Oracle® Database Appliance

X8-2 Deployment and User's Guide

Release 19.14 for Linux x86-64
F51838-01
February 2022

Oracle Database Appliance X8-2 Deployment and User’s Guide, Release 19.14 for Linux x86-64

F51838-01

Copyright © 2019, 2022, 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 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" or "commercial computer software documentation" 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, and MySQL 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 xvii
Documentation Accessibility xvii
Related Documents xviii
Conventions xviii

1 Oracle Database Appliance Checklists

Checklist for System Requirements 1-1
Deployment Checklist for Oracle Database Appliance Bare Metal Systems 1-6
Checklist for Patching Oracle Database Appliance Bare Metal Deployments 1-8
Checklist for Custom Network Address Configuration 1-11

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 X8-2-HA 3-1
Attaching Network Cables to Oracle Database Appliance X8-2-HA 3-4
Attaching Network Cables to Oracle Database Appliance X8-2S or X8-2M 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

Plumbing the Network 4-1
Verifying Cabling and Network Connections 4-3
Downloading Oracle Database Appliance Software 4-5

Installing Oracle Database Appliance Software 4-6
Creating the Appliance 4-7

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 5-6
Provisioning Oracle Database Appliance with Multi-User Access 5-6
Creating, Viewing, and Deleting Users on Oracle Database Appliance with Multi-User Access 5-9
Activating a New User on Oracle Database Appliance with Multi-User Access 5-10
Granting and Revoking Resource Access on Oracle Database Appliance with Multi-User Access 5-11
Viewing Roles, Operations, and Entitlements on Oracle Database Appliance with Multi-User Access 5-12
Managing Databases and Database Homes on Oracle Database Appliance with Multi-User Access 5-13
Changing the Password for a User Account on Oracle Database Appliance with Multi-User Access 5-14
Resetting the Password for a Locked User Account on Oracle Database Appliance with Multi-User Access 5-15

6 Oracle Database Appliance Postinstallation Tasks

Configuring CPU Core Count 6-1
Securing Oracle ILOM Service Processors 6-2
Changing Oracle Database Appliance Passwords 6-2

7 Patching Oracle Database Appliance

About Patching Oracle Database Appliance 7-2
About Patching Pre-Checks 7-4
About Enabling NUMA on Oracle Database Appliance 7-6
Minimizing Downtime When Patching Oracle Database Appliance 7-7
Patching Oracle Database Appliance Bare Metal Systems Using the Command-Line 7-8
Patching Oracle Database Appliance DB Systems Using the Command-Line 7-16
Patching Oracle Database Appliance Bare Metal Systems Using the Browser User Interface 7-19
Patching Oracle Database Appliance DB Systems Using the Browser User Interface 7-21
Updating Oracle Database Appliance Repository with Database Clone Files For Bare Metal Systems 7-23

Patching Existing Database Homes on Bare Metal Systems Using the Browser User Interface 7-24
Patching Non-Provisioned Oracle Database Appliance Systems 7-25
Applying Additional Patches and Updates on Bare Metal Systems 7-26
Enabling Kernel Updates 7-26
Applying Out-of-Cycle Database Patches 7-27
Cleaning Up the Patch Repository 7-29
About Upgrading to a Different Database Home 7-29

8 Re-imaging Oracle Database Appliance

Uninstalling Oracle Database Appliance Components 8-1
Reimaging an Oracle Database Appliance Baremetal System 8-4
Performing Secure Erase of Data on Storage Disks 8-5

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 Transparent Database Encryption (TDE) in Oracle Database Appliance 9-6
Working with Databases 9-7
Viewing Databases 9-8
Creating a Database Using the Browser User Interface 9-9
Managing Database Homes on Oracle ACFS Storage 9-12
Creating a Database Using ODACLI Commands 9-13
Cloning a Database from Backup 9-14
Cloning an Oracle ACFS Database Using the Browser User Interface 9-16
Cloning an Oracle ACFS Database Using Command Line Interface 9-17
Modifying a Database 9-18
Moving a Database from One Oracle Home to Another 9-19
Upgrading a Database 9-20
Deleting a Database 9-20
Working with Database Homes 9-21
About Managing Multiple Oracle Homes on Oracle Database Appliance 9-21
Viewing Database Homes 9-22
Creating a Database Home 9-22
Deleting a Database Home 9-23
Migrating Databases 9-23
About Migrating Databases 9-23
Configuring a Static Listener 9-24

Migrating a Database 9-25
Registering a Database 9-27
About Registering a TDE-Enabled Database 9-28
About Managing Multiple Database Instances Using Instance Caging 9-33
Oracle EM Express and DB Console 9-34

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 X8-2S and X8-2M 10-2
Adding NVMe Storage Disks 10-3
Managing Storage on High-Availability Systems 10-5
About Storage Options for Oracle Database Appliance X8-2-HA 10-5
Preparing for a Storage Upgrade for a Virtualized Platform 10-7
Adding Solid-State Drives (SSDs) for Data Storage 10-9
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 X8-2 Network Interfaces 11-2
Viewing Configured Networks and Network Interfaces 11-5
Creating a Network 11-5
Creating a Physical Network 11-5
Updating a Network 11-6
Adding a Network Interface 11-6
Deleting a Network Interface 11-7
Deleting a Network Interface Using Browser User Interface 11-7
Deleting a Network Using the Browser User Interface 11-7

12 Backup, Restore and Recover Databases

About Database Backup and Recovery Options 12-1
Creating a Mount Point for NFS Backup Location 12-9
Configuring Agent Proxy Settings for Object Store Access 12-10
Creating a Database Backup Policy 12-11
Updating a Database with a Backup Policy 12-12
Updating a Backup Policy 12-13
Backing Up a Database Using the Browser User Interface 12-13
Viewing Backup Reports 12-14

Recovering a Database Using the Browser User Interface 12-14
Deleting a Backup Using the Browser User Interface 12-14
Deleting a Backup Policy 12-15
Using the CLI to Backup, Restore, and Recover 12-15
Preparing for Database Backup to Disk 12-16
Preparing for Database Backup to NFS Location 12-17
Preparing for Database Backup to Oracle Cloud 12-18
Backing Up a Database with the CLI 12-19
Recovering a Database with the CLI 12-20
Restoring a Database with the CLI 12-22
Restoring Archivelog with the CLI 12-25
Deleting Backups with the CLI 12-25

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

About Oracle Data Guard on Oracle Database Appliance 13-1
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 13-21
Enabling Oracle Active Data Guard on Existing Configurations 13-22
Deconfiguring Oracle Data Guard on Oracle Database Appliance 13-22

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 14-5
Viewing CPU Pools in a Bare Metal or KVM Deployment 14-6
Modifying a CPU Pool in a Bare Metal or KVM Deployment 14-6
Deleting a CPU Pool in a Bare Metal or KVM Deployment 14-7
Viewing CPU Pool Audit Records in a Bare Metal or KVM Deployment 14-8
About Over-Subscribed CPU Pools 14-8

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 Database Systems on Oracle Database Appliance 15-7
Managing Database Systems in KVM Deployment 15-9
- Creating a Database System in a KVM Deployment 15-10
- Listing Database Systems in a KVM Deployment 15-15
- Describing a Database System in a KVM Deployment 15-15
- Modifying a Database System in a KVM Deployment 15-16
- Attaching and Detaching a Network for a Database System in a KVM Deployment 15-18
- Starting a Database System in a KVM Deployment 15-19
- Stopping a Database System in a KVM Deployment 15-19
- Deleting a Database System in a KVM Deployment 15-20
- Managing Shared CPU Pool with Database System in a KVM Deployment 15-21
- Managing Backup, Restore, and Recovery on a Database System in a KVM Deployment 15-24
- Managing Oracle Data Guard on a Database System in a KVM Deployment 15-25

Setting Up and Configuring a KVM Deployment for Applications 15-25
Managing VM Storage in KVM Deployment 15-26
- Creating a VM Storage in a KVM Deployment 15-26
- Viewing VM Storage in a KVM Deployment 15-27
- Modifying VM Storage in a KVM Deployment 15-27
- Deleting a VM Storage in a KVM Deployment 15-28

Managing Virtual Networks in KVM Deployment 15-28
- Creating a Virtual Network in a KVM Deployment 15-29
- Viewing Virtual Networks in a KVM Deployment 15-30
- Starting and Stopping Virtual Networks in a KVM Deployment 15-31
- Modifying a Virtual Network in a KVM Deployment 15-31
- Deleting a Virtual Network in a KVM Deployment 15-32

Managing Virtual Disks in KVM Deployment 15-32
- Creating a Virtual Disk in a KVM Deployment 15-33
- Viewing Virtual Disks in a KVM Deployment 15-34
- Cloning a Virtual Disk in a KVM Deployment 15-34
- Modifying a Virtual Disk in a KVM Deployment 15-35
- Deleting a Virtual Disk in a KVM Deployment 15-36

Managing Virtual Machines in KVM Deployment 15-36
- Creating a Virtual Machine in a KVM Deployment 15-37
- Cloning a Virtual Machine in a KVM Deployment 15-41
- Modifying a Virtual Machine in a KVM Deployment 15-42
- Viewing Virtual Machines in a KVM Deployment 15-43

Starting and Stopping Virtual Machines in a KVM Deployment 15-43
Deleting a Virtual Machine in a KVM Deployment 15-44
About Overcommitting Memory or CPUs in an Oracle Database Appliance KVM System 15-45
Example JSON File to Create a Single-Node DB System 15-45
Example JSON File to Create a High-Availability DB System 15-48

16 Oracle Database Appliance Command-Line Interface

About Oracle Database Appliance Command-Line Interface 16-2
Managing ODACLI Privileges and Security with SUDO 16-5
ODACLI Command Changes with Multi-User Access on Oracle Database Appliance 16-7
odacli Network Commands 16-13
odacli configure-firstnet 16-14
odacli configure-pmem 16-16
odacli create-network 16-17
odacli delete-network 16-18
odacli delete-networkinterface 16-20
odacli describe-network 16-21
odacli describe-networkinterface 16-22
odacli list-networks 16-23
odacli list-networkinterfaces 16-24
odacli update-network 16-25
odacli Multi-User Access Commands 16-26
odacli list-domains 16-28
odacli describe-domain 16-28
odacli list-tenants 16-29
odacli describe-tenant 16-30
odacli create-user 16-31
odacli activate-user 16-32
odacli list-users 16-33
odacli describe-user 16-34
odacli authorize-user 16-35
odacli delete-user 16-35
odacli list-user-roles 16-37
odacli describe-user-role 16-38
odacli list-user-entitlements 16-38
odacli describe-user-entitlement 16-40
odacli list-user-operations 16-41
odacli describe-user-operation 16-41
odacli list-resources-access 16-42
odacli describe-resource-access 16-43

odacli grant-resource-access 16-44
odacli revoke-resource-access 16-45
odacli reset-password 16-46
odacli change-password 16-47
odacli Apply Patch and Update Commands 16-48
odacli cleanup-patchrepo 16-49
odacli create-prepatchreport 16-50
odacli describe-prepatchreport 16-52
odacli list-prepatchreports 16-57
odacli delete-prepatchreport 16-58
odacli list-availablepatches 16-59
odacli describe-component 16-60
odacli describe-latestpatch 16-67
odacli list-agentconfig-parameters 16-68
odacli update-agentconfig-parameters 16-70
odacli delete-agentconfig-parameters 16-73
odacli update-dbhome 16-74
odacli update-dcsadmin 16-75
odacli update-dcsagent 16-76
odacli update-dcscomponents 16-77
odacli update-registry 16-78
odacli update-repository 16-80
odacli update-server 16-81
odacli update-storage 16-83
odaadmcli orachk 16-83
odacli Appliance Commands 16-84
odacli set-credential 16-85
odacli create-appliance 16-85
odacli describe-system 16-86
odacli list-featuretracking 16-89
odacli Backup and Recovery Commands 16-91
odacli create-backup 16-93
odacli create-backupconfig 16-95
odacli create-objectstoreswift 16-96
odacli delete-backup 16-97
odacli delete-backupconfig 16-98
odacli describe-backupreport 16-99
odacli delete-objectstoreswift 16-100
odacli describe-schedule 16-101
odacli irestore-database 16-102
odacli list-backupconfigs 16-105

odacli list-backupreports 16-106
odacli list-objectstoreswifts 16-106
odacli list-schedules 16-107
odacli list-tdebackupreports 16-108
odacli recover-database 16-109
odacli restore-archivelog 16-111
odacli restore-tdewallet 16-112
odacli update-backupconfig 16-113
odacli update-objectstoreswift 16-114
odacli update-schedule 16-115
odacli Oracle Data Guard Commands 16-116
odacli configure-dataguard 16-116
odacli deconfigure-dataguard 16-121
odacli describe-dataguardstatus 16-123
odacli failover-dataguard 16-123
odacli list-dataguardstatus 16-124
odacli reinstate-dataguard 16-125
odacli switchover-dataguard 16-125
odacli CPU Core Commands 16-126
odacli list-cpucores 16-126
odacli describe-cpucore 16-127
update-cpucore 16-127
odacli CPU Pool Commands for Bare Metal and KVM 16-129
odacli create-cpupool 16-130
odacli remap-cpupools 16-131
odacli list-cpupools 16-132
odacli describe-cpupool 16-133
odacli modify-cpupool 16-133
odacli delete-cpupool 16-134
odacli list-auditrecords 16-134
odacli describe-auditrecord 16-135
odacli Database Commands 16-136
odacli clone-database 16-137
odacli create-database 16-139
odacli delete-database 16-143
odacli describe-database 16-145
odacli list-databases 16-146
odacli modify-database 16-147
odacli move-database 16-151
odacli register-database 16-152
odacli upgrade-database 16-155

odacli DBHome Commands 16-157
odacli configure-dbhome-storage 16-157
odacli list-dbhome-storages 16-158
odacli describe-dbhome-storage 16-160
odacli modify-dbhome-storage 16-161
odacli create-dbhome 16-161
odacli delete-dbhome 16-162
odacli describe-dbhome 16-163
odacli list-dbhomes 16-164
odacli Database Storage Commands 16-164
odacli create-dbstorage 16-165
odacli delete-dbstorage 16-166
odacli describe-dbstorage 16-167
odacli describe-dgstorage 16-168
odacli list-dbstorages 16-171
odacli list-dgstorages 16-171
odacli modify-dbstorage 16-172
odacli update-dgstorage 16-174
odacli Job Commands 16-175
odacli create-jobdata-retention-policy 16-175
odacli create-purge-jobdata-job 16-177
odacli delete-jobdata-retention-policy 16-181
odacli describe-job 16-182
odacli list-jobs 16-182
odacli list-jobdata-retention-policy 16-186
odacli list-purge-jobdata-job 16-187
odacli list-scheduled-executions 16-188
Log Commands 16-189
odaadmcli manage diagcollect 16-189
odacli create-auto-logclean-policy 16-191
odacli create-logcleanjob 16-192
odacli describe-logcleanjob 16-194
odacli list-auto-logclean-policy 16-194
odacli list-logcleanjobs 16-195
odacli list-logspaceusage 16-196
odacli Oracle Auto Service Request Commands 16-197
odacli configure-asr 16-197
odacli delete-asr 16-199
odacli describe-asr 16-200
odacli test-asr 16-200
odacli update-asr 16-201

odacli OS Commands 16-203
odacli list-osconfigurations 16-203
odacli update-osconfigurations 16-204
odaadmcli Hardware Monitoring Commands 16-205
odaadmcli show cooling 16-205
odaadmcli show env_hw 16-206
odaadmcli show fs 16-206
odaadmcli show memory 16-207
odaadmcli show network 16-208
odaadmcli show power 16-209
odaadmcli show processor 16-209
odaadmcli show server 16-210
Storage Commands 16-211
odaadmcli asmops diskgroup 16-211
odaadmcli expand storage 16-212
odacli list-dgdisks 16-213
odacli validate-storagetopology 16-215
odaadmcli power disk 16-215
odaadmcli show controller 16-216
odaadmcli show disk 16-217
odaadmcli show diskgroup 16-219
odaadmcli show raidsyncstatus 16-220
odaadmcli show storage 16-220
odaadmcli stordiag 16-221

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 Database System Commands for KVM 17-5
odacli create-dbsystem 17-6
odacli modify-dbsystem 17-7
odacli modify-dbfileattributes 17-9
odacli list-dbsystems 17-10
odacli describe-dbsystem 17-11
odacli describe-dbsystem-image 17-17
odacli start-dbsystem 17-18
odacli stop-dbsystem 17-18
odacli delete-dbsystem 17-19
odacli VM Storage Commands for KVM 17-21
odacli create-vmstorage 17-21

odacli list-vmstorages 17-22
odacli describe-vmstorage 17-22
odacli modify-vmstorage 17-23
odacli delete-vmstorage 17-23
odacli Virtual Disk Commands for Oracle KVM 17-23
odacli create-vdisk 17-24
odacli list-vdisks 17-25
odacli describe-vdisk 17-25
odacli modify-vdisk 17-25
odacli clone-vdisk 17-26
odacli delete-vdisk 17-26
odacli CPU Pool Commands for Bare Metal and KVM 17-27
odacli create-cpupool 17-28
odacli remap-cpupools 17-29
odacli list-cpupools 17-30
odacli describe-cpupool 17-31
odacli modify-cpupool 17-31
odacli delete-cpupool 17-32
odacli list-auditrecords 17-32
odacli describe-auditrecord 17-33
odacli VM Network Commands for KVM 17-34
odacli create-vnetwork 17-35
odacli list-vnetworks 17-36
odacli describe-vnetwork 17-36
odacli start-vnetwork 17-36
odacli stop-vnetwork 17-37
odacli modify-vnetwork 17-37
odacli delete-vnetwork 17-38
odacli Virtual Machine Commands for Oracle KVM 17-38
odacli create-vm 17-39
odacli list-vms 17-41
odacli describe-vm 17-41
odacli modify-vm 17-41
odacli start-vm 17-44
odacli stop-vm 17-44
odacli migrate-vm 17-45
odacli clone-vm 17-45
odacli delete-vm 17-46

18 Troubleshooting Oracle Database Appliance

Viewing Details About DCS Error Messages 18-2
Collecting Diagnostics Data Using the BUI 18-2
Resolving Errors When Updating DCS Components During Patching 18-3
Viewing Component Information on the Appliance 18-8
Errors When Logging into the Browser User Interface 18-11
Errors when re-imaging Oracle Database Appliance 18-12
Using Oracle Autonomous Health Framework for Running Diagnostics 18-12
- About Installing Oracle Autonomous Health Framework 18-13
- Using the Oracle ORAchk Health Check Tool 18-14
- Generating and Viewing Oracle ORAchk Health Check Tool Reports in the Browser User Interface 18-15
- Generating and Viewing Database Security Assessment Reports in the Browser User Interface 18-15
- Running Oracle Trace File Analyzer (TFA) Collector Commands 18-16
- Sanitizing Sensitive Information in Diagnostic Collections 18-18
- Enabling Adaptive Classification and Redaction (ACR) 18-20
- Sanitizing Sensitive Information in Oracle Trace File Analyzer Collections 18-21
- Sanitizing Sensitive Information in Oracle ORAchk Output 18-21

Running the Disk Diagnostic Tool 18-23
Running the Oracle Database Appliance Hardware Monitoring Tool 18-23
Configuring a Trusted SSL Certificate for Oracle Database Appliance 18-24
- Option 1: Creating a Key and Java Keystore and Importing a Trusted Certificate 18-25
- Option 2: Packaging an Existing PEM-format Key and Certificates in a New Java Keystore 18-27
- Option 3: Converting an Existing PKCS or PFX Keystore to a Java Keystore 18-27
- Configuring the DCS Server to Use Custom Keystore 18-28
- Configuring the DCS Agent for Custom Certificate 18-29

Disabling the Browser User Interface 18-29
Preparing Log Files for Oracle Support Services 18-30

A Oracle Database Appliance Software Configuration Defaults

Directory Paths for Oracle Database Appliance A-1
Location of Log Files A-2
Oracle Groups and User Configurations for Oracle Database Appliance A-3

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-4
Usable Space on Oracle Database Appliance X8-2S and X8-2M B-4
Usable Space on Oracle Database Appliance X8-2-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-2
Example JSON Files to Create a Single Node Appliance with the CLI C-2
Example JSON File to Create a Multi-Node 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-3
Database Shapes for the DSS Class D-4

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. The following documents are published in the Oracle Database Appliance online documentation library:

- *Oracle Database Appliance Release Notes*
- *Oracle Database Appliance Licensing Information User Manual*
- *Oracle Database Appliance Security Guide*
- *Oracle Database Appliance Accessibility Guide*
- *Oracle Database Appliance X8-2 Deployment and User's Guide*
- *Oracle Database Appliance X7-2 Deployment and User's Guide*
- *Oracle Database Appliance X6-2-HA Deployment and User's Guide*
- *Oracle Database Appliance X6-2S/M/L Deployment and User's Guide*
- *Oracle Database Appliance X5-2 Deployment and User's Guide*
- Oracle Database Appliance Setup Posters and Booklets (a full-size printed copy ships with Oracle Database Appliance)
- *Oracle Database Appliance Owner's Guide*
- *Oracle Database Appliance Service Manual*
- *Oracle Database Appliance Series Safety and Compliance Guide*

For more information about using Oracle Database, go to http://docs.oracle.com/en/database/ and select the database release from the menu. See the following documents in the Oracle Database online documentation library:

- *Oracle Database Security Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database SQL Language Quick Reference*
- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Database Backup and Recovery Reference*
- *Oracle Database Utilities*
- *Oracle Automatic Storage Management Administrator's Guide*

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 Patching Oracle Database Appliance Bare Metal Deployments
  Follow the checklist items to apply patches to Oracle Database Appliance bare metal deployments.
- 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:

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

Component	Information Required	Comments
Timezone	The time zone where you plan to operate the Oracle Database Appliance system.	Timezone:
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. Obtain the following additional network information for virtualized platforms: • OVM Node0 Dom0 Host Name • OVM Node0 Dom0 IP Address • OVM Node1 Dom0 Host Name • OVM Node1 Dom0 IP Address	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
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.
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
Information for Virtualized Platform Deployments Only	Base Memory: Memory allocated to the ODA_BASE domain.	Use this space to note your values.

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.

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

Component	Tasks
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*.
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*.

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

Component	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.
- 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.
- 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.
- 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 Patching Oracle Database Appliance Bare Metal Deployments

Follow the checklist items to apply patches to Oracle Database Appliance bare metal deployments.

Table 1-3 Checklist for Patching Oracle Database Appliance Bare Metal Deployments

Component	Tasks
Verify Appliance is running and network and cabling are correctly configured	On High-Availability systems only: Verify cabling: `# /opt/oracle/dcs/bin/odacli validate-storagetopology` Verify interconnect networking. Run the commands `# ethtool p1p1` and `# ethtool p1p2`.

Table 1-3 (Cont.) Checklist for Patching Oracle Database Appliance Bare Metal Deployments

Component	Tasks
Check the current version	Checking the current version enables you to verify if a patch is applicable to your appliance. You can then determine whether to upgrade directly to the target patch version or whether you need to upgrade to an interim version. Run: `# /opt/oracle/dcs/bin/odacli describe-component -v`
Download the Patches from My Oracle Support	Log into My Oracle Support and download the bundle patches for the release. Refer to the *Oracle Database Appliance Release Notes* for the latest software.
Run Patching Pre-checks and Patch Oracle Database Appliance Using the Browser User Interface	Upload the zip files to a temporary location on Oracle Database Appliance and unzip the patch. Run the command on Node0. The patches are propagated to the other node automatically. Log into the Browser User Interface with the `oda-admin` user name and password. Follow the steps described in the topic *Patching Oracle Database Appliance Using the Browser User Interface*.

Table 1-3 (Cont.) Checklist for Patching Oracle Database Appliance Bare Metal Deployments

Component	Tasks
Run Patching Pre-checks and Patch Oracle Database Appliance Using the CLI	1. Upload the zip files to a temporary location on Oracle Database Appliance and unzip the patch. Run the command on Node0. The patches are propagated to the other node automatically. 2. Update the repository: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /opt/`*`software_file1`*`,`*`software_file2`*`,`*`software_file3`* 3. Confirm update: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli describe-job -i `*`job_ID`* 4. Update the DCS admin: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsadmin -v `*`version`* 5. Update the DCS components: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v `*`version`* 6. Update the DCS agent: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsagent -v `*`version`* 7. Run the patching prechecks: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -s -v `*`version`* 8. Update the server: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-server -v `*`version`* 9. Update storage: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-storage -v `*`version`*

Table 1-3 (Cont.) Checklist for Patching Oracle Database Appliance Bare Metal Deployments

Component	Tasks
	10. Update database: `[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome -v version` Follow the steps and run the commands described in the topic *Patching Oracle Database Appliance Using the CLI*.
Run the Cleanup Script (Optional)	If your patch 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

- 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.
- 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 1-4 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 subnet 192.168.16.0/24 is 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 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 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 Hardware Model	Deployment Use Case
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, 12.2.0.1, and 12.1.0.2 and 21.3 (in DB system)
Oracle Database Standard Edition 2	Oracle Database release 19c, 12.2.0.1, and 12.1.0.2 and 21.3 (in DB system)

> **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, update, test, and delete Oracle ASR
- **Patch 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 Database System
  - List and describe Database System
  - Modify, delete, start, and stop Database 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
- **Object 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
- **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 X8-2-HA
  Connect the storage cables for Oracle Database Appliance X8-2-HA.
- Attaching Network Cables to Oracle Database Appliance X8-2-HA
  Connect Oracle Database Appliance X8-2-HA to either a 10GBase-T (copper) or 25GbE SFP28 (fiber) network.
- Attaching Network Cables to Oracle Database Appliance X8-2S or X8-2M
  Connect Oracle Database Appliance X8-2S or X8-2M 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 X8-2-HA

Connect the storage cables for Oracle Database Appliance X8-2-HA.

Cabling the Storage for Oracle Database Appliance X8-2-HA

Connect the storage cables for Oracle Database Appliance X8-2-HA.

