Oracle® Cloud at Customer

Administering Oracle Database Exadata Cloud at Customer (Gen 1/OCI-C)

Release 18.4.6
E97325-43
August 2022

ORACLE

Oracle Cloud at Customer Administering Oracle Database Exadata Cloud at Customer (Gen 1/OCI-C),
Release 18.4.6

E97325-43

Contents

Preface

Audience x
Documentation Accessibility x
Related Documents x
Conventions xi

1 Getting Started with Exadata Cloud at Customer

About Oracle Database Exadata Cloud at Customer 1-1
About Exadata Cloud at Customer Instances 1-3
    Exadata System Configuration 1-3
    Exadata Storage Configuration 1-5
    Multiple VM Clusters 1-7
    CPU Oversubscription 1-9
About Exadata Cloud at Customer Database Deployments 1-10
    Service Level 1-10
    Oracle Database Software Release 1-11
    Oracle Grid Infrastructure Software Release 1-11
    Oracle Database Software Edition 1-12
    Oracle Database Type 1-12
    Automatic Backup Configuration 1-12
    Data Security 1-13
        Security of Data at Rest 1-13
        Security of Data in Transit 1-14
Before You Begin with Exadata Cloud at Customer 1-14
About Exadata Cloud at Customer Roles and Users 1-15
Accessing the My Services Dashboard and the Oracle Database Cloud Service Console 1-16
Identifying the Exadata Cloud at Customer Release Version 1-17
Using the Exadata Cloud at Customer REST APIs 1-17
Typical Workflow for Using Exadata Cloud at Customer 1-18

2 Managing the Exadata Cloud at Customer Life Cycle

Creating an Exadata Cloud at Customer Instance 2-1
Creating a Database Deployment 2-6
Creating a Database Deployment Using a Cloud Backup 2-17
    Creating a Cloud Backup of an On-Premises Database 2-18
    Replacing the Database by Using the Oracle Database Cloud Service Console 2-20
    Replacing the Database by Using ibkup Actions 2-22
Creating a Clone Database Deployment from a Snapshot Master 2-30
Viewing All Database Deployments 2-33
Viewing Detailed Information for a Database Deployment 2-34
Viewing Activities for Database Deployments in an Identity Domain 2-34
Stopping, Starting, and Removing Database Instances 2-35
Stopping, Starting, and Restarting Compute Nodes 2-37
Scaling an Exadata Cloud at Customer Instance 2-39
    Modifying the Compute Node Processing Power 2-39
    Adding Exadata Storage 2-41
    Adding Compute Nodes 2-42
Suspending and Resuming an Exadata Cloud at Customer Instance 2-44
Creating and Managing Snapshots of a Database Deployment 2-44
Deleting a Database Deployment 2-47
Deleting an Exadata Cloud at Customer Instance 2-48

3 Managing Network Access to Exadata Cloud at Customer

About Network Access to Exadata Cloud at Customer 3-1
Generate a Secure Shell (SSH) Public/Private Key Pair 3-2
    Generate an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility 3-2
    Generate an SSH Key Pair on Windows Using the PuTTYgen Program 3-3
Creating an SSH Tunnel to a Compute Node Port 3-4
    Creating an SSH Tunnel Using the ssh Utility on Linux 3-4
    Creating an SSH Tunnel Using the PuTTY Program on Windows 3-5
Controlling Network Access to Exadata Cloud at Customer 3-6
Defining a Custom Host Name or Domain Name for Exadata Cloud at Customer 3-7
Defining a Custom SCAN Host Name for Exadata Cloud at Customer 3-8
Using Network Encryption and Integrity 3-8

4 Accessing Exadata Cloud at Customer

Connecting to a Compute Node Through Secure Shell (SSH) 4-1
    Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms 4-1

Connecting to a Compute Node Using the PuTTY Program on Windows 4-2
Accessing Enterprise Manager Database Express 18c, or later 4-3
Accessing Enterprise Manager Database Express 12c 4-5
Accessing Enterprise Manager 11g Database Control 4-7
Connecting Remotely to the Database by Using Oracle Net Services 4-9

5 Administering Exadata Cloud at Customer

Using Exadata I/O Resource Management 5-1
Adding an SSH Public Key 5-2
Removing an SSH Public Key 5-3
Updating the Cloud Tooling on Exadata Cloud at Customer 5-4
Administering VM Clusters 5-6
Viewing Information About Existing VM Clusters 5-6
Creating a VM Cluster 5-7
Modifying an Existing VM Cluster 5-9
Deleting a VM Cluster 5-13
Administering Oracle Homes 5-14
Viewing Information About Oracle Homes 5-14
Viewing Information About Oracle Homes by Using the dbaascli Utility 5-15
Moving a Database to Another Oracle Home 5-15
Creating an Oracle Home 5-16
Deleting an Oracle Home 5-16
Administering Software Images 5-17
Viewing Information About Downloaded Software Images 5-18
Viewing Information About Available Software Images 5-19
Downloading a Software Image 5-19
Activating a Software Image 5-20
Deleting a Software Image 5-21
Administering a Data Guard Configuration 5-22
Performing a Switchover Operation 5-22
Performing a Manual Failover Operation 5-23
Reinstating a Failed Primary Database 5-23
Configuring Clients for Automatic Failover 5-24
Administering Pluggable Databases 5-25
Registering a Database 5-26
Maintaining the Manageability of Exadata Cloud at Customer 5-28
Loading Data into the Oracle Database on Exadata Cloud at Customer 5-29
Tuning Oracle Database Performance on Exadata Cloud at Customer 5-30
Monitoring and Managing Oracle Database on Exadata Cloud at Customer 5-30
Monitoring and Managing Exadata Storage Servers on Exadata Cloud at Customer 5-31

Managing the Log and Diagnostic Files on Exadata Cloud at Customer 5-32

6 Backing Up and Restoring Databases on Exadata Cloud at Customer

About Backing Up Database Deployments on Exadata Cloud at Customer 6-1
Viewing Detailed Backup Configuration Information 6-4
Customizing the Automatic Backup Configuration 6-5
Creating an On-Demand Backup 6-9
Creating an On-Demand Backup by Using the bkup_api Utility 6-9
Deleting a Backup 6-11
Updating the Password for Backing Up to Cloud Storage 6-11
Changing to a Different Backup Destination 6-12
Changing the Backup Destination 6-13
Disabling and Re-enabling Scheduled Backups 6-15
Validating Backups 6-15
Restoring from the Most Recent Backup 6-16
Restoring from the Most Recent Backup by Using the bkup_api Utility 6-17
Restoring from a Specific Backup 6-18
Restoring from a Specific Backup by Using the bkup_api Utility 6-19
Restoring to a Specific Point in Time 6-20
Restoring to a Specific Point in Time by Using the bkup_api Utility 6-21
Restoring System and Database Configuration Files 6-22
Recover Backups Using the dbaasapi Utility 6-23
Manually Restoring from a Backup 6-24
View Details About Automatic Backups 6-24

7 Patching Exadata Cloud at Customer

About Patching Exadata Cloud at Customer 7-1
The exadbcpatchmulti Command 7-3
Listing Available Patches 7-6
Listing Available Patches by Using the exadbcpatchmulti Command 7-7
Listing Available Patches by Using the dbaascli Command 7-7
Checking Prerequisites Before Applying a Patch 7-8
Checking Prerequisites Before Applying a Patch by Using the exadbcpatchmulti Command 7-9
Checking Prerequisites Before Applying a Patch by Using the dbaascli Command 7-10
Applying a Patch 7-11
Applying a Patch by Using the exadbcpatchmulti Command 7-13
Applying a Patch by Using the dbaascli Command 7-15
How Applying a Patch Impacts the Oracle Homes 7-16
Listing Applied Patches 7-17

Rolling Back a Patch or Failed Patch 7-18
Rolling Back a Patch or Failed Patch by Using the exadbcpatchmulti Command 7-18
Rolling Back a Patch or Failed Patch by Using the dbaascli Command 7-20

8 Configuring Database Features, Database Options, and Companion Products

Using Oracle Data Guard in Exadata Cloud at Customer 8-1
Using Oracle Multitenant in Exadata Cloud at Customer 8-3
Creating and Activating a Master Encryption Key for a PDB 8-4
Exporting and Importing a Master Encryption Key for a PDB 8-6
Using Oracle GoldenGate Cloud Service with Exadata Cloud at Customer 8-7
Tablespace Encryption 8-7
Creating Encrypted Tablespaces 8-8
Managing Tablespace Encryption 8-8
Managing Huge Pages 8-9

9 Migrating Oracle Databases to Exadata Cloud at Customer

Choosing a Migration Method 9-1
Migration Methods 9-3
Conventional RMAN Backup and Recovery 9-3
Conventional Data Pump Export and Import 9-4
Conventional Data Pump Export and Import: Example 9-5
Transportable Tablespaces 9-6
Data Pump Transportable Tablespace: Example 9-8
Data Pump Full Transportable Export and Import 9-10
Transportable Tablespaces with Cross-Platform Incremental Backup 9-11
Transportable Database 9-13
Data Guard Physical Standby 9-14
Advanced Data Guard Migration Options 9-15
Unplugging and Plugging a Pluggable Database 9-15
Plugging in a Non-CDB 9-16
Cloning a Remote PDB or Non-CDB 9-17

10 Frequently Asked Questions for Exadata Cloud at Customer

A Characteristics of a Newly Created Deployment

Linux User Accounts A-1

Locations of Installed Software A-2
Oracle Database Characteristics A-3
Location of Diagnostic and Log Files A-3
Oracle Data Guard Configuration A-3

B Oracle Cloud Pages for Administering Exadata Cloud at Customer

Instances Page B-1
Activity Page B-3
SSH Access Page B-5
Overview Page B-6
Backup Page B-8
Patching Page B-9
Snapshots Page B-9
Create Instance: Instance Page B-9
Create Instance: Instance Details Page B-11
Create Instance: Confirmation Page B-16

C The dbaascli Utility

dbaascli cswlib download C-3
dbaascli cswlib list C-4
dbaascli database bounce C-4
dbaascli database changepassword C-4
dbaascli database move C-5
dbaascli database start C-5
dbaascli database status C-6
dbaascli database stop C-6
dbaascli database update C-6
dbaascli database upgrade C-8
dbaascli dbhome create C-8
dbaascli dbhome info C-9
dbaascli dbhome purge C-9
dbaascli dbimage activateBP C-9
dbaascli dbimage list C-10
dbaascli dbimage purge C-10
dbaascli grid upgrade C-11
dbaascli listener bounce C-11
dbaascli listener start C-12
dbaascli listener status C-12
dbaascli listener stop C-12

dbaascli orec keep list C-12
dbaascli orec keep tag C-13
dbaascli orec latest C-13
dbaascli orec list C-13
dbaascli orec pitr C-13
dbaascli orec scn C-14
dbaascli patch db apply C-14
dbaascli patch db list C-15
dbaascli patch db prereq C-16
dbaascli patch db switchback C-16
dbaascli patch tools apply C-17
dbaascli patch tools list C-18
dbaascli pdb checkdb C-18
dbaascli pdb checknode C-18
dbaascli pdb checkpdb C-19
dbaascli pdb close C-19
dbaascli pdb connect_info C-19
dbaascli pdb connect_string C-20
dbaascli pdb create C-20
dbaascli pdb delete C-21
dbaascli pdb info C-21
dbaascli pdb local_clone C-21
dbaascli pdb open C-22
dbaascli pdb remote_clone C-22
dbaascli pdb rename C-23
dbaascli pdb resize C-23
dbaascli registerdb begin C-24
dbaascli registerdb prereqs C-24
dbaascli registerdb registry C-24
dbaascli tde rotate masterkey C-25
dbaascli tde status C-25

D ExaCLI Commands

E Cloud Notification Service Events

Preface

This document describes how to manage and monitor Oracle Database Exadata Cloud at Customer and provides references to related documentation.

Topics

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This document is intended for Oracle Cloud users who want to manage and monitor Oracle Database Exadata Cloud at Customer.

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, see these Oracle resources:

- *Getting Started with Oracle Cloud*
- *Using Oracle Cloud Infrastructure Object Storage Classic*
- *What's New for Oracle Database Exadata Cloud at Customer*
- *Known Issues for Oracle Database Exadata Cloud at Customer*
- *REST API for Oracle Database Exadata Cloud at Customer*
- *Deploying Oracle Database Exadata Cloud at Customer*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
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.

1

Getting Started with Exadata Cloud at Customer

This section describes how to get started with Oracle Database Exadata Cloud at Customer for administrators and application owners.

Topics

- About Oracle Database Exadata Cloud at Customer
- About Exadata Cloud at Customer Instances
- About Exadata Cloud at Customer Database Deployments
- Before You Begin with Exadata Cloud at Customer
- About Exadata Cloud at Customer Roles and Users
- Accessing the My Services Dashboard and the Oracle Database Cloud Service Console
- Identifying the Exadata Cloud at Customer Release Version
- Using the Exadata Cloud at Customer REST APIs
- Typical Workflow for Using Exadata Cloud at Customer

About Oracle Database Exadata Cloud at Customer

> **Note:**
>
> This is the administration guide for Gen 1 Exadata Cloud at Customer. Gen 1 is the first generation of Exadata Cloud at Customer, which is deployed in conjunction with Oracle Cloud at Customer using Oracle Cloud Infrastructure Classic (OCI-C).
>
> If you are looking for Gen 2 Exadata Cloud at Customer, which is deployed in conjunction with Oracle Cloud Infrastructure, please use the latest documentation for Oracle Exadata Cloud@Customer.

Oracle Database Exadata Cloud at Customer enables you to apply the combined power of Exadata and Oracle Cloud inside your own data center. You have full access to the features and operations available with Oracle Database, but with Oracle owning and managing the Exadata infrastructure.

Each **Exadata Cloud at Customer instance** is based on an Exadata system configuration that contains some compute nodes (database servers) and Exadata Storage Servers, all tied together by a high-speed, low-latency InfiniBand network and intelligent Exadata software.

The following starter configurations are offered:

- **Base System**: Containing two compute nodes and three Exadata Storage Servers. Previously known as an Eighth Rack, a Base System is an entry-level configuration that contains Exadata Storage Servers with significantly less storage capacity and compute nodes with significantly less memory and processing power than other configurations.
- **Quarter Rack**: Containing two compute nodes and three Exadata Storage Servers.
- **Half Rack**: Containing four compute nodes and six Exadata Storage Servers.
- **Full Rack**: Containing eight compute nodes and 12 Exadata Storage Servers.

Each starter configuration is equipped with a fixed amount of memory, storage, and network resources. However, you can choose how many compute node CPU cores are enabled, up to a fixed maximum for each configuration. This choice enables you to scale a starter configuration to meet workload demands and only pay for the compute node resources that you need.

With Exadata Cloud at Customer, you can optionally add compute nodes or Exadata Storage Servers to create custom configurations up to the size of a Full Rack. This capability enables much greater flexibility so that you can scale your Exadata Cloud at Customer configuration to accommodate your workload requirements.

The Exadata Cloud at Customer compute nodes are each configured with at least one virtual machine (VM). You have root privilege for the Exadata compute node VMs, so you can load and run additional software on the Exadata compute nodes. However, you do not have administrative access to the Exadata infrastructure components, including the physical compute node hardware, network switches, power distribution units (PDUs), integrated lights-out management (ILOM) interfaces, or the Exadata Storage Servers, which Oracle administers.

Exadata Cloud at Customer is provisioned with database storage provided by Exadata Storage Servers. The storage is allocated to disk groups managed by Oracle Automatic Storage Management (ASM). You have administrative access to the ASM disk groups but no direct administrative access is provided, or required, for the Exadata Storage Servers. Exadata Cloud at Customer users seamlessly benefit from the intelligent performance and scalability of Exadata.

Subscription to Exadata Cloud at Customer can include all of the required Oracle Database software licenses, or you can choose to bring Oracle Database software licenses that you already own to Exadata Cloud at Customer. If you choose to use Oracle Database software licenses that are included with the Exadata Cloud at Customer subscription, then the included Oracle Database software licenses contain all of the features of Oracle Database Enterprise Edition, plus all of the database enterprise management packs and all of the Enterprise Edition options, such as Oracle Database In-Memory and Oracle Real Application Clusters (RAC). Exadata Cloud at Customer also comes with cloud-specific software tools that assist with automated backup, patching and upgrade operations.

Within each Exadata Cloud at Customer instance, you can create numerous database deployments. Apart from the inherent storage and processing capacity of your Exadata configuration, there is no set maximum for the number of database deployments that you can create.

When you provision a database deployment, it is configured according to best-practices, with your Oracle database already running, and with default backup jobs already scheduled. You have full administrative privileges for your database, and you can connect to your database by using Oracle Net Services. You are responsible for database administration tasks such as creating tablespaces and managing database users. You can also customize the default automated maintenance setup, and you control the recovery process in response to a database failure.

About Exadata Cloud at Customer Instances

- Exadata System Configuration
- Exadata Storage Configuration
- Multiple VM Clusters
- CPU Oversubscription

Exadata System Configuration

Oracle Database Exadata Cloud at Customer is offered in the following starter system configurations:

- **Base System**: Containing two compute nodes and three Exadata Storage Servers.

  Previously known as an Eighth Rack, a Base System is an entry-level configuration that contains Exadata Storage Servers with significantly less storage capacity and compute nodes with significantly less memory and processing power than other configurations.
- **Quarter Rack**: Containing two compute nodes and three Exadata Storage Servers.
- **Half Rack**: Containing four compute nodes and six Exadata Storage Servers.
- **Full Rack**: Containing eight compute nodes and 12 Exadata Storage Servers.

Exadata Cloud at Customer system configurations are all based on Oracle Exadata X6 or Oracle Exadata X7 systems.

Each starter system configuration is equipped with a fixed amount of memory, storage, and network resources. However, you can choose how many compute node (database server) CPU cores are enabled. This choice enables you to scale an Exadata Cloud at Customer configuration to meet workload demands and only pay for the processing power that you require. Each database server must contain the same number of enabled CPU cores.

Starter System Specifications

The following table outlines the technical specifications for each Exadata Cloud at Customer system configuration based on Oracle Exadata X6 hardware.

Specification	Eighth Rack	Quarter Rack	Half Rack	Full Rack
Number of Compute Nodes	2	2	4	8
— Total Maximum Number of Enabled CPU Cores	68	84	168	336
— Total RAM Capacity	480 GB	1440 GB	2880 GB	5760 GB
Number of Exadata Storage Servers	3	3	6	12
— Total Raw Flash Storage Capacity	19.2 TB	38.4 TB	76.8 TB	153.6 TB
— Total Raw Disk Storage Capacity	144 TB	288 TB	576 TB	1152 TB

Specification	Eighth Rack	Quarter Rack	Half Rack	Full Rack
— Total Usable Storage Capacity	42 TB	84 TB	168 TB	336 TB

The following table outlines the technical specifications for each Exadata Cloud at Customer system configuration based on Oracle Exadata X7 hardware.

Specification	Base System	Quarter Rack	Half Rack	Full Rack
Number of Compute Nodes	2	2	4	8
— Total Maximum Number of Enabled CPU Cores	44	92	184	368
— Total RAM Capacity	480 GB	1440 GB	2880 GB	5760 GB
Number of Exadata Storage Servers	3	3	6	12
— Total Raw Flash Storage Capacity	19.2 TB	76.8 TB	153.6 TB	307.2 TB
— Total Raw Disk Storage Capacity	144 TB	360 TB	720 TB	1440 TB
— Total Usable Storage Capacity	42.7 TB	106.9 TB	213.8 TB	427.6 TB

Elastic System Configuration

With Exadata Cloud at Customer, you can optionally add compute nodes or Exadata Storage Servers to create custom configurations. This option enables much greater flexibility so that you can scale your Exadata Cloud at Customer configuration to accommodate your workload requirements.

This capability, known as elastic scaling, is subject to the following conditions:

- For elastic scaling of Exadata Storage Servers:
  - Each Exadata Cloud at Customer system configuration can have an absolute maximum of 12 Exadata Storage Servers.
  - The Exadata Cloud at Customer system configuration must be based on Oracle Exadata X7 hardware or Oracle Exadata X6 hardware.
  - The Exadata Cloud at Customer service instance must be enabled to support multiple virtual machine (VM) clusters.
- For elastic scaling of Exadata compute nodes:
  - Each Exadata Cloud at Customer system configuration can have an absolute maximum of eight compute nodes. However, the practical maximum is more limited if you don't have enough free IP addresses for the additional compute nodes.

    Specifically, if your system is configured so that each VM cluster client network subnet is defined using a `/28` CIDR block (`*N.N.N.N*/28`) and each VM cluster backup network subnet is defined using a `/29` CIDR block, then your immediate expansion potential is limited to five compute servers. In such cases, expansion to more than five compute servers requires the redeployment of Exadata Cloud at Customer, which includes deleting and re-creating all of the VM clusters and database deployments on the system.

  - The Exadata Cloud at Customer system configuration must be based on Oracle Exadata X7 hardware. You cannot add compute nodes to a starter system configuration based on Oracle Exadata X6 hardware.
  - The Exadata Cloud at Customer service instance must be enabled to support multiple virtual machine (VM) clusters.

The following table outlines the key additional resources provided by each compute node that is added using elastic scaling.

Specification	Exadata Base System Compute Node X7	Exadata Compute Node X7
Maximum Number of Additional CPU Cores	22	46
Additional RAM Capacity	240 GB	720 GB

The following table outlines the key additional resources provided by each Exadata Storage Server that is added using elastic scaling.

Specification	Exadata Storage Server X6	Exadata Base System Storage Server X7	Exadata Storage Server X7
Additional Raw Flash Storage Capacity	12.8 TB	6.4 TB	25.6 TB
Additional Raw Disk Storage Capacity	96 TB	48 TB	120 TB
Additional Usable Storage Capacity	28 TB	14 TB	35.6 TB

Exadata Storage Configuration

As part of configuring each Oracle Database Exadata Cloud at Customer instance, the storage space inside the Exadata Storage Servers is configured for use by Oracle Automatic Storage Management (ASM). By default, the following ASM disk groups are created:

- The DATA disk group is primarily intended for the storage of Oracle Database data files.
- The RECO disk group is primarily used for storing the Fast Recovery Area (FRA), which is an area of storage where Oracle Database can create and manage various files related to backup and recovery, such as RMAN backups and archived redo log files.

In addition, you can optionally create the SPARSE disk group. The SPARSE disk group is required to support Exadata Cloud at Customer snapshots. Exadata snapshots enable space-efficient clones of Oracle databases that can be created and destroyed very quickly and easily. Snapshot clones are often used for development, testing, or other purposes that require a transient database.

For Exadata Cloud at Customer instances that are based on Oracle Exadata X6 hardware, there are additional system disk groups that support various operational purposes. The DBFS disk group is primarily used to store the shared Oracle Clusterware files (Oracle Cluster Registry and voting disks), while the ACFS disk groups underpin shared file systems that are used to store software binaries (and patches) and files associated with the cloud-specific tooling that resides on your Exadata Cloud at Customer compute nodes. You must not remove or disable any of the system disk groups or related ACFS file systems. You should not store your own data, including Oracle Database data files or backups, inside the system disk groups or related ACFS file systems. Compared to the other disk groups, the system disk groups are so small that they are typically ignored when discussing the overall storage capacity.

For Exadata Cloud at Customer instances that are based on Oracle Exadata X7 hardware, there are no additional system disk groups. On such instances, a small amount of space is allocated from the DATA disk group to support the shared file systems that are used to store software binaries (and patches) and files associated with the cloud-specific tooling. You should not store your own data, including Oracle Database data files or backups, inside the system related ACFS file systems.

Although the disk groups are commonly referred to as DATA, RECO and so on, the ASM disk group names contain a short identifier string that is associated with your Exadata Database Machine environment. For example, the identifier could be `C2`, in which case the DATA disk group would be named `DATAC2`, the RECO disk group would be named `RECOC2`, and so on.

As an input to the configuration process, you must make decisions that determine how storage space in the Exadata Storage Servers is allocated to the ASM disk groups:

- **Database backups on Exadata Storage** — select this configuration option if you intend to perform database backups to the Exadata storage within your Exadata Cloud at Customer environment. If you select this option more space is allocated to the RECO disk group, which is used to store backups on Exadata storage. If you do not select this option, more space is allocated to the DATA disk group, which enables you to store more information in your databases.

  > **Note:**
  >
  > Take care when setting this option. Depending on your situation, you may have limited options for adjusting the space allocation after the storage in configured.

- **Create sparse disk group?** — select this configuration option if you intend to use snapshot functionality within your Exadata Cloud at Customer environment. If you select this option the SPARSE disk group is created, which enables you to use Exadata Cloud at Customer snapshot functionality. If you do not select this option, the SPARSE disk group is not created and Exadata Cloud at Customer snapshot functionality will not be available on any database deployments that are created in the environment.

  > **Note:**
  >
  > Take care when setting this option. You cannot later enable Exadata Cloud at Customer snapshot functionality if you do not select the option to create the SPARSE disk group.

The following table outlines the proportional allocation of storage amongst the DATA, RECO, and SPARSE disk groups for each possible configuration:

Configuration settings	DATA disk group	RECO disk group	SPARSE disk group
Database backups on Exadata Storage: No **Create sparse disk group?: No**	80 %	20 %	0 % The SPARSE disk group is not created.
Database backups on Exadata Storage: Yes **Create sparse disk group?: No**	40 %	60 %	0 % The SPARSE disk group is not created.
Database backups on Exadata Storage: No **Create sparse disk group?: Yes**	60 %	20 %	20 %
Database backups on Exadata Storage: Yes **Create sparse disk group?: Yes**	35 %	50 %	15 %

Multiple VM Clusters

If your Exadata system environment is enabled to support multiple virtual machine (VM) clusters, then you can define up to 8 clusters and specify how the overall Exadata system resources are allocated to them.

In a configuration with multiple VM clusters, each VM cluster is allocated a portion of the overall Exadata system resources. By default, the resource allocation is dedicated to the VM cluster with no over-provisioning or resource sharing. On the compute nodes, a separate VM is defined for each VM cluster, and each VM is allocated a dedicated portion of the available compute node memory and local disk resources. Compute node CPU resources can be dedicated to each VM cluster, or you may choose to implement CPU oversubscription to increase the utilization of the compute node CPU resources. Each VM cluster is also allocated a dedicated portion of the overall Exadata storage.

Network isolation between VM clusters is also implemented. For each VM cluster, the client network and backup network each use a dedicated IP subnet.

VM cluster networks are isolated from each other and all other VM Clusters at layer 2 (Ethernet). Isolation is implemented within the Oracle Cloud at Customer network switches, which ensures that network traffic (including data) for each VM cluster is separated from other VM clusters. This additional capability provides enhanced security for databases running on VM Clusters. If you require a direct network connection between VM clusters, to enable database links between databases in different VM clusters for example, then you must submit a service request to Oracle to enable the connection.

Configuration of multiple VM clusters is only possible on environments where it is required when creating an Exadata Cloud at Customer instance. See Creating an Exadata Cloud at Customer Instance. If you are not prompted to configure a shape for the first VM cluster when you create an Exadata Cloud at Customer instance, then the Exadata system supporting the service instance is not equipped to support multiple VM clusters.

On Exadata Cloud at Customer environments where support for multiple VM clusters is enabled, you must specify the following attributes to configure the resources that are allocated to each VM cluster:

- **Cluster Name** — specifies the name that is used to identify the cluster.

- **Database backups on Exadata Storage** — this option configures the Exadata storage to enable local database backups on Exadata storage.
- **Create sparse disk group?** — this option creates a disk group that is based on sparse grid disks. You must select this option to enable Exadata Cloud at Customer snapshots. Exadata snapshots enable space-efficient clones of Oracle databases that can be created and destroyed very quickly and easily.
- **Database backups on ZDLRA** — this option enables database backups on Oracle Zero Data Loss Recovery Appliance (ZDLRA) storage. If you do not select this option then you will not be able to select ZDLRA as a backup location when you configure a database deployment on the cluster.
- **Exadata Storage (TB)** — specifies the total amount of Exadata storage (in TB) that is allocated to the VM cluster. This storage is allocated evenly from all of the Exadata Storage Servers. You must specify a value greater than 3 TB and up to the amount of remaining unallocated Exadata storage space.
- **CPU Cores per Node** — specifies the number of CPU cores that are allocated to each active node in the VM cluster. You must specify a value greater than 2 and up to the number of remaining unallocated CPU cores.
- **Memory (GB) per Node** — specifies the amount of memory (in GB) that is allocated to each active node in the VM cluster. You must specify a value greater than 30 GB and up to the amount of remaining unallocated memory, and you should factor in any plans for more VM clusters.

> **Note:**
>
> Take care when specifying the memory allocation because:
>
> – After the VM cluster is created, you cannot decrease the memory allocation; however, you may increase the memory allocation by using unallocated memory.
>
> – You cannot create another VM cluster unless there is 30 GB of remaining unallocated memory. In that case, you would need to delete an existing VM cluster before you can create another one.

- **Local Storage (GB)** — specifies the amount of local disk storage (in GB) that is allocated to each active node in the VM cluster. You must specify a value greater than 60 GB and up to the amount of remaining unallocated local storage space, and you should factor in any plans for more VM clusters.

> **Note:**
>
> Take care when specifying the local disk storage because:
>
> – In addition to the storage specified in this attribute, each VM cluster requires 137 GB of local disk storage to support software images for the VM cluster. Consequently, the minimum amount of local disk storage consumed by a VM cluster is 197 GB (137 GB + 60 GB).
>
> – For Exadata Cloud at Customer configurations based on Oracle Exadata X7 systems, the total amount of local disk storage that can be allocated to VM clusters is 1237 GB. For Exadata Cloud at Customer configurations based on Oracle Exadata X6 systems, the total amount of local disk storage that can be allocated to VM clusters is 483 GB by default, or up to 1237 GB on systems with upgraded local disk storage.
>
> – After the VM cluster is created, you cannot modify the amount of local storage.
>
> – If all of the local disk storage is allocated, or if there is not at least 197 GB of remaining unallocated local disk storage, then you cannot create another VM cluster. In that case, you must delete an existing VM cluster before you can create another one.

When you create a VM cluster, you must also specify the compute nodes that are part of the cluster. A cluster must contain at least one active node, and may contain any number of active nodes up to the capacity of the Exadata Cloud at Customer instance. See Creating a VM Cluster. You can also add compute nodes to an existing VM cluster or remove compute nodes from an existing VM cluster. See Modifying an Existing VM Cluster.

CPU Oversubscription

CPU oversubscription is a feature that works in conjunction with multiple virtual machine (VM) clusters to increase the overall utilization of your compute node CPU resources.

CPU oversubscription enables you to allocate more virtual compute node CPU cores to your VM clusters than the number of physical CPU cores that are enabled in the service instance. With CPU oversubscription enabled, the total number of virtual CPU cores that are available for allocation to the VM clusters is two times the number of enabled physical CPU cores, and the CPU core allocation for each individual VM cluster is limited to the number of enabled physical CPU cores.

When you provision an Exadata Cloud at Customer instance you can choose to enable CPU oversubscription. See Creating an Exadata Cloud at Customer Instance. You can also enable CPU oversubscription on an existing an Exadata Cloud at Customer instance. See Modifying the Compute Node Processing Power. However, note that you cannot disable CPU oversubscription after it is enabled.

By using CPU oversubscription, you can better utilize compute node CPU resources when some VM clusters are busy but others are not. However, CPU oversubscription forces physical CPU resources to be time-shared during busy periods.

For example, consider an Exadata Cloud at Customer instance with 24 CPU cores enabled:

- Without CPU oversubscription, you can only allocate up to 24 CPU cores to all of the VM clusters running on the instance.

Now consider if the Exadata Cloud at Customer instance supports 2 VM clusters, with each cluster allocated 12 CPU cores. In this case:

  - When one cluster is busy and the other cluster is idle, then the busy one can only use its allocation of 12 CPU cores while the remaining CPU resources remain idle.
  - When both clusters are busy, then they can use their dedicated CPU allocations with no impact on each other.

- With CPU oversubscription enabled, you can allocate up to 48 virtual CPU cores across all of the VM clusters running on the instance. In this case, each individual VM cluster is limited to 24 virtual CPU cores.

  Now consider if the Exadata Cloud at Customer instance supports 2 VM clusters, with each cluster allocated 24 virtual CPU cores. In this case:

  - If one cluster is busy and the other cluster is idle, then the busy one can benefit by using its virtual CPU allocation to use all of the 24 available physical CPU cores.
  - If both clusters are busy, then both must share the 24 physical CPU cores.

About Exadata Cloud at Customer Database Deployments

- Service Level
- Oracle Database Software Release
- Oracle Grid Infrastructure Software Release
- Oracle Database Software Edition
- Oracle Database Type
- Automatic Backup Configuration
- Data Security

Service Level

When creating a database deployment on Oracle Database Exadata Cloud at Customer, ensure that you select **Oracle Database Exadata Cloud Service** as the service level option.

Ignore other service level options, as these relate to Oracle Database Cloud Services that are implemented on non-Exadata systems.

> **Note:**
>
> Before you can create a database deployment with Exadata Cloud at Customer, you must have access to an Exadata Cloud at Customer instance. If you do not have access you will not see **Oracle Database Exadata Cloud Service** in the list of service level options.

Oracle Database Software Release

When creating a database deployment on Oracle Database Exadata Cloud at Customer, you choose one of the following Oracle Database software releases:

- **Oracle Database 11g Release 2**
- **Oracle Database 12c Release 1**
- **Oracle Database 12c Release 2**
- **Oracle Database 18c**
- **Oracle Database 19c**

Oracle Grid Infrastructure Software Release

Oracle Grid Infrastructure provides infrastructure services, such as cluster management and storage management, to Oracle Database. On Exadata Cloud at Customer, one installation of Oracle Grid Infrastructure supports all of the database deployments in each Exadata Cloud at Customer instance or VM cluster.

The Oracle Database software release version that you select for the starter database deployment determines the Oracle Grid Infrastructure software release version that is configured on your Exadata Cloud at Customer instance or VM cluster. The starter database is the very first database deployment that you create after the creation of your Exadata Cloud at Customer instance or VM cluster.

Your starter database configuration and the resulting Oracle Grid Infrastructure version is important because the Oracle Database software release cannot be newer than the underlying Oracle Grid Infrastructure version. For example, you cannot have a database using Oracle Database 19c running in conjunction with Oracle Grid Infrastructure 18c or earlier.

If you select the latest available Oracle Database software release version for your starter database, then the corresponding Oracle Grid Infrastructure version is installed. If you pick an earlier Oracle Database software release version for your starter database, then the second most recent Oracle Grid Infrastructure version is installed.

For example, for new starter database deployments on a system having Oracle Database 19c as the latest available Oracle Database software release version:

- If you select Oracle Database 19c as the Oracle Database software release version, then Oracle Grid Infrastructure 19c is installed.
- If you select Oracle Database 18c, or earlier, as the Oracle Database software release version, then Oracle Grid Infrastructure 18c is installed.

After Oracle Grid Infrastructure is installed:

- On systems configured with the latest Oracle Grid Infrastructure version, you can create database deployments that use any Oracle Database release version.
- On systems configured with an earlier Oracle Grid Infrastructure version, you can only create database deployments that use the corresponding Oracle Database version or earlier.

- If you want to use a later Oracle Database version on a system that is already configured with an earlier Oracle Grid Infrastructure version, then you must manually upgrade Oracle Grid Infrastructure.

  For details about upgrading to Oracle Grid Infrastructure 19c, see My Oracle Support note 2709296.1. For details about other Oracle Grid Infrastructure releases, contact Oracle Support.

Oracle Database Software Edition

When creating a database deployment on Oracle Database Exadata Cloud at Customer, **Enterprise Edition - Extreme Performance** is the only available choice. This provides all the features of Oracle Database Enterprise Edition, plus all the database enterprise management packs and all the Enterprise Edition options, such as Oracle Database In-Memory and Oracle Real Application Clusters (RAC).

Oracle Database Type

When creating a database deployment on Oracle Database Exadata Cloud at Customer, you choose one of the following database types:

- **Database Clustering with RAC** — creates a clustered database that uses Oracle Real Application Clusters. You can specify to run the clustered database instances on one or more compute nodes (database servers) in the Exadata Cloud at Customer environment.
- **Database Clustering with RAC and Data Guard Standby** — creates two clustered databases with one acting as the primary database and one acting as the standby database in an Oracle Data Guard configuration. Each database uses Oracle Real Application Clusters, with clustered database instances on one or more compute nodes (database servers).

Automatic Backup Configuration

Oracle Database Exadata Cloud at Customer provides automatic built-in database backup facilities. Automatic backups can be stored on:

- **Cloud storage** — uses an Oracle Storage Cloud container. This container becomes associated with Oracle Database Backup Cloud Service, which Exadata Cloud at Customer uses to perform backups to cloud storage.
- **Exadata storage** — uses storage from the local Exadata Storage Servers that is allocated to the RECO disk group. Database backups are managed in the Fast Recovery Area (FRA), which is located in the RECO disk group.
- **ZDLRA storage** — uses Oracle Zero Data Loss Recovery Appliance (ZDLRA).

When creating a database deployment on Exadata Cloud at Customer, you choose the destination for automatic backups. Your choices are:

- **Both Cloud Storage and Exadata Storage** — enables two separate backup sets containing periodic full (RMAN level 0) backups and daily incremental backups. The backup to cloud storage uses an Oracle Storage Cloud container, with a seven day cycle between full backups and an overall retention period of 30 days. The backup to Exadata storage uses space in the RECO disk group, with a seven day cycle between full backups and a seven day retention period.

> **Note:**
>
> This option is only available if you provisioned for database backups on Exadata storage. See Exadata Storage Configuration.

- **Cloud Storage Only** — uses an Oracle Storage Cloud container to store periodic full (RMAN level 0) backups and daily incremental backups, with a seven day cycle between full backups and an overall retention period of 30 days.

> **Note:**
>
> To eliminate possible confusion or contention, do not use a single cloud storage container as a backup destination for multiple database deployments.

- **ZDLRA Storage Only** — uses the Recovery Appliance to store one full (RMAN level 0) backup and daily incremental (RMAN level 1) backups. The Recovery Appliance creates virtual full backups from each daily incremental and validates those backups to ensure that they are always recoverable.

> **Note:**
>
> This option is only available if you configured your Exadata Cloud at Customer environment to enable database backups on ZDLRA. See Creating an Exadata Cloud at Customer Instance.

- **None** — no automatic backups are configured.

Data Security

In Oracle Database Exadata Cloud at Customer databases, data security is provided for data in transit and data at rest. Security of data in transit is achieved through network encryption. Security of data at rest is achieved through encryption of data stored in database data files and backups.

Data in Oracle Database files, including backups, is secured by the use of encryption implemented through a key management framework. Security of data across the network is provided by native Oracle Net Services encryption and integrity capabilities.

Topics

- Security of Data at Rest
- Security of Data in Transit

Security of Data at Rest

Oracle Database Exadata Cloud at Customer uses Oracle Transparent Data Encryption (TDE) to encrypt data in the database data files and in backups. Encrypted data is also protected in temporary tablespaces, undo segments, redo logs and during internal database operations such as JOIN and SORT.

TDE includes a keystore (referred to as a wallet in Oracle Database 11g and previous releases) to securely store master encryption keys, and a management framework to securely and efficiently manage the keystore and perform key maintenance operations.

TDE is the underlying mechanism used for default tablespace encryption and encrypted backups. It uses a two-tiered, key-based architecture to transparently encrypt and decrypt data. The master encryption key is stored in the software keystore. For tablespace encryption, this master encryption key is used to encrypt the tablespace encryption key, which in turn is used to encrypt and decrypt data in the tablespace. Refer to Tablespace Encryption for details on the implementation of tablespace encryption by default in Exadata Cloud at Customer.

When a database deployment is created on Exadata Cloud at Customer, a local auto-login software keystore is created. The keystore is local to the compute node and is protected by a system-generated password. The auto-login software keystore is automatically opened when accessed.

The keystore location is specified in the `ENCRYPTION_WALLET_LOCATION` parameter in the `$ORACLE_HOME/network/admin/`*`dbname`*`/sqlnet.ora` file, and can also be located in the database by querying `V$ENCRYPTION_WALLET`.

The Oracle keystore stores a history of retired TDE master encryption keys, which enables you to change them and still be able to decrypt data that was encrypted under an earlier TDE master encryption key.

For additional information on TDE and the keystore, refer to "Introduction to Transparent Data Encryption" in *Oracle Database Advanced Security Guide* for Release 18, 12.2 or 12.1 or "Securing Stored Data Using Transparent Data Encryption" in *Oracle Database Advanced Security Administrator’s Guide* for Release 11.2.

By default, backups to Cloud Storage for Enterprise Edition databases are encrypted. Recovery Manager (RMAN) performs transparent encryption using the auto-login software keystore. Refer to "Configuring Backup Encryption" in *Oracle Database Backup and Recovery User's Guide* for Release 18, 12.2 or 12.1 or "Encrypting RMAN Backups" in *Oracle Database Backup and Recovery User's Guide* for Release 11.2.

Security of Data in Transit

Oracle Database Exadata Cloud at Customer uses native Oracle Net Services encryption and integrity capabilities to secure connections to the database.

Refer to Using Network Encryption and Integrity for details on how to check your configuration and verify the use of native Oracle Net Services encryption and integrity.

Before You Begin with Exadata Cloud at Customer

Before you begin using Oracle Database Exadata Cloud at Customer, you should be familiar with the following technologies:

- Oracle Cloud

  See *Getting Started with Oracle Cloud*.
- Oracle Cloud Infrastructure Object Storage Classic

Exadata Cloud at Customer uses Oracle Database Backup Cloud Service to back up to cloud storage. Database Backup Cloud Service, in turn, uses Oracle Cloud Infrastructure Object Storage Classic containers as repositories for cloud backups. Before you can create a container, you must have access to Oracle Cloud Infrastructure Object Storage Classic. See About Oracle Cloud Infrastructure Object Storage Classic in *Using Oracle Cloud Infrastructure Object Storage Classic*.

Before you create an Exadata Cloud at Customer instance:

- (Optional) Create a Secure Shell (SSH) public/private key pair. The SSH keys are used to facilitate secure access to the compute nodes that support your database deployments. See Generate a Secure Shell (SSH) Public/Private Key Pair.
- (Optional) Create a cloud storage backup location in Oracle Cloud Infrastructure Object Storage Classic.

  If you want to automatically back up your database to cloud storage, you must associate it with a cloud storage container. See Creating Containers in *Using Oracle Cloud Infrastructure Object Storage Classic*.

About Exadata Cloud at Customer Roles and Users

In addition to the roles and privileges described in Learn About Cloud Account Roles in *Getting Started with Oracle Cloud*, there are additional roles containing privileges that specifically apply to Oracle Database Exadata Cloud at Customer.

Cloud user accounts with these roles must exist before users can access and use Exadata Cloud at Customer. The cloud administrator can create Exadata Cloud at Customer administrators by creating cloud user accounts and assigning them the desired roles.

For Exadata Cloud at Customer implementations where the Cloud Control Plane is provided by an Oracle Cloud Machine X6 system, the privileges for Exadata Cloud at Customer are managed using Oracle Identity Cloud Service (IDCS). Under IDCS, most privileges are managed through a series of application roles that are associated with the `ExadataCM` application.

The following list summarizes the application roles and privileges that are associated with the `ExadataCM` application:

- The `ExadataCM Service Based Entitlement Administrator` role enables a user to manage the Exadata Cloud at Customer instance. The following table summarizes key privileges that are associated with the `ExadataCM Service Based Entitlement Administrator` role:

Description of Privileges	More Information
Create, modify, or delete an Exadata Cloud at Customer instance.	Creating an Exadata Cloud at Customer Instance Scaling an Exadata Cloud at Customer Instance Suspending and Resuming an Exadata Cloud at Customer Instance Deleting an Exadata Cloud at Customer Instance
Create, modify, or delete an Exadata Cloud at Customer virtual machine (VM) cluster.	Administering VM Clusters

Description of Privileges	More Information
Monitor and manage service instance usage.	Viewing Service Details and Monitoring Service Usage in *Managing and Monitoring Oracle Cloud*

- The EXADATACM_ADMINISTRATOR role enables a user to manage database deployments that are associated with the Exadata Cloud at Customer instance. The following table summarizes key privileges that are associated with the EXADATACM_ADMINISTRATOR role:

Description of Privilege	More Information
Create or delete database deployments.	Creating a Database Deployment Deleting a Database Deployment
Patch, back up or restore database deployments.	Patching Exadata Cloud at Customer Backing Up and Restoring Databases on Exadata Cloud at Customer
Create and manage snapshots of a database deployment.	Creating and Managing Snapshots of a Database Deployment

In addition to the application roles that are associated with the ExadataCM application, each Exadata Cloud at Customer instance is associated with the ExadataCM_*Instance* application in IDCS, where *Instance* is the Exadata Cloud at Customer instance name. The ExadataCM_*Instance* application contains the ExadataCM Instance Administrator role, which enables a user to manage the Exadata Cloud at Customer instance by using REST APIs as described in Using the Oracle Cloud My Services REST APIs.

For information about managing users and role assignments under IDCS, see Add Users to a Cloud Account with Identity Cloud Service in *Getting Started with Oracle Cloud*.

For older Exadata Cloud at Customer implementations where the Cloud Control Plane is provided by an Oracle Cloud Machine X5 system, the privileges for Exadata Cloud at Customer are delegated using a similar set of roles that are managed by using the My Services application. For information about managing users and role assignments in this environment, see Add Users to a Traditional Cloud Account in *Getting Started with Oracle Cloud*.

Accessing the My Services Dashboard and the Oracle Database Cloud Service Console

To access the My Services dashboard on Oracle Database Exadata Cloud at Customer:

1. Open your web browser and go to URL that was provided by your tenant administrator.

   The Sign In page opens.

2. Sign in with your Exadata Cloud at Customer credentials.

   The My Services dashboard opens.

To access the Oracle Database Cloud Service console:

1. Go to the My Services dashboard.
2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and choose **Open Service Console**.

> **Note:**
>
> If the tile is not visible, click Customize Dashboard and use the resulting dialog to show the tile.

The Oracle Database Cloud Service console opens and displays the Instances Page, which contains a list of database deployments. If a Welcome page is displayed, click **Instances** next to Database Cloud Service to display the Instances Page.

Identifying the Exadata Cloud at Customer Release Version

To find the Exadata Cloud at Customer release version for your current implementation:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the user menu, located on the right side of the banner at the top of the My Services dashboard, and choose the **About** menu option.

   The About My Services dialog displays the Cloud Control Plane release version for your Exadata Cloud at Customer implementation.

   Based on the Cloud Control Plane release version, you can use the following table to determine the Exadata Cloud at Customer release version. For example, if the Cloud Control Plane version displayed in the dialog is 18.4.4, then the Exadata Cloud at Customer release version is 18.4.6, or a later maintenance release.

Cloud Control Plane Release Version	Exadata Cloud at Customer Release Version
17.2.2	17.2.3
18.1.4	18.1.4 or 18.1.4.2
18.3.2	18.1.4.4.3
18.4.4	18.4.6 or later

Using the Exadata Cloud at Customer REST APIs

You can programmatically provision and manage Oracle Database Exadata Cloud at Customer database deployments by using REST (REpresentational State Transfer) application programming interfaces (APIs).

Each REST API call maps to a HTTP request: getting an object (`GET`), adding an object (`POST`), updating an object (`PUT`), and deleting an object (`DELETE`). The HTTP response code indicates whether the request was successful. Each object for which you can perform the `GET`, `POST`, `PUT`, and `DELETE` requests is identified uniquely by its URI.

To access Exadata Cloud at Customer by using the REST API you must use the REST endpoint URL that is associated with your service instance. For details, see *REST API for Oracle Database Exadata Cloud at Customer*.

Using the Oracle Cloud My Services REST APIs

In addition to the Exadata Cloud at Customer REST APIs, you can also use the Oracle Cloud My Services REST APIs to perform the following functions on Exadata Cloud at Customer:

- Create and modify an Exadata Cloud at Customer instance.
- Scale the number of enabled CPU cores in the Exadata Cloud at Customer instance.

For details, see *Oracle Cloud My Services API*, *Managing Exadata Instances*, and *Using REST APIs to Enable and Disable CPU Cores on Exadata Cloud at Customer*.

Typical Workflow for Using Exadata Cloud at Customer

To start using Oracle Database Exadata Cloud at Customer, refer to the following tasks as a guide:

Task	Description	More Information
Add and manage users and roles	Create accounts for your users and assign them appropriate privileges. Assign the necessary Exadata Cloud at Customer roles.	Adding Users and Assigning Roles in *Getting Started with Oracle Cloud*, and About Exadata Cloud at Customer Roles and Users
Create an SSH key pair	Create SSH public/private key pairs to facilitate secure access to the compute nodes associated with your database deployments.	Generate a Secure Shell (SSH) Public/Private Key Pair
Create a service instance.	Use a wizard to create a new service instance, which provisions the Exadata Database Machine that hosts your database deployments.	Creating an Exadata Cloud at Customer Instance
Create a database deployment	Use a wizard to create a new database deployment.	Creating a Database Deployment
Enable network access	Permit access to network services associated with your database deployments.	About Network Access to Exadata Cloud at Customer
Load data into the database	Use standard Oracle Database tools to load data into your databases.	Loading Data into the Oracle Database on Exadata Cloud at Customer
Monitor database deployments	Check on the health and performance of individual database deployments.	Monitoring and Managing Oracle Database on Exadata Cloud at Customer
Monitor the service	Check on the day-to-day operation of your service, monitor performance, and review important notifications.	Managing and Monitoring Oracle Cloud Services in *Managing and Monitoring Oracle Cloud*
Patch a database deployment	Apply a patch or roll back a patch.	Patching Exadata Cloud at Customer
Back up a database deployment	Back up a database or restore a database from a backup.	Backing Up and Restoring Databases on Exadata Cloud at Customer

2
Managing the Exadata Cloud at Customer Life Cycle

This section describes tasks to manage the life cycle of Oracle Database Exadata Cloud at Customer.

Topics

- Creating an Exadata Cloud at Customer Instance
- Creating a Database Deployment
- Creating a Database Deployment Using a Cloud Backup
- Creating a Clone Database Deployment from a Snapshot Master
- Viewing All Database Deployments
- Viewing Detailed Information for a Database Deployment
- Viewing Activities for Database Deployments in an Identity Domain
- Stopping, Starting, and Removing Database Instances
- Stopping, Starting, and Restarting Compute Nodes
- Scaling an Exadata Cloud at Customer Instance
- Suspending and Resuming an Exadata Cloud at Customer Instance
- Creating and Managing Snapshots of a Database Deployment
- Deleting a Database Deployment
- Deleting an Exadata Cloud at Customer Instance

Creating an Exadata Cloud at Customer Instance

When you create an Oracle Database Exadata Cloud at Customer instance, you provision the Exadata Database Machine that hosts your Exadata Cloud at Customer database deployments. To create an Exadata Cloud at Customer instance, use the Create New Oracle Database Exadata Cloud Service Instance wizard as described in the following procedure.

Before You Begin

Before you create an Exadata Cloud at Customer instance, ensure that you have an active Exadata Cloud at Customer subscription in place.

If you do not have a valid subscription in place, then the Create New Oracle Database Exadata Cloud Service Instance wizard does not show the options required to create and provision an Exadata Cloud at Customer instance.

Procedure

To create an Exadata Cloud at Customer instance:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click **Create Instance**, and then click the **Create** button associated with Exadata Cloud at Customer in the All Services list.

   The Create New Oracle Database Exadata Cloud Service Instance wizard starts and the Instance Details page is displayed.

3. On the Instance Details page, specify configuration details for your Exadata Cloud at Customer instance. Then, click **Next**.

   a. In the **Instance Details** section, specify the following attributes associated with your Exadata Cloud at Customer instance.

