example JSON Files to Create a Single Node Appliance with the CLI

Follow these JSON file examples to create a JSON file to deploy a single node appliance with the command `odacli create-appliance`.

Use the example JSON files and the information located in the readme as a template to create a file for your environment. Examples and the readme are located in the `/opt/oracle/dcs/sample` directory.

> **Note:**
>
> If you do not enter your network and Oracle ILOM information correctly based on your setup, then network access is lost to both the host and Oracle ILOM.

When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

Note: To configure Oracle ASR during provisioning, always specify the asrType parameter in the JSON file.

Example C-1 JSON File to Create a Single Node Oracle Database Appliance with Role Separation

The following is an example of a JSON file that creates a single node appliance on Oracle Database Appliance. The example uses role separation.

```
   {
 "instance" : {
   "name" : "odambox",
   "instanceBaseName" : "odambox-c",
   "dbEdition" : "EE",
   "timeZone" : "UTC",
   "ntpServers" : ["10.0.3.14"],
   "dnsServers" : ["10.0.4.10","10.0.4.11","10.0.4.12"],
   "domainName" : "example.com",
   "isRoleSeparated" : true,
   "osUserGroup" : {
     "groups" : [ {
```

```
        "groupId" : 1001,
        "groupName" : "oinstall",
        "groupRole" : "oinstall"
      }, {
        "groupId" : 1002,
        "groupName" : "dbaoper",
        "groupRole" : "dbaoper"
      }, {
        "groupId" : 1003,
        "groupName" : "dba",
        "groupRole" : "dba"
      }, {
        "groupId" : 1004,
        "groupName" : "asmadmin",
        "groupRole" : "asmadmin"
      }, {
        "groupId" : 1005,
        "groupName" : "asmoper",
        "groupRole" : "asmoper"
      }, {
        "groupId" : 1006,
        "groupName" : "asmdba",
        "groupRole" : "asmdba"
      } ],
      "users" : [ {
        "userId" : 1000,
        "userName" : "oracle",
        "userRole" : "oracleUser"
      }, {
        "userId" : 1001,
        "userName" : "grid",
        "userRole" : "gridUser"
      } ]
    }
  },
  "sysOraHomeStorage": {
        "volumeSize": "160",
        "diskgroup": "DATA"
    }
  "nodes" : [ {
    "nodeNumber" : "0",
    "nodeName" : "odambox",
    "network" : [ {
       "nicName" : "btbond1",
       "ipAddress" : "10.0.1.11",
       "subNetMask" : "255.255.240.0",
       "gateway" : "10.0.0.1",
       "networkType" : [ "Public" ],
       "isDefaultNetwork" : true
      },
    ],
    "ilom" : {
      "ilomName":"odambox-c",
      "ipAddress":"10.0.2.10",
      "subNetMask":"255.255.240.0",
      "gateway":"10.0.0.1"
```

```
    }
} ],
"grid" : {
  "diskGroup" : [ {
    "diskGroupName" : "DATA",
    "redundancy" : "NORMAL",
    "diskPercentage" :80
  }, {
    "diskGroupName" : "RECO",
    "redundancy" : "NORMAL",
    "diskPercentage" :20
  } ],
"scan" : {
   "scanName":"odambox-scan",
   "ipAddresses":[
  "10.0.1.11"
   ]
  },
"vip":[
   {
  "nodeNumber":"0",
  "vipName":"odambox-vip",
  "ipAddress":"10.0.1.11"
   }
  ],
  "language" : "en",
  "enableAFD":"TRUE"
},
"database" : {
  "dbName" : "myTestDb",
  "databaseUniqueName":"myTestDb_sea1kj",
  "dbEdition" : "EE",
  "dbVersion" : "19.21.0.0",
  "dbHomeId":null,
  "instanceOnly" : false,
  "isCdb" : true,
  "pdBName" : "pdb1",
  "pdbAdminuserName" : "pdbuser",
  "enableTDE":true,
  "adminPassword" : "password",
  "dbType" : "SI",
  "dbTargetNodeNumber" : null,
  "dbClass" : "OLTP",
  "dbShape" : "odb1",
  "dbStorage" : "ACFS",
  "dbCharacterSet" : {
    "characterSet" : "AL32UTF8",
    "nlsCharacterset" : "AL16UTF16",
    "dbTerritory" : "AMERICA",
    "dbLanguage" : "AMERICAN"
  },
  "dbConsoleEnable" : false,
  "backupConfigId":null,
  "rmanBkupPassword": null
},
"asr" :{
```

```
    "asrType": "INTERNAL",
    "userName":"john.smith@example.com",
    "password":"password",
    "proxyServerName":"www-proxy.example.com",
    "proxyPort":"80",
    "proxyUserName":"proxy-user",
    "proxyPassword":"proxy-password",
    "httpsPortNumber":"16163"
  }
}
```

Example C-2 JSON File to Create a Single Node Oracle Database Appliance without Role Separation

The following is an example of a JSON file that creates an Oracle Database Appliance without using role separation. This example creates two groups (oinstall and dba) and one user ("oracle").

```
{
  "instance" : {
    "name" : "odambox",
    "instanceBaseName" : "odambox-c",
    "dbEdition" : "EE",
    "timeZone" : "UTC",
    "ntpServers" : ["10.0.3.14"],
    "dnsServers" : ["10.0.4.10","10.0.4.11","10.0.4.12"],
    "domainName" : "example.com",
    "isRoleSeparated" : false,
    "osUserGroup" : {
      "groups" : [ {
        "groupId" : 1001,
        "groupName" : "oinstall",
        "groupRole" : "oinstall"
      }, {
        "groupId" : 1002,
        "groupName" : "dba",
        "groupRole" : "dba"
      } ],
     "users" : [ {
        "userId" : 1000,
        "userName" : "oracle",
        "userRole" : "oracleUser"
      } ]
    }
  },
  "nodes" : [ {
    "nodeNumber" : "0",
    "nodeName" : "odambox",
    "network" : [ {
       "nicName" : "btbond1",
       "ipAddress" : "10.0.1.11",
       "subNetMask" : "255.255.240.0",
       "gateway" : "10.0.0.1",
       "networkType" : [ "Public" ],
       "isDefaultNetwork" : true
      },
```

```
  ],
  "ilom" : {
    "ilomName":"odambox-c",
    "ipAddress":"10.0.2.10",
    "subNetMask":"255.255.240.0",
    "gateway":"10.0.0.1"
   }
} ],
"grid" : {
  "diskGroup" : [ {
    "diskGroupName" : "DATA",
    "redundancy" : "NORMAL",
    "diskPercentage" :80
  }, {
    "diskGroupName" : "RECO",
    "redundancy" : "NORMAL",
    "diskPercentage" :20
  } ],
"scan" : {
   "scanName":"odambox-scan",
   "ipAddresses":[
  "10.0.1.11"
   ]
  },
"vip":[
   {
  "nodeNumber":"0",
  "vipName":"odambox-vip",
  "ipAddress":"10.0.1.11"
   }
  ],
  "language" : "en",
  "enableAFD":"TRUE"
},
"database" : {
  "dbName" : "myTestDb",
  "databaseUniqueName":"myTestDb_sea1kj",
  "dbEdition" : "EE",
  "dbVersion" : "19.21.0.0",
  "dbHomeId":null,
  "instanceOnly" : false,
  "isCdb" : true,
  "pdBName" : "pdb1",
  "pdbAdminuserName" : "pdbuser",
  "enableTDE":true,
  "adminPassword" : "password",
  "dbType" : "SI",
  "dbTargetNodeNumber" : null,
  "dbClass" : "OLTP",
  "dbShape" : "odb1",
  "dbStorage" : "ACFS",
  "dbCharacterSet" : {
    "characterSet" : "AL32UTF8",
    "nlsCharacterset" : "AL16UTF16",
    "dbTerritory" : "AMERICA",
    "dbLanguage" : "AMERICAN"
```

```
    },
    "dbConsoleEnable" : false,
    "backupConfigId":null,
    "rmanBkupPassword": null
  },
  "asr" :{
    "asrType": "INTERNAL",
    "userName":"john.smith@example.com",
    "password":"password",
    "proxyServerName":"www-proxy.example.com",
    "proxyPort":"80",
    "proxyUserName":"proxy-user",
    "proxyPassword":"proxy-password",
    "httpsPortNumber":"16163"
  }
}
```