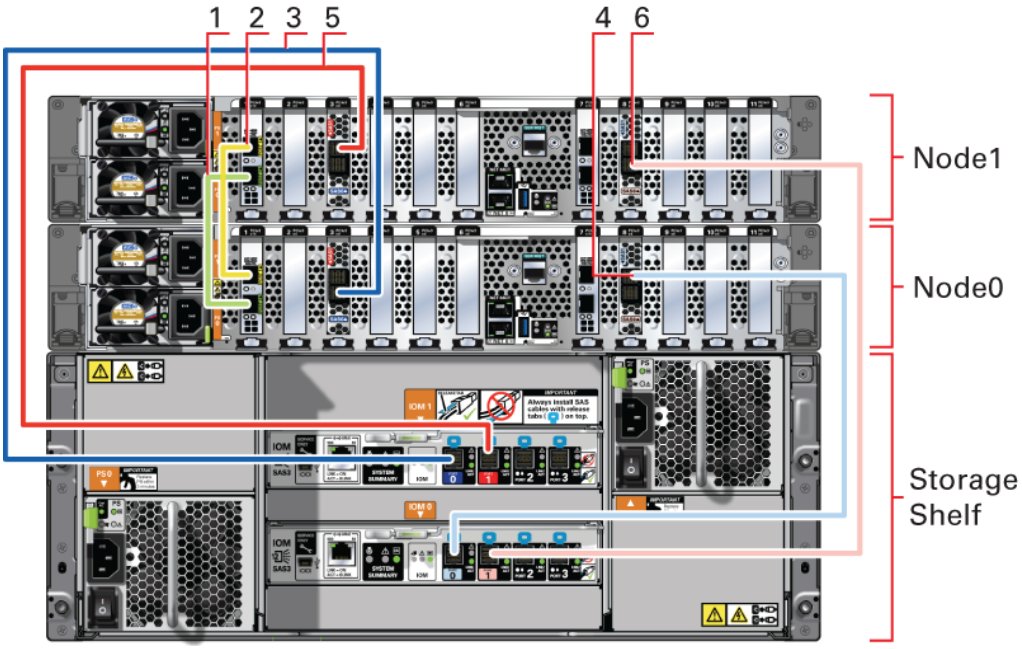


Table 3-1 Storage Cabling for Oracle Database Appliance X8-2-HA

Callout Number	Purpose	Start - Compute Node0	End - Compute Node1
1	Connect green SFP+ 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 SFP+ 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 3 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 8 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 3 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 8 in Node1	Connect into light red port in bottom IO Module (PORT 1)

Cabling the Storage Expansion Shelf

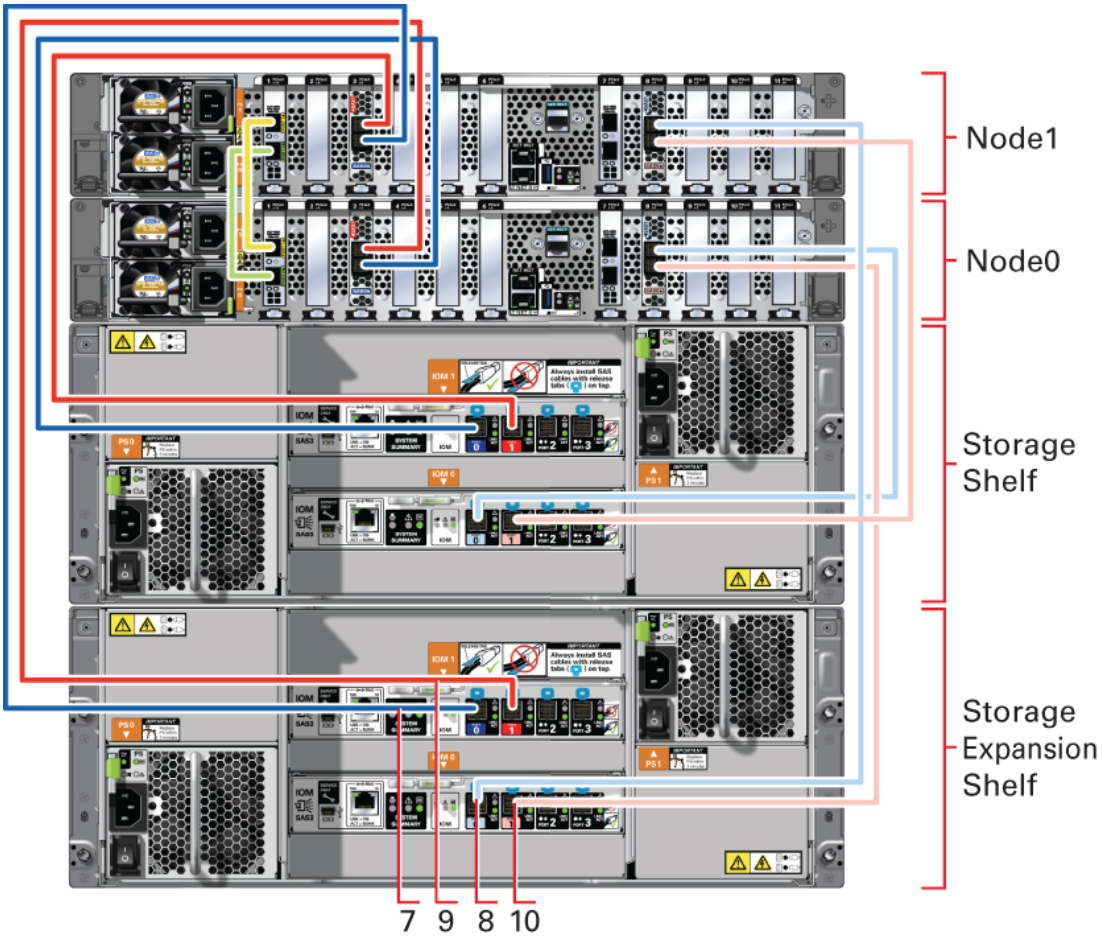


Table 3-2 Cabling the Storage Expansion Shelf for Oracle Database Appliance X8-2-HA

Callout Number	Purpose	Start - Compute Nodes	End - Expansion Shelf
7	Connect dark blue SAS cable	Connect into dark blue port (SAS0) in PCIe slot 3 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 8 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 3 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 8 in Node0	Connect into light red port in bottom IO Module (PORT 1)

Attaching Network Cables to Oracle Database Appliance X8-2-HA

Connect Oracle Database Appliance X8-2-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 X8-2-HA

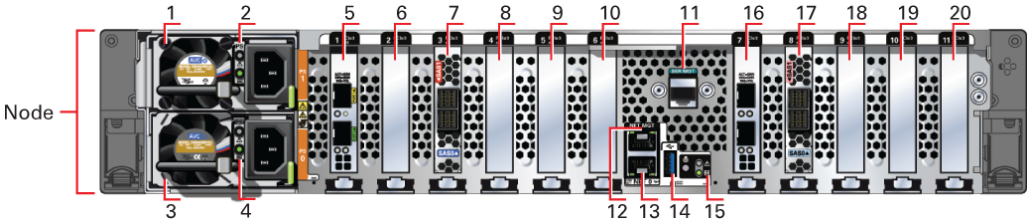


Table 3-3 Network Ports and Cabling

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: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
7	PCIe card slot 3: provides two SAS3 connectors used to connect the servers to the storage shelf and the storage expansion shelf
8-10	PCI card slots 4-6: filler panels
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

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

Callout	Description
15	System status indicators: Locate LED: white, Service Required LED: amber, Power/OK LED: green
16	PCIe card slot 7: Oracle Dual Port 25Gb Ethernet Adapter or Oracle Quad Port 10GBase-T Adapter
17	PCIe card slot 8: provides two SAS3 connectors used to connect the servers to the storage shelf and the storage expansion shelf
18	PCIe card slot 9: filler panel
19	PCIe card slot 10: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
20	PCIe card slot 11: 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 X8-2S or X8-2M

Connect Oracle Database Appliance X8-2S or X8-2M 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 X8-2S or X8-2M

Connect the Fiber and Copper Network Cables for Oracle Database Appliance X8-2S or X8-2M

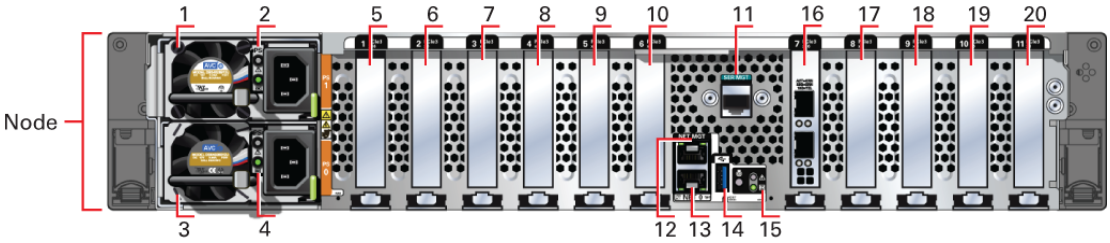


Table 3-8 Network Ports and Power Cabling

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: Oracle Database Appliance X8-2S: filler panel Oracle Database Appliance X8-2M: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter
7-8	PCIe card slot 3 and PCI card slot 4: Oracle Database Appliance X8-2S: filler panel Oracle Database Appliance X8-2M: Oracle NVMe Switch PCIe card
9-10	PCI card slots 5-6: filler panels
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 7: Oracle Dual Port 25Gb Ethernet Adapter or Oracle Quad Port 10GBase-T Adapter
17	PCIe card slot 8: Oracle Database Appliance X8-2S: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter Oracle Database Appliance X8-2M: Oracle NVMe Switch PCIe card
18	PCIe card slot 9: Oracle NVMe Switch PCIe card
19	PCIe card slot 10: filler panel, or optional Oracle Dual Port 25Gb Ethernet Adapter, or optional Oracle Quad Port 10GBase-T Adapter

Table 3-8 (Cont.) Network Ports and Power Cabling

Callout	Description
20	PCIe card slot 11: 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 3-9 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.

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 X8-2-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 X8-2-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

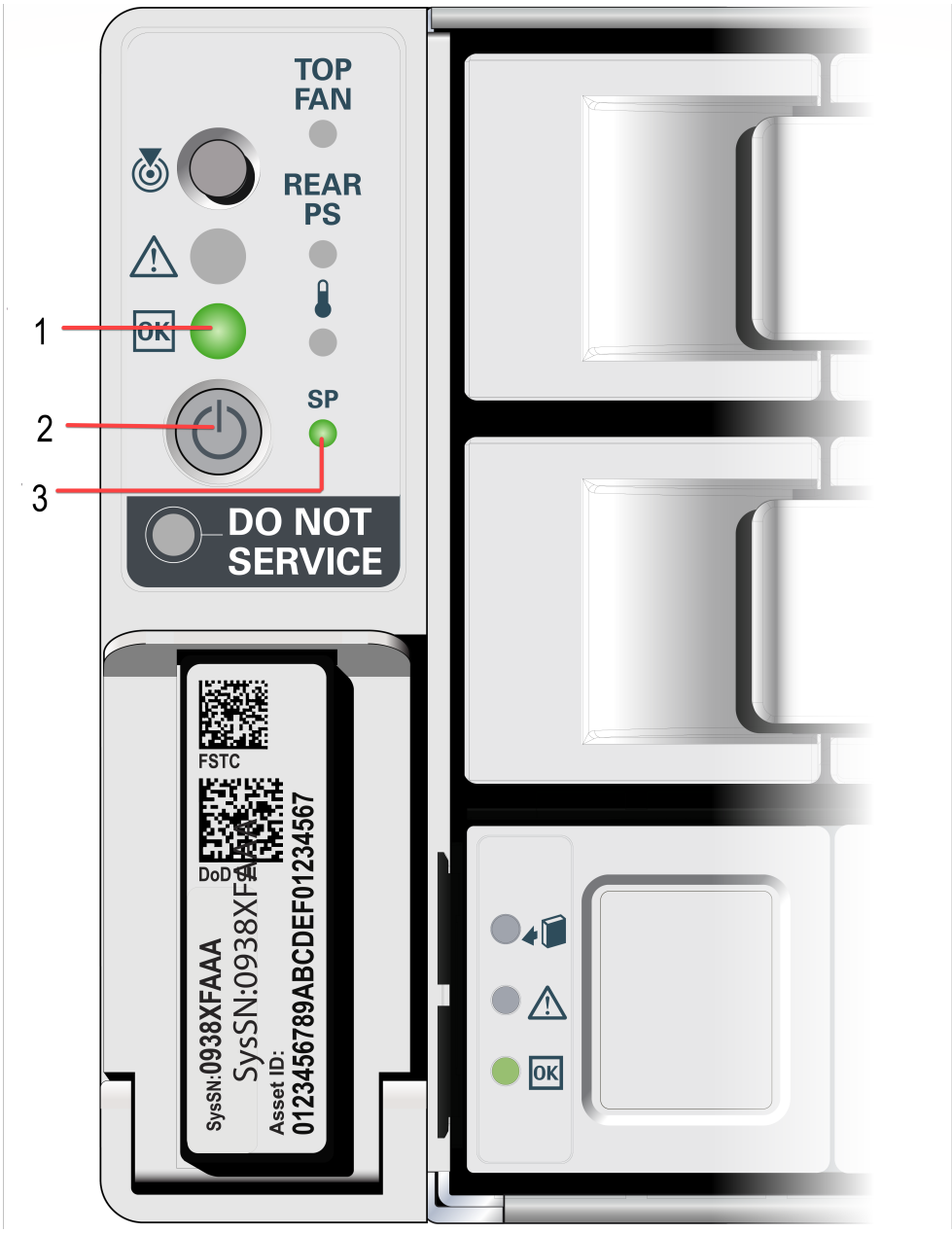


Table 3-13 Description of Callouts for Powering On Oracle Database Appliance

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.

- 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.
- Verifying Cabling and Network Connections
  After powering on Oracle Database Appliance, verify that the network connections are set up correctly.
- 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.

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.

1. Connect to Oracle ILOM remote console, then log into Oracle Database Appliance as `root`.
2. Run the command `configure-firstnet` on **both** nodes.

```
# /opt/oracle/dcs/bin/odacli configure-firstnet
Select the Interface to configure the network on (btbond1 btbond2 btbond3
btbond4 btbond5 btbond6) [btbond1]:btbond3
Configure DHCP on btbond3 (yes/no) [no]:
INFO: You have chosen Static configuration
Use VLAN on btbond3 (yes/no) [no]:
Enter the IP address to configure : 10.31.106.12
Enter the Netmask address to configure : 255.255.240.0
```

```
Enter the Gateway address to configure[10.31.96.1] :
INFO: Restarting the network
```

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 `btbond1` interface without configuring DHCP.

```
# /opt/oracle/dcs/bin/odacli configure-firstnet
Select the Interface to configure the network on (btbond1 btbond2)
[btbond1]:
Configure DHCP on btbond1 (yes/no) [no]:
INFO: You have chosen Static configuration
Use VLAN on btbond1 (yes/no) [no]:
Enter the IP address to configure : 10.209.13.109
Enter the Netmask address to configure : 255.255.252.0
Enter the Gateway address to configure[10.209.12.1] :
INFO: Plumbing the IPs now
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
Using bonding public interface (yes/no) [yes]:
Select the Interface to configure the network on (btbond1) [btbond1]:
Configure DHCP on btbond1 (yes/no) [no]:
INFO: You have chosen Static configuration
Use VLAN on btbond1 (yes/no) [no]:yes
Configure VLAN on btbond1, input VLAN ID [2 - 4094] 122
INFO: using network interface btbond1.122
Enter the IP address to configure : 192.0.2.24
Enter the Netmask address to configure : 192.0.2.24
Enter the Gateway address to configure[192.0.2.24] :
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
```

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.

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 : X7-2
INFO : Check for Environment(Bare Metal or Virtual Machine)
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 : 3b:00.0
SUCCESS : External LSI SAS controller 1 : 5e: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 : X7-2
INFO : Check for Environment(Bare Metal or Virtual Machine)
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 : 3b:00.0
SUCCESS : External LSI SAS controller 1 : 5e: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.

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 19.14:

   ```
   p30403673_1914000_Linux-x86-64.zip
   p30403662_1914000_Linux-x86-64.zip
   ```

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

```
unzip p30403673_1914000_Linux-x86-64.zip
unzip p30403662_1914000_Linux-x86-64.zip
```

The GI and RDBMS Clone files for release 19.14 are extracted.

```
odacli-dcs-19.14.0.0.0-date-GI-19.14.0.0.zip
odacli-dcs-19.14.0.0.0-date-DB-19.14.0.0.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 19.14:

```
put odacli-dcs-19.14.0.0.0-date-GI-19.14.0.0.zip
put odacli-dcs-19.14.0.0.0-date-DB-19.14.0.0.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.

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 19.14:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
   odacli-dcs-19.14.0.0.0-date-DB-19.14.0.0.zip,/tmp/odacli-dcs-19.14.0.0.0-
   date-GI-19.14.0.0.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.

> **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.

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 section, 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.

6. In the Network Information section, 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**.

   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.

   Oracle 12.1 and later databases can use Oracle ASM storage.

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

n. **Database Role**: Select the Database role.

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

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

p. **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.

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

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

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

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

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

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

c. **SNMP Version**: Select **V2** or **V3**. V3 is the default and recommended version.

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

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

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

g. **(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**.

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

i. **(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.

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 cannot enable multi-user access on Oracle Database Appliance DB systems. DB systems do not require role separation since you can create only one database on each DB system.

- 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.
- 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 execute certain operations such as get-disks and release-disks on the DCS agent.
- **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.

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 read-only list of all databases in the system is displayed.
4. Click the Database Home link on the left hand pane.
5. Click **Show All Database Homes**. A read-only list of all database homes in the system is displayed.

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 and confirm the **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

- 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 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 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.
- 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.

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 update-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
   ```

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.

7

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 Pre-Checks
  Patching pre-checks 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.
- 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 agent and components 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 Existing Database Homes on Bare Metal Systems Using the Browser User Interface
  Use the Browser User Interface to patch database homes to the latest release on bare metal systems.
- 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.

Related Topics

- Checklist for Patching Oracle Database Appliance Bare Metal Deployments
  Follow the checklist items to apply patches to Oracle Database Appliance bare metal deployments.

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 patches
- BIOS
- Hardware drivers
- Hardware Management Pack (HMP) and firmware drivers for various components
- Oracle Appliance Manager
- Oracle Linux
- Oracle VM Server
- 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 Pre-Checks

Patching pre-checks are designed to detect and flag problems that might be encountered during patching.

The pre-checks 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 pre-checks 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 pre-checks are available when you run these commands.

Components Verified by Patching Pre-Checks

Check	Component Verified
System patching pre-checks	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 pre-checks	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 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: • Validates whether orachk is running • 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 pre-checks	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
Storage pre-checks	Validates storage patch location and tag Validates patching location and tag Validates command execution Validates Oracle ASM disk groups status Validates whether server update can be rolling or not

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 19.14, 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 19.14 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 on Oracle Database Appliance release 19.12. When you patch to Oracle Database Appliance release 19.14 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 19.14 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 database 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.

Oracle Database Appliance does **not** apply patches in a rolling manner for high-availability systems. However, there are recommendations to minimize downtime during patching.

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.
- 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, HDD, NVMe) 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.

See Also:

Oracle Database High Availability Overview and Best Practices Guide in the Oracle Database 19c Documentation Library

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 file, and Oracle Database clone files and update the repository on the bare metal system.

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.

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 33702951), 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 19.14:

   ```
   p33702951_1914000_Linux-x86-64.zip
   ```

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

   ```
   unzip p33702951_1914000_Linux-x86-64.zip
   ```

   The zip file contains the following software file:

   ```
   oda-sm-19.14.0.0.0-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 19.14:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -
f /tmp/oda-sm-19.14.0.0.0-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
19.14.0.0.0
```

7. Update the DCS components:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
19.14.0.0.0
```

If you are patching from Oracle Database Appliance release 19.9, then the `odacli update-dcscomponents` command updates the Zookeeper, installs MySQL, migrates metadata from Derby to MySQL and updates other DCS components such as the DCS Agent, DCS CLI, and DCS Controller on Oracle Database Appliance.

If you are patching from Oracle Database Appliance release 19.10 or later, then 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
19.14.0.0.0
```

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 19.14:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
odacli-dcs-19.14.0.0.0-date-GI-19.14.0.0.zip
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -f /tmp/
odacli-dcs-19.14.0.0.0-date-DB-19.14.0.0.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.

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

For example, for 19.14:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -s -v
19.14.0.0.0
```

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

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli list-prepatchreports
```

For example:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli list-prepatchreports
3797791f-72ac-42b6-8bb8-ed5e5c9b32c1 Patch pre-checks for [OS, GI,
ORACHKSERVER] June 30, 2020 6:29:30 AM UTC Running
```

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 the `--local` option is not available with the `odacli update-server` command in this release.

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

For example, for 19.14:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-server -v 19.14.0.0.0
```

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 created before the 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 19.14:

    ```
    [root@oda1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -
    st -v 19.14.0.0.0
    ```

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

    ```
    [root@oda1 opt]# /opt/oracle/dcs/bin/odacli list-prepatchreports
    ```

    For example:

    ```
    [root@oda1 opt]# odacli list-prepatchreports

    ID                                       Description
    Created                                  Status
    ----------------------------------------------------------------------
    ----------------------------------------
    b030d6e2-bb10-465c-81d3-ccaa48cea681 Patch pre-checks for [STORAGE]
    December 8, 2021 11:26:45 AM UTC Success
    ```

    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.

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

For example, for 19.14:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-storage -v 19.14.0.0.0
--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 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 19.14:**

    a. 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-19.14.0.0.0-date-DB-19.14.0.0.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 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 7c67c5b4-f585-4ba9-865f-c719c63c0a6e -v 19.14.0.0.0
    [root@oda1 opt]# odacli describe-prepatchreport -i 39ef1eeb-70d3-47ad-
    b3f5-48960ca0607b
    ```

    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.

c. If the pre-patch report does not show any errors, then update the database home to the latest release:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --
dbhomeid DB_Home_ID -v release_number
```

For example, for 19.14:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --
dbhomeid 7c67c5b4-f585-4ba9-865f-c719c63c0a6e -v 19.14.0.0.0
```

19. Confirm that the patch jobs are successful:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli list-jobs
```

20. 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 X6-2S, 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 on Bare Metal Systems Using the Browser User Interface
  Use the Browser User Interface to patch database homes to the latest release on bare metal systems.

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**
  With this release, 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 database 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.

Ensure that you have patched your Oracle Database Appliance bare metal deployment to release 19.14 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 (patch 33702951), Oracle Grid Infrastructure clone files (patch 30403673), and Oracle Database clone files (patch 30403662) 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
   19.14.0.0.0
   ```

4. Update the DCS components:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
19.14.0.0.0
```

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
19.14.0.0.0
```

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 19.14:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli create-prepatchreport -s
-v 19.14.0.0.0
```

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

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli list-prepatchreports
```

For example:

```
[root@dbsystem1 opt]# odacli list-prepatchreports
ID Description Created Status
----------------------------------------
--------------------------------------------------------------------------
- ---------------------------------- ----------
67029f6b-db5d-4f65-9367-c5986c3050d4 Patch pre-checks for [OS, GI,
ORACHKSERVER] December 8, 2021 8:44:24 AM GMT Success
```

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 the `--local` option is not available with the `odacli update-server` command in this release.

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-server -v
version
```

For example, for 19.14:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-server -v
19.14.0.0.0
```

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 19.14:**

    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-19.14.0.0.0-date-DB-19.14.0.0.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 19.14.0.0.0
[root@dbsystem1 opt]# odacli describe-prepatchreport -i
39ef1eeb-70d3-47ad-b3f5-48960ca0607b
```

c. Update the database home to the latest release:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --
dbhomeid DB_Home_ID -v release_number
```

For example, for 19.14:

```
[root@dbsystem1 opt]# /opt/oracle/dcs/bin/odacli update-dbhome --
dbhomeid 7c67c5b4-f585-4ba9-865f-c719c63c0a6e -v 19.14.0.0.0
```

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 database 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 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 with the `oda-admin` user name and password.

   `https://Node0-host-ip-address:7093/mgmt/index.html`

3. Click **Patch 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 and DCS Components After Updating the Repository and Before Patching

Before patching your appliance using the Browser User Interface, you must update the DCS admin and DCS components manually, through CLI commands.

Follow these steps:

1. Update DCS admin:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsadmin -v
   19.14.0.0.0
   ```

2. Confirm that the update is successful:

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

3. Update the DCS components:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
   19.14.0.0.0
   ```

   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 Agent, Server, and Storage Using the Browser User Interface

1. Do not have any jobs running or pending during the update window.
2. Click the **Appliance** tab, and then click **Patch Manager** in the Browser User Interface.

3. Wait for the Component Details to load. If you just updated the Patch Repository, click **Refresh** in the Component Details section of the page.

   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. On the Patch Manager page, click **Refresh** to refresh the Component Details. Select **Update Server**, then click **Apply Patch** to begin the job to patch the server components. Select **Precheck** to run patching prechecks. You must run the patching prechecks before updating the server and other components.

   The DCS agent is automatically updated whenever the Server or Storage components are updated.

   When the job finishes, go to the next step. Click **Activity** for job status.

5. On the Patch Manager page, click **Refresh** to refresh the Component Details. Select **Update Storage**, then click **Apply Patch** to begin the job to patch the storage components. Select **Precheck** if you want to run patching prechecks.

   For high-availability environment, you can select the **Rolling** check box to perform rolling patching of storage components.

6. Click **Activity** to monitor the progress, and verify that the job completes successfully.

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 **Patch 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 and DCS Components After Updating the Repository and Before Patching

Before patching your DB system using the Browser User Interface, you must update the DCS admin and DCS components manually, through CLI commands.

Follow these steps:

1. Update DCS admin:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcsadmin -v
   19.14.0.0.0
   ```

2. Confirm that the update is successful:

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

3. Update the DCS components:

   ```
   [root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-dcscomponents -v
   19.14.0.0.0
   ```

   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 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 **Patch Manager** in the Browser User Interface.
4. Wait for the Component Details to load. If you just updated the Patch Repository, click **Refresh** in the Component Details section of the page.

   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. Select **Precheck** to run patching prechecks.

6. On the Patch Manager page, click **Refresh** to refresh the Component Details. Select **Update Server**, then click **Apply Patch** to begin the job to patch the server components.

   The DCS agent is automatically updated whenever the server components are updated.

   When the job finishes, go to the next step. Click **Activity** for job status.
7. On the Patch Manager page, click **Refresh** to refresh the Component Details. Select **Precheck** if you want to run patching prechecks.
8. 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_1914000_Linux-x86-64.zip`, and extract the Oracle Database clone software:

   For example, download the Oracle Database clone software patch for 19.14:

   ```
   unzip /tmp/p30403662_1914000_Linux-x86-64.zip
   odacli-dcs-19.14.0.0.0-date-DB-19.14.0.0.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 19.14:

```
[root@oda1 opt]# /opt/oracle/dcs/bin/odacli update-repository -
f /tmp/odacli-dcs-19.14.0.0.0-date-DB-19.14.0.0.zip
```

Follow the same steps to update the repository with the database clone software for other supported database versions such as 12.2, 12.1, and so on.

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 Existing Database Homes on Bare Metal Systems Using the Browser User Interface

Use the Browser User Interface to patch database homes to the latest release on bare metal systems.

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.

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, and then select the patch version for the database home.
3. Select the database home you want to patch, and click **Patch**. Select **Precheck** to run pre-checks before patching the database.

   Click **Activity** for job status.
4. 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.
5. Click **Ignore Precheck Failures** to ignore failures reported in the prechecks reported. It is recommended that you fix errors reported in the precheck results.
6. Click **Ignore Missing Patches** to ignore missing patches.
7. 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 19.10 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 19.10 or later.
2. Update the repository with the server patch for the Oracle Database Appliance release you want to patch to, for example 19.10 or later.
3. For running `odacli update-server` on Oracle Database Appliance imaged with ISO of release 19.10:

   ```
   odacli update-server -v 19.10.0.0.0 -sko -f
   ```

   For Oracle Database Appliance release 19.10, 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.

Related Topics

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

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://linux.oracle.com
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/cd/E37670_01/E39380/html/ksplice-enhanced-offline.html

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
- Oracle Database 12.1 to 12.2

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*.

Related Topics

- odacli upgrade-database
  You can use the `odacli upgrade-database` command to upgrade a database from a supported release.

8

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.

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] [-idmusers
dbuser1,dbuser2,dbuser3][-erasedata] [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 oracle -dbuser oracle -idmusers
dbuser1,dbuser2,dbuser3
```

Table 8-1 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 oracle -dbuser oracle`
groups	Describes the comma-separated list of groups. The default groups are `oinstall,dba,asmadmin,asmoper,asmdba`.

Table 8-1 (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`

`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.

Related Topics

- Checklist for Patching Oracle Database Appliance Bare Metal Deployments
  Follow the checklist items to apply patches to Oracle Database Appliance bare metal deployments.

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.
- 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.

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:**
>
> 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 8-1 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

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
  With this release, 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 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.
- 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 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

With this release, 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.

Starting with Oracle Database Appliance release 19.14, 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/`.

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/odaorahome/`.