      - **Name** — enter a name for your service instance.
      - **Region** — select the region (data center) that hosts your Exadata Cloud at Customer instance.
      - **Plan** — select the available plan from the list. A plan is associated with a set of attributes that apply to a service. For Exadata Cloud at Customer, only one plan is available.
      - **Rack Size** — select the rack configuration for your service instance. See Exadata System Configuration for a description of the available rack configurations. Your subscription may impose limits on the available rack sizes that are displayed.

        In addition to selecting the correct rack configuration, ensure that you select the appropriate configuration option for your situation:

        – If the service instance uses Exadata X6 hardware, ensure that you select Eighth Rack, Quarter Rack, Half Rack, or Full Rack.

        – If the service instance uses Exadata X7 hardware, ensure that you select a Rack Size that is associated with Exadata X7.

        – If the service instance uses Oracle Database licenses that are provided by you rather than licenses that are provided are part of the service subscription, ensure that you select a Rack Size that is associated with the bring-your-own-license (BYOL) option.

        – If the service instance uses Exadata infrastructure that is billed separately from the database server OCPUs, ensure that you select a Rack Size that is associated with Exadata X7 Infrastructure.

      - **BYOL enabled** — check this option to indicate that the Exadata Cloud at Customer instance uses Oracle Database licenses that are provided by you rather than licenses that are provided are part of the service subscription.

        This option does not display if it is incompatible with your subscription. It only affects the billing that is associated with the service instance, and has no effect on the technical configuration of the Exadata Cloud at Customer instance.

      - **Additional Number of OCPU (Cores)** — enter the number of additional CPU cores that you want to enable. (Optional)

        Ensure that you only use this field for service instances based on Exadata X6.

Use this field to specify the number of additional CPU cores to enable for the service instance. This number is in addition to the minimum number of enabled CPU cores for each rack size. The additional CPU cores specified in this setting are allocated evenly among the compute nodes associated with the Exadata Cloud at Customer instance.

See Exadata System Configuration for details about the maximum number of CPU cores that are available for each Exadata rack size. Your subscription may impose more limits on the number of CPU cores that you can enable.

This option does not display if it is incompatible with your subscription.

- **Additional Number of OCPU (Cores) X7** — enter the number of additional CPU cores that you want to enable. (Optional)

  Ensure that you only use this field for service instances based on Exadata X7 hardware without the BYOL or separate infrastructure options.

  This field is functionally equivalent to Additional Number of OCPU (Cores), except that it only applies to Exadata Cloud at Customer instances that are created with a Rack Size that is associated with an Exadata X7 system.

- **Additional Number of OCPU (Cores) BYOL** — enter the number of additional CPU cores that you want to enable. (Optional)

  Ensure that you only use this field for service instances based on Exadata X7 hardware with the bring-your-own-license (BYOL) option.

  This field is functionally equivalent to Additional Number of OCPU (Cores), except that it only applies to Exadata Cloud at Customer instances that are created with a Rack Size that is associated with the BYOL option.

- **Total Number of OCPU (Cores)** — enter the number of CPU cores that you want to enable.

  Ensure that you only use this field for service instances based on Exadata X7 Infrastructure where the Exadata infrastructure is billed separately from the database server OCPUs.

  Use this field to specify the total number of CPU cores to enable for the service instance. The CPU cores specified in this setting must be allocated evenly among the compute nodes associated with the Exadata Cloud at Customer instance.

  See Exadata System Configuration for details about the maximum number of CPU cores that are available for each Exadata rack size.

  Your subscription may also impose limits on the number of CPU cores that you can enable.

- **Suspend On Create** — check this option to suspend the Exadata Cloud at Customer instance immediately after it is created. When a service instance is suspended, the associated compute nodes are shut down and the service does not consume any compute node CPU cores.

  This option is only available for service instances that use Exadata X7 Infrastructure where the Exadata infrastructure is billed separately from the database server OCPUs.

- **CPU Oversubscription** — check this option to enable CPU oversubscription on the Exadata Cloud at Customer instance. See CPU Oversubscription.

> **Note:**
>
> After it is enabled, you cannot disable CPU oversubscription.

- **Availability Domain** — specifies the infrastructure zone to place the Exadata Cloud at Customer instance. (Optional)

  Use this setting to configure Exadata Cloud at Customer instances in different infrastructure zones to facilitate high availability in an Oracle Data Guard configuration.

  This option does not display if availability domains are not supported in the Region that you selected to host the instance.
- **Exadata ID** — specifies the identifier of the Exadata system rack to use for the Exadata Cloud at Customer instance. The choices in the list are based information provided during the provisioning of your Exadata Cloud at Customer systems.

b. In the **Administrator Details** section, provide information about the administrator of your Exadata Database Machine environment.
- **Email** — enter an email address for the Exadata system administrator.
- **User Name** — enter a user name for the Exadata system administrator. Alternatively, check the **Use email as user name** option to copy the Email entry into the User Name field.
- **First Name** — enter the first name of the Exadata system administrator.
- **Last Name** — enter the last name of the Exadata system administrator.

4. Use the VM Cluster Details section to specify configuration details for the first VM cluster on your Exadata Cloud at Customer instance. Then, click **Create Service Instance**.
- **Exadata System Name** — enter a name for your first VM cluster. This name is also used in the cluster name for the Oracle Grid Infrastructure installation.
- **Database backups on Exadata Storage** — check this option to configure the Exadata storage to enable local database backups.

  > **Note:**
  >
  > Take care when setting this option because your choice has a profound effect on the storage allocation and your backup options, which cannot be easily changed. See Exadata Storage Configuration for more information about the effects of each configuration alternative.
- **Create sparse disk group?** — check this option to create a disk group that is based on sparse grid disks. You must select this option to enable Exadata Cloud at Customer snapshots. Exadata snapshots enable space-efficient clones of Oracle databases that can be created and destroyed very quickly and easily.

> **Note:**
>
> Take care when setting this option because your choice has a profound effect on the storage allocation and your ability to use snapshots, which cannot be easily changed. See Exadata Storage Configuration for more information about the effects of each configuration alternative. See also Creating and Managing Snapshots of a Database Deployment.

- **Database backups on ZDLRA** — check this option to enable database backups on Oracle Zero Data Loss Recovery Appliance (ZDLRA) storage. If you do not select this option, then you cannot select ZDLRA as a backup location when you configure a database deployment.
- **Exadata Storage (TB)** — specifies the total amount of Exadata storage (in TB) that is allocated to the first VM cluster. This storage is allocated evenly from all of the Exadata Storage Servers. Specify a value up to the total amount of Exadata storage that is associated with your selected Rack Size, factoring in any plans for more VM clusters.
- **Client Network** — specifies the client network subnet that is allocated to the first VM cluster. Choose from the list of available subnets. The choices in the list are based on the network definitions that were defined during the provisioning of your Exadata Cloud at Customer system. Your choice also automatically selects the backup network subnet for the VM cluster, which is associated with the selected client network subnet.
- **OCPUs** — specifies the number of CPU cores that are allocated to each active node in the first VM cluster. Specify a value within the suggested range, factoring in any plans for more VM clusters.
- **Memory (GB)** — specifies the amount of memory (in GB) that is allocated to each active node in the first VM cluster. Specify a value greater than 30 GB and up to the amount of remaining unallocated memory, factoring in any plans for more VM clusters.

> **Note:**
>
> Take care when specifying the memory allocation because:
>
> – After the VM cluster is created, you cannot decrease the memory allocation; however, you may increase the memory allocation by using unallocated memory.
>
> – You cannot create another VM cluster unless there is 30 GB of remaining unallocated memory. In that case, you would need to delete an existing VM cluster before you can create another one.

- **DB Home Storage (GB)** — specifies the amount of local disk storage (in GB) that is allocated to each active node in the first VM cluster. Specify a value greater than 60 GB and up to the amount of remaining unallocated local storage space, factoring in any plans for more VM clusters.

> **Note:**
>
> Take care when specifying the local disk storage because:
>
> – In addition to the storage specified in this attribute, each VM cluster requires 137 GB of local disk storage to support software images for the VM cluster. Consequently, the minimum amount of local disk storage consumed by a VM cluster is 197 GB (137 GB + 60 GB).
>
> – For Exadata Cloud at Customer configurations based on Oracle Exadata X7 systems, the total amount of local disk storage that can be allocated to VM clusters is 1237 GB. For Exadata Cloud at Customer configurations based on Oracle Exadata X6 systems, the total amount of local disk storage that can be allocated to VM clusters is 483 GB by default, or up to 1237 GB on systems with upgraded local disk storage.
>
> – After the VM cluster is created, you cannot modify the amount of local storage.
>
> – If all of the local disk storage is allocated, or if there is not at least 197 GB of remaining unallocated local disk storage, then you cannot create another VM cluster. In that case, you must delete an existing VM cluster before you can create another one.

- The check boxes beside each node (`Node 1`, `Node 2`, and so on) enable you to specify the active compute nodes in the first VM cluster. Use the check boxes to specify the nodes that are included in the cluster.

5. Click **Create** in the confirmation dialog to proceed, or click **Cancel** in the confirmation dialog to step back into the wizard.

   Clicking **Create** in the confirmation dialog starts the process to create the service instance. This process is fully automated and takes approximately one to two hours to complete. During this time, you cannot access the service instance. After the process is completed, the service instance becomes active and you can create database deployments.

   If you need to change a setting, click **Cancel** in the confirmation dialog to step back into the wizard. You can also click **Cancel** at any time to exit the wizard without creating a new service instance.

Creating a Database Deployment

To create a database deployment on Oracle Database Exadata Cloud at Customer, use the Create Instance wizard as described in the following procedure.

However, before using the Create Instance wizard, you need to make sure that you have all of the necessary information, as described in Before You Begin. Additionally, after your database deployment is created you need to perform a few follow-on tasks to make sure your deployment is accessible and up-to-date, as described in After Your Database Deployment Is Created.

Before You Begin

Before you create a database deployment, ensure you have created or acquired information about the following:

- An active Exadata Cloud at Customer instance

  Before you can create a database deployment, you must have an active Exadata Cloud at Customer instance in place.

  If you do not have an active service instance in place, then the Create Instance wizard will not show the options required to create a database deployment on Exadata Cloud at Customer.

  See Creating an Exadata Cloud at Customer Instance.

- An SSH public/private key pair (Optional)

  An SSH public key is used for authentication when you use an SSH client to connect to a compute node associated with the deployment. When you connect, you must provide the private key that matches the public key.

  You can have the wizard create a public/private key pair for you, or you can create one beforehand and upload or paste its private key value. If you want to create a key pair beforehand, you can use a standard SSH key generation tool. See Generate a Secure Shell (SSH) Public/Private Key Pair.

  When creating a database deployment on Exadata Cloud at Customer, the Create Instance wizard checks if an SSH public key is already registered on the Exadata system. If no key exists, you will be prompted for a new public key during the creation process. Otherwise, the existing key is used.

- A cloud storage backup location (Optional)

  If you want to automatically back up your database to cloud storage, you must associate an Oracle Cloud Infrastructure Object Storage Classic container with the database deployment. You can create the container beforehand and provide the wizard with information about it, or you can have the wizard create the container for you. If you want to create the container beforehand, see Creating Containers in *Using Oracle Cloud Infrastructure Object Storage Classic* for instructions.

  Whether you create the container beforehand or have the wizard do it for you, you are prompted for the following information about the container:

  - The name of the container.
  - The user name and password of a user who has read/write access to the container.

  > **Note:**
  >
  > To eliminate possible confusion or contention, do not use a single cloud storage container as a backup destination for multiple database deployments.

- A ZDLRA backup location (Optional)

  If you want to automatically back up your database to an Oracle Zero Data Loss Recovery Appliance (ZDLRA), you must associate a ZDLRA backup location with the database deployment.

  Before you create the database deployment you must configure your recovery appliance. See Configuring Recovery Appliance for Protected Database Access.

  Then, when you select ZDLRA as a backup location in the Create Instance wizard, you are prompted for the following information:

  - The TNS connect string to connect to the ZDLRA.
  - The ZDLRA virtual private catalog (VPC) user name and password.

- An existing cloud backup of an Oracle database created using Oracle Database Backup Cloud Service, which meets the criteria for instantiation from backup (Optional)

  When you create a database deployment, you can have the database populated, or instantiated, from the data stored in a Database Backup Cloud Service backup. To use this approach the following criteria must be met:

  - The backed-up database must be version 18, 12.2.0.1, 12.1.0.2 or 11.2.0.4 with the latest patch set update (PSU) applied.
  - If the backed-up database uses Oracle Database version 12.1.0.2, or later, it must be a multitenant container database (CDB). Exadata Cloud at Customer does not support non-CDB databases for Oracle Database 12c, or later.
  - The backed-up database must use File System or ASM as its storage method for data files.

  If you wish to instantiate your database using a backup from another Exadata Cloud at Customer database deployment in the same identity domain, then you must specify the source database deployment by selecting from a list of the available deployments.

  If you wish to instantiate your database using any other Database Backup Cloud Service backup, you are prompted for the following information:

  - The database ID of the backed-up database.
  - The decryption method for the backup, which is the password associated with the backup for backups that use password encryption, or a zip file containing the source database's wallet directory and contents for backups that use Transparent Data Encryption (TDE).
  - The name of the Oracle Cloud Infrastructure Object Storage Classic container where the backup is stored.
  - The user name and password of an Oracle Cloud user who has read access to the container.

- If you intend to create a database deployment with an Oracle Data Guard configuration, ensure that you have the required network configuration in place to support Oracle Data Guard.

  See Using Oracle Data Guard in Exadata Cloud at Customer.

- If you intend for the database deployment to use an existing set of Oracle binaries in an existing Oracle Home directory location, ensure that you know the Oracle Home name.

  See Viewing Information About Oracle Homes.

- If you intend for the database deployment to use an existing set of Oracle binaries in an existing Oracle Home directory location, ensure that the software version in the existing Oracle Home directory location is not older than the current default software image. If the current default software image is newer than the software version in the existing Oracle Home directory location, then the database creation may fail or experience problems.

To avoid this situation, you can use any of the following approaches:

  - Before you create the new database deployment, patch the existing Oracle Home directory location to bring it up to the release level of the default software image.
  - Before you create the new database deployment, activate the software image that matches the release level of the software in the existing Oracle Home directory location. Then, after creating the database deployment, re-activate the newer software image, if required.
  - Create the new database deployment in a new Oracle Home directory location, which will use the default software image.

See Administering Software Images.

- If your Exadata Cloud at Customer system environment is configured to support multiple VM clusters, ensure that you know the name of the VM cluster that will host the database deployment.

  See Viewing Information About Existing VM Clusters.

Procedure

To create a database deployment on Exadata Cloud at Customer:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click **Create Instance**.

   The Create Instance wizard starts.

3. On the Instance page, specify basic attributes for your database deployment. Then, click **Next**.

   - **Instance Name** — enter a name for your database deployment.
   - **Description** — enter a description for your database deployment. (Optional)
   - **Notification Email** — enter an email address that receives notifications from the database deployment creation operation. (Optional)
   - **Exadata System** — select an available Oracle Exadata Database Machine configuration to host the database deployment. The list contains the Oracle Exadata Database Machines that are associated with your active Exadata Cloud at Customer instances.

     If you later select Database Clustering with RAC and Data Guard Standby as the Database Type, the Exadata System specifies the system hosting the primary database.
   - **Cluster** — select a VM cluster to host the database deployment.

     This option is only available if your system environment is enabled to support multiple VM clusters.
   - **Hostnames** — specify one or more compute nodes that you want to host the database instances for this database deployment.

     If you previously selected Database Clustering with RAC and Data Guard Standby as the Database Type, then this selection applies to the primary database.

- **Tags** — specifies tags for the database deployment. (Optional)

  Tagging enables you to group database deployments that share similar characteristics or are used for a similar purpose. Click the plus icon to create a new tag.

- **Service Level** — select **Oracle Database Exadata Cloud Service** from the list.

  Ignore other service level options, as these relate to Oracle Database Cloud Services that are implemented on non-Exadata systems.

  > **Note:**
  >
  > If **Oracle Database Exadata Cloud Service** is not available in the list of service level choices, you do not have active Exadata Cloud at Customer instance. You need to obtain a subscription and create an Exadata Cloud at Customer instance before you can create a database deployment.

- **Software Release** — select the Oracle Database software release that you want to run in your database deployment.

  Your choices for software release are:

  - **Oracle Database 11g Release 2**
  - **Oracle Database 12c Release 1**
  - **Oracle Database 12c Release 2**
  - **Oracle Database 18c**
  - **Oracle Database 19c**

  > **Note:**
  >
  > - The Oracle Database software release version that you select for the starter database deployment determines the Oracle Grid Infrastructure software release version that is configured on your Exadata Cloud at Customer instance or VM cluster. The starter database is the very first database deployment that you create after the creation of your Exadata Cloud at Customer instance or VM cluster.
  > - For non-starter database deployments, your software release options may be limited by the Oracle Grid Infrastructure software release version that is configured on your Exadata Cloud at Customer instance or VM cluster. If you select an option that is incompatible with your Oracle Grid Infrastructure software installation, then the deployment will fail and an error message will be returned.
  >
  > See Oracle Grid Infrastructure Software Release.

- **Software Edition** — the only valid option for use with Exadata Cloud at Customer is **Enterprise Edition — Extreme Performance**.

- **Database Type** — select one of the following options:
  - **Database Clustering with RAC** — creates a clustered database that uses Oracle Real Application Clusters. You can specify to run the clustered database instances on one or more compute nodes (database servers) in the Exadata Cloud at Customer environment.
  - **Database Clustering with RAC and Data Guard Standby** — creates two clustered databases with one acting as the primary database and one acting as the standby database in an Oracle Data Guard configuration. Each database uses Oracle Real Application Clusters, with clustered database instances on one or more compute nodes (database servers).

4. On the Instance Details page, configure details for your database deployment. Then, click **Next**.

   a. In the **Database Configuration** section, set the database name, administrator password, and other database configuration options.

      - **DB Name** — enter a name for the database instances.
      - **PDB Name** — enter a name for the default pluggable database (PDB).

        This setting is available only for databases that use Oracle Database 12c, or later.

        > **Note:**
        >
        > This setting is ignored if you select the "Create Instance from Existing Backup" option. In this case, the new database contains the PDBs contained in the backup.

      - **Administration Password** and **Confirm Password** — enter and then re-enter an administration password.

        The administration password is used to configure administration accounts and functions in the database deployment, including the password for the Oracle Database SYS and SYSTEM users.

        > **Note:**
        >
        > Ensure that you remember the administration password associated with your database deployment.

      - **Oracle Homes** — specify the option to create a new Oracle Home directory location, or select an existing Oracle Home location from the list.
      - **Oracle Home Name** — if you previously selected the option to create a new Oracle Home directory location, you can optionally specify a name prefix for the new Oracle Home location. If specified, the value becomes the first part of the full Oracle Home name, which also includes a string identifying the Oracle Database release and latest applied bundle patch, along with numeric identifiers that are used to uniquely identify the Oracle Home location. If you do not specify a value, then the new Oracle Home location is given a system-generated name.

- **SSH Public Key** — provide the SSH public key to be used for authentication when using an SSH client to connect to a compute node that is associated with your database deployment.

  Click **Edit** to specify the key by using one of the following options:

  - Upload a file containing the public key value.
  - Input, or paste in a public key value. Ensure that the value you input does not contain line breaks or end with a line break.
  - Create a new system-generated key pair. If you select this option, you will be prompted to download a file containing the system-generated keys. Ensure that you keep the generated private key in a secure location.

  > **Note:**
  >
  > The SSH Public Key field is not displayed if the selected Exadata Cloud at Customer environment already contains a previously specified SSH key.

- Optionally, expand **Advanced Settings** and set the following:

  - **Application Type** — select the application type that best suits your application:

    * **Transactional (OLTP)** — configures the database for a transactional workload, with a bias towards high volumes of random data access.
    * **Decision Support or Data Warehouse** — configures the database for a decision support or data warehouse workload, with a bias towards large data scanning operations.

    > **Note:**
    >
    > The Application Type setting is only displayed when you create the starter database, which is the very first database deployment that you create after the creation of your Exadata Cloud at Customer instance or VM cluster. Subsequent database deployments are created with a standardized database configuration.

  - **Character Set** — specify the database character set for the database. The database character set is used for:

    * Data stored in SQL `CHAR` data types (`CHAR`, `VARCHAR2`, `CLOB`, and `LONG`).
    * Identifiers such as table names, column names, and PL/SQL variables.
    * Entering and storing SQL and PL/SQL source code.

> **Note:**
>
> The Character Set setting is ignored if you select the "Create Instance from Existing Backup" option. In this case, the new database inherits the character set from the backup.

– **National Character Set** — specify the national character set for the database. The national character set is used for data stored in SQL `NCHAR` data types (`NCHAR`, `NCLOB`, and `NVARCHAR2`).

> **Note:**
>
> The National Character Set setting is ignored if you select the "Create Instance from Existing Backup" option. In this case, the new database inherits the national character set from the backup.

– **Enable Oracle GoldenGate** — configures the database for use as the replication database of an Oracle GoldenGate Cloud Service instance. See Using Oracle GoldenGate Cloud Service with Exadata Cloud at Customer.

b. In the **Backup and Recovery Configuration** section, choose an automatic backup option and associated backup settings for your database deployment.

Backup Destination — select how automatic backups are to be configured:

- **Both Cloud Storage and Exadata Storage** — enables two separate backup sets containing periodic full (RMAN level 0) backups and daily incremental backups. The backup to cloud storage uses an Oracle Storage Cloud container, with a seven day cycle between full backups and an overall retention period of 30 days. The backup to Exadata storage uses space in the RECO disk group, with a seven day cycle between full backups and a seven day retention period.

> **Note:**
>
> This option is only available if you provisioned for database backups on Exadata storage. See Exadata Storage Configuration.

- **Cloud Storage Only** — uses an Oracle Storage Cloud container to store periodic full (RMAN level 0) backups and daily incremental backups, with a seven day cycle between full backups and an overall retention period of 30 days.

> **Note:**
>
> To eliminate possible confusion or contention, do not use a single cloud storage container as a backup destination for multiple database deployments.

- **ZDLRA Storage Only** — uses the Recovery Appliance to store one full (RMAN level 0) backup and daily incremental (RMAN level 1) backups. The Recovery Appliance creates virtual full backups from each daily incremental and validates those backups to ensure that they are always recoverable.

> **Note:**
>
> This option is only available if you configured your Exadata Cloud at Customer environment to enable database backups on ZDLRA. See Creating an Exadata Cloud at Customer Instance.

- **None** — no automatic backups are configured.

If you select **Both Cloud Storage and Exadata Storage** or **Cloud Storage Only**, the following fields and options are displayed:

- **Cloud Storage Container** — enter the URL of an Oracle Cloud Infrastructure Object Storage Classic container. To eliminate possible confusion or contention, a separate cloud storage container is recommended for each database deployment.

  The URL has the general form:

  ```
  storage-instance-endpoint/container
  ```

  where `storage-instance-endpoint` is the REST endpoint URL for the storage service instance, and `container` is the name of the storage container.

  To determine the `storage-instance-endpoint` value, see Finding the REST Endpoint URL for Your Service Instance in *Using Oracle Cloud Infrastructure Object Storage Classic*.

- **Username** — enter the user name of a user who has read/write access to the container specified in **Cloud Storage Container**.
- **Password** — enter the password of the user specified in **Username**.
- **Create Cloud Storage Container** — select this option to create a new storage container. To use this option you must specify a new Cloud Storage container using the previously specified format. You must also provide the Cloud Storage user name and password in the preceding fields, and the specified user must have the Service Administrator role for the specified Oracle Storage Cloud Service instance.

  > **Note:**
  >
  > If you select this option, the new storage container is created as soon as you click **Next** on the Instance Details page, and the storage container remains even if you cancel out of the wizard without creating a new database deployment. If this occurs, you can use the storage container for a future database deployment or you can manually delete the container. If you want to delete the container, see Deleting Containers in *Using Oracle Cloud Infrastructure Object Storage Classic* for instructions.

If you select **ZDLRA Storage Only**, the following fields and options are displayed:

- **ZDLRA Storage Connection String** — enter the TNS connect string to connect to the Recovery Appliance.

- **Username** — enter the virtual private catalog (VPC) user name for the Recovery Appliance specified in **Zdlra Storage Container**.
- **Password** — enter the password of the user specified in **Username**.

c. Complete the **Initialize Data From Backup** section if you are having the new database populated, or instantiated, from the data stored in a Database Backup Cloud Service backup.

- **Create Instance from Existing Backup** — select Yes to populate the new database with data stored in an existing Database Backup Cloud Service backup.
- **On-Premises Backup?** — use this option to indicate the origin of the source database backup. The origin of the backup may be another Exadata Cloud at Customer database deployment in the same identity domain or another database that was backed up to cloud storage using Database Backup Cloud Service.

  If you select this option you are indicating that the source database backup is not from another currently operational Exadata Cloud at Customer database deployment in the same identity domain. In this case, the following fields and options are displayed:

  - **Database ID** — enter the database id of the database from which the existing backup was created. You can get this value by querying the backup source database as follows:

    ```
    SQL> SELECT dbid FROM v$database;
    ```

  - **Decryption Method** — click **Edit** and provide the information necessary to decrypt the existing backup:

    * For a backup that uses Transparent Data Encryption (TDE), select **Upload Wallet File** then click **Browse** and specify a zip file containing the source database's TDE wallet directory, and the contents of that directory.

      > **Note:**
      >
      > If the source database is from another Exadata Cloud at Customer database deployment, its TDE wallet directory is `/u02/app/oracle/admin/`*`dbname`*`/tde_wallet` or `/var/opt/oracle/dbaas_acfs/`*`dbname`*`/tde_wallet`.

    * For a backup that uses password encryption, select **Paste RMAN Key Value** and paste the password (key value) used to encrypt the backup.

    > **Note:**
    >
    > For database deployments using Oracle Database 12.2, or later, only backups using TDE are supported.

  - **Cloud Storage Container** — enter the name of the Oracle Cloud Infrastructure Object Storage Classic container where the existing backup is stored; use this format:

    `instance-id_domain/container`

where *instance* is the name of the Oracle Cloud Infrastructure Object Storage Classic instance, *id_domain* is the id of the identity domain, and *container* is the name of the container.

- **Username** — enter the user name of an Oracle Cloud user who has read access to the container specified in **Cloud Storage Container**.
- **Password** — enter the password of the user specified in **Username**.

If you deselect **On-Premises Backup?** you are indicating that the source database backup is from another currently operational Exadata Cloud at Customer database deployment in the same identity domain. In this case, the following field is displayed:

- **Source Instance Name** — specify the database deployment that is associated with the source database backup that you want to use.
- **Backup Tag** — specify the backup tag that identifies the specific source database backup that you want to use.

d. Complete the **Standby Database** section if you previously selected Database Clustering with RAC and Data Guard Standby as the Database Type.

- **Standby Database Configuration** — influences the location of the Oracle Data Guard standby database. Select from the following options:
  - **High Availability** — indicates that the standby database is placed on a different Exadata system in the same region (data center) as the primary database, thus providing isolation at the Exadata system infrastructure level.
  - **Disaster Recovery** — indicates that the standby database is placed in a different region (data center) from the primary database, thus providing isolation at the Exadata system infrastructure level and geographical separation to protect against catastrophic data center failure.
- **Exadata System** — select an available Oracle Exadata Database Machine configuration to host the standby database. The list contains the Oracle Exadata Database Machines that are associated with your active Exadata Cloud at Customer instances.

  Your selection is validated when you leave the Instance Details page, and you will be notified if the selection is not consistent with your Standby Database Configuration specification.

  > **Note:**
  >
  > The Exadata System used to host the standby database must exist in the same identity domain as the Exadata System previously specified on the Instance page that is used to host the primary database.

- **Cluster** — select a VM cluster to host the standby database.

  This option is only available if your Exadata Cloud at Customer system environment is enabled to configure multiple VM clusters.
- **Hostnames** — specify one or more compute nodes that you want to host the database instances for the standby database.

> **Note:**
>
> The number of compute nodes that you specify here must match the number of compute nodes that you specified for the primary database.

5. On the Confirmation page, review the configuration settings. If you are satisfied, click **Create**.

   If you need to change a setting, use the navigation bar or **Back** button at the top of the wizard to step back through the pages in the wizard. Click **Cancel** to cancel out of the wizard without creating a new database deployment.

   Clicking **Create** starts the process to create the database deployment. This process is fully automated and takes some time to complete. You should not access or manipulate the database deployment until the creation process is completed and the deployment is listed in the Oracle Database Cloud Service console.

After Your Database Deployment Is Created

After your database deployment is created, you should perform the following actions:

- **Update cloud tooling**

  While the base images used to create Exadata Cloud at Customer database deployments are updated regularly, it is possible that even more recent updates to the cloud tooling are available. Therefore, you should check for and apply any updates to the cloud tooling. See Updating the Cloud Tooling on Exadata Cloud at Customer.

- **Apply database patches**

  While the base images used to create Exadata Cloud at Customer database deployments are updated regularly, it is possible that a newer patch set update (PSU) or bundle patch (BP) is available. Therefore, you should check for and apply any database patches that are available. See Applying a Patch.

Creating a Database Deployment Using a Cloud Backup

You can create an Oracle Database Exadata Cloud at Customer database deployment whose database is instantiated from a cloud backup created using Oracle Database Backup Cloud Service.

This technique is called instantiate-from-backup and the database that is the origin of the backup is called the source database. Instantiate-from-backup can be used in the following ways:

- You can use the Create Instance wizard to perform an instantiate-from-backup operation during the creation of a new database deployment. See Creating a Database Deployment.
- Alternatively, you can use the instantiate-from-backup function to replace the database associated with an existing database deployment. See Replacing the Database by Using the Oracle Database Cloud Service Console and Replacing the Database by Using `ibkup` Actions.

In any case, the source database backup must meet certain suitability requirements. These include:

- If the source database is from an existing database deployment, ensure that the database deployment has been backed up to cloud storage. For more information, see About Backing Up Database Deployments on Exadata Cloud at Customer.
- If the source database is an on-premises Oracle database, ensure that the database is suitable for instantiation in the cloud and then create a cloud backup. For instructions, see Creating a Cloud Backup of an On-Premises Database.
- The source database backup must use Oracle Database version 18, 12.2.0.1, 12.1.0.2, or 11.2.0.4 with the latest patch set update (PSU) applied.
- If the source database uses Oracle Database version 12.1.0.2, or later, it must be a multitenant container database (CDB). Exadata Cloud at Customer does not support non-CDB databases for Oracle Database 12c, or later.
- The source database uses File System or ASM as its storage method for data files.

After completing an instantiate-from-backup operation, the resulting database deployment exhibits the following characteristics:

- The database uses the SID that you specified when creating the database deployment.
- The database files and data are from the source database backup.
- The database identifier (`dbid` value in `V$DATABASE`) will be different from the source database identifier.
- The Oracle Net listener is configured with services for the database, and PDBs if applicable.

Creating a Cloud Backup of an On-Premises Database

Use the `ibackup` utility to create a backup of an on-premises Oracle Database, which can then be used to replace an Oracle Database Exadata Cloud at Customer database.

The `ibackup` utility enables you to:

- Perform a pre-check of the on-premises database to ensure that you can generate a backup that is suitable for replacing a cloud database.
- Generate an Oracle Database backup, as well as additional files, that you can use to replace the database on an Exadata Cloud at Customer database deployment as part of an instantiate-from-backup operation.

Prerequisites

Ensure that the on-premises database you intend to back up, as well as the Exadata Cloud at Customer database you intend to replace, meet the requirements described in Creating a Database Deployment Using a Cloud Backup.

The source on-premises database must also meet the following additional criteria:

- The on-premises database host must be a Linux X64 (OEL 6 or OEL 7) system.
- The database character set of your on-premises database must be compatible with the Exadata Cloud at Customer database that you intend to replace.
- Non-Oracle software on the on-premises database host must meet the following minimum release requirements:

- Java: Release 7 or higher. Java must be in the default path.
- Python: Above Release 2.6 and below Release 3.0.

Procedure

Perform these tasks:

1. Download a zip file containing the `ibackup` utility to the on-premises database host. Use `wget` on the on-premises database host to download the `OracleCloud_ibackup_Setup.zip` file from Oracle Cloud Infrastructure Object Storage Classic:

   ```
   $ wget https://storage.us2.oraclecloud.com/v1/dbcsswlibp-usoracle29538/
   ibackup/OracleCloud_ibackup_Setup.zip
   ```

2. On the on-premises database host:
   1. Log in as the `oracle` user.
   2. Unzip the `OracleCloud_ibackup_Setup.zip` file. Files are extracted into the `ibackup` directory.
   3. Switch to the `root` user and run the following command to set the ownership of the files in the `ibackup` directory:

      ```
      # chown -R oracle:oinstall ibackup
      ```

   4. Return to being the `oracle` user and navigate to the `ibackup` directory:

      ```
      $ cd ibackup
      ```

   5. Edit the `backup.cfg` file as follows:
      - Set the encryption mode for the database backup. Set `TDE=y` if the database uses Transparent Data Encryption. Set `TDE=n` to use RMAN key encryption.
      - Set the value for `target_db` to `18.0.0`, `12.2.0.1`, `12.1.0.2`, or `11.2.0.4`, depending on the version of the Exadata Cloud at Customer database deployment where you intend to instantiate the backup.
      - Set the value for `oss_user` to the user name of a user who has read/write access to the storage container specified in `oss_url`.
      - Set the value for `oss_url` to the URL of the Oracle Cloud Infrastructure Object Storage Classic container that will be used to store the database backup.
      - You can set the value for `oss_url` to the password of the user specified in `oss_user`. If you specify a value for `oss_passwd`, the password is obfuscated the first time you run the `ibackup` tool. If you do not enter a password value, you are prompted for the password when you run the tool.
      - If you set `TDE=n`, set the `rman_key` value to the RMAN encryption key. Otherwise, leave this value blank.

3. Run a pre-check on the source on-premises database. The pre-check does not generate a backup file.

   ```
   $ ./ibackup -d dbname
   ```

In the above command, *dbname* is the name of the source database. Examine the pre-check results.

4. Generate a backup:

   ```
   $ ./ibackup -d dbname -b -i
   ```

   Optionally, you can use the –f option to ignore fix-up log failures when generating a backup:

   ```
   $ ./ibackup -d dbname -b -i -f
   ```

   In addition to the Oracle Database backup, the following files are also generated in the /var/opt/oracle/ibackup/ibkup directory:

   - tde_wallet.zip — The TDE wallet directory. This file is generated only if TDE was enabled in the on-premises database. Copy this file to a secure and accessible location. This file is required to import the Oracle backup in an instantiate-from-backup operation.
   - TDE_README.txt — Instructions on how to unzip the tde_wallet.zip file. This is important because the instantiate-from-backup operation expects a defined structure for the TDE wallet directory.
   - Import.json — Template file to import the backup using ibkup actions with the dbaasapi utility.
   - oss_file.cfg — Oracle Cloud Infrastructure Object Storage Classic information used to save the backup.

   Use these files when replacing the database on an Exadata Cloud at Customer database deployment as part of an instantiate-from-backup operation.

Replacing the Database by Using the Oracle Database Cloud Service Console

You can use the Oracle Database Cloud Service console to replace the database for an Exadata Cloud at Customer database deployment using an instantiate-from-backup operation.

Before You Begin

If you wish to replace your database using a backup from another currently operational Exadata Cloud at Customer database deployment in the same identity domain, then you must specify the source database deployment by selecting from a list of the available deployments.

If you wish to replace your database using any other backup, you are prompted for the following information:

- The database ID of the backed-up database.
- The decryption method for the backup, which is the password associated with the backup for backups that use password encryption, or a zip file containing the source database’s wallet directory and contents for backups that use Transparent Data Encryption (TDE).

- The name of the Oracle Cloud Infrastructure Object Storage Classic container where the backup is stored.
- The user name and password of an Oracle Cloud user who has read access to the container.

Procedure

1. Open the Instances page of the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the database deployment whose database you wish to replace.

   The Oracle Database Cloud Service Overview page is displayed.

3. From the action menu (☰) next to the database deployment name, choose **Replace Database using Backup**.

   The Replace Database using Backup window is displayed.

4. Specify attributes in the Replace Database using Backup window:

   On-Premises Backup? — use this option to indicate the origin of the source database backup.

   If you select this option you are indicating that the source database backup is not from another currently operational Exadata Cloud at Customer database deployment in the same identify domain. In this case, the following fields and options are displayed:

   - **Database ID** — enter the database id of the database from which the existing backup was created. You can get this value by querying the backup source database as follows:

     ```
     SQL> SELECT dbid FROM v$database;
     ```

   - **Decryption Method** — provide the information necessary to decrypt the existing backup:

     - For a backup that uses Transparent Database Encryption (TDE), select **Upload Wallet File** then click **Browse** and specify a zip file containing the source database's TDE wallet directory, and the contents of that directory.

       > **Note:**
       >
       > If the source database is from another Exadata Cloud at Customer database deployment, its TDE wallet directory is `/u02/app/oracle/admin/`*`dbname`*`/tde_wallet` or `/var/opt/oracle/dbaas_acfs/`*`dbname`*`/tde_wallet`.

     - For a backup that uses password encryption, select **Paste RMAN Key Value** and paste the password (key value) used to encrypt the backup.

     > **Note:**
     >
     > For database deployments using Oracle Database 12c Release 2 (12.2), or later, only backups using TDE are supported.

- **Cloud Storage Container** — enter the name of the Oracle Cloud Infrastructure Object Storage Classic container where the existing backup is stored; use this format:

  ```
  instance-id_domain/container
  ```

  where *`instance`* is the name of the Oracle Cloud Infrastructure Object Storage Classic instance, *`id_domain`* is the id of the identity domain, and *`container`* is the name of the container.
- **Username** — enter the user name of an Oracle Cloud user who has read access to the container specified in **Cloud Storage Container**.
- **Password** — enter the password of the user specified in **Username**.
- **Administration Password** and **Confirm Password** — enter and then re-enter a new administration password.

  The administration password is used to configure administration accounts and functions in the database deployment, including the password for the Oracle Database SYS and SYSTEM users in the newly replaced database.

  > **Note:**
  >
  > Ensure that you remember the administration password associated with your database deployment.

If you deselect **On-Premises Backup?** you are indicating that the source database backup is from another currently operational Exadata Cloud at Customer database deployment in the same identity domain. In this case, the following fields are displayed:

- **Source Instance Name** — specify the database deployment whose database backup you want to use.
- **Backup Tag** — a list of backups available for the specified database deployment.
- **Administration Password** and **Confirm Password** — enter and then re-enter a password for the Oracle Database SYS and SYSTEM users in the newly replaced database.

5. Click **Replace Database** and confirm that you want to replace the database when prompted.

   The database deployment is put into Maintenance status and the operation begins. The process is fully automated and takes some time to complete. You should not access or manipulate the database deployment until the process is completed.

Replacing the Database by Using `ibkup` Actions

You can perform an instantiate-from-backup operation by using `ibkup` actions with the `dbaasapi` utility to replace the existing database in an Exadata Cloud at Customer database deployment with a database sourced from a cloud backup.

The `dbaasapi` utility operates by reading a JSON file containing instructions and other information and writing its results to a JSON file specified in the input file. In essence, it is a command-line utility that operates like a REST API endpoint, accepting a JSON request body and producing a JSON response body. The `dbaasapi` utility checks that the operation being requested does not conflict with any operation already in progress and then runs the operation asynchronously: that is, it starts the requested operation and then returns terminal control to you.

Here are the tasks you perform to replace the database by using `ibkup` actions:

1. Copy the TDE wallet from the source database to the Exadata Cloud at Customer deployment, if necessary.
2. Create `dbaasapi` input files for `ibkup begin` and `ibkup status` actions.
3. Run the `ibkup begin` action.
4. Run the `ibkup status` action to monitor progress of the `ibkup` operation.
5. Upon completion of the `ibkup` operation, confirm that the source database now resides on the Exadata Cloud at Customer deployment.

Copy the Source Database TDE Wallet

If the cloud backup you are using was created using Transparent Data Encryption (TDE) or dual-mode encryption, you need to copy the TDE wallet from the source database to the database deployment.

> **Note:**
>
> If the source database is from another Exadata Cloud at Customer database deployment, its backup was created using Transparent Data Encryption (TDE). All cloud backups from Exadata Cloud at Customer use TDE as the backup encryption mode.

1. On an Exadata Cloud at Customer compute node that is associated with your target database deployment, connect as the `oracle` user and create a directory to store the source database TDE wallet along with other files that you create in later steps. For example:

   ```
   $ mkdir -p /home/oracle/ibkup
   ```

2. Copy the source database's `tde_wallet` directory to the newly created directory.

   If the source database is from another Exadata Cloud at Customer database deployment, its `tde_wallet` directory is located at `/u02/app/oracle/admin/`*`dbname`*`/tde_wallet` or `/var/opt/oracle/dbaas_acfs/`*`dbname`*`/tde_wallet`, where *`dbname`* is the name of the database. You can find the location of the `tde_wallet` directory by querying `V$ENCRYPTION_WALLET`.

3. Ensure that the `tde_wallet` files are owned by `oracle` and only accessible by `oracle`. For example:

   ```
   $ chown -R oracle:oinstall /home/oracle/ibkup/tde_wallet
   $ chmod 600 /home/oracle/ibkup/tde_wallet/*
   ```

Create `dbaasapi` Input Files

1. Use a secure shell utility like SSH or PuTTY to connect as the `opc` user to the compute node that is associated with your target database deployment. For instructions, see Connecting to a Compute Node Through Secure Shell (SSH).
2. The `dbaasapi` utility must be run as the `root` user. Start a root-user shell:

   ```
   $ sudo -s
   #
   ```

3. Navigate to the directory where you previously stored the source database TDE wallet.

   ```
   # cd /home/oracle/ibkup
   ```

   If you did not copy the source database TDE wallet, create a directory to store your request and response files and then navigate to it.
4. Create a `begin-request.json` file to pass to `dbaasapi` to perform the `ibkup begin` action.

   Here is an example that uses a password-encrypted backup:

   ```
   # cat begin-request.json
   {
     "object": "db",
     "action": "begin",
     "operation": "ibkup",
     "params": {
       "dbname": "crmdb",
       "dbid": "1428538966",
       "oss_url": "https://mystore.storage.oraclecloud.com/v1/Storage-mystore/
   IBKUP",
       "oss_user": "storageadmin",
       "oss_passwd": "pa55word",
       "decrypt_key": "backup",
       "passwd": "Welcome-1",
       "dbsize": "30GB"
     },
     "outputfile": "/home/oracle/ibkup/begin-response.json",
     "FLAGS": ""
   }
   ```

   The JSON object for the `ibkup begin` action supports the following parameters. All parameters are required unless identified as optional.

Parameter	Description
`object`	The value `"db"`.
`action`	The value `"begin"`.
`operation`	The value `"ibkup"`.

Parameter	Description

`params`	An object containing parameters that provide details for the `ibkup begin` action. This object has the following parameters:
	• `dbname`: The name of the target database that you are replacing. You can get this value by querying the target database:
	`SQL> SELECT name FROM v$database;`
	• `dbid`: The database id of the source database. You can get this value by querying the source database:
	`SQL> SELECT dbid FROM v$database;`
	• `oss_url`: The URL of the container that stores the source database's backup.
	• `oss_user`: The user name of an Oracle Cloud user who has read privileges for the container that stores the source database's backup.
	• `oss_passwd`: The password of the `oss_user` user.
	• `rman_handle`: (Optional) The RMAN handle of a targeted backup that contains controlfile and spfile backups. The `ibkup begin` action uses the controlfile and spfile in this backup.
	Use the `rman_tag` parameter to specify the RMAN tag of a backup supported by this controlfile and spfile. If you do not specify an RMAN tag, the latest backup supported by this controlfile and spfile is used.
	Oracle recommends that you provide both a handle and a tag to use a specific backup or provide no handle and no tag to use the latest backup.
	You can view RMAN handles and tags by using the RMAN `LIST BACKUP` command.
	• `rman_tag`: (Optional) The RMAN tag of a targeted full backup. The `ibkup begin` action uses this backup.
	Use the `rman_handle` parameter to specify the RMAN handle of a backup containing controlfile and spfile backups that support this RMAN tag. If you do not specify an RMAN handle, the latest controlfile and spfile are used. If the RMAN handle does not support the specified RMAN tag, a `datafile not found` error occurs.
	Oracle recommends that you provide both a handle and a tag to use a specific backup or provide no handle and no tag to use the latest backup.
	You can view RMAN handles and tags by using the RMAN `LIST BACKUP` command.
	• `decrypt_key`: (Optional) The key (password) used to encrypt the backup.
	Provide this parameter if you created the backup using password encryption or dual-mode encryption.
	Note: you cannot use this option when replacing the database on a database deployment using Oracle Database 12c Release 2 (12.2) or later, because only backups using TDE are supported for such deployments.
	• `decrypt_wallet`: (Optional) The fully qualified path of the wallet directory you copied from the source database

Parameter	Description
	to the DBCS deployment you created; for example: /home/oracle/ibkup/tde_wallet. Provide this parameter if you created the backup using Transparent Data Encryption (TDE) or dual-mode encryption. • passwd: The administrator (SYS and SYSTEM) password to use for the target database after the replacement operation concludes. • dbsize: The size of the source database. For Exadata Cloud at Customer, provide an estimate of the source database size.
outputfile	The fully qualified name of the output file for dbaasapi to use; for example: "/home/oracle/ibkup/begin-response.json".
FLAGS	The value "" (an empty string).

5. Create a status-request.json file to pass to dbaasapi to perform the ibkup status action. Here is an example:

```
# vim status-request.json
{
  "object": "db",
  "action": "status",
  "operation": "ibkup",
  "id": "TBD",
  "params": {
    "dbname": "crmdb"
  },
  "outputfile": "/home/oracle/ibkup/status-response.json",
  "FLAGS": ""
}
```

In this example, the value of the id parameter is "TBD" because the corresponding ibkup begin action has not been run yet.

The JSON object for the ibkup status action supports the following parameters. All parameters are required.

Parameter	Description
object	The value "db".
action	The value "status".
operation	The value "ibkup".
id	The ID number of the action you want status for.
params	An object containing parameters that provide details for the ibkup status action. This object has the following parameters: • dbname: The name of the database on the target database that is being replaced.
outputfile	The fully qualified name of the output file for dbaasapi to use; for example: "/home/oracle/ibkup/status-response.json".

Parameter	Description
FLAGS	The value "" (an empty string).

Run the ibkup begin Action

1. Use dbaasapi to run the ibkup begin action:

   ```
   # /var/opt/oracle/dbaasapi/dbaasapi -i begin-request.json
   ```

2. View the output file to confirm that the action has started and note the id of the action; for example:

   ```
   # cat /home/oracle/ibkup/begin-response.json
   {
      "msg" : "",
      "object" : "db",
      "status" : "Starting",
      "errmsg" : "",
      "outputfile" : "",
      "action" : "begin",
      "id" : "19",
      "operation" : "ibkup",
      "logfile" : "/var/opt/oracle/log/crmdb/dbaasapi/db/ibkup/19.log"
   }
   ```

   The key parameters in this response are as follows:

Parameter	Description
status	The status of the operation; one of: "Error", "Starting", "InProgress" or "Success".
action	The value "begin", which is the ibkup action you requested.
id	The ID number assigned to this action. Use this number in subsequent ibkup status actions to check the status of the overall ibkup operation.
operation	The value "ibkup", which is the operation you requested.
logfile	The log file for the ibkup operation. You can poll this log file to monitor progress of the operation. However, you should run the ibkup status action to monitor progress because it provides extra status information along with a definitive indication of when the operation is finished.

Run the ibkup status Action to Monitor Progress

1. Update the status-request.json input file with id value of the ibkup operation that you have started. Edit the status-request.json file, replacing the id parameter value of "TBD" with the ID number reported in the begin-response.json file.

2. Use dbaasapi to run the ibkup status action and view the response; for example:

   ```
   # /var/opt/oracle/dbaasapi/dbaasapi -i status-request.json
   # cat status-response.json
   {
      "msg" : "  -> 15 03 * * 6 oracle /var/opt/oracle/cleandb/cleandblogs.pl\\n\
   \n#### Completed OCDE Successfully ####",
      "object" : "db",
      "status" : "Success",
   ```

```
    "errmsg" : "",
    "outputfile" : "",
    "action" : "begin",
    "id" : "19",
    "operation" : "ibkup",
    "logfile" : "/var/opt/oracle/log/crmdb/dbaasapi/db/ibkup/19.log"
}
```

3. Rerun the `ibkup status` action regularly until the response indicates that the operation is finished.

Confirm Successful Completion

Confirm successful completion of the instantiate-from-backup operation by connecting to the replacement database and verifying that it contains the expected structure and data. For example, you could query `V$PDBS` to ensure that the database contains the expected PDBs, or you could query a specific application table to ensure that it contains the expected data. Also ensure that the expected database instances are up and running.

1. Connect as the **`oracle`** user to a compute node that is associated with your target database deployment.

   See Connecting to a Compute Node Through Secure Shell (SSH).

2. Configure the Oracle Database environment variable settings:

   ```
   $ . oraenv
   ```

3. Ensure that of the expected database instances are running:

   ```
   $ srvctl status database -d dbname
   ```

4. Connect to the replacement database and confirm that it contains the expected structure and data.

   For example, you could query `V$PDBS` to ensure that the database contains the expected PDBs:

   ```
   $ sqlplus / as sysdba
   SQL> select name, open_mode, restricted from v$pdbs;
   ```

5. Check that services registered with the Oracle Net Listener include the services from the source database:

   ```
   $ lsnrctl status
   ```

More About `ibkup` Actions

The preceding instantiate-from-backup tasks showed the use of two `ibkup` actions; `begin` and `status`. Here is more information about what these two actions do, along with information about two other `ibkup` actions; `prereqs` and `restore`.

- The `begin` action:
  1. Validates the format and completeness of the input file.
  2. Creates the output file, which includes an ID number for use in subsequent `status` actions.
  3. Releases terminal control.
  4. Performs the same value-validation checks that the `prereqs` action performs.

    5. Takes a backup of the current database deployment environment, should the need arise to restore the environment after a failed `ibkup` operation.
    6. Replaces the current database using the backup of the source database.

- The `status` action:
    1. Validates the format and completeness of the input file.
    2. Retrieves the current status of operation whose ID number was provided in the input file.
    3. Creates the output file, which contains the retrieved status information.
- The `prereqs` action takes an input file of the same format as the `begin` and `restore` actions, except that the value of the `action` parameter must be `"prereqs"`.

  You can use the `prereqs` action to test whether the input file you intend to use for either the `begin` action or the `restore` action is valid and that the backup specified in the file is available.

  The `prereqs` action does as follows:
    1. Validates the format and completeness of the input file.
    2. Creates the output file, which includes an ID number for use in subsequent `status` actions.
    3. Releases terminal control.
    4. Checks that the values provided in the input file would be valid if used in the input file for a `begin` or `restore` action. It confirms access to the backup, including use of the decryption key and wallet as necessary, and that the backup's database ID matches the provided `dbid`.
- The `restore` action takes an input file of the same format as the `begin` action, except that the value of the `action` parameter must be `"restore"`.

  If a `begin` operation fails, you can use the `restore` action to reset the database deployment's environment so that you can attempt the `begin` operation again, after determining the cause of the failure and correcting the problem.

  After you use the `restore` action, you need to reboot the compute nodes that are associated with the database deployment to ensure that the environment is completely reset. For instructions, see Stopping, Starting, and Restarting Compute Nodes.