Example JSON File to Create a High-Availability Appliance with the CLI

Follow the JSON file example to create a JSON file to deploy a high-availability appliance with the command `odacli create-appliance`.

Use the example JSON files and the information located in the readme as a template to create a file for your environment. Examples and the readme are located in the `/opt/oracle/dcs/sample` directory.

> **Note:**
>
> It is important to review the readme and the examples carefully before creating your JSON file. If you do not enter your network and Oracle ILOM information correctly based on your setup, then network access is lost to both the host and Oracle ILOM.

Example C-3 JSON File to Create a high-availability Oracle Database Appliance with Role Separation

The following is an example of a JSON file that creates a high-availability appliance on Oracle Database Appliance bare metal platform. The example uses role separation. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
   "instance":{
      "name":"odahabox",
      "instanceBaseName":"odahabox",
      "dbEdition":"EE",
      "timeZone":"UTC",
      "ntpServers" : ["10.0.3.14"],
      "dnsServers" : ["10.0.4.10","10.0.4.11","10.0.4.12"],
      "domainName":"example.com",
      "isRoleSeparated":true,
      "osUserGroup":{
```

```
      "groups":[
         {
            "groupId":1001,
            "groupName":"oinstall",
            "groupRole":"oinstall"
         },
         {
            "groupId":1002,
            "groupName":"dbaoper",
            "groupRole":"dbaoper"
         },
         {
            "groupId":1003,
            "groupName":"dba",
            "groupRole":"dba"
         },
         {
            "groupId":1004,
            "groupName":"asmadmin",
            "groupRole":"asmadmin"
         },
         {
            "groupId":1005,
            "groupName":"asmoper",
            "groupRole":"asmoper"
         },
         {
            "groupId":1006,
            "groupName":"asmdba",
            "groupRole":"asmdba"
         }
      ],
      "users":[
         {
            "userId":101,
            "userName":"grid",
            "userRole":"gridUser"
         },
         {
            "userId":102,
            "userName":"oracle",
            "userRole":"oracleUser"
         }
      ]
   },
   "objectStoreCredentials":null
},
"sysOraHomeStorage": {
     "volumeSize": "160",
     "diskgroup": "DATA"
 }
"nodes":[
   {
      "nodeNumber":"0",
      "nodeName":"odahaboxc1n2",
      "network":[
```

```
            {
               "nicName":"btbond1",
               "ipAddress":"10.31.98.133",
               "subNetMask":"255.255.240.0",
               "gateway":"10.31.96.1",
               "networkType":[
                  "Public"
               ],
               "isDefaultNetwork":true
            }
         ],
         "ilom":{
            "ilomName":"odahabox2-c",
            "ipAddress":"10.31.16.140",
            "subNetMask":"255.255.240.0",
            "gateway":"10.31.16.1"
         }
      },
      {
         "nodeNumber":"1",
         "nodeName":"odahaboxc1n1",
         "network":[
            {
               "nicName":"btbond1",
               "ipAddress":"10.31.98.132",
               "subNetMask":"255.255.240.0",
               "gateway":"10.31.96.1",
               "networkType":[
                  "Public"
               ],
               "isDefaultNetwork":true
            }
         ],
         "ilom":{
            "ilomName":"odahabox1-c",
            "ipAddress":"10.31.16.139",
            "subNetMask":"255.255.240.0",
            "gateway":"10.31.16.1"
         }
      }
   ],
   "grid":{
      "diskGroup":[
         {
            "diskGroupName":"DATA",
            "redundancy":"HIGH",
            "diskPercentage":80
         },
         {
            "diskGroupName":"RECO",
            "redundancy":"HIGH",
            "diskPercentage":20
         },
      ],
      "scan":{
         "scanName":"odahaboxc1-scan",
```