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 19.14

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 19.14

If you have patched your appliance from Oracle Database Appliance release 19.6 or later to 19.14, 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 19.14

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 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:

```
odacli modify-database -in db_name -g target_node | -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, backup, recover, and restore TDE-enabled databases, backup and restore TDE wallet, re-keying of TDE master encryption key, and changing the password of password-protected Keystore.

Note:

To enable Transparent Data Encryption (TDE), you must deploy Enterprise Edition Oracle Database 19c or later.

See Also:

Using Transparent Data Encryption in the *Oracle Database Advanced Security Guide*

Caution:

When you enable TDE, 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.

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 Using the Browser User Interface
  Create a database from an existing Oracle ACFS database using the Browser User Interface.
- Cloning an Oracle ACFS Database Using Command Line Interface
  Create a database from an existing Oracle ACFS database using CLI commands.
- 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.
- Deleting a Database
  Use the Oracle Appliance Manager Browser User Interface to delete an Oracle database.

Viewing Databases

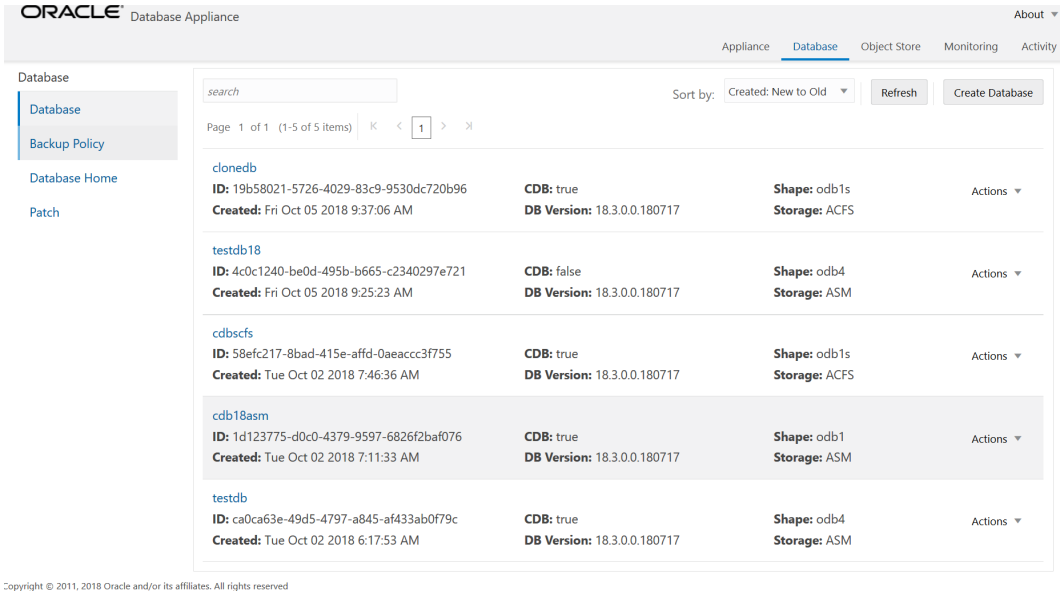
Use the Oracle Appliance Manager Browser User Interface to display a list of Oracle databases and database details, create, upgrade, and delete a database in Oracle Database Appliance.

On the Oracle Database Appliance Virtualized Platform, see the command `oakcli list databases`.

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.

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 19.14 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 Create Database 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.

   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. 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 19.14. Selecting **Yes** enables you to use the console to manage the database.

p. 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.

q. In the **Confirm Password** field, enter the password again to confirm.

r. In the **Characterset** field, select an option from the drop-down list. The default is AL32UTF8.

s. In the **National Characterset** field, select an option from the drop-down list. The default is AL16UTF16.

t. In the **Language** field, select a database language from the drop-down list. The default is American.

u. In the **Territory** field, select a territory or location for the database from the drop-down list. The default is America.

v. 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.

6. 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 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. 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 19.14, 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  19.14.0.0.0

   (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 19.14.0.0.0.

```
# odacli  create-dbhome -v 19.14.0.0.0
```

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 `19.14.0.0.0` and a database home:

   ```
   # odacli create-database -n PRODDB -v 19.14.0.0.0 -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 ObjectStore:

   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, additional fields appear on the page and are 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.

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. Click **Create**.
12. 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.

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.

    ```
    # odacli describe-job -i jobId
    ```

Cloning an Oracle ACFS Database Using the Browser User Interface

Create a database from an existing Oracle ACFS database using the Browser User Interface.

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.

Follow these steps to create a database from an existing Oracle ACFS 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. Click **Clone a Database**, then click **Next** to display the Clone Database page.
5. Select the **Source Database** from which you want to create the database.

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

6. Select the **Database Shape** (template) for your new database. The database shape you select determines the total memory allocated to the database.
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. 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.
9. Specify the Networks.
10. 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.
11. Click **Create**.
12. 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.

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.

    ```
    # odacli describe-job -i jobId
    ```

Cloning an Oracle ACFS Database Using Command Line Interface

Create a database from an existing Oracle ACFS database using CLI commands.

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.
- 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.

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 databases (CDBs)
- The source database must be in the OPEN state.
- The source database must not be in backup mode.
- The source database must be in archive mode.

> **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.

Follow this procedure to clone a database:

- Run the `odacli clone-database` command.

  ```
  odacli clone-database --databaseUniqueName --dbname --dbshape --
  dbtype --json --sourcedbname
  ```

  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.

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. 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 -in 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.
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, then click **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.

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. 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 19.14.0.0.0 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.

> **Note:**
>
> The `odacli register-database` command is supported only for primary database of type single-instance.

Follow these steps to register a database:

1. If you have Oracle Database version 12.1 with SQL patches, 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 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.

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`.

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:
            SEHA 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:
          SEHA 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 Oracle Database 18c, 12.2.0.1, and 12.1.0.2. Both 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 X8-2S and X8-2M systems.
- Managing Storage on High-Availability Systems
  Understand the storage for your Oracle Database Appliance X8-2-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 X8-2M and X8-2-HA models provide storage expansion options from the base configuration. In addition, on Oracle Database Appliance X8-2-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 update-agentconfig-parameters -n ASMRM_CPU_RQ -v 50 -d "CPU RUN
QUEUE THRESHOLD" -u
odacli update-agentconfig-parameters -n ASMRM_MAX_HDD_DISK_RQ -v 2 -d
"HDD DISK QUEUE THRESHOLD" -u
odacli update-agentconfig-parameters -n ASMRM_MAX_SSD_DISK_RQ -v 32 -d
"SSD DISK QUEUE THRESHOLD" -u
odacli update-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 X8-2S and X8-2M systems.

- About Storage on Oracle Database Appliance X8-2S and X8-2M
  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 X8-2 storage to add NVMe disks or replace existing NVMe disks.

About Storage on Oracle Database Appliance X8-2S and X8-2M

Understand the storage for your Oracle Database Appliance single-node system.

Oracle Database Appliance X8-2S has two 6.4TB 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 X8-2M has two 6.4TB 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 X8-2M in this release, you can set the storage on X8-2M 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 X8-2S and X8-2M

Configuration	Oracle Database Appliance X8-2S	Oracle Database Appliance X8-2M
Base Configuration	2 x 6.4 TB NVMe = 12.8 TB NVMe	2 x 6.4 TB NVMe = 12.8 TB NVMe
Storage addition options	None	10 x 6.4 TB NVMe storage drives for a total storage of 76.8 TB NVMe. Order Qty 1 - 7600927 (2-pack 6.4TB 2.5-inch NVMe PCIe 3.0 SSD v2 with coral-d bracket for Oracle Database Appliance X8-2M) Pack of two 6.4 TB NVMe disks to a maximum of 12 disks.

Adding NVMe Storage Disks

Depending on the available drives, you can expand Oracle Database Appliance X8-2 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. 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 `odaadmcli show diskgroup` command to display and review Oracle Automatic Storage Management (Oracle ASM) disk group information.
4. Use orachk to confirm Oracle ASM and CRS health.

Review and perform these best practices before adding storage.

Adding NVMe Storage Disks

The default configuration for Oracle Database Appliance X8-2S or X8-2M includes two (2) NVMe disks. You cannot expand storage for Oracle Database Appliance X8-2S.

For Oracle Database Appliance X8-2M, 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 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 disks one at a time in the slots and power on the device.
   For example, to add two (2) NVMe disks, insert the disks in slots 2 and 3. To add four (4) NVMe drives, insert the disks in slots 2 to 5.

   ```
   # odaadmcli power disk on slot_number
   ```

   For example, when adding four (4) NVMe disks:

   ```
   # odaadmcli power disk on pd_02
   # odaadmcli power disk on pd_03
   # odaadmcli power disk on pd_04
   # odaadmcli power disk on pd_05
   ```

   Allow at least one minute between inserting each disk.
3. Run the `odaadmcli expand storage` command to add the new storage drives:

   ```
   # odaadmcli expand storage -ndisk number_of_disks
   ```

   For example, to add two (2) NVMe drives:

   ```
   #odaadmcli expand storage -ndisk 4
   Precheck passed.
   Check the progress of expansion of storage by executing 'odaadmcli
   show disk'
   Waiting for expansion to finish ...
   ```

4. Run the `odaadmcli show disk` command to ensure that all disks are listed, are online, and are in a good state.

   ```
   # odaadmcli show disk
   ```

Managing Storage on High-Availability Systems

Understand the storage for your Oracle Database Appliance X8-2-HA system.

- About Storage Options for Oracle Database Appliance X8-2-HA
  Oracle Database Appliance High-Availability systems have options for high performance and high capacity storage configurations.
- Preparing for a Storage Upgrade for a Virtualized Platform
  Review and perform these best practices before adding storage to the base shelf or adding the expansion shelf.
- 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 X8-2-HA and X8-2M 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 X8-2-HA

Oracle Database Appliance High-Availability systems have options for high performance and high capacity storage configurations.

The base configuration of Oracle Database Appliance X8-2HA 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 X8-2-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 X8-2-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 X8-2-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 using only SSD, 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 252 TB HDD raw storage for DATA and RECO (18 HDDs, each with 14 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 596.16 TB raw storage for DATA, RECO, and FLASH, with 92.16 TB SSD and 504 TB HDD.

Table 10-2 Storage Options for Oracle Database Appliance X8-2-HA

Configuration	Oracle Database Appliance X8-2-HA SSD-Only Configuration for High Performance	Oracle Database Appliance X8-2-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 14 TB. • 6 x 7.68 TB SSD = 46 TB SSD • 18 x 14TB HDD = 252 TB HDD • Total storage on the first JBOD = 298 TB, with 46 TB SSD and 252 TB HDD

Table 10-2 (Cont.) Storage Options for Oracle Database Appliance X8-2-HA

Configuration	Oracle Database Appliance X8-2-HA SSD-Only Configuration for High Performance	Oracle Database Appliance X8-2-HA SSD and HDD Configuration for High Capacity
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 = 368.64 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 = 298 TB, with 46 TB SSD and 252 TB HDD • Total storage including both JBODs = 596 TB, with 92 TB SSD and 504 TB HDD

Preparing for a Storage Upgrade for a Virtualized Platform

Review and perform these best practices before adding storage to the base shelf or adding the expansion shelf.

1. Update Oracle Database Appliance to the latest Patch Bundle before expanding storage.
2. Confirm both nodes are at the same version and patch bundle level for software and firmware.

```
# oakcli show version -detail
```

```
# oakcli inventory -q
```

> **Note:**
>
> If `oakd` is not running on either node, fix the problem before adding storage.

3. Check the disk health of the existing storage disks.

Run the check on both nodes and use the default checks option to check the `NetworkComponents`, `OSDiskStorage`, `SharedStorage`, and `SystemComponents`.

```
# oakcli validate -d
```

4. Run the command `oakcli show diskgroup` on each node to display and review Oracle Automatic Storage Management (Oracle ASM) disk group information.

```
# oakcli show diskgroup data


# oakcli show diskgroup reco


 # oakcli show diskgroup redo
```

5. Confirm Oracle ASM and CRS health on both nodes.

   Run the `oakcli orachk` command on each node. If there is a problem connecting to either node, then check the `/etc/bashrc` file and remove (or remark out) any values in the profile for `root` ; `oracle` ; `grid users`

   Run `oakcli orachk` on Node 0:

```
# oakcli orachk
...

Checking Status of Oracle Software Stack - Clusterware, ASM, RDBMS

. . . . . .
--------------------------------------------------------------------------------------------------
------
Oracle Stack Status
--------------------------------------------------------------------------------------------------
------
Host Name CRS Installed  ASM HOME   RDBMS Installed    CRS UP    ASM UP    RDBMS UP DB Instance
Name
--------------------------------------------------------------------------------------------------
------
odax3rm1       Yes           No          Yes              No        No
No          ........
--------------------------------------------------------------------------------------------------
------

 ...
```

   Run `oakcli orachk` on Node 1:

```
# oakcli orachk
...

Checking Status of Oracle Software Stack - Clusterware, ASM, RDBMS

. . . . . .
--------------------------------------------------------------------------------------------------
------
Oracle Stack Status
--------------------------------------------------------------------------------------------------
------
Host Name CRS Installed  ASM HOME   RDBMS Installed    CRS UP    ASM UP    RDBMS UP DB Instance
Name
--------------------------------------------------------------------------------------------------
------
odax3rm2      Yes           Yes           Yes            Yes       Yes        Yes      b22S2
b23S2 b24S2
--------------------------------------------------------------------------------------------------
```

```
------

...
```

6. Confirm communications between the nodes and that SSH is working using the same password for `oracle`, `root` and `grid`.

   From each node:

   a. ssh to both nodes.

   b. Ping both nodes.

7. Confirm that there is at least 10 GB of space available on each node.

```
[root@oda]# df -h


[root@odb]# df -h
```

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 X8-2-HA and X8-2M 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 X8-2-HA.

You must fully populate the base configuration before you can add an expansion shelf to Oracle Database Appliance X8-2-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.

> **Note:**
>
> You can follow the same procedure to add storage to the base configuration on Virtualized Platform by using `oakcli` command equivalent of the `odacli` and `odaadmcli` commands in the procedure.

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
Precheck passed.
Check the progress of expansion of storage by executing 'odaadmcli
show disk'
Waiting for expansion to finish ...
```

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 - grid user -c /opt/oracle/oak/bin/stordiag/asm_script.sh 1 6
   ```

   For example:

   ```
   # /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_S09_1399765506p1,
'/dev/mapper/SSD_E0_S11_1399765508p1' name SSD_E0_S09_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_S09_1399765506p2,
'/dev/mapper/SSD_E0_S11_1399765508p2' name SSD_E0_S09_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 `oak_storage_conf.xml` file must show 12. If you added 12 disks to the base configuration, then the `oak_storage_conf.xml` file must show 18.

   ```
   # cat /opt/oracle/oak/conf/oak_storage_conf.xml
   ```

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 X8-2-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:**
>
> You can follow the same procedure to add storage to the base configuration on Virtualized Platform by using `oakcli` command equivalent of the `odacli` or `odaadmcli` commands in the procedure.

> **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. If this is a new deployment or re-image of Oracle Database Appliance, perform the following steps in order:
   a. Power on the base storage.
   b. Power on Node 0.
   c. Power on Node 1.

   > **Caution:**
   >
   > Do not power on the expansion shelf yet.

3. 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 :
> ```

4. Power on the storage expansion shelf and wait for 20 minutes before issuing the CLI command for storage expansion.
5. 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.

6. 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.

7. Run the `odaadmcli show enclosure` command to check the health of components in expansion shelf.

   ```
   # odaadmcli show enclosure
   ```

8. Run the `odaadmcli expand storage` command.

   ```
   # odaadmcli expand storage -ndisk 24 -enclosure 1

   Precheck passed.
   Check the progress of expansion of storage by executing 'odaadmcli show
   disk'
   Waiting for expansion to finish ...
   ```

   It takes approximately 30 to 40 minutes to add all of the disks to the configuration.

9. 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
   ```

10. 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
    ```

11. Run the `odacli describe-component` command to verify that all firmware components in the storage expansion are current.

```
# odacli describe-component
```

12. 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 X8-2 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- and multi-node 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 X8-2 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 X8-2-HA.

Table 11-1 Network Interfaces for Oracle Database Appliance X8-2-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 7 (required)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p7p1, p7p2 (p7p3, p7p4)	btbond1 (btbond2)	Either one of the IP addresses: btbond1 or btbond2, can be assigned during deployment. The remaining IP address can be assigned post deployment.
PCIe Slot 2 (optional)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p2p1, p2p2 (p2p3, p2p4)	btbond3 (btbond4)	Assigned post deployment
PCIe Slot 10 (optional)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p10p1, p10p2 (p10p3, p10p4)	btbond5 (btbond6)	Assigned post deployment
On-board	1GBase-T	em1	Not applicable	Assigned post deployment

The following table lists the default network interfaces for Oracle Database Appliance X8-2-HA Virtualized Platform.

Table 11-2 Network Interfaces for Oracle Database Appliance X8-2-HA Virtualized Platform

PCIe Network Card Slot	PCIe Network Port	Network Interfaces	Network Bonds	IP Addresses
PCIe Slot 1 (Interconnect)	2 x 10/25GbE SFP28 ports	eth0, eth1	icbond0/priv1	Node 0: 192.168.16.24 Node 1: 192.168.16.25
PCIe Slot 7 (required)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	eth3, eth4 (eth5, eth6)	bond0/net1 (bond1/net2)	Assigned during deployment
PCIe Slot 2 (optional)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	eth7, eth8 (eth9, eth10)	bond2/net3 (bond3/net4)	Assigned during deployment

Table 11-2 (Cont.) Network Interfaces for Oracle Database Appliance X8-2-HA Virtualized Platform

PCIe Network Card Slot	PCIe Network Port	Network Interfaces	Network Bonds	IP Addresses
PCIe Slot 10 (optional)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	eth11, eth12 (eth13, eth14)	bond4/net5 (bond5/net6)	Assigned during deployment
On-board	1GBase-T	eth2	Not applicable	Assigned post deployment

The following table lists the default network interfaces for Oracle Database Appliance bare metal systems on Oracle Database Appliance X8-2S and X8-2M.

Table 11-3 Network Interfaces for Oracle Database Appliance X8-2S and X8-2M Bare Metal Deployment

PCIe Network Card Slot	PCIe Network Port	Network Interfaces	Network Bonds	IP Addresses
PCIe Slot 1	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p7p1, p7p2 (p7p3, p7p4)	btbond1 (btbond2)	btbond1 and btbond2 can be assigned during deployment.
PCIe Slot 2 (optional for Oracle Database Appliance X8-2M)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p2p1, p2p2 (p2p3, p2p4)	btbond3 (btbond4)	Can be assigned during deployment. If not assigned during deployment, then it can be assigned post deployment.
PCIe Slot 8 (optional for Oracle Database Appliance X8-2S)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p8p1, p8p2 (p8p3, p8p4)	btbond3 (btbond4)	Can be assigned during deployment. If not assigned during deployment, then it can be assigned post deployment.
PCIe Slot 10 (optional)	4 x 10GBase-T ports or 2 x 10/25GbE SFP28 ports	p10p1, p10p2 (p10p3, p10p4)	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.
- Creating a Mount Point for NFS Backup Location
  Create a mount point for the Network File System (NFS) location, to set up the NFS backup and recovery option.
- 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 logsarchivelog 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.

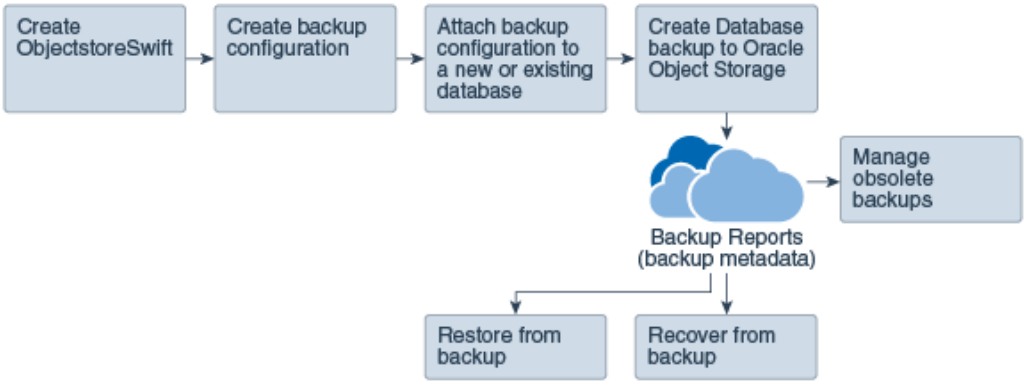
  > **Caution:**
  >
  > Save the password that you use during backups in a safe location, you must provide the password during recovery operations.

- 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

Related Topics

- Creating a Mount Point for NFS Backup Location
  Create a mount point for the Network File System (NFS) location, to set up the NFS backup and recovery option.

Creating a Mount Point for NFS Backup Location

Create a mount point for the Network File System (NFS) location, to set up the NFS backup and recovery option.

Follow these steps to create a mount point for the NFS location:

1. Follow these steps on the source machine:
   a. Create a sharable location on the source machine and give full permissions to this directory.

      ```
      # mkdir /tmp/nfs_test
      chmod 774 /tmp/nfs_test
      ```

   b. Add entries in the `/etc/exports` file in the format *`shared_location destination_IPs (permissions)`*.

      For example:

      ```
      # cat /etc/exports
      /tmp/nfs_test 192.0.2.1(rw,sync)
      /tmp/nfs_test 192.0.2.2(rw,sync)
      ```

   c. Restart the NFS server on the host.

      ```
      # service nfs  restart
      ```

   d. Check the export list for the entries.

      ```
      # showmount -e
      Export list for oda1:
      /tmp/nfs_test 192.0.2.1,192.0.2.2
      ```

2. Follow these steps on the client machine:
   a. Create a client location on the client machine as the `oracle` user.

      ```
      # sudo -E -u oracle mkdir /tmp/client_location
      ```

b. Mount this location with the source location in the format *mount_server:source_folder client_location*.

```
# mount -t nfs 192.0.2.3:/tmp/nfs_test /tmp/client_location
```

c. Check if the mount details are correct:

```
# mount
192.0.2.3:/tmp/nfs_test on /tmp/client_location type nfs
(rw,vers=4,addr=192.0.2.3,clientaddr=192.0.2.2)
```

Specify this NFS location in the backup configuration, either through the Browser User Interface or CLI commands.

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).

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 `update-agentconfig-parameters` command.

```
 # odacli update-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, 2018 4:56:53 PM IST
Message:

Task Name Start Time End Time Status
------------------------------------
```

For more information about the `update-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, 2018 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, 2018 4:56:53 PM IST
HttpProxyPort 80 October 23, 2018 4:56:53 PM IST
HttpsProxyHost Https proxy server host October 23, 2018 12:23:21 AM IST
HttpsProxyPort Https proxy server port October 23, 2018 12:23:21 AM IST
OSPatchRepos Repo list for OS patching October 23, 2018 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 update-agentconfig-parameters
  Use the `odacli update-agentconfig-parameters` command to modify configuration variables used by the appliance.
- 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 -in 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

- Creating a Mount Point for NFS Backup Location
  Create a mount point for the Network File System (NFS) location, to set up the NFS backup and recovery option.

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 and the automatic backups for the archivelog.

   ```
   # 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 update-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.
- 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 update-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.

   ```
   # 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.

Patching Oracle Data Guard primary and standby databases using ODACLI commands

Before patching 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. 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.