  The `restore` action does as follows:
    1. Validates the format and completeness of the input file.
    2. Creates the output file, which includes an ID number for use in subsequent `status` actions.
    3. Releases terminal control.
    4. Terminates any `begin` action that is in progress.
    5. Kills all processes related to the `begin` action. If it cannot kill one or more processes, it exits with an error status.
    6. Restores the database deployment environment to its state before the first `begin` action.

Creating a Clone Database Deployment from a Snapshot Master

You can create a clone database deployment that is based on a snapshot master database.

> **Note:**
>
> You must create a snapshot master before you can create a clone. See Creating a Snapshot Master.

When you create a clone deployment, Exadata Cloud at Customer creates a new database deployment that uses sparse data files based on the snapshot master database.

As data blocks change, the changed blocks are written to sparse files maintained in the SPARSE disk group. Thus the data files are a combination of the original data blocks in the snapshot master and the changed blocks in the sparse files. This mechanism enables multiple clones to share the snapshot master data files while changes are written to separate sparse data files for each clone. This is especially space-efficient when much of the data in the clone remains unchanged from the original values.

Procedure

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Locate the snapshot master you want to use as the basis for your clone deployment and choose **Create Database Clone** from the snapshot master's action menu (☰).

   The Instance page of the Create Instance wizard is displayed.

3. On the Instance page, specify basic attributes for your clone database deployment. Then, click **Next**.

   For a clone deployment, only the following fields may be set. The other fields contain values that are inherited from the snapshot master database deployment and cannot be changed.

   - **Instance Name** — enter a name for your database deployment.
   - **Description** — enter a description for your database deployment. (Optional)
   - **Notification Email** — enter an email address that receives notifications from the database deployment creation operation. (Optional)
   - **Hostnames** — specify one or more compute nodes that you want to host the database instances for this clone database deployment.
   - **Tags** — specifies tags for the database deployment. (Optional)

Tagging enables you to group database deployments that share similar characteristics or are used for a similar purpose. Click the plus icon to create a new tag.

4. On the Instance Details page, configure details for your clone database deployment. Then, click **Next**.

   For a clone deployment, only the following fields may be set. The other fields contain values that are inherited from the snapshot master database deployment and cannot be changed.

   a. In the **Database Configuration** section, set the following options:

      - **Cluster** — select a VM cluster to host the database deployment.

        This option is only available if your system environment is enabled to support multiple VM clusters.

      - **DB Name** — enter a name for the database instances.

      - **Administration Password** and **Confirm Password** — enter and then re-enter an administration password.

        The administration password is used to configure administration accounts and functions in the database deployment, including the password for the Oracle Database SYS and SYSTEM users.

        > **Note:**
        >
        > Ensure that you remember the administration password associated with your database deployment.

   b. In the **Backup and Recovery Configuration** section, choose an automatic backup option and associated backup settings for your clone database deployment.

      Backup Destination — select how automatic backups are to be configured:

      - **Both Cloud Storage and Exadata Storage** — enables two separate backup sets containing periodic full (RMAN level 0) backups and daily incremental backups. The backup to cloud storage uses an Oracle Storage Cloud container, with a seven day cycle between full backups and an overall retention period of 30 days. The backup to Exadata storage uses space in the RECO disk group, with a seven day cycle between full backups and a seven day retention period.

        > **Note:**
        >
        > This option is only available if you provisioned for database backups on Exadata storage. See Exadata Storage Configuration.

      - **Cloud Storage Only** — uses an Oracle Storage Cloud container to store periodic full (RMAN level 0) backups and daily incremental backups, with a seven day cycle between full backups and an overall retention period of 30 days.

> **Note:**
>
> To eliminate possible confusion or contention, do not use a single cloud storage container as a backup destination for multiple database deployments.

- **ZDLRA Storage Only** — uses the Recovery Appliance to store one full (RMAN level 0) backup and daily incremental (RMAN level 1) backups. The Recovery Appliance creates virtual full backups from each daily incremental and validates those backups to ensure that they are always recoverable.

  > **Note:**
  >
  > This option is only available if you configured your Exadata Cloud at Customer environment to enable database backups on ZDLRA. See Creating an Exadata Cloud at Customer Instance.

- **None** — no automatic backups are configured.

If you select **Both Cloud Storage and Exadata Storage** or **Cloud Storage Only**, the following fields and options are displayed:

- **Cloud Storage Container** — enter the URL of an Oracle Cloud Infrastructure Object Storage Classic container. To eliminate possible confusion or contention, a separate cloud storage container is recommended for each database deployment.

  The URL has the general form:

  ```
  storage-instance-endpoint/container
  ```

  where *`storage-instance-endpoint`* is the REST endpoint URL for the storage service instance, and *`container`* is the name of the storage container.

  To determine the *`storage-instance-endpoint`* value, see Finding the REST Endpoint URL for Your Service Instance in *Using Oracle Cloud Infrastructure Object Storage Classic*.

- **Username** — enter the user name of a user who has read/write access to the container specified in **Cloud Storage Container**.
- **Password** — enter the password of the user specified in **Username**.
- **Create Cloud Storage Container** — select this option to create a new storage container. To use this option you must specify a new Cloud Storage container using the previously specified format. You must also provide the Cloud Storage user name and password in the preceding fields, and the specified user must have the Service Administrator role for the specified Oracle Storage Cloud Service instance.

> **Note:**
>
> If you select this option, the new storage container is created as soon as you click **Next** on the Instance Details page, and the storage container remains even if you cancel out of the wizard without creating a new database deployment. If this occurs, you can use the storage container for a future database deployment or you can manually delete the container. If you want to delete the container, see Deleting Containers in *Using Oracle Cloud Infrastructure Object Storage Classic* for instructions.

If you select **ZDLRA Storage Only**, the following fields and options are displayed:

- **ZDLRA Storage Connection String** — enter the TNS connect string to connect to the Recovery Appliance.
- **Username** — enter the virtual private catalog (VPC) user name for the Recovery Appliance specified in **Zdlra Storage Container**.
- **Password** — enter the password of the user specified in **Username**.

5. On the Confirmation page, review the configuration settings. If you are satisfied, click **Create**.

   If you need to change a setting, use the navigation bar or **Back** button at the top of the wizard to step back through the pages in the wizard. Click **Cancel** to cancel out of the wizard without creating a new clone database deployment.

   Clicking **Create** starts the process to create the clone database deployment. This process is fully automated and takes some time to complete. You should not access or manipulate the clone deployment until the creation process is completed and the clone is listed in the Oracle Database Cloud Service console.

Viewing All Database Deployments

From the Oracle Database Cloud Service Console, you can:

- View the total resources allocated across all Oracle Database Exadata Cloud at Customer database deployments.
- View the details for each deployment.
- Use the search field to filter the list to include only the deployments that contain a given string in their name.

To view all database deployments:

- Open the Oracle Database Cloud Service console.

  For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

  The Oracle Database Cloud Service console opens and displays the Instances Page, which contains a list of database deployments.

> **Note:**
>
> If a Welcome page is displayed, click **Services** next to Database Cloud Service to display the Instances Page.

Viewing Detailed Information for a Database Deployment

From the Oracle Database Cloud Service Overview page, you can:

- View a summary of details for a database deployment on Oracle Database Exadata Cloud at Customer, such as description, subscription mode, and so on.
- View the total resources allocated to the deployment.
- View the details and status information for each node associated with the deployment.

To view detailed information for a database deployment:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click on the name of the database deployment for which you want to view more information.

   The Oracle Database Cloud Service Overview Page is displayed.

Viewing Activities for Database Deployments in an Identity Domain

Use the Activity page to view activities for database deployments on Oracle Database Exadata Cloud at Customer in your identity domain. You can restrict the list of activities displayed using search filters.

To view activities for your database deployments:

1. Open the Activity page:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. Click **Activity**.

      The Activity Page is displayed, showing the list of all activities started within the past 24 hours. You can use the Start Time Range field to specify a start time range other than the default of the previous 24 hours.

2. Use the options in the Search Activity Log section to filter the results to meet your needs. You can search on start time range, full or partial service name, activity status, and operation type. Click **Search**. View the results in the table that follows.

Stopping, Starting, and Removing Database Instances

From the Oracle Database Cloud Service console, you can stop or start a database instance on a specific compute node. You can also remove a database instance from a compute node.

Topics

- Stopping a Database Instance
- Starting a Database Instance
- Removing a Database Instance

Stopping a Database Instance

You can stop individual database instances associated with an Exadata Cloud at Customer database deployment from the Oracle Database Cloud Service console. When you stop a database instance, the database instance is shut down but remains as a cluster resource that is associated with the compute node.

To stop a database instance:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. In the list, click the name of a database deployment that is associated with the database instance that you want to stop.

   The Oracle Database Cloud Service Overview page is displayed. The page contains the list of compute node resources, with a separate action menu (☰) for each entry.

3. Locate the compute node that hosts the database instance that you want to stop. From the associated action menu, select **Stop DB Instance** and then confirm the action.

   While the database instance is being stopped, the database deployment is placed into a maintenance state. After the database instance is stopped, the associated compute node shows an icon containing a red alert triangle to indicate that the instance is stopped (⚠).

Starting a Database Instance

You can start individual database instances associated with an Exadata Cloud at Customer database deployment from the Oracle Database Cloud Service console. You can start a previously stopped database instance that is already associated with a compute node. Or, you can also start a database instance on a node that is not associated with the database deployment, which is displayed in the Oracle Database Cloud Service console with a shaded appearance.

Before you start a database instance, particularly on a new node, you should check to ensure that there is sufficient memory available to support the database instance. You can examine a running database instance on another node to determine its memory use and then check the node where you want to start the database instance.

To start a database instance:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. In the list, click the name of a database deployment for which you want to start a database instance.

   The Oracle Database Cloud Service Overview page is displayed. The page contains the list of compute node resources, with a separate action menu (☰) for each entry.

3. Locate the compute node that you want to start the database instance on. From the associated action menu, select **Start DB Instance** and then confirm the action.

   While the database instance is being started, the database deployment is placed into a maintenance state.

Removing a Database Instance

When you remove a database instance, the database instance is removed as a clustered database resource. If the database instance is running, it is automatically shut down before it is removed.

> **Note:**
>
> - You cannot remove a database instance if it is the only preferred instance for an Oracle Database service.
> - You cannot remove a database instance while a backup operation is running on the corresponding compute node.

To remove a database instance:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. In the list, click the name of a database deployment that is associated with the database instance that you want to remove.

   The Oracle Database Cloud Service Overview page is displayed. The page contains the list of compute node resources, with a separate action menu (☰) for each entry.

3. Locate the compute node that contains the database instance that you want to remove. From the associated action menu, select **Remove DB Instance** and then confirm the action.

   While the database instance is being removed, the database deployment is placed into a maintenance state. After the operation is completed, the database instance is removed from the database definition in Oracle Clusterware, and the associated compute node is displayed with a shaded appearance in the Oracle Database Cloud Service console.

Stopping, Starting, and Restarting Compute Nodes

From the Oracle Database Cloud Service console, you can stop, start, and restart the compute node virtual machines (VMs) that are associated with a database deployment on Oracle Database Exadata Cloud at Customer.

> **Note:**
>
> It is also possible to stop and start a compute node by connecting to the compute node and using an operating system command, such as `shutdown` or `reboot`. However, Oracle recommends that you use the Oracle Database Cloud Service console to stop and start the compute nodes, rather than using an operating system command.

Topics

- Stopping a Compute Node
- Starting a Stopped Compute Node
- Restarting a Compute Node
- Viewing Past Stop, Start and Restart Activity

Stopping a Compute Node

You can stop individual compute node VMs associated with an Exadata Cloud at Customer database deployment from the Oracle Database Cloud Service console. When you stop a compute node, the node is not available to any of your Exadata Cloud at Customer databases that share the compute node. If you stop all of the compute nodes associated with an Exadata Cloud at Customer environment, you effectively stop all of your databases running on the environment.

To stop a compute node:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. In the list, click the name of a database deployment that is associated with the compute node that you want to stop.

   The Oracle Database Cloud Service Overview page is displayed. The page contains the list of compute node resources, with a separate action menu (☰) for each entry.

3. From the action menu that is associated with the compute node you want to stop, select **Stop Compute Node**, and then confirm the action.

   While the compute node is being stopped, the database deployment is placed into a maintenance state. After the compute node is stopped, the associated icon shows a red alert triangle to indicate that the node is stopped (⚠).

> ⚠ **Caution:**
>
> Do not use the `halt`, `shutdown`, or `shutdown -h` commands to shut down a compute node. Doing so stops the compute node indefinitely and requires manual intervention by Oracle Cloud system administrators to restart the compute node.

Starting a Stopped Compute Node

To start a stopped compute node VM:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. In the list, click the name of a database deployment that is associated with the compute node that you want to start.

   The Oracle Database Cloud Service Overview page is displayed. The page contains the list of compute node resources, with a separate action menu (☰) for each entry.

3. From the action menu that is associated with the compute node that you want to start, select **Start Compute Node**, and then confirm the action.

   While the compute node is being started, the database deployment is placed into a maintenance state.

Restarting a Compute Node

When you restart a compute node VM, the compute node is stopped and then immediately started again.

To restart a compute node:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. In the list, click the name of a database deployment that is associated with the compute node that you want to restart.

   The Oracle Database Cloud Service Overview page is displayed. The page contains the list of compute node resources, with a separate action menu (☰) for each entry.

3. From the action menu that is associated with the compute node you want to restart, select **Restart Compute Node**, and then confirm the action.

   While the compute node is being restarted, the database deployment is placed into a maintenance state.

Viewing Past Stop, Start and Restart Activity

You can see information about past stop, start, and restart activity for an Exadata Cloud at Customer database deployment by viewing the activity log:

1. View the Overview page for the database deployment:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. In the list of deployments, click the name of the database deployment whose past activity you want to view.

      The Oracle Database Cloud Service Overview page is displayed.

2. To expand the activity log, click the triangle icon beside the Activity title.

   The activity log shows information about past operations performed on the database deployment, with the most recent activity first.

3. To see details about an operation, click the triangle icon beside the operation.

   If an operation failed, the details include information about why it failed.

Scaling an Exadata Cloud at Customer Instance

You can scale an Oracle Database Exadata Cloud at Customer instance by:

- Modifying the Compute Node Processing Power
- Adding Exadata Storage
- Adding Compute Nodes

Modifying the Compute Node Processing Power

If an Exadata Cloud at Customer instance requires more compute node processing power, you can scale up the number of enabled CPU cores in the current Exadata system configuration. A capability known as bursting enables you to modify the compute node processing power temporarily, or you can add compute node processing power on a more permanent basis.

The maximum number of enabled CPU cores depends on your system configuration. See Exadata System Configuration. However, your subscription may impose extra limits.

You can also choose to enable the CPU oversubscription feature.

To modify the number of enabled CPU cores or enable the CPU oversubscription feature in an existing Exadata Cloud at Customer instance:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Modify**.

   The Modify Oracle Database Exadata Cloud Service Instance wizard starts and the Instance Details page is displayed.

4. If it is displayed on the Instance Details page, select the **OCPUs** option to indicate that you want to modify the number of enabled CPU cores within an existing Exadata Cloud at Customer instance.

5. On the Instance Details page, specify the type of scaling operation that you want to perform and use the slider control to set the number of enabled CPU cores on each compute node. You can also choose to enable CPU oversubscription. Then, click **Modify**.

   a. If it is not already enabled, you can choose to enable CPU oversubscription on the Exadata Cloud at Customer instance by selecting the **CPU Oversubscription** option. See CPU Oversubscription.

      > **Note:**
      >
      > After it is enabled, you cannot disable CPU oversubscription.

   b. Specify the type of scaling operation that you want to perform by selecting the **Subscription** option or the **Burst** option:

      - Select **Subscription** if you want to scale the service in line with a subscription change.

        To use this option, you must first adjust your subscription and purchase the additional CPU core entitlements. Thereafter, the slider control enables the placement of the additional CPU cores on your compute nodes.

      - Select **Burst** if you want to temporarily scale the service instance.

        With bursting, you can quickly scale up beyond your subscription level to cater for workload peaks. You can also scale back to the subscription level at any time.

        CPU cores beyond your subscription level are charged separately using an hourly rate for the bursting period. These charges begin immediately after you enable the CPU cores in your service instance, and continue until you scale back to the subscription level.

        > **Note:**
        >
        > For non-metered subscriptions only, the maximum number of enabled CPU cores available with bursting is limited to twice the number of CPU cores in the associated service subscription. For example, if your service subscription contains 11 enabled CPU cores on each compute node, then the bursting maximum is 22 CPU cores on each compute node. This limit does not apply to other subscription types, such as Universal Credits.

   c. Use the slider control to set the new number of enabled CPU cores on each compute node. When you make a change, the change is reflected in the **Configuration after Update** summary. At any point, you can click **Reset** to return the slider to its original setting.

> **Note:**
>
> The slider setting represents the total number of enabled CPU cores for each compute node and not the number of additional CPU cores to enable.

6. On the Confirmation page, review the configuration settings. If you are satisfied, click **Modify**.

   Otherwise, click **Cancel** on the Confirmation page and again on the Instance Details page to cancel the wizard without updating the service instance.

Modifying the number of enabled CPU cores and enabling CPU oversubscription are online operations. Compute nodes are not rebooted because of these operations.

If the Exadata Cloud at Customer instance is not enabled to support multiple virtual machine (VM) clusters, then changes to the number of enabled CPU cores are automatically reflected in each compute node user domain (DomU). However, if the instance is enabled to support multiple VM clusters, then changes to the number of enabled CPU cores in the an Exadata Cloud at Customer instance are not automatically applied to the underlying VM clusters. In this case, you must also modify the VM clusters separately to utilize the enabled CPU resources in the an Exadata Cloud at Customer instance. See Modifying an Existing VM Cluster.

The Oracle Database instance caging feature is enabled by explicitly setting the `CPU_COUNT` database initialization parameter. That setting is not affected by modifying the number of enabled CPU cores. Therefore, if you are using the instance caging feature you must explicitly alter the `CPU_COUNT` setting to match any changes to the service instance. If `CPU_COUNT` is set to `0` (its default setting), then instance caging is disabled and Oracle Database continuously monitors the number of CPUs reported by the operating system and uses the current count.

Adding Exadata Storage

You can increase the Exadata storage in your current Exadata system configuration by adding Exadata Storage Servers subject to the following conditions:

- Each Exadata Cloud at Customer system configuration can have an absolute maximum of 12 Exadata Storage Servers.
- The Exadata Cloud at Customer system configuration must be based on Oracle Exadata X7 hardware or Oracle Exadata X6 hardware.
- The Exadata Cloud at Customer service instance must be enabled to support multiple virtual machine (VM) clusters.

To add Exadata Storage Servers to your current Exadata system configuration, you must first procure subscriptions for the additional storage servers. Then, an Oracle field engineer physically installs the Exadata Storage Server hardware and registers it in the system.

After the Exadata Storage Server hardware is physically provisioned and registered, you can use the following procedure to add the Exadata storage to your Exadata Cloud at Customer instance:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Modify**.

   The Modify Oracle Database Exadata Cloud Service Instance wizard starts and the Instance Details page is displayed.

4. On the Instance Details page, select the **Storage** option to indicate that you want to add Exadata storage to the Exadata Cloud at Customer instance.

5. Confirm that the Instance Details page shows the correct number of Exadata Storage Servers to be added. Then, click **Modify**.

6. On the Confirmation page, review the configuration settings. If you are satisfied, click **Modify**.

   Otherwise, click **Cancel** on the Confirmation page and again on the Instance Details page to cancel the wizard without updating the service instance.

Ensure that the addition of the Exadata Storage Servers completes successfully before you attempt to add anymore Exadata Storage Servers or compute nodes. If the operation fails for any reason, the failure must be remedied before any attempt to add anymore Exadata Storage Servers or compute nodes. Otherwise, the system may be left in an inconsistent state requiring manual intervention by Oracle.

Adding Exadata storage to the Exadata system configuration is an online operation, which does not affect the availability of your database deployments.

Adding Exadata Storage Servers triggers a rebalance operation, which occurs in the background and is designed to have minimal impact on the I/O performance of your system. Specifically, the existing storage allocation for each VM cluster is rebalanced across all of the Exadata Storage Servers, including the new ones. And, the newly added storage is proportionally distributed to the VM clusters, increasing the allocated storage for each VM cluster.

If you want to further modify the amount of storage that is allocated to a VM cluster, you must do it separately for each VM cluster. See Modifying an Existing VM Cluster.

Adding Compute Nodes

You can increase compute resources in your current Exadata system configuration by adding Exadata compute nodes subject to the following conditions:

- Each Exadata Cloud at Customer system configuration can have an absolute maximum of eight compute nodes. However, the practical maximum is more limited if you don't have enough free IP addresses for the additional compute nodes.

  Specifically, if your system is configured so that each VM cluster client network subnet is defined using a `/28` CIDR block (*`N.N.N.N`*`/28`) and each VM cluster backup network subnet is defined using a `/29` CIDR block, then your immediate expansion potential is limited to five compute servers. In such cases, expansion to more than five compute servers requires the redeployment of Exadata Cloud at Customer, which includes deleting and re-creating all of the VM clusters and database deployments on the system.

- The Exadata Cloud at Customer system configuration must be based on Oracle Exadata X7 hardware. You cannot add compute nodes to a starter system configuration based on Oracle Exadata X6 hardware.
- The Exadata Cloud at Customer service instance must be enabled to support multiple virtual machine (VM) clusters.

To add compute nodes to your current Exadata system configuration, you must first procure subscriptions for the additional compute nodes. Then, an Oracle field engineer physically installs the compute node hardware and registers it in the system.

For non-metered subscriptions only, before you use the following procedure to add the compute nodes to your Exadata Cloud at Customer instance, you must disable (scale back to zero) CPU core bursting. See Modifying the Compute Node Processing Power.

After the compute node hardware is physically provisioned and registered, you can use the following procedure to add the compute nodes to your Exadata Cloud at Customer instance:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Modify**.

   The Modify Oracle Database Exadata Cloud Service Instance wizard starts and the Instance Details page is displayed.

4. On the Instance Details page, select the **Compute Nodes** option to indicate that you want to add compute nodes to the Exadata Cloud at Customer instance.
5. Confirm that the Instance Details page shows the correct number of compute nodes to be added. Then, click **Modify**.
6. On the Confirmation page, review the configuration settings. If you are satisfied, click **Modify**.

   Otherwise, click **Cancel** on the Confirmation page and again on the Instance Details page to cancel the wizard without updating the service instance.

Ensure that the addition of the compute nodes completes successfully before you attempt to add anymore compute nodes or Exadata Storage Servers. If the operation fails for any reason, the failure must be remedied before any attempt to add anymore compute nodes or Exadata Storage Servers. Otherwise, the system may be left in an inconsistent state requiring manual intervention by Oracle.

Adding compute nodes to your Exadata Cloud at Customer instance is an online operation, which does not affect the availability of your existing VM clusters and database deployments.

After the compute nodes are added to your Exadata Cloud at Customer instance, you can use the new compute nodes when you create a VM cluster. You can also add the new compute nodes to an existing VM cluster. See Creating a VM Cluster and Modifying an Existing VM Cluster.

Suspending and Resuming an Exadata Cloud at Customer Instance

On Exadata Cloud at Customer instances that use Exadata X7 Infrastructure, you can suspend and later resume the service instance. When you suspend an Exadata Cloud at Customer instance, the associated compute nodes are shut down and the service does not consume any compute node CPU cores.

This facility is only available on Exadata Cloud at Customer instances that use Exadata X7 Infrastructure where the Exadata infrastructure is billed separately from the database server OCPUs.

To suspend or resume an Exadata Cloud at Customer instance:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and then choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Modify**.

   The Modify Oracle Database Exadata Cloud Service Instance wizard starts and the Instance Details page is displayed.

4. On the Instance Details page, set the option to suspend or resume the service instance. Then, click **Modify**.

   The current status of the service instance is displayed on the Instance Details page.

   If the service instance is Active, then you can suspend the service instance by checking the Suspend Service checkbox.

   If the service instance is Suspended, then you can resume the service instance by unchecking the Suspend Service checkbox.

5. On the Confirmation page, review the configuration settings. If you are satisfied, click **Modify**.

   You can click **Cancel** to cancel out of the wizard without suspending or reactivating the service instance.

Creating and Managing Snapshots of a Database Deployment

Oracle Database Exadata Cloud at Customer supports the creation of snapshot master databases, which can be used as the basis for creating space-efficient clone databases that can be created and destroyed very quickly and easily. Snapshot clones are often used for development, testing, or other purposes that require a transient database.

> **Note:**
>
> - To create and manage snapshot masters and clones, your Exadata Cloud at Customer instance must be configured with the SPARSE disk group. See Creating an Exadata Cloud at Customer Instance.
> - Exadata Cloud at Customer supports snapshots only with Oracle Database 12c, or later. You cannot create a snapshot master of a database using Oracle Database 11g Release 2.

The snapshot master is a completely independent database deployment that contains a read-only copy of the source database. It can continue to function as a snapshot master even if the source database deployment is deleted. When you create a snapshot master, the source database deployment is put into maintenance status while a copy is taken.

When you create a snapshot clone, Exadata Cloud at Customer creates another database deployment that references the snapshot master. To record changes in the clone database, Oracle writes altered blocks to the SPARSE disk group without changing the snapshot master. Thus, you can create several space-efficient clones based on the same snapshot master to use for various purposes, including application testing or branched application development work.

Here are the tasks for creating and managing snapshot masters and clones:

- Creating a Snapshot Master
- Creating a Clone Database Deployment from a Snapshot Master
- Listing Clone Database Deployments Created from a Snapshot Master
- Deleting a Clone Database Deployment
- Deleting a Snapshot Master

Creating a Snapshot Master

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the name of the database deployment for which you want to add a snapshot master.

   The Oracle Database Cloud Service Overview page is displayed.

3. Click the Administration tile and then click the Snapshots tab.

   The Oracle Database Cloud Service Snapshots page is displayed. Any snapshot masters already associated with the deployment are shown in the Available Snapshot Masters list.

4. Click **Create Snapshot Master** and specify the attributes associated with the snapshot master. Then, click **Create**.

   - **Snapshot Master Name** — enter a name for the snapshot master.
   - **DB Name** — enter a name for the database instances.

- **Administration Password** and **Confirm Password** — enter and then re-enter an administration password.

  The administration password is used to configure administration accounts and functions in the snapshot master database deployment, including the password for the Oracle Database SYS and SYSTEM users.

  > **Note:**
  >
  > Ensure that you remember the administration password associated with your snapshot master database deployment.

- **Hostnames** — specify one or more compute nodes that you want to host the database instances for this snapshot master.
- **Description** — enter a description of the snapshot master. (Optional)
- **ACFS** — select this option to use Oracle ASM Cluster File System (ACFS) to store the Oracle binaries for this snapshot master.

  > **Note:**
  >
  > By using ACFS, you can reduce the amount of local storage space consumed by Oracle binaries on each compute node. However, this option is only recommended for non-production databases on Exadata Cloud at Customer.

- **Clone Source Oracle Home** — select this option to clone the existing Oracle Home directory location and binaries into a new Oracle Home location for the snapshot master to use. If you do not select this option, then a new Oracle Home directory location is created and populated with binaries from the current default software image for the corresponding Oracle Database software version.

Listing Clone Database Deployments Created from a Snapshot Master

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the name of the snapshot master deployment that you are interested in.

   The Oracle Database Cloud Service Overview page is displayed.

3. Locate the Associations region on the Oracle Database Cloud Service Overview page.

   Click the triangular expand icon (▶) before Associations to view the list of clone deployments that are associated with the snapshot master.

   If the Associations region is not displayed, no clone deployments have been created from the snapshot master.

Deleting a Clone Database Deployment

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Locate the snapshot clone that you want to delete and choose **Delete** from that snapshot clone’s action menu (☰).
3. In the Delete Service window, confirm that you want to delete the clone database deployment by clicking **Delete**.

   Once deleted, the associated entry is removed from the list of database deployments displayed in the Oracle Database Cloud Service console.

Deleting a Snapshot Master

> **Note:**
>
> You cannot delete a snapshot master that has clone database deployments associated with it. You must first delete the clone deployments, as described in Deleting a Clone Database Deployment.

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.
2. Locate the snapshot master that you want to delete and choose **Delete** from that snapshot master’s action menu (☰).
3. In the Delete Service window, confirm that you want to delete the snapshot master by clicking **Delete**.

   If the window warns you that you cannot delete the snapshot master because there are existing clones, click **Close** and then delete the clones before trying to delete the snapshot master.

   Once deleted, the associated entry is removed from the list of database deployments displayed in the Oracle Database Cloud Service console.

Deleting a Database Deployment

When you no longer require a database deployment on Oracle Database Exadata Cloud at Customer, you can delete it.

To delete a database deployment:

1. Open the Instances page of the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.
2. Select **Delete** from the action menu (☰) corresponding to the database deployment that you want to delete.

   You are prompted to confirm the deletion.

3. Use the confirmation dialog to confirm that you want to delete the database deployment. Optionally, you can also select the option to delete the backups associated with the database deployment.

> **Note:**
>
> The option to delete the backups associated with the database deployment only exists for deployments that are created using Exadata Cloud at Customer release 17.1.5, or later.

Once deleted, the entry is removed from the list of database deployments displayed on the Oracle Database Cloud Service console.

Deleting an Exadata Cloud at Customer Instance

When you delete an Oracle Database Exadata Cloud at Customer instance you delete all of the software and data on the system, including all of the database deployments hosted on the system.

> **Note:**
>
> Deleting an Exadata Cloud at Customer instance may fail when the instance contains existing VM clusters or existing snapshot clone database deployments. To avoid this problem ensure that you delete all snapshot clone database deployments and all VM clusters before you attempt to delete an Exadata Cloud at Customer instance. See Deleting a Database Deployment and Deleting a VM Cluster.

To delete an Exadata Cloud at Customer instance:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Select **Delete** from the action menu (☰) corresponding to the service instance that you want to delete.

   A confirmation dialog appears.

4. Review the details in the confirmation dialog. Click **Delete** to delete the service instance, or click **Cancel** to return to the Service Details page without deleting the service instance.

Note:

Clicking **Delete** starts the process to delete the service instance. This process is fully automated and takes some time to complete. During this time you may still see the service instance listed in the Service Details page; however, you cannot access the service instance.

3

Managing Network Access to Exadata Cloud at Customer

Topics

- About Network Access to Exadata Cloud at Customer
- Generate a Secure Shell (SSH) Public/Private Key Pair
- Creating an SSH Tunnel to a Compute Node Port
- Controlling Network Access to Exadata Cloud at Customer
- Defining a Custom Host Name or Domain Name for Exadata Cloud at Customer
- Defining a Custom SCAN Host Name for Exadata Cloud at Customer
- Using Network Encryption and Integrity

About Network Access to Exadata Cloud at Customer

Network access to the compute nodes associated with Oracle Database Exadata Cloud at Customer is primarily provided by Secure Shell (SSH) connections on port 22. Other network protocols and services may also be used.

SSH Access on Port 22

SSH is a cryptographic network protocol that uses two keys, one public and one private, to provide secure communication between two networked computers. Port 22 is the standard TCP/IP port that is assigned to the SSH servers.

The public key is stored in the compute nodes associated with your Exadata Cloud at Customer environment. If no public key is associated with your Exadata Cloud at Customer environment you will be prompted to specify a public key when you create a database deployment. You can add a new SSH key to your Exadata Cloud at Customer environment by using the SSH Access menu option, which can be found in the action menu (☰) that is associated with each database deployment.

When you access any Exadata Cloud at Customer compute node using SSH, you must provide the private key that matches the public key.

For more information about generating the required SSH public/private key pair, see Generate a Secure Shell (SSH) Public/Private Key Pair.

Access to Other Network Services

With Exadata Cloud at Customer, the Exadata system environment resides inside your data center and no firewall is provided between Exadata Cloud at Customer and your network. Therefore, unless your network imposes other limitations, you may inherently configure and use all other network services on Exadata Cloud at Customer.

Generate a Secure Shell (SSH) Public/Private Key Pair

Several tools exist to generate SSH public/private key pairs. The topics in this section show how to generate an SSH key pair on UNIX, UNIX-like, and Windows platforms.

Topics:

- Generate an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility
- Generate an SSH Key Pair on Windows Using the PuTTYgen Program

Generate an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility

UNIX and UNIX-like platforms (including Solaris and Linux) include the ssh-keygen utility to generate SSH key pairs.

To generate an SSH key pair on UNIX and UNIX-like platforms using the ssh-keygen utility:

1. Navigate to your home directory:

   ```
   $ cd $HOME
   ```

2. Run the ssh-keygen utility, providing as ***filename*** your choice of file name for the private key:

   ```
   $ ssh-keygen -b 2048 -t rsa -f filename
   ```

   The ssh-keygen utility prompts you for a passphrase for the private key.

3. Enter a passphrase for the private key, or press Enter to create a private key without a passphrase:

   ```
   Enter passphrase (empty for no passphrase): passphrase
   ```

   > **Note:**
   >
   > While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

   The ssh-keygen utility prompts you to enter the passphrase again.

4. Enter the passphrase again, or press Enter again to continue creating a private key without a passphrase:

   ```
   Enter the same passphrase again: passphrase
   ```

5. The ssh-keygen utility displays a message indicating that the private key has been saved as *filename* and the public key has been saved as *filename*.pub. It also displays information about the key fingerprint and randomart image.

Generate an SSH Key Pair on Windows Using the PuTTYgen Program

The PuTTYgen program is part of PuTTY, an open source networking client for the Windows platform.

To generate an SSH key pair on Windows using the PuTTYgen program:

1. Download and install PuTTY or PuTTYgen.

   To download PuTTY or PuTTYgen, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

2. Run the PuTTYgen program.

   The PuTTY Key Generator window is displayed.

3. Set the **Type of key to generate** option to **SSH-2 RSA**.
4. In the **Number of bits in a generated key** box, enter **2048**.
5. Click Generate to generate a public/private key pair.

   As the key is being generated, move the mouse around the blank area as directed.

6. (Optional) Enter a passphrase for the private key in the **Key passphrase** box and reenter it in the **Confirm passphrase** box.

   > **Note:**
   >
   > While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

7. Click **Save private key** to save the private key to a file. To adhere to file-naming conventions, you should give the private key file an extension of `.ppk` (PuTTY private key).

   > **Note:**
   >
   > The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY as your SSH client. It cannot be used with other SSH client tools. Refer to the PuTTY documentation to convert a private key in this format to a different format.

8. Select all of the characters in the **Public key for pasting into OpenSSH authorized_keys file** box.

   Make sure you select all the characters, not just the ones you can see in the narrow window. If a scroll bar is next to the characters, you aren't seeing all the characters.

9. Right-click somewhere in the selected text and select **Copy** from the menu.
10. Open a text editor and paste the characters, just as you copied them. Start at the first character in the text editor, and do not insert any line breaks.

11. Save the text file in the same folder where you saved the private key, using the `.pub` extension to indicate that the file contains a public key.
12. If you or others are going to use an SSH client that requires the OpenSSH format for private keys (such as the `ssh` utility on Linux), export the private key:
    a. On the **Conversions** menu, choose **Export OpenSSH key**.
    b. Save the private key in OpenSSH format in the same folder where you saved the private key in `.ppk` format, using an extension such as `.openssh` to indicate the file's content.

Creating an SSH Tunnel to a Compute Node Port

To create an SSH tunnel to a port on a compute node associated with Oracle Database Exadata Cloud at Customer, you use Secure Shell (SSH) client software that supports tunneling.

Several SSH clients that support tunneling are freely available. The following sections show how to use SSH clients on the Linux and Windows platforms to connect to a compute node using an SSH tunnel.

> **Note:**
>
> An SSH tunnel cannot be used to connect to an Exadata Cloud at Customer database using the SCAN listeners because an SSH tunnel is a point-to-point connection to a specific port on a specific host IP address. However, the SCAN listeners route incoming connections to any of the available node listeners, which listen on a different set of virtual IP addresses. See Connecting Remotely to the Database by Using Oracle Net Services.

Creating an SSH Tunnel Using the ssh Utility on Linux

The Linux platform includes the ssh utility, an SSH client that supports SSH tunneling.

Before you use the ssh utility to create an SSH tunnel, you need the following:

- The IP address of the target compute node.

  The IP addresses associated with a database deployment on Oracle Database Exadata Cloud at Customer are listed on the details page associated with the database deployment. See Viewing Detailed Information for a Database Deployment.
- The SSH private key file that pairs with the public key used during the database deployment creation process.
- The port number for which you want to create an SSH tunnel.

To create an SSH tunnel for a port using the ssh utility on Linux:

1. In a command shell, set the file permissions of the private key file so that only you have access to it:

```
$ chmod 600 private-key-file
```

private-key-file is the path to the SSH private key file that matches the public key used during the database deployment creation process.

2. Run the ssh utility:

   ```
   $ ssh -i private-key-file -L local-port:target-ip-address:target-port opc@target-
   ip-address
   ```

   where:

   - *private-key-file* is the path to the SSH private key file.
   - *local-port* is the number of an available port on your Linux system. Specify a port number greater than 1023 and less than 49152 to avoid conflicts with ports that are reserved for the system. As a good practice, and for the sake of simplicity, you should specify the same port number as the one to which you are creating a tunnel.
   - *target-ip-address* is the IP address of the target compute node in *x.x.x.x* format.
   - *target-port* is the port number to which you want to create a tunnel.

3. If this is the first time you are connecting to the target compute node, the ssh utility prompts you to confirm the public key. In response to the prompt, enter **yes**.

After the SSH tunnel is created, you can access the port on the target compute node by specifying localhost:*local-port* on your Linux system.

Creating an SSH Tunnel Using the PuTTY Program on Windows

PuTTY is a freely available SSH client program for Windows that supports SSH tunneling.

Before you use the ssh utility to create an SSH tunnel, you need the following:

- The IP address of the target compute node.

  The IP addresses associated with a database deployment on Oracle Database Exadata Cloud at Customer are listed on the details page associated with the database deployment. See Viewing Detailed Information for a Database Deployment.

- The SSH private key file that pairs with the public key used during the database deployment creation process.
- The port number for which you want to create an SSH tunnel.

To create an SSH tunnel for a port using the PuTTY program on Windows:

1. Download and install PuTTY.

   To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

2. Run the PuTTY program.

   The PuTTY Configuration window is displayed, showing the Session panel.

3. Configure SSH connectivity:

   a. In **Host Name (or IP address)** box, enter the IP address of the target compute node.

   b. Confirm that the **Connection type** option is set to **SSH**.

   c. In the Category tree, expand **Connection** if necessary and then click **Data.**

      The Data panel is displayed.

   d. In **Auto-login username** box, enter **oracle**.

   e. Confirm that the **When username is not specified** option is set to **Prompt**.
   f. In the Category tree, expand **SSH** and then click **Auth**.

      The Auth panel is displayed.
   g. Click the **Browse** button next to the **Private key file for authentication** box. Then, in the Select private key file window, navigate to and open the private key file that matches the public key used during the database deployment creation process.
4. Add a forwarded port:
   a. In the Category tree, click **Tunnels**.

      The Tunnels panel is displayed.
   b. In the **Source Port** box, enter the number of an available port on your system. Specify a port number greater than 1023 and less than 49152 to avoid conflicts with ports that are reserved for the system. As a good practice, and for the sake of simplicity, you should specify the same port number as the one to which you are creating a tunnel.
   c. In the **Destination box**, enter the IP address of the target compute node, a colon, and the port number to which you want to create a tunnel; for example, 192.0.2.100:1521.
   d. Confirm that the **Local** and **Auto** options are set.
   e. Click Add to add the forwarded port.

      The new forwarded port appears in the **Forwarded ports** list.
5. In the Category tree, click **Session**.

   The Session panel is displayed.
6. In the **Saved Sessions** box, enter a name for this connection configuration. Then, click **Save**.
7. Click **Open** to open the connection.

   The PuTTY Configuration window is closed and the PuTTY window is displayed.
8. If this is the first time you are connecting to the target compute node, the PuTTY Security Alert window is displayed, prompting you to confirm the public key. Click **Yes** to continue connecting.

After the SSH tunnel is created, you can access the port on the target compute node by specifying `localhost:`*`local-port`* on your system, where *`local-port`* is the source port that you specified when creating the tunnel.

Controlling Network Access to Exadata Cloud at Customer

You can control network access to your Oracle Database Exadata Cloud at Customer by listing network addresses that are either invited to connect, or excluded from connecting as follows:

- You can use Oracle Net Services valid node checking to define a list that Oracle Net Services uses to allow or disallow connections from. You enable and control valid node checking by setting parameters in the `sqlnet.ora` file, which is typically located at `$ORACLE_HOME/network/admin/`*`dbname`*`/sqlnet.ora`. Oracle Net Services valid node checking only controls Oracle Net Services connections.

Connections by other means, such as SSH, are not arbitrated by Oracle Net Services valid node checking.

To enable Oracle Net Services valid node checking, set `TCP.VALIDNODE_CHECKING = yes` in the `sqlnet.ora` file. To control Oracle Net Services valid node checking use the following parameters:

- `TCP.EXCLUDED_NODES` specifies clients that are denied access to the database. The parameter can be set to a list of host names or addresses and the list may include wildcards for IPv4 addresses and CIDR (Classless Inter-Domain Routing) notation for IPv4 and IPv6 addresses. For example:

  ```
  TCP.EXCLUDED_NODES=(finance.us.example.com, mktg.us.example.com,
  192.168.2.25, 172.30.*, 2001:DB8:200C:417A/32)
  ```

- `TCP.INVITED_NODES` specifies clients that are allowed access to the database. This list takes precedence over the `TCP.EXCLUDED_NODES` parameter if both lists are present. The parameter can be set to a list of host names or addresses and the list may include wildcards for IPv4 addresses and CIDR notation for IPv4 and IPv6 addresses. For example:

  ```
  TCP.INVITED_NODES=(sales.us.example.com, hr.us.example.com,
  192.168.*, 2001:DB8:200C:433B/32)
  ```

Defining a Custom Host Name or Domain Name for Exadata Cloud at Customer

You can associate a custom host name or domain name to the public IP address of a compute node associated with your Oracle Database Exadata Cloud at Customer environment.

To associate a custom host name to the public IP address of a compute node, contact the administrator of your DNS (Domain Name Service) and request a custom DNS record for the compute node’s public IP address. For example, if your domain is `example.com` and you wanted to use `clouddb1` as the custom host name for a compute node, you would request a DNS record that associates `clouddb1.example.com` to your compute node's public IP address.

To associate a custom domain name to the public IP address of a compute node:

1. Register your domain name through a third-party domain registration vendor, such as `Register.com`, `Namecheap`, and so on. For example, `example.com`.
2. Resolve your domain name to the IP address of the Exadata Cloud at Customer compute node, using the third-party domain registration vendor console. For more information, refer to the third-party domain registration documentation.

You can obtain the public IP address of a compute node by viewing details as described in Viewing Detailed Information for a Database Deployment.

Defining a Custom SCAN Host Name for Exadata Cloud at Customer

Single Client Access Name (SCAN) is an Oracle Grid Infrastructure feature that provides a single name for clients to access Oracle databases running in a cluster.

By default, every database deployment on Oracle Database Exadata Cloud at Customer is associated with a SCAN, and the SCAN is associated with 3 virtual IP addresses (VIPs). Each SCAN VIP is also associated with a SCAN listener, that provides a connection endpoint for Oracle database connections using Oracle Net Services. To maximize availability, Oracle Grid Infrastructure distributes the SCAN VIPs and SCAN listeners across the available cluster nodes. In addition, the SCAN VIPs and SCAN listeners are automatically migrated to a surviving node in the case of node shutdown or failure. The aim is to ensure that Oracle clients always have a single, reliable set of connection endpoints that can service all of the databases running in the cluster.

You can define a custom host name for the SCAN VIP addresses associated with Exadata Cloud at Customer. To do so, contact the administrator of your DNS (Domain Name Service) and request a custom DNS record that resolves to all three of the SCAN VIP addresses. For example, if your domain is `example.com` and you wanted to use `db1scan` as the custom SCAN host name, you would request a DNS record that resolves `db1scan.example.com` to the three SCAN VIP addresses associated with your database deployments. You can obtain the SCAN VIP addresses by viewing details as described in Viewing Detailed Information for a Database Deployment.

Using Network Encryption and Integrity

To secure connections to your Oracle Database Exadata Cloud at Customer databases, you can use native Oracle Net Services encryption and integrity capabilities.

Encryption of network data provides data privacy so that unauthorized parties are not able to view data as it passes over the network. In addition, integrity algorithms protect against data modification and illegitimate replay.

Oracle Database provides the Advanced Encryption Standard (AES), DES, 3DES, and RC4 symmetric cryptosystems for protecting the confidentiality of Oracle Net Services traffic. It also provides a keyed, sequenced implementation of the Message Digest 5 (MD5) algorithm or the Secure Hash Algorithm (SHA-1 and SHA-2) to protect against integrity attacks.

By default, database deployments on Exadata Cloud at Customer are configured to enable native Oracle Net Services encryption and integrity. Also, by default, Oracle Net Services clients are configured to enable native encryption and integrity when they connect to an appropriately configured server. If your Oracle Net Services client is configured to explicitly reject the use of native encryption and integrity then connection attempts will fail.

You can check your configuration and verify the use of native Oracle Net Services encryption and integrity as follows. For more general information about configuring native Oracle Net Services encryption and integrity, see "Configuring Oracle Database Network Encryption and Data Integrity" in *Oracle Database Security Guide* for Release

18, 12.2 or 12.1 or "Configuring Network Data Encryption and Integrity for Oracle Servers and Clients" in *Database Advanced Security Administrator's Guide* for Release 11.2.

Checking your Exadata Cloud at Customer environment

The following procedure outlines the basic steps required to confirm that native Oracle Net Services encryption and integrity are enabled in your Exadata Cloud at Customer environment.

> **Note:**
>
> The procedure relates to a single compute node and a single database or group of databases that share a set of Oracle binaries. For Exadata Cloud at Customer , you should confirm that the configuration settings are consistent across all of the compute nodes and database deployments in the environment.

1. In a command shell, connect to the compute node as the `oracle` user. See Connecting to a Compute Node Through Secure Shell (SSH).
2. Configure your Oracle Database environment variable settings:

   ```
   $ . oraenv
   ```

3. Change directories to the location of the `sqlnet.ora` configuration file. For example:

   ```
   $ cd $ORACLE_HOME/network/admin/dbname
   $ ls sqlnet.ora
   sqlnet.ora
   ```

4. View the `sqlnet.ora` file and confirm that it contains the following parameter settings:

   ```
   SQLNET.ENCRYPTION_SERVER = required
   SQLNET.CRYPTO_CHECKSUM_SERVER = required
   ```

   The `required` setting enables the encryption or integrity service and disallows the connection if the client side is not enabled for the security service. This is the default setting for database deployments on Exadata Cloud at Customer.

Checking your Oracle Net Services Client Configuration

The following procedure outlines the basic steps required to confirm that native encryption and integrity are enabled in your Oracle Net Services client configuration.

1. In a command shell, connect to the Oracle Net Services client.
2. Change directories to the location of the `tnsnames.ora` and `sqlnet.ora` configuration files, for example:

   ```
   $ cd $ORACLE_HOME/network/admin
   $ ls *.ora
   sqlnet.ora tnsnames.ora
   ```

3. View the `sqlnet.ora` file and confirm that it *does not* contain the following parameter settings:

   ```
   SQLNET.ENCRYPTION_CLIENT = rejected
   SQLNET.CRYPTO_CHECKSUM_CLIENT = rejected
   ```

   The `rejected` setting explicitly disables the encryption or integrity service, even if the server requires it. When a client with an encryption or integrity service setting of `rejected` connects to a server with the `required` setting, the connection fails with the following error: `ORA-12660: Encryption or crypto-checksumming parameters incompatible`.

   Because native Oracle Net Services encryption and integrity are enabled in your Exadata Cloud at Customer environment by default, any parameter setting other than `rejected`, or no setting at all, would result in the use of native encryption and integrity.

Verifying the use of Native Encryption and Integrity

You can verify the use of native Oracle Net Services encryption and integrity by connecting to your Oracle database and examining the network service banner entries associated with each connection. This information is contained in the `NETWORK_SERVICE_BANNER` column of the `V$SESSION_CONNECT_INFO` view. The following example shows the SQL command used to display the network service banner entries associated with the current connection:

```
SQL> select network_service_banner
     from v$session_connect_info
     where sid in (select distinct sid from v$mystat);
```

The following example output shows banner information for the available encryption service and the crypto-checksumming (integrity) service, including the algorithms in use:

```
NETWORK_SERVICE_BANNER
-----------------------------------------------------------------------
--------------
TCP/IP NT Protocol Adapter for Linux: Version 12.1.0.2.0 - Production
Encryption service for Linux: Version 12.1.0.2.0 - Production
AES256 Encryption service adapter for Linux: Version 12.1.0.2.0 -
Production
Crypto-checksumming service for Linux: Version 12.1.0.2.0 - Production
SHA1 Crypto-checksumming service adapter for Linux: Version 12.1.0.2.0
- Production
```

If native Oracle Net Services encryption and integrity was not in use, the banner entries would still include entries for the available security services; that is, the services linked into the Oracle Database software. However, there would be no entries indicating the specific algorithms in use for the connection. The following output shows an example:

```
NETWORK_SERVICE_BANNER
-----------------------------------------------------------------------
```

```
---------------
TCP/IP NT Protocol Adapter for Linux: Version 12.1.0.2.0 - Production
Encryption service for Linux: Version 12.1.0.2.0 - Production
Crypto-checksumming service for Linux: Version 12.1.0.2.0 - Production
```

4

Accessing Exadata Cloud at Customer

This section describes how to access tools, utilities and interfaces available in Oracle Database Exadata Cloud at Customer.

Topics

- Connecting to a Compute Node Through Secure Shell (SSH)
- Accessing Enterprise Manager Database Express 18c, or later
- Accessing Enterprise Manager Database Express 12c
- Accessing Enterprise Manager 11g Database Control
- Connecting Remotely to the Database by Using Oracle Net Services

Connecting to a Compute Node Through Secure Shell (SSH)

You use Secure Shell (SSH) client software to establish a secure connection and gain local access the tools, utilities, and other resources on an Oracle Database Exadata Cloud at Customer compute node.

By default, Exadata Cloud at Customer is configured to enable SSH connections using the `opc` and `oracle` operating system user accounts. For tasks that use other operating system user accounts, you can start a command shell as the `root` user by connecting to `opc` and running `sudo -s`.

Several SSH clients are freely available. The following sections show how to use SSH clients on UNIX, UNIX-like, and Windows platforms to connect to a compute node associated with Exadata Cloud at Customer.

Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms

UNIX and UNIX-like platforms (including Solaris and Linux) include the ssh utility, an SSH client.

Before You Begin

Before you use the ssh utility to connect to a compute node, you need the following:

- The IP address of the compute node

  The IP address of a compute node associated with a database deployment on Oracle Database Exadata Cloud at Customer is listed on the Oracle Database Cloud Service Overview page. See Viewing Detailed Information for a Database Deployment.
- The SSH private key file that matches the public key associated with the deployment.

Procedure

To connect to a compute node using the ssh utility on UNIX and UNIX-like platforms:

1. In a command shell, set the file permissions of the private key file so that only you have access to it:

   ```
   $ chmod 600 private-key-file
   ```

   `private-key-file` is the path to the SSH private key file that matches the public key that is associated with the deployment.

2. Run the ssh utility:

   ```
   $ ssh -i private-key-file user-name@node-ip-address
   ```

   where:

   - *`private-key-file`* is the path to the SSH private key file.
   - *`user-name`* is the operating system user you want to connect as:
     - Connect as the user **`oracle`** to perform most operations; this user does not have root access to the compute node.
     - Connect as the user **`opc`** to perform operations that require root access to the compute node, such as backing up or patching; this user can use the `sudo` command to gain root access to the compute node.
   - *`node-ip-address`* is the IP address of the compute node in *`x.x.x.x`* format.