```
            "ipAddresses":[
               "10.31.98.182",
               "10.31.98.183"
            ]
         },
         "vip":[
            {
               "nodeNumber":"0",
               "vipName":"odahaboxc1n2-vip",
               "ipAddress":"10.31.98.159"
            },
            {
               "nodeNumber":"1",
               "vipName":"odahaboxc1n1-vip",
               "ipAddress":"10.31.98.158"
            }
         ],
         "language":"en",
         "enableAFD":"TRUE"
      },
      "database":{
         "dbName":"myTestDb",
         "databaseUniqueName":"myTestDb_sea1kj",
         "dbVersion":"19.21.0.0",
         "dbHomeId":null,
         "instanceOnly":false,
         "isCdb":true,
         "pdBName":"pdb1",
         "pdbAdminuserName":"pdbuser",
         "enableTDE":true,
         "adminPassword":"password",
         "dbType":"RAC",
         "dbTargetNodeNumber":null,
         "dbClass":"OLTP",
         "dbShape":"odb1",
         "dbStorage":"ACFS",
         "dbCharacterSet":{
            "characterSet":"AL32UTF8",
            "nlsCharacterset":"AL16UTF16",
            "dbTerritory":"AMERICA",
            "dbLanguage":"AMERICAN"
         },
         "dbConsoleEnable":false,
         "backupConfigId":null
      },
      "asr":null
}
```

Example C-4 JSON File to Create a high-availability Oracle Database Appliance with Enterprise Edition High Availability Enabled

The following is an example of a JSON file that creates a high-availability appliance on Oracle Database Appliance bare metal platform. The example shows enabling of Enterprise Edition

High Availability. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "clusterName": "cl-test",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
                    "id": 1005,
                    "role": "asmoper"
                },
                {
                    "name": "asmdba",
                    "id": 1006,
                    "role": "asmdba"
                }
            ],
            "users": [
                {
                    "name": "grid",
                    "id": 1000,
                    "role": "gridUser"
                },
                {
                    "name": "oracle",
                    "id": 1001,
                    "role": "oracleUser"
```

```
                }
            ]
        }
    },
    "database": {
        "name": "dbtest",
        "uniqueName": "dbtest",
        "domainName": "test_domain",
        "version": "19.15.0.0.220419",
        "edition": "EE",
        "type": "SI",
        "dbClass": "OLTP",
        "shape": "odb2",
        "role": "PRIMARY",
        "targetNodeNumber": "0",
        "enableDbConsole": false,
        "enableEEHA": true,
        "enableUnifiedAuditing": true,
        "redundancy": "MIRROR", <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
        "characterSet": {
            "characterSet": "AL32UTF8",
            "nlsCharacterset": "AL16UTF16",
            "dbTerritory": "AMERICA",
            "dbLanguage": "AMERICAN"
        },
         "enableTDE": false,
         "isCdb": true,
            "pdbName": "pdb1",
            "pdbAdminUser": "pdbadmin"
    },
    "network": {
        "domainName": "test_domain",
        "ntpServers": [],
        "dnsServers": [
            "xx.xxx.xx.xxx"
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xx.xx.xxx.xx"
            },
            {
                "name": "node2",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xx.xx.xxx.xx"
```

```
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "test-scan",
        "scanIps": [
            "xx.xx.xxx.xx",
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en"
    }
}
```

Example C-5 JSON File to Create a high-availability Oracle Database Appliance with Standard Edition High Availability Enabled

The following is an example of a JSON file that creates a high-availability appliance on Oracle Database Appliance bare metal platform. The example shows enabling of Standard Edition High Availability. When using the example to create your JSON file, change the definitions to match your environment. The password must meet password requirements.

```
{
    "system": {
        "name": "test_example",
        "shape": "odb2",
        "customMemorySize": "24G",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        "clusterName": "cl-test",
        "enableRoleSeparation": true,
        "customRoleSeparation": {
            "groups": [
                {
                    "name": "oinstall",
                    "id": 1001,
                    "role": "oinstall"
                },
                {
                    "name": "dbaoper",
                    "id": 1002,
                    "role": "dbaoper"
                },
                {
                    "name": "dba",
                    "id": 1003,
                    "role": "dba"
                },
                {
                    "name": "asmadmin",
                    "id": 1004,
                    "role": "asmadmin"
                },
                {
                    "name": "asmoper",
```

```
                        "id": 1005,
                        "role": "asmoper"
                    },
                    {
                        "name": "asmdba",
                        "id": 1006,
                        "role": "asmdba"
                    }
                ],
                "users": [
                    {
                        "name": "grid",
                        "id": 1000,
                        "role": "gridUser"
                    },
                    {
                        "name": "oracle",
                        "id": 1001,
                        "role": "oracleUser"
                    }
                ]
            }
        },
        "database": {
            "name": "dbtest",
            "uniqueName": "dbtest",
            "domainName": "test_domain",
            "version": "19.15.0.0.220419",
            "edition": "SE",
            "type": "SI",
            "dbClass": "OLTP",
            "shape": "odb2",
            "role": "PRIMARY",
            "targetNodeNumber": "0",
            "enableDbConsole": false,
            "enableSEHA": true,
            "enableUnifiedAuditing": true,
            "redundancy": "MIRROR", <<< if diskgroup redundancy is FLEX, then
database redundancy must be set to one of "HIGH" or "MIRROR", otherwise, can
be null
            "characterSet": {
                "characterSet": "AL32UTF8",
                "nlsCharacterset": "AL16UTF16",
                "dbTerritory": "AMERICA",
                "dbLanguage": "AMERICAN"
            },
             "enableTDE": false,
             "isCdb": true,
                "pdbName": "pdb1",
                "pdbAdminUser": "pdbadmin"
        },
        "network": {
            "domainName": "test_domain",
            "ntpServers": [],
            "dnsServers": [
                "xx.xxx.xx.xxx"
```

```
        ],
        "nodes": [
            {
                "name": "node1",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 0,
                "vipName": "node1-vip",
                "vipAddress": "xx.xx.xxx.xx"
            },
            {
                "name": "node2",
                "ipAddress": "xx.xx.xxx.xx",
                "netmask": "xxx.xxx.xxx.x",
                "gateway": "xx.xx.xxx.x",
                "number": 1,
                "vipName": "node2-vip",
                "vipAddress": "xx.xx.xxx.xx"
            }
        ],
        "publicVNetwork": "vnet1",
        "scanName": "test-scan",
        "scanIps": [
            "xx.xx.xxx.xx",
            "xx.xx.xxx.xx"
        ]
    },
    "grid": {
        "language": "en"
    }
}
```

D

Database Shapes for Oracle Database Appliance

Use the information in this appendix to select database shapes, or templates, for your planned databases.

- About Database Shapes and Classes
  Review this information to help determine the database shape (also known as a database template) and class to use based on the common workload your databases perform.
- Database Shapes for the OLTP Class
  Use Oracle Database Appliance OLTP Database Shapes if your database workload is primarily online transaction processing (OLTP).
- Database Shapes for the In-Memory Class
  Use Oracle Database Appliance In-Memory (IMDB) database shapes if your database workload can fit in memory, and can benefit from in-memory performance capabilities.
- Database Shapes for the DSS Class
  Use DSS database shapes if your database workload is primarily decision support services (DSS) or data warehousing.

About Database Shapes and Classes

Review this information to help determine the database shape (also known as a database template) and class to use based on the common workload your databases perform.

Oracle Database Appliance shapes define databases with parameters selected specifically to optimize performance on Oracle Database Appliance. In addition, these shapes help you to set up appropriate instance caging.

Oracle Database Appliance enables you to consolidate many databases into a single system. Consolidation can minimize idle resources, maximize efficiency, and lower costs. By using instance caging in conjunction with Oracle Database Resource Manager (the Resource Manager), you can provide desired levels of service across multiple instances on a single Oracle Database Appliance.

Oracle Database Appliance shapes are already tuned for the size of each database instance workload. They are designed to run on a specific number of cores. Caging ensures that each database workload is restricted to the set of cores allocated by the shape, enabling multiple databases to run concurrently with no performance degradation, up to the capacity of Oracle Database Appliance. You can select database shape sizes larger than your current needs to provide for planned growth, which you accommodate later by adjusting System Global Area (SGA) and Program Global Area (PGA) sizes as well as the number of cores.