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;
   ```

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 dbsystem.
- 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 dbsystem.

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 dbsystem CPU pool is similar to the VM CPU pool, but it only applies to a dbsystem.

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).

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 cpupool. 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 dbsystem automatically creates an internal dbsystem CPU pool associated with this dbsystem. The internal dbsystem 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 dbsystem CPU pool which can be shared among dbsystems. 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 update-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 update-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 update-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 Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to manage shared CPU pool with database 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.
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

- 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-dbsystem
  Use the `odacli modify-dbsystem` command to modify a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- 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.
- 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

After patching the bare metal system, use the command `odacli remap-cpupools` to modify CPU pools.

See the topic *About Enabling NUMA on Oracle Database Appliance* for more information.

Using Browser User Interface to Modify 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. Click **Yes** to confirm the action.
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

- 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-dbsystem
  Use the `odacli modify-dbsystem` command to modify a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- 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.
- 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.

> **Note:**
>
> You cannot enable multi-user access on Oracle Database Appliance DB systems. DB systems do not require role separation since you can create only one database on each DB system.

- 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 Database Systems on Oracle Database Appliance
  Understand how you can deploy KVM-based database systems on Oracle Database Appliance to run your Oracle Database environments.
- Managing Database Systems in KVM Deployment
  Use ODACLI to create, list, describe, start, stop, and delete database 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 two-node 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/about.html#about-kvm

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:

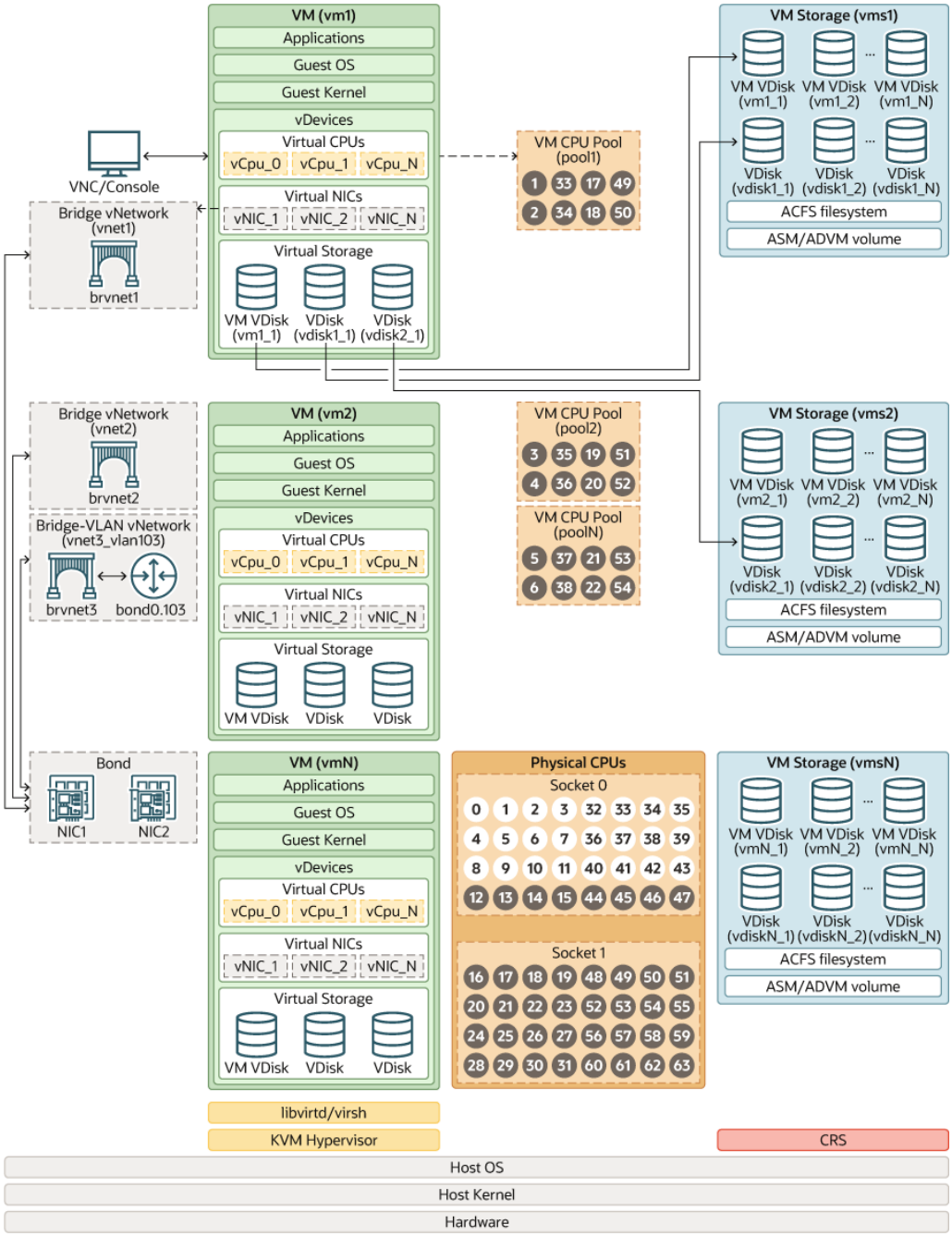
- Deploy database and applications in a single hardware environment.
- Use your CPU cores (and memory) efficiently.
- Use virtual local area networks (VLANs) to provide separate networks to different virtual machines.
- Use VM storage to grow storage for the virtual machine repository.
- Enables easy 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.

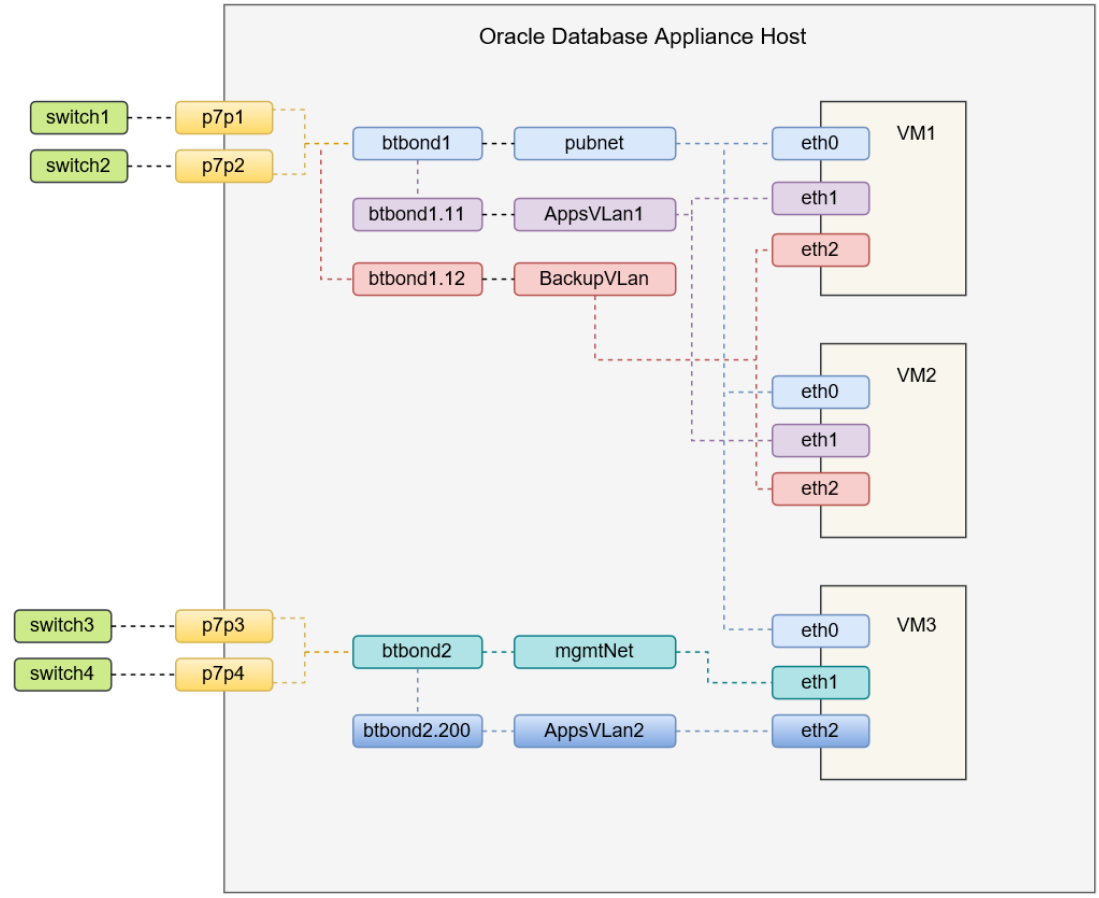
Understand the terminology of the various components you need to set up for an Oracle Database Appliance KVM deployment. The ODACLI tooling handles all the required configuration for your Oracle Database Appliance KVM deployment.

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 Database Systems on Oracle Database Appliance

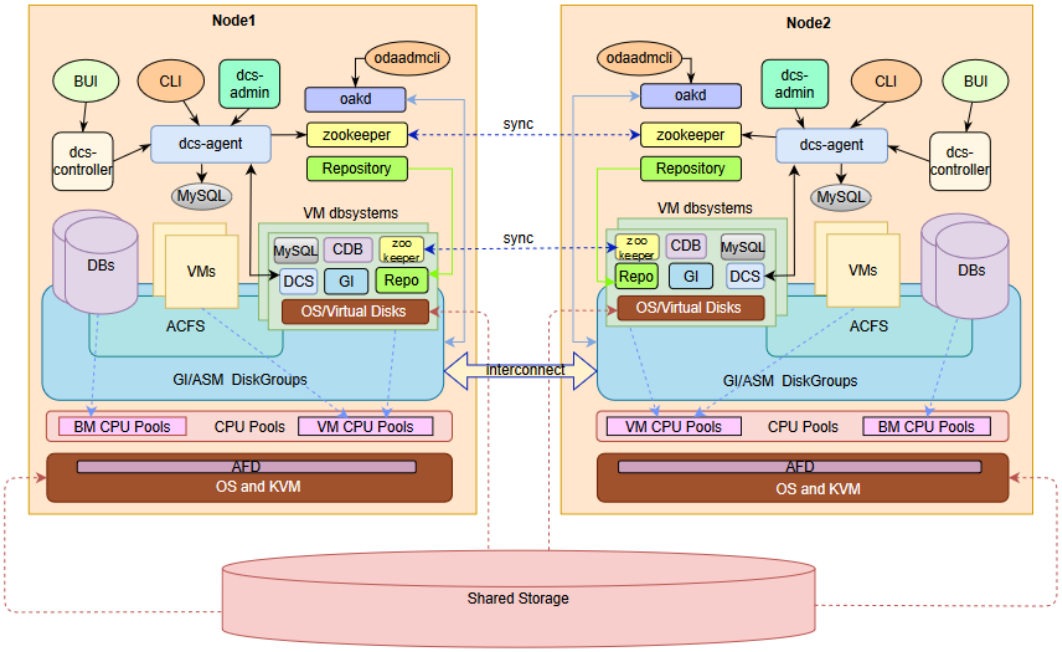
Understand how you can deploy KVM-based database systems on Oracle Database Appliance to run your Oracle Database environments.

About KVM Database Systems

KVM database systems enable hard partitioning for Oracle Database licensing, where each KVM database system has its own CPU pool that is automatically assigned during KVM database system creation. Oracle Database Appliance simplifies the management of KVM database systems with the built-in Browser User Interface (BUI) or ODACLI Command Line Interface (ODACLI).

About KVM Database System Architecture

You can provision and configure a database system on an Oracle KVM. The database can be a single-instance Oracle Database or an Oracle RAC Database with two instances.

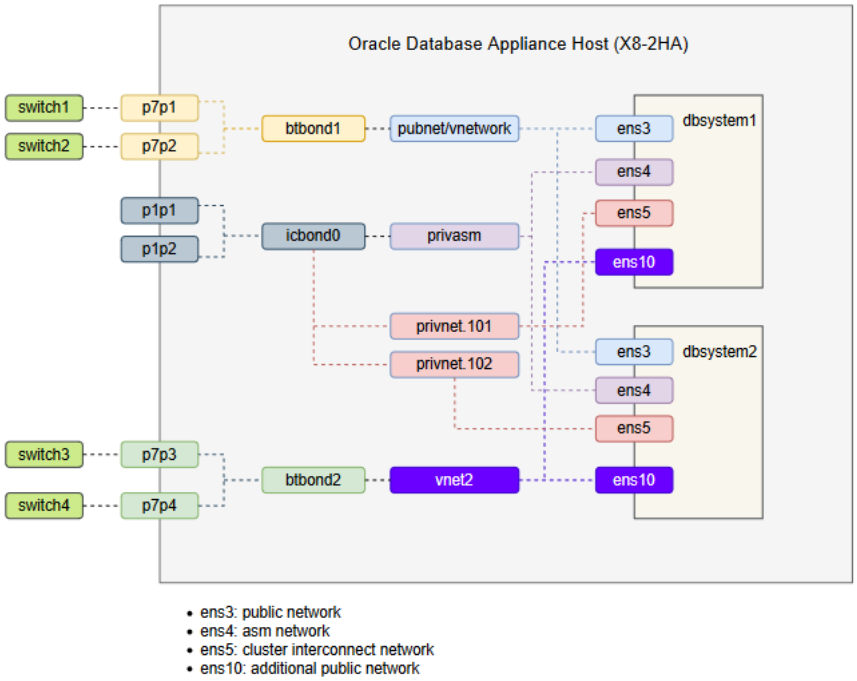


About KVM Database System Networks

Each Oracle Database Appliance database 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 database system, the public network of the database 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 database system. You can also create a different vnetwork and use it as public network for the dbsystem.

- Oracle ASM network and database system cluster interconnect network are created and managed internally. 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 addresses 192.168.16.0/24 and 192.168.17.0/24 are available for use by Oracle Database Appliance database 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 Database System

The following restrictions apply when you deploy KVM database system on Oracle Database Appliance:

- You can create only one database for each database system.
- The shape of the database system is derived from the shape you select for the database. The CPU cores for the database system are set as defined in the database shape. For example, if you select odb4 as the database shape, in which the database uses 4 cores, then the database system is created with 4 cores. odb1s and odb1 shapes are not supported by the database system.
- You cannot use Oracle ACFS for storage within the database system. The database running inside the dbsystem uses Oracle ASM storage. The VM is created on Oracle ACFS based storage.
- Oracle Database Appliance release 19.13 supports creation of databases of Oracle Database releases 21.4, 21.3, 19.13, 19.12, and 19.11 on 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 database system JSON payload, with the attribute `version` for `database`.

- You cannot configure CPU Pools or run VM commands within the database system.

Managing Database Systems in KVM Deployment

Use ODACLI to create, list, describe, start, stop, and delete database systems in an Oracle Database Appliance KVM deployment.

- Creating a Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to create a database system in a KVM deployment.
- Listing Database Systems in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to list database systems in a KVM deployment.
- Describing a Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to describe a database system in a KVM deployment.
- Modifying a Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to modify a database system in a KVM deployment.
- Attaching and Detaching a Network for a Database System in a KVM Deployment
  Use ODACLI commands to attach or detach networks for database system in a KVM deployment.
- Starting a Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to start a database system in a KVM deployment.
- Stopping a Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to stop a database system in a KVM deployment.
- Deleting a Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to delete a database system in a KVM deployment.
- Managing Shared CPU Pool with Database System in a KVM Deployment
  Use ODACLI commands or the Browser User Interface to manage shared CPU pool with database system in a KVM deployment.
- Managing Backup, Restore, and Recovery on a Database System in a KVM Deployment
  Understand the backup, restore, and recovery operations supported on a database system in a KVM deployment.
- Managing Oracle Data Guard on a Database System in a KVM Deployment
  Understand the Oracle Data Guard operations supported on a database system in a KVM deployment.

Creating a Database System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to create a database system in a KVM deployment.

Important:

Oracle Database Appliance release 19.14 supports creation of database versions 19.11, 19.12, 19.13, 19.14, 21.3, 21.4, and 21.5 on DB systems. 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.

NOT_SUPPORTED:

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 19.14, 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 database 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 database system fails. Use the `odacli modify-dbfileattributes` command to set up the file ownerships.

Using ODACLI to Create a KVM Database System

Use the command `odacli create-dbsystem` to create a KVM database system.

Follow these steps:

1. Download the Oracle Database Appliance KVM DB System Image (patch 32451228) 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.

   ```
   p32451228_1914000_Linux-x86-64.zip
   ```

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

   ```
   unzip p32451228_1914000_Linux-x86-64.zip
   odacli-dcs-19.14.0.0.0-ODAVM.zip
   ```

3. Update the repository with the image. The database system version must be the same as the Oracle Database Appliance version.

   ```
   # /opt/oracle/dcs/bin/odacli update-repository -f /tmp/odacli-
   dcs-19.14.0.0.0-ODAVM.zip
   ```

4. Run the `odacli describe-dbsystem-image` command to query the supported Oracle Grid Infrastructure and Oracle Database versions for this database 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            19.14.0.0.0         19.14.0.0.0

GI              19.14.0.0.220118    19.14.0.0.220118
                19.13.0.0.211019    19.13.0.0.211019
                19.12.0.0.210720    19.12.0.0.210720
                19.11.0.0.210420    19.11.0.0.210420
                21.5.0.0.220118     21.5.0.0.220118
                21.4.0.0.211019     21.4.0.0.211019
                21.3.0.0.210720     21.3.0.0.210720

DB              19.14.0.0.220118    19.14.0.0.220118
                19.13.0.0.211019    19.13.0.0.211019
                19.12.0.0.210720    19.12.0.0.210720
                19.11.0.0.210420    19.11.0.0.210420
                21.5.0.0.220118     21.5.0.0.220118
                21.4.0.0.211019     21.4.0.0.211019
                21.3.0.0.210720     21.3.0.0.210720
```

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 19.14, deploy Oracle Grid Infrastructure release 19.14. To deploy Oracle Database release 21.5, deploy Oracle Grid Infrastructure release 21.5.

6. Create the `prov.json` file as per examples provided in this chapter.
Oracle Database 21c 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 dbsystem.

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 prompts `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 prompts `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 dbsystem 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 Database 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 section, 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.
   - **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.
   - **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.
   - (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 section, specify the client access network details, depending on whether it is a single-node or two-node DB system deployment:
   - **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. 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**.
   - 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. Enter 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.
    - (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**.

- **Sys and PDB Admin User Password** and **Confirm Password**: Provide a password for the database.
- **Shape**: Select a database shape from the list.
- 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

- Example JSON File to Create a High-Availability 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`.
- 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 Database Systems in a KVM Deployment

Use ODACLI commands or the Browser User Interface to list database systems in a KVM deployment.

Using ODACLI to List KVM Database Systems

Use the command `odacli list-dbsystems` to list KVM database systems.

```
# odacli list-dbsystems
```

Using Browser User Interface to List KVM Database 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 Database System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to describe a database system in a KVM deployment.

Using ODACLI to Describe a KVM Database System

Use the command `odacli describe-dbsystem` to describe a KVM database system.

```
# odacli describe-dbsystem -n dbsystem_name
```

Using Browser User Interface to Describe a KVM Database 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 Database System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to modify a database system in a KVM deployment.

Using ODACLI to Modify a KVM Database System

Use the command `odacli modify-dbsystem` to change the database shape or the CPU pool association for the KVM database system.

```
# odacli modify-dbsystem -n name [-cp cpu_pool_name] [-no-cp] [-avn
v_network] [-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]
```

You can attach a CPU pool to the database system, or remove the CPU pool from the database system.

You can scale up (increase) or scale down (decrease) the database shape with the `odacli modify-dbsystem` command. The dbsystem is automatically restarted for the new shape to take effect.

```
# odacli modify-dbsystem --name dbsystem1 --shape odb4
```

You can use `odacli modify-dbsystem` to attach or detach the shared dbsystem CPU pool. Attaching shared dbsystem CPU pool removes the internal CPU pool created when the dbsystem is created. When detaching shared CPU pool from the dbsystem, an internal CPU pool is automatically created and attached to the dbsystem. One shared dbsystem CPU pool can be attached to multiple dbsystems. Oversubscribing to shared dbsystem CPU pool is allowed, but there could be potential performance impact if shared CPU pool is oversuscribed.

> **Note:**
>
> The `odacli modify-dbsystem -s shape` command only works on Oracle Database Appliance release 19.11 and later dbsystems. The CPU pool commands are available on Oracle Database Appliance release 19.11 and later dbsystems.

Using Browser User Interface to Modify a KVM Database 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 DB System shape, attach or detach CPU pool for the DB System, attach or detach networks, or enable or disable NUMA for the DB system.
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 dbsystem to change the database shape.

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-dbsystem
  Use the `odacli modify-dbsystem` command to modify a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- 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.
- 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 Database System in a KVM Deployment

Use ODACLI commands to attach or detach networks for database system in a KVM deployment.

Using ODACLI to Attach or Detach a Network to a KVM Database System

You can attach a new vnetwork to the dbsystem 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 dbsystem in a KVM deployment. The vnetwork must exist before you can attach it to the dbsystem.

Follow these steps to attach a network to a DB System:

1. Identify the name of vnetwork bridge to attach to the dbsystem using the `odacli list-vnetworks` and `odacli describe-vnetwork -n` *`vnetwork_name`* commands.
2. Use the `odacli modify-dbsystem` command to attach network to dbsystem:

   ```
   # 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 dbsystem. The SCAN name (sn), SCAN IP address (sip) and Virtual IP address (vip) are used when the network type is `database` or `dataguard`.

To detach a network in the DB system, run the following command:

```
# odacli modify-dbsystem -n name -dvn vnetwork_name
```

Using Browser User Interface to Modify a KVM Database 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 networks for the DB system.
5. Click the Detach Networks field to view the networks, select the network 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 Database System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to start a database system in a KVM deployment.

Using ODACLI to start a KVM Database System

Use the command `odacli start-dbsystem` to start a KVM database system.

```
# odacli start-dbsystem -n dbsystem_name
```

Using Browser User Interface to Start a KVM Database 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 Database System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to stop a database system in a KVM deployment.

Using ODACLI to Stop a KVM Database System

Use the command `odacli stop-dbsystem` to stop a KVM database system.

```
# odacli stop-dbsystem -n dbsystem_name
```

Using Browser User Interface to Stop a KVM Database 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 Database System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to delete a database system in a KVM deployment.

Using ODACLI to Delete a KVM Database System

Use the command `odacli delete-dbsystem` to delete a KVM database system.

```
# odacli delete-dbsystem -n dbsystem_name -f
```

If you do not specify the `-f` option, then the operation deletes the database system and Oracle Clusterware files, and retains the database files. With the `-f` option, the operation deletes the database system, Oracle Clusterware files, and the database files.

Using Browser User Interface to Delete a KVM Database 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 database 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 database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Managing Shared CPU Pool with Database System in a KVM Deployment

Use ODACLI commands or the Browser User Interface to manage shared CPU pool with database system in a KVM deployment.

Using ODACLI to Manage Shared CPU Pool with Database System

Creating a dbsystem automatically creates an internal dbsystem CPU pool associated with this dbsystem. The internal dbsystem CPU pool is created and managed internally. It cannot be created, modified, or deleted manually. Starting with release 19.11, 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 dbsystem CPU pool cannot be local, it must exist on both nodes always.

Associating Shared DB System CPU Pool

A shared dbsystem CPU pool can be associated to a new dbsystem or existing dbsystem. To create a new dbsystem with shared dbsystem CPU pool, add the "cpuPoolName" parameter in the `create-dbsystem json` file. The associated internal CPU pool is 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 dbsystem.

```
# 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 dbsystem CPU pool to an existing dbsystem removes the internal CPU pool associated to the dbsystem. A single shared DB System CPU pool can be shared by multiple DB Systems. Oversubscription to the dbsystem 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 dbsystem:

```
# 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 dbsystem CPU pool type and associated dbsystems. The internal dbsystem CPU pool has type "DB_SYSTEM" and shared dbsystem 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 dbsystem 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 while associated to 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

- odacli modify-dbsystem
  Use the `odacli modify-dbsystem` command to modify a database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).
- 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 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 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 Backup, Restore, and Recovery on a Database System in a KVM Deployment

Understand the backup, restore, and recovery operations supported on a database system in a KVM deployment.

About Backup, Restore, and Recovery on a Database System

Similar to the bare metal systems, you can use ODACLI commands to configure and perform backup, restore, and recovery operations on Database Systems. Since only one database is supported per dbsystem, you must delete the database using `odacli delete-database` in the standby dbsystem before restoring the database to the dbsystem using iRestore. 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 Oracle Data Guard on a Database System in a KVM Deployment

Understand the Oracle Data Guard operations supported on a database system in a KVM deployment.

About Using Oracle Data Guard on a Database System

Similar to the bare metal systems, you can configure and use Oracle Data Guard on Database Systems using ODACLI commands. Since only one database is supported per DB system, you must delete the database in the standby DB system using `odacli delete-database`, before restoring the standby database, and then configuring Oracle Data Guard. 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
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
```

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 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`.

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 Shared: 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 the kickstart configuration file is available over the network that guest VM can access.

For example:

```
odacli create-vm --name odaksvm --memory 4G --source /u01/kvm/images/
OL77_x86_64.iso --vmstorage myvms --size 10G --vcpus 2 --extra-args
"console=tty0 console=ttyS0,115200 ks=http://test: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.

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 Oracle KVM Database 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_example",
        "shape": "odb2",
        "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": "db19",
        "uniqueName": "db19",
        "domainName": "example.com",
        "version": "19.14.0.0.220118",
        "edition": "EE",
        "type": "SI",
        "dbClass": "OLTP",
        "shape": "odb2",
        "role": "PRIMARY",
        "targetNodeNumber": null,
        "enableDbConsole": false,
        "enableFlashStorage": false,
        "enableFlashCache": false,
        "enableUnifiedAuditing": 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": "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"
    }
}
```

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",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
    "database": {
        "name": "db19",
        "uniqueName": "db19",
```

```
        "domainName": "example.com",
        "version": "19.14.0.0.220118",
        "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"
    }
}
```

Example JSON File to Create a High-Availability 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 15-3 JSON File to Create Two-Node Oracle KVM Database System without Role Separation

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. 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",
        "timeZone": "America/Los_Angeles",
        "diskGroup": "DATA",
        "cpuPoolName": "shared_dbsystem_cpupool",
        },
    "database": {
        "name": "dbtest",
        "uniqueName": "dbtest",
        "domainName": "test_domain",
        "version": "19.14.0.0.220118",
        "edition": "SE",
        "type": "SI",
        "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"
        },
        "enableSEHA": true,
        "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 15-4 JSON File to Create Two-Node Oracle KVM Database System with Role Separation

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. 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",
        "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": "19.14.0.0.220118",
            "edition": "SE",
            "type": "SI",
            "dbClass": "OLTP",
            "shape": "odb2",
            "role": "PRIMARY",
            "targetNodeNumber": "0",
            "enableDbConsole": false,
            "enableUnifiedAuditing": 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"
        },
        "enableSEHA": true,
        "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"
    }
}
```

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 Job Commands
  Use the `odacli list-jobs` and `odacli describe-job` commands to display job details.
- 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.
- 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, diskgroups, 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.

Example 16-1 Command-Line Interface Syntax

```
# odacli create-database -h

Usage: create-database [options]
Options:
 * --adminpassword, -m
       Password for SYS,SYSTEM and PDB Admin
```

```
  --backupconfigid, -bi
       Backup Config ID
  --cdb, -c
       Create Container Database (Inverse option: --no-cdb/-no-c)
  --characterset, -cs
       Character Set (default:AL32UTF8)        Default: AL32UTF8
  --databaseUniqueName, -u
       database unique name
  --dbclass, -cl
       Database Class EE: OLTP/DSS/IMDB, SE: OLTP     Default: OLTP
  --dbconsole, -co
       Enable Database Console (Inverse option: --no-dbconsole/-no-co)

  --dbhomeid, -dh
       Database Home ID (Use Existing DB Home)
  --dblanguage, -l
       Datbase Language (default:AMERICAN)        Default: AMERICAN
* --dbname, -n
       Database Name
  --dbshape, -s
       Database Shape{odb1s,odb1,odb2,etc.}        Default: odb1
  --dbstorage, -r
       Database Storage {ACFS|ASM}        Default: ACFS
  --dbterritory, -dt
       Database Territory (default:AMERICA)        Default: AMERICA

  --dbtype, -y
       Database Type: SI        Default: SI
  --help, -h
       get help
  --instanceonly, -io
       Create Instance Only (For Standby)
  --json, -j
       json output
  --nlscharacterset, -ns
       NLS Character Set (default:AL16UTF16)        Default: AL16UTF16
   --no-cdb, -no-c
    Won't create Container Database (Inverse option: --cdb/-c)
    --no-dbconsole, -no-co
    Disable Database Console (Inverse option: --dbconsole/-co)
  --pdbadmin, -d
       Pluggable Database Admin User
  --pdbname, -p
       Pluggable Database Name
  --version, -v
       Database Version
```

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` or `odacliadm 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 16-2 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 -m
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 16-3 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.

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 Command	Roles Required	Changes in command option
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 update-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 update-agentconfig-parameters	ODA-ADMINISTRATOR	No change
odacli delete-agentconfig-parameters	ODA-ADMINISTRATOR	No change
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 Command	Roles Required	Changes in command option
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 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 update-backupconfig	ODA-ADMINISTRATOR, ODA-DB	Must have primary or shared access

ODACLI Command	Roles Required	Changes in command option
odacli update-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 update-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 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 Command	Roles Required	Changes in command option
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 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 configure-dbhome-storage	ODA-ADMINISTRATOR	No change
odacli list-dbhome-storages	ODA-ADMINISTRATOR	No change

ODACLI Command	Roles Required	Changes in command option
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 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 update-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 Command	Roles Required	Changes in command option
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 update-asr	ODA-ADMINISTRATOR	No change
odacli list-osconfigurations	ODA-ADMINISTRATOR	No change
odacli update-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
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 `configure-firstnet` to configure the first network in the appliance after racking and connecting the power and network cables.
- odacli configure-pmem
  Use the `odacli configure-pmem` command to configure PMEM in the system.
- 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 update-network
  Use the `odacli update-network` command to update an existing network configuration.

odacli configure-firstnet

Use the command `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.

Example 16-4 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 16-5 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 ]
```

odacli configure-pmem

Use the `odacli configure-pmem` command to configure PMEM in the system.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli configure-pmem [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.

Usage Notes

- By default, PMEM is configured in `MemoryMode`.
- When PMEM is running in `MemoryMode`, the `sysctl.conf` hugepage parameter can be set up based on different types of databases:
  - OLTP: 50% of total memory
  - In-memory: 50% of total memory
  - DSS: 25% of total memory
  - TimesTen: 80% of total memory

Example 16-6 Configure PMEM

```
# odacli configure-pmem
Configure PMEM in the system.
INFO: PMEM is configured in 'MemoryMode' mode.
```

```
Please select the database type - 1) OLTP 2) In-memory 3) DSS 4) TimesTen
[4] 3
Setup the system to use PMEM in 'MemoryMode' mode for 'DSS' type database
The system will be reboot after finish the setup
Do you want to continue [Y/N] y
```

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` *`interface`* `-p` *`ipaddress`* `-w {Public|Dataguard|Database|Management|Backup|Other} -s` *`subnetmask`* `-g` *`gateway`*`[-h] [-sip] [-sn] [-vs] [-j]`

Parameters

Parameter	Description
`--defaultnetwork, -d`	Identifies the default network.
`--gateway, -g`	Defines the network gateway. The gateway is required for the default network.
`--help, -h`	(Optional) Displays help for using the command.
`--interface, -n`	Defines the name of the network interface.
`--interfacetype, -t`	Defines the network interface type {BOND|PHYSICAL|VLAN|INTERNAL} (non case-sensitive).
`--ipaddress, -p`	Defines the network IP address. This can be specified in the format IPAddress or nodeNumber0:IPAddress0,nodeNumber1:IPAddress1,...
`--json, -j`	(Optional) Displays JSON output.
`--networktype, -w`	Defines the type of network. Options are: {Public|Dataguard|Database|Management|Backup|Other}. The default is Other.
`--name, -m`	Name of the network.
`--no-defaultnetwork, no-d`	Identifies a network as not the default network. Use `--defaultnetwork -d` to identify a default network.
`--subnetmask, -s`	Defines the Network Subnet Mask.
`--scanips, -sip`	Defines the SCAN IP addresses (in the format scanip0,scanip1,...)
`--scanname, -sn`	Defines the SCAN name.
`--nodeNumber, -u`	Defines the node number for High-Availability deployments. **Note:** This option is deprecated and will be desupported in a future release.

Parameter	Description
`--vips, -vs`	List of virtual IP addresses. They can be specified in the format vipname0:nodeNumber0:vip0,vipname1:nodenumber1:vip1, ...
`--vlanid, -v`	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 16-7 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 sfpbond1 -p 192.0.2.15 -w Backup -s
255.255.255.0 -no-d
```

Example 16-8 Creating a VLAN Network

```
odacli create-network -g 10.31.131.129 -n btbond1 -t VLAN -p
0:10.31.131.132,1:10.31.131.133 -m vlan150 -w Database -no-d -sip
10.31.131.134,10.31.131.135 -sn scaoda702c1-vlan150-scan -s
255.255.255.128 -vs scaoda702c1n1-vlan150-
vip:0:10.31.131.136,scaoda702c1n2-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 [-m] [-h] [-j]
```

Parameters

Parameter	Description
`--id, -i`	Defines the network identity.
`--name, -m`	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 16-9 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" : "December 8, 2021 23:14:32 PM EDT",
     "endTime" : "December 8, 2021 23:14:32 PM EDT",
     "status" : "Running",
     "taskDescription" : null,
     "parentTaskId" : "TaskSequential_135",
     "jobId" : "c26d217e-419b-4a91-8680-7b06bcfe9828",
     "tags" : [ ],
     "reportLevel" : "Info",
     "updatedTime" : "December 8, 2021 23:14:32 PM EDT"
   },{
   "taskId" : "TaskZJsonRpcExt_142",
     "taskName" : "Setting up Network",
     "taskResult" : "Network setup success",
     "startTime" : "December 8, 2021 23:14:32 PM EDT",
     "endTime" : "December 8, 2021 23:14:32 PM EDT",
     "status" : "Success",
     "taskDescription" : null,
     "parentTaskId" : "TaskParallel_141",
     "jobId" : "c26d217e-419b-4a91-8680-7b06bcfe9828",
     "tags" : [ ],
     "reportLevel" : "Info",
     "updatedTime" : "December 8, 2021 23:14:32 PM EDT"    } ],
```

```
  "createTimestamp" : "December 8, 2021 23:14:32 PM EDT",
  "description" : "Network service delete",
  "updatedTime" : "December 8, 2021 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 id [-h]
```

Parameters

Parameter	Description
-id	Defines the network interface name.
--help, -h	(Optional) Displays help for using the command.

Usage Notes

You cannot delete the network interface card in slot 7.

Example 16-10 Deleting a Network Interface

```
[root@oak ~]/opt/oracle/dcs/bin/odacli delete-networkinterface -m
btbond4
{
"jobId" : "a81066e2-417e-4df6-b810-08df24c646c1",
"status" : "Running",
"message" : null,
"reports" : ,
"createTimestamp" : "December 8, 2021 00:01:09 AM UTC",
"resourceList" : ,
"description" : "Network Interface Deletion with name btbond4",
"updatedTime" : "December 8, 2021 00:01:09 AM UTC"
}
```

Example 16-11 Deleting btbond1 Network Interface is Not Allowed

```
[root@oak ~]# /opt/oracle/dcs/bin/odacli delete-networkinterface -m
btbond1
DCS-10001:Internal error encountered: Delete operation is not allowed
on network interface : btbond1.
```

Example 16-12 Deleting Network Interface When Network Exists is Not Allowed

```
[root@oak ~]# opt/oracle/dcs/bin/odacli delete-networkinterface -m 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 [-j][-u][-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, -m`	Specifies the network name.
`--nodeNumber, -u`	(Optional) Displays the network interface information for the specified node. **Note:** This option is deprecated and will be desupported in a future release.