3. If this is the first time you are connecting to the compute node, the ssh utility prompts you to confirm the public key. In response to the prompt, enter **`yes`**.

Connecting to a Compute Node Using the PuTTY Program on Windows

PuTTY is a freely available SSH client program for Windows.

Before You Begin

Before you use the PuTTY program to connect to a compute node, you need the following:

- The IP address of the compute node

  The IP address of a compute node associated with a database deployment on Oracle Database Exadata Cloud at Customer is listed on the Oracle Database Cloud Service Overview page. See Viewing Detailed Information for a Database Deployment.

- The SSH private key file that matches the public key associated with the deployment. This private key file must be in the PuTTY .ppk format. If the private key file was originally created on the Linux platform, you can use the PuTTYgen program to convert it to the .ppk format.

Procedure

To connect to a compute node using the PuTTY program on Windows:

1. Download and install PuTTY.

   To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

2. Run the PuTTY program.

   The PuTTY Configuration window is displayed, showing the Session panel.

3. In **Host Name (or IP address)** box, enter the IP address of the compute node.
4. Confirm that the **Connection type** option is set to **SSH**.
5. In the Category tree, expand **Connection** if necessary and then click **Data**.

   The Data panel is displayed.

6. In **Auto-login username** box, enter the user you want to connect as:
   - Connect as the user **oracle** to perform most operations; this user does not have root access to the compute node.
   - Connect as the user **opc** to perform operations that require root access to the compute node, such as backing up or patching; this user can use the `sudo` command to gain root access to the compute node.
7. Confirm that the **When username is not specified** option is set to **Prompt**.
8. In the Category tree, expand **SSH** and then click **Auth**.

   The Auth panel is displayed.

9. Click the **Browse** button next to the **Private key file for authentication** box. Then, in the Select private key file window, navigate to and open the private key file that matches the public key that is associated with the deployment.
10. In the Category tree, click **Session**.

    The Session panel is displayed.

11. In the **Saved Sessions** box, enter a name for this connection configuration. Then, click **Save**.
12. Click **Open** to open the connection.

    The PuTTY Configuration window is closed and the PuTTY window is displayed.

13. If this is the first time you are connecting to the VM, the PuTTY Security Alert window is displayed, prompting you to confirm the public key. Click **Yes** to continue connecting.

Accessing Enterprise Manager Database Express 18c, or later

Enterprise Manager Database Express (EM Express), a web-based tool for managing Oracle Database, is available on Oracle Database Exadata Cloud at Customer database deployments created using Oracle Database 18c, or later.

You can access EM Express in the following ways:

- Using the Open EM Console menu item
- Using a direct URL
- Using an SSH tunnel

Using the Open EM Console Menu Item to Access EM Express

1. Open the Instances page of the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. From the action menu (☰) for the deployment, select **Open EM Console**.

3. If your browser displays a warning that your connection is not secure or not private, use the browser's advanced option to ignore the warning and continue.

   You get this warning because Exadata Cloud at Customer database deployments use a self-signed certificate to provide HTTPS (secure HTTP) connectivity, and such certificates are considered suspicious by many web browsers.

4. When prompted for a user name and password, enter the name of a user with the `DBA` privilege (such as `SYS` or `SYSTEM`) and the password.

   Enter a PDB name if you want to access a specific PDB or leave it blank to access the root container.

   If you want to connect with `SYSDBA` privileges, select **as SYSDBA**.

   After entering or selecting the required values, click **Login**.

The **Open EM Console** menu item is also available from the action menu (☰) on the Oracle Database Cloud Service Instance Overview page.

Using a Direct URL to Access EM Express

1. In your web browser, go to the following URL:

   ```
   https://node-ip-address:EM-Express-port/em
   ```

   where *`node-ip-address`* is the public IP address of the compute node hosting EM Express, and *`EM-Express-port`* is the EM Express port used by the database.

2. If your browser displays a warning that your connection is not secure or not private, use the browser's advanced option to ignore the warning and continue.

   You get this warning because Exadata Cloud at Customer database deployments use a self-signed certificate to provide HTTPS (secure HTTP) connectivity, and such certificates are considered suspicious by many web browsers.

3. When prompted for a user name and password, enter the name of a user with `DBA` privilege (such as `SYS` or `SYSTEM`) and the password.

   Enter a PDB name if you want to access a specific PDB or leave it blank to access the root container.

   If you want to connect with `SYSDBA` privileges, select **as SYSDBA**.

   After entering or selecting the required values, click **Login**.

Using an SSH Tunnel to Access EM Express

1. Create an SSH tunnel to the EM Express port on the compute node hosting EM Express. For information about creating an SSH tunnel, see Creating an SSH Tunnel to a Compute Node Port.

2. After creating the SSH tunnel, go to the following URL:

```
https://localhost:EM-Express-port/em
```

where *EM-Express-port* is the EM Express port used by the database.

3. When prompted for a user name and password, enter the name of a user with DBA privilege (such as SYS or SYSTEM) and the password.

   Enter a PDB name if you want to access a specific PDB or leave it blank to access the root container.

   If you want to connect with SYSDBA privileges, select **as SYSDBA**.

   After entering or selecting the required values, click **Login**.

Finding the EM Express Network Port

When a database deployment is created, Exadata Cloud at Customer automatically sets a port for EM Express. You do not need to perform any manual configuration steps. Each database deployment is allocated a unique port number in a range starting with 5500, 5501, 5502, and so on.

To find the port that is in use for a specific database, connect to the database as a database administrator and execute the query shown in the following example:

```
SQL> select dbms_xdb_config.getHttpsPort() from dual;

DBMS_XDB_CONFIG.GETHTTPSPORT()
------------------------------
                          5502
```

Accessing Enterprise Manager Database Express 12c

Enterprise Manager Database Express 12c (EM Express), a web-based tool for managing Oracle Database 12c, is available on Oracle Database Exadata Cloud at Customer database deployments created using Oracle Database 12c Release 1 (12.1) or Oracle Database 12c Release 2 (12.2).

Before you access EM Express to manage your database you must determine, and in some cases configure, the network port that is used to access EM Express as follows:

- **To manage the CDB.** When a database deployment is created, Exadata Cloud at Customer automatically sets a port for EM Express access to the CDB. You do not need to perform any manual configuration steps. Each database deployment is allocated a unique port number. The allocations use ports in a range starting with 5500, 5501, 5502, and so on.
- **To manage a PDB with Oracle Database 12c Release 1 (version 12.1).** For a version 12.1 database deployment, you must manually set a port for each PDB you want to manage using EM Express. See Setting the Port for EM Express to Manage a PDB.
- **To manage a PDB with Oracle Database 12c Release 2 (version 12.2).** With Oracle Database 12c Release 2, EM Express can be configured to access the CDB and all PDBs on a single port, which is known as the global port. For version 12.2 database deployments created after early December 2016, the global port is set by default. For deployments created prior to December 2016, see Setting the Global Port for EM Express to Manage a CDB and the PDBs (Oracle Database 12.2 Only).

> **Note:**
>
> To confirm the port that is in use for a specific database, connect to the database as a database administrator and execute the query shown in the following example:
>
> ```
> SQL> select dbms_xdb_config.getHttpsPort() from dual;
>
> DBMS_XDB_CONFIG.GETHTTPSPORT()
> ------------------------------
>                           5502
> ```

Setting the Port for EM Express to Manage a PDB

In Oracle Database 12c Release 1, a unique HTTPS port must be configured for the root container (CDB) and each PDB that you manage using EM Express.

To configure a HTTPS port so that you can manage a PDB with EM Express:

1. Invoke SQL*Plus and log in to the PDB as the `SYS` user with `SYSDBA` privileges.
2. Execute the `DBMS_XDB_CONFIG.SETHTTPSPORT` procedure.

   ```
   SQL> exec dbms_xdb_config.sethttpsport(port-number)
   ```

Setting the Global Port for EM Express to Manage a CDB and the PDBs (Oracle Database 12.2 Only)

In Oracle Database 12c Release 2, you can configure a single port (known as the global port), which enables you to use EM Express to connect to all of the PDBs in the CDB using the HTTPS port for the CDB.

In database deployments created after early December 2016, the global port is set by default.

To configure the global port in deployments created before December 2016:

1. Invoke SQL*Plus and log in to the root container (CDB) as the `SYS` user with `SYSDBA` privileges.
2. Execute the `DBMS_XDB_CONFIG.SETGLOBALPORTENABLED` procedure.

   ```
   SQL> exec dbms_xdb_config.SetGlobalPortEnabled(TRUE)
   ```

Accessing EM Express Using the EM Express Port

If the EM Express port is not blocked, you can access EM Express by directing your browser to the URL `https://`*`node-ip-address`*`:`*`EM-Express-port`*`/em`, where *`node-ip-address`* is the public IP address of the compute node hosting EM Express, and *`EM-Express-port`* is the EM Express port used by the database.

You can also access EM Express to manage the CDB in 12.1 or the root container and PDBs through the global port in 12.2 through the Oracle Database Cloud Service console:

1. Open the Instances page of the Oracle Database Cloud Service console.

For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. From the action menu (☰) for the deployment, select **Open EM Console**.

   The EM Express login page is displayed.

3. Enter the name of a user with the `DBA` privilege (such as `SYS` or `SYSTEM`) and the password. To connect with `SYSDBA` privileges, select **as sysdba**. Then click **Login**.

This option is also available from the action menu (☰) on the Oracle Database Cloud Service Instance Overview page.

Accessing EM Express Using an SSH Tunnel

To access EM Express when its port is blocked, you must create an SSH tunnel to the EM Express port on the compute node hosting EM Express. See Creating an SSH Tunnel to a Compute Node Port.

After the SSH tunnel is created, you can access EM Express by directing your browser to the URL `https://localhost:`*`EM-Express-port`*`/em`.

After the EM Express login page is displayed, enter the name of a user with the `DBA` privilege (such as `SYS` or `SYSTEM`) and the password. To connect with `SYSDBA` privileges, select **as sysdba**. Then click **Login**.

Accessing Enterprise Manager 11g Database Control

Enterprise Manager 11g Database Control (Database Control), a web-based tool for managing Oracle Database 11g, is available on Oracle Database Exadata Cloud at Customer database deployments created using Oracle Database 11g Release 2.

You can access Database Control in the following ways:

- Using the Open EM Console menu item
- Using a direct URL
- Using an SSH tunnel

Using the Open EM Console Menu Item to Access Database Control

1. Open the Instances page of the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. From the action menu (☰) for the deployment, select **Open EM Console**.

3. If your browser displays a warning that your connection is not secure or not private, use the browser's advanced option to ignore the warning and continue.

   You get this warning because Exadata Cloud at Customer database deployments use a self-signed certificate to provide HTTPS (secure HTTP) connectivity, and such certificates are considered suspicious by many web browsers.

4. When prompted for a user name and password, enter the name of a user with the `DBA` privilege (such as `SYS` or `SYSTEM`) and the password.

   If you want to connect with `SYSDBA` privileges, select **as SYSDBA**.

After entering or selecting the required values, click **Login**.

The **Open EM Console** menu item is also available from the action menu (☰) on the Oracle Database Cloud Service Instance Overview page.

Using a Direct URL to Access Database Control

1. In your web browser, go to the following URL:

   ```
   https://node-ip-address:DB-Control-port/em
   ```

   where *node-ip-address* is the public IP address of the compute node hosting Database Control, and *DB-Control-port* is the Database Control port used by the database.

2. If your browser displays a warning that your connection is not secure or not private, use the browser's advanced option to ignore the warning and continue.

   You get this warning because Exadata Cloud at Customer database deployments use a self-signed certificate to provide HTTPS (secure HTTP) connectivity, and such certificates are considered suspicious by many web browsers.

3. When prompted for a user name and password, enter the name of a user with the `DBA` privilege (such as `SYS` or `SYSTEM`) and the password.

   If you want to connect with `SYSDBA` privileges, select **as SYSDBA**.

   After entering or selecting the required values, click **Login**.

Using an SSH Tunnel to Access Database Control

1. Create an SSH tunnel to the Database Control port on the compute node hosting Database Control. For information about creating an SSH tunnel, see Creating an SSH Tunnel to a Compute Node Port.

2. After creating the SSH tunnel, go to the following URL:

   ```
   https://localhost:DB-Control-port/em
   ```

   where *DB-Control-port* is the Database Control port used by the database.

3. When prompted for a user name and password, enter the name of a user with the `DBA` privilege (such as `SYS` or `SYSTEM`) and the password.

   If you want to connect with `SYSDBA` privileges, select **as SYSDBA**.

   After entering or selecting the required values, click **Login**.

Finding the Database Control Network Port

When a database deployment is created, Exadata Cloud at Customer automatically sets a port for Database Control. You do not need to perform any manual configuration steps. Each database deployment is allocated a unique port number in a range starting with 1158, 1159, 1160, and so on.

To find the port that is in use for a specific database, investigate the `REPOSITORY_URL` entry in the `$ORACLE_HOME/`*`Hostname_SID`*`/sysman/config/emd.properties` file. In the preceding file name, *Hostname* is the host name of the compute node hosting Database Control, and *SID* is the Oracle Database system identifier (SID).

Connecting Remotely to the Database by Using Oracle Net Services

Oracle Database Exadata Cloud at Customer supports remote database access by using Oracle Net Services.

Because Exadata Cloud at Customer uses Oracle Grid Infrastructure, you can make connections by using Single Client Access Name (SCAN), which is a feature that provides a consistent mechanism for clients to access all of the Oracle databases running in a cluster.

By default, the SCAN is associated with three virtual IP addresses (VIPs). Each SCAN VIP is also associated with a SCAN listener that provides a connection endpoint for Oracle Database connections using Oracle Net Services. To maximize availability, Oracle Grid Infrastructure distributes the SCAN VIPs and SCAN listeners across the available cluster nodes. In addition, the SCAN VIPs and SCAN listeners are automatically migrated to a surviving node if a node fails or shuts down. The aim is to ensure that Oracle Database clients always have a single, reliable set of connection endpoints that can service all of the databases running in the cluster.

The SCAN listeners are in addition to the Oracle Net Listeners that run on every node in the cluster, which are also known as the node listeners. When an Oracle Net Services connection comes through SCAN, the SCAN listener routes the connection to one of the node listeners and plays no further part in the connection. A combination of factors including listener availability, database instance placement, and workload distribution determines which node listener receives each connection.

Before You Can Connect

To make a remote database connection on Exadata Cloud at Customer by using Oracle Net Services, ensure that there is no firewall or other network control device that prevents access to the Oracle Net Listener port (typically port 1521). For maximum efficiency, also ensure access to the Oracle Notification Service port (typically port 6200). If ONS is inaccessible, the Oracle Net Services connection functions but some operations are subject to extra wait times.

To make a remote database connection by using Oracle Net Services, you also need the following information:

- The IP addresses for your SCAN VIPs. These IP addresses are contained in the detailed information associated with each database deployment. See Viewing Detailed Information for a Database Deployment.
- The database identifier, either the database SID or service name. For database deployments running Oracle Database 11g, you can identify the database by using the SID. For deployments running Oracle Database 12c, or later, connecting to the database by specifying the database SID connects you to the CDB (container database). To connect to a PDB (pluggable database), specify the service name of the pluggable database by using the following format:

  ```
  pdb.network-domain
  ```

  where *pdb* is the name of the PDB and *network-domain* is the network domain name associated with your Exadata Cloud at Customer environment; for example:

  ```
  PDB1.us2.oraclecloud.com
  ```

You can determine the network domain name associated with your Exadata Cloud at Customer environment by viewing details as described in Viewing Detailed Information for a Database Deployment.

Creating an Oracle Net Services Connection by Using SCAN

To create an Oracle Net Services connection by using the SCAN listeners you can choose between two approaches. You can:

- Use a connect descriptor that references all of the SCAN VIPs.

  This approach requires you to supply all of the SCAN VIP addresses and allows Oracle Net Services to connect to an available SCAN listener. A Net Services alias is typically used to provide a convenient name for the connect descriptor. For example:

  ```
  alias-name = (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=SCAN-VIP-1)(PORT=1521))
      (ADDRESS=(PROTOCOL=tcp)(HOST=SCAN-VIP-2)(PORT=1521))
      (ADDRESS=(PROTOCOL=tcp)(HOST=SCAN-VIP-3)(PORT=1521)))
    (CONNECT_DATA=
      (sid-or-service-entry)))
  ```

  where:

  – `alias-name` is the name you use to identify the alias.

  – `SCAN-VIP-[1-3]` are the IP addresses for the SCAN VIPs.

  – `sid-or-service-entry` identifies the database SID or service name using one of the following formats:

    * `SID=sid-name`; for example, `SID=ORCL`.

    * `SERVICE_NAME=service-name`; for example, `SERVICE_NAME=PDB1.us2.oraclecloud.com`.

  > **Note:**
  >
  > By default, Oracle Net Services randomly selects one of the addresses in the address list to balance the load between the SCAN listeners.

  A suitable connect descriptor is contained in the database deployment connect string, which you can obtain by viewing details as described in Viewing Detailed Information for a Database Deployment. For database deployments running Oracle Database 11g, you can use the supplied connect string to connect to your database. For deployments running Oracle Database 12c, or later, you must modify the supplied connect string to specify the service name of the PDB or CDB that you want to connect to.

- Use a connect identifier that references a custom SCAN name.

  Using this approach, the SCAN name resolves to one of the three SCAN VIPs and the corresponding SCAN listener handles the connection. See Defining a Custom SCAN Host Name for Exadata Cloud at Customer.

To create an Oracle Net Services connection using a customer SCAN name, you can use the easy connect method to specify a connect identifier with the following format:

```
SCAN-name:1521/sid-or-service-entry
```

For example:

```
exa1scan.example.com:1521/ORCL
```

or

```
exa1scan.example.com:1521/PDB1.us2.oraclecloud.com
```

5 Administering Exadata Cloud at Customer

This section describes tasks for administering your Oracle Database Exadata Cloud at Customer environment and the Oracle databases contained therein.

Topics

- Using Exadata I/O Resource Management
- Adding an SSH Public Key
- Removing an SSH Public Key
- Updating the Cloud Tooling on Exadata Cloud at Customer
- Administering VM Clusters
- Administering Oracle Homes
- Administering Software Images
- Administering a Data Guard Configuration
- Administering Pluggable Databases
- Registering a Database
- Maintaining the Manageability of Exadata Cloud at Customer
- Loading Data into the Oracle Database on Exadata Cloud at Customer
- Tuning Oracle Database Performance on Exadata Cloud at Customer
- Monitoring and Managing Oracle Database on Exadata Cloud at Customer
- Monitoring and Managing Exadata Storage Servers on Exadata Cloud at Customer
- Managing the Log and Diagnostic Files on Exadata Cloud at Customer

Using Exadata I/O Resource Management

Oracle Database Exadata Cloud at Customer provides an interface for Exadata I/O Resource Management (IORM) that enables prioritization of I/O resources amongst different databases.

Exadata IORM allows workloads and databases to share I/O resources automatically according to user-defined policies. Exadata Cloud at Customer provides a simple interface to enable IORM across multiple databases.

This facility uses a system of shares that are allocated amongst all of the databases that run on the Exadata system. Each database is assigned a share value between 1 and 32, with 1 being the lowest share, and 32 being the highest share. The share value represents the relative importance of each database.

Every database is automatically assigned a default share value of 1. In this state, every database receives an even share of the available I/O resources. Increasing the share value for a specific database increases its relative importance, and consequently decreases the amount of I/O available for all of the other databases.

For example, on an Exadata system with four databases, one share is allocated to each database by default. This ensures that each database is allocated 1 out of every 4 I/Os when the system becomes loaded enough for IORM to intervene. If the share value for one database is changed to 2, the total number of shares increases to 5. Now, when IORM is required, the database with a share value of 2 is allocated 2 out of every 5 I/Os, while the databases with a share value of 1 are each allocated 1 out of every 5 I/Os.

In addition to prioritizing access to I/O resources, the share value also prioritizes access to Exadata flash storage resources. The available flash storage space is divided up according to the total number of allocated shares, and each database is allocated an amount of space according to its share value. Consequently, databases with a larger share value are given access to proportionally more flash storage space.

Adjusting IORM share values for databases

To adjust the IORM share values for databases:

1. Open the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Locate a database deployment that you want to adjust the IORM share for, and from the associated action menu (☰) select **Update Exadata IORM**.

   The Exadata I/O Resource Management dialog is displayed.

3. In the Exadata I/O Resource Management dialog, use the **Shares** fields to specify the share value for each database deployment on the corresponding Exadata Cloud at Customer instance or VM cluster.

4. When you are satisfied, click **Save** to implement the settings. Alternatively, click **Cancel** to leave the dialog without updating any of the share values.

Implementing a custom IORM policy

In addition to prioritizing between databases, Exadata IORM can manage resources across different workload categories, both within a single database and across multiple databases, by using a custom IORM policy. To implement a custom IORM policy, you must submit a Service Request to Oracle Support. When you submit the Service Request, you must specify the custom IORM policy that you wish to implement by providing the `ALTER IORMPLAN` command to apply to the Exadata Storage Servers. You will be notified through the Service Request when the policy is enabled.

For details about submitting the Service Request see How to Request Service Configuration for Oracle Database Exadata Cloud Service. Also, see the *Oracle Exadata Storage Server Software User's Guide* for details about the `ALTER IORMPLAN` command.

Adding an SSH Public Key

Should the need arise, you can add an SSH public key to your Oracle Database Exadata Cloud at Customer environment. After you add the public key, you can provide the matching private key to connect to a compute node using SSH as either the `opc` or the `oracle` user.

To add an SSH public key:

1. Go to the SSH Access page for a database deployment that is associated with the compute nodes that you want to add a public key to:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. From the action menu (☰) for the database deployment, select **SSH Access**.

      The **Add New Key** overlay is displayed.

2. Specify the new public key using one of the following methods:

   - Select **Upload a new SSH Public Key value** and click **Choose File** to select a file that contains the public key.
   - Select **Key value** and specify the new public key value in the text area. Make sure the value does not contain line breaks or end with a line break.

3. Click **Add New Key**.

> **Note:**
>
> Although you can add an SSH key using the action menu (☰) for a database deployment, every SSH key provides system-wide access to the compute nodes that are associated with the database deployment. You are not required to add an SSH key for every database deployment, and you cannot create a specific association between an SSH key and a database deployment in order to provide isolated access to the database deployment.

Removing an SSH Public Key

Should the need arise, you can remove an SSH public key from your Oracle Database Exadata Cloud at Customer environment. After you remove the public key, you can no longer use the matching private key to connect to a compute node using SSH as either the `opc` or the `oracle` user.

To remove an SSH public key you must edit the `authorized_keys` files for the `opc` and `oracle` users on every compute node in your Exadata Cloud at Customer environment.

> **Note:**
>
> The following describes the procedure for each compute node and must be repeated across your compute nodes.

To remove an SSH public key on a compute node:

1. Connect to the compute node as the **`opc`** user.

   See Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Delete the line containing the SSH public key that you want to remove from the `authorized_keys` files associated with the `opc` user (`/home/opc/.ssh/authorized_keys`) and the `oracle` user (`/home/oracle/.ssh/authorized_keys`).

   > **Caution:**
   >
   > The `authorized_keys` files may contain numerous keys and altering or removing the wrong key may result in a loss of functionality. To minimize the likelihood of an error make a copy of each `authorized_keys` file before making any modification. Also, rather than deleting the line containing the public key that you wish to remove, you can disable the key by tagging it with the `@revoked` marker. For example:
   >
   > ```
   > @revoked ssh-rsa AAAAB5W...
   > ```

4. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Updating the Cloud Tooling on Exadata Cloud at Customer

You can update the cloud-specific tooling included on an Exadata Cloud at Customer compute node by downloading and applying a software package containing the updated tools.

> **Note:**
>
> It is highly recommended to maintain the same version of cloud tooling across your Exadata Cloud at Customer environment.

To check the installed cloud tooling release and check for updates:

1. Connect to the compute node as the **`opc`** user.

   See Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Use the following command to display information about the installed cloud tooling and to list the available updates.

   ```
   # dbaascli patch tools list
   ```

   The command output displays:

- The version of the cloud tooling that is installed on the compute node.
- A list of available updates.
- Notification of the cloud tooling version that is installed on the other compute nodes in the VM cluster.

> **Note:**
>
> If the command fails or does not provide the expected output, then check the `oss_container_url` parameter setting in the configuration file at `/var/opt/oracle/exapatch/exadbcpatch.cfg`.
>
> The expected URL contains the region and subscription identifiers for your Exadata Cloud at Customer instance in the following form:
>
> ```
> https://
> storage.<region_id>.ocm.<subscription_id>.oraclecloudatcustomer.co
> m/v1/Storage-idcs-cloudinfra-<region_id>/dbaas_patch/shome
> ```
>
> For example, the parameter setting may be similar to:
>
> ```
> oss_container_url="https://
> storage.us1.ocm.1234567.oraclecloudatcustomer.com/v1/Storage-idcs-
> cloudinfra-us1/dbaas_patch/shome"
> ```
>
> You can also validate the accessibility of the storage container by using the `curl` command. In this case, use the `oss_container_url` value without the trailing `shome` directory. For example:
>
> ```
> $ curl -v https://
> storage.us1.ocm.1234567.oraclecloudatcustomer.com/v1/Storage-idcs-
> cloudinfra-us1/dbaas_patch
> ```

To update the cloud-specific tooling:

1. Connect to the compute node as the **`opc`** user.

   See Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Download and apply the cloud tooling update:

   - To update to the latest available cloud tooling, use the following command:

     ```
     # dbaascli patch tools apply --patchid LATEST
     ```

- To update to a specific cloud tooling release, use the following command:

  ```
  # dbaascli patch tools apply --patchid patch-id
  ```

  In the preceding command, *patch-id* is a cloud tooling patch identifier, as reported in the output of the `dbaascli patch tools list` command.

Administering VM Clusters

If your Exadata system environment is enabled to support multiple virtual machine (VM) clusters, then you can define up to 8 clusters and specify how the overall Exadata system resources are allocated to them.

Topics

- Viewing Information About Existing VM Clusters
- Creating a VM Cluster
- Modifying an Existing VM Cluster
- Deleting a VM Cluster

Viewing Information About Existing VM Clusters

To view information about existing VM clusters:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and then choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Manage Clusters**.

   The cluster administration page is displayed, which contains two tables:

   - The VM Clusters table displays information about all of the VM clusters that are defined on your Exadata Cloud at Customer system environment. Each row in the table outlines the resource allocation, configuration options, and status for an existing VM cluster.
   - The Resource Overview table outlines:
     - The resources that are allocated to existing VM clusters.
     - The resources that are not allocated to existing VM clusters and are available to be allocated to new VM clusters.
     - The total resources that are contained in your Exadata Cloud at Customer system environment.

Creating a VM Cluster

To create a VM cluster:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and then choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Manage Clusters**.

   The cluster administration page is displayed.

4. In the cluster administration page, click **Create VM Cluster**.

   The Create VM Cluster dialog is displayed.

5. Use the Create VM Cluster dialog to specify the following resource allocations and configuration options for the new VM cluster, factoring in any plans for more VM clusters. Then, click **Create** and confirm your intention in the resulting dialog.

   - **Cluster Name** — enter a name for the VM cluster.
   - **Database backups on Exadata Storage** — check this option to configure the Exadata storage to enable local database backups.

     > **Note:**
     >
     > Take care when setting this option because your choice has a profound effect on the storage allocation and your backup options, which cannot be easily changed. See Exadata Storage Configuration for more information about the effects of each configuration alternative.

   - **Create sparse disk group?** — check this option to create a disk group that is based on sparse grid disks. You must select this option to enable Exadata Cloud at Customer snapshots. Exadata snapshots enable space-efficient clones of Oracle databases that can be created and destroyed very quickly and easily.

     > **Note:**
     >
     > Take care when setting this option because your choice has a profound effect on the storage allocation and your ability to use snapshots, which cannot be easily changed. See Exadata Storage Configuration for more information about the effects of each configuration alternative. See also Creating and Managing Snapshots of a Database Deployment.

   - **Database backups on ZDLRA** — check this option to enable database backups on Oracle Zero Data Loss Recovery Appliance (ZDLRA) storage. If you do not select this option, then you cannot select ZDLRA as a backup location when you configure a database deployment.

- **Exadata Storage (TB)** — specifies the total amount of Exadata storage that is allocated to the VM cluster. This storage is allocated evenly from all of the Exadata Storage Servers. Specify a value greater than 3 TB and up to the amount of remaining unallocated Exadata storage space.
- **Client Network** — specifies the client network subnet that is allocated to the VM cluster. Choose from the list of available subnets. The choices in the list are based on the network definitions that were defined during the provisioning of your Exadata Cloud at Customer system. Your choice also automatically selects the backup network subnet for the VM cluster, which is associated with the selected client network subnet.
- **CPU Cores** — specifies the number of CPU cores that are allocated to each active node in the VM cluster. Specify a value greater than 2 and up to the number of remaining unallocated CPU cores.
- **Memory (GB)** — specifies the amount of memory that is allocated to each active node in the VM cluster. Specify a value greater than 30 GB and up to the amount of remaining unallocated memory, factoring in any plans for more VM clusters.

> **Note:**
>
> Take care when specifying the memory allocation because:
>
> – After the VM cluster is created, you cannot decrease the memory allocation; however, you may increase the memory allocation by using unallocated memory.
>
> – You cannot create another VM cluster unless there is 30 GB of remaining unallocated memory. In that case, you would need to delete an existing VM cluster before you can create another one.

- **DB Home Storage (GB)** — specifies the amount of local disk storage that is allocated to each active node in the VM cluster. Specify a value greater than 60 GB and up to the amount of remaining unallocated local storage space, factoring in any plans for more VM clusters.

> **Note:**
>
> Take care when specifying the local disk storage because:
>
> – In addition to the storage specified in this attribute, each VM cluster requires 137 GB of local disk storage to support software images for the VM cluster. Consequently, the minimum amount of local disk storage consumed by a VM cluster is 197 GB (137 GB + 60 GB).
>
> – For Exadata Cloud at Customer configurations based on Oracle Exadata X7 systems, the total amount of local disk storage that can be allocated to VM clusters is 1237 GB. For Exadata Cloud at Customer configurations based on Oracle Exadata X6 systems, the total amount of local disk storage that can be allocated to VM clusters is 483 GB by default, or up to 1237 GB on systems with upgraded local disk storage.
>
> – After the VM cluster is created, you cannot modify the amount of local storage.
>
> – If all of the local disk storage is allocated, or if there is not at least 197 GB of remaining unallocated local disk storage, then you cannot create another VM cluster. In that case, you must delete an existing VM cluster before you can create another one.

- The check boxes beside each node (`node-1`, `node-2`, and so on) enable you to specify the active compute nodes in the VM cluster. Use the check boxes to specify the nodes that are included in the cluster.

After you complete the Create VM Cluster dialog, a message is displayed indicating that the VM cluster is being created.

Modifying an Existing VM Cluster

> **Note:**
>
> - Before you modify an existing VM cluster you must create a database deployment in the VM cluster. If you attempt to modify a VM cluster before creating any database deployment, then the operation fails.
> - Ensure that you do not modify an existing VM cluster while another modify operation is progressing. If you attempt to perform concurrent VM cluster modifications, then operations may fail.

To modify an existing VM cluster:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and then choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Manage Clusters**.

   The cluster administration page is displayed.

4. Click the action menu (☰) that is associated with the VM cluster and choose **Modify**.

5. Use the Modify VM Cluster dialog to modify the resource allocation for any of the following attributes. You can only modify one attribute at a time. Other attributes are displayed for reference purposes only and cannot be modified. After you make the change, click **Modify** and confirm your intention in the resulting dialog.

   - **CPU Cores per Node** — specifies the number of CPU cores that are allocated to each node in the VM cluster. Specify a value greater than 2 and up to the number of remaining unallocated CPU cores.

     Modifying the number of CPU cores allocated to each node in the VM cluster is an online operation. Compute nodes are not rebooted because of this operation.

     The Oracle Database instance caging feature is enabled by explicitly setting the `CPU_COUNT` database initialization parameter. That setting is not affected by modifying the number of CPU cores that are allocated to the VM cluster. Therefore, if you are using the instance caging feature you must explicitly alter the `CPU_COUNT` setting to match changes to the VM cluster. If `CPU_COUNT` is set to `0` (its default setting), then instance caging is disabled and Oracle Database continuously monitors the number of CPUs reported by the operating system and uses the current count.

   - **Memory (GB) per Node** — specifies the amount of memory (in GB) that is allocated to each node in the VM cluster. Specify a value greater than the current allocation and up to the amount of remaining unallocated memory.

     To process a memory change, the associated compute nodes are rebooted in a rolling manner, one compute node at a time, to minimize impact on the cluster.

     > **Note:**
     >
     > You can only use the Modify VM Cluster dialog to increase the memory allocation.

   - **Exadata Storage (TB)** — specifies the total amount of Exadata storage (in TB) that is allocated to the VM cluster. This storage is allocated evenly from all of the Exadata Storage Servers. Specify a value greater than 3 TB and up to the amount of remaining unallocated Exadata storage space.

     You may reduce the Exadata storage allocation for a VM cluster. However, you must ensure that the new amount covers the existing contents, and you should also allow for anticipated data growth.

     Modifying the Exadata storage allocated to the VM cluster is an online operation. Compute nodes are not rebooted because of this operation.

   - **Add/Remove Nodes** — indicates that you want to add a node to the VM cluster or remove a node from the VM cluster.

After you select this option, use the check boxes beside each node (`node-1`, `node-2`, and so on) to specify the nodes that are included in the cluster. If you check a previously unchecked box, then the corresponding node is added to the VM cluster. If you uncheck a previously checked box, then the corresponding node is removed from the VM cluster.

Node addition and removal are online operations. Unchanged compute nodes are not rebooted because of node addition or removal.

Note the following requirements, recommendations, and restrictions for node addition and removal:

- You must always have at least one node in the VM cluster.
- You can only add or remove one node at a time. To add or remove multiple nodes you must use the Modify VM Cluster dialog multiple times.
- Before you add a new compute node to an existing VM cluster:
  - Update the cloud-specific tooling on the existing compute nodes in the VM cluster. See Updating the Cloud Tooling on Exadata Cloud at Customer.
  - Undo any operating system customization on the existing compute nodes in the VM cluster. The newly added node will use a standard software image with settings that may cause performance or stability issues if added to a customized cluster. Examples of potentially problematic customization include changing the timezone and NTP settings.
  - If you have upgraded the Exadata software release in the cluster, then check with Oracle to ensure that the Exadata software release version in your compute node user domains (DomU) matches the Exadata system software version in the management domains (Dom0). If you attempt to add a node with mismatched software versions, then the operation fails.
  - Ensure that there is at least one running database instance for every existing database deployment on the VM cluster.

    If you do not follow this recommendation and add a compute node to a VM cluster while a database is completely shut down, then you need to run the following command before you can start the database on the added compute node:

    ```
    #  dbaascli database configureinstance --dbname dbname --node
    nodename
    ```

    In the preceding command, *`dbname`* specifies the name of the database and *`nodename`* specifies the hostname of the new compute node. The command must be run as the `root` user on a compute node that hosts a running instance of the database.
  - If the existing VM cluster was created using Exadata Cloud at Customer before release 18.4.6, then ensure that Oracle Grid Infrastructure is correctly configured to add the new compute node:
    a. Connect to a compute node in the cluster as the `opc` user and start a command shell as the `root` user.

       See Connecting to a Compute Node Through Secure Shell (SSH).
    b. Examine the contents of the configuration file at `/var/opt/oracle/creg/grid/grid.ini`:

If the file contains entries for the Oracle Automatic Storage Management (ASM) `SYS` and `ASMSNMP` user passwords, then take note of the passwords and proceed to the next step. If present, the entries look like:

```
sys=password
asmsnmp=password
```

If the file does not contain the ASM `SYS` and `ASMSNMP` user passwords, then look for the `wallet_loc` entry. If the `wallet_loc` entry is not present, then proceed to the next step and supply passwords of your choice when prompted. If the `wallet_loc` entry is present, then your cluster is correctly configured to add new compute nodes and you do not need to proceed to the next step.

c. Run the following command twice, once to set the ASM `SYS` user password and again to set the ASM `ASMSNMP` user password:

```
# dbaascli grid changepasswd --dbname grid
```

When prompted, supply the ASM user name (`SYS` or `ASMSNMP`) and password.

* Ensure that the file permissions are correctly configured on the existing nodes.

On every existing node in the VM cluster:

a. Connect to the compute node as the `opc` user and start a command shell as the `root` user.

See Connecting to a Compute Node Through Secure Shell (SSH).

b. Become the `grid` user:

```
# su - grid
$
```

c. Observe the `ORACLE_HOME` directory location for Oracle Grid Infrastructure. For example:

```
$ echo $ORACLE_HOME
/u01/app/12.2.0.1/grid
```

d. Exit the `grid` user session:

```
$ exit
#
```

e. As the `root` user, set an environment variable that references the `ORACLE_HOME` directory location for Oracle Grid Infrastructure that you observed previously. For example:

```
# export GRID_HOME=/u01/app/12.2.0.1/grid
```

f. Run the following command sequence to set the required file privileges:

```
# chgrp oinstall $GRID_HOME/srvm/admin/logging.properties
# chmod 0644 $GRID_HOME/srvm/admin/logging.properties
# chgrp oinstall $GRID_HOME/bin/oradaemonagent
# chmod 0644 $GRID_HOME/bin/oradaemonagent
# chgrp oinstall $GRID_HOME/bin/osdbagrp0
# chmod 0644 $GRID_HOME/bin/osdbagrp0
# chgrp oinstall $GRID_HOME/bin/rawutl0
# chmod 0644 $GRID_HOME/bin/rawutl0
# chgrp oinstall $GRID_HOME/bin/*.ouibak
# chmod 0644 $GRID_HOME/bin/*.ouibak
```

- If the existing VM cluster uses Exadata software release 19.2, then ensure that host naming is correctly configured on the existing nodes. You can determine the current Exadata software release by executing the `imageinfo` command on any compute node.

  On every existing node in the VM cluster:

  a. Connect to the compute node as the `opc` user and start a command shell as the `root` user.

     See Connecting to a Compute Node Through Secure Shell (SSH).

  b. Edit `/etc/nsswitch.conf` to ensure that the `hosts` entry concludes with `myhostname`.

     For example, the line in `/etc/nsswitch.conf` containing:

     ```
     hosts: files dns
     ```

     changes to:

     ```
     hosts: files dns myhostname
     ```

- After you add a new compute node to an existing VM cluster:
  - Update the cloud-specific tooling to ensure the same version across all of the compute nodes in the VM cluster. See Updating the Cloud Tooling on Exadata Cloud at Customer.
  - Start the required database instances on the new nodes. See Starting a Database Instance.
- Before you remove a compute node, note the following:
  - When you remove a compute node, all database instances on the node are deleted.
  - You cannot remove a node if doing so would delete a database instance that is the only preferred instance for an Oracle Database service.
  - You cannot remove a node while a backup operation is running on it.

Deleting a VM Cluster

To delete an existing VM cluster:

1. Open the My Services dashboard.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the action menu (☰) in the tile that is associated with Exadata Cloud at Customer and then choose **View Details**.

   The Service Details page is displayed, with the Overview tab showing.

3. Locate your service instance in the list. Click the action menu (☰) located beside the service instance name and choose **Manage Clusters**.

   The cluster administration page is displayed.

4. Click the action menu (☰) that is associated with the VM cluster and choose **Delete**. Then, confirm the action in the resulting dialog.

   A message is displayed indicating that the cluster is being deleted.

Administering Oracle Homes

An Oracle Home is a directory location on the compute nodes that contains Oracle Database binaries. Oracle Database Exadata Cloud at Customer enables multiple database deployments to share a set of Oracle Database binaries in a shared Oracle Home directory location.

Topics

- Viewing Information About Oracle Homes
- Moving a Database to Another Oracle Home
- Creating an Oracle Home
- Deleting an Oracle Home

Viewing Information About Oracle Homes

You can view information about Oracle Home directory locations by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in Other Ways to View Information About Oracle Homes at the end of this topic.

Viewing Information About Oracle Homes by Using the Oracle Database Cloud Service Console

1. Open the Instances page of the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the database deployment for which you want to view Oracle Home information.

   The Oracle Database Cloud Service Overview page is displayed.

3. Click Show more... in the Instance Overview section to reveal the Oracle Home Name that is associated with the database deployment.

Other Ways to View Information About Oracle Homes

- You can use the dbaascli utility. See Viewing Information About Oracle Homes by Using the dbaascli Utility.

Viewing Information About Oracle Homes by Using the dbaascli Utility

You can view information about Oracle Home directory locations by using the dbhome info subcommand of the dbaascli utility as follows:

1. Connect to a compute node as the **opc** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Execute the dbaascli command with the dbhome info subcommand:

   ```
   # dbaascli dbhome info
   ```

4. When prompted, press Enter to view information about all Oracle Homes registered in your Exadata Cloud at Customer environment, or specify an Oracle Home name to view information only about that Oracle Home.

5. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Moving a Database to Another Oracle Home

Moving a database to another Oracle Home enables you to consolidate existing Oracle Homes and manage the storage that they consume. You can move a database to another Oracle Home by using the database move subcommand of the dbaascli utility as follows:

1. Connect to a compute node as the **opc** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Ensure that all of the database instances associated with the database deployment are up and running.

   ```
   # dbaascli database status --dbname dbname
   ```

   In the preceding command, *dbname* specifies the name of the database that you want to check.

   Restart any database instances that are not running and open.

4. Execute the dbaascli command with the database move subcommand:

   ```
   # dbaascli database move --dbname dbname --ohome oracle_home
   ```

In the preceding command:

- *dbname* — specifies the name of the database that you want to move.
- *oracle_home* — specifies the path to an existing Oracle Home directory location, which you want the specified database to use.

When performing a move operation to an Oracle Home with a different patch level, if the database is part of an Oracle Data Guard configuration, then ensure that you move the standby database to the new patchset before you move the primary database.

5. Exit the root-user command shell:

```
# exit
$
```

Creating an Oracle Home

You can create a new Oracle Home directory location and software installation, without creating a new database deployment, by using the `dbhome create` subcommand of the `dbaascli` utility as follows:

1. Connect to a compute node as the **opc** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Run the `dbaascli` command with the `dbhome create` subcommand:

```
# dbaascli dbhome create --version software_version
```

   In the preceding command, *software_version* specifies an Oracle Database software version. For example, `11204`, `12102`, `12201`, or `18000`. The latest available bundle patch for the specified software version is automatically used.

   To see information about Oracle Database software images that are available in your Exadata Cloud at Customer environment, including software version and bundle patch details, use the dbaascli dbimage list command.

   When prompted, type `yes` to confirm that the installation is based on a local software image.

4. Exit the root-user command shell:

```
# exit
$
```

Deleting an Oracle Home

If an Oracle Home directory does not support any database deployments, you can delete it by using the `dbhome purge` subcommand of the `dbaascli` utility as follows:

1. Connect to a compute node as the **opc** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Execute the `dbaascli` command with the `dbhome purge` subcommand:

   ```
   # dbaascli dbhome purge
   ```

4. When prompted, enter:
   - `1` — if you want to specify the Oracle Home name for the location being purged.
   - `2` — if you want to specify the Oracle Home directory path for the location being purged.

5. When next prompted, enter the Oracle Home name or directory path for the location being purged.

   If your entries are valid and the Oracle Home is not associated with a database deployment, then the Oracle binaries are removed from the Oracle Home directory location and the associated metadata is removed from the system.

6. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Administering Software Images

Oracle maintains a library of cloud software images and provides capabilities to view the library and download images to your Oracle Database Exadata Cloud at Customer instance. Using these facilities, you can control the version of Oracle binaries that is used when a new set of Oracle binaries is installed.

When you create a new database deployment with a new Oracle Home directory location, the Oracle Database binaries are sourced from a software image that is stored in your Exadata Cloud at Customer instance. Over time, the software images in your Exadata Cloud at Customer instance will become old if they are not maintained. Using an old software image means that you need to apply patches to newly installed binaries to bring them up to date, which is unnecessarily laborious and possibly prone to error.

Software image administration uses the `dbaascli` utility, which is part of the cloud-specific tooling included in Exadata Cloud at Customer. To use the latest enhancements, you should update to the latest version of the cloud tools. See Updating the Cloud Tooling on Exadata Cloud at Customer.

> **Note:**
>
> If you create a new database deployment in an existing Oracle Home directory location, the software version in the existing Oracle Home directory location must not be older than the current default software image. If the current default software image is newer than the software version in the existing Oracle Home directory location, then the database creation will fail.
>
> To avoid this situation, you can use any of the following options:
>
> - Before you create the new database deployment, patch the existing Oracle Home directory location to bring it up to the release level of the default software image.
> - Before you create the new database deployment, activate the software image that matches the release level of the software in the existing Oracle Home directory location. Then, after creating the database deployment, re-activate the newer software image, if required.
> - Create the new database deployment in a new Oracle Home directory location, which will use the default software image.

Topics

- Viewing Information About Downloaded Software Images
- Viewing Information About Available Software Images
- Downloading a Software Image
- Activating a Software Image
- Deleting a Software Image

Viewing Information About Downloaded Software Images

You can view information about Oracle Database software images that are downloaded to your Exadata Cloud at Customer environment by using the `dbimage list` subcommand of the `dbaascli` utility as follows:

1. Connect to a compute node as the **`opc`** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Execute the `dbaascli` command with the `dbimage list` subcommand:

   ```
   # dbaascli dbimage list
   ```

   The command displays a list of software images that are downloaded to your Exadata Cloud at Customer environment, including version and bundle patch information.

4. Exit the root-user command shell:

```
# exit
$
```

Viewing Information About Available Software Images

You can view information about Oracle Database software images that are available to download to your Exadata Cloud at Customer environment by using the `cswlib list` subcommand of the `dbaascli` utility as follows:

1. Connect to a compute node as the **`opc`** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Execute the `dbaascli` command with the `cswlib list` subcommand:

   ```
   # dbaascli cswlib list [ --oss_uri download_location ]
   ```

   The command displays a list of available software images, including version and bundle patch information that you can use to download the software image.

   The `--oss_uri` option can be used to specify the location of the software image library. Normally, the `--oss_uri` option is not required because the location of the software image library is automatically derived from configuration information in the Exadata Cloud at Customer environment. However, if you experience any difficulties with the automatic location, you can lodge a service request (SR) with Oracle Support and they may instruct you to use this option to specify an alternate location. The *`download_location`* is specified as a uniform resource identifier (URI) to a cloud storage container that contains available software images.

4. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Downloading a Software Image

You can download available software images and make them available in your Exadata Cloud at Customer environment by using the `cswlib download` subcommand of the `dbaascli` utility as follows:

1. Connect to a compute node as the **`opc`** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Execute the `dbaascli` command with the `cswlib download` subcommand:

   ```
   # dbaascli cswlib download --version software_version --bp software_bp [--
   bp_update ( yes | no )] [--cdb ( yes | no )] [--oss_uri download_location]
   ```

In the preceding command:

- *software_version* — specifies an Oracle Database software version. For example, 11204, 12102, 12201, 18000, 19000.
- *software_bp* — identifies a bundle patch release. For example, APR2018, JAN2019, OCT2019, and so on.
- --bp_update — optionally indicates whether the downloaded software image becomes the current default software image. Default is no.
- --cdb — optionally specifies whether the downloaded software image supports the Oracle multitenant architecture. Default is yes. If you specify --cdb no, then a separate software image is downloaded that contains binaries to support non-container databases (non-CDB).
- *download_location* — optionally specifies the location of the software image library. The location is specified as a uniform resource identifier (URI) to a cloud storage container that contains available software images.

  Normally, the --oss_uri option is not required because the location of the software image library is automatically derived from configuration information in the Exadata Cloud at Customer environment. However, if you experience any difficulties with the automatic location, you can lodge a service request (SR) with Oracle Support and they may instruct you to use this option to specify an alternate location.

4. Exit the root-user command shell:

```
# exit
$
```

Activating a Software Image

You can use the following procedure to activate a specific software image, making it the default software image for the corresponding software release in your Exadata Cloud at Customer environment:

1. Connect to a compute node as the **opc** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Execute the dbaascli command with the dbimage activateBP subcommand:

```
# dbaascli dbimage activateBP --version software_version --bp software_bp [--
cdb ( yes | no )]
```

   In the preceding command:

   - *software_version* — specifies the Oracle Database software version. For example, 11204, 12102, 12201, 18000, 19000.
   - *software_bp* — identifies the bundle patch release. For example, APR2018, JAN2019, OCT2019, and so on.

- `--cdb` — optionally specifies whether to activate a software image that supports the Oracle multitenant architecture. Default is `yes`. If you specify `--cdb no`, then the command acts on the software image that contains binaries to support non-container databases (non-CDB).

   The command fails and outputs an error message if the specified software image is not already downloaded to your Exadata Cloud at Customer environment.

4. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Deleting a Software Image

You can use the following procedure to delete a software image from your Exadata Cloud at Customer environment:

1. Connect to a compute node as the **`opc`** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Execute the `dbaascli` command with the `dbimage purge` subcommand:

   ```
   # dbaascli dbimage purge --version software_version --bp software_bp [--cdb ( yes
   | no )]
   ```

   In the preceding command:

   - *`software_version`* — specifies the Oracle Database software version. For example, `11204`, `12102`, `12201`, `18000`, `19000`.
   - *`software_bp`* — identifies the bundle patch release. For example, `APR2018`, `JAN2019`, `OCT2019`, and so on.
   - `--cdb` — optionally specifies whether to remove the software image that supports the Oracle multitenant architecture. Default is `yes`. If you specify `--cdb no`, then the software image that contains binaries to support non-container databases (non-CDB) is removed.

   If the command will remove a software image that is not currently available in the software image library, and therefore cannot be downloaded again, then the command pauses and prompts for confirmation.

   You cannot remove the current default software image for any software version. To avoid this restriction, you must make another software image the current default.

4. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Administering a Data Guard Configuration

Oracle Database Exadata Cloud at Customer provides several commands and features to simplify the administration of database deployments that contain an Oracle Data Guard configuration.

Topics

- Performing a Switchover Operation
- Performing a Manual Failover Operation
- Reinstating a Failed Primary Database
- Configuring Clients for Automatic Failover

Performing a Switchover Operation

You can perform a switchover to the standby database in your Oracle Data Guard configuration by using the Oracle Database Cloud Service console.

A switchover operation enables the primary database to switch roles with the standby database. There is no data loss during a switchover. After a switchover, each database continues to participate in the Oracle Data Guard configuration in its new role. A switchover is typically used to reduce primary database downtime during planned outages, such as operating system or hardware upgrades, or rolling upgrades of the Oracle Database software and patch sets. For more information, see "Switchovers" in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1 or 11.2.

Performing a Switchover Operation by Using the Oracle Database Cloud Service Console

1. Go to the Overview page for the database deployment you want to perform a switchover on:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. In the list of deployments, click the name of the database deployment you want to perform the switchover on.

      The Oracle Database Cloud Service Overview page is displayed.

2. To ensure the Overview page reflects the current role of each database, click the Refresh Configuration icon.

3. From the action menu (☰) located beside the deployment name or beside any of the compute nodes, select **Switchover**, and then confirm the action.

   The deployment shows a status of Maintenance in the Oracle Database Cloud Service console until the switchover is complete.

4. Refresh the page occasionally.

   Database Role will be updated to reflect the new role for each database.

Performing a Manual Failover Operation

You can perform a manual failover to the standby database in your Oracle Data Guard configuration by using the Oracle Database Cloud Service console.