> **Note:**
>
> Oracle strongly recommends that you use the Oracle Database Appliance shapes, because they implement best practices and are configured specifically for Oracle Database Appliance.

> **Important:**
>
> Except for the largest database shape, all other database shapes supported on the Oracle Database Appliance bare metal system are also supported on Oracle Database Appliance DB system.

Choosing a Database Shape

Database shapes are configured specifically for the type of database workload that you want to carry out on your databases on Oracle Database Appliance. Choose the shape that best matches the common workload your databases perform (OLTP, DSS, In-Memory).

The database sizing tables provide shape names and sizing based on the number of CPUs and memory attributes for each type of database workload.

Identify the shape type that is appropriate to your database workload and hardware:

- Use Oracle Database Appliance OLTP Database Shapes if your database workload is primarily online transaction processing (OLTP).
- Use Oracle Database Appliance DSS database shapes if your database workload is primarily decision support services (DSS) or data warehousing.
- Use Oracle Database Appliance In-Memory (IMDB) database shapes if your database workload can fit in memory, and can benefit from in-memory performance capabilities.

Use the database shape tables to help select the best shapes for your databases. When using these tables remember the following:

- The information in the tables assumes that you are creating local disk backups. Consider the space requirements for your database and the policy for local disk backups versus external backups. Typically, external backups have more space available for the database than local backups.
- The log file size assumes three (3) REDO log groups for each instance with a log switch every 15 minutes when the system is running at full capacity.

Database Shapes for the OLTP Class

Use Oracle Database Appliance OLTP Database Shapes if your database workload is primarily online transaction processing (OLTP).

The tables list the online transaction processing (OLTP) database shape sizes for Oracle Database Appliance X9-2S, X9-2L, and X9-2-HA.

The information in the table assumes the following:

- You are creating local disk backups. Consider the space requirements for your database and the policy for local disk backups versus external backups. Typically, external backups have more space available for the database than local backups.

Table D-1 Oracle Database Appliance X9-2S OLTP Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb1s	1	2	1	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb1	1	4	2	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb2	2	8	4	400	4	32	16
odb4	4	16	8	800	4	64	32
odb6	6	24	12	1200	8	256	48
odb08	8	32	16	1600	8	256	64
odb10	10	40	20	2000	8	256	80
odb12	12	48	24	2400	16	256	96
odb16	16	64	32	3200	16	256	128

Table D-2 Oracle Database Appliance X9-2-HA and X9-2L OLTP Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb1s	1	2	1	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb1	1	4	2	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb2	2	8	4	400	4	32	16
odb4	4	16	8	800	4	64	32
odb6	6	24	12	1200	8	256	48
odb08	8	32	16	1600	8	256	64

Table D-2 (Cont.) Oracle Database Appliance X9-2-HA and X9-2L OLTP Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb10	10	40	20	2000	8	256	80
odb12	12	48	24	2400	16	256	96
odb16	16	64	32	3200	16	256	128
odb20	20	80	40	4000	16	256	160
odb24	24	96	48	4800	16	256	192
odb28	28	112	56	5600	16	256	224
odb32	32	128	64	6400	16	256	256

Database Shapes for the In-Memory Class

Use Oracle Database Appliance In-Memory (IMDB) database shapes if your database workload can fit in memory, and can benefit from in-memory performance capabilities.

The table lists the In-Memory database shape sizes for Oracle Database Appliance X9-2S, X9-2L, and X9-2-HA.

The information in the table assumes the following:

- You are creating local disk backups. Consider the space requirements for your database and the policy for local disk backups versus external backups. Typically, external backups have more space available for the database than local backups.

Table D-3 Oracle Database Appliance X9-2S In-Memory Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	In-Memory (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb1s	1	2	1	1	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb1	1	4	2	2	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb2	2	8	4	4	400	4	32	16
odb4	4	16	8	8	800	4	64	32
odb6	6	24	12	12	1200	8	256	48
odb08	8	32	16	16	1600	8	256	64
odb10	10	40	20	20	2000	8	256	80
odb12	12	48	24	24	2400	16	256	96

Table D-3 (Cont.) Oracle Database Appliance X9-2S In-Memory Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	In-Memory (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb16	16	64	32	32	3200	16	256	128

Table D-4 Oracle Database Appliance X9-2L and X9- 2-HA In-Memory Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	In-Memory (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb1s	1	2	1	1	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb1	1	4	2	2	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb2	2	8	4	4	400	4	32	16
odb4	4	16	8	8	800	4	64	32
odb6	6	24	12	12	1200	8	256	48
odb08	8	32	16	16	1600	8	256	64
odb10	10	40	20	20	2000	8	256	80
odb12	12	48	24	24	2400	16	256	96
odb16	16	64	32	32	3200	16	256	128
odb20	20	80	40	42	4000	16	256	160
odb24	24	96	48	48	4800	16	256	192
odb28	28	112	56	56	5600	16	256	224
odb32	32	128	64	64	6400	16	256	256

Database Shapes for the DSS Class

Use DSS database shapes if your database workload is primarily decision support services (DSS) or data warehousing.

The table lists the DSS database shape sizes for Oracle Database Appliance X9-2S, X9-2L, and X9-2-HA.

The information in the table assumes the following:

- You are creating local disk backups. Consider the space requirements for your database and the policy for local disk backups versus external backups. Typically, external backups have more space available for the database than local backups.

Table D-5 Oracle Database Appliance X9-2S DSS Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Redo log file size (GB)	Log buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb1s	1	1	2	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb1	1	2	4	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb2	2	4	8	400	4	32	16
odb4	4	8	16	800	4	64	32
odb6	6	12	24	1200	8	256	48
odb08	8	16	32	1600	8	256	64
odb10	10	20	40	2000	8	256	80
odb12	12	24	48	2400	16	256	96
odb16	16	32	64	3200	16	256	128

Table D-6 Oracle Database Appliance X9-2L and X9-2-HA DSS Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb1s	1	1	2	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb1	1	2	4	200	4	32	Not Applicable. odb-01s not supported for DB system.
odb2	2	4	8	400	4	32	16
odb4	4	8	16	800	4	64	32
odb6	6	12	24	1200	8	256	48
odb08	8	16	32	1600	8	256	64
odb10	10	20	40	2000	8	256	80
odb12	12	24	48	2400	16	256	96
odb16	16	32	64	3200	16	256	128

Table D-6 (Cont.) Oracle Database Appliance X9-2L and X9-2-HA DSS Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)	DB System Memory (GB) Applicable only for DB Systems
odb20)	20	40	80	4000	16	256	160
odb24	24	48	96	4800	16	256	192
odb28	28	56	112	5600	16	256	224
odb32	32	64	128	6400	16	256	256

Glossary

Index

Numerics

10GBase-T (copper) network ports, *3-4*, *3-6*
25GbE SFP28 (fiber) network ports, *3-4*, *3-6*