Example 16-13 Displaying Network Details

Enter the following command to display the details of the public network:

```
# odacli describe-network -m 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, 2020 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:  scaoda1-scan
               Scan Ips:  10.31.117.64 10.31.117.65
                   Vips:  [scaoda1-vip on node0: 10.31.116.61, scaoda2-
vip on node1: 10.31.116.63]
   Associated Databases: odacn db122a db122b db122c db122d
```

Example 16-14 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 -u 1

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, 2018 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 [-j][-h][-u]
```

Parameters

Parameter	Description
`--id, -i`	Identifies the network interface 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.

Parameter	Description
`--nodeNumber, -u`	(Optional) Displays the network interface information for the specified node. **Note:** This option is deprecated and will be desupported in a future release.

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 number option (`-u`).

Example 16-15 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 -u 1

Network Interface details
----------------------------------------------------------------
                     ID:  fe1bf0a7-f56e-44cd-9a84-f374c0aa4b61
                   Name:  eth2
                    NIC:  eth2
                   Type:  PHYSICAL
                Members:  eth2
        Slave Interface:  true
                Created:  October 16, 2018 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  [-j] [-h]
```

Parameters

Parameter	Description
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.
`--nodeNumber, -u`	(Optional) Displays the network interface information for the specified node. **Note:** This option is deprecated and will be desupported in a future release.

Example 16-16 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  [-j] [-h] [-u]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.
`--nodeNumber`, `-u`	(Optional) Displays the network interface information for the specified node. **Note:** This option is deprecated and will be desupported in a future release.

Example 16-17 Displaying a List of Network Interfaces

Use the `odacli list-networkinterfaces` command to display a list of network interfaces:

```
# odacli list-networkinterfaces -u 1
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 update-network

Use the `odacli update-network` command to update an existing network configuration.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To update a network:

```
odacli  update-network -i id [-p IP address] [-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.
`--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,...
`--json, -j`	(Optional) Displays JSON output.
`--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.
`--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.

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 16-18 Updating a Network

The following example updates network ID `192.0.0.2` and designates the network as a backup network:

```
# odacli update-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:**
>
> 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.

- 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 16-19

```
[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 16-20

```
# 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 16-21

```
[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.
`--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 16-22

```
[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.
`--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 16-23

```
[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, 2021 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, 2021 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 16-24

```
[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 16-25

```
[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 16-26

```
[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 16-27

```
[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 16-28

```
[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 16-29

```
[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 16-30

```
[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 16-31

```
[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 16-32

```
[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 16-33

```
[odaadmin@oda1 bin]$ odacli list-user-operations

Operation Name           Operation Description      Is Active
---------------------------------
--------------------------------------------------------------------------------
----------------- ----------
migrate-ovm2kvm          Migrate OVM to KVM         true
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 16-34

```
[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.

Parameter	Description
`--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 16-35

```
[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.
`--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 16-36

```
[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.

Parameter	Description
`--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 16-37

```
[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 16-38

```
[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 16-39

```
[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 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 16-40

```
[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 update-agentconfig-parameters
  Use the `odacli update-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-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 components for Zookeeper 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 and Zookeeper.
- odacli update-registry
  Use the `odacli update-registry` command to update the registry of components 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.
- 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 [-cl` *`clones`*`] | [-v` *`version`* `[-comp` *`component`*`]]| [-l` *`local`*`|-n` *`NodeID`*`]`

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, -n`	(Optional) Cleans up the repository on the specified nodes.
`--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 16-41 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, 2019 06:44:28 AM UTC",
"resourceList" : ,
"description" : "Cleanup patchrepos",
"updatedTime" : "April 17, 2019 06:44:28 AM UTC"
```

Example 16-42 Deleting Clone Files

```
# odacli cleanup-patchrepo -cl
{
"jobId" : "5d8549a2-1a5e-4b4f-9867-c1f671c659c4",
"status" : "Created",
"message" : null,
"reports" : ,
"createTimestamp" : "April 17, 2019 06:43:45 AM UTC",
"resourceList" : ,
"description" : "Cleanup patchrepos",
"updatedTime" : "April 17, 2019 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 [-d] [-i] [-f] [-l] [-n] [-v] [-h] [-s] [-
r] [-st] [-c os [-local]]
```

Parameters

Parameter	Description
`--dbhome, -d`	(Optional) Specifies the database home component for running the pre-checks.
`--dbhomeid, -i`	(Optional) Specifies the IDs of the database homes for running the pre-checks.
`--files, -f`	(Optional) Specifies files required to update repository, comma-separated if multiple files are required.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--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, -n`	(Optional) Runs patch pre-checks on specified nodes.
`--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.

Usage Notes

Use the `odacli create-prepatchreport` command to generate a pre-check report.

Example 16-43 Creating Pre-Check Report

```
# odacli create-prepatchreport -v 19.14.0.0.0 -s

Job details
----------------------------------------------------------------
                     ID: e07993d5-0a06-44e7-819a-36d708c52bb6
            Description: Patch pre-checks for [OS, ILOM, GI, ORACHKSERVER]
                 Status: Created
                Created: April 20, 2020 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
---------------------------------------- ----------------------------------
----------------------------
```

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

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli describe-prepatchreport -i
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--jobid, -i`	Specifies the Job ID for the pre-check report.

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

```
# odacli describe-prepatchreport -i e07993d5-0a06-44e7-819a-36d708c52bb6

Patch pre-check report
------------------------------------------------------------------------
                 Job ID:  e07993d5-0a06-44e7-819a-36d708c52bb6
            Description:  Patch pre-checks for [OS, ILOM, GI, ORACHKSERVER]
                 Status:  FAILED
                Created:  April 20, 2020 1:01:59 AM PDT
                 Result:  One or more pre-checks failed for [OS, ORACHK]

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

Pre-Check                      Status   Comments
------------------------------ --------
--------------------------------------
__OS__
Validate supported versions    Success   Validated minimum supported
versions.
Validate patching tag          Success   Validated patching tag:
19.6.0.0.0.
Is patch location available    Success   Patch location is
available.
Validate if ODABR is installed Failed    ODABR utility is not installed on
                                         node:
scaoda8m004.
Space checks for OS upgrade    Success   Validated space
checks.
Install OS upgrade software    Success   Extracted OS upgrade patches
into
                                         /root/oda-upgrade. Do not remove
this
                                         directory untill OS upgrade
completes.
Verify OS upgrade by running   Success   Results stored
in:
preupgrade checks                        '/root/preupgrade-
results/
```

```
preupg_results-200420010731.tar.gz' .
                                          Read complete report
file
                                          '/root/preupgrade/
result.html' before
                                          attempting OS
upgrade.
Validate custom rpms installed  Success   No additional RPMs found
installed on

node:scaoda8m004.
Scheduled jobs check            Failed    Scheduled jobs found.
Disable
                                          scheduled jobs before
attempting OS
                                          upgrade.
__ILOM__
Validate supported versions     Success   Validated minimum supported
versions.
Validate patching tag           Success   Validated patching tag:
19.6.0.0.0.
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

__GI__
Validate supported GI versions  Success   Validated minimum supported
versions.
Validate available space        Success   Validated free space
under /u01
Verify DB Home versions         Success   Verified DB Home
versions
Validate patching locks         Success   Validated patching
locks
Validate clones location exist  Success   Validated clones
location
Validate ODABR snapshots exist  Success   No ODABR snaps found on the
node.

__ORACHK__
Running orachk                  Failed    Orachk validation
failed: .
Verify diagsnap configuration   Failed    diagsnap or pstack are
configured to
                                          collect first failure
diagnostic
Software home                   Failed    Software home check
failed
```

Example 16-47 Example of a successful patch pre-checks report

```
# odacli describe-prepatchreport -i aec9373c-96aa-43ce-9aae-8091ec9cd4eb
Patch pre-check report
------------------------------------------------------------------
                 Job ID:  aec9373c-96aa-43ce-9aae-8091ec9cd4eb
            Description:  Pre-Check report for patching [DB]
                 Status:  COMPLETED
                 Result:  All pre-checks succeeded
Node Name
---------------
node1
Pre-Check                      Status     Comments
------------------------------ ----------
-----------------------------------
__DB__
Validate patching tag          Success     Validated patching tag:
12.2.1.2.0
Validate available space       Success     Validated free space
required

under /u01
Is system provisioned          Success     Verified system is
provisioned
Is patch location available    Success     Patch location is
available
Validate minimum agent version Success     Validated minimum agent
version
Verify DBHome patch tag        Success     Verified DB Home patch
tag
Is GI upgraded                 Success     Validated GI is
upgraded
Patch location validation      Success     Successfully validated
location
Patch verification             Success     Patch 26710464 not applied on
DB

home
Is patch rollback required     Success     No DB patch is required to
rollback
Opatch updation                Success     Successfully updated the opatch
in

DbHome
Patch conflict check           Success     No patch conflicts found on
DBHome
```

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 16-48 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
Report ID	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 16-49 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 16-50 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 16-51 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 in Oracle Database Appliance, then the output displays the version details for each of the installed homes. 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][-n]
```

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, -n`	(Optional) Describes the components for a specific node on high availability (HA) systems. This option is not needed to display component details on single node systems. {0|1}
`--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 16-52 Displaying local and shared storage components and the required updates

```
[root@node1 ~]# odacli describe-component
System Version
---------------
19.13.0.0.0

System node Name
---------------
node1

Local System Version
---------------
19.13.0.0.0

Component                                 Installed Version    Available
Version
----------------------------------------- --------------------
```

```
--------------------
OAK                                       19.13.0.0.0           up-to-
date

GI                                        19.13.0.0.211019      up-to-
date

DB {
[ OraDB19000_home1,OraDB19000_home2,      19.13.0.0.211019      up-to-
date
OraDB19000_home3 ]

[ OraDB12201_home1 ]                      12.2.0.1.211019       up-to-
date
[ OraDB12102_home1 ]                      12.1.0.2.211019       up-to-
date
}

DCSCONTROLLER                             19.13.0.0.0           up-to-
date

DCSCLI                                    19.13.0.0.0           up-to-
date

DCSAGENT                                  19.13.0.0.0           up-to-
date

DCSADMIN                                  19.13.0.0.0           up-to-
date

OS                                        7.9                   up-to-
date

ILOM                                      5.0.2.24.r141466      up-to-
date

BIOS                                      52050300              up-to-
date

SHARED CONTROLLER FIRMWARE                16.00.08.00           up-to-
date

FIRMWAREEXPANDER                          0310                  up-to-
date

LOCAL DISK FIRMWARE                       1132                  up-to-
date

SHARED DISK FIRMWARE {
[ c0d0,c0d1,c0d2,c0d3,c0d4,c0d5,c1d0,     RXA0                  up-to-
date
c1d1,c1d2,c1d3,c1d4,c1d5 ]

[ c0d6,c0d7,c0d8,c0d9,c0d10,c0d11,c0d12,  A144                  up-to-
date
```

```
c0d13,c0d14,c0d15,c0d16,c0d17,c0d18,

c0d19,c0d20,c0d21,c0d22,c0d23,c1d6,c1d7,

c1d8,c1d9,c1d10,c1d11,c1d12,c1d13,c1d14,

c1d15,c1d16,c1d17,c1d18,c1d19,c1d20,

c1d21,c1d22,c1d23 ]

[ c2d0,c2d1 ]                           1132                up-to-
date
}
HMP                                     2.4.8.0.600         up-to-
date
```

Example 16-53 Displaying patch details for components for High-Availability Oracle Database Appliance Systems for a release

```
# /opt/oracle/dcs/bin/odacli describe-component -v
System Version
---------------
18.8.0.0.0

System node Name
---------------
node1

Local System Version
---------------
18.8.0.0.0

Component                               Installed Version    Available
Version
--------------------------------------- --------------------
--------------------
OAK                                      18.8.0.0.0           up-to-date

GI                                       18.8.0.0.191015      up-to-date

DB                                       18.8.0.0.191015      up-to-date

DCSAGENT                                 18.8.0.0.0           up-to-date

ILOM                                     4.0.4.41.r130359     up-to-date

BIOS                                     25080100             up-to-date

OS                                       6.10                 up-to-date

FIRMWARECONTROLLER                       11.05.03.00          up-to-date

FIRMWAREEXPANDER                         0018                 001e
```

```
FIRMWAREDISK {
[ c0d0,c0d1 ]                                A7E0                 up-to-
date
[ c1d0,c1d1,c1d2,c1d3,c1d4,c1d5,c1d6,        A72A                 a7e0
c1d7,c1d8,c1d9,c1d10,c1d11,c1d12,c1d13,
c1d14,c1d15,c1d16,c1d17,c1d18,c1d19,
c2d0,c2d1,c2d2,c2d3,c2d4,c2d5,c2d6,c2d7,
c2d8,c2d9,c2d10,c2d11,c2d12,c2d13,c2d14,
c2d15,c2d16,c2d17,c2d18,c2d19 ]
[ c1d20,c1d21,c1d22,c2d20,c2d21,c2d22 ]   944A                 up-to-
date
}

System node Name
---------------
node2

Local System Version
---------------
18.8.0.0.0

Component                                Installed Version
Available Version
---------------------------------------- --------------------
--------------------
OAK                                       18.8.0.0.0           up-to-
date

GI                                        18.8.0.0.191015      up-to-
date

DB                                        18.8.0.0.191015      up-to-
date

DCSAGENT                                  18.8.0.0.0           up-to-
date

ILOM                                      4.0.2.27.b.r125869
4.0.4.41.r130359

BIOS                                      25060400
25080100

OS                                        6.10                 up-to-
date

FIRMWARECONTROLLER                        11.05.03.00          up-to-
date

FIRMWAREEXPANDER                          0018                 001e

FIRMWAREDISK {
[ c0d0,c0d1 ]                             A7E0                 up-to-
date
[ c1d0,c1d1,c1d2,c1d3,c1d4,c1d5,c1d6,     A72A                 a7e0
```

```
c1d7,c1d8,c1d9,c1d10,c1d11,c1d12,c1d13,
c1d14,c1d15,c1d16,c1d17,c1d18,c1d19,
c2d0,c2d1,c2d2,c2d3,c2d4,c2d5,c2d6,c2d7,
c2d8,c2d9,c2d10,c2d11,c2d12,c2d13,c2d14,
c2d15,c2d16,c2d17,c2d18,c2d19 ]
[ c1d20,c1d21,c1d22,c2d20,c2d21,c2d22 ]   944A                  up-to-date
}
```

Example 16-54 List DB Home Details

```
# odacli describe-component -d

System Version
---------------
18.7.0.0.0

System node Name
---------------
node1

Local System Version
---------------
18.7.0.0.0

Component                                Installed Version    Available
Version
---------------------------------------- --------------------
--------------------
DB                                        12.2.0.1.190716       up-to-date

System node Name
---------------
node2

Local System Version
---------------
18.7.0.0.0

Component                                Installed Version    Available
Version
---------------------------------------- --------------------
--------------------
DB                                        12.2.0.1.190716       up-to-date
```

Example 16-55 List the Server Components and Versions

```
# odacli describe-component -s
System Version
---------------
18.7.0.0.0

System node Name
---------------
node1
```

```
Local System Version
---------------
18.7.0.0.0

Component                                Installed Version
Available Version
---------------------------------------- --------------------
--------------------
OAK                                       18.7.0.0.0           up-to-
date

GI                                        18.7.0.0.190716      up-to-
date

ILOM                                      4.0.4.38.r130206     up-to-
date

BIOS                                      52010400             up-to-
date

OS                                        6.10                 up-to-
date

System node Name
---------------
node2

Local System Version
---------------
18.7.0.0.0

Component                                Installed Version
Available Version
---------------------------------------- --------------------
--------------------
OAK                                       18.7.0.0.0           up-to-
date

GI                                        18.7.0.0.190716      up-to-
date

ILOM                                      4.0.4.38.r130206     up-to-
date

BIOS                                      52010400             up-to-
date

OS                                        6.10                 up-to-
date
```

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.

Parameter	Description
`--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    December 8, 2021 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  December 8, 2021 4:14:10 AM
UTC
HttpProxyPort   Http proxy server port   December 8, 2021 4:14:10 AM
UTC
HttpsProxyHost   Https proxy server host  December 8, 2021 4:14:10 AM
UTC
HttpsProxyPort  Https proxy server port  December 8, 2021 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 update-agentconfig-parameters

Use the `odacli update-agentconfig-parameters` command to modify configuration variables used by the appliance.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli update-agentconfig-parameters
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--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.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--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.

Usage Notes

Use the `odacli update-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 update-agentconfig-parameters -n HttpsProxyPort -v 80 -d "Http
proxy port"
-n HttpsProxyPort -v 80 -d "Https proxy port" -u

update-agentconfig-parameters -n HttpsProxyPort -n HttpProxyPort -r

update-agentconfig-parameters -n HttpsProxyHost -v 90 -a

update-agentconfig-parameters -n OSPatchRepos -v
```

Example 16-61 Setting Oracle RDBMS Parameters

```
# odacli update-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                       November 22, 2021 5:24:45 PM UTC
```

Example 16-62 Updating the RDBMS Parameter for all DB systems in the Bare Metal System

```
# odacli update-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                          November 10, 2021
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                          November 10, 2021
12:08:15 AM UTC
```

You can set the value for a specific DB system:

```
# odacli update-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                          November 10, 2021
12:08:15 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 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 update-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:  December 8, 2021 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-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` *`dbhomeid`* `-v` *`version`* `[-f] [-imp] [-p] [-j] [-h]`

Parameters

Parameter	Description
`--dbhomeid, -i`	Defines the Oracle Database Home to update.
`--help-h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--precheck, -p`	(Optional) Analyzes the patch.
`--force, -f`	(Optional) Ignores precheck error and forces the operation.
`--version, -v`	Defines the Oracle Database Appliance version to update. For example, 19.11.0.0.0.
`----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.

Example 16-65 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 19.14.0.0.0
```

odacli update-dcsadmin

Use the `odacli update-dcsadmin` command to update the DCS admin components for Zookeeper upgrade.

File Path

$ORACLE_HOME/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 16-66 Updating DCS Admin

```
# odacli update-dcsadmin -v 18.7.0.0.0
{
  "jobId" : "4c238b93-e641-4f9d-9f86-93d0574dd234",
  "status" : "Created",
```

```
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "July 12, 2019 03:11:53 AM UTC",
  "resourceList" : [ ],
  "description" : "DcsAdmin patching",
  "updatedTime" : "July 12, 2019 03:11:53 AM UTC"
}
```

odacli update-dcsagent

Use the `odacli update-dcsagent` command to update the agent.

File Path

$ORACLE_HOME`/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 16-67 Updating the Agent

To update the `dcs-agent` to version 19.14:

```
# odacli update-dcsagent -v 19.14
{
  "jobId" : "77e454d3-eb68-4130-a247-7633f8d6192b",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
```

```
  "createTimestamp" : "December 8, 2021 14:09:24 PM CST",
  "description" : "DcsAgent patching",
  "updatedTime" : "December 8, 2021 14:09:24 PM CST"
}
```

odacli update-dcscomponents

Use the `odacli update-dcscomponents` command to update the DCS components such as the DCS agent and Zookeeper.

File Path

$ORACLE_HOME/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 Zookeeper.

Example 16-68 Updating DCS Components

```
# odacli update-dcscomponents -v 18.7.0.0.0
{
  "jobId" : "4c238b93-e641-4f9d-9f86-93d0574dd234",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "July 12, 2019 03:11:53 AM UTC",
  "resourceList" : [ ],
  "description" : "DcsComponents patching",
  "updatedTime" : "July 12, 2019 03:11:53 AM UTC"
}
```

odacli update-registry

Use the `odacli update-registry` command to update the registry of components when you apply patches manually.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli update-registry  -n component  [-f] [-j] [-h]
```

Parameters

Parameter	Description
`--component`, `-n`	Defines the Oracle Database Appliance component for the registry update. The values can be `system {gihome, dbnode, sysinstance}` or `all {dbhome, db, dbstorage, asr}`. You can also specify the individual component to be refreshed. **Note:** Refresh system components before you refresh all components.
`-f`	(Optional) Forces the refresh of the specified component even if it already exists in the appliance registry metadata.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.

Usage Notes

The `update-registry` command updates the registry of components when you apply patches manually.

The components must be updated in the order: system, dbhome, db,dbstorage, otherwise the `update-registry` command displays an error. To update all components, you can specify the component as `all` in the `update-registry` command.

The `update-registry` command rediscovers only the following DCS components resources:

- System resources : Oracle Grid Infrastructure (gihome), Oracle Database node (dbnode), System instance (sysinstance)
- Database Home (dbhome)
- Database (db)
- Database storage (dbstorage)
- Oracle ASR

The `update-registry` command does not rediscover any other components not mentioned in the 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.

Example 16-69 Updating the Registry with the -force option

```
# odacli update-registry -n system
DCS-10112:Specified components are already discovered.

Using force flag option to rediscover and update the sytem components though
it already exists in appliance registry
# odacli update-registry -n system -f

Job details
----------------------------------------------------------------
                     ID: 752b56d8-2bcd-4a29-ab96-196925fc5c13
            Description: Discover System Components : system
                 Status: Created
                Created: July 31, 2019 1:02:51 PM UTC
                Message:

Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------

# odacli describe-job -i 752b56d8-2bcd-4a29-ab96-196925fc5c13

Job details
----------------------------------------------------------------
                     ID: 752b56d8-2bcd-4a29-ab96-196925fc5c13
            Description: Discover System Components : system
                 Status: Success
                Created: July 31, 2019 1:02:51 PM UTC
                Message:

Task Name Start Time End Time Status
---------------------------------------- ----------------------------------
---------------------------------- ----------
Rediscover SysInstance July 31, 2019 1:02:51 PM UTC July 31, 2019 1:03:24 PM
UTC Success
Rediscover DBNode July 31, 2019 1:03:24 PM UTC July 31, 2019 1:03:24 PM UTC
Success
Rediscover GiHome July 31, 2019 1:03:24 PM UTC July 31, 2019 1:03:26 PM UTC
Success
```

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 16-70 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-19.14.0.0.0-
date-server.zip,/tmp/oda-sm-19.14.0.0.0-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" : "December 8, 2021 14:13:45 AM CST",
  "description" : "Repository Update",
  "updatedTime" : "December 8, 2021 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   December 8, 2021
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 [-p] [-c] -v version [-f][-ng] [-j] [-h]
```

Parameters

Parameter	Description
`--precheck, -p`	(Optional) Analyzes the patch.
`--version, -v`	(Mandatory) Defines the version to update.
`--component,-c`	(Optional) Component that is to be upgraded. Supported value is OS.

Parameter	Description
`--nogrid,-ng`	(Optional) Skips upgrade of Oracle Grid Infrastructure.
`--skip-orachk, -sko`	(Optional) Skips ORAchk errors and continues patching operation.
`--force, -f`	(Optional) Ignores precheck error and forces the patching operation.
`--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.

> **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 16-71 Updating the Server

Run the `odacli update-server` command to update the server to 19.14.0.0.0. On a high-availability system, run the command on one node. The patches are applied automatically to both nodes.

```
# odacli update-server -v 19.14.0.0.0
{
  "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" : "December 8, 2021 14:13:45 PM CST",
  "resourceList" : [ ],
  "description" : "Server Patching",
```

```
  "updatedTime" : "December 8, 2021 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 16-72 Updating the Storage

```
# odacli update-storage -v 19.14.0.0.0
```

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 16-73 Running odaadmcli orachk

```
[root@oak bin] # ./odaadmcli orachk
INFO: 2019-09-04 16:41:26: Running orachk under /opt/
oracle.SupportTools/orachk
```

Example 16-74 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 set-credential
  Use the `odacli set-credential` command to change the `oda-admin` user credentials.
- 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 list-featuretracking
  Use the `odacli list-featuretracking` command to display the latest feature tracking report.

odacli set-credential

Use the `odacli set-credential` command to change the `oda-admin` user credentials.

Syntax

To reset the `oda-admin` user credentials in interactive mode:

```
odacli set-credential --username username [-j] [-h]
```

Parameters

Parameter	Description
`--username`, `-u`	User name required to access the Oracle Appliance Manager Browser User Interface. The default user name is `oda-admin`.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

Only `root` user can reset the `oda-admin` user credentials.

Example 16-75 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

$ORACLE_HOME/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.

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 16-76 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 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 16-77 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 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 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 BackupReport.
- 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 update-backupconfig
  Use the command `odacli update-backupconfig` to update a backup configuration.
- odacli update-objectstoreswift
  Use the command `odacli update-objectstoreswift` to change the credentials for the ObjectStore account.
- odacli update-schedule
  Use the command `odacli update-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 | -in 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.
`--dbName, -in`	Defines the Database name.
`--dbid, -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.
`--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.
- 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 16-78 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 16-79 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 16-80 Create a Backup of only the TDE Wallet

```
# odacli create-backup -in testdb -c TDEWallet
```

Example 16-81 Create a Backup of the database and TDE Wallet

```
# odacli create-backup -in 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 16-82 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 16-83 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 16-84 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 16-85 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 delete-backup

Use the command `odacli delete-backup` to delete obsolete backups and backup reports.

File Path

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli  delete-backup -i Database Resource ID -in [-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.
`--dbid, -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.
`--dbName, -in`	Defines the Database resource name.

Parameter	Description
`--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 16-86 Delete a Level 0 or Level 1 Backup

```
# odacli delete-backup -i 20576eb1-bc32-4e34-bf97-fda0b60ca15b
```

Example 16-87 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

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli  delete-backupconfig  -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).
`--backupconfigname, -in`	Defines the Backup Config name.

Parameter	Description
`--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 16-88 Deleting a Backup Configuration

Delete a backup configuration named `production`

```
# odacli delete-backupconfig -in 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 16-89 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" : "January 08, 2018 12:43:14 PM UTC",
  "pitrSCN" : "1175058",
  "resetLogsTimeStamp" : "January 08, 2018 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 delete-objectstoreswift

Use the command `odacli delete-objectstoreswift` to delete the credentials for the ObjectStore account.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
# odacli  delete-objectstoreswift -i Object Store Swift id -in 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.
`--objectstoreswiftid, -i`	Defines the Object Store Swift identifier (ID).
`--objectstoreswiftName, -in`	Defines the Object Store Swift name.

Usage Notes

You cannot delete the Object Store credentials if they are attached to a backup configuration.

Example 16-90 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

$ORACLE_HOME`/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.
`--scheduleid, -id`	Defines the schedule with an identifier (ID).

Example 16-91 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 BackupReport.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli irestore-database -r [-n] [-bp] [-cl] [-nn] [-co] [-s] [-ro] [-dr] [-y] [-c] [-oid] [-tf] [-bl] [-t] [-tl] [-cp] [-dh] [-dn] [-de] [-f] [-rd] [-on] [-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.
`--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.
`--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.

Parameter	Description
`--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.
`--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.
`--cpupool, -cp`	(Optional) Specifies the CPU pool name.

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.

- 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.

Example 16-92 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 16-93 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 16-94 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 16-95 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 16-96 Restoring to a standby database when Oracle Data Guard is configured

```
# odacli irestore-database -r backup_report.json -u dbUniqueName -ro STANDBY
```

Example 16-97 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, 2020 11:20:08 AM UTC",
  "resourceList" : [ ],
  "description" : "Database service recovery with db name: irmendb3",
  "updatedTime" : "August 10, 2020 11:20:08 AM UTC"
}
```

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 16-98 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 16-99 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 16-100 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

$ORACLE_HOME/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.