A failover operation changes the standby database to the primary role in response to a primary database failure. If the primary database was not operating in either maximum protection mode or maximum availability mode before the failure, some data loss may occur. If Flashback Database is enabled on the primary database, it can be reinstated as a standby for the new primary database once the reason for the failure is corrected. A failover is typically used only when the primary database becomes unavailable, and there is no possibility of restoring it to service within a reasonable period of time. For more information, see "Failovers" in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1 or 11.2.

Performing a Manual Failover Operation by Using the Oracle Database Cloud Service Console

1. Go to the Overview page for the database deployment you want to perform the failover on:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. In the list of deployments, click the name of the database deployment you want to perform the failover on.

      The Oracle Database Cloud Service Overview page is displayed.

2. To ensure the Overview page reflects the current role of each database, click the Refresh Configuration icon.

3. From the action menu (☰) located beside the deployment name or beside any of the compute nodes, select **Failover**, and then confirm the action.

   The deployment shows a status of Maintenance in the Oracle Database Cloud Service console until the operation is complete.

4. Refresh the page occasionally.

   Database Role will be updated to reflect the new role for each database.

Reinstating a Failed Primary Database

You can reinstate a failed primary database after a failover by using the Oracle Database Cloud Service console.

After performing a failover to the standby database, you may be able to restore your original disaster-recovery solution by reinstating the failed primary database. You can use the Data Guard broker's reinstate capability to make the failed primary database a viable standby database for the new primary. For more information, see "Reenabling Disabled Databases After a Role Change" in *Oracle Data Guard Broker* for Release 18, 12.2, 12.1 or 11.2.

Reinstating a Failed Primary Database by Using the Oracle Database Cloud Service Console

1. Go to the Overview page for the database deployment you want to perform the reinstate on:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. In the list of deployments, click the name of the database deployment you want to perform the reinstate on.

      The Oracle Database Cloud Service Overview page is displayed.

2. To ensure the Overview page reflects the current role of each database, click the Refresh Configuration icon.

3. From the action menu (☰) located beside the deployment name or beside any of the primary database's compute nodes, select **Reinstate**, and then confirm the action.

   The deployment has a status of Maintenance in the Oracle Database Cloud Service console until the operation is complete.

4. Refresh the page occasionally.

   Database Role will be updated to reflect the new role for each database.

Configuring Clients for Automatic Failover

By using pre-defined network service names, application clients can automatically reconnect to a new primary database following a role transition.

Your Data Guard configuration on Oracle Database Exadata Cloud at Customer is pre-configured to provide automatic transition of application connections from a failed primary database to a new primary database after a Data Guard role transition has taken place.

The following network service names are pre-defined:

- *dbname*_dg: This service is used to connect to the primary database. If the database uses Oracle Database 12c Release 1, or later, this service connects to the root container.
- *dbname*_dg_ro: This service is used to connect to the standby database. If the database uses Oracle Database 12c Release 1, or later, this service connects to the root container.
- *PDBname*_dg: In an Oracle Data Guard configuration using Oracle Databases 12c Release 1, or later, this service is defined and is used to connect to the default PDB of the primary database.
- *PDBname*_dg_ro: In an Oracle Data Guard configuration using Oracle Databases 12c Release 1, or later, this service is defined and is used to connect to the default PDB of the standby database.

The services are managed on each database through the use of pre-defined triggers. Following a role transition, the trigger is fired to start the services on the new primary database. By using the pre-defined network service names in your application connections, your application clients will be automatically directed to the new primary database following a role transition.

See Client Failover Best Practices for Highly Available Oracle Databases: Oracle Database 11g Release 2 or Client Failover Best Practices for Highly Available Oracle Databases Oracle Database 12c for detailed information.

Administering Pluggable Databases

Oracle Database Exadata Cloud at Customer provides APIs to administer Oracle Multitenant pluggable databases (PDBs).

The PDB administration APIs are provided using the `dbaascli` utility, which is part of the cloud-specific tooling included in Exadata Cloud at Customer. The APIs can be used interactively or programmatically, and provide a simple mechanism for performing PDB lifecycle management operations. The APIs are specifically customized for Exadata Cloud at Customer and operate in conjunction with the database configuration standards that are implemented in Exadata Cloud at Customer.

The PDB administration APIs only support databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

The following table outlines the supported PDB lifecycle management operations:

PDB Lifecycle Operation	**`dbaascli` Command**
Create a new PDB.	dbaascli pdb create
Delete a PDB.	dbaascli pdb delete
Create a new PDB as a clone of an existing PDB in the same container database.	dbaascli pdb local_clone
Create a new PDB as a clone of an existing PDB in another container database.	dbaascli pdb remote_clone
Open a PDB.	dbaascli pdb open
Close a PDB.	dbaascli pdb close
Rename a PDB.	dbaascli pdb rename
Modify the size limits for a PDB.	dbaascli pdb resize
Display information about a container database.	dbaascli pdb checkdb
Display status information about a PDB.	dbaascli pdb checkpdb
Display detailed information about a PDB.	dbaascli pdb info
Display status information about PDBs that are associated with a specific container database and a specific compute node.	dbaascli pdb checknode
Display Oracle Net connect string information for a PDB.	dbaascli pdb connect_string
Return network connection information for a PDB.	dbaascli pdb connect_info

Registering a Database

Database registration enables you to use the Oracle Database Exadata Cloud at Customer console and cloud tooling to administer a database that was not created using the Exadata Cloud at Customer console.

Exadata Cloud at Customer includes APIs for database registration that support two main use-cases:

- You can use the database registration APIs to register a database that you manually created on Exadata Cloud at Customer, using the Oracle Database Configuration Assistant (DBCA) for example.
- You can use database registration as part of the process to migrate an existing database from another platform to Exadata Cloud at Customer.

In both cases, after the database is registered you can use the Oracle Database Exadata Cloud at Customer console and cloud tooling to perform various administration functions, such as backup, recovery, and patching.

Database registration uses the `dbaascli` utility, which is part of the cloud-specific tooling included in Exadata Cloud at Customer. To use the latest enhancements, you should update to the latest version of the cloud tools. See Updating the Cloud Tooling on Exadata Cloud at Customer.

Registering a Replacement Database

You can register a replacement database by using the following procedure:

1. Create a database deployment using the Oracle Database Exadata Cloud at Customer console or REST API. See Creating a Database Deployment.

   Configure the database deployment as follows:

   - Set the DB Name attribute to match the database name of the replacement database.
   - Set the Software Release attribute to match the Oracle Database software release used by the replacement database.
   - Set the Character Set attribute to match the character set of the replacement database.
2. Shut down the newly created database.
3. Remove the database files and replace them with the replacement database. The replacement database files must be moved into the same location as the database you are replaced. You can manually copy the replacement database files into place from a cold backup, or you can restore from a backup using Oracle Recovery Manager (RMAN).
4. Start the replacement database.
5. Perform the registration prerequisite check and resolve any issues that it uncovers.

   Run the following command as the `root` user on one of the database servers:

   ```
   # dbaascli registerdb prereqs --dbname dbname [ --db_unique_name
   dbuniquename ]
   ```

In the preceding command:

- *dbname* specifies the database name (`DB_NAME`) of the replacement database.
- *dbuniquename* optionally specifies the globally unique name (`DB_UNIQUE_NAME`) of the replacement database. You only need to specify this parameter of the unique name differs from the database name.

The prerequisite check uncovers issues with the replacement database that you must resolve before successful registration. The following list outlines common issues:

- Required database attributes such as the database name, software version, or character set do not match the original database.
- The database is not in `READ WRITE` mode.
- Archive log mode and block change tracking are not enabled.
- Transparent Database Encryption (TDE) is not configured, or is not properly configured. For example, the TDE keystore is not the expected location, or the TDE keystore type is not `AUTOLOGIN`.

6. Register the database by running the following command as the `root` user on one of the database servers:

   ```
   # dbaascli registerdb begin --dbname dbname [ --db_unique_name
   dbuniquename ]
   ```

   In the preceding command, specify the same values for *dbname* and *dbuniquename* that you used in the prerequisite check.

Registering a Manually Created or Migrated Database

You can register a manually created or migrated database by using the following procedure:

1. Manually create or migrate the database. Ensure that the database meets the following requirements:
   - The `/etc/oratab` file contains an entry for the database.
   - Oracle Transparent Database Encryption (TDE) is configured on the database.
   - The database is in `READ WRITE` mode.
   - Archive log mode and block change tracking are enabled.
2. Generate a cloud registration file (`creg.ini`) by running the following command as the `root` user on one of the database servers:

   ```
   # dbaascli registerdb registry --action initialize --db dbname [ --udb
   dbuniquename ]
   ```

   In the preceding command:

   - *dbname* specifies the database name (`DB_NAME`) of the database that you are registering.
   - *dbuniquename* optionally specifies the globally unique name (`DB_UNIQUE_NAME`) of the database. You only need to specify this parameter of the unique name differs from the database name.
3. Perform the registration prerequisite check and resolve any issues that it uncovers.

Run the following command as the root user on one of the database servers:

```
# dbaascli registerdb prereqs --dbname dbname [ --db_unique_name
dbuniquename ]
```

In the preceding command, specify the same values for *dbname* and *dbuniquename* that you used in the preceding step.

The prerequisite check uncovers issues with the replacement database that you must resolve before successful registration.

4. Register the database by running the following command as the root user on one of the database servers:

   ```
   # dbaascli registerdb begin --dbname dbname [ --db_unique_name
   dbuniquename ]
   ```

   In the preceding command, specify the same values for *dbname* and *dbuniquename* that you used in the preceding steps.

5. Create a database deployment using the Oracle Database Exadata Cloud at Customer console or REST API. See Creating a Database Deployment.

   Configure the database deployment as follows:

   - Set the DB Name attribute to match the database name of the database that you are registering.
   - Set the Software Release attribute to match the Oracle Database software release used by the database.

Maintaining the Manageability of Exadata Cloud at Customer

The following best practices ensure that your Oracle Database Exadata Cloud at Customer instances stay manageable.

To keep your Exadata Cloud at Customer instances manageable, follow these guidelines:

- Wherever possible, use Oracle-supplied cloud interfaces (Web consoles, cloud-specific tools, or REST APIs) to perform lifecycle management and administrative operations. For example, you should use the Oracle Database Cloud Service console to create databases instead of manually running the Oracle Database Configuration Assistant (DBCA), and you should use the Oracle Database Cloud Service console or the exadbcpatchmulti command to apply Oracle Database patches instead of manually running opatch.
- Do not change the compute node OS user accounts for the opc and oracle users.
- Do not alter the SSH keys that are used to automate Exadata Cloud at Customer. There are three Exadata Cloud at Customer automation keys, which are stored in one or more of the following files:
  - /root/.ssh/authorized_keys
  - /home/opc/.ssh/authorized_keys

  - `/home/oracle/.ssh/authorized_keys`

  You can identify the Exadata Cloud at Customer automation keys by using the following key comments:

  - `PUBLIC_KEY` identifies a key that is associated with Database Cloud Services automation.
  - `OEDA_PUB` identifies a key that is associated with the Oracle Exadata Deployment Assistant.
  - `EXACLOUD_KEY` identifies a key that is associated with Exadata Cloud at Customer infrastructure automation.

- Apply the quarterly Patch Set Updates (PSUs) regularly, every quarter if possible.
- Do not change the ports for Oracle Net Listener, or Enterprise Manager.

Loading Data into the Oracle Database on Exadata Cloud at Customer

You load data into an Oracle database on Oracle Database Exadata Cloud at Customer using the same tools you would use for an Oracle database on another system.

The following sections outline several common tools and techniques used to load data into an Oracle database. Also, see Migrating Oracle Databases to Exadata Cloud at Customer for additional techniques and more specific information about migrating existing Oracle databases to Exadata Cloud at Customer.

Using SQL*Loader to Load Data into the Database

SQL*Loader is a high-speed data loading utility that loads data from external files into tables in an Oracle database. SQL*Loader accepts input data in a variety of formats, can perform filtering, and can load data into multiple Oracle database tables during the same load session. SQL*Loader provides three methods for loading data: Conventional Path Load, Direct Path Load, and External Table Load.

For information, see "SQL Loader" in *Oracle Database Utilities* for Release 18, 12.2, 12.1 or 11.2.

Using Oracle Data Pump Import to Load Data into the Database

Oracle Data Pump is an Oracle Database feature that offers very fast bulk data and metadata movement between Oracle databases. Oracle Data Pump provides two high-speed, parallel utilities: Export (expdp) and Import (impdp). Data Pump automatically manages multiple, parallel streams for maximum throughput of unload and load operations. The degree of parallelism can be adjusted on-the-fly.

For information, see "Data Pump Import" in *Oracle Database Utilities* for Release 18, 12.2, 12.1 or 11.2.

Using Transportable Tablespaces to Load Data into the Database

Transportable Tablespaces is an Oracle Database feature that copies a set of tablespaces from one Oracle database to another. Moving data using transportable tablespaces can be much more efficient than performing either an export/import or unload/load of the same data. This is because the tablespace datafiles are copied to the destination location, which avoids the cost of formatting the data into Oracle blocks. Also, in some circumstances, your

Transportable Tablespace can contain previously encrypted or compressed data, which avoids the cost of decrypting and re-encrypting, or expanding and re-compressing the data.

For information, see "Transporting Tablespaces Between Databases" in *Oracle Database Administrator's Guide* for Release 18, 12.2, 12.1 or 11.2.

Using Pluggable Databases (PDBs) to Load Data into the Database

The multitenant architecture of Oracle Database 12c and later releases supports the moving of a pluggable database (PDB) from one container database (CDB) to another. This capability makes it easy to load data into Exadata Cloud at Customer, provided that the source data is already inside a PDB on Oracle Database 12c or a later release.

For information about PDBs and how to unplug, move, and plug them, see "Overview of Configuring and Managing a Multitenant Environment" in *Oracle Multitenant Administrator’s Guide* for Release 18 or "Overview of Managing a Multitenant Environment" in *Oracle Database Administrator's Guide* for Release 12.2 or 12.1.

Tuning Oracle Database Performance on Exadata Cloud at Customer

You tune the performance of Oracle Database on Oracle Database Exadata Cloud at Customer using the same tools you would use for an Oracle database running on any system in your data center. Exadata Cloud at Customer does not place any special restrictions on performance tuning.

The *Oracle Database Performance Tuning Guide* for Release 18, 12.2, 12.1 or 11.2 provides extensive information about how to use Oracle Database performance tools to optimize database performance. It also describes performance best practices and includes performance-related reference information.

Additionally, the Enterprise Manager Tuning and Performance option packs are included in all Exadata Cloud at Customer database deployments. These option packs provide several utilities to assist in maintaining performance and identifying and correcting performance issues.

If your performance tuning activities indicate that you need more computing power or more storage, you can scale Exadata Cloud at Customer to satisfy the need. See Scaling an Exadata Cloud at Customer Instance.

Monitoring and Managing Oracle Database on Exadata Cloud at Customer

To monitor and manage the Oracle database deployed on Oracle Database Exadata Cloud at Customer, you can use the standard management tool provided with the version of the database:

- For Oracle Database 18c, or later, use Enterprise Manager Database Express. See Accessing Enterprise Manager Database Express 18c, or later.
- For Oracle Database 12c, use Enterprise Manager Database Express 12c. See Accessing Enterprise Manager Database Express 12c.

- For Oracle Database 11g, use Enterprise Manager 11g Database Control. See Accessing Enterprise Manager 11g Database Control.

> **Note:**
>
> See Maintaining the Manageability of Exadata Cloud at Customer .

Monitoring and Managing Exadata Storage Servers on Exadata Cloud at Customer

You can use the ExaCLI utility to perform monitoring and management functions on the Exadata Storage Servers that are associated with your Oracle Database Exadata Cloud at Customer instance.

ExaCLI is a command-line administration tool that runs on Exadata compute nodes and storage server nodes and enables you to manage other nodes remotely. On Exadata Cloud at Customer, ExaCLI is configured on the compute nodes to enable access to a set of essential Exadata Storage Server monitoring and management functions, which can be performed without direct administrative access to the Exadata Storage Servers. See Using the ExaCLI Utility.

The ExaCLI utility is located on each Exadata Cloud at Customer compute node at `/usr/local/sbin/exacli`. To use the ExaCLI utility you require:

- The ExaCLI command, which specifies the operation that you want to perform.

  For Exadata Storage Server targets, ExaCLI supports the same command syntax as CellCLI. See CellCLI Command Reference. However, note that ExaCLI on Exadata Cloud at Customer only supports a specific set of monitoring and management operations. For a full list of the supported operations, see ExaCLI Commands.
- The username and password to connect to the Exadata Storage Server.

  On Exadata Cloud at Customer, the preconfigured user is `cloud_user_`*`clustername`*, where *`clustername`* is the name of the virtual machine (VM) cluster that is being used. You can determine the name of the VM cluster by running the following command as the `grid` user on any cluster node:

  ```
  $ crsctl get cluster name
  ```

  The password for `cloud_user_`*`clustername`* is initially set to the administration password that you specify when creating the starter database deployment on the VM cluster.
- The network address of the Exadata Storage Server that is the target of the operation.

  You can determine the IP addresses for your Exadata Storage Servers by examining the `/etc/oracle/cell/network-config/cellip.ora` file on any Exadata Cloud at Customer compute node.

Managing the Log and Diagnostic Files on Exadata Cloud at Customer

The software components in Oracle Database Exadata Cloud at Customer generate a variety of log and diagnostic files, and not all these files are automatically archived and purged. Thus, managing the identification and removal of these files to avoid running out of file storage space is an important administrative task.

Database deployments on Exadata Cloud at Customer include the `cleandblogs` script to simplify this administrative task. The script runs daily as a `cron` job on each compute node to archive key files and remove old log and diagnostic files.

The `cleandblogs` script operates by using the `adrci` (Automatic Diagnostic Repository [ADR] Command Interpreter) tool to identify and purge target diagnostic folders and files for each Oracle Home listed in `/etc/oratab`. It also targets Oracle Net Listener logs, audit files and core dumps.

On Exadata Cloud at Customer, the script is run separately as the `oracle` user to clean log and diagnostic files that are associated with Oracle Database, and as the `grid` user to clean log and diagnostic files that are associated with Oracle Grid Infrastructure.

The `cleandblogs` script uses a configuration file to determine how long to retain each type of log or diagnostic file. You can edit the file to change the default retention periods. The file is located at `/var/opt/oracle/cleandb/cleandblogs.cfg` on each compute node.

The following table lists the parameters that appear in the `cleandblogs.cfg` file, providing a description and the default retention period in days for each file type.

Parameter	Description and Default Value
`AlertRetention`	Alert log (`alert_*instance*.log`) retention value in days. Default value: 14
`ListenerRetention`	Listener log (`listener.log`) retention value in days. Default value: 14
`AuditRetentionDB`	Database audit (`*.aud`) retention value in days. Default value: 1
`CoreRetention`	Core dump/files (`*.cmdp*`) retention value in days. Default value: 7
`TraceRetention`	Trace file (`*.tr*` and `*.prf`) retention value in days. Default value: 7
`longpRetention`	Data designated in the Automatic Diagnostic Repository (ADR) as having a long life (the `LONGP_POLICY` attribute). For information about ADR, see "Automatic Diagnostic Repository (ADR)" in *Oracle Database Administrator's Guide* for Release 18, 12.2, 12.1 or 11.2. Default value: 14

Parameter	Description and Default Value
`shortpRetention`	Data designated in the Automatic Diagnostic Repository (ADR) as having a short life (the `SHORTP_POLICY` attribute). For information about ADR, see "Automatic Diagnostic Repository (ADR)" in *Oracle Database Administrator's Guide* for Release 18, 12.2, 12.1 or 11.2. Default value: 7
`LogRetention`	Log file retention in days for files under `/var/opt/oracle/log` and log files in ACFS under `/var/opt/oracle/dbaas_acfs/log`. Default value: 14
`LogDirRetention`	cleandblogs log file retention in days. Default value: 14
`ScratchRetention`	Temporary file retention in days for files under `/scratch`. Default value: 7

Archiving Alert Logs and Listener Logs

When cleaning up alert and listener logs, `cleandblogs` first archives and compresses the logs, operating as follows:

1. The current log file is copied to an archive file that ends with a date stamp.
2. The current log file is emptied.
3. The archive file is compressed using `gzip`.
4. Any existing compressed archive files older than the retention period are deleted.

Running the cleandblogs Script Manually

The `cleandblogs` script automatically runs daily on each compute node, but you can also run the script manually if the need arises.

1. Connect to the compute node as the `oracle` user to clean log and diagnostic files that are associated with Oracle Database, or connect as the `grid` user to clean log and diagnostic files that are associated with Oracle Grid Infrastructure.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Change to the directory containing the `cleandblogs` script:

   ```
   $ cd /var/opt/oracle/cleandb
   ```

3. Run the `cleandblogs` script:

   ```
   $ ./cleandblogs.pl
   ```

   When running the script manually, you can specify an alternate configuration file to use instead of `cleandblogs.cfg` by using the `--pfile` option:

   ```
   $ ./cleandblogs.pl --pfile config-file-name
   ```

4. Close your connection to the compute node:

   ```
   $ exit
   ```

6

Backing Up and Restoring Databases on Exadata Cloud at Customer

This section explains how to back up and restore Oracle databases on Oracle Database Exadata Cloud at Customer.

Topics

- About Backing Up Database Deployments on Exadata Cloud at Customer
- Viewing Detailed Backup Configuration Information
- Customizing the Automatic Backup Configuration
- Creating an On-Demand Backup
- Deleting a Backup
- Updating the Password for Backing Up to Cloud Storage
- Changing to a Different Backup Destination
- Disabling and Re-enabling Scheduled Backups
- Validating Backups
- Restoring from the Most Recent Backup
- Restoring from a Specific Backup
- Restoring to a Specific Point in Time
- Restoring System and Database Configuration Files
- Recover Backups Using the dbaasapi Utility
- Manually Restoring from a Backup
- View Details About Automatic Backups

About Backing Up Database Deployments on Exadata Cloud at Customer

By backing up your Oracle Database Exadata Cloud at Customer database deployments, you can protect against data loss if a failure occurs.

About Automatic Database Backups

Exadata Cloud at Customer provides a backup feature that automatically backs up the Oracle database associated with a database deployment. This feature is built over Oracle Recovery Manager (RMAN) and exposed through a simple set of system utilities that are installed on your Exadata system. It also relies on Oracle Database Backup Cloud Service, which in turn uses an Oracle Storage Cloud Service container, when cloud storage is selected as a backup location.

When you create a database deployment on Exadata Cloud at Customer, you must choose from the following automatic backup configuration options:

- **Both Cloud Storage and Exadata Storage** — enables two separate backup sets containing periodic full (RMAN level 0) backups and daily incremental backups. The backup to cloud storage uses an Oracle Storage Cloud container, with a seven day cycle between full backups and an overall retention period of 30 days. The backup to Exadata storage uses space in the RECO disk group, with a seven day cycle between full backups and a seven day retention period.

  > **Note:**
  >
  > This option is only available if you provisioned for database backups on Exadata storage. See Exadata Storage Configuration.

- **Cloud Storage Only** — uses an Oracle Storage Cloud container to store periodic full (RMAN level 0) backups and daily incremental backups, with a seven day cycle between full backups and an overall retention period of 30 days.

  > **Note:**
  >
  > To eliminate possible confusion or contention, do not use a single cloud storage container as a backup destination for multiple database deployments.

- **ZDLRA Storage Only** — uses the Recovery Appliance to store one full (RMAN level 0) backup and daily incremental (RMAN level 1) backups. The Recovery Appliance creates virtual full backups from each daily incremental and validates those backups to ensure that they are always recoverable.

  > **Note:**
  >
  > This option is only available if you configured your Exadata Cloud at Customer environment to enable database backups on ZDLRA. See Creating an Exadata Cloud at Customer Instance.

- **None** — no automatic backups are configured.

Default Automatic Database Backup Configuration

The default automatic backup configuration follows a set of Oracle best-practice guidelines:

- Automatic backups are scheduled daily.
- Backups consist of periodic full backups of the database, followed by daily incremental backups:
  - For backups to **Both Cloud Storage and Exadata Storage** or **Cloud Storage Only**, the default interval between full backups is seven days.
  - For backups to **ZDLRA Storage Only**, the Recovery Appliance creates and validates virtual full backups from each daily incremental backup.

- The retention period defines the period for which backups are maintained, as follows:
  - For backups to **Both Cloud Storage and Exadata Storage**, two separate backups are maintained with different retention periods. By default, the backup to Exadata storage has a seven day retention period and the backup to cloud storage has a thirty day retention period.
  - For backups to **Cloud Storage Only**, the default retention period is thirty days.
  - For backups to **ZDLRA Storage Only**, the retention period is controlled by the policy that is implemented in the Recovery Appliance.
- After the initial retention period, for daily incremental backups to **Both Cloud Storage and Exadata Storage** or **Cloud Storage Only**, the oldest daily incremental backup is automatically merged into the oldest full backup.
- The user data residing in backups is encrypted by default, regardless of the backup destination.
- For database deployments where the Database Type is Database Clustering with RAC and Data Guard Standby, automatic backups are executed on the original primary site; that is, the Exadata system that was initially configured as the primary site, regardless of any role switches.

You can customize some aspects of the backup configuration for your database deployment. See Customizing the Automatic Backup Configuration.

On-Demand Database Backups

You can also create on-demand database backups that use the automatic backup configuration. These backups can be initiated at any time, using the Oracle Database Cloud Service console or the `bkup_api` utility. By default, on-demand backups are managed using the same retention policy as automatic backups.

Optionally, you can configure an on-demand backup as a long-term backup. If you use this option, the backup is not managed using the retention policy for automatic backups. Long-term backups remain until you explicitly remove them from the system.

In addition to complete database backups, you can use the `bkup_api` utility to perform an on-demand backup of an individual pluggable database (PDB).

Viewing Exadata Storage for Database Backups

If your Exadata Cloud at Customer environment is provisioned for database backups on Exadata storage, the backups are stored in the Fast Recovery Area (FRA), which resides in the RECO disk group.

You can confirm that the RECO disk group is used to store the FRA, and view the contents, by using command line tools on the Exadata compute nodes:

1. Connect to a compute node as the `oracle` user.

   See Connecting to a Compute Node Through Secure Shell (SSH).

2. Configure your Oracle Database environment variable settings:

   ```
   $ . oraenv
   ```

3. Confirm that the database is configured to use the RECO disk group to store the Fast Recovery Area:

   ```
   $ sqlplus / as sysdba
   SQL> show parameter DB_RECOVERY_FILE_DEST
   ```

4. Connect to the compute node as the `opc` user.

   See Connecting to a Compute Node Through Secure Shell (SSH).

5. Become the `grid` user:

   ```
   $ sudo -s
   # su - grid
   ```

6. List the ASM disk groups:

   ```
   $ asmcmd lsdg
   ```

7. List the contents of the FRA:

   ```
   $ asmcmd ls FRA_LOCATION
   ```

   where *`FRA_LOCATION`* is the location associated with the `DB_RECOVERY_FILE_DEST` database parameter setting.

Additional Database Backup Options

In addition to the Exadata Cloud at Customer automatic database backup capabilities, you can separately and manually perform Oracle Database backup and recovery operations by using Oracle RMAN or other Oracle Database backup and recovery tools and techniques.

Manually configured backups can use the same cloud storage or Exadata storage locations as the database backups provided by Exadata Cloud at Customer, or they may use other storage locations. If you create manual backups on local Exadata storage, it is recommended that you provision for database backups on Exadata storage in your Exadata Cloud at Customer instance. For more information, see Exadata Storage Configuration.

When implementing a manual backup and recovery scheme, you are responsible for considering all of the associated requirements, including network bandwidth, storage capacity and data security.

Viewing Detailed Backup Configuration Information

You can use the `get_config_info` command of the `bkup_api` utility to view detailed backup configuration settings for database deployments. Optionally, the output can be used to create a file containing JSON formatted output.

1. Connect to a compute node as the **`opc`** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Use the `get_config_info` subcommand to display information about the current backup configuration:

   ```
   # /var/opt/oracle/bkup_api/bkup_api get_config_info --all --dbname dbname [--json
   json_destination]
   ```

   where *dbname* is the database name and *json_destination* is the name of a file to be generated containing JSON formatted output.

4. Exit the root-user command shell and disconnect from the compute node:

   ```
   # exit
   $ exit
   ```

Customizing the Automatic Backup Configuration

You can customize many of the characteristics of the automatic backup configuration.

Topics

- Customizing Backup Settings by Using a Generated Configuration File
- Customizing Which System Files Are Backed Up
- Customizing Which Database Configuration Files Are Backed Up

Customizing Backup Settings by Using a Generated Configuration File

You can customize backup settings for a database deployment by generating a file containing the current customizable settings, editing the file, and then using the file to update the backup settings.

To generate a configuration file with the current backup settings and use it to update the settings:

1. Connect as the **opc** user to a compute node.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Use the `bkup_api get config` command to generate a file containing the current backup settings for the database deployment:

   ```
   # /var/opt/oracle/bkup_api/bkup_api get config [--file=filename] --dbname=dbname
   ```

   where *filename* is an optional parameter used to specify a name for the file that is generated and *dbname* is the database name for the database that you want to act on.

4. Edit the parameter values in the generated file to change any settings you want to customize in the backup configuration.

   The following parameters can be modified to customize the backup configuration:

Parameter	Description
bkup_cron_entry	Enables the automatic backup configuration. Valid values are yes and no.
bkup_cfg_files	Enables backup of system and database configuration files. Valid values are yes and no.
bkup_daily_time	Start time of the automatic daily backup expressed in 24-hour time as hh:mm.
bkup_archlog_cron_entry	Enables automatic backups of archived database log files. Valid values are yes and no. Default value is yes. Setting bkup_archlog_cron_entry to no disables automatic archive log clean-up jobs. This setting is valid only when the associated database has no automatic database backups configured.
bkup_archlog_frequency	Interval in minutes between automatic backups of archived database log files. Valid values are 15, 20, 30, 60, and 120. Default value is 60.
bkup_disk	Enables backups to local Exadata storage. Valid values are yes and no.
bkup_disk_recovery_window	Retention period for backups on local Exadata storage, expressed as a number of days up to 14. Only applicable when bkup_disk is set to yes. Default value is 7.
bkup_oss	Enables backups to cloud storage. Valid values are yes and no.
bkup_oss_recovery_window	Retention period for backups to cloud storage, expressed as a number of days up to 90. Only applicable when bkup_oss is set to yes. Default value is 30.
bkup_oss_url	Location of the storage container that is used for backup to cloud storage. Only applicable when bkup_oss is set to yes.
bkup_oss_user	User name of the Oracle Cloud user having write privileges on the cloud storage container specified in bkup_oss_url. Only applicable when bkup_oss is set to yes.
bkup_oss_passwd	Password of the Oracle Cloud user having write privileges on the cloud storage container specified in bkup_oss_url. Only applicable when bkup_oss is set to yes.
bkup_oss_l0_day	Day of the week when a level 0 backup is taken and stored on cloud storage. Valid values are mon, tue, wed, thu, fri, sat, sun. Only applicable when bkup_oss is set to yes. Default value is sun.
bkup_zdlra	Enables backups to a Recovery Appliance. Valid values are yes and no.
bkup_zdlra_url	Location of the Recovery Appliance that is being used for backups. Only applicable when bkup_zdlra is set to yes.
bkup_zdlra_user	The virtual private catalog (VPC) user name for the Recovery Appliance specified in bkup_zdlra_url. Only applicable when bkup_zdlra is set to yes.
bkup_zdlra_passwd	Password of the Recovery Appliance user specified in bkup_zdlra_url. Only applicable when bkup_zdlra is set to yes.
bkup_rman_compression	Level of compression applied to automatic backups. Valid values are basic, low, medium, and high. Default value is low.

Parameter	Description
bkup_set_section_size	Enables the use of the RMAN multisection backup feature. Valid values are yes and no.
bkup_section_size	RMAN section size that is used for automatic backups. Default value is 64G. Only applicable when bkup_set_section_size is set to yes.
bkup_channels_node	Number of RMAN channels that are used for automatic backups. Valid values are between 1 and 32. Default value is 4.
bkup_use_rcat	Enables the use of an existing RMAN recovery catalog. Valid values are yes and no.
bkup_rcat_user	Recovery catalog user name. Only applicable when bkup_use_rcat is set to yes.
bkup_rcat_passwd	Password for recovery catalog user specified in bkup_rcat_user. Only applicable when bkup_use_rcat is set to yes.
bkup_rcat_conn	Connection string for the RMAN recovery catalog. Only applicable when bkup_use_rcat is set to yes.

5. Use the `bkup_api set config` command to update the backup settings using the file containing your updated backup settings:

   ```
   # /var/opt/oracle/bkup_api/bkup_api set config --file=filename --dbname=dbname
   ```

   where *filename* is used to specify the name of the file that contains the updated backup settings and *dbname* is the database name for the database that you are acting on.

6. You can use the `bkup_api configure_status` command to check the status of the configuration update:

   ```
   # /var/opt/oracle/bkup_api/bkup_api configure_status
   ```

7. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Note that any changes you make by using the `bkup_api` command are not reflected in the Oracle Database Exadata Cloud at Customer console.

Customizing Which System Files Are Backed Up

If your backup configuration includes `bkup_cfg_files=yes`, then each backup includes system configuration files and directories specified in the `oscfg.spec` file.

To change which system files and directories are backed up:

1. Connect as the **oracle** user to a compute node.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Edit the contents of the `oscfg.spec` file. This file is located under `/var/opt/oracle/dbaas_acfs/bkup/`*dbname*, where *dbname* is the name of the database that is associated with the backup configuration.

   Following is an example of the default contents of the `oscfg.spec` file:

   ```
   ## OS Configuration Files
   #
   ```

```
# Doc Spec
oscfg.spec
#
# Directories
/etc/rc.d
/home/oracle/bkup
#
# Single files
/home/oracle/.bashrc
/etc/crontab
/etc/sysctl.conf
/etc/passwd
/etc/group
/etc/oraInst.loc
/etc/oratab
/etc/fstab
```

Customizing Which Database Configuration Files Are Backed Up

If your backup configuration includes `bkup_cfg_files=yes`, then each backup includes database configuration files and directories specified in the `dbcfg.spec` file.

To change which database configuration files are backed up:

1. Connect as the **`oracle`** user to a compute node.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Edit the contents of the `dbcfg.spec` file. This file is located under `/var/opt/oracle/dbaas_acfs/bkup/`*`dbname`*, where *`dbname`* is the name of the database that is associated with the backup configuration.

   Following is an example of the contents of the `dbcfg.spec` file:

   ```
   ### Oracle_Home configuration files.
   #
   # Doc Spec
   dbcfg.spec
   # DB id
   dbid
   #
   # Directories
   /u02/app/oracle/product/dbversion/dbhome_n/admin/dbname/xdb_wallet
   /u02/app/oracle/admin/dbname/xdb_wallet
   /u02/app/oracle/admin/dbname/db_wallet
   # Note: tde_wallet must be backed up in a different location than
   DATA bkup.
   /u02/app/oracle/admin/dbname/tde_wallet
   /u02/app/oracle/admin/dbname/cat_wallet
   #/u01/app/oraInventory
   #
   # Single files
   /var/opt/oracle/dbaas_acfs/dbname/opc/opcdbname.ora
   /u02/app/oracle/product/dbversion/dbhome_n/dbs/opcdbname.ora
   /u02/app/oracle/product/dbversion/dbhome_n/dbs/orapwinstancename
   ```

```
/u02/app/oracle/product/dbversion/dbhome_n/network/admin/listener.ora
/u02/app/oracle/product/dbversion/dbhome_n/network/admin/sqlnet.ora
/u02/app/oracle/product/dbversion/dbhome_n/network/admin/tnsnames.ora
/u02/app/oracle/product/dbversion/dbhome_n/rdbms/lib/env_rdbms.mk
/u02/app/oracle/product/dbversion/dbhome_n/rdbms/lib/ins_rdbms.mk
#
# Creg
/var/opt/oracle/creg/instancename.ini
#
```

Creating an On-Demand Backup

You can create an on-demand backup of an Oracle Database Exadata Cloud at Customer database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in Other Ways to Create an On-Demand Backup at the end of this topic.

Creating an On-Demand Backup by Using the Oracle Database Cloud Service Console

1. Open the Instances page of the Oracle Database Cloud Service console.

   For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

2. Click the database deployment for which you want to create a backup.

   The Oracle Database Cloud Service Overview page is displayed.

3. Click the Administration tile.

   The Oracle Database Cloud Service Backup page is displayed.

4. Click **Backup Now**.

   The Backup Now dialog is displayed.

5. Make a selection for the **Keep Forever** option and then click **Backup**.

   The **Keep Forever** option controls the backup retention policy, as follows:

   - **No** — specifies that the backup is produced and maintained in accordance with the automatic backup retention policy that is associated with the database deployment.
   - **Yes** — specifies that the backup is a long-term backup, which is produced and maintained independently of the automatic backup retention policy that is associated with the database deployment. Long-term backups remain until you explicitly remove them from the system.

Other Ways to Create an On-Demand Backup

- You can use the `bkup_api` utility. See Creating an On-Demand Backup by Using the bkup_api Utility.

Creating an On-Demand Backup by Using the bkup_api Utility

You can use the `bkup_api` utility to create an on-demand backup of a complete database or an individual pluggable database (PDB):

1. Connect as the **`opc`** user to a compute node. In a Data Guard configuration, connect to a compute node hosting the primary database.

For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Enter the `bkup_api` command:

   - To create a backup that follows the current retention policy, use the following `bkup_api` command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api bkup_start --dbname=dbname
     ```

     where *`dbname`* is the database name for the database that you want to back up.

   - To create an on-demand backup of a specific PDB, use the following `bkup_api` command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api bkup_start --dbname=dbname --
     pdb=pdbname
     ```

   - To create a long-term backup of the complete database that persists until you delete it, use the following `bkup_api` command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api bkup_start --keep --dbname=dbname
     ```

     By default, the long-term backup is given a timestamp-based tag. To specify a custom backup tag, add the `--tag` option to the `bkup_api` command. For example, to create a long-term backup with the tag `monthly`, use the following command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api bkup_start --keep --tag=monthly --
     dbname=dbname
     ```

   - To create an on-demand RMAN level 0 backup, use the following `bkup_api` command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api bkup_start --level0 --dbname=dbname
     ```

     You can use this option to manually perform an RMAN level 0 (full) backup if the scheduled weekly level 0 backup fails or following a major structural change in the database, such as adding a new data file or tablespace. This option is only valid for backup configurations that use cloud storage only.

   - To create an on-demand backup that includes an image copy of the database data files, use the following `bkup_api` command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api bkup_start --datafiles --
     dbname=dbname
     ```

     You can use this option to manually perform a full image backup to cloud storage if the scheduled weekly full backup fails or following a major structural change in the database, such as adding a new data file or tablespace. This option is only valid for backup configurations that use cloud storage and local Exadata storage.

4. After you start an on-demand backup, the backup process runs in the background. To check the progress of the backup process, run the following `bkup_api` command on the same compute node where the backup is running:

```
# /var/opt/oracle/bkup_api/bkup_api bkup_status --dbname=dbname
```

5. Exit the root-user command shell and disconnect from the compute node:

   ```
   # exit
   $ exit
   ```

Deleting a Backup

You can delete long-term backups created using the `bkup_api` utility with the `--keep` option.

You cannot delete backups that are associated with the automatic backup configuration, whether they were created using the `bkup_api` utility or the Oracle Database Cloud Service console. These backups are deleted automatically based on the retention period that is associated with the automatic backup configuration.

To delete a long-term backup of a database deployment on Oracle Database Exadata Cloud at Customer:

1. Connect to a compute node as the **`opc`** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. List the available long-term backups:

   ```
   # /var/opt/oracle/bkup_api/bkup_api list --keep --dbname=dbname
   ```

   where *`dbname`* is the database name for the database that you want to act on.

   A list of available backups is displayed. Note the tag of the backup that you want to delete.

4. Delete the backup:

   ```
   # /var/opt/oracle/bkup_api/bkup_api bkup_delete --bkup=backup-tag --dbname=dbname
   ```

   where *`backup-tag`* is the tag of the backup you want to delete.

5. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Updating the Password for Backing Up to Cloud Storage

Whenever the password is changed for an Oracle Cloud user whose credentials are used for backing up to an Oracle Storage Cloud container, you need to update the user's password in the corresponding backup configuration.

Because Oracle Cloud requires users to change their passwords on a regular basis, you need to perform this task regularly.

You can update the password by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in Other Ways to Update the Password at the end of this topic.

Updating the Password by Using the Oracle Database Cloud Service Console

1. Go to the Backup page of the deployment whose backup credentials you want to update:
   1. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.
   2. Click the name of the database deployment whose backup credentials you want to update.

      The Overview page for the deployment is displayed.
   3. Click the Administration tile.

      The Backup page for the deployment is displayed.
2. Click **Configure Backups**.

   The Configure Backups window is displayed.
3. Enter the Cloud user name and new password.
4. Click **Save** and then confirm the operation.

Other Ways to Update the Password

- You can use the `bkup_api` utility. See Customizing the Automatic Backup Configuration.

Changing to a Different Backup Destination

With Oracle Database Exadata Cloud at Customer, you can change the backup destination for your database deployment after creating it.

The instructions in this topic describe how to switch backup destinations for an existing database deployment. Specifically, the following changes are possible using the instructions in this topic:

- From **None** to **Both Cloud Storage and Local Storage**
- From **None** to **Cloud Storage Only**
- From **Both Cloud Storage and Local Storage** to **Cloud Storage Only**
- From **Cloud Storage Only** to **Both Cloud Storage and Local Storage**

For background information on the destinations, see About Backing Up Database Deployments on Exadata Cloud at Customer.

> **Note:**
>
> The Oracle Database Cloud Service console does not recognize backup configuration changes made outside the console. Therefore, depending on what backup destination change you make, the console will not reflect the new backup destination and may not list any completed backups. If the backups are not displayed, you will not be able to use the Oracle Database Cloud Service console to perform recovery.

Prerequisites

- If you are switching to the backup destination Both Cloud Storage and Local Storage or Cloud Storage Only, you must have an Oracle Storage Cloud Service container in your account that is reserved for backups. If you don’t have one, you must create one. See Creating Containers in *Using Oracle Storage Cloud Service*, or see the tutorial Oracle Storage Cloud Service: Creating Containers Using the REST API.

Changing the Backup Destination

Oracle Database Exadata Cloud at Customer allows you to change the backup destination for your database deployments after creating them.

Before changing the backup destination, make sure you have performed applicable Prerequisites.

1. Connect as the **oracle** user to a compute node that is associated with your database deployment.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Configure your Oracle Database environment variable settings:

   ```
   $ . oraenv
   ```

3. Start an RMAN session:

   ```
   $ rman target=/
   ...
   RMAN>
   ```

4. Delete any existing backups.

   ```
   RMAN> delete backup;
   ```

   All RMAN-managed backups for the database are deleted. This process may take several minutes.

5. Exit the RMAN session:

   ```
   RMAN> exit;
   ```

6. Close your connection to the compute node as the **oracle** user.

7. Connect as the **opc** user to a compute node that is associated with your database deployment.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

8. Start a root-user command shell:

```
$ sudo -s
#
```

9. Enter the `bkup_api` command to generate a file containing the current backup settings:

```
# /var/opt/oracle/bkup_api/bkup_api get config [--file=filename] --
dbname=dbname
```

   where *`filename`* is an optional parameter used to specify a name for the file that will be generated and *`dbname`* is the database name for the database that you want to act on.

10. Edit the parameter values in the generated file to change the backup destination. The following parameters are used to customize the backup destination:

Parameter	Description
`bkup_disk`	Enables backups to local Exadata storage. Valid values are `yes` and `no`.
`bkup_disk_recovery_window`	Retention period for backups on local Exadata storage. Value is expressed as a number of days between 1 and 14. Only applicable when `bkup_disk` is set to `yes`.
`bkup_oss`	Enables backups to cloud storage. Valid values are `yes` and `no`.
`bkup_oss_l0_day`	Day of the week when a level 0 backup is taken and stored on cloud storage. Valid values are `mon`, `tue`, `wed`, `thu`, `fri`, `sat`, `sun`. Only applicable when `bkup_oss` is set to `yes`.
`bkup_oss_recovery_window`	Retention period for backups to cloud storage. Value is expressed as a number of days between 1 and 90. Only applicable when `bkup_oss` is set to `yes`.
`bkup_oss_url`	Location of the storage container that is used for backup to cloud storage. Only applicable when `bkup_oss` is set to `yes`.
`bkup_oss_user`	User name of the Oracle Cloud user having write privileges on the cloud storage container specified in `bkup_oss_url`. Only applicable when `bkup_oss` is set to `yes`.
`bkup_oss_passwd`	Password of the Oracle Cloud user having write privileges on the cloud storage container specified in `bkup_oss_url`. Only applicable when `bkup_oss` is set to `yes`.

11. Enter the `bkup_api` command to update the backup settings using the file you generated:

```
# /var/opt/oracle/bkup_api/bkup_api set config --file=filename --
dbname=dbname
```

   where *`filename`* is used to specify the name of the file that contains the updated backup settings and *`dbname`* is the database name for the database that you are acting on.

12. You can use this `bkup_api` command to check the status of the update:

```
# /var/opt/oracle/bkup_api/bkup_api configure_status
```

13. Exit the root-user command shell:

```
# exit
$
```

Disabling and Re-enabling Scheduled Backups

You can disable and re-enable regularly scheduled backups of a database deployment.

You can disable and re-enable scheduled backups by generating a file containing the current settings, editing the file, and then using the file to update the backup settings. To generate a configuration file with the current backup settings and use it to update the settings:

1. Connect as the **opc** user to a compute node that is associated with the database deployment.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Enter the `bkup_api` command to generate a file containing the current backup settings:

   ```
   # /var/opt/oracle/bkup_api/bkup_api get config [--file=filename] --dbname=dbname
   ```

   where *filename* is an optional parameter used to specify a name for the file that will be generated and *dbname* is the database name for the database that you want to act on.

4. Edit the `bkup_cron_entry` parameter in the generated file to disable or re-enable scheduled backups.

5. Enter this `bkup_api` command to update the backup settings using the file you generated:

   ```
   # /var/opt/oracle/bkup_api/bkup_api set config --file=filename --dbname=dbname
   ```

   where *filename* is used to specify the name of the file that contains the updated backup settings and *dbname* is the database name for the database that you want to act on.

6. You can use this `bkup_api` command to check the status of the update:

   ```
   # /var/opt/oracle/bkup_api/bkup_api configure_status
   ```

7. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Validating Backups

Regular backup validation is recommended to ensure that backups are complete and corruption-free. You can use the `reval_start` command of the `bkup_api` utility to validate backups.

During backup validation, the backup sets that are associated with the specified database are read and checked for corruption. Internally, the RMAN `restore validate` command is used to stream data from the backup to the database server for validation purposes. This process acts like a normal restore, except that the data is discarded after the validation without writing to any data files.

Use the following procedure to perform backup validation:

> **Note:**
>
> Oracle recommends that you perform this procedure at least monthly.

1. Connect as the **opc** user to a compute node that is associated with the database deployment. Then, start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Enter the `bkup_api` command to perform backup validation:

   ```
   # /var/opt/oracle/bkup_api/bkup_api reval_start --dbname=dbname
   ```

   where *dbname* is the database name of the database that is the subject of the backup validation operation.

   In the command output, take note of the backup request unique identifier (UUID).

3. You can use this `bkup_api` command to check the status of the backup validation:

   ```
   # /var/opt/oracle/bkup_api/bkup_api --uuid=UUID --dbname=dbname
   ```

   In the preceding command:

   - *UUID* is the backup request UUID from the output of the `bkup_api` command to perform backup validation.
   - *dbname* is the database name of the database that is the subject of the backup validation operation.

Restoring from the Most Recent Backup

You can restore the most recent backup and perform complete recovery on an Oracle Database Exadata Cloud at Customer database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in Other Ways to Restore from the Most Recent Backup at the end of this topic.

Restoring from the Most Recent Backup by Using the Oracle Database Cloud Service Console

1. Go to the Backup page of the deployment you want to restore and recover:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. Click the database deployment you want to restore and recover.

      The Oracle Database Cloud Service Overview page is displayed.

   c. Click the Administration tile.

      The Oracle Database Cloud Service Backup page is displayed.

2. Click **Recover**.

   The Database Recovery overlay is displayed.

3. In the list of recovery options, select **Latest**. Then, click **Recover**.

   The restoration and recovery process performs these steps:

   - Shut down the database
   - Prepare for recovery
   - Perform the recovery
   - Restart the database after recovery

Other Ways to Restore from the Most Recent Backup

- You can use the `bkup_api` utility. See Restoring from the Most Recent Backup by Using the bkup_api Utility.
- You can use Oracle Recovery Manager (RMAN) to manually restore a database on Exadata Cloud at Customer. See Manually Restoring from a Backup.

Restoring from the Most Recent Backup by Using the bkup_api Utility

You can use the `bkup_api` utility to restore from the most recent backup and perform complete recovery on a complete database, or recover a specific pluggable database (PDB):

1. Connect as the **`opc`** user to a compute node that is associated with the database deployment. In a Data Guard configuration, connect to the compute node hosting the primary database.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Enter the following `bkup_api` command:

   ```
   # /var/opt/oracle/bkup_api/bkup_api recover_start --latest --dbname=dbname
   ```

   where *`dbname`* is the database name for the database that you want to recover.

   If you want to recover a specific PDB, then add the `--pdb=`*`pdbname`* option, where *`pdbname`* is the PDB name.

   > **Note:**
   >
   > - It is recommended to perform a complete database backup after every PDB recovery.
   > - PDB recovery does not restore database files (control files, spfiles, data files and so on). Therefore, if a file is missing you must recover the entire database.

If you want to restore system and database configuration files included in the backup, then add the `--cfgfiles` option. Without this option, only the database is restored.

4. After you enter a `bkup_api recover_start` command, the recovery process runs in the background. To check the progress of the recovery process, enter the following `bkup_api` command:

   ```
   # /var/opt/oracle/bkup_api/bkup_api recover_status --dbname=dbname
   ```

5. Exit the root-user command shell and disconnect from the compute node:

   ```
   # exit
   $ exit
   ```

Restoring from a Specific Backup

You can restore a specific backup and perform recovery to that backup on an Oracle Database Exadata Cloud at Customer database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in Other Ways to Restore from a Specific Backup at the end of this topic.

Restoring from a Specific Backup by Using the Oracle Database Cloud Service Console

1. Go to the Backup page of the deployment you want to restore and recover:
   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. Click the database deployment you want to restore and recover.

      The Oracle Database Cloud Service Overview page is displayed.

   c. Click the Administration tile.

      The Oracle Database Cloud Service Backup page is displayed.

2. In the list of backups, locate the backup you want to restore from.

3. Click the action menu (☰) that is associated with the backup you want to restore from. Choose **Recover** and then confirm the action.

   The restoration and recovery process performs these steps:

   - Shut down the database
   - Prepare for recovery
   - Perform the recovery
   - Restart the database after recovery

Other Ways to Restore from a Specific Backup

- You can use the `bkup_api` utility. See Restoring from a Specific Backup by Using the bkup_api Utility.
- You can use Oracle Recovery Manager (RMAN) to manually restore a database on Exadata Cloud at Customer. See Manually Restoring from a Backup.

Restoring from a Specific Backup by Using the bkup_api Utility

You can use the `bkup_api` utility to restore and recover using a specific backup of a complete database or a specific pluggable database (PDB):

1. Connect as the **`opc`** user to a compute node that is associated with the database deployment. In a Data Guard configuration, connect to the compute node hosting the primary database.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. List the available backups by using the following `bkup_api` command:

   ```
   # /var/opt/oracle/bkup_api/bkup_api list --dbname=dbname
   ```

   where *`dbname`* is the database name for the database that you want to recover.