A

ACFS, *B-1*, *B-2*
ACFS mount points, *B-2*
ACR, *19-25*, *19-27*
administrative account users, *9-2*
administrative accounts, *9-2*
agent
    update, *16-81*–*16-83*
agent proxy setting, *12-17*
ASM, *B-1*
ASM Cluster File System
    See Oracle ACFS
ASR
    configure, *16-238*, *16-241*, *18-3*, *18-5*, *18-7*
    delete, *16-240*
    details, *16-240*
    documentation, *18-13*
    modify, *16-242*
    register, *18-3*, *18-5*, *18-7*
    test, *16-243*
Automatic Storage Management
    See Oracle ASM

B

back up database, *12-19*
backup and recovery
    CLI commands, *16-106*
    RMAN, *9-3*
backup databases, *12-1*
backup location
    disk group, *1-1*
backup policy, *12-22*, *12-23*
    update, *12-19*
backup to external FRA, *12-18*
backup to internal FRA, *12-18*
backup to NFS location, *12-18*
backup to Object Store, *12-18*
bare metal patching, *8-9*, *8-35*, *8-38*
bill of materials, *19-15*
bond
    network interface, *11-1*, *11-2*
bridge
    network interface, *11-1*
Browser User Interface, *C-1*
    about, *2-4*
    actions, *2-4*
BUI
    clone vdisk, *15-42*
    clone vm, *15-48*
    create CPU pool, *14-3*
    create dbsystem, *15-16*
    create vdisk, *15-40*
    create vm, *15-45*
    create vmstorage, *15-34*
    create vnetwork, *15-37*
    delete cpupool, *14-6*
    delete dbsystem, *15-28*
    delete vdisk, *15-44*
    delete vm, *15-51*
    delete vnetwork, *15-39*
    delete-vmstorage, *15-36*
    describe CPU pool, *14-5*
    describe dbsystem, *15-23*
    describe storage, *15-34*
    describe vdisk, *15-42*
    describe vm, *15-50*
    describe vnetwork, *15-38*
    list dbsystems, *15-22*
    modify cpupool, *14-6*
    modify dbsystem, *15-24*, *15-26*, *15-29*
    modify vdisk, *15-43*
    modify vm, *15-49*
    modify vmstorage, *15-35*
    modify vnetwork, *15-39*
    remap cpupools, *14-4*
    start dbsystem, *15-27*
    start vm, *15-51*
    start vnetwork, *15-38*
    stop dbsystem, *15-27*

C

cabling, *3-1*, *3-4*
    validate, *16-259*
cleanup, *8-42*

cleanup utility, *7-1*, *7-7*
cleanup.pl, *7-1*, *7-7*, *19-19*
CLI commands
  configure, *16-13*
cloning database, *9-87*
Configuration Audit Tool (ORAchk), *19-21*
configure CLI commands
  configure-firstnet, *16-13*
configure-firstnet, *4-4*
CPU
  commands, *16-145*
CPU core
  current configuration, *16-146*
  enable, *16-146*
CPU cores
  configuration history, *16-145*
  list, *16-145*
cpu pool
  creating, *14-3*
CPU pool
  audit record, *14-7*
  deleting, *14-6*
  modifying, *14-6*
  over-subscribed, *14-7*
  remapping, *14-4*
  viewing, *14-5*
cpu pools
  about, *14-1*
cpupool
  create, *16-149*, *17-30*
  delete, *16-152*, *17-33*
  describe, *16-151*, *17-32*
  modify, *16-152*, *17-33*
cpupools
  auditrecord, *16-153*, *16-154*, *17-34*, *17-35*
  list, *16-151*, *17-32*
  remap, *16-150*, *17-31*
create appliance, *4-9*

D

DATA disk group, *B-3*
database, *1-1*, *9-101*, *D-2*, *D-4*, *D-5*
  backup policy, *12-19*
  block size, *1-1*
  clone from backup, *9-86*
  create, *9-81*
  delete, *9-100*, *9-102*
  details, *9-81*
  home, *9-101*
    create multiple, *9-101*
    multiple, *9-101*
  language, *1-1*
  list, *9-81*
  modify, *9-90*
  move, *9-91*
database *(continued)*
  register, *16-180*
  template
    See shape, database
  territory, *1-1*
  upgrade, *9-92*, *9-93*
database backup
  delete, *12-21*
database classes, *D-1*
database home
  acfs, *9-83*
  create, *9-102*
  details, *9-101*
  display details, *16-190*
  list, *9-101*
database home storage
  configure, *16-186*
  describe, *16-188*
  list, *16-187*
  modify, *16-189*
database homes
  ACFS, *9-3*
Database Security Assessment report, *19-22*
database shapes, *D-1*, *D-2*, *D-4*, *D-5*
  about, *D-1*
database storage, *B-1*
  commands, *16-193*
  create, *16-193*
  display details, *16-195*
  list, *16-199*
database, register, *9-106*, *9-107*
db system
  modifying, *15-24*
DB system
  backup, restore and recovery, *15-29*
  creating, *15-16*
  delete, *15-28*
  describing, *15-23*
  modifying, *15-29*
  network, *15-26*
  Oracle Data Guard, *15-32*
  start, *15-27*
  stop, *15-27*
db system patching, *8-25*
DB systems
  list, *15-22*
dbsystem
  create, *17-6*
  delete, *17-21*
  describe, *17-11*, *17-18*
  modify, *17-7*, *17-10*
  start, *17-19*
  stop, *17-20*
dbsystems
  list, *17-11*
dcserr, *19-5*

default groups and users, *A-5*
deleting backups, *12-31*
DHCP (Dynamic Host Configuration Protocol)
  connect to Oracle ILOM, *3-9*
  Oracle ILOM configuration, *3-9*
diagnostics, *19-6*
directory paths, *A-1*
disk
  replace, *10-3*, *10-5*
disk group redundancy, *B-1*
disk group sizes, *A-5*
disk group storage
  describe, *16-197*
  list, *16-199*
domain
  definition, *15-4*
downloading software, *4-6*
DSS
  database shape, *D-5*
dynamic default parameters
  using Browser User Interface, *8-43*
Dynamic Host Configuration Protocol
  See DHCP

E

electrical power cords
  connecting, *3-9*
EM Express, *9-113*
Enterprise Edition High Availability, *9-6*
error correlation report, *19-2*
external FRA, *12-1*

F

flex diskgroup, *9-81*, *9-83*, *9-90*

G

Grid user, *9-2*
groups and users defaults, *A-5*
guest virtual machine
  definition, *15-4*

H

Host Public Addresses
  IP address, *1-8*

I

IMDB
  database shape, *D-4*
install software, *4-7*
installed version
  display, *16-49*, *16-52*, *16-62*, *16-63*, *16-65*
instance caging, *9-113*
  enable, *9-113*
Integrated Lights Out Manager
  See Oracle ILOM
internal FRA, *12-1*