Example 16-101 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
---------------------------------------- -------------------------
-------------------------------------------------
------------------------------ --------
201672f6-c80c-4ff9-99be-2bc8be480e66     metastore maintenance     internal
metastore maintenance                     0 0 0 1/1 * ? *
true
e86ff1e9-8607-41f8-8d0a-cf59dc1e52a9     AgentState metastore cleanup
```

```
internal agentstateentry metastore maintenance     0 0 0 1/1 * ?
*                 true
17640773-aa8b-4b4f-b170-3a68d5329cbb     bom maintenance           bom
reports generation                              0 0 1 ? * SUN
*                 true
cae98cb1-e8ab-48c6-aaf9-22d9ab4097b1     Big File Upload Cleanup
clean up expired big file uploads.                0 0 1 ? * SUN
*                 true
745333ad-8845-4444-b591-7ca7d19a4d91     feature_tracking_job
Feature tracking job                              0 0 20 ? * WED
*                true
42b6894f-f680-4316-9958-f4ecf501d781     Log files Cleanup
Auto log file purge bases on policy               0 0 3 1/1 * ?
*                 true
```

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 16-102 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] [-i] [-h] [-j] [-r] [-t] [-p] [-s] [-
in] [-tp] [-tl] [-tz] [-cp]
```

Parameters

Parameter	Description
`--backupReport`, `-br`{Regular-L0|Regular-L1|Longterm}	(Optional) JSON input for a backup report.
`--dbName`, `-in`	Defines the Database Name.
`--dbid`, `-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 16-103 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 16-104 Recovering a Database to the latest

```
# odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t
Latest -p
```

Example 16-105 Recovering a Database to an SCN

```
# odacli recover-database -i b5fc646e-01a6-4c8b-8286-7633346c4 -t SCN -
s 392375947
```

Example 16-106 Restoring the TDE Wallet while performing database recovery, if TDE wallets are lost

```
# odacli recover-database -in db1 -t latest -tp -tl /u01/NFS_TDE/
testbackups/test2-c/database/91276171/tdbacf2/tdewallet/
ewallet_2020021714442745.p12
Enter TDE wallet password:
```

Example 16-107 Recovering a Database that was associated with a CPU Pool

```
# odacli recover-database -t latest -in irmendb4  -cp testcpupool3
{
  "jobId" : "b09f2d3f-2b01-4dbb-9797-122ba786e0e4",
```

```
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "August 10, 2020 14:56:57 PM UTC",
  "resourceList" : [ ],
  "description" : "Create recovery-latest for db : irmendb4",
  "updatedTime" : "August 10, 2020 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 -fs from_scn -ts to_scn -ft from_timestamp -tt
to_timestamp [-h] [-j]
```

Parameters

Parameter	Description
`--dbname, -in`	Specifies the Database Name.
`--dbid, -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 16-108 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 [-in] [-tl] [-h] [-j]
```

Parameters

Parameter	Description
`--dbname, -in`	Specifies the Database Name.
`--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.

Example 16-109 Restoring TDE wallet for a database from a backup location

```
# odacli restore-tdewallet -in testdb -tl /u01/NFS_TDE/testbackups/
test2-c/database/91276171/tdbacf2/tdewallet/
ewallet_2020021714442745.p12
Enter TDE wallet password:
```

odacli update-backupconfig

Use the command `odacli update-backupconfig` to update a backup configuration.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli update-backupconfig [-i] [-in] [-ca] -d backup destination{Disk|
ObjectStore|None} [-c] [-cr] [-h] [-j] [-no-cr] [-dc] [-ec] [-o][-w]
```

Parameters

Parameter	Description
`--backupconfigid, -i`	Defines the Backup Config ID.
`--backupconfigname, -in`	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.
`--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.
`--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.

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 16-110 Revise a Backup Configuration for Disk

Create a backup configuration named `production` that backs up to disk with a 14 day recovery window.

```
# odacli update-backupconfig -d Disk -in production -w 14
```

odacli update-objectstoreswift

Use the command `odacli update-objectstoreswift` to change the credentials for the ObjectStore account.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

```
# odacli  update-objectstoreswift [-h] [-j] [-in] [-i] [-p] [-u]
Object Store user name
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.
`--objectstoreswiftid`, `-i`	Defines the Object Store Swift identifier (ID).
`--objectstoreswiftName`, `-in`	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.

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 update-backupconfig` with `objectstore` as the destination. The credentials are not validated against endpoint URL and tenancy.

Example 16-111 Changing the Oracle Casper ObjectStore Password

```
# odacli  update-objectstoreswift -i Object Store Swift id -p
```

Example 16-112 Changing the Oracle ObjectStore User Name

```
# odacli  update-objectstoreswift -i Object Store Swift id -u Object Store
user name
```

odacli update-schedule

Use the command `odacli update-schedule` to update the schedule for a database, or to disable the database backup schedule.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
#  odacli  update-schedule  [-x] [-t] [-d] [-e] [-h] [-j] [-id]
```

Parameters

Parameter	Description
--cronExpression, -x	(Optional) Defines the date and time for the update.
--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.
--scheduleid, -id	Defines the schedule with an identifier (ID).

Usage Notes

Backups incur overhead on the system. When possible, do not schedule backups to run when users are trying to access data.

Use a utility, such as www.croncronmaker.com, to generate a valid cron expression.

Example 16-113 Change What Time the Backup Occurs

Edit the cron expression to change the time of scheduled backups for a given schedule ID.

```
#  odacli  update-schedule -i scheduleid  -x "0 0 13 1/1 * ? *" -e
```

Example 16-114 Disable Scheduled Database Backups

```
#  odacli  update-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 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 16-115 Configuring Oracle Data Guard interactively on the commad 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 16-116 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 16-117 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]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--dgconfigid, -i`	Specifies the Oracle Data Guard Configuration ID.

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 16-118 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 [-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.
`--dgid, -i`	Specifies the Oracle Data Guard Status ID.

Usage Notes

Example 16-119 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.
`--dgid, -i`	Specifies the Oracle Data Guard Status ID.

Parameter	Description
`--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 16-120 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 [-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 16-121 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.
`--dgid, -i`	Specifies the Oracle Data Guard Status ID.
`--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 16-122 Performing reinstate operation on Oracle Data Guard

```
# odacli reinstate-dataguard -i 75f23ce0-006a-4aeb-ba6d-b4b60275c19 -u
houdg02
```

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. The default is false.
`--dgid, -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 16-123 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.
- update-cpucore
  Use the `odacli update-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 16-124 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 16-125 Displaying the Current Core Configuration

```
# odacli describe-cpucore

Node  Cores  Modified                       Job Status
----- ------ ------------------------------ ---------------
0      8     July 25, 2016 9:39:59 AM SGT   Configured
```

update-cpucore

Use the `odacli update-cpucore` command to enable the number of CPU cores in the system.

File Path

```
/opt/oracle/dcs/bin/odacli
```

Syntax

```
odacli update-cpucore  -c cores [-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.

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 16-126 Enabling CPU Cores

The following command enables 8 CPU cores.

```
# odacli update-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" : "June 30, 2021 10:29:37 AM UTC",
  "resourceList" : [ ],
  "description" : "CPU cores service update",
  "updatedTime" : "June 30, 2021 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 [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the CPU Pool name.
`--cores, -c`	Specifies the CPU Pool cores.
`--bm, -bm`	Specifies the CPU Pool as Bare Metal.
`--vm, -vm`	Specifies the CPU Pool as Virtual Machine.
`--dbsystem, -dbs`	Specifies the CPU Pool type as DB System.
`--socket, -s`	Specifies the socket to use. If not specified, then use all available sockets.
`--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 guest machines. VM CPU pools can be assigned to multiple VMs.
- Attach dbsystem CPU pool to dbsystems. dbsystem CPU pool can be assigned to multiple dbsystems. After you create a dbsystem CPU pool, you can create a new dbsystem 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 dbsystem.
- 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 mustbe 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.

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 `odacli remap-cpupools` command, after updating the bare metal CPU cores with the command `odacli update-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

- 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 database 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 -n
node_ID -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-id, -n`	(Optional) Specifies the node ID.
`--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.

Parameter	Description
--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] [-s] [-t] [-j] [-f] [-nn] [-p] [-h] [-sh|-no-sh] [-cp]
```

Parameters

Parameter	Description
`--databaseUniqueName, -u`	Defines a unique name for the database.
`--dbname, -n`	Defines the name given to the new database (`dbname`.)
`--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.
`--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.
`--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 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.

Example 16-127 Cloning a Database

The following example creates a clone database snap1 from source database acfsdb1.

```
# /opt/oracle/dcs/bin/odacli clone-database -n snap1 -u snap1u -f
acfsdb1
```

Example 16-128 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 new 'SYS user' password:
Confirm 'SYS user' password:
{
  "jobId" : "74a80586-3523-4367-93a0-38401647c5a0",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "August 09, 2020 10:25:40 AM UTC",
  "resourceList" : [ ],
  "description" : "Snap Database service creation with db name:
dbclone",
  "updatedTime" : "August 09, 2020 10:25:40 AM UTC"
}
```

Example 16-129 Cloning a Database and associating it with a different CPU Pool

```
# odacli clone-database -f dbcpupin -n db1clone -u db1cloneUnique -cp
testdbcpupool2
Enter new 'SYS user' password:
Confirm 'SYS user' password:
{
  "jobId" : "2fe285b5-2e3b-4e80-9121-e7c8380952b3",
  "status" : "Created",
  "message" : null,
  "reports" : [ ],
  "createTimestamp" : "August 09, 2020 10:36:55 AM UTC",
  "resourceList" : [ ],
  "description" : "Snap Database service creation with db name:
db1clone",
```

```
  "updatedTime" : "August 09, 2020 10:36:55 AM UTC"
}
```

Example 16-130 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:
```

Related Topics

- Cloning an Oracle ACFS Database Using Command Line Interface
  Create a database from an existing Oracle ACFS database using CLI commands.

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] [-h] [-rd] [-sh|-no-sh] [-t] [-
cp]
```

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.
`--cpupool, -cp`	Defines the CPU pool name. The CPU pool must be of type bare metal (BM).
`--dbEdition, -de`	Defines the Database Edition.

Parameter	Description
--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.
--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.

Parameter	Description
--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.
--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.
--enable-seha, -sh	Specifies whether you want to enable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases.
--targethost, -th	(Optional) Specifies the target host name (applicable only for Oracle RAC One Node Database and single-instance Oracle Database, where supported).
--targetnode, -g	(Optional) Specifies the target node number (applicable only for Oracle RAC One Node Database and single-instance Oracle Database, where supported). The values are 0 or 1.
--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.

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. 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 12.1 or 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 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.

> **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, 19.11.0.0, 18.8.0.0, 12.2.0.1, or12.1.0.2, 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.

> **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 16-131 Creating a TDE-enabled database

```
# 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 16-132 Creating a Database in Interactive Mode

This example creates a 19.14.0.0.0 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" : "December 8, 2021 03:54:03 AM EDT",
   "description" : "Database service creation with db name: hrdb",
   "updatedTime" : "December 8, 2021 03:54:03 AM EDT"
}
```

Example 16-133 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 19.14.0.0.0
# odacli create-database -n hrmsdb2 -v 19.14.0.0.220118
```

The following statement creates a new database against a home with a specific Oracle Database Bundle:

```
# odacli create-database -n hrmsdb3 -v 19.14.0.0.220118
```

Example 16-134 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
```

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  [-ea] [-h] [-j] [-fd]
```

Parameters

Parameter	Description
`--dbid, -i`	Identifies the database home identifier (ID) to display. Use the `odacli list-databases` command to obtain the `--dbid`.
`--force, -fd`	(Optional) Forces the delete operation.
`--eraseAll, -ea`	(Optional) Removes all files.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.

Usage

> **Note:**
>
> 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.

Example 16-135 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  12.1.0.2      true
OLTP
7e28bf52-1a09-49fd-9391-841838d2c42f   crmdb   12.1.0.2      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

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To display database details:

```
odacli describe-database -i dbid [-h] [-j]
```

Parameters

Parameter	Description
--dbid, -i	Identifies the database home identifier (ID) to display. Use the `odacli list-databases` command to obtain the `dbid`.
--json, -j	(Optional) Displays JSON output.
--help, -h	(Optional) Displays help for using the command.

Example 16-136

Display information for a database.

```
# odacli describe-database -i 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:
           SEHA 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 16-137 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: 19.14.0.0.0
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: December 8, 2021, 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.

Parameter	Description
-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 16-138 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     19.14.0.0.0    true
fb4d02f3-2413-47ca-8584-a768e23ec2e7   ee12db      19.14.0.0.0    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] [-lb] [-no-bkp] [-th] [-g]
[-cl] [-rd] [-s] [-an] [-ctm] [-bi] [-bin] [noab] [-bp] [-id] [-in] [-y] [-
cp] [-no-cpupool] [-rkt] [-ctp] [-h] [-j]
```

Parameters

Parameter	Description
--databaseid, -id	Defines the database identifier.
--dbName, -in	Defines the database name to be updated. Specify either --dbName or --dbid.
--dbid, -i	Defines the database ID to be updated. Specify either --dbName or --dbid.

Parameter	Description
--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-autobackup, -noab	(Optional) Disables automatic database backups.
--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.
--disable-seha, -no-sh	Disables Standard Edition High Availability for single-instance Oracle Database 19c and later.
--enable-seha, -sh	Enables Standard Edition High Availability for single-instance Oracle Database 19c and later.
--targethost, -th	(Optional) Specifies the target host name (applicable only for Oracle RAC One Node Database and single-instance Oracle Database, where supported).
--targetnode, -g	(Optional) Specifies the target node number (applicable only for Oracle RAC One Node Database and single-instance Oracle Database, where supported). The values are 0 or 1.

Parameter	Description
`--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.
`--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 16-139 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" : "August 26, 2019 06:48:58 AM UTC",
  "resourceList" : [ ],
  "description" : "modify-database service with db ids: 1941d594-
c777-4eca-9fce-18b778d5c153",
  "updatedTime" : "August 26, 2019 06:48:58 AM UTC"
}
```

Example 16-140 Converting Oracle RAC or Oracle RAC One Node Database to Single-instance Oracle Database

```
# odacli modify-database -in 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" : "April 21, 2020 00:32:55 AM UTC",
  "resourceList" : [ {
    "resourceId" : "de2a7959-e31b-4dee-b600-6b16dfb78c34",
    "resourceType" : "DB",
    "resourceNewType" : null,
    "jobId" : "4b548365-dac5-4557-a6a8-2f65c2b725e7",
    "updatedTime" : "April 21, 2020 00:32:55 AM UTC"
  } ],
  "description" : "Modify database : db18",
  "updatedTime" : "April 21, 2020 00:32:55 AM UTC"
}
```

Example 16-141 Specifying the target node when converting database

```
# odacli modify-database -in db18 -y SI -g node_number
# odacli modify-database -in db18 -y SI -th node_name
```

Example 16-142 Changing the TDE Wallet Password

```
# odacli modify-database -in testdb -ctp
Enter current 'TDE wallet' password:
Enter new 'TDE wallet' password:
Confirm 'TDE wallet' password:
```

Example 16-143 Re-keying the TDE Master Encryption Password

```
# odacli modify-database -in testdb -rkt
Enter current 'TDE wallet' password:
```

Example 16-144 Removing a CPU pool

```
# odacli modify-database -in 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" : "August 09, 2020 16:18:54 PM UTC",
  "resourceList" : [ {
    "resourceId" : "ff9c138b-a003-40cf-b1fc-0b33cd955ebf",
    "resourceType" : "DB",
    "resourceNewType" : null,
    "jobId" : "c1d05fb3-828a-4644-af9a-9d82b4ef5a89",
    "updatedTime" : "August 09, 2020 16:18:54 PM UTC"
  } ],
  "description" : "Modify database : dbcpupin",
  "updatedTime" : "August 09, 2020 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] [-in dbname] [-dh dest-dbhome-id][-dhn dest-dbhome-name][-j] [-h]
```

Parameters

Parameter	Description
`--dbid, -i`	(Optional) Defines the database ID.
`--dbName, -in`	(Optional) Defines the database name.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output. The default is false.
`--dest-db-home-id, -dh`	(Optional) Defines the destination database home ID.
`--dest-dbhome-name, -dhn`	(Optional) Defines the destination database home name.

Usage Notes

Example 16-145 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] [-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.
`--no-dbconsole`, `-no-co`	(Optional) Disables Database Console. Use the `-co`flag to enable Database Console.

Parameter	Description
`--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.
`--enable-seha, -sh`	Specifies whether you want to enable Standard Edition High-Availability for Oracle Database Standard Edition 19c and later single-instance databases.
`--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 only on the primary database.
- 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 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.

Example 16-146 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" : "December 8, 2021 05:55:49 AM EDT",
  "description" : "Database service registration with db service name:
crmdb.example.com",
  "updatedTime" : "December 8, 2021 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: December 8, 2021 5:55:49 AM EDT
                Message:

Task Name                      Start Time
------------------------------ ------------------------------
restore control file           December 8, 2021 5:55:49 AM EDT
move spfile to right location  December 8, 2021 5:56:08 AM EDT
register DB with clusterware   December 8, 2021 5:56:13 AM EDT
reset db parameters            December 8, 2021 5:57:05 AM EDT
Running DataPatch              December 8, 2021 5:57:36 AM EDT

(Continued)
  End Time                     Status
------------------------------ --------
December 8, 2021 5:56:08 AM EDT   Success
December 8, 2021 5:56:13 AM EDT   Success
December 8, 2021 5:57:05 AM EDT   Success
```

```
December 8, 2021 5:57:36 AM EDT   Success
December 8, 2021 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 -from
source dbhome id -to destination dbhome id [-ctm] [-j] [-h] [-sh|-no-sh]
```

Parameters

Parameter	Description
`--databaseids, -i`	(Optional) Defines the Database IDs to upgrade. You can use a comma separated list of database IDs.
`--destDbHomeId, -to`	The DB HOME ID of the destination database home.
`--sourceDbHomeId, -from`	The DB HOME ID 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.
`--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.
`--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 16-147 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
```

Related Topics

- 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.

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.

Parameter	Description
`--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 16-148 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 19.14, 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 16-149 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 16-150 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
`--ohomeid, -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 16-151 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 16-152 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, 19.14.0.0.220118. If you use the database version without specifying the bundle patch number, then the latest bundle patch is used.

Example 16-153 Creating an Oracle Database Home

The following example creates an Oracle Database Home version 19.14.0.0.220118.

```
# odacli  create-dbhome -v 19.14.0.0.220118
```

odacli delete-dbhome

Use the `odacli delete-dbhome` command to delete database home that is not associated with a database.

File Path

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli delete-dbhome  -i [-h]
```

Parameters

Parameter	Description
`--id, -i`	Identifies the database home using a database identifier (ID).
`--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 16-154 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

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To display details about Oracle Database Home:

```
odacli describe-dbhome -i dbhomeid [-h] [-j] [-v]
```

Parameters

Parameter	Description
-i *dbhomeid*	Identifies the database home ID. Use the `odacli list-dbhomes` command to get the *dbhomeid*.
--help, -h	(Optional) Displays help for using the command.
--json, -j	(Optional) Displays JSON output. The default is false.
-v*dbversion*	(Optional) Identifies the Database Home Version. Use the `odacli list-dbhomes –v` command to get the *dbversion*.

Example 16-155 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: OraDB12102_home1
                      Version: 12.1.0.2
                Home Location: /u01/app/orauser/product/12.1.0.2/dbhome_1
                      Created: Jun 2, 2016 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 16-156 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 update-dgstorage
  Use the `odacli update-dgstorage` command to release disks.

odacli create-dbstorage

Use the `odacli create-dbstorage` command to create the file system for database migrations.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli create-dbstorage  -n dbname -s dataSize -u databaseUniqueName -r [ASM|
ACFS] [-h]
```

Parameters

Parameter	Description
`--dbname`, `-n`	Defines the name of the database.
`--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*

Parameter	Description
`--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
`--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 16-157 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

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli delete-dbstorage  -i [-h]
```

Parameters

Parameter	Description
`--id, -i`	Identifies the database storage using a database identifier (ID).
`--help, -h`	(Optional) Displays help for using the command.

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 16-158 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

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

```
odacli describe-dbstorage  -i [-j] [-h]
```

Parameters

Parameter	Description
`--id, -i`	Identifies the database storage.
`--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 16-159 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 16-160 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 16-161 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 16-162 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.

Parameter	Description
`--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 16-163 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 resized to match the new size specified.
- 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 16-164 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 update-dgstorage

Use the `odacli update-dgstorage` command to release disks.

File Path

`/opt/oracle/dcs/bin/odacli`

Syntax

To release disks:

```
odacli update-dgstorage [-h] [-f] [-rd] [-j]
```

Parameters

Parameter	Description
`--json, -j`	(Optional) Displays JSON output.
`--help, -h`	(Optional) Displays help for using the command.
`--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.

Usage Notes

- Run the `odacli list-dgdisks` to view details of the disk, before running the `odacli update-dgstorage` command.
- The `odacli update-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 16-165 Releasing a physical disk HDD_E0_S10_577135552 which has two logical partitions on two different disk groups

```
# odacli update-dgstorage -rd HDD_E0_S10_577135552
Job details
----------------------------------------------------------------
                     ID:  46bc736c-417b-4ac2-826e-aa857db50e29
            Description:  Release disks: [HDD_E0_S10_577135552]
                 Status:  Running
                Created:  July 7, 2020 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:  July 7, 2020 6:24:27 PM PDT
                Message:  Release ASM disks
```

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.
- 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-job
  Use the `odacli list-purge-jobdata-job` 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 16-166 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 16-167 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 16-168 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 16-169 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 16-170 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 16-171 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 16-172 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

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To view a specific job, status, and tasks:

```
odacli describe-job   -i jobid [-j] [-h]
```

Parameters

Parameter	Description
`--jobid`, `-i` *jobid*	Identifies the job. To get the job identifier (jobid), run the `list-jobs` command.
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-173 Displaying Details for a Job

To display details of a specific job with jobid `02df22c8-c21f-4162-8265-97f7826c243a`:

```
# odacli describe-job -i 02df22c8-c21f-4162-8265-97f7826c243a
```

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.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

To list jobs and view job details and status:

```
odacli list-jobs [-j] [-h] [-o] [-f] [-t] [-hd] [-tl] [-k] [-s]
```

Parameters

Parameter	Description
`--json`, `-j`	(Optional) Displays JSON output.
`--help`, `-h`	(Optional) Displays help for using the command.
`--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`

Parameter	Description
--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
--keyword, -k	(Optional) Specifies the keyword that the job description contains. Example: # odacli list-jobs -k OraDB
--status, -s	(Optional) Specifies the status of the job, such as Created, Scheduled, Running, Failure, Success, InternalError, Waiting. Example: # odacli list-jobs -s Success

Example 16-174 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 16-175 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 16-176 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-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 16-177 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-job

Use the `odacli list-purge-jobdata-job` 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-job [-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 16-178 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] [-l] [-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 16-179 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
```

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 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 16-180 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

$ORACLE_HOME`/opt/oracle/dcs/bin/odacli`

Syntax

To list jobs and view job details and status:

```
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`.
`--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.

Parameter	Description
`--usageOverPercentage, -uPer`	Starts purge when log usage is over the specified number of percentage of total partition space. Valid range is 10-50.

Example 16-181 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 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 16-182 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 [-h]
```

Parameters

Parameter	Description
`--help, -h`	(Optional) Displays help for using the command.
`--jobid, -i`	Displays the summary of the log cleanup with the specified job ID.

Example 16-183 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 16-184 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 16-185 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 16-186 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
- odacli delete-asr
- odacli describe-asr
- odacli test-asr
- odacli update-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 username [-r proxyserver] [-t proxyport]
[-y proxyuser] [-s snmpversion] -i[asrip] -e [internal|external] [-j]
[-h]
```

Parameters

Parameter	Description
`--asrip`, `-i`	(Optional) Identifies the external Oracle ASR Manager IP address.
`--asrtype`, `-e` {internal | external}	Defines the Oracle ASR Configuration Type. The default is internal.
`--help`, `-h`	(Optional) Displays help for using the command.
`--json`, `-j`	(Optional) Displays JSON output.
`--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.
`--snmpversion`, `-s [V3]`	(Optional) Defines Simple Network Management Protocol (SNMP) Version 3.
`--username`, `-u`	Defines the Oracle ASR user name. The user name is the My Oracle Support user name under which the server is registered.

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 -i 198.51.100.1`

Example 16-187 Configuring Oracle ASR with a Proxy Server

This example configures Oracle ASR for user name scott.tiger@example.com. A proxy server, www-proxy.example.com, and port 80 are defined.

```
# odacli configure-asr -u john.smithl@example.com -t 80 -r www-
proxy.example.com
Enter new 'ASR user' password:
Confirm 'ASR user' password:

Job details
----------------------------------------------------------------

ID: d99559b6-d98d-4cb7-b44d-8577cab26667
Description: Configure ASR
Status: Created
```

```
Created: December 8, 2021 6:12:19 AM WSST
Message:

Task Name          Start Time          End Time        Status
 ---------------- ----------------- ----------------- ----------
```

Example 16-188 Configuring an External Oracle ASR

This example configures Oracle Database Appliance to use an external Oracle ASR instance at IP address 10.20.30.40.

```
# odacli configure-asr --asrip 10.20.30.40 --asrtype External
{
  "jobId" : "ea054a2f-d18d-4253-83bc-b57434e3598e",
  "status" : "Created",
  "message" : "Please run the script '/tmp/activateExternalAssets.pl' on the
ASRManager host once the current job is successful.",
  "reports" : [ ],
  "createTimestamp" : "December 8, 2021 22:12:34 PM EST",
  "description" : "Configure ASR",
  "updatedTime" : "December 8, 2021 22:12:34 PM EST"
}
```

When the job completes successfully, run the `/tmp/activateExternalAssets.pl` script on the Oracle ASR Manager host.

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 16-189 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 describe-asr

Use the `odacli describe-asr` command to display Oracle Auto Service Request (Oracle ASR) configuration details.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
describe-asr [-h]
```

Parameters

Parameter	Description
--help, -h	(Optional) Displays help for using the command.

Example 16-190 Displaying Oracle ASR Details

```
# odacli describe-asr

ASR details
----------------------------------------------------------------
ID: e841d74a-687b-4e87-9548-1baa2090d48e
Name: UserName: scott.tiger@example.com
ProvyServerName: www-proxy.example.com
ProxyPort: 80
ProxyUserName:
SnmpVersion: V3
State: N/A
Created: July 15, 2016 8:53:54 AM PST
Updated: July 15, 2016 8:53:54 AM PST
```

odacli test-asr

Use the `odacli test-asr` command to test the Oracle Auto Service Request (Oracle ASR) configuration.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
# odacli test-asr [-h]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Usage Notes

This command internally invokes the SNMP test trap by invoking `/SP/alertmgmt/rules/1 testrule=true`.

Example 16-191 Testing the Oracle ASR Configuration

```
# odacli test-asr

Job details
----------------------------------------------------------------
ID: ec6783f4-551d-4686-ab1b-22e2d9e59c98
Description: Test ASR
Status: Created
Created: July 25, 2016 9:03:15 AM SGT
Message:

Task Name                      Start Time            End Time          Status
------------------------------ --------------------- ----------------
----------
```

odacli update-asr

Use the `odacli update-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

```
  update-asr  -u username [-r proxyserver] [-t proxyport] [-y proxyuser] [-s
snmpversion] -i[asrip] -e [internal|external] [-j] [-h]
```

Parameters

Parameter	Description
`--asrip`, `-i`	(Optional) Identifies the external Oracle ASR Manager IP address.

Parameter	Description
`--asrtype, -e` [internal | external]	Defines the Oracle ASR Configuration Type. The default is internal.
`--help, -h`	(Optional) Displays help for using the command.
`--json, -j`	(Optional) Displays JSON output.
`--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.
`--snmpversion, -s [v3]`	(Optional) Defines Simple Network Management Protocol (SNMP) Version 3.
`--username, -u`	Defines the Oracle ASR user name. The user name is the My Oracle Support user name under which the server is registered.

Usage Notes

- Enter the password during Oracle ASR configuration.
- All log files for Oracle ASR are located in the `/var/opt/asrmanager/log/` directory.
- You cannot use the `update-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.
- To configure an external Oracle ASR Manager, you must define the Oracle ASR Configuration Type as external (`-e external`). For example, `odacli update-asr -e external -i 198.51.100.1`

Example 16-192 Updating Oracle ASR with 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 update-asr -u scott.tiger@example.com --proxyserver www-
proxy2.example.com --proxyport 80
Enter new 'ASR user' password:
Confirm 'ASR user' password:


Job details
----------------------------------------------------------
ID: 79cb2baa-1644-45c5-a004-a303e3111807
Description: Update ASR
Status: Created
Updated: December 8, 2021 9:53:54 AM PST
Message:
```

```
Task Name          Start Time          End Time        Status
---------------- ---------------- ----------------- ----------
```

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 update-osconfigurations
  Use the command `odacli update-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 16-193 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 update-osconfigurations

Use the command `odacli update-osconfigurations` to update the HugePage and memlock values.