   If you want to list the available backups for a specific PDB, then add the `--pdb=`*`pdbname`* option, where *`pdbname`* is the PDB name.

4. Commence the recovery by using the following `bkup_api` command:

   ```
   # /var/opt/oracle/bkup_api/bkup_api recover_start -b backup-tag --dbname=dbname
   ```

   where *`backup-tag`* is the tag for the specific backup that you want to use and *`dbname`* is the database name for the database that you want to recover.

   If you want to recover a specific PDB, then add the `--pdb=`*`pdbname`* option, where *`pdbname`* is the PDB name.

   > **Note:**
   >
   > - It is recommended to perform a complete database backup after every PDB recovery.
   > - PDB recovery does not restore database files (control files, spfiles, data files and so on). Therefore, if a file is missing you must recover the entire database.

   If you want to restore system and database configuration files included in the backup, then add the `--cfgfiles` option. Without this option, only the database is restored.

5. After you enter a `bkup_api recover_start` command, the recovery process runs in the background. To check the progress of the recovery process, enter the following `bkup_api` command:

   ```
   # /var/opt/oracle/bkup_api/bkup_api recover_status --dbname=dbname
   ```

6. Exit the root-user command shell and disconnect from the compute node:

```
# exit
$ exit
```

Restoring to a Specific Point in Time

You can restore from a backup and perform recovery to a specific point in time on an Oracle Database Exadata Cloud at Customer database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in Other Ways to Restore to a Specific Point in Time at the end of this topic.

Restoring to a Specific Point in Time by Using the Oracle Database Cloud Service Console

1. Go to the Backup page of the deployment you want to restore and recover:
   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. Click the database deployment you want to restore and recover.

      The Oracle Database Cloud Service Overview page is displayed.

   c. Click the Administration tile.

      The Oracle Database Cloud Service Backup page is displayed.

2. Click **Recover**.

   The Database Recovery overlay is displayed.

3. In the list of recovery options, select **Date and Time** or **System Change Number** (SCN) to indicate how you want to specify the end point of the recovery operation. Then, enter the appropriate value.

   > **Note:**
   >
   > If specified, the recovery date and time values are subject to the UTC time zone.

4. Click **Recover**.

   The restoration and recovery process performs these steps:

   - Shut down the database
   - Prepare for recovery
   - Perform the recovery
   - Restart the database after recovery

Other Ways to Restore to a Specific Point in Time

- You can use the `bkup_api` utility. See Restoring to a Specific Point in Time by Using the bkup_api Utility.
- You can use Oracle Recovery Manager (RMAN) to manually restore a database on Exadata Cloud at Customer. See Manually Restoring from a Backup.

Restoring to a Specific Point in Time by Using the bkup_api Utility

You can use the `bkup_api` utility to restore and recover a complete database, or recover a specific pluggable database (PDB), to a specific point in time.

1. Connect as the **`opc`** user to a compute node that is associated with the database deployment. In a Data Guard configuration, connect to the compute node hosting the primary database.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Enter the `bkup_api` command:

   - To recover to a specific Oracle Database system change number (SCN), enter the following `bkup_api` command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api recover_start --scn scn --dbname=dbname
     ```

     where *`scn`* is the SCN of the desired recovery point, and *`dbname`* is the database name for the database that you want to recover.

   - To recover to a specific point in time, enter the following `bkup_api` command:

     ```
     # /var/opt/oracle/bkup_api/bkup_api recover_start -t 'timestamp' --
     dbname=dbname
     ```

     where *`timestamp`* is the recovery point in time expressed in the following format: `DD-MON-YYY HH24:MM:SS`, and *`dbname`* is the database name for the database that you want to recover.

     By default, the recovery point in time is subject to the UTC time zone. If you want to use the current database server OS time zone setting, then add the `--nonutc` command line argument as follows:

     ```
     # /var/opt/oracle/bkup_api/bkup_api recover_start -t 'timestamp' --nonutc --
     dbname=dbname
     ```

   Additionally:

   - If you want to recover a specific PDB to a specific point in time or SCN, then add the `--pdb=`*`pdbname`* option, where *`pdbname`* is the PDB name.

     Note:

     – It is recommended to perform a complete database backup after every PDB recovery.

     – PDB recovery does not restore database files (control files, spfiles, data files and so on). Therefore, if a file is missing you must recover the entire database.

- If you want to restore system and database configuration files included in the backup, then add the `--cfgfiles` option. Without this option, only the database is restored.

4. After you enter a `bkup_api recover_start` command, the recovery process runs in the background. To check the progress of the recovery process, enter the following `bkup_api` command:

   ```
   # /var/opt/oracle/bkup_api/bkup_api recover_status --dbname=dbname
   ```

5. Exit the root-user command shell and disconnect from the compute node:

   ```
   # exit
   $ exit
   ```

Restoring System and Database Configuration Files

You can use the `download_cfg` command of the `bkup_api` utility to restore system and database configuration files from an automatic database backup.

1. Connect to the compute node as the **`opc`** user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Use the `download_cfg` subcommand to retrieve the system and database configuration files from an automatic database backup:

   ```
   # /var/opt/oracle/bkup_api/bkup_api download_cfg --dbname=dbname
   ```

   where *`dbname`* specifies the database name for the database that is associated with the backup.

   By default, the system and database configuration files are retrieved from the most recent backup. If you would like to specify differently, you can add one of the following options:

   - Use `--date=`*`backupdate`* to identify a specific backup date that you want to retrieve the system and database configuration files from. The expected format for *`backupdate`* is `MM/DD/YYYY`.

   - Use `--file=`*`filename`* to identify a specific backup file containing system or database configuration files. The *`filename`* that you specify must be the complete name for the file as shown in the object store.

   You can view a list of system and database configuration file backups by using the following command:

   ```
   # /var/opt/oracle/bkup_api/bkup_api list_cfgfiles --dbname=dbname
   ```

   You can use the output from this command to identify a specific *`backupdate`* or *`filename`* to target for retrieval.

   By default, the compressed tar files that contain the system and database configuration files are downloaded to `/var/opt/oracle/log/`*`dbname`*`/cfgfiles`. To specify a different location where you would like to download the system and database configuration files, add the `--location=`*`location`* option.

4. After you retrieve the system and database configuration files from an automatic database backup, you must manually extract the required files from the compressed tar files and copy them to their original locations.

5. Exit the root-user command shell and disconnect from the compute node:

   ```
   # exit
   $ exit
   ```

Recover Backups Using the dbaasapi Utility

You can use the `dbaasapi` utility to restore backup files and perform complete recovery on a database deployment:

1. Connect as the **`opc`** user to a compute node that is associated with the database deployment.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Start a root-user command shell:

   ```
   $ sudo -s
   #
   ```

3. Create a JSON input file with the required recovery command parameters. For example:

   ```
   {
       "action": "start",
       "object": "db",
       "operation": "recover",
       "outputfile": "output-file",
       "params": {
            "dbname": "dbname",
            "cfgfiles": "yes",
            "bkup_tag": "backup-tag"
       }
   }
   ```

   Where valid values in the `params` tag are:

   - `"dbname": "dbname"` — specifies the name of the database being recovered.
   - `"cfgfiles": "yes"` — optionally specifies that you want to recover the system and database configuration files. By default, configuration files are not recovered.
   - One of the following recovery types must also be specified:
     - `"latest": "yes"` — recovers the latest available backup.
     - `"bkup_tag": "backup-tag"` — recovers the backup having the specified *`backup-tag`*.
     - `"scn": "SCN"` — performs a point in time recovery to the specified system change number (*`SCN`*).
     - `"timestamp": "timestamp"` — performs a point in time recovery to the specified *`timestamp`*. The *`timestamp`* value must be UTC and of the format `dd-MMM-yyyy HH:mm:ss`.

4. Restore the backup and perform recovery using the `orec` subcommand of the `dbaascli` utility:

```
# /var/opt/oracle/dbaasapi/dbaasapi -i input-file.json
```

The restoration and recovery process performs these steps:

- Shut down the database
- Extract and restore configuration files, if `"cfgfiles": "yes"` was specified
- Prepare for recovery
- Perform the recovery
- Restart the database after recovery

5. Exit the root-user command shell:

```
# exit
$
```

Manually Restoring from a Backup

Oracle Database Exadata Cloud at Customer provides a backup feature that backs up the Oracle database associated with a database deployment. This feature is built over Oracle Recovery Manager (RMAN).

To manually restore a database backup, you can use the RMAN utility. For information about using RMAN, see the *Oracle Database Backup and Recovery User's Guide* for Release 18, 12.2, 12.1, or 11.2.

View Details About Automatic Backups

You can view historical details about scheduled and on-demand backups that use an automatic backup configuration.

You can use the `bkup_api` utility to view a list of backup jobs, which includes database backups and archive log backups. You can also view details for a specific job.

1. Connect as the **opc** user to the compute node. In a Data Guard configuration, connect to the compute node hosting the primary database. For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH)
2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Enter the following `bkup_api` command to view a list of backup jobs:

```
# /var/opt/oracle/bkup_api/bkup_api list jobs [--all] --dbname=dbname [--
pdb=pdbname]
```

where:

- `--all` optionally specifies that the job list contains database backups and archive log backups. Without this option, the list does not contain archive log backup jobs.
- `--dbname=dbname` specifies the database name for the database that is the subject of the backups.
- `--pdb=pdbname` optionally specifies that the job list only contains backups for the specified pluggable database (PDB).

For example:

```
# /var/opt/oracle/bkup_api/bkup_api list jobs --all --dbname=db12c
DBaaS Backup API V1.5 @2016 Multi-Oracle home
-> This node is not master. Running on Elastic configuration.
-> Action : list_jobs
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
UUID                             | DATE                | STATUS  |
TAG                | TYPE
e1120cba477711e9a16d0010ee68ed9a | 2019-03-15 23:13:11 | success |
TAG20190315T231806 | incremental
0b4ead48484111e9ac030010ee68ed9a | 2019-03-16 23:13:11 | success |
TAG20190316T231816 | incremental
35ec694a490a11e98e130010ee68ed9a | 2019-03-17 23:13:13 | success |
TAG20190317T231854 | full
2fa5e16849cf11e9a5b30010ee68ed9a | 2019-03-18 22:43:06 | success |
None               | archivelog
60a08be849d311e987fb0010ee68ed9a | 2019-03-18 23:13:20 | success |
TAG20190318T231751 | incremental
915f149e49d711e991f60010ee68ed9a | 2019-03-18 23:43:06 | success |
None               | archivelog
f35926d249df11e99f440010ee68ed9a | 2019-03-19 00:43:06 | success |
None               | archivelog
54f3887649e811e9af320010ee68ed9a | 2019-03-19 01:43:06 | success |
None               | archivelog
```

4. To view detailed information for a specific backup job, choose a job UUID from the list of backup jobs and enter the following command:

```
# /var/opt/oracle/bkup_api/bkup_api --uuid=uuid --dbname=dbname
```

For example:

```
# /var/opt/oracle/bkup_api/bkup_api --uuid=60a08be849d311e987fb0010ee68ed9a --
dbname=db12c
DBaaS Backup API V1.5 @2016 Multi-Oracle home
-> This node is not master. Running on Elastic configuration.
@ STARTING CHECK STATUS 60a08be849d311e987fb0010ee68ed9a
[ REQUEST TICKET ]
[UUID    ->  60a08be849d311e987fb0010ee68ed9a
[DBNAME  ->  db12c
[STATE   ->  success
[ACTION  ->  create-backup-incremental
[STARTED ->  2019-03-18 23:13:20
[ENDED   ->  2019-03-18 23:18:30.921995
[PID     ->  282732
[TAG     ->  TAG20190318T231751
[LOG     ->  API:: Wallet is in open state
[LOG     ->  API:: Oracle database state is up and running
[LOG     ->  API:: DB instance: db12c
[LOG     ->  API:: Validating the backup repository ......
[LOG     ->  API::      All backup pieces are ok
[LOG     ->  API:: Performing backup to local storage (primary backup)
[LOG     ->  API:: Executing rman instructions
[LOG     ->  API:: ...... OK
[LOG     ->  API:: Backup to local storage is completed
[LOG     ->  API:: Clean MOTD.
[LOG     ->  API:: Performing backup to cloud storage (secondary backup)
[LOG     ->  API:: Executing rman instructions
[LOG     ->  API:: ......OK
[LOG     ->  API:: Backup to cloud storage is completed
[LOG     ->  API:: Clean MOTD.
```

```
[LOG     ->  API:: Starting backup of config files
[LOG     ->  API:: at time: 2019-03-18:23:17:51
[LOG     ->  API:: Determining the oracle database id
[LOG     ->  API::  DBID: 4075204950
[LOG     ->  API:: Creating directories to store config files
[LOG     ->  API:: Determining the oracle database id
[LOG     ->  API::  DBID: 4075204950
[LOG     ->  API:: Enabling RAC exclusions for config files.
[LOG     ->  API:: Compressing config files into tar files
[LOG     ->  API:: Uploading config files to cloud storage
[LOG     ->  API:: Completed at time: 2019-03-18:23:18:18
[LOG     ->  API:: at time: 2019-03-18:23:18:18
[LOG     ->  API:: Config files backup ended successfully
[LOG     ->  API:: Clean MOTD.
[LOG     ->  API:: All requested tasks are completed
[ END TICKET ]
```

5. Exit the root-user command shell and disconnect from the compute node:

   ```
   # exit
   $ exit
   ```

7

Patching Exadata Cloud at Customer

This section explains how to apply a patch to Oracle Database Exadata Cloud at Customer, and roll back the patch as necessary.

Topics

- About Patching Exadata Cloud at Customer
- The exadbcpatchmulti Command
- Listing Available Patches
- Checking Prerequisites Before Applying a Patch
- Applying a Patch
- Listing Applied Patches
- Rolling Back a Patch or Failed Patch

About Patching Exadata Cloud at Customer

Routine Patching Facilities in Exadata Cloud at Customer

Oracle Database Exadata Cloud at Customer provides facilities to manage routine patching of the Oracle Database and Oracle Grid Infrastructure software, and the compute node user domain (DomU) operating system environment. These facilities are provided by the Oracle Database Cloud Service console and also by means of using the `exadbcpatchmulti` and `dbaascli` utilities. You are responsible for managing patches and updates to the Oracle Database and Oracle Grid Infrastructure software on the compute nodes, and the compute node user domain (DomU) operating system environment.

This document focuses on using the facilities provided by Exadata Cloud at Customer to perform the following patching operations:

- The exadbcpatchmulti Command
- Listing Available Patches
- Checking Prerequisites Before Applying a Patch
- Applying a Patch
- Listing Applied Patches
- Rolling Back a Patch or Failed Patch

Manually Patching Oracle Database and Oracle Grid Infrastructure Software

In general, Oracle recommends that you use the facilities provided by Exadata Cloud at Customer to perform routine patching of Oracle Database and Oracle Grid Infrastructure software. However, if you encounter a problem that requires a patch which is not included in any routine patch set, then work with Oracle Support Services to identify and apply the appropriate patch.

For general information about patching Oracle Database, see "Patch Set Updates and Requirements for Upgrading Oracle Database" in the *Oracle Database Upgrade Guide* for Release 18, 12.2, 12.1, or 11.2.

Manually Patching the Compute Node Operating System

Oracle recommends that you use the facilities provided by Exadata Cloud at Customer to perform routine patching of the compute node user domain (DomU) operating system environment.

However, if you want to apply a patch or perform an Exadata software update that is not displayed in the list of available patches (see Listing Available Patches), then you can manually update operating system components on the compute nodes by using standard Exadata tools and techniques outlined at Updating Exadata Database Servers in the *Oracle Exadata Database Machine Maintenance Guide*.

By using standard Exadata tools and techniques, you are able to apply Exadata software release updates to the compute nodes at your convenience. For feature release updates only, Oracle recommends that you lodge a service request with Oracle Support Services to ensure that Oracle is aware of your plans. A feature release update is an update that changes any of the first four digits in the Exadata software release identifier. For example, upgrading from Exadata software release 12.1.2.2.0 to release 12.1.2.3.0 would be a feature release update. However, upgrading from Exadata software release 12.1.2.3.0 to release 12.1.2.3.4 would not be considered a feature release update. You can determine the current Exadata software release by executing the `imageinfo` command on any compute node.

Patching Performed by Oracle

Patches and updates to all other system components are managed and performed by Oracle. This includes the physical compute nodes (Dom0), network switches, power distribution units (PDUs), integrated lights-out management (ILOM) interfaces, and the Exadata Storage Servers.

In all but rare exceptional circumstances, you will receive advance communication about these updates through the Cloud Notification Portal to help you plan for them. If there are corresponding recommended updates for your compute node virtual machine environment, then Oracle will provide notification about these. There is no option to opt out of any updates.

Wherever possible, scheduled updates are performed in a manner that preserves service availability throughout the update process. However, there may be some noticeable impact on performance and throughput as individual system components are unavailable for a period of time during the update process.

For example, the compute nodes may need to be rebooted when a service is updated. In such cases, wherever possible, the compute nodes would be rebooted in a rolling manner, one at a time, to ensure that the service, and the Oracle databases contained therein, remain available throughout the process. However, while each compute node is being rebooted it is not available for a short period of time. Consequently, the service may not be able to cater for the same workload while each individual server is unavailable.

For further information about the standard update policies and practices that apply to Exadata Cloud at Customer see Oracle Database Cloud Exadata Service Supported Software Versions and Planning for Updates.

The exadbcpatchmulti Command

You can use the `exadbcpatchmulti` utility to perform assisted patching operations for Oracle Grid Infrastructure and Oracle Database on Exadata Cloud at Customer. The `exadbcpatchmulti` utility is located under `/var/opt/oracle/exapatch` on every compute node.

> **Note:**
>
> - The `exadbcpatchmulti` command requires root administration privileges. Therefore, you need to connect to the compute node as the `opc` user and then start a root-user command shell to perform patching operations.
> - The `exadbcpatchmulti` command uses the cloud-specific tooling included on your Exadata Cloud at Customer compute nodes, and specific patches may require functionality provided by a specific version of the tools. Therefore, it is recommended to update to the latest version of the cloud tools before performing any patching operations. See Updating the Cloud Tooling on Exadata Cloud at Customer.
> - The `exadbcpatchmulti` command only works along with patches for Oracle Grid Infrastructure and Oracle Database. It cannot be used along with operating system patches.
> - Each `exadbcpatchmulti` action also has an equivalent action that is available by using the `dbaascli` command. The `dbaascli` command variants are provided separately in this document.

The syntax for the `exadbcpatchmulti` command depends on the action being performed, which is specified as the first argument to the command. The following list outlines the available patching actions and the syntax of the `exadbcpatchmulti` command for each action. Detailed procedures and examples for each action are provided separately in this document.

- To list the available patch identifiers for an Oracle Home directory:

  ```
  # /var/opt/oracle/exapatch/exadbcpatchmulti -list_patches
  -oh=hostname:oracle_home [-sshkey=sshkey_file]
  ```

- To check prerequisites before applying a patch:
  - On specific instances:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async patchid
    -instance1=hostname1:oracle_home
    [-instance2=hostname2:oracle_home ...]
    [-dbnames=dbname[,dbname2 ...]] [-sshkey=sshkey_file]
    ```

  - By specifying only database names:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async patchid
    -dbnames=dbname[,dbname2 ...] [-alldbs] [-sshkey=sshkey_file]
    ```

- To apply a patch:
  - On specific instances:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
    -instance1=hostname1:oracle_home
    [-instance2=hostname2:oracle_home ...]
    [-dbnames=dbname[,dbname2 ...]] [-run_datasql=(0|1)] [-
    sshkey=sshkey_file]
    ```

  - By specifying only database names:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
    -dbnames=dbname[,dbname2 ...] [-run_datasql=(0|1)] [-alldbs] [-
    sshkey=sshkey_file]
    ```

- To report the status of a patching operation for an Oracle Home directory:

  ```
  # /var/opt/oracle/exapatch/exadbcpatchmulti -get_status patchtxn
  -oh=hostname:oracle_home [-sshkey=sshkey_file]
  ```

- To roll back a previously applied patch or a failed patch:
  - On specific instances:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async
    patchid
    -instance1=hostname1:oracle_home
    [-instance2=hostname2:oracle_home ...]
    [-dbnames=dbname[,dbname2 ...]] [-run_datasql=(0|1)] [-
    sshkey=sshkey_file]
    ```

  - By specifying only database names:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async
    patchid
    -dbnames=dbname[,dbname2 ...] [-run_datasql=(0|1)] [-alldbs] [-
    sshkey=sshkey_file]
    ```

The following table describes the arguments shown in the syntax for the `exadbcpatchmulti` command.

Argument	Description
patchid	Identifies the patch to be pre-checked, applied, or rolled back. To list the applicable patch identifiers for an Oracle Home directory, execute the `exadbcpatchmulti` command with the `-list_patches` action.

Argument	Description
-instance*N* = *hostname*:*oracle_home*	Specifies a compute node and Oracle Home directory that is the target of the specified patching action. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory. If you specify this option, then you explicitly identify the nodes and Oracle Home directory locations that you want to patch. You can patch all of your nodes using one command or you may patch some nodes in one run and patch the rest later. If you use this argument to specify a shared Oracle Home directory and you do not specify the -dbnames argument, then the patching action is applied to all of the databases that share the specified Oracle Home.
-dbnames = *dbname*[,*dbname2* ...]	Specifies the database names for the databases that are the target of the specified patching action. If you use this argument to patch databases that use a shared Oracle Home and you do not specify the -alldbs option, then a new Oracle Home containing the patched Oracle Database binaries is created and the databases are moved to the new Oracle Home.
-alldbs	Specifies that you want to apply the specified patching operation to all of the databases that share the same Oracle Database binaries (Oracle Home) as the databases specified in the -dbnames argument.
-run_datasql = (0|1)	Use this argument to control execution of the SQL commands associated with the patch or rollback operation: • -run_datasql=1 executes the SQL commands associated with the patch or rollback operation. • -run_datasql=0 does not execute the SQL commands associated with the patch or rollback operation. If the argument is not specified, the patch-related SQL is executed by default; equivalent to -run_datasql=1. The SQL commands associated with a patch should only be performed after all of the compute nodes are patched or rolled back. Ensure that you specify -run_datasql=0 if you are patching, or rolling back, a subset of nodes and further nodes remain to be patched or rolled back. This argument can only be specified along with a patching or rollback operation acting on a set of compute nodes. If you have patched, or rolled back, all of your nodes and you did not specify this argument, you need to manually execute the SQL commands associated with the patch or rollback operation, which typically involves running the catbundle.sql script for Oracle Database 11g or the datapatch utility for Oracle Database 12c, or later. Refer to the patch documentation for full details.
-oh = *hostname*:*oracle_home*	Specifies the compute node and Oracle Home directory location that is used to search for applicable patches or to report on the current status of a patching operation. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

Argument	Description
-sshkey = *sshkey_file*	Specifies an SSH private key associated with the opc user, which is used to connect to compute nodes in the cluster. Typically this file is located at /home/opc/.ssh/id_rsa.
patchtxn	This argument is only used to report the status of a patching operation for an Oracle Home directory. It specifies the identifier for the patching operation under investigation. The identifier is output to the terminal and also recorded in the log file shortly after the commencement of a pre-check, patch, or rollback operation.

When you run the exadbcpatchmulti command, its activity is recorded in the log file at /var/opt/oracle/log/exadbcpatch/exadbcpatch.log. Log files for previous patching operations are maintained in the same directory and each log file contains a timestamp within its name.

Listing Available Patches

Viewing Available Patches by Using the Oracle Database Cloud Service Console

1. Go to the Patching page for the database deployment on which you want to check patching:
   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.
   b. Click the database deployment on which you want to check patching.

      The Oracle Database Cloud Service Overview page is displayed.
   c. Click the Administration tile and then click the Patching tab.

      The Oracle Database Cloud Service Patching page is displayed.
2. A list of patches you can apply appears in the Available Patches section.

> **Note:**
>
> The Oracle Database Cloud Service Patching page shows only the few most recent patches that are associated with the database deployment for each patching category; that is, operating system patches, database patches or grid infrastructure patches.

Other Ways to View Available Patches

For Oracle Database and Oracle Grid Infrastructure patches, you can use the exadbcpatchmulti or dbaascli utility. See:

- Listing Available Patches by Using the exadbcpatchmulti Command
- Listing Available Patches by Using the dbaascli Command

Listing Available Patches by Using the exadbcpatchmulti Command

You can produce a list of available Oracle Database or Oracle Grid Infrastructure patches using the `exadbcpatchmulti` command as follows:

1. Connect to the compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `exadbcpatchmulti` command with the `-list_patches` action:

   ```
   # /var/opt/oracle/exapatch/exadbcpatchmulti -list_patches
    -oh=hostname:oracle_home [-sshkey=sshkey_file]
   ```

   In the preceding command:

   - `-oh` specifies a compute node and Oracle Home directory for which you want to list the available patches. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.
   - `-sshkey` optionally specifies the location of the SSH private key of the `opc` user, which is used to connect to compute nodes in the cluster.

   For example:

   ```
   # /var/opt/oracle/exapatch/exadbcpatchmulti -list_patches
   -oh=hostname1:/u02/app/oracle/product/12.1.0.2/dbhome_1
   -sshkey=/home/opc/.ssh/id_rsa
   ```

   > **Note:**
   >
   > The list of available patches is determined by interrogating the database to establish the patches that have already been applied. When a patch is applied, the corresponding database entry is made as part of the SQL patching operation, which is executed at the end of the patch workflow. Therefore, the list of available patches may include partially applied patches along with patches that are currently being applied.

Listing Available Patches by Using the dbaascli Command

You can produce a list of available Oracle Database or Oracle Grid Infrastructure patches using the `dbaascli` command as follows:

1. Connect to a compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `dbaascli patch db list` command:

   ```
   # dbaascli patch db list --oh hostname:oracle_home
   ```

   In the preceding command, `--oh` specifies a compute node and Oracle Home directory for which you want to list the available patches. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

For example:

```
# dbaascli patch db list --oh hostname1:/u02/app/oracle/product/
12.1.0.2/dbhome_1
```

> **Note:**
>
> The list of available patches is determined by interrogating the database to establish the patches that have already been applied. When a patch is applied, the corresponding database entry is made as part of the SQL patching operation, which is executed at the end of the patch workflow. Therefore, the list of available patches may include partially applied patches along with patches that are currently being applied.

Checking Prerequisites Before Applying a Patch

Before you apply a patch, you can check its prerequisites to make sure that it can be successfully applied.

The prerequisites-checking operation:

- Confirms that the patch is available for download.
- Confirms connectivity to the required compute nodes.
- Verifies that there is enough space to apply the patch.
- Runs additional commands to validate that the specific patch requirements are met.

Checking Prerequisites Before Applying a Patch by Using the Oracle Database Cloud Service Console

Before You Begin

The patching processes use the cloud-specific tooling included in your Exadata Cloud at Customer environment, and specific patches may require functionality provided by a specific version of the tools. Therefore, it is recommended to update to the latest version of the cloud tools before performing any patching operations. See Updating the Cloud Tooling on Exadata Cloud at Customer.

Procedure

1. Go to the Patching page for the database deployment on which you want to check patching:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. Click the database deployment on which you want to check patching.

      The Oracle Database Cloud Service Overview page is displayed.

c. Click the Administration tile and then click the Patching tab.

   The Oracle Database Cloud Service Patching page is displayed. A list of patches you can apply appears in the Available Patches section.

   > **Note:**
   >
   > The Oracle Database Cloud Service Patching page shows only the few most recent patches that are associated with the database deployment for each patching category; that is, operating system patches, database patches or grid infrastructure patches.

2. Click the action menu (☰) that is associated with the patch whose prerequisites you want to check, and then select **Precheck**.

   If further input is required, specify the required details in the Patch Precheck Service window and click **Precheck** to continue. The Patch Precheck Service window displays in the following circumstances:

   - If you have previously checked prerequisites for the selected patch, the Patch Precheck Service window shows the results of the previous check and asks if you want to perform another set of prerequisite checks.

   The Patching page redisplays, showing a status message indicating that prerequisite checks are in progress.

3. Refresh the Patching page occasionally to update the status message.

   Note that prerequisite checking can take several minutes to complete.

4. When the prerequisite checks are completed, the Precheck results link is displayed.

   Click Precheck results to display the results of the prerequisite checks.

Other Ways to Check Prerequisites Before Applying a Patch

For Oracle Database and Oracle Grid Infrastructure patches, you can use the `exadbcpatchmulti` or `dbaascli` utility. See:

- Checking Prerequisites Before Applying a Patch by Using the exadbcpatchmulti Command
- Checking Prerequisites Before Applying a Patch by Using the dbaascli Command

Checking Prerequisites Before Applying a Patch by Using the exadbcpatchmulti Command

You can perform an Oracle Database or Oracle Grid Infrastructure patch prerequisites-checking (pre-check) operation by using the `exadbcpatchmulti` command as follows:

1. Connect to the compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `exadbcpatchmulti` command with the `-precheck_async` action:

- On specific instances:

  ```
  # /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async 
  patchid
  -instance1=hostname1:oracle_home
  [-instance2=hostname2:oracle_home ...]
  [-dbnames=dbname[,dbname2 ...]] [-sshkey=sshkey_file]
  ```

- By specifying only database names:

  ```
  # /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async 
  patchid
  -dbnames=dbname[,dbname2 ...] [-alldbs] [-sshkey=sshkey_file]
  ```

In the preceding command:

- *`patchid`* identifies the patch to be pre-checked.

  > **Note:**
  >
  > For details about how to find the available patch identifiers, see Listing Available Patches.

- `-instance`*`N`* specifies a compute node and Oracle Home directory that is subject to the pre-check operation. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.
- `-dbnames` specifies the database names for the databases that are the target of the pre-check operation.
- `-alldbs` specifies that you want to pre-check all of the databases that share the same Oracle Database binaries (Oracle Home) as the specified databases.
- `-sshkey` optionally specifies the location of the SSH private key of the `opc` user, which is used to connect to compute nodes in the cluster.

For example:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async 12345678
-instance1=hostname1:/u02/app/oracle/product/12.1.0.2/dbhome_1
-instance2=hostname2:/u02/app/oracle/product/12.1.0.2/dbhome_1
```

Checking Prerequisites Before Applying a Patch by Using the dbaascli Command

You can perform an Oracle Database or Oracle Grid Infrastructure patch prerequisites-checking (pre-check) operation by using the `dbaascli` command as follows:

1. Connect to a compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `dbaascli patch db prereq` command:
   - On specific instances:

     ```
     # dbaascli patch db prereq --patchid patchid
     --instance1 hostname:oracle_home [--dbnames dbname[,dbname2 ...]]
     ```

   - By specifying only database names:

     ```
     # dbaascli patch db prereq --patchid patchid
     --dbnames dbname[,dbname2 ...] [-alldbs]
     ```

   In the preceding commands:
   - *`patchid`* identifies the patch to be pre-checked.

     > **Note:**
     >
     > For details about how to find the available patch identifiers, see Listing Available Patches.

   - `--instance1` specifies a compute node and Oracle Home directory that is subject to the pre-check operation. In this context, the Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.
   - `--dbnames` specifies the database names for one or more databases that are the target of the pre-check operation.
   - `-alldbs` specifies that you want to pre-check all of the databases that share the same Oracle Database binaries (Oracle Home) as the specified databases.

   For example:

   ```
   # dbaascli patch db prereq --patchid 12345678
   --instance1 hostname1:/u02/app/oracle/product/12.1.0.2/dbhome_1
   ```

Applying a Patch

Applying a Patch by Using the Oracle Database Cloud Service Console

Before You Begin

- The patching processes use the cloud-specific tooling included in your Exadata Cloud at Customer environment, and specific patches may require functionality provided by a specific version of the tools. Therefore, it is recommended to update to the latest version of the cloud tools before performing any patching operations. See Updating the Cloud Tooling on Exadata Cloud at Customer.
- After you apply an Oracle Database patch, Oracle recommends that you also update the corresponding software image to the same release level. Doing this ensures that databases and pluggable databases (PDBs) created after the update use images that match the updated software. See Administering Software Images.
- After you apply an Oracle Database patch, the Oracle Home name is updated to reflect the release level of the newly applied patch. See How Applying a Patch Impacts the Oracle Homes.

Procedure

1. Go to the Patching page of the database deployment to which you want to apply a patch:

   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.

   b. Click the database deployment to which you want to apply a patch.

      The Oracle Database Cloud Service Overview page is displayed.

   c. Click the Administration tile and then click the Patching tab.

      The Oracle Database Cloud Service Patching page is displayed. A list of patches you can apply appears in the Available Patches section.

      > **Note:**
      >
      > The Oracle Database Cloud Service Patching page shows only the few most recent patches that are associated with the database deployment for each patching category; that is, operating system patches, database patches or grid infrastructure patches.

2. Click the action menu (☰) that is associated with the patch you want to apply, and then select **Patch**.

   The Patch Service window displays.

3. If the database deployment uses a shared Oracle Home, you can select the option to patch all of the database deployments that are associated with the shared Oracle Home.

   If you select this option, then all of the databases are patched at the same time by updating the existing Oracle Home.

   If you do not specify this option, then only the current database deployment is patched. In this case, when you patch the first database that is associated with a shared Oracle Home, it is moved to use a new patched Oracle Home, and when you patch subsequent databases from the original shared Oracle Home, they are moved to use the same patched Oracle Home as the first database. This enables you to patch individual databases while maintaining the original sharing affinity.

4. Enter a note that you wish to associate with the patch. Then, click **Patch**.

   The Patch Service window closes and the patching operation begins.

   The Administration tile shows the starting time of the patching operation and a **Patching...** message replaces the **Patch** button.

   When the patching operation completes, the Patching page shows the completion time of the patching operation, and a log of the operation’s activities appears in the Details of Last Patching Activity section. If the operation was successful, the patch is removed from the list Available Patches list. If the operation fails, the patch remains in the list and you should check the Details of Last Patching Activity section for information about the failure.

> **Note:**
>
> Patching operations are performed in a rolling manner, one compute node at a time, in order to minimize impact on the database. For database deployments where the Database Type is Database Clustering with RAC and Data Guard Standby, the standby site is patched first followed by the primary site.

Other Ways to Apply a Patch

For Oracle Database and Oracle Grid Infrastructure patches, you can use the `exadbcpatchmulti` or `dbaascli` utility. See:

- Applying a Patch by Using the exadbcpatchmulti Command
- Applying a Patch by Using the dbaascli Command

> **Note:**
>
> After you apply an Oracle Database patch, the Oracle Home name is updated to reflect the release level of the newly applied patch. However, when you apply a patch by using the `exadbcpatchmulti` or `dbaascli` utility, the modified Oracle Home name is not updated in the Oracle Database Cloud Service console. For details, see How Applying a Patch Impacts the Oracle Homes.

Applying a Patch by Using the exadbcpatchmulti Command

You can apply a patch by using the `exadbcpatchmulti` command.

The patching operation:

- Can be used to patch some or all of your compute nodes using one command.
- Can be used to patch one or many of your databases using one command.
- Coordinates multi-node patching in a rolling manner.
- Can execute patch-related SQL after patching all the compute nodes in the cluster.

You can perform a patching operation using the `exadbcpatchmulti` command as follows:

1. Connect to a compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `exadbcpatchmulti` command with the `-apply_async` action:

   - On specific instances:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
     -instance1=hostname1:oracle_home
     [-instance2=hostname2:oracle_home ...]
     [-dbnames=dbname[,dbname2 ...]] [-run_datasql=(0|1)] [-
     sshkey=sshkey_file]
     ```

- By specifying only database names:

  ```
  # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
  -dbnames=dbname[,dbname2 ...] [-run_datasql=(0|1)] [-alldbs] [-
  sshkey=sshkey_file]
  ```

In the preceding commands:

- *patchid* identifies the patch to be applied.

  > **Note:**
  >
  > For details about how to find the available patch identifiers, see Listing Available Patches.

- `-instanceN` specifies a compute node and Oracle Home directory that is subject to the patching operation. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

  If you use this argument to specify a shared Oracle Home directory and you do not specify the `-dbname` argument, then all of the databases that share the specified Oracle Home are patched.

- `-dbnames` specifies the database names for the databases that are the target of the patching operation.

  If you use this option to patch a database that uses a shared Oracle Home, and you do not specify the `-alldbs` option, then the database is moved to another Oracle Home that contains the patched Oracle Database binaries.

- `-alldbs` patches all of the databases that share the same Oracle Database binaries (Oracle Home) as the databases specified in the `-dbnames` argument.

- `--run_datasql` controls execution of the SQL commands associated with the patch:

  - `--run_datasql 1` executes the SQL commands associated with the patch.
  - `--run_datasql 0` does not execute the SQL commands associated with the patch.

  If the argument is not specified, the patch-related SQL is executed by default; equivalent to `--run_datasql 1`.

  Patch-related SQL should only be executed after all of the compute nodes are patched. Ensure that you specify `--run_datasql 0` if you are patching a node and further nodes remain to be patched.

  This argument can only be specified along with a patching operation on a compute node. If you have patched all of your nodes and you did not specify this argument, you need to manually execute the SQL commands associated with the patch, which typically involves running the `catbundle.sql` script for Oracle Database 11g or the `datapatch` utility for Oracle Database 12c, or later. Refer to the patch documentation for full details.

- `-sshkey` optionally specifies the location of the SSH private key of the `opc` user, which is used to connect to compute nodes in the cluster.

For example:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async 23456789
-instance1=hostname1:/u02/app/oracle/product/12.1.0.2/dbhome_1
-instance2=hostname2:/u02/app/oracle/product/12.1.0.2/dbhome_1
-run_datasql=1 -sshkey=/home/opc/.ssh/id_rsa
```

> **Note:**
>
> To save time and storage space, the patching operation may move a database to another Oracle Home that already contains the patched Oracle Database binaries.

Applying a Patch by Using the dbaascli Command

You can perform a patching operation using the `dbaascli` command as follows:

1. Connect to a compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `dbaascli patch db apply` command:

   - On specific instances:

     ```
     # dbaascli patch db apply --patchid patchid
     --instance1 hostname:oracle_home
     [--dbnames dbname[,dbname2 ...]] [--run_datasql (0|1)]
     ```

   - By specifying only database names:

     ```
     # dbaascli patch db apply --patchid patchid
     --dbnames dbname[,dbname2 ...] [--run_datasql (0|1)] [-alldbs]
     ```

   In the preceding commands:

   - *`patchid`* identifies the patch to be applied.

     > **Note:**
     >
     > For details about how to find the available patch identifiers, see Listing Available Patches.

   - `--instance1` specifies a compute node and Oracle Home directory that is subject to the patching operation. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

     If you use this argument to specify a shared Oracle Home directory and you do not specify the `--dbname` argument, then all of the databases that share the specified Oracle Home are patched.

   - `--dbnames` specifies the database names for the databases that are the target of the patching operation.

If you use this option to patch a database that uses a shared Oracle Home, and you do not specify the `-alldbs` option, then the database is moved to another Oracle Home that contains the patched Oracle Database binaries.

- `-alldbs` patches all of the databases that share the same Oracle Database binaries (Oracle Home) as the databases specified in the `--dbnames` argument.
- `--run_datasql` controls execution of the SQL commands associated with the patch:
  - `--run_datasql 1` executes the SQL commands associated with the patch.
  - `--run_datasql 0` does not execute the SQL commands associated with the patch.

  If the argument is not specified, the patch-related SQL is executed by default; equivalent to `--run_datasql 1`.

  Patch-related SQL should only be executed after all of the compute nodes are patched. Ensure that you specify `--run_datasql 0` if you are patching a node and further nodes remain to be patched.

  This argument can only be specified along with a patching operation on a compute node. If you have patched all of your nodes and you did not specify this argument, you need to manually execute the SQL commands associated with the patch, which typically involves running the `catbundle.sql` script for Oracle Database 11g or the `datapatch` utility for Oracle Database 12c, or later. Refer to the patch documentation for full details.

For example:

```
# dbaascli patch db apply --patchid 23456789
--instance1 hostname1:/u02/app/oracle/product/12.1.0.2/dbhome_1
--run_datasql 1
```

Note:

To save time and storage space, the patching operation may move a database to another Oracle Home that already contains the patched Oracle Database binaries.

How Applying a Patch Impacts the Oracle Homes

This topic describes how applying a patch with `exadbcpatchmulti` or `dbaascli` impacts the Oracle Homes in your environment.

After you apply an Oracle Database patch, the Oracle Home name is updated to reflect the release level of the newly applied patch.

For example, the Oracle Home name for a deployment using Oracle Database 18c with the April 2019 patch set update would be similar to `OraHome100_18600_dbru190416_0`. But, after applying the January 2020 patch set update, the Oracle Home name would change to `OraHome100_18900_dbru200114_0`.

This occurs regardless of whether you use the Oracle Database Cloud Service console to apply, or whether you apply the patch by using the `exadbcpatchmulti` or `dbaascli` utility.

However, when you apply a patch by using the `exadbcpatchmulti` or `dbaascli` utility, the modified Oracle Home name is not updated in the Oracle Database Cloud Service console. Consequently, the updated Oracle Home name is not displayed in the console pages that display details associated with the existing database deployments, and it is not visible in the list of existing Oracle Homes when you use the console to create a new database deployment.

To display the correct Oracle Home details for any existing database deployment, you can use the `dbaascli dbhome info` command. See Viewing Information About Oracle Homes by Using the dbaascli Utility.

To create a database deployment using a patched Oracle Home name that does not appear in the Oracle Database Cloud Service console, you can use the following options:

- **Oracle Homes** — leave the default, which specifies the option to create a new Oracle Home directory location. Do not select an existing Oracle Home location from the list.
- **Oracle Home Name** — specify the full name of the patched Oracle Home that you want to use for the new database deployment.

After you use this procedure, the patched Oracle Home name becomes visible in the list of existing Oracle Homes when you use the console to create another new database deployment.

Listing Applied Patches

You can produce a list of applied patches to determine which patches have been applied.

You can use the `opatch` utility to determine the patches that have been applied to an Oracle Database or Grid Infrastructure installation.

To produce a list of applied patches for an Oracle Database installation, proceed as follows:

1. Connect to a compute node as the `oracle` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Set the ORACLE_HOME variable to the location of the Oracle Database installation you wish to examine. For example:

   ```
   $ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
   ```

3. Execute the `opatch` command with the `lspatches` option:

   ```
   $ $ORACLE_HOME/OPatch/opatch lspatches
   ```

To produce a list of applied patches for Oracle Grid Infrastructure, proceed as follows:

1. Connect to a compute node as the `opc` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Become the `grid` user:

   ```
   $ sudo -s
   # su - grid
   ```

3. Execute the `opatch` command with the `lspatches` option:

   ```
   $ $ORACLE_HOME/OPatch/opatch lspatches
   ```

Rolling Back a Patch or Failed Patch

Rolling Back a Patch or Failed Patch by Using the Oracle Database Cloud Service Console

To roll back the last patch or failed patch attempt by using the Oracle Database Cloud Service console:

1. Go to the Patching page of the database deployment on which you want to roll back a patch:
   a. Open the Oracle Database Cloud Service console.

      For detailed instructions, see Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.
   b. Click the database deployment on which you want to roll back a patch.

      The Oracle Database Cloud Service Overview page is displayed.
   c. Click the Administration tile and then click the Patching tab.

      The Oracle Database Cloud Service Patching page is displayed.
2. Click **Rollback**.

   The Patching page redisplays, showing a status message that your request has been submitted, the Administration tile shows the starting time of the rollback operation, and a **Rolling back...** message replaces the **Rollback** button.

> **Note:**
>
> Rollback operations are performed with a minimum of impact on the functioning of the database. However, during a patch rollback operation the database may be shut down for a short period of time, thus making it inaccessible.

Other Ways to Roll Back a Patch or Failed Patch

For Oracle Database and Oracle Grid Infrastructure patches, you can use the `exadbcpatchmulti` or `dbaascli` utility. See:

- Rolling Back a Patch or Failed Patch by Using the exadbcpatchmulti Command
- Rolling Back a Patch or Failed Patch by Using the `dbaascli` Command

Rolling Back a Patch or Failed Patch by Using the exadbcpatchmulti Command

You can roll back a patch or failed patch attempt by using the `exadbcpatchmulti` command. The command:

- Can be used to roll back a patch on some or all of your compute nodes using one command.
- Can be used to roll back a patch on one or many databases using one command.
- Coordinates multi-node operations in a rolling manner.
- Can execute rollback-related SQL after rolling back the patch on all the compute nodes in the cluster.

You can perform a patch rollback operation using the `exadbcpatchmulti` command as follows:

1. Connect to the compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `exadbcpatchmulti` command with the `-rollback_async` action:

   - On specific instances:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async patchid
     -instance1=hostname1:oracle_home
     [-instance2=hostname2:oracle_home ...]
     [-dbnames=dbname[,dbname2 ...]] [-run_datasql=(0|1)] [-
     sshkey=sshkey_file]
     ```

   - By specifying only database names:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async patchid
     -dbnames=dbname[,dbname2 ...] [-run_datasql=(0|1)] [-alldbs] [-
     sshkey=sshkey_file]
     ```

   In the preceding command:

   - *`patchid`* identifies the patch to be rolled back.
   - `-instance`*`N`* specifies a compute node and Oracle Home directory that is subject to the rollback operation. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

     If you use this argument to specify a shared Oracle Home directory and you do not specify the `-dbnames` argument, then all of the databases that share the specified Oracle Home are rolled back.
   - `-dbnames` specifies the database names for the databases that are the target of the rollback operation.
   - `-alldbs` specifies that you want to roll back all of the databases that share the same Oracle Database binaries (Oracle Home) as the databases specified in the `-dbnames` argument.
   - `-run_datasql` controls execution of the rollback-related SQL commands associated with the patch:
     - `-run_datasql=1` executes the rollback-related SQL commands associated with the patch.
     - `-run_datasql=0` does not execute the rollback-related SQL commands associated with the patch.

If the argument is not specified, the rollback-related SQL is executed by default; equivalent to `-run_datasql=1`.

> **Note:**
>
> – Rollback-related SQL should only be executed after all of the compute nodes are rolled back. Therefore, ensure that you specify `-run_datasql=0` if you are rolling back a subset of nodes and further nodes remain to be rolled back.
>
> – This argument can only be specified along with a rollback operation on a set of compute nodes. Therefore, if you have rolled back all of your nodes and you did not specify this argument, you need to manually execute the SQL commands associated with the rollback operation. Refer to the patch documentation for further details.

- `-sshkey` optionally specifies the location of the SSH private key for the `opc` user, which is used to connect to compute nodes in the cluster.