J

JSON file, *5-12*, *5-17*, *5-21*, *5-25*, *15-52*, *15-62*, *C-1*, *C-2*, *C-7*

K

key vault
  about, *9-9*
KVM, *17-1*, *17-3*
  definition, *15-3*
  deploying, *15-33*
  overcomit, *15-52*
KVM commands
  odacli clone-vdisk, *17-28*
  odacli clone-vm, *17-47*
  odacli create-cpupool, *16-149*, *17-30*
  odacli create-dbsystem, *17-6*
  odacli create-vdisk, *17-26*
  odacli create-vm, *17-40*
  odacli create-vmstorage, *17-23*
  odacli create-vnetwork, *17-36*
  odacli delete-cpupool, *16-152*, *17-33*
  odacli delete-dbsystem, *17-21*
  odacli delete-vdisk, *17-28*
  odacli delete-vm, *17-47*
  odacli delete-vmstorage, *17-25*
  odacli delete-vnetwork, *17-39*
  odacli describe-auditrecord, *16-154*, *17-35*
  odacli describe-cpupool, *16-151*, *17-32*
  odacli describe-dbsystem, *17-11*
  odacli describe-dbsystem-image, *17-18*
  odacli describe-vdisk, *17-27*
  odacli describe-vm, *17-42*
  odacli describe-vmstorage, *17-24*
  odacli describe-vnetwork, *17-37*
  odacli list-auditrecords, *16-153*, *17-34*
  odacli list-cpupools, *16-151*, *17-32*
  odacli list-dbsystems, *17-11*
  odacli list-vdisks, *17-26*
  odacli list-vms, *17-42*
  odacli list-vmstorages, *17-23*
  odacli list-vnetworks, *17-37*
  odacli migrate-vm, *17-46*
  odacli modify-cpupool, *16-152*, *17-33*
  odacli modify-dbfileattributes, *17-10*
  odacli modify-dbsystem, *17-7*
  odacli modify-vdisk, *17-27*

KVM commands *(continued)*
odacli modify-vm, *17-43*
odacli modify-vmstorage, *17-24*
odacli modify-vnetwork, *17-39*
odacli remap-cpupools, *16-150*, *17-31*
odacli start-dbsystem, *17-19*
odacli start-vm, *17-45*
odacli start-vnetwork, *17-38*
odacli stop-dbsystem, *17-20*
odacli stop-vm, *17-46*
odacli stop-vnetwork, *17-38*

L

LED
status when blinking, *3-11*
status when steady, *3-11*
status when steady on, *3-11*
LED indicators
NVMe, *10-3*, *10-5*
Linux
and virtualization, *15-2*, *15-7*, *15-10*
linux kernel core extractor, *19-3*
listener.ora, configure, *9-104*
load configuration, *C-1*
log files, *A-4*
sending to Oracle Support Services, *19-32*
logs
storage diagnostic, *16-95*, *16-230*

M

migration
from an existing database, *9-3*
multi-node system, *10-6*
multi-user access
appliance, *5-2*, *5-7*
create user, *5-10*, *5-32*, *9-50*
entitlements, *5-35*
manage database, *5-36*
manage database homes, *5-36*
manage password, *5-11*, *5-38*
odacli commands, *16-6*
passwordless, *5-7*
provisioning, *5-9*, *5-29*
reset-password, *5-33*, *5-38*
resources, *5-34*
roles, *5-35*
shared resource, *5-5*
user roles, *5-4*
multiple database instances
instance caging, *9-113*
multiple Oracle homes, *9-101*
create, *9-101*
My Oracle Support, *19-32*
send log files for troubleshooting, *19-32*

N

network
configure, *16-13*
create, *11-4*, *11-5*
delete, *11-6*, *11-7*
interfaces, *11-4*
list, *11-4*
update, *11-5*
network cabling, *3-6*
network infrastructure, *11-1*, *11-2*
network interface
add, *11-6*
default values, *11-1*, *11-2*
delete, *11-6*
network ports, *3-4*, *3-6*
network time protocol service (NTP service), *1-1*
NFS backup, *12-9*, *12-12*, *12-15*
NTP service
See network time protocol service
NVMe
disk location, *10-3*, *10-5*
LED indicators, *10-3*, *10-5*

O

oakcli commands
show power
example, *19-31*
Object Store, *12-17*
credentials, *16-111*, *16-116*, *16-117*, *16-125*, *16-133*
odaadmcli commands
asmops diskgroup, *16-256*
expand storage, *16-257*
hardware configuration, *16-249*
hardware monitoring, *16-249*
manage diagcollect, *16-230*, *19-32*
orachk, *16-95*
power disk, *16-259*
show controller, *16-260*
show cooling, *16-250*
show disk, *16-261*
show diskgroup, *16-262*
show env_hw, *16-250*
show fs, *16-251*
show localdisk, *16-249*
show memory, *16-252*
show network, *16-252*
show power, *16-253*
show processor, *16-254*
show raidsyncstatus, *16-263*
show server, *16-254*
show storage, *16-264*
storage, *16-255*
stordiag, *16-265*

odacli cleanup-patchrepo, *8-42*
odacli commands
- acr, *16-244*
- appliance, *16-96*
- apply patch, *16-46*
- backup, *16-106*
- cleanup-patchrepo, *16-48*
- clone-database, *16-155*
- clone-vdisk, *15-42*
- clone-vm, *15-48*
- configure-asr, *16-238*
- configure-dataguard, *16-135*
- configure-dbhome-storage, *16-186*
- CPU core, *16-145*
- cpu pool, *16-148*, *17-29*
- create database, *9-101*
- create vdisk, *15-40*
- create-appliance, *16-99*
- create-auto-logclean-policy, *16-231*
- create-backup, *16-108*
- create-backupconfig, *16-109*
- create-cpupool, *14-3*
- create-database, *16-165*
- create-dbhome, *16-190*
- create-dbstorage, *16-193*
- create-dbsystem, *15-16*
- create-diagcollection, *16-203*
- create-jobdata-retention-policy, *16-207*
- create-logcleanjob, *16-233*
- create-network, *16-16*
- create-objectstoreswift, *16-111*
- create-okvserverconfig, *16-223*
- create-prepatchreport, *16-49*
- create-purge-jobdata-job, *16-208*
- create-vm, *15-45*
- create-vmstorage, *15-34*
- create-vnetwork, *15-37*
- database, *16-154*
- database storage, *16-193*
- dataguard, *16-135*
- DBHome, *16-185*
- dbsystem, *17-5*
- deconfigure-dataguard, *16-140*
- delete-asr, *16-240*
- delete-backup, *16-112*
- delete-backupconfig, *16-113*
- delete-cpupool, *14-6*
- delete-database, *16-172*
- delete-dbhome, *16-190*
- delete-dbstorage, *16-195*
- delete-dbsystem, *15-28*
- delete-diagcollection, *16-205*
- delete-jobdata-retention-policy, *16-212*
- delete-network, *16-18*
- delete-networkinterface, *16-19*
- delete-objectstoreswift, *16-117*