File Path

$ORACLE_HOME/opt/oracle/dcs/bin/odacli

Syntax

```
odacli update-osconfigurations [-h] [-hs] [-j] [-m]
```

Parameters

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.
`--hugespace`, `-hs`	(Optional) Updates the HugePage value.
`--json`, `-j`	(Optional) Displays JSON output. The default is false.
`--memlock`, `-m`	(Optional) Updates the memlock value.

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 update memlock or hugepage. If no option is provided, then both parameters are updated to the suggested values.

Example 16-194 Updating the HugePage and Memlock Parameters to the Suggested Values

```
odacli update-osconfigurations
{
"jobId" : "954cf7a5-9cad-451c-8820-3140a716af26",
"status" : "Created",
"message" : "Successfully submitted a request to configure OS
parameters.",
"reports" : [ ],
"createTimestamp" : "February 06, 2018 00:03:51 AM MST",
"resourceList" : [ ],
"description" : "Configuring OS Parameter",
"updatedTime" : "February 06, 2018 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: February 6, 2018 12:03:51 AM MST
Message:
Task Name             Start Time                          End Time                            Status
--------------------  ----------------------------------  ----------------------------------
-------
Setting up memlock.   February 6, 2018 12:03:51 AM MST   February 6, 2018 12:03:51 AM MST
Success
Setting up HugeSpace  February 6, 2018 12:03:51 AM MST   February 6, 2018 12:03:51 AM MST
Success
```

odaadmcli Hardware Monitoring Commands

Use the `hardware monitoring` commands to display hardware configurations.

- 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 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 16-195 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 16-196 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 16-197 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 16-198 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 16-199 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 16-200 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 16-201 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.

Parameter	Description
--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 16-202 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 16-203 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.

Parameter	Description
`--help`, `-h`	(Optional) Displays help for using the command.

Example 16-204 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 16-205 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 16-206 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 16-207 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 16-208 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 16-209 Listing All Diskgroups

```
# odaadmcli show diskgroup

DiskGroups
----------
DATA
RECO
```

Example 16-210 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 16-211 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 16-212 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 16-213 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 Database System Commands for KVM
  Use the odacli database 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-insensitive.

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 update-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 update-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 update-objectstoreswift`
- `odacli list-osconfigurations`
- `odacli update-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 update-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 Database System Commands for KVM

Use the odacli database 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 database 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 database 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 -p json_payload_path [-j] [-h]
```

Parameters

Parameter	Description
`--params, -p`	(Mandatory) Input parameters file path for DB system creation.
`--json, -j`	(Optional) Displays output in JSON format.
`--help, -h`	(Optional) Displays help for using the command.

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:  January 5, 2021 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 two-node 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 database system on Oracle Database Appliance Kernel-based Virtual Machine (KVM).

Syntax

```
odacli modify-dbsystem -n name [-cp cpu_pool_name] [-no-cp] [-avn v_network] [-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]
[-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Describes the name of the DB system.
`--attach-vnetwork, -avn`	(Optional) Specifies the name of the vnetwork to attach.
`--cpupool, -cp`	(Optional) Specifies the shared DB System CPU Pool name.
`--detach-vnetwork, -dvn`	(Optional) Specifies the name of the vnetworks to detach.
`--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.
`--scan-ips, -sip`	(Optional) Specifies the SCAN IP of the network.

Parameter	Description
`--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.
`--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.
`--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 dbsystem 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 database shape, that is, scale up (increase) or scale down (decrease):

    ```
    odacli modify-dbsystem -n dbsystem name -s new_shape
    ```

  - Enable NUMA for the existing dbsystem. NUMA is enabled on Oracle Database Appliance release 19.12. For existing dbsytems created before release 19.12, use the `odacli modify-dbsystem --enable_numa` command to enable the NUMA for the dbsystem. This command includes enabling the NUMA for the VM configuration, operation system, and the database inside the dbsystem.

    ```
    odacli modify-dbsystem --name dbsystem name --enable_numa
    ```

  - Attach or detach new network to the dbsystem. 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 dbsystem. 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 dbsystem:

```
odacli modify-dbsystem -n dbsystem1 -dvn vnet2
```

Example

Modify a KVM database system:

```
# odacli modify-dbsystem -n example_system1 -cp example_pool -s new_shape
```

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 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 19.14, 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 [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	(Mandatory) Describes the name of the DB system.
`--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 on a single-node system.

```
# odacli describe-dbsystem -n example_system1

DB System
details
--------------------------------------------------------------------------------
---
                         ID:  0656fe7b-111a-4bd3-b1f0-b3832812d328
                       Name:  scaoda8m018c4
                      Shape:  odb2
                     Memory:  16.00 GB
                     Status:  CONFIGURED
                    Created:  2021-01-19 00:07:20 UTC
                    Updated:  2021-01-19 00:39:16 UTC

 CPU Pool
--------------------------
                       Name:  cpu_pool1
            Number of cores:  2
                       Node:  node1
          Effective CPU set:  14-15,46-47
                Online CPUs:  14, 15, 46, 47
               Offline CPUs:  NONE


 VM Storage
--------------------------
                 Disk group:  DATA
                Volume name:  SC0E8DB73A
              Volume device:  /dev/asm/sc0e8db73a-250
                       Size:  200.00 GB
                Mount Point:  /u05/app/sharedrepo/example_system
```

```
 VMs
--------------------------
                     Host:  host_name
                  VM Name:  vm_name
             VM Host Name:  vm_host_name
             Target State:  ONLINE
            Current State:  ONLINE


 VNetworks
--------------------------
                     Host:  vm_host_name
                  VM Name:  vm_name
                   Public:  xxx.xxx.xxx.xxx   / xxx.xxx.xxx.xxx  /
ens3 / BRIDGE(pubnet)
                      ASM:  xxx.xxx.xxx.xxx  / xxx.xxx.xxxx.xxx /
ens4 / BRIDGE(privasm) VLAN(priv0.100)

Databases
--------------------------
                     Name:  c1kdb
              Resource ID:  a98ea0d7-ec75-4dd7-a6b3-f72e0b644e3f
              Unique name:  c1kdb
              Database ID:  1053377603
              Domain name:  domain_name
               DB Home ID:  5ea6c0a1-d2ae-4b7f-96e9-109a569cd09b
                    Shape:  odb1
                  Version:  19.10.0.0.210119
                  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:  true
                 PDB name:  pdb1
           PDB admin user:  pdbadmin
```

View details of a KVM database system on a High Availability system, without InfiniBand.

```
# odacli describe-dbsystem -n example_system2

DB System details
--------------------------------------------------------------------------------
---
                        ID:  04f718f8-ddc9-4b45-8e8e-07d14165eb9c
                      Name:  example_system2
                     Shape:  odb2
                    Memory:  16.00 GB
                    Status:  CONFIGURED
                   Created:  2021-01-19 10:00:08 UTC
                   Updated:  2021-01-19 10:04:02 UTC

CPU Pool
--------------------------
                      Name:  c4bfa26faa
           Number of cores:  2
                      Node:  n1
         Effective CPU set:  0-1,20-21
               Online CPUs:  0, 1, 20, 21
              Offline CPUs:  NONE


                      Node:  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
              Target State:  ONLINE
             Current State:  ONLINE


                      Host:  host2
                   VM Name:  y4bfa26faa
              VM Host Name:  example_domain2
              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
DDB System details
```

```
--------------------------------------------------------------------------------
---
                        ID:  0c5280a1-3ee1-438e-ad11-1cee23de17fd
                      Name:  example_system3
                     Shape:  odb2
                    Status:  CONFIGURED
                    Memory:  16.00 GB
                   Created:  2020-12-10 02:11:11 UTC
                   Updated:  2020-12-10 02:11:11 UTC

 CPU Pool
--------------------------
                      Name:  ce2ab55255
           Number of cores:  2
                      Node:  node1
         Effective CPU set:  4-5,24-25
               Online CPUs:  4, 5, 24, 25
              Offline CPUs:  NONE


                      Node:  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
              Target State:  ONLINE
             Current State:  ONLINE


                      Host:  host2
                   VM Name:  ye2ab55255
              VM Host Name:  host_name2
              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:  19.10.0.0.210119
                  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:
```

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 database 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 operation deletes the database system and Oracle Clusterware files, and retains the database files. With the `-f` option, the operation deletes the database system, Oracle Clusterware files, and the database files.

Example

Delete a KVM database system.

```
# odacli delete-dbsystem -n example_system
WARNING: Use --force option to delete the database and datafiles.
Do you confirm delete DB System 'example_system'? (Options: y/n,
default: n):

Job details
----------------------------------------------------------------
                     ID:  4d70e75d-1bc0-4917-a50a-63cd6a108d1e
            Description:  DB System example_system deletion
                 Status:  Success
                Created:  January 16, 2021 9:19:54 AM UTC
                Message:

Task Name                                Start
Time                          End Time
Status
----------------------------------------
----------------------------------
---------------------------------- ----------

# odacli describe-job -i 4d70e75d-1bc0-4917-a50a-63cd6a108d1e

Job details
----------------------------------------------------------------
                     ID:  4d70e75d-1bc0-4917-a50a-63cd6a108d1e
            Description:  DB System example_system deletion
                 Status:  Success
                Created:  January 16, 2021 9:19:54 AM UTC
                Message:

Task Name                                Start
Time                          End Time
Status
----------------------------------------
----------------------------------
---------------------------------- ----------
Remove DB System from Clusterware        January 16, 2021 9:19:54 AM
UTC    January 16, 2021 9:19:54 AM UTC    Success
Deprovision DB System VM(s)              January 16, 2021 9:19:54 AM
UTC    January 16, 2021 9:19:54 AM UTC    Success
Delete DB System ACFS filesystem         January 16, 2021 9:19:54 AM
UTC    January 16, 2021 9:20:04 AM UTC    Success
Delete DB System ASM volume              January 16, 2021 9:20:04 AM
```

```
UTC    January 16, 2021 9:20:11 AM UTC    Success
Delete DB System VNetworks               January 16, 2021 9:20:04 AM UTC
January 16, 2021 9:20:11 AM UTC    Success
Delete DB System metadata               January 16, 2021 9:20:11 AM UTC
January 16, 2021 9:20:11 AM 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 [-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).
`--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 [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the virtual disk name
`--increment, -i`	Incremental storage size to be modified. Format is K|M|G|T
`--json, -j`	(Optional) Displays output in JSON format.
`--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
`--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 [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the CPU Pool name.
`--cores, -c`	Specifies the CPU Pool cores.
`--bm, -bm`	Specifies the CPU Pool as Bare Metal.
`--vm, -vm`	Specifies the CPU Pool as Virtual Machine.
`--dbsystem, -dbs`	Specifies the CPU Pool type as DB System.
`--socket, -s`	Specifies the socket to use. If not specified, then use all available sockets.
`--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 guest machines. VM CPU pools can be assigned to multiple VMs.
- Attach dbsystem CPU pool to dbsystems. dbsystem CPU pool can be assigned to multiple dbsystems. After you create a dbsystem CPU pool, you can create a new dbsystem 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 dbsystem.

- 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 mustbe 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.

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 `odacli remap-cpupools` command, after updating the bare metal CPU cores with the command `odacli update-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

- 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 database 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 -n
node_ID -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-id, -n`	(Optional) Specifies the node ID.
`--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.
`--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

Parameter	Description
`--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 -src source -vc vcpus -vms vmstorage -cp cpupool -
d description -g graphics
-mm max-mem-size -mvc max-vcpu-count -ost os-type -osv os_variant -pn preferred_node -
s disk_size -vd vdisks
-vn virtual_networks -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, FTP or NFS.
`--vcpus`, `-vc`	Specifies the number of vCPUs to use.
`--vmstorage`, `-vms`	Specifies the VM storage name
`--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

Parameter	Description
`--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-type, -ost`	(Optional) Specifies the operating system type
`--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_option -cfg -avd attach_vdisks -avn
attach_vnetworks -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  [-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-option, -bo`	(Optional) Specifies the the boot option 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.
`--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
`--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
```

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 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 database 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
`--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 -to node [-j] [-h]
```

Parameters

Parameter	Description
`--name, -n`	Specifies the virtual machine name to be migrated
`--target-node, -to`	Specifies the destination node
`--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 `n1`.

```
odacli migrate-vm -n vm1 -to n1
```

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

Troubleshooting Oracle Database Appliance

Understand tools you can use to validate changes and troubleshoot Oracle Database Appliance problems.

- 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.
- Configuring a Trusted SSL Certificate for Oracle Database Appliance
  The Browser User Interface and DCS Controller use SSL-based HTTPS protocol for secure communication. Understand the implications of this added security and the options to configure SSL certificates.
- 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.

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
```

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 Zookeeper, 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/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 Zookeeper, 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 19.14.0.0.0
{
  "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" : "February 22, 2021 02:37:37 AM CST",
  "description" : "Update-dcscomponents job completed and is not part of
Agent
job list",
  "updatedTime" : "February 22, 2021 02:39:10 AM CST"
}
```

The pre-check report log file at the location `/opt/oracle/dcs/log/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
```

When the `odacli update-dcscomponents` command fails:

On Oracle Database Appliance single-node systems, the command output is as follows:

```
# ./odacli update-dcscomponents -v 19.14.0.0.0

DCS-10008:Failed to update DCScomponents: 19.10.0.0.0
Internal error while patching the DCS components :
DCS-10231:Cannot proceed. Pre-checks for update-dcscomponents failed.
Refer to
/opt/oracle/dcs/log/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 19.14.0.0.0

Internal error while patching the DCS components :
DCS-10231:Cannot proceed. Pre-checks for update-dcscomponents failed.
Refer to
/opt/oracle/dcs/log/jobId-dcscomponentsPreCheckReport.log
on node 0 and /opt/oracle/dcs/log/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/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
```

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
```

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
```

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
```

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
```

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
```

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.

Viewing the Bill of Materials in the Browser User Interface

Use the Appliance tab in the Browser User Interface to 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 **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 18-1 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 18-2 Example Command to View the Bill of Materials from the Command Line for Virtualized Platforms

```
# oakcli describe-system -b
```

Example 18-3 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.
- 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 19.14.
- 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.
- 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 19.14.

When you provision or patch your appliance to Oracle Database Appliance release 19.14, 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 19.14, 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 19.14 Baremetal Systems for New Installation

When you provision or upgrade to Oracle Database Appliance 19.14, 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 19.14 Virtualized Platform

When you provision or upgrade to Oracle Database Appliance 19.14, 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 19.14, 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 18-1 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.

Table 18-1 (Cont.) Command Options for tfactl Tool

Option	Description
-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.
-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.

Table 18-1 (Cont.) Command Options for tfactl Tool

Option	Description
`-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.
`-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 18-4 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 18-5 Block dumps after redaction

```
14A533F40 ******** ******** ******** ******** [****************]
14A533F50 ******** ******** ******** ******** [****************]
14A533F60 ******** ******** ******** ******** [****************]
```

Example 18-6 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 18-7 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.

Example 18-8 Describing current status of ACR

```
bash-4.2# odacli describe-acr
Trace File Redaction: Enabled
```

Example 18-9 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 18-10 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 18-11 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
}
```

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 18-12 Masking or Sanitizing Sensitive Data in a Specific Collection

```
tfactl diagcollect -srdc ORA-00600 -mask
tfactl diagcollect -srdc ORA-00600 -sanitize
```

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 18-13 Sanitizing Sensitive Information in Specific Collection IDs

```
orachk -sanitize comma_delimited_list_of_collection_IDs
```

Example 18-14 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 18-15 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 18-16 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 18-17 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.

Configuring a Trusted SSL Certificate for Oracle Database Appliance

The Browser User Interface and DCS Controller use SSL-based HTTPS protocol for secure communication. Understand the implications of this added security and the options to configure SSL certificates.

The Browser User Interface provides an added layer of security using certificates and encryption, when an administrator interacts with the appliance. Encryption of data ensures that:

- Data is sent to the intended recipient, and not to any malicious third-party.
- When data is exchanged between the server and the browser, data interception cannot occur nor can the data be edited.

When you connect to the Browser User Interface through HTTPS, the DCS Controller presents your browser with a certificate to verify the identity of appliance. If the web browser finds that the certificate is not from a trusted Certificate Authority (CA), then the browser assumes it has encountered an untrusted source, and generates a security alert message. The security alert dialog boxes display because Browser User

Interface security is enabled through HTTPS and SSL, but you have not secured your Web tier properly with a trusted matching certificate from a Certificate Authority. It is possible to purchase commercial certificates from a Certificate Authority or create your own and register them with a Certificate Authority. However, the server and browser certificates must use the same public certificate key and trusted certificate to avoid the error message produced by the browser.

There are three options to configure your certificates:

- Create your own key and Java keystore, ensure it is signed by a Certificate Authority (CA) and then import it for use.
- Package an existing Privacy Enhanced Mail (PEM) format key and certificates in a new Java keystore.
- Convert an existing PKCS or PFX keystore to a Java keystore and configure it for the Browser User Interface.

> **Note:**
>
> For Oracle Database Appliance High-Availability hardware models, run the configuration steps on **both** nodes.

The following topics explain how to configure these options:

- Option 1: Creating a Key and Java Keystore and Importing a Trusted Certificate
  Use `keytool`, a key and certificate management utility, to create a keystore and a signing request.
- Option 2: Packaging an Existing PEM-format Key and Certificates in a New Java Keystore
  Use the OpenSSL tool to package Privacy Enhanced Mail (PEM) files in a PKCS keystore.
- Option 3: Converting an Existing PKCS or PFX Keystore to a Java Keystore
  If you have an existing PKCS or PFX keystore for your server's domain, convert it to a Java keystore.
- Configuring the DCS Server to Use Custom Keystore
  After packaging or converting your keystore into Java keystore, configure the DCS server to use your keystore.
- Configuring the DCS Agent for Custom Certificate
  After you import the certificate into the keystore, configure the DCS agent to use the same certificate.

Option 1: Creating a Key and Java Keystore and Importing a Trusted Certificate

Use `keytool`, a key and certificate management utility, to create a keystore and a signing request.

1. Create the keystore:

```
keytool -genkeypair -alias your.domain.com -storetype jks -keystore
your.domain.com.jks -validity 366 -keyalg RSA -keysize 4096
```

2. The command prompts you for identifying data:

```
1. What is your first and last name? your.domain.com
2. What is the name of your organizational unit? yourunit
3. What is the name of your organization? yourorg
4. What is the name of your City or Locality? yourcity
5. What is the name of your State or Province? yourstate
6. What is the two-letter country code for this unit? US
```

3. Create the certificate signing request (CSR):

```
keytool -certreq -alias your.domain.com -file your.domain.com.csr
-keystore your.domain.com.jks -ext san=dns:your.domain.com
```

4. Request a Certificate Authority (CA) signed certificate:
   a. In the directory where you ran Step 1 above, locate the file *your.domain.com*.csr.
   b. Submit the file to your Certificate Authority (CA).

      Details vary from one CA to another. Typically, you submit your request through a website; then the CA contacts you to verify your identity. CAs can send signed reply files in a variety of formats, and CAs use a variety of names for those formats. The CA's reply must be in PEM or PKCS#7 format.
   c. There may be a waiting period for the CA's reply.
5. Import the CA's reply. The CA's reply will provide one PKCS file or multiple PEM files.
   a. Copy the CA's files into the directory where you created the keystore in Step 1 above.
   b. Use keytool to export the certificate from the keystore:

      ```
      keytool -exportcert -alias your.domain.com -file /opt/oracle/dcs/
      conf/keystore-cert.crt
      -keystore your.domain.name.jks
      ```
6. Use `keytool` to import the keystore certificate and the CA reply files:

```
keytool -importcert -trustcacerts -alias your.domain.com
-file /opt/oracle/dcs/conf/keystore-cert.crt  -keystore /opt/
oracle/dcs/conf/dcs-ca-certs
```

To import PKCS file, run the command:

```
keytool -importcert -trustcacerts -alias your.domain.com -file
CAreply.pkcs -keystore /opt/oracle/dcs/conf/dcs-ca-certs
```

CAreply.pkcs is the name of the PKCS file provided by the CA and *your.domain.com* is the complete domain name of your server.

If the CA sent PEM files, then there may be one file, but most often there are two or three. Import the files to your keystore with commands in the order shown below, after substituting your values:

```
keytool -importcert -alias root -file root.cert.pem -keystore /opt/
oracle/dcs/conf/dcs-ca-certs -trustcacerts
keytool -importcert -alias intermediate -file intermediate.cert.pem /opt/
oracle/dcs/conf/dcs-ca-certs -trustcacerts
keytool -importcert -alias intermediat2 -file intermediat2.cert.pem /opt/
oracle/dcs/conf/dcs-ca-certs -trustcacerts
keytool -importcert -alias your.domain.com -file server.cert.pem /opt/
oracle/dcs/conf/dcs-ca-certs -trustcacerts
```

root.cert.pem is the name of the root certificate file and *intermediate.cert.pem* is the name of the intermediate certificate file. The root and intermediate files link the CA's signature to a widely trusted root certificate that is known to web browsers. Most, but not all, CA replies include roots and intermediates. *server.cert.pem* is the name of the server certificate file. The file links your domain name with your public key and the CA's signature.

Option 2: Packaging an Existing PEM-format Key and Certificates in a New Java Keystore

Use the OpenSSL tool to package Privacy Enhanced Mail (PEM) files in a PKCS keystore.

If you have an existing private key and certificates for your server's domain in PEM format, importing them into a Java keystore requires the OpenSSL tool. OpenSSL can package the PEM files in a PKCS keystore. Java keytool can then convert the PKCS keystore to a Java keystore.

1. Install OpenSSL.
2. Copy your private key, server certificate, and intermediate certificate into one directory.
3. Package the key and certificates into a PKCS keystore as follows:

```
openssl pkcs12 -export -in server.cert.pem -inkey private.key.pem -
certfile
intermediate.cert.pem -name "your.domain.com" -out your.domain.com.p12
```

server.cert.pem is the name of the server certificate file, *your.domain.com* is the complete domain name of your server, *private.key.pem* is the private counterpart to the public key in *server.cert.pem*, and *intermediate.cert.pem* is the name of the intermediate certificate file.

Convert the resulting PKCS keystore file, *your.domain.com*.p12 into a Java keystore.

Option 3: Converting an Existing PKCS or PFX Keystore to a Java Keystore

If you have an existing PKCS or PFX keystore for your server's domain, convert it to a Java keystore.

1. Run the command:

```
keytool -importkeystore -srckeystore your.domain.com.p12 -
srcstoretype PKCS12
-destkeystore /opt/oracle/dcs/conf/dcs-ca-certs -deststoretype jks
```

your.domain.com.p12 is the existing keystore file and *your.domain.com* is the complete domain name of your server.

2. Configure the DCS server as explained in the topic *Configuring the DCS Server to Use Custom Keystore*.

Configuring the DCS Server to Use Custom Keystore

After packaging or converting your keystore into Java keystore, configure the DCS server to use your keystore.

1. Login to the appliance.

```
ssh -l root oda-host-name
```

2. Generate the obfuscated keystore password:

```
java -cp /opt/oracle/dcs/bin/dcs-controller-n.n.n.-SNAPSHOT.jar
 org.eclipse.jetty.util.security.Password keystore-password
```

For example:

```
[root@oda]# java -cp /opt/oracle/dcs/bin/dcs-controller-2.4.18-
SNAPSHOT.jar
org.eclipse.jetty.util.security.Password test
12:46:33.858 [main] DEBUG org.eclipse.jetty.util.log
- Logging to Logger[org.eclipse.jetty.util.log] via
org.eclipse.jetty.util.log.Slf4jLog
12:46:33.867 [main] INFO org.eclipse.jetty.util.log
- Logging initialized @239ms to org.eclipse.jetty.util.log.Slf4jLog
test
OBF:1z0f1vu91vv11z0f
MD5:098f6bcd4621d373cade4e832627b4f6
[root@scaoda7s001 conf]#
```

Copy the password that starts with OBF:.

3. Update the DCS controller configuration file.

```
cd /opt/oracle/dcs/conf
```

Update the following parameters in `dcs-controller.json`:

```
"keyStorePath": "keystore-directory-path/your.domain.com.jks"
 "trustStorePath": /opt/oracle/dcs/conf/dcs-ca-certs
```

```
"keyStorePassword": "obfuscated keystorepassword"
"certAlias": "your.domain.com"
```

4. Restart the DCS Controller.

```
systemctl stop initdcscontroller
systemctl start initdcscontroller
```

5. Access the Browser User Interface at `https://oda-host-name:7093/mgmt/index.html`.

Configuring the DCS Agent for Custom Certificate

After you import the certificate into the keystore, configure the DCS agent to use the same certificate.

1. Update the DCS agent configuration file:

```
cd /opt/oracle/dcs/conf
```

Update the following parameters in the `dcs-agent.json` file:

```
"keyStorePath": "keystore-directory-path/your.domain.com.jks"
 "trustStorePath": /opt/oracle/dcs/conf/dcs-ca-certs
 "keyStorePassword": "obfuscated keystorepassword"
 "certAlias": "your.domain.com"
```

2. Restart the DCS agent:

```
systemctl stop initdcsagent
systemctl start initdcsagent
```

3. Access the agent at `https://oda-host-name:7070`.
4. Update the CLI certificates.

```
cp -f /opt/oracle/dcs/conf/dcs-ca-certs
/opt/oracle/dcs/dcscli/dcs-ca-certs
```

5. Update the DCS command-line configuration files:

```
[root@]# cd /opt/oracle/dcs/dcscli
```

Update the following parameters in `dcscli-adm.conf` and `dcscli.conf`:

```
TrustStorePath=/opt/oracle/dcs/conf/dcs-ca-certs
TrustStorePassword=keystore_password
```

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.

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 18-18 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.
- 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.

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` 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/`.

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 Zookeeper logs are stored in the directory `/opt/zookeeper/log/`. The Zookeeper logs are `zkMonitor.log`, `zookeeper.log`, and `zookeeper.out`.

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-1 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 X8-2S and X8-2M
  Review the table for the approximate amount of usable space for Oracle Database Appliance X8-2S and X8-2M.
- Usable Space on Oracle Database Appliance X8-2-HA
  Review the table for the approximate amount of usable space for high performance and high capacity storage on Oracle Database Appliance X8-2-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 Performanc e)	`/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 X8-2S and X8-2M

Review the table for the approximate amount of usable space for Oracle Database Appliance X8-2S and X8-2M.

Oracle Database Appliance X8-2S and X8-2M 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 X8-2S and X8-2M

Number of Drives	Raw Storage	Usable Storage with Normal Redundancy (Double Mirroring)	Usable Storage with High Redundancy (Triple Mirroring)
X8-2S/X8-2M (2 drives)	12.8 TB	5.8 TB	Not Applicable
X8-2M (4 drives)	25.6 TB	9.9 TB	6.6 TB
X8-2M (6 drives)	38.4 TB	14.8 TB	9.9 TB
X8-2M (8 drives)	51.2 TB	19.8 TB	13.2 TB
X8-2M (10 drives)	64 TB	24.7 TB	16.5 TB
X8-2M (12 drives)	76.8 TB	29.7 TB	19.8 TB

Usable Space on Oracle Database Appliance X8-2-HA

Review the table for the approximate amount of usable space for high performance and high capacity storage on Oracle Database Appliance X8-2-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 X8-2-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 X8-2-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 X8-2-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 X8-2-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: 252 TB	SSDs: 17.8 TB HDDs: 97 TB	SSDs: 11.9 TB HDDs: 65 TB
12 SSD drives and 36 HDD drives	SSDs: 92 TB HDDs: 504 TB	SSDs: 35.6 TB HDDs: 195 TB	SSDs: 23.7 TB HDDs: 130 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.
`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.	Total usable storage after formatting to Oracle ASM disk groups. For example, for 2 NVMe drives, total usable storage is 5.8TB.
`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, and save an existing configuration as a JSON file 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 Multi-Node Appliance with the CLI
  Follow the JSON file example to create a JSON file to deploy a multi-node 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, and save an existing configuration as a JSON file 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.