For example:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async 34567890
-instance1=hostname1:/u02/app/oracle/product/12.1.0.2/dbhome_1
-instance2=hostname2:/u02/app/oracle/product/12.1.0.2/dbhome_1
-run_datasql=1 -sshkey=/home/opc/.ssh/id_rsa
```

Rolling Back a Patch or Failed Patch by Using the `dbaascli` Command

You can perform a patch rollback operation using the `dbaascli` command as follows:

1. Connect to the compute node as the `opc` user and start a command shell as the `root` user.

   For detailed instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Execute the `dbaascli patch db switchback` command:

   - On specific instances:

     ```
     # dbaascli patch db switchback --patchid patchid
     --instance1 hostname:oracle_home
     [--dbnames dbname[,dbname2 ...]] [--run_datasql (0|1)]
     ```

   - By specifying only database names:

     ```
     # dbaascli patch db switchback --patchid patchid
     --dbnames dbname[,dbname2 ...] [--run_datasql (0|1)] [-alldbs]
     ```

   In the preceding commands:

   - *`patchid`* identifies the patch to be rolled back.
   - `--instance1` specifies a compute node and Oracle Home directory that is subject to the rollback operation. In this context, an Oracle Home directory

may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

If you use this argument to specify a shared Oracle Home directory and you do not specify the `--dbnames` argument, then all of the databases that share the specified Oracle Home are rolled back.

- `--dbnames` specifies the database names for the databases that are the target of the rollback operation.
- `-alldbs` specifies that you want to roll back all of the databases that share the same Oracle Database binaries (Oracle Home) as the databases specified in the `--dbnames` argument.
- `--run_datasql` controls execution of the rollback-related SQL commands associated with the patch:
  - `--run_datasql 1` executes the rollback-related SQL commands associated with the patch.
  - `--run_datasql 0` does not execute the rollback-related SQL commands associated with the patch.

  If the argument is not specified, the rollback-related SQL is executed by default; equivalent to `--run_datasql 1`.

> **Note:**
>
> - Rollback-related SQL should only be executed after all of the compute nodes are rolled back. Therefore, ensure that you specify `--run_datasql 0` if you are rolling back a node and further nodes remain to be rolled back.
> - This argument can only be specified along with a rollback operation on a set of compute nodes. Therefore, if you have rolled back all of your nodes and you did not specify this argument, you need to manually execute the SQL commands associated with the rollback operation. Refer to the patch documentation for further details.

For example:

```
# dbaascli patch db switchback --patchid 34567890
-instance1 hostname1:/u02/app/oracle/product/12.1.0.2/dbhome_1
```

8 Configuring Database Features, Database Options, and Companion Products

Oracle Database Exadata Cloud at Customer provides special capabilities for certain Oracle Database features and options and for certain companion products.

Topics

- Using Oracle Data Guard in Exadata Cloud at Customer
- Using Oracle Multitenant in Exadata Cloud at Customer
- Using Oracle GoldenGate Cloud Service with Exadata Cloud at Customer
- Tablespace Encryption
- Managing Huge Pages

Using Oracle Data Guard in Exadata Cloud at Customer

When creating an Oracle Database Exadata Cloud at Customer database deployment, you can create an Oracle Data Guard configuration.
Oracle Data Guard enables Oracle databases to survive disasters and data corruptions by providing a comprehensive set of services that create, maintain, manage, and monitor a standby database. Oracle Data Guard maintains the standby database as a copy of the primary database. If the primary database becomes unavailable because of a planned or an unplanned outage, you can switch the standby database to the primary role, minimizing the downtime associated with the outage.

About Oracle Data Guard in Exadata Cloud at Customer

In general, an Oracle Data Guard configuration contains one primary database, which is the database that is accessed by most of your applications, and up to thirty standby destinations, connected by Oracle Net Services. However, the Oracle Data Guard configuration in Exadata Cloud at Customer specifically includes one primary database and one standby database.

A standby database is a transactionally consistent copy of the primary database. Once created, Oracle Data Guard automatically maintains each standby database by transmitting redo data from the primary database and then applying the redo to the standby database. In an Oracle Data Guard configuration on Exadata Cloud at Customer, the standby database is a physical standby database. A physical standby database provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis. A physical standby database is kept synchronized with the primary database, through Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.

See "Oracle Data Guard Configurations" in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1 or 11.2 for additional information.

Exadata Cloud at Customer also includes Oracle Active Data Guard. Oracle Active Data Guard provides read-only access to the physical standby database while it is synchronized with the primary database, enabling minimal latency between reporting and transactional data. With the Oracle Active Data Guard feature known as real-time query, Redo Apply can be active while the physical standby database is open, thus allowing queries to return results that are identical to what would be returned from the primary database. See "Opening a Physical Standby Database" in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1 or 11.2 for additional information about real-time query.

Before You Create an Oracle Data Guard Configuration

Before you create a database deployment that uses an Oracle Data Guard configuration, you must ensure that both the primary and the standby sites are configured to enable the network communication that is required for Oracle Data Guard.

In summary, network traffic must be able to flow:

- Between both sites (primary and standby).
- In both directions (inbound and outbound at each site).
- On both the administration network (if available) and the client network, and across all combinations (client-to-client, administration-to-administration, client-to-administration, and administration-to-client).
- On both the SSH port (22) and the Oracle Net Services port (1521).

Creating an Oracle Data Guard Configuration

To create an Oracle Data Guard configuration in Exadata Cloud at Customer, make the following choices in the Create Instance wizard:

- On the Instance page, choose **Database Clustering with RAC and Data Guard Standby** as the Database Type.
- On the Instance Details page, complete the **Standby Database** section:
  - **Standby Database Configuration** — influences the location of the Oracle Data Guard standby database. Select from the following options:
    - **High Availability** — indicates that the standby database is placed on a different Exadata system in the same region (data center) as the primary database, thus providing isolation at the Exadata system infrastructure level.
    - **Disaster Recovery** — indicates that the standby database is placed in a different region (data center) from the primary database, thus providing isolation at the Exadata system infrastructure level and geographical separation to protect against catastrophic data center failure.
  - **Exadata System** — select an available Oracle Exadata Database Machine configuration to host the standby database. The list contains the Oracle Exadata Database Machines that are associated with your active Exadata Cloud at Customer instances.

    Your selection is validated when you leave the Instance Details page, and you will be notified if the selection is not consistent with your Standby Database Configuration specification.

> **Note:**
>
> The Exadata System used to host the standby database must exist in the same identity domain as the Exadata System previously specified on the Instance page that is used to host the primary database.

– **Cluster** — select a VM cluster to host the standby database.

  This option is only available if your Exadata Cloud at Customer system environment is enabled to configure multiple VM clusters.

– **Hostnames** — specify one or more compute nodes that you want to host the database instances for the standby database.

> **Note:**
>
> The number of compute nodes that you specify here must match the number of compute nodes that you specified for the primary database.

For further details, see Creating a Database Deployment.

When you make these choices, Exadata Cloud at Customer creates an Oracle Data Guard configuration with a primary database and a single standby database, hosting the databases on two independent Exadata systems.

The configuration includes Oracle Active Data Guard. See "Opening a Physical Standby Database" in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1, or 11.2 for more information on the real-time query and the automatic block media recovery features of Oracle Active Data Guard.

With Exadata Cloud at Customer, you can use the Oracle Database Cloud Service console to perform key Data Guard operations, such as switchover, failover and reinstating a failed primary database, as described in Administering a Data Guard Configuration.

You can also manage primary and standby databases by using the SQL command-line interface or the Oracle Data Guard broker interfaces. The broker provides a command-line interface (DGMGRL) and a graphical user interface through Oracle Enterprise Manager Cloud Control.

Using Oracle Multitenant in Exadata Cloud at Customer

By default, an Oracle Multitenant environment is created when you create an Oracle Database Exadata Cloud at Customer database deployment that uses Oracle Database 12c, or later.

The multitenant architecture enables an Oracle database to function as a multitenant container database (CDB) that includes zero, one, or many pluggable databases (PDBs). A PDB is a portable collection of schemas, schema objects, and non-schema objects that appears to an Oracle Net Services client as a non-CDB. All Oracle databases before Oracle Database 12c were non-CDBs.

Topics

- Creating and Activating a Master Encryption Key for a PDB

- Exporting and Importing a Master Encryption Key for a PDB

Note:

See also: Can I create a non-container database (non-CDB)?

Creating and Activating a Master Encryption Key for a PDB

To use Oracle Transparent Data Encryption (TDE) in a pluggable database (PDB), you must create and activate a master encryption key for the PDB.

In a multitenant environment, each PDB has its own master encryption key which is stored in a single keystore used by all containers.

To determine whether you need to create and activate an encryption key for the PDB, perform the following steps:

1. Invoke SQL*Plus and log in to the database as the `SYS` user with `SYSDBA` privileges.
2. Set the container to the PDB:

   ```
   SQL> ALTER SESSION SET CONTAINER = pdb;
   ```

3. Query `V$ENCRYPTION_WALLET` as follows:

   ```
   SQL> SELECT wrl_parameter, status, wallet_type FROM v$encryption_wallet;
   ```

   If the `STATUS` column contains a value of `OPEN_NO_MASTER_KEY` you need to create and activate the master encryption key.

To create and activate the master encryption key in a PDB, perform the following steps:

1. Set the container to the PDB:

   ```
   SQL> ALTER SESSION SET CONTAINER = pdb;
   ```

2. Create and activate a master encryption key in the PDB by executing the following command:

   ```
   SQL> ADMINISTER KEY MANAGEMENT SET KEY USING TAG 'tag' FORCE KEYSTORE
   IDENTIFIED BY keystore-password WITH BACKUP USING 'backup_identifier';
   ```

   In the above command:

   - *`keystore-password`* is the keystore password. By default, the keystore password is set to the value of the administration password that is specified when the database deployment is created.
   - The optional `USING TAG 'tag'` clause can be used to associate a tag with the new master encryption key.
   - The `WITH BACKUP` clause, and the optional `USING 'backup_identifier'` clause, can be used to create a backup of the keystore before the new master encryption key is created.

   See also `ADMINISTER KEY MANAGEMENT` in *Oracle Database SQL Language Reference* for Release 18 or 12.2.

> **Note:**
>
> To enable key management operations while the keystore is in use, Oracle Database 12c Release 2, and later, includes the `FORCE KEYSTORE` option to the `ADMINISTER KEY MANAGEMENT` command. This option is also available for Oracle Database 12c Release 1 with the October 2017, or later, bundle patch.
>
> If your Oracle Database 12c Release 1 deployment does not have the October 2017, or later, bundle patch installed, you can perform the following alternative steps:
>
> - Close the keystore.
> - Open the password-based keystore.
> - Create and activate a master encryption key in the PDB by using `ADMINISTER KEY MANAGEMENT` without the `FORCE KEYSTORE` option.
> - Update the auto-login keystore by using `ADMINISTER KEY MANAGEMENT` with the `CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE` option.

3. Query `V$ENCRYPTION_WALLET` again to verify that the `STATUS` column is set to `OPEN`:

   ```
   SQL> SELECT wrl_parameter, status, wallet_type FROM v$encryption_wallet;
   ```

4. Query `V$INSTANCE` and take note of the value in the `HOST_NAME` column, which identifies the database server that contains the newly updated keystore files:

   ```
   SQL> SELECT host_name FROM v$instance;
   ```

5. Copy the updated keystore files to all of the other database servers.

   To distribute the updated keystore you must perform the following actions on each database server that does not contain the updated keystore files:

   a. Connect to the root container and query `V$ENCRYPTION_WALLET`. Take note of the keystore location contained in the `WRL_PARAMETER` column:

      ```
      SQL> SELECT wrl_parameter, status FROM v$encryption_wallet;
      ```

   b. Copy the updated keystore files.

      You must copy all of the updated keystore files from a database server that is already been updated. Use the keystore location observed in the `WRL_PARAMETER` column of `V$ENCRYPTION_WALLET`.

   c. Open the updated keystore:

      ```
      SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE open FORCE KEYSTORE IDENTIFIED BY
      keystore-password CONTAINER=all;
      ```

> **Note:**
>
> To enable key management operations while the keystore is in use, Oracle Database 12c Release 2, and later, includes the `FORCE KEYSTORE` option to the `ADMINISTER KEY MANAGEMENT` command. This option is also available for Oracle Database 12c Release 1 with the October 2017, or later, bundle patch.
>
> If your Oracle Database 12c Release 1 deployment does not have the October 2017, or later, bundle patch installed, you can perform the following alternative steps:
>
> - Close the keystore before copying the updated keystore files.
> - Copy the updated keystore files.
> - Open the updated keystore by using `ADMINISTER KEY MANAGEMENT` without the `FORCE KEYSTORE` option.

6. Query `GV$ENCRYPTION_WALLET` to verify that the `STATUS` column is set to `OPEN` across all of the database instances:

```
SQL> SELECT wrl_parameter, status, wallet_type FROM gv$encryption_wallet;
```

Exporting and Importing a Master Encryption Key for a PDB

You must export and import the master encryption key for any encrypted PDBs you plug into your database deployment.

If your source PDB is encrypted, you must export the master encryption key and then import it. In a multitenant environment, each PDB has its own master encryption key which is stored in a single keystore used by all containers.

You can export and import all of the TDE master encryption keys that belong to the PDB by exporting and importing the TDE master encryption keys from within a PDB. Export and import of TDE master encryption keys support the PDB unplug and plug operations. During a PDB unplug and plug, all of the TDE master encryption keys that belong to a PDB, as well as the metadata, are involved.

See "Exporting and Importing TDE Master Encryption Keys for a PDB" in *Oracle Database Advanced Security Guide* for Release 18, 12.2 or 12.1.

See "ADMINISTER KEY MANAGEMENT" in *Oracle Database SQL Language Reference* for Release 18, 12.2 or 12.1.

To export the master encryption keys, perform the following steps:

1. Invoke SQL*Plus and log in to the PDB.
2. Export the master encryption key by executing the following command:

```
SQL> ADMINISTER KEY MANAGEMENT EXPORT ENCRYPTION KEYS WITH SECRET "secret"
TO 'filename' IDENTIFIED BY keystore-password;
```

To import the master encryption key perform the following steps:

1. Invoke SQL*Plus and log in to the PDB.
2. Export the master encryption key by executing the following command:

```
SQL> ADMINISTER KEY MANAGEMENT IMPORT ENCRYPTION KEYS WITH SECRET "secret" FROM
'filename' IDENTIFIED BY keystore-password;
```

Using Oracle GoldenGate Cloud Service with Exadata Cloud at Customer

Oracle GoldenGate Cloud Service is a secure, high performance data integration and replication service that can replicate data in real time from on-premises databases to databases in Oracle Database Exadata Cloud at Customer.
You must create an Exadata Cloud at Customer database deployment that is properly configured for use as a GoldenGate Cloud Service replication target before you create a GoldenGate Cloud Service instance.

To properly configure an Exadata Cloud at Customer database deployment for use as a replication target:

- You must configure the database deployment for use as a replication database.

  You can configure the database deployment for use as a replication database by setting the **Enable Oracle GoldenGate** option on the Instance Details page of the Create Instance wizard.
- The target database must be network accessible on the listener port.

  You must ensure that access to the Oracle Net Listener port is enabled through any firewall that sits between your Exadata Cloud at Customer environment and the GoldenGate Cloud Service instance. Also, ensure that you always use an encrypted Oracle Net Services connection. See Using Network Encryption and Integrity.

Once you have created and properly configured an Exadata Cloud at Customer database deployment for use as a replication target, you can create an Oracle GoldenGate Cloud Service instance that uses it. See Provision an Oracle GoldenGate Cloud Service Instance in *Using Oracle GoldenGate Cloud Service*.

Tablespace Encryption

By default, all new tablespaces that you create in an Exadata Cloud at Customer database are encrypted.

However, the tablespaces that are initially created in conjunction with the database deployment may not be encrypted by default.

- For database deployments that use Oracle Database 12c Release 2 or later, only the `USERS` tablespaces initially created in conjunction with the database deployment are encrypted. No other tablespaces are encrypted including the non-`USERS` tablespaces in:
  - The root container (`CDB$ROOT`).
  - The seed pluggable database (`PDB$SEED`).
  - The first PDB, which is created in conjunction with the database deployment.
- For database deployments that use Oracle Database 12c Release 1 or Oracle Database 11g, none of the tablespaces initially created in conjunction with the database deployment are encrypted.

For further information about the implementation of tablespace encryption in Exadata Cloud at Customer, along with how it impacts various deployment scenarios, see Oracle Database Tablespace Encryption Behavior in Oracle Cloud.

Topics

- Creating Encrypted Tablespaces
- Managing Tablespace Encryption

Creating Encrypted Tablespaces

User-created tablespaces are encrypted by default.

By default, any new tablespaces created by using the SQL `CREATE TABLESPACE` command are encrypted with the AES128 encryption algorithm. You do not need to include the `USING 'encrypt_algorithm'` clause to use the default encryption.

You can specify another supported algorithm by including the `USING 'encrypt_algorithm'` clause in the `CREATE TABLESPACE` command. Supported algorithms are AES256, AES192, AES128, and 3DES168.

Managing Tablespace Encryption

You can manage the software keystore (known as an Oracle wallet in Oracle Database 11*g*), the master encryption key, and control whether encryption is enabled by default.

Managing the Master Encryption Key

Tablespace encryption uses a two-tiered, key-based architecture to transparently encrypt (and decrypt) tablespaces. The master encryption key is stored in an external security module (software keystore). This master encryption key is used to encrypt the tablespace encryption key, which in turn is used to encrypt and decrypt data in the tablespace.

When a database deployment is created on Exadata Cloud at Customer, a local software keystore is created. The keystore is local to the compute nodes and is protected by the administration password specified during the deployment process. The auto-login software keystore is automatically opened when the database is started.

You can change (rotate) the master encryption key by using the `ADMINISTER KEY MANAGEMENT` SQL statement. For example:

```
SQL> ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY USING TAG 'tag'
IDENTIFIED BY password WITH BACKUP USING 'backup';

keystore altered.
```

See "Managing the TDE Master Encryption Key" in *Oracle Database Advanced Security Guide* for Release 18, 12.2 or 12.1 or "Setting and Resetting the Master Encryption Key" in *Oracle Database Advanced Security Administrator's Guide* for Release 11.2.

Controlling Default Tablespace Encryption

The ENCRYPT_NEW_TABLESPACES initialization parameter controls the default encryption of new tablespaces. In Exadata Cloud at Customer databases, this parameter is set to CLOUD_ONLY by default.

Values of this parameter are as follows.

Value	Description
ALWAYS	During creation, tablespaces are transparently encrypted with the AES128 algorithm unless a different algorithm is specified in the ENCRYPTION clause.
CLOUD_ONLY	Tablespaces created in an Exadata Cloud at Customer database are transparently encrypted with the AES128 algorithm unless a different algorithm is specified in the ENCRYPTION clause. For non-cloud databases, tablespaces are only encrypted if the ENCRYPTION clause is specified. This is the default value.
DDL	During creation, tablespaces are not transparently encrypted by default, and are only encrypted if the ENCRYPTION clause is specified.

> **Note:**
>
> With Oracle Database 12c Release 2 (12.2), or later, you can no longer create a new unencrypted tablespace on Exadata Cloud at Customer. An error message is returned if you set ENCRYPT_NEW_TABLESPACES to DDL and issue a CREATE TABLESPACE command without specifying an ENCRYPTION clause.

Managing Huge Pages

Huge Pages provide considerable performance benefits for Oracle Database on systems with large amounts of memory. Oracle Database Exadata Cloud at Customer provides configuration settings that make use of Huge Pages by default; however, you can make manual adjustments to optimize the configuration of Huge Pages.

Huge Pages is a feature integrated into the Linux kernel 2.6. Enabling Huge Pages makes it possible for the operating system to support large memory pages. Using Huge Pages can improve system performance by reducing the amount of system CPU and memory resources required to manage Linux page tables, which store the mapping between virtual and physical memory addresses. For Oracle Databases, using Huge Pages can drastically reduce the number of page table entries associated with the System Global Area (SGA).

On Exadata Cloud at Customer environments, a standard page is 4 KB, while a Huge Page is 2 MB by default. Therefore, an Oracle Database on Exadata Cloud at Customer with a 50 GB SGA requires 13,107,200 standard pages to house the SGA, compared with only 25,600 Huge Pages. The result is much smaller page tables, which require less memory to store and fewer CPU resources to access and manage.

The configuration for Huge Pages varies depending on when the associated Exadata Cloud at Customer instance was created. Two configurations exist:

- Default Configuration of Huge Pages — After Exadata Cloud at Customer release 17.1.5

- Default Configuration of Huge Pages — Prior to Exadata Cloud at Customer release 17.1.5

Default Configuration of Huge Pages — After Exadata Cloud at Customer release 17.1.5

For Exadata Cloud at Customer instances created with release 17.1.5, or later, a fixed portion of system memory is reserved for Huge Pages in the operating system on each compute node, and this allocation can be used by any database deployment. The precise allocation is determined by the Application Type setting that is associated with the starter database deployment:

- **Transactional (OLTP) —** 70% of the system memory is reserved for Huge Pages.
- **Decision Support or Data Warehouse —** 50% of the system memory is reserved for Huge Pages.

The starter database deployment is configured with the instance parameter setting `USE_LARGE_PAGES=ONLY`. This setting forces the SGA to use Huge Pages.

Additional database deployments are configured with the instance parameter setting `USE_LARGE_PAGES=TRUE`. This setting uses available Huge Pages for the SGA and reverts to standard memory pages when the Huge Page allocation is exhausted.

Default Configuration of Huge Pages — Prior to Exadata Cloud at Customer release 17.1.5

For Exadata Cloud at Customer instances created prior to release 17.1.5, Huge Pages are configured only for the starter database deployment, which is the first database deployment that is created after the creation of the Exadata Cloud at Customer instance. The number of Huge Pages configured in the operating system is based on the size of the SGA.

The starter database deployment is configured with the instance parameter setting `USE_LARGE_PAGES=ONLY`. This setting forces the SGA to use Huge Pages.

Additional database deployments are not configured to use Huge Pages by default. To use Huge Pages with additional databases you must perform a manual configuration.

Adjusting the Configuration of Huge Pages

The configuration of Huge Pages for Oracle Database is a two-step process:

- At the operating system level, the overall amount of memory allocated to Huge Pages is controlled by the `vm.nr_hugepages` system parameter entry. This setting exists separately on each compute node in the environment, and it is strongly recommended that the setting is consistent across all of the compute nodes.

  To immediately adjust the Huge Page allocation, use the following command on each compute node as the `root` user:

  ```
  # sysctl -w vm.nr_hugepages=value
  ```

  In the command, `value` is the number of Huge Pages that you want to allocate.

On Exadata Cloud at Customer environments, each Huge Page is 2 MB by default. Therefore, to allocate 50 GB of memory to Huge Pages you can execute the following command:

```
# sysctl -w vm.nr_hugepages=25600
```

To persist the Huge Page setting across reboots, you must add an entry in the system parameter file. If your compute nodes use Oracle Linux 6, or earlier, then you can add the entry to the system parameter file at `/etc/sysctl.conf`. For example:

```
# echo "vm.nr_hugepages=25600" >> /etc/sysctl.conf
```

If your compute nodes use Oracle Linux 7, or later, then you can add the entry to a user-specific parameter file under `/etc/sysctl.d/`. For example:

```
# touch /etc/sysctl.d/01-custom.conf
# echo "vm.nr_hugepages=25600" >> /etc/sysctl.d/01-custom.conf
```

Check `/etc/oracle-release` to determine the current Oracle Linux release on your compute nodes.

- At the Oracle Database level, the use of Huge Pages is controlled by the `USE_LARGE_PAGES` instance parameter setting. This setting applies to each database instance in a clustered database and it is strongly recommended that the setting is consistent across all of the database instances associated with a database deployment. The following options are available:
  - `TRUE` — specifies that the database instance can use Huge Pages if they are available. For all versions of Oracle Database after 11.2.0.3, Oracle allocates as much of the SGA as it can using Huge Pages. When the Huge Page allocation is exhausted, standard memory pages are used.
  - `FALSE` — specifies that the database instance does not use Huge Pages. This setting is generally not recommended if Huge Pages are available.
  - `ONLY` — specifies that the database instance must use Huge Pages. With this setting, the database instance fails to start if the entire SGA cannot be accommodated in Huge Pages.

You must ensure that the overall configuration works if you make any adjustments at either the operating system or Oracle Database level.

For more information, see the *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems* for Release 11.2 or 12.1 for a general overview of Huge Pages and more information about configuring Huge Pages. Also, see `USE_LARGE_PAGES` in the *Oracle Database Reference* for Release 11.2, 12.1 or 12.2.

9

Migrating Oracle Databases to Exadata Cloud at Customer

You can migrate your on-premises Oracle databases to Oracle Database Exadata Cloud at Customer using various different approaches based on different tools and technologies.

Topics

- Choosing a Migration Method
- Migration Methods

Choosing a Migration Method

Various different migration methods exist, and each migration method is associated with different benefits, opportunities, requirements and limitations.

Migration Considerations

Some migration methods apply only if specific characteristics of the source (on-premises) and target (Exadata Cloud at Customer) databases match or are compatible. And even when multiple migration methods are technically feasible, other non-technical factors can affect which method you choose.

For example, Exadata Cloud at Customer uses a little-endian platform, so if you are migrating from a big-endian platform, some physical migration approaches are not feasible or require extra processing to achieve. Also, the frequency and length of the available maintenance windows is often a key consideration in determining which migration approaches are applicable.

Some of the characteristics and factors to consider when choosing a migration method are:

- Source and target database versions
- Source platform and operating system
- Source database character set
- Quantity of data, including indexes
- Methods available for data transportation
- Database features and data types used
- Storage for data staging
- Acceptable length of system outage
- Network bandwidth

When choosing the right migration approach, you should clearly define what you need to migrate. For example, do you need to migrate a whole database or a whole tablespace or just a selection of database objects? This will help you to choose an approach that avoids considerable wasted effort in order to migrate data that is not required in the target database.

You should also weigh up the short-term requirement to perform the migration with the long-term impact of using the selected migration approach. Specifically, you may need to rule out what seems to be an easy and convenient migration approach if the resulting database configuration is sub-optimal. For example, Exadata performs best with an ASM AU size of 4 MB and database extents that are a multiple of 4 MB. If the source database extent sizes are not a multiple of 4 MB and it is impractical to reorganize the database before migration, then you might favor a migration approach that allows you to reorganize the database during the migration. If you choose an approach that does not allow the extents to be reorganized, you may be able to deliver a quicker and easier migration; however, you may also end up paying an ongoing performance penalty.

It is also worth noting that sometimes it makes sense to extend a migration method by performing additional data processing, or combine multiple migration methods, to deliver the best result. For example, your situation might determine that Transportable Tablespaces are a convenient way to migrate data into Exadata Cloud at Customer. However, the physical organization of the data in the Transportable Tablespaces may not be ideal for Exadata, so you may choose to redefine the tables by using a series of `CREATE ... AS SELECT` SQL commands, or to reload the data into fresh segments using Data Pump.

Finally, Oracle can offer professional services to assist with all aspects of data migration to Exadata Cloud at Customer. You can engage Oracle to provide specific assistance for your migration efforts, or you can get Oracle to plan and execute the migration for you.

Determining Applicable Methods

To determine which migration methods might be applicable to your migration scenario, gather the following information.

1. The database version of your source database:
2. The architecture of the database, for source databases that use Oracle Database 12c, or later:
   - Container database (CDB). A CDB can support one (single-tenant) or more (multitenant) pluggable databases (PDBs).
   - Non-CDB
3. Your source database host platform and endian format:

   Query `V$DATABASE` to identify the platform name for your source database.

   Platforms are either little-endian or big-endian depending on the byte ordering that they use. Query `V$TRANSPORTABLE_PLATFORM` to view all platforms that support cross-platform tablespace transport, along with the endian format of each platform.

   Exadata Cloud at Customer uses Linux x86–64, which is little endian.
4. The database character set of your source database:

   By default, databases are configured to use the AL32UTF8 database character set on Exadata Cloud at Customer.
5. The target database version that you are migrating to on Exadata Cloud at Customer:

   With Exadata Cloud at Customer, databases that use Oracle Database 12c, or later, are configured to use the CDB architecture.

After gathering this information, consider the following migration method outlines to determine the feasibility of each method to your specific scenario.

Migration Methods

Many methods exist to migrate Oracle databases to Oracle Database Exadata Cloud at Customer.

Which of these methods apply to a given migration scenario depends on several factors, including the version, character set, and platform endian format of the source and target databases.

> **Note:**
>
> This guide does not cover every available migration method. Rather, it focuses the most commonly applicable methods available using tools and technologies that are readily available in Oracle Database. Alternative approaches, such as using data integration technologies or custom code are not considered.

Topics

- Conventional RMAN Backup and Recovery
- Conventional Data Pump Export and Import
- Transportable Tablespaces
- Data Pump Full Transportable Export and Import
- Transportable Tablespaces with Cross-Platform Incremental Backup
- Transportable Database
- Data Guard Physical Standby
- Advanced Data Guard Migration Options
- Unplugging and Plugging a Pluggable Database
- Plugging in a Non-CDB
- Cloning a Remote PDB or Non-CDB

Conventional RMAN Backup and Recovery

You can migrate data to Exadata Cloud at Customer by using Oracle Recovery Manager (RMAN). RMAN facilitates a physical migration approach, which is favored in migration scenarios where physical database re-organization is not necessary.

RMAN is an Oracle Database client that performs backup and recovery tasks on Oracle databases. You can use RMAN to migrate data to Exadata Cloud at Customer simply by transferring a backup of your source database to Exadata Cloud at Customer and restoring it there. You can also restore from backups stored in Oracle Database Backup Cloud Service.

If your source database resides on Linux x86–64 (like Exadata Cloud at Customer), and it uses Oracle Database 11g Release 2, or later, you can use RMAN to restore a backup of your source database on Exadata Cloud at Customer.

If you use conventional RMAN backup and recovery to create a database on Exadata Cloud at Customer, note that it is not automatically registered with the Exadata Cloud at Customer console and cannot use cloud tooling for database administration. To enable the Exadata Cloud at Customer console and cloud tooling, see Registering a Database.

RMAN also provides an active database duplication feature, which performs duplication over a network link between the source and target databases. You must consider the size of your source database, and the speed and reliability of your network connection to determine the feasibility of this approach.

For information about using RMAN, see *Oracle Database Backup and Recovery User's Guide* for Release 18, 12.2, 12.1, or 11.2.

Note:

RMAN provides other options if your source database platform differs from Exadata Cloud at Customer:

- If your source database resides on another little-endian platform, you can use RMAN to transport the entire database to Exadata Cloud at Customer. See Transportable Database.
- If your source database resides on a big-endian platform, then you can only use RMAN along with the Transportable Tablespaces feature of Oracle Database. This option can only be used to migrate your data tablespaces, not administrative tablespaces, such as SYSTEM and SYSAUX. See Transportable Tablespaces.

Conventional Data Pump Export and Import

You can use this method regardless of the endian format and database character set of the source database. You can also use Data Pump to migrate data between different versions of Oracle Database. This method is simple to implement, provides the broadest cross-platform support and enables you to physically re-organize your target database; however, the time and resources required for export and import may rule out this approach for situations with large databases or limited timeframes.

Conventional Data Pump Export and Import uses the Data Pump utilities, `expdp` and `impdp`, to unload (export) and load (import) Oracle Database data and metadata. During an Export, a copy of the source data, and metadata, is written to a binary dump file. After the dump file is transported to the target system, its contents can be imported into another Oracle database. Because of this architecture, Data Pump provides broad support for data migration between different platforms, different Oracle Database versions and databases with different character sets.

In conjunction with using this approach, database administrators can alter the physical properties of database objects in the target database. For example, administrators can optimize table and index extent sizes to suit the characteristics of the target database environment. Therefore, conventional Data Pump Export and Import is well suited for situations where you need to physically re-organize the target database.

In addition to working on whole databases, conventional Data Pump Export and Import provides the flexibility to export and import specific tables, schemas or tablespaces, which makes it well suited for situations where you do not want to migrate the entire database. This capability also enables you to migrate a database in pieces if such an approach is logically valid.

Because of the processing required during export and import, this approach can be more time and resource intensive than other migration approaches. Therefore, other approaches might be preferred for migrations that require minimal downtime.

To migrate your source database, tablespace, schema, or table to Oracle Database Exadata Cloud at Customer using conventional Data Pump Export and Import, perform these tasks:

1. On the source database host, use Data Pump Export to unload part or all of the source database to a dump file.
2. Transfer the resulting dump file to an Exadata Cloud at Customer compute node.
3. On the Exadata Cloud at Customer compute node, use Data Pump Import to load the target database.
4. After verifying that the dump file contents has been imported successfully, you can delete the dump file.

See Oracle Data Pump in *Oracle Database Utilities* for Release 18, 12.2, 12.1, or 11.2.

Conventional Data Pump Export and Import: Example

This example provides a step-by-step demonstration of the tasks required to migrate a schema from an existing Oracle database to Oracle Database Exadata Cloud at Customer.

This example illustrates a schema-mode export and import. The same general procedure applies for a full database, tablespace, or table export and import.

In this example, the source database is on a Linux host.

1. On the source database host, invoke Data Pump Export to export the schema.
   a. On the source database host, create an operating system directory to store the output from the export operation.

      ```
      $ mkdir /u01/app/oracle/admin/orcl/dpdump/for_cloud
      ```

   b. On the source database host, invoke SQL*Plus and log in to the source database as the SYSTEM user.

      ```
      $ sqlplus system
      Enter password: <enter the password for the SYSTEM user>
      ```

   c. Create a directory object in the source database to reference the operating system directory.

      ```
      SQL> CREATE DIRECTORY dp_for_cloud AS '/u01/app/oracle/admin/orcl/dpdump/
      for_cloud';
      ```

   d. Exit from SQL*Plus.
   e. On the source database host, invoke Data Pump Export as the SYSTEM user or another user with the DATAPUMP_EXP_FULL_DATABASE role and export the required schema. In this example, the schema owner is FSOWNER. Provide the password for the user when prompted.

      ```
      $ expdp system SCHEMAS=fsowner DIRECTORY=dp_for_cloud
      ```

2. Transfer the dump file to the target Exadata Cloud at Customer compute node.

In this example, the dump file is copied across the network by using the SCP utility.

a. On the target Exadata Cloud at Customer compute node, create a directory that you will copy the dump file to.

   Choose an appropriate location based on the size of the file that will be transferred.

   ```
   $ mkdir /u01/app/oracle/admin/ORCL/dpdump/from_source
   ```

b. Before using the `scp` command to copy the export dump file, make sure the SSH private key that provides access to the target Exadata Cloud at Customer compute node is available on your source host. For more information about SSH keys, see About Network Access to Exadata Cloud at Customer.

c. On the source database host, use the SCP utility to transfer the dump file to the target Exadata Cloud at Customer compute node.

   ```
   $ scp -i private_key_file \
   /u01/app/oracle/admin/orcl/dpdump/for_cloud/expdat.dmp \
   oracle@compute_node_IP_address:/u01/app/oracle/admin/ORCL/dpdump/
   from_source
   ```

3. On the target Exadata Cloud at Customer compute node, invoke Data Pump Import and import the data into the database.

   a. On the Exadata Cloud at Customer compute node, invoke SQL*Plus and log in to the database as the `SYSTEM` user.

      ```
      $ sqlplus system
      Enter password: <enter the password for the SYSTEM user>
      ```

   b. Create a directory object in the Exadata Cloud at Customer database.

      ```
      SQL> CREATE DIRECTORY dp_from_source AS '/u01/app/oracle/admin/ORCL/
      dpdump/from_source';
      ```

   c. If they do not exist, create the tablespace(s) for the objects that will be imported.

   d. Exit from SQL*Plus.

   e. On the Exadata Cloud at Customer compute node, invoke Data Pump Import and connect to the database. Import the data into the database.

      ```
      $ impdp system SCHEMAS=fsowner DIRECTORY=dp_from_source
      ```

4. After verifying that the data has been imported successfully, you can delete the `expdat.dmp` file.

Transportable Tablespaces

This method provides broad cross-platform migration support, and limited support for source and destination databases with different character sets. You can also use the transportable tablespace feature to migrate data to a later version of Oracle Database. This method is often chosen when migrating between platforms with different endian formats, or in cases where physical re-organization is not necessary.

The transportable tablespace method is generally much faster than a conventional export and import of the same data because you do not have to unload and reload the data. Rather, the source data files are transported to the destination system and attached to the target database. For basic migrations using this feature, you use Data

Pump to export and import only the metadata associated with the objects in the tablespace.

The transportable tablespace method provides broad cross-platform support with some limitations. If you are migrating from a big-endian platform to Exadata Cloud at Customer (little-endian), extra processing is required to perform a conversion. Ideally, the source and target database character sets should be the same (AL32UTF8). However, there are limited situations where another source character set can be supported. Administrative tablespaces, such as SYSTEM and SYSAUX, cannot be included in a transportable tablespace set. For details regarding the requirements and limitations for transportable tablespaces, see Transporting Tablespaces Between Databases in *Oracle Database Administrator's Guide* for Release 18, 12.2, 12.1, or 11.2.

To perform a basic migration using the transportable tablespace method, you perform these tasks:

1. Select a self-contained set of tablespaces. That is, there should be no references from objects inside the set of tablespaces to objects outside the set of tablespaces.

   For example, there should be no:

   - Indexes for tables outside the tablespace set.
   - Partitioned tables having partitions outside the tablespace set.
   - Referential integrity constraints that point to objects outside the tablespace set.
   - LOB columns that point to LOBs outside the tablespace set.

   You can use the `TRANSPORT_SET_CHECK` procedure in the `DBMS_TTS` package to determine whether a set of tablespaces is self-contained.
2. On the source database, place the set of tablespaces into read-only mode.
3. On the source database host, execute Data Pump Export to unload the metadata associated with the tablespace set.
4. Transfer the Data Pump Export dump file and the tablespace datafiles to an Exadata Cloud at Customer compute node.
5. On the Exadata Cloud at Customer compute node, load the tablespace data files into ASM and Exadata Storage Server. If required, perform an endian format conversion at this stage.

   You can load and convert the data files by using the RMAN `CONVERT` command, or the `PUT_FILE` procedure in the `DBMS_FILE_TRANSFER` package.
6. On the Exadata Cloud at Customer compute node, use Data Pump Import to load the metadata associated with the tablespace set.
7. Set the tablespaces on the Exadata Cloud at Customer database to read-write mode.
8. After verifying that the data has been imported successfully, you can delete the dump file.

As an alternative to this basic migration procedure, you can use RMAN to migrate a transportable tablespace set. By using RMAN you can avoid the requirement to place the source tablespaces into read-only mode. You can also use a database backup as the migration source, and you can specify a target point in time, SCN, or restore point during your recovery window and transport tablespace data as it existed at that time. See Creating Transportable Tablespace Sets in *Oracle Database Backup and Recovery User's Guide* for Release 18, 12.2, 12.1, or 11.2.

Data Pump Transportable Tablespace: Example

This example provides a step-by-step demonstration of the tasks required to migrate tablespaces from an existing Oracle database to Oracle Database Exadata Cloud at Customer.

This example performs a migration of the FSDATA and FSINDEX tablespaces, which contain objects owned by the FSUSER database user.

In this example, the source database is on a big-endian AIX-based host.

1. Verify that the source tablespace set is self-contained.

   a. On the source database host, invoke SQL*Plus and log in to the source database as the SYSTEM user.

   ```
   $ sqlplus system
   Enter password: <enter the password for the SYSTEM user>
   ```

   b. Use the TRANSPORT_SET_CHECK procedure in the DBMS_TTS package to determine if the tablespace set is self-contained.

   ```
   SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('FSDATA,FSINDEX', TRUE);
   ```

   c. Examine the TRANSPORT_SET_VIOLATIONS view. If the tablespace set examined by DBMS_TTS.TRANSPORT_SET_CHECK is self-contained, this view is empty. Otherwise, you must resolve any violation before you proceed.

   ```
   SQL> SELECT * FROM TRANSPORT_SET_VIOLATIONS;
   ```

2. On the source database, place the set of tablespaces that will be transported into read-only mode.

   ```
   SQL> ALTER TABLESPACE fsindex READ ONLY;
   SQL> ALTER TABLESPACE fsdata READ ONLY;
   ```

3. On the source database host, execute Data Pump Export to unload the metadata associated with the tablespace set.

   a. Create an operating system directory to store the output from the export operation.

   ```
   $ mkdir /u01/app/oracle/admin/orcl/dpdump/for_cloud
   ```

   b. Create a directory object in the source database to reference the operating system directory.

   ```
   SQL> CREATE DIRECTORY dp_for_cloud AS '/u01/app/oracle/admin/orcl/dpdump/
   for_cloud';
   ```

   c. Determine the name(s) of the data files that belong to the FSDATA and FSINDEX tablespaces by querying DBA_DATA_FILES. These files will also be listed in the export output.

   ```
   SQL> SELECT file_name FROM dba_data_files
     2  WHERE tablespace_name in ('FSDATA','FSINDEX');

   FILE_NAME
   -----------------------------------------------------------------
   /u01/app/oracle/oradata/orcl/fsdata01.dbf
   /u01/app/oracle/oradata/orcl/fsindex01.dbf
   ```

   d. Invoke Data Pump Export to perform the transportable tablespace export.

On the source database host, invoke Data Pump Export and connect to the source database. Export the source tablespaces using the `TRANSPORT_TABLESPACES` option. Provide the password for the `SYSTEM` user when prompted.

```
$ expdp system TRANSPORT_TABLESPACES=fsdata,fsindex TRANSPORT_FULL_CHECK=YES
DIRECTORY=dp_for_cloud
```

4. Transfer the dump file and tablespace data files to the target Exadata Cloud at Customer compute node.

   In this example, the files are copied across the network by using the SCP utility.

   a. On the target Exadata Cloud at Customer compute node, create a directory that you will copy the dump file to.

      Choose an appropriate location based on the size of the file that will be transferred.

      ```
      $ mkdir /u01/app/oracle/admin/ORCL/dpdump/from_source
      ```

   b. Before using the `scp` command to copy the export dump file, make sure the SSH private key that provides access to the target Exadata Cloud at Customer compute node is available on your source host. For more information about SSH keys, see About Network Access to Exadata Cloud at Customer.

   c. On the source database host, use the SCP utility to transfer the dump file and tablespace data files to the target Exadata Cloud at Customer compute node.

      ```
      $ scp -i private_key_file \
      /u01/app/oracle/admin/orcl/dpdump/for_cloud/expdat.dmp \
      oracle@compute_node_IP_address:/u01/app/oracle/admin/ORCL/dpdump/from_source

      $ scp -i private_key_file \
      /u01/app/oracle/oradata/orcl/fsdata01.dbf \

      oracle@compute_node_IP_address:/u01/app/oracle/admin/ORCL/dpdump/from_source

      $ scp -i private_key_file \
      /u01/app/oracle/oradata/orcl/fsindex01.dbf \
      oracle@compute_node_IP_address:/u01/app/oracle/admin/ORCL/dpdump/from_source
      ```

5. On the target Exadata Cloud at Customer compute node, convert and load the tablespace data files into ASM and Exadata Storage Server.

   In this example, the data files are converted to little-endian format and loaded into ASM by using the RMAN `CONVERT` command.

   a. Invoke RMAN and log in to the target database as the `SYSTEM` user.

      ```
      $ rman target system
      target database password: <enter the password for the SYSTEM user>
      ```

   b. Use the `CONVERT` command to convert and load the data files into ASM.

      Take note of the ASM file names for your converted data files.

      ```
      RMAN> convert datafile
      2> '/u01/app/oracle/admin/ORCL/dpdump/from_source/fsdata01.dbf',
      3> '/u01/app/oracle/admin/ORCL/dpdump/from_source/fsindex01.dbf'
      4> to platform="Linux x86 64-bit"
      5> from platform="AIX-Based Systems (64-bit)"
      6> format '+DATA_SYSNAME';

      Starting converstion at target at ...
      ...
      input file name=/u01/app/oracle/admin/ORCL/dpdump/from_source/fsdata01.dbf
      ```

```
converted datafile=+DATA_SYSNAME/ORCL/datafile/fsdata01.277.821069105
...

input file name=/u01/app/oracle/admin/ORCL/dpdump/from_source/
fsindex01.dbf
converted datafile=+DATA_SYSNAME/ORCL/datafile/fsindex01.278.419052810