odacli commands *(continued)*
- delete-okvserverconfig, *16-226*
- delete-prepatchreport, *16-63*
- delete-vdisk, *15-44*
- delete-vm, *15-51*
- delete-vmstorage, *15-36*
- delete-vnetwork, *15-39*
- describe-asr, *16-240*
- describe-backupconfig, *16-112*
- describe-backupreport, *16-114*
- describe-component, *16-65*
- describe-cpucore, *16-146*
- describe-cpupool, *14-5*
- describe-database, *16-173*
- describe-dataguardstatus, *16-141*
- describe-dbhome, *16-191*
- describe-dbhome-storage, *16-188*
- describe-dbstorage, *16-195*
- describe-dbsystem, *15-23*
- describe-dgstorage, *16-197*
- describe-diagcollection, *16-204*
- describe-job, *16-213*
- describe-latestpatch, *16-71*
- describe-logcleanjob, *16-234*
- describe-network, *16-20*
- describe-networkinterface, *16-22*
- describe-objectstoreswift, *16-116*
- describe-okvserverconfig, *16-227*
- describe-prepatchreport, *16-52*
- describe-schedule, *16-118*
- describe-system, *16-100*
- describe-vdisk, *15-42*
- describe-vm, *15-50*
- describe-vmstorage, *15-34*
- describe-vnetwork, *15-38*
- diagnostics, *16-203*
- export-asrconfig, *16-241*
- failover-dataguard, *16-142*
- irestore-database, *16-118*
- jobs, *16-206*
- key vault, *16-222*
- list scheduler jobs, *16-221*
- list-auto-logclean-policy, *16-235*
- list-backupconfig, *16-124*
- list-backupreports, *16-125*
- list-cpucores, *16-145*
- list-cpupools, *14-5*
- list-databases, *16-175*
- list-dataguardstatus, *16-143*
- list-dbhome-storage, *16-187*
- list-dbhomes, *16-192*
- list-dbstorages, *16-199*
- list-dbsystems, *15-22*
- list-dgdisks, *16-257*
- list-diagcollections, *16-205*
- list-featuretracking, *16-103*

odacli commands *(continued)*
list-jobdata-retention-policy, *16-219*
list-jobs, *16-215*
list-logcleanjobs, *16-236*
list-logspaceusage, *16-236*
list-networkinterfaces, *16-23*
list-networks, *16-23*
list-okvserverconfigs, *16-228*
list-osconfigurations, *16-247*
list-prepatchreports, *16-62*
list-purge-jobdata-jobs, *16-220*
list-scheduled-executions, *16-221*
list-schedules, *16-126*
list-tdebackupreports, *16-127*
list-vdisks, *15-42*
list-vms, *15-50*
list-vmstorages, *15-34*
list-vnetworks, *15-38*
modify-asr, *16-242*
modify-backupconfig, *16-132*
modify-cpupool, *14-6*, *15-29*
modify-database, *16-175*
modify-dbhome-storage, *16-189*
modify-dbstorage, *16-200*
modify-dbsystem, *15-24*, *15-26*
modify-dgstorage, *16-202*
modify-network, *16-24*
modify-objectstoreswift, *16-133*
modify-osconfigurations, *16-248*
modify-schedule, *16-134*
modify-vdisk, *15-43*
modify-vm, *15-49*
modify-vmstorage, *15-35*
modify-vnetwork, *15-39*
move-database, *16-179*
multi-user access, *16-26*
network, *16-13*
odacli activate-user, *16-31*
odacli authorize-user, *16-34*
odacli change-password, *16-46*
odacli copy-okvtdewallet, *16-223*
odacli create-appliance, *5-12*, *5-17*, *C-1*, *C-2*, *C-7*
odacli create-dbsystem, *5-21*, *5-25*, *15-52*, *15-62*
odacli create-user, *16-30*
odacli delete-agentconfig-parameters, *16-76*
odacli delete-auto-logclean-policy, *16-232*
odacli delete-user, *16-35*
odacli describe-acr, *16-244*
odacli describe-domain, *16-28*
odacli describe-resource-access, *16-42*
odacli describe-tenant, *16-29*
odacli commands *(continued)*
odacli describe-user, *16-33*
odacli describe-user-entitlement, *16-39*
odacli describe-user-operation, *16-40*
odacli describe-user-role, *16-37*
odacli disable-acr, *16-246*
odacli enable-acr, *16-245*
odacli grant-resource-access, *16-43*
odacli list-acr, *16-246*
odacli list-agentconfig-parameters, *16-72*
odacli list-availablepatches, *16-63*
odacli list-dgstorages, *16-199*
odacli list-domains, *16-27*
odacli list-nodes, *16-103*
odacli list-objectstoreswifts, *16-125*
odacli list-pendingjobs, *16-219*
odacli list-resources-access, *16-41*
odacli list-tenants, *16-29*
odacli list-user-entitlements, *16-38*
odacli list-user-operations, *16-40*
odacli list-user-roles, *16-36*
odacli list-users, *16-32*
odacli modify-agentconfig-parameters, *16-74*
odacli modify-asmport, *16-96*
odacli modify-cpucore, *16-146*
odacli ping-agent, *16-102*
odacli redact-acr, *16-245*
odacli reset-password, *16-45*
odacli revoke-resource-access, *16-44*
Oracle ASR, *16-237*
OS, *16-247*
recover-database, *16-128*
recovery, *16-106*
register-database, *16-180*
register-dataguard, *16-144*
reinstate-dataguard, *16-143*
remap-cpupools, *14-4*
restore-archivelog, *16-130*
restore-tdewallet, *16-130*
set-credential, *16-99*
show -h
with Oracle Database Appliance Hardware Monitoring Tool, *19-31*
start-dbsystem, *15-27*
start-vm, *15-51*
start-vnetwork, *15-38*
stop-dbsystem, *15-27*
stop-vm, *15-51*
stop-vnetwork, *15-38*
storage, *16-255*
switchover-dataguard, *16-144*
test-asr, *16-243*