Saving an Appliance Configuration from the Browser User Interface

You can save an appliance configuration and use it to create another appliance.

1. Log into the Browser User Interface.
2. If you have already configured your appliance, then in the Create Appliance page, click **Save Configuration**.

3. The configuration is saved as a JSON file on the local machine, being used to access 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.

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.

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" : "12.2.0.1",
    "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",
    "snmpVersion":"v2"
  }
}
```

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" : "12.2.0.1",
    "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",
    "snmpVersion":"v2"
  }
}
```

Example JSON File to Create a Multi-Node Appliance with the CLI

Follow the JSON file example to create a JSON file to deploy a multi-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:**
>
> 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 Multi-Node Oracle Database Appliance with Role Separation

The following is an example of a JSON file that creates a multi-node 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
      },
      {
         "diskGroupName":"REDO",
         "redundancy":"HIGH",
         "diskPercentage":null
      }
   ],
   "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":"12.2.0.1",
      "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
}
```

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 and to acquire an appropriate license.

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 odb1 and odb1s, 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 X8-2S, X8-2M, and X8-2-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 X8-2S OLTP Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)
odb1s	1	2	1	200	4	32
odb1	1	4	2	200	4	32
odb2	2	8	4	400	4	32
odb4	4	16	8	800	4	64
odb6	6	24	12	1200	8	256
odb08	8	32	16	1600	8	256
odb10	10	40	20	2000	8	256
odb12	12	48	24	2400	16	256
odb16	16	64	32	3200	16	256

Table D-2 Oracle Database Appliance X8-2-HA and X8-2M OLTP Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)
odb1s	1	2	1	200	4	32
odb1	1	4	2	200	4	32
odb2	2	8	4	400	4	32
odb4	4	16	8	800	4	64
odb6	6	24	12	1200	8	256
odb08	8	32	16	1600	8	256
odb10	10	40	20	2000	8	256
odb12	12	48	24	2400	16	256
odb16	16	64	32	3200	16	256
odb20	20	80	40	4000	16	256
odb24	24	96	48	4800	16	256
odb28	28	112	56	5600	16	256
odb32	32	128	64	6400	16	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 X8-2S, X8-2M, and X8-2-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 X8-2S In-Memory Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	In-Memory (GB)	Processes	Log File Size (GB)	Log Buffer (MB)
odb1s	1	2	1	1	200	4	32
odb1	1	4	2	2	200	4	32
odb2	2	8	4	4	400	4	32
odb4	4	16	8	8	800	4	64
odb6	6	24	12	12	1200	8	256
odb08	8	32	16	16	1600	8	256
odb10	10	40	20	20	2000	8	256
odb12	12	48	24	24	2400	16	256
odb16	16	64	32	32	3200	16	256

Table D-4 Oracle Database Appliance X8-2M and X8- 2-HA In-Memory Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	In-Memory (GB)	Processes	Log File Size (GB)	Log Buffer (MB)
odb1s	1	2	1	1	200	4	32
odb1	1	4	2	2	200	4	32
odb2	2	8	4	4	400	4	32
odb4	4	16	8	8	800	4	64
odb6	6	24	12	12	1200	8	256
odb08	8	32	16	16	1600	8	256
odb10	10	40	20	20	2000	8	256
odb12	12	48	24	24	2400	16	256
odb16	16	64	32	32	3200	16	256
odb20	20	80	40	42	4000	16	256
odb24	24	96	48	48	4800	16	256
odb28	28	112	56	56	5600	16	256
odb32	32	128	64	64	6400	16	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 X8-2S, X8-2M, and X8-2-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 X8-2S DSS Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Redo log file size (GB)	Log buffer (MB)
odb1s	1	1	2	200	4	32
odb1	1	2	4	200	4	32
odb2	2	4	8	400	4	32
odb4	4	8	16	800	4	64
odb6	6	12	24	1200	8	256
odb08	8	16	32	1600	8	256
odb10	10	20	40	2000	8	256
odb12	12	24	48	2400	16	256
odb16	16	32	64	3200	16	256

Table D-6 Oracle Database Appliance X8-2M and X8-2-HA DSS Database Shapes

Shape	CPU Cores	SGA (GB)	PGA (GB)	Processes	Log File Size (GB)	Log Buffer (MB)
odb1s	1	1	2	200	4	32
odb1	1	2	4	200	4	32
odb2	2	4	8	400	4	32
odb4	4	8	16	800	4	64
odb6	6	12	24	1200	8	256
odb08	8	16	32	1600	8	256
odb10	10	20	40	2000	8	256
odb12	12	24	48	2400	16	256
odb16	16	32	64	3200	16	256
odb20)	20	40	80	4000	16	256
odb24	24	48	96	4800	16	256
odb28	28	56	112	5600	16	256
odb32	32	64	128	6400	16	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, *18-18*, *18-20*
administrative account users, *9-2*
administrative accounts, *9-2*
agent
    update, *16-75*–*16-78*
agent proxy setting, *12-10*
ASM, *B-1*
ASM Cluster File System
    See Oracle ACFS
ASR
    configure, *16-197*
    delete, *16-199*
    details, *16-200*
    test, *16-200*
    update, *16-201*
Automatic Storage Management
    See Oracle ASM

B

back up database, *12-13*
backup and recovery
    CLI commands, *16-91*
    RMAN, *9-3*
backup databases, *12-1*
backup location
    disk group, *1-1*
backup policy, *12-16*, *12-17*
    update, *12-13*
backup to external FRA, *12-11*
backup to internal FRA, *12-11*
backup to NFS location, *12-11*
backup to Object Store, *12-11*
bare metal patching, *7-8*, *7-23*, *7-25*
bill of materials, *18-8*
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, *15-34*
    clone vm, *15-41*
    create CPU pool, *14-3*
    create dbsystem, *15-10*
    create vdisk, *15-33*
    create vm, *15-37*
    create vmstorage, *15-26*
    create vnetwork, *15-29*
    delete cpupool, *14-7*
    delete dbsystem, *15-20*
    delete vdisk, *15-36*
    delete vm, *15-44*
    delete vnetwork, *15-32*
    delete-vmstorage, *15-28*
    describe CPU pool, *14-6*
    describe dbsystem, *15-15*
    describe storage, *15-27*
    describe vdisk, *15-34*
    describe vm, *15-43*
    describe vnetwork, *15-30*
    list dbsystems, *15-15*
    modify cpupool, *14-6*
    modify dbsystem, *15-16*, *15-18*, *15-21*
    modify vdisk, *15-35*
    modify vm, *15-42*
    modify vmstorage, *15-27*
    modify vnetwork, *15-31*
    remap cpupools, *14-5*
    start dbsystem, *15-19*
    start vm, *15-43*
    start vnetwork, *15-31*
    stop dbsystem, *15-19*

C

cabling, *3-1*, *3-4*
    validate, *16-215*

cleanup, *7-29*
cleanup utility, *8-1*
cleanup.pl, *8-1*, *18-12*
CLI commands
  configure, *16-14*
cloning database, *9-16*, *9-17*
Configuration Audit Tool (ORAchk), *18-14*
configure CLI commands
  configure-firstnet, *16-14*
configure-firstnet, *4-1*
CPU
  commands, *16-126*
CPU core
  current configuration, *16-127*
  enable, *16-127*
CPU cores
  configuration history, *16-126*
  list, *16-126*
cpu pool
  creating, *14-3*
CPU pool
  audit record, *14-8*
  deleting, *14-7*
  modifying, *14-6*
  over-subscribed, *14-8*
  remapping, *14-5*
  viewing, *14-6*
cpu pools
  about, *14-1*
cpupool
  create, *16-130*, *17-28*
  delete, *16-134*, *17-32*
  describe, *16-133*, *17-31*
  modify, *16-133*, *17-31*
cpupools
  auditrecord, *16-134*, *16-135*, *17-32*, *17-33*
  list, *16-132*, *17-30*
  remap, *16-131*, *17-29*
create appliance, *4-7*

D

DATA disk group, *B-4*
database, *1-1*, *9-21*, *D-2*–*D-4*
  backup policy, *12-12*
  block size, *1-1*
  clone from backup, *9-14*
  create, *9-9*
  delete, *9-20*, *9-23*
  details, *9-8*
  home, *9-21*
    create multiple, *9-21*
    multiple, *9-21*
  language, *1-1*
  list, *9-8*
database *(continued)*
  modify, *9-18*
  move, *9-19*
  register, *16-152*
  template
    See shape, database
  territory, *1-1*
  upgrade, *9-20*
database backup
  delete, *12-14*
database classes, *D-1*
database home
  acfs, *9-12*
  create, *9-22*
  details, *9-22*
  display details, *16-162*
  list, *9-22*
database home storage
  configure, *16-157*
  describe, *16-160*
  list, *16-158*
  modify, *16-161*
database homes
  ACFS, *9-3*
Database Security Assessment report, *18-15*
database shapes, *D-1*–*D-4*
  about, *D-1*
database storage, *B-1*
  commands, *16-164*
  create, *16-165*
  display details, *16-166*, *16-167*
  list, *16-171*
database system
  backup, restore and recovery, *15-24*
  creating, *15-10*
  delete, *15-20*
  describing, *15-15*
  modifying, *15-16*, *15-21*
  network, *15-18*
  Oracle Data Guard, *15-25*
  start, *15-19*
  stop, *15-19*
database systems
  list, *15-15*
database, register, *9-27*, *9-28*
db system patching, *7-16*
dbsystem
  create, *17-6*
  delete, *17-19*
  describe, *17-11*, *17-17*
  modify, *17-7*, *17-9*
  start, *17-18*
  stop, *17-18*
dbsystems
  list, *17-10*

dcserr, *18-2*
default groups and users, *A-3*
deleting backups, *12-25*
DHCP (Dynamic Host Configuration Protocol)
  connect to Oracle ILOM, *3-9*
  Oracle ILOM configuration, *3-9*
diagnostics, *18-2*
directory paths, *A-1*
disk
  replace, *10-3*
disk group redundancy, *B-1*
disk group sizes, *A-3*
disk group storage
  describe, *16-168*
  list, *16-171*
domain
  definition, *15-4*
downloading software, *4-5*
DSS
  database shape, *D-4*
Dynamic Host Configuration Protocol
  See DHCP

E

electrical power cords
  connecting, *3-9*
EM Express, *9-34*
external FRA, *12-1*

F

flex diskgroup, *9-9*, *9-12*, *9-18*

G

Grid user, *9-2*
groups and users defaults, *A-3*
guest virtual machine
  definition, *15-4*

H

Host Public Addresses
  IP address, *1-11*

I

IMDB
  database shape, *D-3*
install software, *4-6*
installed version
  display, *16-50*, *16-52*, *16-57*, *16-58*, *16-60*
instance caging, *9-33*
  enable, *9-33*
Integrated Lights Out Manager
  See Oracle ILOM
internal FRA, *12-1*

J

JSON file, *15-45*, *15-48*, *C-2*, *C-7*

K

KVM, *17-1*, *17-3*
  definition, *15-3*
  deploying, *15-25*
  overcomit, *15-45*
KVM commands
  odacli clone-vdisk, *17-26*
  odacli clone-vm, *17-45*
  odacli create-cpupool, *16-130*, *17-28*
  odacli create-dbsystem, *17-6*
  odacli create-vdisk, *17-24*
  odacli create-vm, *17-39*
  odacli create-vmstorage, *17-21*
  odacli create-vnetwork, *17-35*
  odacli delete-cpupool, *16-134*, *17-32*
  odacli delete-dbsystem, *17-19*
  odacli delete-vdisk, *17-26*
  odacli delete-vm, *17-46*
  odacli delete-vmstorage, *17-23*
  odacli delete-vnetwork, *17-38*
  odacli describe-auditrecord, *16-135*, *17-33*
  odacli describe-cpupool, *16-133*, *17-31*
  odacli describe-dbsystem, *17-11*
  odacli describe-dbsystem-image, *17-17*
  odacli describe-vdisk, *17-25*
  odacli describe-vm, *17-41*
  odacli describe-vmstorage, *17-22*
  odacli describe-vnetwork, *17-36*
  odacli list-auditrecords, *16-134*, *17-32*
  odacli list-cpupools, *16-132*, *17-30*
  odacli list-dbsystems, *17-10*
  odacli list-vdisks, *17-25*
  odacli list-vms, *17-41*
  odacli list-vmstorages, *17-22*
  odacli list-vnetworks, *17-36*
  odacli migrate-vm, *17-45*
  odacli modify-cpupool, *16-133*, *17-31*
  odacli modify-dbfileattributes, *17-9*
  odacli modify-dbsystem, *17-7*
  odacli modify-vdisk, *17-25*
  odacli modify-vm, *17-41*
  odacli modify-vmstorage, *17-23*
  odacli modify-vnetwork, *17-37*
  odacli remap-cpupools, *16-131*, *17-29*

KVM commands *(continued)*
odacli start-dbsystem, *17-18*
odacli start-vm, *17-44*
odacli start-vnetwork, *17-36*
odacli stop-dbsystem, *17-18*
odacli stop-vm, *17-44*
odacli stop-vnetwork, *17-37*

L

LED
status when blinking, *3-11*
status when steady, *3-11*
status when steady on, *3-11*
LED indicators
NVMe, *10-3*
Linux
and virtualization, *15-2*, *15-7*
listener.ora, configure, *9-24*
load configuration, *C-1*
log files, *A-2*
sending to Oracle Support Services, *18-30*
logs
storage diagnostic, *16-83*, *16-189*

M

migration
from an existing database, *9-3*
multi-node system, *10-5*
multi-user access
appliance, *5-2*
create user, *5-9*
entitlements, *5-12*
manage database, *5-13*
manage database homes, *5-13*
manage password, *5-14*
odacli commands, *16-7*
provisioning, *5-6*
reset-password, *5-10*, *5-15*
resources, *5-11*
roles, *5-12*
shared resource, *5-6*
user roles, *5-4*
multiple database instances
instance caging, *9-33*
multiple Oracle homes, *9-21*
create, *9-21*
My Oracle Support, *18-30*
send log files for troubleshooting, *18-30*

N

network
configure, *16-14*
create, *11-5*
delete, *11-7*
interfaces, *11-5*
list, *11-5*
update, *11-6*
network cabling, *3-6*
network infrastructure, *11-1*, *11-2*
network interface
add, *11-6*
default values, *11-1*, *11-2*
delete, *11-7*
network ports, *3-4*, *3-6*
network time protocol service (NTP service), *1-1*
NFS backup, *12-9*
NTP service
See network time protocol service
NVMe
disk location, *10-3*
LED indicators, *10-3*

O

oakcli commands
show power
example, *18-23*
Object Store, *12-10*
credentials, *16-96*, *16-100*, *16-106*, *16-114*
odaadmcli commands
asmops diskgroup, *16-211*
expand storage, *16-212*
hardware configuration, *16-205*
hardware monitoring, *16-205*
manage diagcollect, *16-189*, *18-30*
orachk, *16-83*
power disk, *16-215*
show controller, *16-216*
show cooling, *16-205*
show disk, *16-217*
show diskgroup, *16-219*
show env_hw, *16-206*
show fs, *16-206*
show memory, *16-207*
show network, *16-208*
show power, *16-209*
show processor, *16-209*
show raidsyncstatus, *16-220*
show server, *16-210*
show storage, *16-220*
storage, *16-211*
stordiag, *16-221*
odacli cleanup-patchrepo, *7-29*

odacli commands
- appliance, *16-84*
- apply patch, *16-48*
- backup, *16-91*
- cleanup-patchrepo, *16-49*
- clone-database, *16-137*
- clone-vdisk, *15-34*
- clone-vm, *15-41*
- configure-asr, *16-197*
- configure-dataguard, *16-116*
- configure-dbhome-storage, *16-157*
- configure-pmem, *16-16*
- CPU core, *16-126*
- cpu pool, *16-129*, *17-27*
- create database, *9-21*
- create vdisk, *15-33*
- create-appliance, *16-85*
- create-auto-logclean-policy, *16-191*
- create-backup, *16-93*
- create-backupconfig, *16-95*
- create-cpupool, *14-3*
- create-database, *16-139*
- create-dbhome, *16-161*
- create-dbstorage, *16-165*
- create-dbsystem, *15-10*
- create-jobdata-retention-policy, *16-175*
- create-logcleanjob, *16-192*
- create-network, *16-17*
- create-objectstoreswift, *16-96*
- create-prepatchreport, *16-50*
- create-purge-jobdata-job, *16-177*
- create-vm, *15-37*
- create-vmstorage, *15-26*
- create-vnetwork, *15-29*
- database, *16-136*
- database storage, *16-164*
- dataguard, *16-116*
- DBHome, *16-157*
- dbsystem, *17-5*
- deconfigure-dataguard, *16-121*
- delete-asr, *16-199*
- delete-backup, *16-97*
- delete-backupconfig, *16-98*
- delete-cpupool, *14-7*
- delete-database, *16-143*
- delete-dbhome, *16-162*
- delete-dbstorage, *16-166*
- delete-dbsystem, *15-20*
- delete-jobdata-retention-policy, *16-181*
- delete-network, *16-18*
- delete-networkinterface, *16-20*
- delete-objectstoreswift, *16-100*
- delete-prepatchreport, *16-58*
- delete-vdisk, *15-36*
- delete-vm, *15-44*

odacli commands *(continued)*
- delete-vmstorage, *15-28*
- delete-vnetwork, *15-32*
- describe-asr, *16-200*
- describe-backupreport, *16-99*
- describe-component, *16-60*
- describe-cpucore, *16-127*
- describe-cpupool, *14-6*
- describe-database, *16-145*
- describe-dataguardstatus, *16-123*
- describe-dbhome, *16-163*
- describe-dbhome-storage, *16-160*
- describe-dbstorage, *16-167*
- describe-dbsystem, *15-15*
- describe-dgstorage, *16-168*
- describe-job, *16-182*
- describe-latestpatch, *16-67*
- describe-logcleanjob, *16-194*
- describe-network, *16-21*
- describe-networkinterface, *16-22*
- describe-prepatchreport, *16-52*
- describe-schedule, *16-101*
- describe-system, *16-86*
- describe-vdisk, *15-34*
- describe-vm, *15-43*
- describe-vmstorage, *15-27*
- describe-vnetwork, *15-30*
- failover-dataguard, *16-123*
- irestore-database, *16-102*
- jobs, *16-175*
- list scheduler jobs, *16-188*
- list-auto-logclean-policy, *16-194*
- list-backupconfig, *16-105*
- list-backupreports, *16-106*
- list-cpucores, *16-126*
- list-cpupools, *14-6*
- list-databases, *16-146*
- list-dataguardstatus, *16-124*
- list-dbhome-storage, *16-158*
- list-dbhomes, *16-164*
- list-dbstorages, *16-171*
- list-dbsystems, *15-15*
- list-dgdisks, *16-213*
- list-featuretracking, *16-89*
- list-jobdata-retention-policy, *16-186*
- list-jobs, *16-182*
- list-logcleanjobs, *16-195*
- list-logspaceusage, *16-196*
- list-networkinterfaces, *16-24*
- list-networks, *16-23*
- list-osconfigurations, *16-203*
- list-prepatchreports, *16-57*
- list-purge-jobdata-job, *16-187*
- list-scheduled-executions, *16-188*
- list-schedules, *16-107*

odacli commands *(continued)*
list-tdebackupreports, *16-108*
list-vdisks, *15-34*
list-vms, *15-43*
list-vmstorages, *15-27*
list-vnetworks, *15-30*
modify-cpupool, *14-6*, *15-21*
modify-database, *16-147*
modify-dbhome-storage, *16-161*
modify-dbstorage, *16-172*
modify-dbsystem, *15-16*, *15-18*
modify-vdisk, *15-35*
modify-vm, *15-42*
modify-vmstorage, *15-27*
modify-vnetwork, *15-31*
move-database, *16-151*
multi-user access, *16-26*
network, *16-13*
odacli activate-user, *16-32*
odacli authorize-user, *16-35*
odacli change-password, *16-47*
odacli create-appliance, *C-2*, *C-7*
odacli create-dbsystem, *15-45*, *15-48*
odacli create-user, *16-31*
odacli delete-agentconfig-parameters, *16-73*
odacli delete-user, *16-35*
odacli describe-domain, *16-28*
odacli describe-resource-access, *16-43*
odacli describe-tenant, *16-30*
odacli describe-user, *16-34*
odacli describe-user-entitlement, *16-40*
odacli describe-user-operation, *16-41*
odacli describe-user-role, *16-38*
odacli grant-resource-access, *16-44*
odacli list-agentconfig-parameters, *16-68*
odacli list-availablepatches, *16-59*
odacli list-dgstorages, *16-171*
odacli list-domains, *16-28*
odacli list-objectstoreswifts, *16-106*
odacli list-resources-access, *16-42*
odacli list-tenants, *16-29*
odacli list-user-entitlements, *16-38*
odacli list-user-operations, *16-41*
odacli list-user-roles, *16-37*
odacli list-users, *16-33*
odacli reset-password, *16-46*
odacli revoke-resource-access, *16-45*
odacli update-agentconfig-parameters, *16-70*
Oracle ASR, *16-197*
OS, *16-203*
odacli commands *(continued)*
recover-database, *16-109*
recovery, *16-91*
register-database, *16-152*
reinstate-dataguard, *16-125*
remap-cpupools, *14-5*
restore-archivelog, *16-111*
restore-tdewallet, *16-112*
set-credential, *16-85*
show -h
with Oracle Database Appliance Hardware Monitoring Tool, *18-23*
start-dbsystem, *15-19*
start-vm, *15-43*
start-vnetwork, *15-31*
stop-dbsystem, *15-19*
stop-vm, *15-43*
stop-vnetwork, *15-31*
storage, *16-211*
switchover-dataguard, *16-125*
test-asr, *16-200*
update, *16-48*
update database home, *16-74*
update repository, *16-80*
update-asr, *16-201*
update-backupconfig, *16-113*
update-cpucore, *16-127*
update-dcsadmin, *16-75*
update-dcsagent, *16-76*
update-dcscomponents, *16-77*
update-dgstorage, *16-174*
update-network, *16-25*
update-objectstoreswift, *16-114*
update-osconfigurations, *16-204*
update-registry, *16-78*
update-schedule, *16-115*
update-server, *16-81*
update-storage, *16-83*
upgrade-database, *16-155*
validate storage topology, *16-215*
validate-storagetopology, *16-215*
vdisk, *17-23*
vm, *17-38*
vmstorage, *17-21*
vnetwork, *17-34*
odacli create-appliance
example JSON files, *C-2*, *C-7*
readme, *C-2*
odacli create-dbsystem
example JSON files, *15-45*, *15-48*
odaeraser.py, *8-5*
OINSTALL group, *9-2*
OLTP
database shape, *D-2*

Optimal Flexible Architecture
and Oracle homes, A-1
ORAchk report, 18-15
Oracle ACFS, 15-4
Oracle ACFS (Oracle Automatic Storage Management Cluster File System)
virtual storage, 15-4
Oracle ASM, 10-1, A-3
Oracle ASM (Oracle Automatic Storage Management)
data migration, 9-3
user group, A-3
Oracle ASR
commands, 16-197
configure, 16-197
Oracle ASR (Oracle Auto Service Request), 1-1
Oracle Data Guard, 13-1
active data guard, 13-22
CLI commands, 16-116
configure, 13-2
deconfigure, 13-22
failover, 13-12
migrating, 13-15
network, 13-14
patching, 13-20
reinstate, 13-12
status, 13-12
switchover, 13-12
upgrade database, 13-21
Oracle Database
See database
Oracle Database Appliance Hardware Monitoring Tool, 18-23
Oracle Database Appliance Manager
software inventory, 2-3
Oracle Database Resource Manager
instance caging, 9-33
Oracle Enterprise Manager Database Express, 9-34
Oracle home
multiple, 9-21
Oracle ILOM (Oracle Integrated Lights Out Manager), 1-1
configure, 3-9
Oracle Integrated Lights Out Manager
See Oracle ILOM
Oracle Ksplice, 7-26
Oracle KVM, 15-2, 15-7
Oracle KVM command-line interface
alphabetical list of commands, 17-1
Oracle ORAchk, 18-21
Oracle ORAchk Health Check Tool, 18-14
Oracle Support Services
See My Oracle Support
Oracle Trace File Analyzer Collection, 18-21
Oracle user, 9-2
OS configuration
current, 16-203, 16-204
suggested, 16-203, 16-204
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, 14-8

P

patches
software inventory, 2-3
patching
appliance, 7-2, 7-7
using Browser User Interface, 7-19, 7-21, 7-24
patching Oracle Database Appliance, 7-8, 7-16, 7-23, 7-25
performance tuning, 14-8
plumb network, 4-1
power cords
connecting, 3-9
processor_group_name, 14-1
proxy setting, 12-10

R

readme, C-2
recover database, 12-20
recovering database, 12-14
reimage appliance, 8-4
resource management
CPU pools, 14-8
Oracle KVM, 15-2, 15-7
supported virtual operating systems, 15-2, 15-7
restore archivelog, 12-25
restore baremetal system, 8-4
restore database, 12-22
RMAN (Recovery Manager utility)
database management, 9-3
RPM drift, 18-8

S

sanitize diagnostics, 18-18, 18-20
save configuration, C-1
scheduler
jobs, 16-188
Scheduling Database Backups, 12-19

secure erase tool, *8-5*
odaeraser.py, *8-5*
sending log files to Oracle Support Services, *18-30*
shapes, *D-1*
database
choosing, *D-1*
OLTP, *D-2–D-4*
space management, *B-2*
SQL*Loader, *9-3*
Standard Edition High Availability, *9-5*
start up the system, *3-11*
static listener, configure, *9-24*
storage, *10-1*, *10-5*, *B-4*
expand, *10-3*
single-node, *10-2*
storage cables, *3-1*
storage disks
location, *10-3*

T

TDE, *9-6*
TFA command, *18-16*
TFA options, *18-16*
Trace File Analyzer (TFA) Collector, *18-16*
troubleshooting, *18-2*, *18-3*, *18-12*
log files for Oracle Support Services, *18-30*
turning on Oracle Database Appliance, *3-11*

U

uninstall, *8-1*
update-dcscomponents, *18-3*
upgrade
database, *7-29*
usable storage, *B-4*

V

viewing backup reports, *12-14*
virtual disk
clone, *17-26*
cloning, *15-34*
create, *17-24*
creating, *15-33*
delete, *17-26*
deleting, *15-36*
describe, *17-25*
list, *17-25*
modify, *17-25*
modifying, *15-35*
viewing, *15-34*
virtual disks, *15-4*
virtual local area network, *11-1*, *11-2*
virtual machine
about, *15-4*
assemblies, *15-4*
clone, *17-45*
cloning, *15-41*
create, *17-39*
creating, *15-37*
delete, *17-46*
deleting, *15-44*
describe, *17-41*
list, *17-41*
migrate, *17-45*
modify, *17-41*
modifying, *15-42*
start, *15-43*, *17-44*
stop, *15-43*, *17-44*
templates, *15-4*
viewing, *15-43*
virtual network
creating, *15-29*
deleting, *15-32*
modifying, *15-31*
start, *15-31*
stop, *15-31*
viewing, *15-30*
virtual storage
about, *15-4*
virtualization, *15-2*, *15-7*
VLAN, *11-1*
vm storage
creating, *15-26*
deleting, *15-28*
modifying, *15-27*
viewing, *15-27*
vmstorage
create, *17-21*
delete, *17-23*
describe, *17-22*
modify, *17-23*
vmstorages
list, *17-22*
vnetwork
create, *17-35*
delete, *17-38*
describe, *17-36*
modify, *17-37*
start, *17-36*
stop, *17-37*
vnetworks
list, *17-36*

W

web interface
web interface *(continued)*
Oracle Enterprise Manager Database Express, *9-34*