odacli commands *(continued)*
update, *16-46*
update database, *16-78*
update database home, *16-79*
update repository, *16-91*
update-dcsadmin, *16-81*
update-dcsagent, *16-81*
update-dcscomponents, *16-82*
update-registry, *16-83*
update-server, *16-92*
update-server-postcheck, *16-95*
update-storage, *16-94*
upgrade-database, *16-183*
validate storage topology, *16-259*
validate-storagetopology, *16-259*
vdisk, *17-25*
vm, *17-40*
vmstorage, *17-22*
vnetwork, *17-35*
odacli create-appliance
example JSON files, *5-12*, *5-17*, *C-1*, *C-2*, *C-7*
readme, *C-1*
odacli create-dbsystem
example JSON files, *5-21*, *5-25*, *15-52*, *15-62*
odaeraser.py, *7-6*
OINSTALL group, *9-2*
OLTP
database shape, *D-2*
Optimal Flexible Architecture
and Oracle homes, *A-1*
ORAchk report, *19-22*
Oracle ACFS, *15-4*
Oracle ACFS (Oracle Automatic Storage Management Cluster File System)
virtual storage, *15-4*
Oracle ASM, *10-1*, *A-5*
Oracle ASM (Oracle Automatic Storage Management)
data migration, *9-3*
user group, *A-5*
Oracle ASR
commands, *16-237*
configure, *16-238*, *16-241*, *18-3*, *18-5*, *18-7*
register, *18-3*, *18-5*, *18-7*
Oracle ASR (Oracle Auto Service Request), *1-1*
configuring, *18-1*
Oracle Data Guard, *13-2*
active data guard, *13-23*
CLI commands, *16-135*
configure, *13-2*
deconfigure, *13-28*
failover, *13-12*
migrating, *13-15*
network, *13-14*
patching, *13-20*
register, *13-23*
Oracle Data Guard *(continued)*
reinstate, *13-12*
status, *13-12*
switchover, *13-12*
upgrade database, *13-22*
Oracle Database
See database
Oracle Database Appliance Hardware Monitoring Tool, *19-31*
Oracle Database Appliance Manager
software inventory, *2-3*
Oracle Database Resource Manager
instance caging, *9-113*
Oracle Enterprise Manager Database Express, *9-113*
Oracle home
multiple, *9-101*
Oracle ILOM (Oracle Integrated Lights Out Manager), *1-1*
configure, *3-9*
Oracle Integrated Lights Out Manager
See Oracle ILOM
oracle key vault
create database, *9-12*, *9-16*, *9-50*
manage database, *9-24*, *9-25*, *9-30*, *9-38*, *9-48*
manage server configuration, *9-22*, *9-51*
migrate acfs cdb database, *9-72*
migrate acfs database, *9-52*
migrate asm database, *9-58*, *9-65*
Oracle Ksplice, *8-39*
Oracle KVM, *15-2*, *15-7*, *15-10*
Oracle KVM command-line interface
alphabetical list of commands, *17-1*
Oracle ORAchk, *19-29*
Oracle ORAchk Health Check Tool, *19-21*
Oracle Support Services
See My Oracle Support
Oracle Trace File Analyzer Collection, *19-28*
Oracle user, *9-2*
OS configuration
current, *16-247*, *16-248*
suggested, *16-247*, *16-248*
OSASM group, *9-2*
OSDBA, *9-2*
OSDBA for ASM group, *9-2*
OSDBA group, *9-2*
OSOPER groups, *9-2*
over-subscribed CPU pool, *14-7*

P

patches
software inventory, *2-3*
patching
appliance, *8-2*, *8-7*

patching *(continued)*
using Browser User Interface, *8-31*, *8-33*, *8-36*, *8-37*
patching Oracle Database Appliance, *8-9*, *8-25*, *8-35*, *8-38*
performance tuning, *14-7*
plumb network, *4-4*
port numbers, *A-2*
power cords
connecting, *3-9*
processor_group_name, *14-1*
protocols, *A-2*
proxy setting, *12-17*

R

readme, *C-1*
recover database, *12-26*
recovering database, *12-20*
register
Oracle ASR, *18-3*, *18-5*, *18-7*
reimage appliance, *7-4*
resource management
CPU pools, *14-7*
Oracle KVM, *15-2*, *15-7*, *15-10*
supported virtual operating systems, *15-2*, *15-7*, *15-10*
restore archivelog, *12-31*
restore baremetal system, *7-4*
restore database, *12-28*
RMAN (Recovery Manager utility)
database management, *9-3*
RPM drift, *19-15*

S

sanitize diagnostics, *19-25*, *19-27*
sanitize entities, *19-28*
save configuration, *C-1*
scheduler
jobs, *16-221*
Scheduling Database Backups, *12-25*
secure erase tool, *7-6*
odaeraser.py, *7-6*
sending log files to Oracle Support Services, *19-32*
shapes, *D-1*
database
choosing, *D-1*
OLTP, *D-2*, *D-4*, *D-5*
SI (Support Identifier)
Oracle ASR requirement, *18-1*
space management, *B-2*
SQL*Loader, *9-3*
ssh configuration, *19-2*
Standard Edition High Availability, *9-5*
start up the system, *3-11*
static listener, configure, *9-104*
storage, *10-1*, *10-6*, *B-3*
expand, *10-3*, *10-5*
single-node, *10-2*
storage cables, *3-1*
storage disks
location, *10-3*, *10-5*

T

TDE, *9-7*
TFA command, *19-23*
TFA options, *19-23*
Trace File Analyzer (TFA) Collector, *19-23*
trace file redaction, *19-28*
troubleshooting, *19-2*, *19-3*, *19-5*, *19-6*, *19-19*
log files for Oracle Support Services, *19-32*
turning on Oracle Database Appliance, *3-11*

U

uninstall, *7-1*, *7-7*
update
database, *8-8*
update-dcscomponents, *19-6*
upgrade
database, *8-42*
usable storage, *B-3*

V

viewing backup reports, *12-20*
virtual disk
clone, *17-28*
cloning, *15-42*
create, *17-26*
creating, *15-40*
delete, *17-28*
deleting, *15-44*
describe, *17-27*
list, *17-26*
modify, *17-27*
modifying, *15-43*
viewing, *15-42*
virtual disks, *15-4*
virtual local area network, *11-1*, *11-2*
virtual machine
about, *15-4*
assemblies, *15-4*
clone, *17-47*
cloning, *15-48*
create, *17-40*
creating, *15-45*
delete, *17-47*
deleting, *15-51*

virtual machine *(continued)*
describe, *17-42*
list, *17-42*
migrate, *17-46*
modify, *17-43*
modifying, *15-49*
start, *15-51*, *17-45*
stop, *15-51*, *17-46*
templates, *15-4*
viewing, *15-50*
virtual network
creating, *15-37*
deleting, *15-39*
modifying, *15-39*
start, *15-38*
stop, *15-38*
viewing, *15-38*
virtual storage
about, *15-4*
virtualization, *15-2*, *15-7*, *15-10*
VLAN, *11-1*
vm storage
creating, *15-34*
deleting, *15-36*
modifying, *15-35*
vm storage *(continued)*
viewing, *15-34*
vmstorage
create, *17-23*
delete, *17-25*
describe, *17-24*
modify, *17-24*
vmstorages
list, *17-23*
vnetwork
create, *17-36*
delete, *17-39*
describe, *17-37*
modify, *17-39*
start, *17-38*
stop, *17-38*
vnetworks
list, *17-37*

W

web interface
Oracle Enterprise Manager Database Express, *9-113*