...
```

6. On the target Exadata Cloud at Customer compute node, use Data Pump Import to load the metadata associated with the tablespace set.

   a. Invoke SQL*Plus and log in to the target database as the `SYSTEM` user.

   b. Create a directory object in the target database that points to the operating system directory containing the Data Pump dump file.

      ```
      SQL> CREATE DIRECTORY dp_from_source AS '/u01/app/oracle/admin/ORCL/
      dpdump/from_source';
      ```

   c. If they do not already exist, create user accounts for the owners of the objects that will be imported into the target database.

      ```
      SQL> CREATE USER fsowner
        2  PROFILE default
        3  IDENTIFIED BY fspass
        4  TEMPORARY TABLESPACE temp
        5  ACCOUNT UNLOCK;
      ```

   d. Invoke Data Pump Import and import the tablespace metadata into the target database. Use the `TRANSPORT_DATAFILES` option and specify the file names for the data files that are converted and loaded into ASM.

      ```
      $ impdp system DIRECTORY=dp_from_source \
      TRANSPORT_DATAFILES='+DATA_SYSNAME/ORCL/datafile/
      fsdata01.277.821069105', \
      '+DATA_SYSNAME/ORCL/datafile/fsindex01.278.419052810'
      ```

7. On the target database, set the `FSDATA` and `FSINDEX` tablespaces to `READ WRITE` mode.

   ```
   SQL> ALTER TABLESPACE fsdata READ WRITE;
   Tablespace altered.
   SQL> ALTER TABLESPACE fsindex READ WRITE;
   Tablespace altered.
   ```

8. After verifying that the data has been imported successfully, you can delete the `expdat.dmp` dump file.

Data Pump Full Transportable Export and Import

Like transportable tablespaces, this method provides broad cross-platform migration support, limited support for source and destination databases with different character sets, and it can be used to migrate data to a later version of Oracle Database. It simplifies the process of migrating complete databases and leverages the transportable tablespace feature where possible.

Data Pump full transportable export and import is an extension of basic transportable tablespaces, which can be used to migrate the entire contents of your source database to Exadata Cloud at Customer.

You perform a full transportable export by specifying the parameters FULL=YES and TRANSPORTABLE=ALWAYS when you execute the Data Pump Export. When a full transportable export is performed, a mix of data movement methods are used:

- Objects residing in transportable tablespaces have only their metadata unloaded into the dump file and the data is moved when you copy the data files to the target database.
- Objects residing in non-transportable tablespaces (for example, SYSTEM and SYSAUX) have both their metadata and data unloaded into the dump file.

For details regarding the requirements and limitations for full transportable export, see Transporting Databases in *Oracle Database Administrator's Guide* for Release 18, 12.2, 12.1, or 11.2.

To migrate your source database to Exadata Cloud at Customer using the Data Pump full transportable export and import, you perform these tasks:

1. On the source database, place all the user-defined tablespaces into read-only mode.
2. On the source database host, execute Data Pump Export and perform a full transportable export.

   To perform a full transportable export, Specify the parameters FULL=YES and TRANSPORTABLE=ALWAYS.
3. Transfer the Data Pump Export dump file and the datafiles for all of the user-defined tablespaces to an Exadata Cloud at Customer compute node.
4. On the Exadata Cloud at Customer compute node, load the user-defined tablespace data files into ASM and Exadata Storage Server. If required, perform an endian format conversion at this stage.

   You can load and convert the data files by using the RMAN CONVERT command, or the PUT_FILE procedure in the DBMS_FILE_TRANSFER package.
5. On the Exadata Cloud at Customer compute node, use Data Pump Import to load the metadata associated with the user-defined tablespaces, along with the data and metadata exported from the source database's non-transportable tablespaces.
6. Set the user-defined tablespaces on the Exadata Cloud at Customer database to read-write mode.
7. After verifying that the data has been imported successfully, you can delete the dump file.

Transportable Tablespaces with Cross-Platform Incremental Backup

This method uses transportable tablespaces in conjunction with cross-platform incremental backup. By using this combination, the downtime required for the migration can be reduced significantly; however, this comes at the cost of using more administration and processing resources overall. It also provides the benefits associated with transportable tablespaces; namely, broad cross-platform migration support, limited support for source and destination databases with different character sets, and the ability to migrate data to a later version of Oracle Database.

A migration using transportable tablespaces in conjunction with cross-platform incremental backup in accomplished in three phases:

1. Preparation.
   a. Use RMAN to backup your source tablespaces.
   b. Transfer the backups to an Exadata Cloud at Customer compute node.

   c. On the Exadata Cloud at Customer compute node, load the tablespace data files into ASM and Exadata Storage Server. If required, perform an endian format conversion at this stage.

      You can load and convert the data files by using the RMAN `CONVERT` command, or the `PUT_FILE` procedure in the `DBMS_FILE_TRANSFER` package.

2. Roll forward.

   a. Use RMAN to create an incremental backup on the source system.

   b. Transfer the incremental backup to an Exadata Cloud at Customer compute node.

   c. On the Exadata Cloud at Customer compute node, use RMAN to convert the incremental backup to the target system endian format and apply it to the target data files.

   Repeat the roll forward tasks until the target database is almost up to date with the source database.

   This method relies on the notion that the incremental backups can be taken, transported and applied quicker than the time period covered by each backup. If this is true, each backup will get successively smaller and the target system will catch up with the source system. If the incremental backups take too long to generate and apply, the target system will never catch up and this method cannot be used.

3. Final roll forward and metadata transport.

   a. On the source database, place the source tablespaces into read-only mode.

   b. On the source database host, use RMAN to create the final incremental backup.

   c. On the source database host, execute Data Pump Export to unload the metadata associated with the tablespace set.

   d. Transfer the final incremental backup and the Data Pump dump file to an Exadata Cloud at Customer compute node.

   e. On the Exadata Cloud at Customer compute node, use RMAN to convert the final incremental backup to the target system endian format and apply it to the target data files.

   f. On the Exadata Cloud at Customer compute node, use Data Pump Import to load the metadata associated with the tablespace set.

   g. Set the tablespaces on the Exadata Cloud at Customer database to read-write mode.

By using this method, no downtime is incurred in the preparation and roll forward phases, which is where most of the data transportation occurs. Downtime is only incurred in the final roll forward and metadata transport phase. Consequently, the required downtime depends on the rate of change and the amount of metadata in the source database, rather than its overall size. Therefore, using transportable tablespaces in conjunction with cross-platform incremental backup is a good candidate for situations where data file transfer and conversion would otherwise require unacceptably long downtime.

Note that cross-platform incremental backup does not affect the amount of time it takes to perform metadata export and import. So databases that have very large amounts of metadata will see limited benefit if the migration time is dominated by metadata operations, not data file transfer and conversion.

For information about this approach, including specific requirements and limitations, see *Reduce Transportable Tablespace Downtime using Cross-Platform Incremental Backup* for Oracle Database 11g and 12c.

Transportable Database

This method works in conjunction with RMAN to migrate whole databases between platforms that share the same endian format. The result is a block-for-block replica of the source database. Consequently, the transportable database method is useful in cases where it is not necessary to physically re-organize the source database.

Though conceptually similar, the transportable database method is substantially different from transportable tablespaces. The transportable database method involves copying an entire database, including the `SYSTEM` tablespace, from one platform to another. Because the whole database is copied, containment checks are unnecessary and no Data Pump export and import are required. RMAN is used to perform the required backup, conversion and restoration operations, and you can also use backups stored in Oracle Database Backup Cloud Service.

The transportable database method only works across platforms that share the same endian format. Therefore, your source database must reside on a little-endian platform in order to facilitate transport to Exadata Cloud at Customer.

When you use the transportable database method, the result is a block-for-block copy of the source database, and the target database automatically uses the database character set of the source database. You should carefully consider whether the physical organization and character set of your source database are suitable for use in conjunction with Exadata Cloud at Customer before selecting this approach.

To perform a migration using the transportable database method, you perform a different set of tasks depending on:

- The type of backup used. You can choose between:
  - Image copies, which are file copies generated with the RMAN `BACKUP AS COPY` command, an operating system command such as the UNIX `cp` command, or by the Oracle archiver process.
  - An RMAN backup set, which is one or more binary files that contain backup data in a format that can only be created or restored by RMAN. In general, Oracle recommends using backup sets because they are optimized for use with RMAN.
- The system where conversion is performed. You can choose between:
  - The source system. You might select this option in order to prepare the database as much as possible before using Exadata Cloud at Customer.
  - The target system. You might select this option to minimize any migration impact on the source system.

See Transporting Data Across Platforms in *Oracle Database Backup and Recovery User's Guide* for Release 18, 12.2, 12.1, or 11.2.

Data Guard Physical Standby

An Oracle Data Guard physical standby database is a block-for-block replica of a primary database. You can use Data Guard to replicate your source database to Exadata Cloud at Customer. Afterward, you can decouple the databases and use the physical standby as your new master. You can use this method in conjunction with source databases from a selection of little-endian platforms.

Oracle Data Guard provides a comprehensive set of features that create, maintain, manage, and monitor standby databases. Data Guard is primarily used to maintain standby databases for the purposes of disaster recovery. During normal operations, the standby database is constantly updated with changes from the primary database. If the primary database fails for any reason, the standby database can be used to support the application workload.

Oracle Data Guard can also be used to facilitate data migration. You can start by creating a standby database in the target environment. After the standby is created and brought up to date with the primary database, you can perform a switchover and make the standby the new primary database. Finally, you can decouple the databases and continue using the original standby as your migrated database.

To host a Data Guard physical standby database on Exadata Cloud at Customer, your source database must reside on Linux x86–64 (the same as Exadata Cloud at Customer) or a compatible little-endian platform. Compatible platforms include Linux x86, Windows x86 (32–bit or 64–bit) and Solaris x86. See What differences are allowed between a Primary Database and a Data Guard Physical Standby Database for details about Data Guard support for different platforms. Also, the primary and standby databases must have the same compatibility setting, which means that your source database must be upgraded to a version of Oracle Database supported by Exadata Cloud at Customer before Data Guard is configured.

When you instantiate the Data Guard physical standby database, you use a block-for-block copy of the primary database, and the standby database automatically uses the database character set of the primary database. You should carefully consider whether the physical organization and character set of your source database is suitable for use in conjunction with Exadata Cloud at Customer before selecting this approach.

To perform a database migration using a Data Guard physical standby database, you perform these tasks:

1. Create a database deployment on Exadata Cloud at Customer that will eventually incorporate your migrated database.
2. Manually delete the Exadata Cloud at Customer database that is created in conjunction with the database deployment in step 1.
3. Create the standby database on Exadata Cloud at Customer using the database deployment created in step 1. This will be the migrated database.
4. Configure Transparent Data Encryption (TDE).
5. Configure automatic backups for the migrated database.
6. Perform a Data Guard switchover, so that the migrated database assumes the primary database role in the Data Guard configuration.
7. Register the migrated database with the Exadata Cloud at Customer tooling and Web console.

8. Decouple the databases by stopping the Data Guard redo apply services and removing the initialization parameter settings for Data Guard. At this point, you can decommission your original source database.

For detailed instructions, see Migration to Exadata Cloud using Simple Data Guard Approach with Minimal Downtime.

See also Creating a Physical Standby Database in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1, or 11.2.

Advanced Data Guard Migration Options

You can use advanced migration options based on Oracle Data Guard to perform a database upgrade during the migration while providing limited downtime and easy fallback if issues arise.

You can use a Transient Logical Rolling Upgrade to migrate and upgrade a database. However, to use this technique the source database must be compatible with the prerequisite conditions for creating a logical standby database. This method provides the lowest downtime as the upgrade does not impact the primary database. The upgrade is performed on the migration destination database prior to switching over, leaving the original source database open for use. After completing the upgrade, you can switch over to the upgraded database and all applications can use the migrated and upgraded database.

If your source database is not compatible with the prerequisite conditions for creating a logical standby database, you can use an alternative method where the upgrade is performed to the migrated database after it is switched over to the primary role. Using this method requires longer downtime since no changes can be applied back to the original source database after the upgrade commences.

For details about both methods, see Migration to Exadata Cloud Using Advanced Data Guard Approach with Minimal Downtime.

See also Creating a Logical Standby Database in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1, or 11.2.

Unplugging and Plugging a Pluggable Database

You can use this method only if the source platform is little endian, and the database character set is AL32UTF8 or a compatible subset. It uses the multitenant architecture in Oracle Database 12c, or later, to enable easy migration of pluggable databases (PDBs).

You can migrate a PDB to Oracle Database Exadata Cloud at Customer by unplugging the PDB from the source container database (CDB) and plugging it into a CDB on Exadata Cloud at Customer.

This approach is attractive because of its simplicity. However, the specific requirements for this method make it suitable in fewer situations than other methods, such as transportable tablespaces. The requirements for unplugging and plugging a PDB include:

- The source database must be a PDB, which implies that the source database version is 12.1 or later.
- The source and target platform must have the same endian format, which is little-endian for Exadata Cloud at Customer.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud at Customer. Alternatively, the PDB character set must by

a multibyte character set that is a binary subset of AL32UTF8, such as UTF8 for example; however, complications may arise if the different character sets have different maximum character widths.

To migrate a PDB to Exadata Cloud at Customer by unplugging and plugging a PDB, you perform these tasks:

1. On the source database host, connect to the root container of the source CDB as a user with the `SYSDBA` or `SYSOPER` administrative privilege, and:
   a. Close the source PDB.
   b. Execute the `ALTER PLUGGABLE DATABASE ... UNPLUG INTO` command to generate an XML file containing the PDB metadata.
2. Transfer the XML file and the PDB data files to an Exadata Cloud at Customer compute node.
3. On the Exadata Cloud at Customer compute node, connect to the root container of the target CDB as a user with the `SYSDBA` or `SYSOPER` administrative privilege, and:
   a. Optionally, execute the `DBMS_PDB.CHECK_PLUG_COMPATIBILITY` function to verify that your PDB is compatible with Exadata Cloud at Customer.
   b. Execute the `CREATE PLUGGABLE DATABASE` command to plug in the PDB.
4. Open the target PDB in read-write mode by executing the `ALTER PLUGGABLE DATABASE ... OPEN READ WRITE` command.

See Creating a PDB by Plugging an Unplugged PDB into a CDB in *Oracle Database Administrator's Guide* for Release 18, 12.2, or 12.1.

Alternatively, you can use RMAN to assist in the PDB migration process. By using RMAN you can avoid the requirement to place the source PDB into read-only mode. However, using RMAN requires that you use the `BACKUP FOR TRANSPORT` or `BACKUP TO PLATFORM` command to create a transportable backup of your source PDB. Therefore, using this method requires additional space and processing resources to create the required backup. See Performing Cross-Platform Transport of PDBs in *Oracle Database Backup and Recovery User's Guide* for Release 18, 12.2, or 12.1.

Plugging in a Non-CDB

You can use this method only if the source platform is little endian, and the database character set is AL32UTF8 or a compatible subset. It uses the multitenant architecture in Oracle Database 12c, or later, and provides a way to consolidate several non-CDBs into a multitenant database on Exadata Cloud at Customer.

You can migrate a non-CDB to Oracle Database Exadata Cloud at Customer by plugging the non-CDB into a CDB on Exadata Cloud at Customer. This method is similar to unplugging and plugging a PDB, and has similar requirements and restrictions:

- The source database must be version is 12.1 or later.
- The source and target platform must have the same endian format, which is little-endian for Exadata Cloud at Customer.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud at Customer. Alternatively, the PDB character set must by a multibyte character set that is a binary subset of AL32UTF8, such as

UTF8 for example; however, complications may arise if the different character sets have different maximum character widths.

To migrate to Exadata Cloud at Customer by plugging in a non-CDB, you perform these tasks:

1. On the source database host, invoke SQL*Plus, connect to the source database as a user with the `SYSDBA` or `SYSOPER` administrative privilege, and:
   a. Set the source database to read-only mode.
   b. Execute the `DBMS_PDB.DESCRIBE` procedure to generate an XML file that describes the database files of the non-CDB.
   c. Shut down the source database.
2. Transfer the XML file and the source database data files to an Exadata Cloud at Customer compute node.
3. On the Exadata Cloud at Customer compute node, connect to the root container of the target CDB as a user with the `SYSDBA` or `SYSOPER` administrative privilege, and execute the `CREATE PLUGGABLE DATABASE` command to plug in the source database.
4. Connect to the target PDB as a `SYSDBA` user and execute the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` script to delete unnecessary metadata from the `SYSTEM` tablespace of the new PDB.
5. Open the target PDB in read-write mode by executing the `ALTER PLUGGABLE DATABASE ... OPEN READ WRITE` command.

See Creating a PDB Using a Non-CDB in *Oracle Database Administrator's Guide* for Release 18, 12.2, or 12.1.

Cloning a Remote PDB or Non-CDB

You can use this method only if the source platform is little endian, and the database character set is AL32UTF8 or a compatible subset. It uses the multitenant architecture in Oracle Database 12c, or later, in conjunction with a database link to clone the source database directly over the network. The process is simple; however, it may not be feasible for large databases or situations involving slow or unreliable network links.

Cloning a Remote PDB or Non-CDB is very similar to unplugging and plugging in a PDB or plugging in a Non-CDB. The major difference is that remote cloning uses a database link to transfer the data as part of running the `CREATE PLUGGABLE DATABASE` command. As a result, remote cloning is even simpler than preparing, transporting and plugging in a PDB. However, since remote cloning depends on transporting the data over a database link, you must consider the size of your source database and the speed of your Internet connection in order to determine whether it is a feasible migration approach in your case.

Cloning a Remote PDB or Non-CDB has similar requirements and restrictions compared with unplugging and plugging in a PDB or plugging in a Non-CDB:

- The source database must be version is 12.1 or later.
- The source and target platform must have the same endian format, which is little-endian for Exadata Cloud at Customer.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud at Customer. Alternatively, the PDB character set must by a multibyte character set that is a binary subset of AL32UTF8, such as UTF8 for

example; however, complications may arise if the different character sets have different maximum character widths.

Furthermore, if you are creating a PDB by cloning a non-CDB, then both the target CDB and the source non-CDB must be running Oracle Database 12c version 12.1.0.2, or later.

To migrate a PDB or Non-CDB to Exadata Cloud at Customer using the remote cloning method, you perform these tasks:

1. Place the source PDB or Non-CDB in read-only mode.
2. On the target CDB, create a database link that enables a connection to the source database.
3. On the target CDB, run the `CREATE PLUGGABLE DATABASE` statement and specify the source PDB or the source non-CDB in the `FROM` clause.

   For example, assuming that you have a database link to a source PDB or Non-CDB named `mylink` and the name of your source database is `mydb`, then the following statement creates a cloned PDB named `newpdb`:

   ```
   SQL> CREATE PLUGGABLE DATABASE newpdb FROM mydb@mylink;
   ```

4. If your source is a non-CDB, connect to the target PDB as a `SYSDBA` user and execute the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` script to delete unnecessary metadata from the `SYSTEM` tablespace of the new PDB.
5. Open the target PDB in read-write mode by executing the `ALTER PLUGGABLE DATABASE ... OPEN READ WRITE` command.

See Cloning a Remote PDB or Non-CDB in *Oracle Database Administrator's Guide* for Release 18, 12.2, or 12.1.

10

Frequently Asked Questions for Exadata Cloud at Customer

This section provides answers to frequently asked questions (FAQs) for Oracle Database Exadata Cloud at Customer.

- Who is the service right for?
- Does the Exadata Cloud at Customer support external Oracle Net Services (SQL*Net) connections?
- How is storage allocated?
- How are users defined?
- How can I secure my data?
- Can I load additional third-party software?
- Can I create databases that use non-Exadata storage?
- Is there any additional charge for support?
- What database options are included or available?
- Is this service enabled to use Application Express?
- Can I create a non-container database (non-CDB)?

Who is the service right for?

Exadata Cloud at Customer is an ideal fit for:

- Running business-critical production OLTP or analytic databases at almost any scale without incurring the capital expenditure and complexity of maintaining the underlying IT infrastructure. Oracle Database In-Memory enables ultra-high-performance analytics to be run on dedicated analytic databases or directly on OLTP databases.
- Consolidating a variety of workloads using multiple Oracle databases or Oracle Multitenant.
- Maintaining synchronized Oracle standby or replica databases for disaster recovery and/or query offloading using Oracle Active Data Guard or Oracle GoldenGate.
- Quickly provisioning high-performance Oracle databases for ad-hoc business reasons such as feature development, functionality testing, application certification, proof-of-concept, and try-before-buy.
- Executing time-sensitive large-scale business applications such as launching a web-based marketing campaign, running loyalty programs, and rolling out new business initiatives.

Does the Exadata Cloud at Customer support external Oracle Net Services (SQL*Net) connections?

Yes. Exadata Cloud at Customer supports direct external connections using Oracle Net Services. See Connecting Remotely to the Database by Using Oracle Net Services.

How is storage allocated?

The amount of storage space allocated to Exadata Cloud at Customer is fixed and is based on the system configuration options that you selected when you commenced your service subscription. See Exadata System Configuration and Exadata Storage Configuration.

How are users defined?

Users are defined at various different levels:

- Each Exadata Cloud at Customer deployment comes under the ownership of an administrative user for the overall environment. Additional administrator user accounts can be defined by using the Oracle Database Cloud Service console.
- Each compute node has pre-defined operating system (OS) user accounts, including the `oracle` and `opc` user accounts. Additional OS user accounts may be defined by using the native OS utilities available on each compute node.
- Each Oracle database contains pre-defined database user accounts, including `SYS`, `SYSTEM` and others. Additional database user accounts may be defined by using the SQL `CREATE USER` command or by using the facilities provided by database administration tools such as Enterprise Manager or SQL Developer.

How can I secure my data?

You use standard Oracle Database security features to manage user accounts, authentication, privileges and roles, application security, encryption, network traffic, and auditing. Furthermore, depending on your service configuration and security requirements, you may be able to leverage the advanced security features provided by Oracle Advanced Security, Oracle Label Security, Oracle Real Application Security and Oracle Database Vault.

Can I load additional third-party software?

Customers may load additional software on the database servers. However, to ensure the best performance, Oracle discourages adding software except for agents, such as backup agents and security monitoring agents, on the database servers. See Oracle Exadata Rack Restrictions in *Oracle Exadata Database Machine System Overview*.

Can I create databases that use non-Exadata storage?

Creation of databases on compute nodes that store database data files on non-Exadata storage is not supported. This applies Exadata Cloud at Customer on Oracle Cloud and Exadata Cloud at Customer. For example, it is not supported to create a database on Exadata Cloud at Customer that uses a ZFS file server for data file storage.

Is there any additional charge for support?

No, support is included in the subscription price for this service.

What database options are included or available?

Exadata Cloud at Customer is equipped with Oracle Database Enterprise Edition - Extreme Performance. This provides all the features of Oracle Database Enterprise Edition, plus all the database enterprise management packs and all the Enterprise

Edition options, such as Oracle Database In-Memory and Oracle Real Application Clusters (RAC).

> **Note:**
>
> Some options are dependent on the Oracle Database version in use. For example, Oracle Database In-Memory can only be used with Oracle Database software version 12.1.0.2, or later.

Is this service enabled to use Application Express?

No, by default Oracle Application Express is not enabled on Exadata Cloud at Customer deployments. However, you may manually customize your databases to configure and enable Oracle Application Express.

Can I create a non-container database (non-CDB)?

By default, Exadata Cloud at Customer uses the multitenant architecture for all databases that are created with Oracle Database 12c, or later. However, if you require a non-container database (non-CDB) you can use the procedure at Creating non-CDB databases using Oracle Database 12c on Exadata Cloud at Customer.

A
Characteristics of a Newly Created Deployment

This section provides information about the content and configuration of a newly created database deployment on Oracle Database Exadata Cloud at Customer.

Topics

- Linux User Accounts
- Locations of Installed Software
- Oracle Database Characteristics
- Location of Diagnostic and Log Files
- Oracle Data Guard Configuration

Linux User Accounts

This section provides information about Linux user accounts that are provisioned on Oracle Database Exadata Cloud at Customer.

Every Exadata Cloud at Customer compute node is provisioned with the following operating system user accounts.

User	Description
`opc`	The system administrator account you use in conjunction with the `sudo` command to gain `root` user access to your compute nodes.
`oracle`	The Oracle Database administrator account you use to access the system and perform database administration tasks. A home directory, `/home/oracle`, is created for this user. This user cannot use the `sudo` command to perform operations that require `root` user access.
`root`	The root administrator for the system. You do not have direct access to this account. To perform operations that require `root` user access, execute `sudo -s` as the `opc` user.
`grid`	The Oracle Grid Infrastructure administrator account you use to perform ASM and Oracle Clusterware administration tasks. A home directory, `/home/grid`, is created for this user. This user cannot use the `sudo` command to perform operations that require `root` user access. You do not have direct access to this account. To perform operations that require `grid` user access, execute `sudo -s` as the `opc` user to get `root` access, and then execute `su - grid` to become the `grid` user.

The following environment variable settings are created for the `opc`, `oracle` and `grid` users.

Variable	Description
`HOME`	The home directory of the user, either `/home/opc`, `/home/oracle` or `/home/grid`.
`HOSTNAME`	The host name of the compute node.

Variable	Description
LANG	The system language, en_US.UTF-8.
PATH	The paths to search for executables; set to include: • /sbin • /usr/sbin • /bin • /usr/bin • $HOME
SHELL	The default shell, /bin/bash.
USER	The user name, either opc, oracle or grid.

In addition, the following environment variable settings are created for the grid user only.

Variable	Description
ORACLE_HOME	The Oracle Grid Infrastructure home directory: /u01/app/12.1.0.2/grid, /u01/app/12.2.0.1/grid, or /u01/app/18.0.0/grid.
ORACLE_SID	The ASM system identifier (SID) associated with the ASM instance on the compute node: +ASM*N*, where *N* is a unique number (1, 2, 3, and so on).
PATH	Additional paths to search for executables: • $ORACLE_HOME/bin • $ORACLE_HOME/OPatch

Locations of Installed Software

This section provides information about the locations of installed software on a newly created Oracle Database Exadata Cloud at Customer database deployment.

When a database deployment is created on Exadata Cloud at Customer, software is installed in the following locations.

Software	Installation Location
Oracle Database	$ORACLE_HOME: • Oracle Database: Depending on the selected Oracle Database software release, the Oracle Home directory resides under /u02/app/oracle/product/18.0.0, /u02/app/oracle/product/12.2.0, /u02/app/oracle/product/12.1.0, or /u02/app/oracle/product/11.2.0. • Oracle Grid Infrastructure: /u01/app/18.0.0/grid, /u01/app/12.1.0.2/grid, or /u01/app/12.2.0.1/grid.

Oracle Database Characteristics

When a database deployment is created on Oracle Database Exadata Cloud at Customer, an Oracle database is created using information provided in the Create Instance wizard:

Wizard Page and Field	How Used When Creating the Database
Software Release on the Instance page	Determines which version of Oracle Database is used.
DB Name on the Instance Details page	The database system identifier (SID) of the database.
PDB Name on the Instance Details page (only for Oracle Database 12c, or later)	The name of the default pluggable database (PDB) that is created in the container database.
Administrator Password on the Instance Details page	The password used for the SYS and SYSTEM database users.
Application Type on the Instance Details page (for a starter database deployment only)	Adjusts Oracle Database parameter settings: • **Transactional (OLTP)** — configures the database for a transactional workload, with a bias towards high volumes of random data access. • **Decision Support or Data Warehouse** — configures the database for a decision support or data warehouse workload, with a bias towards large data scanning operations.
Character Set on the Instance Details page	The database character set.
National Character Set on the Instance Details page	The database national character set.
Database Type on the Instance page	Specifies the type of database deployment, which determines key configuration attributes such as whether or not Oracle Data Guard is configured, for example.
Hostnames on the Instance Details page	Specifies the placement of database instances on the compute nodes.
Oracle Home Name on the Instance Details page	Specifies the placement of the Oracle Home directory (containing the Oracle Database binaries) on the compute nodes.

Location of Diagnostic and Log Files

When a database deployment is created on Oracle Database Exadata Cloud at Customer, log files from the creation operation are stored in subdirectories of `/var/opt/oracle/log`.

By default, Oracle Database trace files and log files are stored in subdirectories of `/u02/app/oracle/diag`. Oracle Grid Infrastructure trace files and log files are stored in subdirectories of `/u01/app/grid/diag`.

Oracle Data Guard Configuration

The Oracle Data Guard configuration in an Oracle Database Exadata Cloud at Customer deployment includes a primary database and a single physical standby database.
The Oracle Data Guard configuration in an Oracle Database Exadata Cloud at Customer database deployment has the following characteristics:

- Standby Database Type: Physical. The Oracle Data Guard configuration includes a physical standby database. A physical standby database provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis.
- Data Protection Mode: Maximum Performance. The Oracle Data Guard configuration uses maximum performance protection mode. This protection mode provides the highest level of data protection that is possible without affecting the performance of a primary database. This is accomplished by allowing transactions to commit as soon as all redo data generated by those transactions has been written to the online log. See Oracle Data Guard Protection Modes in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1 or 11.2 for more information on data protection modes.
- Redo Transport Services Mode: Asynchronous (ASYNC). Redo transport services control the automated transfer of redo data from the primary database to one or more archival destinations in an Oracle Data Guard configuration. The Oracle Data Guard configuration is set to asynchronous (`ASYNC` attribute of the `LOG_ARCHIVE_DEST_`*`n`* initialization parameter). The asynchronous redo transport mode transmits redo data asynchronously with respect to transaction commitment. A transaction can commit without waiting for the redo generated by that transaction to be successfully sent to any redo transport destination that uses the asynchronous redo transport mode. See Introduction to Redo Transport Services in *Oracle Data Guard Concepts and Administration* for Release 18, 12.2, 12.1 or 11.2 for more information on redo transport services modes.

B

Oracle Cloud Pages for Administering Exadata Cloud at Customer

This section provides information about what you can do and what you see on each of the Oracle Cloud pages for administering Oracle Database Exadata Cloud at Customer.

Topics

- Instances Page
- Activity Page
- SSH Access Page
- Overview Page
- Backup Page
- Patching Page
- Snapshots Page
- Create Instance: Instance Page
- Create Instance: Instance Details Page
- Create Instance: Confirmation Page

Instances Page

The Oracle Database Cloud Service Instances page displays all deployments on Oracle Database Exadata Cloud at Customer.

Topics

- What You Can Do From the Oracle Database Cloud Service Instances Page
- What You See on the Oracle Database Cloud Service Instances Page

What You Can Do From the Oracle Database Cloud Service Instances Page

Use the Oracle Database Cloud Service Instances page to perform the tasks described in the following topics:

- Viewing All Database Deployments
- Creating a Database Deployment
- Viewing Detailed Information for a Database Deployment
- Deleting a Database Deployment

What You See on the Oracle Database Cloud Service Instances Page

The following table describes the key information shown on the Oracle Database Cloud Service Instances page.

Element	Description
☰ navigation menu	Navigation menu providing access to other Oracle Cloud services in the identity domain.
username ▼	User menu providing access to help, accessibility options, console version information and sign-out.
Dashboard	Click to go to the My Services Dashboard page.
Users	Click to go to the My Services Users page.
Notifications	Click to go to the My Services Notifications page.
Activity	Click to go to the Activity Page.
SSH Access	Click to go to the SSH Access Page.
Welcome!	Click to go to the Oracle Database Cloud Service console Welcome page.
REST APIs	Click to go to the API Catalog Cloud Service.
☰ menu after **REST APIs**	Menu that provides access to Platform Services.
Instances, **OCPUs**, **Memory**, **Storage** and **Public IPs**	Summary of resources being used: • **Instances** — Total number of configured deployments. • **OCPUs** — Total number of Oracle CPUs allocated across all deployments. • **Memory** — Total amount of compute node memory allocated across all deployments. • **Storage** — Total amount of storage allocated across all deployments. • **Public IPs** — Number of public IP addresses allocated across all deployments.
Enter a full or partial service name 🔍	Enter a full or partial deployment name to filter the list of deployments to include only those that contain the string in their name.
Create Instance	Click to create a new database deployment on Exadata Cloud at Customer. See Creating a Database Deployment.
☁	Click to view details for the database deployment or clone deployment.
📷	Click to view details for the snapshot master deployment.
Status	Status of the deployment if it is not running. Status values include “In Progress”, “Maintenance”, “Stopped”, and “Terminating”.
Version	Version of Oracle Database configured on the deployment. For example: 12.1.0.2 or 11.2.0.4.
Edition	Software edition of Oracle Database configured on the deployment.
Created On or **Submitted On**	Date when the deployment was created. During the creation process, the date when the creation request was submitted.
Exadata System	Name of the Exadata Cloud at Customer instance.
OCPUs	Number of Oracle CPUs associated with the deployment.
Memory	Amount of compute node memory in GBs associated with the deployment.

Element	Description
Storage	Amount of storage in GBs associated with the deployment.
☰ menu for each deployment	Menu that provides the following options: • **Open EM Console** — Open the database console, either Enterprise Manager Database Express or Enterprise Manager 11g Database Control. • **SSH Access** — Add an SSH public key. See Adding an SSH Public Key. • **Create Database Clone** — Create a clone database deployment associated with this snapshot master. This option is only available if the deployment is a snapshot master. See Creating a Clone Database Deployment from a Snapshot Master. • **Update Exadata IORM** — Update settings for Exadata I/O resource management (IORM). See Using Exadata I/O Resource Management. • **Delete** — Delete the deployment. See Deleting a Database Deployment or Deleting a Snapshot Master.
Service create and delete history	Listing of attempts to create or delete a deployment. Click the triangle icon next to the title to view the history listing.

Activity Page

The Activity page displays activities for all Oracle Database Exadata Cloud at Customer deployments in your identity domain. You can restrict the list of activities displayed using search filters.

Topics

- What You Can Do From the Activity Page
- What You See on the Activity Page

What You Can Do From the Activity Page

Use the Activity page to view operations for all Exadata Cloud at Customer deployments in your identity domain.

You can use the page’s Search Activity Log section to filter the list of displayed operations based on:

- The time the operation was started
- The status of the operation
- The name of the deployment on which the operation was performed
- The type of the operation

In the table of results, you can:

- Click any column heading to sort the table by that column.
- Click the triangle at the start of an operation’s row to see more details about that operation.

What You See on the Activity Page

The following table describes the key information shown on the Activity page.

Element	Description
Start Time Range	Filters activity results to include only operations started within a specified time range. The range defaults to the previous 24 hours.
Status	Filters operations by status of the operation: • All • Scheduled • Running • Succeeded • Failed You can select any subset of status types. The default value is `All`.
Service Name	Filters the activity results to include operations only for the specified service instance. You can enter a full or partial service instance name.
Service Type	Filters the activity results to include operations only for instances of the specified service type. The default value is the current cloud service.
Operation	Filters the activity results to include selected types of operations. You can select any subset of the given operations. The default value is `All`.
Search	Searches for activities by applying the filters specified by the Start Time Range, Status, Service Name, Service Type and Operation fields, and displays activity results in the table.
Reset	Clears the Start Time Range and Service Name fields, and returns the Status and Operation fields to their default values.
Results per page	Specifies the number of results you want to view per page. The default value is `10`.
▶	Displays status messages for the given operation. Clicking on the resulting downward arrow hides the status messages.
Service Name	Shows the name of the service instance and its identity domain: *service_instance*:*identity_domain* You can sort the column in ascending or descending order.
Service Type	Shows the type of cloud service for this instance. You can sort the column in ascending or descending order.
Operation	Shows the type of operation performed on the service instance. You can sort the column in ascending or descending order.
Status	Shows the status of the operation performed on the service instance. You can sort the column in ascending or descending order.
Start Time	Shows the time the operation started. You can sort the column in ascending or descending order.
End Time	Shows the time the operation ended, if the operation is complete. You can sort the column in ascending or descending order.
Initiated By	Shows the user that initiated the operation. The user can be any user in the identity domain who initiated the operation or, for certain operations such as automated backup, System. You can sort the column in ascending or descending order.

SSH Access Page

The SSH Access page enables you to view and add SSH public keys to Oracle Database Exadata Cloud at Customer deployments in your identity domain. You can restrict the list of deployments displayed using search filters.

Topics

- What You Can Do From the Activity Page
- What You See on the Activity Page

What You Can Do From the SSH Access Page

Use the SSH Access page to view and add SSH public keys to Exadata Cloud at Customer deployments in your identity domain.

You can use the page’s Search section to filter the list of displayed deployments based on deployment name.

In the table of results, you can:

- Click any column heading to sort the table by that column.
- Click the triangle at the start of a deployment’s row to see more details.

What You See on the SSH Access Page

The following table describes the key information shown on the SSH Access page.

Element	Description
Service Name	Filters the results to include SSH keys only for the specified deployment. You can enter a full or partial deployment name.
Service Type	Filters the results to include SSH keys only for deployments of the specified service type. The default value is the current cloud service.
Search	Searches for SSH keys by applying the filters specified by the Service Name and Service Type fields, and displays the results in the table.
Results per page	Specifies the number of results you want to view per page. The default value is `10`.
▶	Displays a description of an item in the results table. Clicking on the resulting downward arrow hides the description.
Service Name	Shows the name of the deployment.
Service Type	Shows the type of cloud service for this deployment.
Last Update	Shows the most recent time the SSH keys for this deployment were updated.

Element	Description
Actions	Click the **Add New Key** button to add a new SSH public key to this deployment. The **Add New Key** overlay is displayed with its **Key value** field displaying the deployment's most recent SSH public key. Specify the new public key using one of the following methods: • Select **Upload a new SSH Public Key value** and click **Choose File** to select a file that contains the public key. • Select **Key value**. Delete the current key value and paste the new public key into the text area. Make sure the value does not contain line breaks or end with a line break.

Overview Page

The Oracle Database Cloud Service Overview page displays overview information for an Oracle Database Exadata Cloud at Customer database deployment.

The following tables describe the elements and options available in the various areas of the Overview page:

- What You See in the Banner Area
- What You See in the Tiles Area
- What You See in the Page Content Area

What You See in the Banner Area

The following table describes the elements and options available in the banner area at the top of the page.

Element	Description
☰ menu	Navigation menu providing access to other Oracle Cloud services in the identity domain.
username ▼	User menu providing access to help, accessibility options, console version information and sign-out.
Dashboard	Click to go to the My Services Dashboard page.
Users	Click to go to the My Services Users page.
Notifications	Click to go to the My Services Notifications page.
▶ (next to the "Oracle Database Cloud Service" link)	Click to see details about the database deployment: description, identity domain, subscription type, user who created the deployment, and when the deployment was created.
Oracle Database Cloud Service link	Click to return to the Instances Page.

Element	Description
☰ (next to the deployment’s name)	Deployment menu that provides the following options: • **Open EM Console** — Open the database console for the deployment, either Enterprise Manager Database Express or Enterprise Manager 11g Database Control. • **Switchover** — Start a switchover operation. (Available only for deployments with a Data Guard configuration.) • **Failover** — Start a manual failover operation. (Available only for deployments with a Data Guard configuration.) • **Reinstate** — Start an operation to reinstate a failed primary as the standby. (Available only for deployments with a Data Guard configuration.) • **SSH Access** — Add an SSH public key to the deployment. See Adding an SSH Public Key. • **Replace Database using Backup** — Replace the database on the deployment using an existing backup stored by Database Backup Cloud Service. See Creating a Database Deployment Using a Cloud Backup. • **View Activity** — Go to the Activity Page to view activities performed on this deployment.

What You See in the Tiles Area

The following table describes the elements and options available in the tiles area at the side of the page.

Element	Description
Overview tile	The current tile, highlighted to indicate that you are viewing the Overview page.
Administration tile	Click to access these pages for the deployment: • Backup Page • Patching Page • Snapshots Page

What You See in the Page Content Area

The following table describes the elements and options available in the main content area of the page.

Element	Description
⟳	Click to refresh the page.
Service Overview section	Displays a summary box followed by information about the deployment. The summary box shows high-level information about the Exadata Cloud Service instance hosting the deployment: compute nodes, OCPUs, memory, and storage. Following the summary box is a listing of information about the deployment, including Oracle Database version, Software edition, backup destination, overall status, and so on. Click the **Show more...** link to see even more information about the deployment.

Element	Description
Resources section	Contains an entry for each compute node of the deployment. Each entry displays information about the compute node and provides a menu to perform actions on the compute node.
⚙	(Available only for deployments with a Data Guard configuration.) Click to poll the status of the Data Guard configuration on the deployment's compute nodes and refresh the information on this page.
☰ (for each compute node)	Compute node menu that provides the following options: • **Start** — Start a stopped compute node. See Stopping, Starting, and Restarting Compute Nodes . • **Stop** — Stop a compute node. See Stopping, Starting, and Restarting Compute Nodes . • **Restart** — Restart a compute node. See Stopping, Starting, and Restarting Compute Nodes . • **Switchover** — Start a switchover operation. (Available only for deployments with a Data Guard configuration.) • **Failover** — Start a manual failover operation. (Available only for deployments with a Data Guard configuration.) • **Reinstate** — Start an operation to reinstate a failed primary as the standby. (Available only for deployments with a Data Guard configuration.)
Data Guard Metrics	(Available only for deployments with a Data Guard configuration.) Displays metrics about the Data Guard configuration.
Network Information	Displays network host name and IP address information.

Backup Page

You use the Backup page to manage backup and recovery of a particular database deployment.

What You See on the Oracle Database Cloud Service Backup Page

The following table describes the key information shown on the Oracle Database Cloud Service Backup page.

Element	Description
Backup Now	Click to create a full backup of the database deployment.
Recover	Click to recover the database deployment to the latest backup or to a specific point in time.
Configure Backups	Click to update the credentials for backing up to cloud storage.
☰ (for each available backup)	Menu that provides the **Recover** option. Choose this option to recover to the given backup.
Recovery History	Listing of recovery operations on the database deployment. Click the triangle icon next to the title to view the listing.

Patching Page

You use the Patching page to view available patches, initiate a patching process, and view details of the last patching process for a particular database deployment.

What You See on the Oracle Database Cloud Service Patching Page

The following table describes the key information shown on the Oracle Database Cloud Service Patching page.

Element	Description
Available Patches	A list of patches you can apply to the deployment.
☰ (for each listed patch)	**Menu** icon provides the following options for the patch: • **Precheck** — Check whether the patch can be successfully applied to the deployment. • **Patch** — Apply the patch to the deployment.
Details of Last Patching Activity	Expand to see a description of the actions taken during the last patching operation.
Rollback	Click to roll back the last patching operation. See Rolling Back a Patch or Failed Patch by Using the Oracle Database Cloud Service Console.

Snapshots Page

You use the Snapshots page to create Exadata snapshot masters, which are read-only copies of an existing database. You can then use these Exadata snapshot masters to quickly and easily create space-efficient clone databases, which are useful for development, testing, or other purposes that require a transient database.

What You See on the Oracle Database Cloud Service Snapshots Page

The following table describes the key information shown on the Oracle Database Cloud Service Snapshots page.

Element	Description
Create Snapshot Master	Click to create an Exadata snapshot master, which can be used to create clone database deployments.
☰ (for each available snapshot master)	Menu that provides the following options: • **Create Database Clone** — Create a clone deployment. • **Delete** — Delete a snapshot master.

Create Instance: Instance Page

Create Instance: Instance is the first page in the wizard you use to create a new database deployment, as described in Creating a Database Deployment.

What You See in the Navigation Area

Element	Description
Cancel	Click to cancel the Create Instance wizard without creating a new database deployment.
Next>	Click to advance to the Create Instance: Instance Details page.

What You See in the Page Content Area

The following table describes the key information shown on the Create Instance: Instance page.

Element	Description
Instance Name	The name for the new database deployment.
Description	(Optional) A description for the new database deployment.
Notification Email	(Optional) An email address that receives notifications from the database deployment creation operation.
Exadata System	This list contains the Oracle Exadata Database Machines that are associated with your existing subscriptions. Exadata Cloud at Customer offers several configurations, as described in Exadata System Configuration.
Hostnames	Specifies the compute nodes that host database instances for the database deployment.
Tags	(Optional) Specifies tags for the database deployment. Tagging enables you to group database deployments that share similar characteristics or are used for a similar purpose.
Service Level	The service level for the new deployment: • **Oracle Database Exadata Cloud Service** — is the only Service Level setting compatible with Exadata Cloud at Customer. All other service level options relate to Oracle Database Cloud Service, which does not use Exadata.
Software Release	The release version of Oracle Database for the new deployment: • **Oracle Database 11g Release 2** • **Oracle Database 12c Release 1** • **Oracle Database 12c Release 2** • **Oracle Database 18c** • **Oracle Database 19c** See Oracle Database Software Release and Oracle Grid Infrastructure Software Release.
Software Edition	The Oracle Database software package for the new deployment: • **Enterprise Edition - Extreme Performance** — is the only Software Edition setting compatible with Exadata Cloud at Customer.
Database Type	The type of deployment to create: • **Database Clustering with RAC** • **Database Clustering with RAC and Data Guard Standby** See Oracle Database Type.

Create Instance: Instance Details Page

Create Instance: Instance Details is a page in the Create Instance wizard you use to create a new database deployment. For more information, see Creating a Database Deployment.

The following tables describe the key information shown on the Create Instance: Instance Details page:

- What You See in the Navigation Area
- What You See in the Database Configuration Section
- What You See in the Backup and Recovery Configuration Section
- What You See in the Initialize Data From Backup Section
- What You See in the Standby Database Section

What You See in the Navigation Area

Element	Description
<Previous	Click to return to the Create Instance: Instance page.
Cancel	Click to cancel the Create Instance wizard without creating a new database deployment.
Next>	Click to advance to the Create Instance: Confirmation page.

What You See in the Database Configuration Section

Element	Description
Cluster	Specifies the VM cluster to host the database deployment. This option is only available if your Exadata Cloud at Customer system environment is enabled to configure multiple VM clusters.
DB Name	The name for the database instance.
PDB Name	The name for the default pluggable database (PDB). This option is available only for Oracle Database 12c, or later. This option is not available if Create Instance from Existing Backup is set to Yes.
Administration Password **Confirm Password**	The administration password, which is used to configure administration accounts and functions in the database deployment, including the password for the Oracle Database SYS and SYSTEM users.
Oracle Homes	Specifies the option to create a new Oracle Home directory location, or an existing Oracle Home location.
Oracle Home Name	(Optional) If you previously selected the option to create a new Oracle Home directory location, you can specify a name prefix for the new Oracle Home location. If specified, the value becomes the first part of the full Oracle Home name, which also includes a string identifying the Oracle Database release and latest applied bundle patch, along with numeric identifiers that are used to uniquely identify the Oracle Home location. If you do not specify a value, then the new Oracle Home location is given a system-generated name.

Element	Description
SSH Public Key **Edit**	The SSH public key to be used for authentication when using an SSH client to connect to a compute node that is associated with your database deployment. Click **Edit** to specify the public key. You can upload a file containing the public key value, paste in the value of a public key, or create a system-generated key pair. If you paste in the value, make sure the value does not contain line breaks or end with a line break. **Note:** The SSH Public Key field will not be displayed if the selected Exadata Cloud at Customer environment already contains a previously specified SSH key.
Advanced Settings: Application Type	Specifies how the database deployment is configured: • **Transactional (OLTP)** — configures the database for a transactional workload, with a bias towards high volumes of random data access. • **Decision Support or Data Warehouse** — configures the database for a decision support or data warehouse workload, with a bias towards large data scanning operations. **Note:** The Application Type field is only displayed when you create the first database deployment on an Exadata system. Subsequent database deployments are created with a standardized database configuration.
Advanced Settings: Character Set	The database character set for the database. The database character set is used for: • Data stored in SQL `CHAR` data types (`CHAR`, `VARCHAR2`, `CLOB`, and `LONG`) • Identifiers such as table names, column names, and PL/SQL variables • Entering and storing SQL and PL/SQL source code This option is not available if Create Instance from Existing Backup is set to Yes.
Advanced Settings: National Character Set	The national character set for the database. The national character set is used for data stored in SQL `NCHAR` data types (`NCHAR`, `NCLOB`, and `NVARCHAR2`). This option is not available if Create Instance from Existing Backup is set to Yes.

Element	Description
Advanced Settings: Enable Oracle GoldenGate	Configures the database for use as the replication database of an Oracle GoldenGate Cloud Service instance. See Using Oracle GoldenGate Cloud Service with Exadata Cloud at Customer.

What You See in the Backup and Recovery Configuration Section

Element	Description
Backup Destination	Controls the destination and configuration of automatic backups: • **Both Cloud Storage and Exadata Storage** — enables two separate backup sets containing periodic full (RMAN level 0) backups and daily incremental backups. The backup to cloud storage uses an Oracle Storage Cloud container, with a seven day cycle between full backups and an overall retention period of 30 days. The backup to Exadata storage uses space in the RECO disk group, with a seven day cycle between full backups and a seven day retention period. **Note:** This option is only available if you provisioned for database backups on Exadata storage. See Exadata Storage Configuration. • **Cloud Storage Only** — uses an Oracle Storage Cloud container to store periodic full (RMAN level 0) backups and daily incremental backups, with a seven day cycle between full backups and an overall retention period of 30 days. **Note:** To eliminate possible confusion or contention, do not use a single cloud storage container as a backup destination for multiple database deployments. • **ZDLRA Storage Only** — uses the Recovery Appliance to store one full (RMAN level 0) backup and daily incremental (RMAN level 1) backups. The Recovery Appliance creates virtual full backups from each daily incremental and validates those backups to ensure that they are always recoverable. **Note:** This option is only available if you configured your Exadata Cloud at Customer environment to enable database backups on ZDLRA. See Creating an Exadata Cloud at Customer Instance. • **None** — no automatic backups are configured. For more information about backups and backup configurations, see About Backing Up Database Deployments on Exadata Cloud at Customer.

Element	Description
Cloud Storage Container	The name of an existing Oracle Storage Cloud Service container or a new one to be created in the format: `instance-id_domain/container` where `instance` is the name of the Oracle Storage Cloud Service instance, `id_domain` is the id of the identity domain, and `container` is the name of the container. This field is only displayed if cloud storage is included in your Backup Destination choice.
Username	The user name of a user who has read/write access to the specified **Cloud Storage Container**. This field is only displayed if cloud storage is included in your Backup Destination choice.
Password	The password of the user specified in **Username**. This field is only displayed if cloud storage is included in your Backup Destination choice.
Create Cloud Storage Container	Create a new Oracle Storage Cloud Service container as part of the database deployment creation. Specify the container name and the Cloud Storage user name and password in the preceding fields. This field is only displayed if cloud storage is included in your Backup Destination choice.
Zdlra Storage Container	The TNS connect string to connect to the Oracle Zero Data Loss Recovery Appliance (ZDLRA). This field is only displayed if your Backup Destination choice is ZDLRA Storage Only.
Username	The virtual private catalog (VPC) user name for the ZDLRA specified in **Zdlra Storage Container**. This field is only displayed if your Backup Destination choice is ZDLRA Storage Only.
Password	The password of the user specified in **Username**. This field is only displayed if your Backup Destination choice is ZDLRA Storage Only.

What You See in the Initialize Data From Backup Section

Element	Description
Create Instance from Existing Backup	Create a database deployment whose database is derived from a cloud backup created using Oracle Database Backup Cloud Service. The other fields and options in the Initialize Data From Backup section only display if Create Instance from Existing Backup is set to Yes.

Element	Description
On-Premises Backup	Indicates the origin of the source database backup. Select this option if the source database backup is not from another Exadata Cloud at Customer database deployment in the same identify domain. In this case, the following fields and options are displayed except for Source Service Name. Deselect this option if the source database backup is from another Exadata Cloud at Customer database deployment in the same identify domain. In this case, only the Source Service Name field is displayed.
Database ID	The database identifier of the database from which the existing backup was created. You can get this value by using the following SQL query: `SQL> SELECT dbid FROM v$database;`
Decryption Method **Edit**	Specifies the information necessary to decrypt the source database backup. Click **Edit** to specify the necessary information. In the resulting dialog: • For a backup that uses Transparent Database Encryption (TDE), select **Upload Wallet File** then click **Browse** and specify a zip file containing the source database’s TDE wallet directory, and the contents of that directory. **Note:** If the source database is from another Exadata Cloud at Customer database deployment, its TDE wallet directory is `/u02/app/oracle/admin/`*`dbname`*`/tde_wallet` or `/var/opt/oracle/dbaas_acfs/`*`dbname`*`/tde_wallet`. • For a backup that uses password encryption, select **Paste RMAN Key Value** and paste the password (key value) used to encrypt the backup.
Cloud Storage Container	The name of the Oracle Cloud Infrastructure Object Storage Classic container where the existing backup is stored; use this format: *`instance-id_domain/container`* where *`instance`* is the name of the Oracle Cloud Infrastructure Object Storage Classic instance, *`id_domain`* is the id of the identity domain, and *`container`* is the name of the container.
Username	The user name of an Oracle Cloud user who has read access to the container specified in **Cloud Storage Container**.
Password	The password of the user specified in **Username**.
Source Service Name	From the list of possible alternatives, specify the database deployment that is associated with the source database backup that you want to use.

What You See in the Standby Database Section

Element	Description
Standby Database Configuration	Controls where the standby database is placed in relation to the primary database: • **High Availability** — indicates that the standby database is placed on a different Exadata system in the same region (data center) as the primary database, thus providing isolation at the Exadata system infrastructure level. • **Disaster Recovery** — indicates that the standby database is placed in a different region (data center) from the primary database, thus providing isolation at the Exadata system infrastructure level and geographical separation to protect against catastrophic data center failure. See Using Oracle Data Guard in Exadata Cloud at Customer for more information.
Exadata System	Select an available Oracle Exadata Database Machine configuration to host the standby database. The list contains the Oracle Exadata Database Machines that are associated with your active Exadata Cloud at Customer instances.
Cluster	Specifies the VM cluster to host the standby database. This option is only available if your Exadata Cloud at Customer system environment is enabled to configure multiple VM clusters.
Hostnames	Specify one or more compute nodes that you want to host the database instances for the standby database.

Create Instance: Confirmation Page

Create Instance: Confirmation is the final page in the Create Instance wizard you use to create a new database deployment. For more information, see Creating a Database Deployment.

What You See on the Create Instance: Confirmation Page

The Create Instance: Confirmation page presents a summary list of all the choices you made on the preceding pages of the Create Instance wizard. In addition, it provides the controls described in the following table.

Element	Description
<Previous	Click to return to the Create Instance: Instance Details page.
Cancel	Click to cancel the Create Instance wizard without creating a new deployment.
Create>	Click to begin the process of creating an Exadata Cloud at Customer deployment. The Create Instance wizard closes and the Oracle Database Cloud Service console is displayed, showing the new deployment with a status of In progress.

C The dbaascli Utility

You can use the dbaascli utility to perform various life-cycle and administration operations on Oracle Database Exadata Cloud at Customer.

To use the dbaascli utility, you must be connected to an Exadata Cloud at Customer compute node. See Connecting to a Compute Node Through Secure Shell (SSH).

Some dbaascli commands can be run as the oracle user, but many commands require root administrator privileges. Refer to each command for specific requirements.

On Exadata Cloud at Customer, the dbaascli utility supports these commands:

- dbaascli cswlib download – downloads available software images and makes them available in your Exadata Cloud at Customer environment.
- dbaascli cswlib list – displays information about Oracle Database software images that are available to download to your Exadata Cloud at Customer environment.
- dbaascli database bounce – shuts down and then restarts the database.
- dbaascli database changepassword – changes the password of the specified user.
- dbaascli database move – moves a database to another Oracle Home.
- dbaascli database start– starts and opens the database instances.
- dbaascli database status – displays the open mode of the database and additional information about the database deployment.
- dbaascli database stop – shuts down the database.
- dbaascli database update – performs database configuration changes.
- dbaascli database upgrade – upgrades the major Oracle Database software release version.
- dbaascli dbhome create – creates an Oracle Home.
- dbaascli dbhome info – displays information about Oracle Homes.
- dbaascli dbhome purge – deletes an unused Oracle Home.
- dbaascli dbimage activateBP – activates the specified software image, making it the current default software image in your Exadata Cloud at Customer environment.
- dbaascli dbimage list – displays information about Oracle Database software images that are downloaded to your Exadata Cloud at Customer environment.
- dbaascli dbimage purge – removes the specified software image from your Exadata Cloud at Customer environment.
- dbaascli grid upgrade – upgrades the Oracle Grid Infrastructure software release version.
- dbaascli listener bounce – stops and restarts the Oracle Net listener.
- dbaascli listener start – starts the Oracle Net listener.

- dbaascli listener status – displays the status of the Oracle Net listener, including a summary of listener configuration settings, listening protocol addresses, and a summary of services registered with the listener.
- dbaascli listener stop – stops the Oracle Net listener.
- dbaascli orec keep list – lists the available long-term backups.
- dbaascli orec keep tag – restores a specific long-term backup and performs recovery.
- dbaascli orec latest – restores the most recent backup and performs complete recovery.
- dbaascli orec list – lists the available normal backups.
- dbaascli orec pitr – restores a backup and performs recovery to the specified point in time.
- dbaascli orec scn – restores a backup and performs recovery to the specified SCN.
- dbaascli patch db apply – applies an Oracle Database or Oracle Grid Infrastructure patch.
- dbaascli patch db list – check whether any Oracle Database or Oracle Grid Infrastructure patches are available.
- dbaascli patch db prereq – check the prerequisites for an Oracle Database or Oracle Grid Infrastructure patch.
- dbaascli patch db switchback – rolls back an Oracle Database or Oracle Grid Infrastructure patch.
- dbaascli patch tools apply – downloads and applies a cloud tooling update.
- dbaascli patch tools list – displays the currently installed cloud tooling version and checks for cloud tooling updates.
- dbaascli pdb checkdb – lists information about a container database.
- dbaascli pdb checknode – lists status information about pluggable databases that are associated with a specific container database and a specific compute node.
- dbaascli pdb checkpdb – lists status information about a pluggable database.
- dbaascli pdb close – closes a pluggable database.
- dbaascli pdb connect_info – returns network connection information for a pluggable database.
- dbaascli pdb connect_string – displays Oracle Net connect string information for a pluggable database.
- dbaascli pdb create – creates a new pluggable database.
- dbaascli pdb delete – deletes a pluggable database.
- dbaascli pdb info – displays more detailed information about a pluggable database.
- dbaascli pdb local_clone – creates a new pluggable database as a clone of an existing PDB in the same container database.
- dbaascli pdb open – opens a pluggable database.
- dbaascli pdb remote_clone – creates a new pluggable database as a clone of an existing PDB in another container database.

- dbaascli pdb rename – renames a pluggable database.
- dbaascli pdb resize – modifies the size limits for a pluggable database.
- dbaascli registerdb begin – registers a database with the Exadata Cloud at Customer tooling.
- dbaascli registerdb prereqs – checks the prerequisites for registering a database with the Exadata Cloud at Customer tooling.
- dbaascli registerdb registry – generates the cloud registration file (`creg.ini`) that is used for registering a database.
- dbaascli tde rotate masterkey – changes (rotates) the master encryption key.
- dbaascli tde status – displays information about the software keystore, including the type and status.

dbaascli cswlib download

The `cswlib download` subcommand of the `dbaascli` utility downloads available software images and makes them available in your Exadata Cloud at Customer environment.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
# dbaascli cswlib download --version software_version --bp software_bp [--bp_update
( yes | no )] [--cdb ( yes | no )] [--oss_uri download_location]
```

In the preceding command:

- *`software_version`* — specifies an Oracle Database software version. For example, `11204`, `12102`, `12201`, `18000`, `19000`.
- *`software_bp`* — identifies a bundle patch release. For example, `APR2018`, `JAN2019`, `OCT2019`, and so on.
- `--bp_update` — optionally indicates whether the downloaded software image becomes the current default software image. Default is `no`.
- `--cdb` — optionally specifies whether the downloaded software image supports the Oracle multitenant architecture. Default is `yes`. If you specify `--cdb no`, then a separate software image is downloaded that contains binaries to support non-container databases (non-CDB).
- *`download_location`* — optionally specifies the location of the software image library. The location is specified as a uniform resource identifier (URI) to a cloud storage container that contains available software images.

  Normally, the `--oss_uri` option is not required because the location of the software image library is automatically derived from configuration information in the Exadata Cloud at Customer environment. However, if you experience any difficulties with the automatic location, you can lodge a service request (SR) with Oracle Support and they may instruct you to use this option to specify an alternate location.

To see Oracle Database software images that are available to download to your Exadata Cloud at Customer environment, use the dbaascli cswlib list command.

dbaascli cswlib list

The `cswlib list` subcommand of the `dbaascli` utility displays information about Oracle Database software images that are available to download to your Exadata Cloud at Customer environment.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
# dbaascli cswlib list [ --oss_uri download_location ]
```

The command displays a list of available software images, including version and bundle patch information that you can use to download the software image.

The `--oss_uri` option can be used to specify the location of the software image library. Normally, the `--oss_uri` option is not required because the location of the software image library is automatically derived from configuration information in the Exadata Cloud at Customer environment. However, if you experience any difficulties with the automatic location, you can lodge a service request (SR) with Oracle Support and they may instruct you to use this option to specify an alternate location. The *`download_location`* is specified as a uniform resource identifier (URI) to a cloud storage container that contains available software images.

dbaascli database bounce

The `database bounce` subcommand of the `dbaascli` utility can be used to shut down and restart the database.

Execute this command as the `oracle` user.

```
dbaascli database bounce --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to bounce.

When this subcommand is executed the database is shut down in immediate mode. The database instance is then restarted and the database is opened. In Oracle Database 12c or later, all PDBs are opened.

dbaascli database changepassword

The `database changepassword` subcommand of the `dbaascli` utility is used to change the password of a database user.

Execute this command as the `root` user.

```
dbaascli database changepassword --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to affect.

Enter the user name and new password when prompted.

dbaascli database move

The `database move` subcommand of the `dbaascli` utility moves a database to another Oracle Home directory location.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
dbaascli database move --dbname dbname --ohome oracle_home [--precheck] [--
standby]
```

In the preceding command:

- `--dbname` *`dbname`* — specifies the name of the database that you want to move.
- `--ohome` *`oracle_home`* — specifies the path to an existing Oracle Home directory location, which you want the specified database to use.
- `--precheck` — optionally performs a prerequisite check. You can use this option to test the environment prior to performing a move.
- `--standby` — optionally specifies that you are moving a standby database. Specify this option if you are moving a standby database in an Oracle Data Guard configuration.

Before performing a move operation, ensure that all of the database instances associated with the database deployment are up and running.

You can use this command to move a database to an Oracle Home with a different patch level. For example, you can move in either direction between 12.2.0.1.200414 and 12.2.0.1.200714 or between 19.3 and 19.7. You can also use this command to move a standby database in an Oracle Data Guard configuration to an Oracle Home with a different release level. For example, 12.2.*X* to 19.*Y*.

However, for a standalone database (not in an Oracle Data Guard configuration) or an Oracle Data Guard primary database, you cannot use this command to move between major release versions, such as 12.2.*X* to 19.*Y*. For such cases, see dbaascli database upgrade. See also, Upgrading to 19c Oracle Database on Gen 1 Exadata Cloud at Customer.

When performing a move operation to an Oracle Home with a different patch level, if the database is part of an Oracle Data Guard configuration, then ensure that you move the standby database to the new patchset before you move the primary database.

The command automatically patches the data dictionary (also known as SQL patching or data patching) as required. Specifically, data dictionary patching automatically occurs when you move a database to an Oracle Home with a different patch level. The exception is for standby databases, where data dictionary patching flows from the primary.

dbaascli database start

The `database start` subcommand of the `dbaascli` utility can be used to start the database instance and open the database.

Execute this command as the `oracle` user.

```
dbaascli database start --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to start.

When this subcommand is executed the database instance is started and the database is opened. In Oracle Database 12c or later, all PDBs are opened.

dbaascli database status

The `database status` subcommand of the `dbaascli` utility can be used to check the status of the database in your database deployment.

Execute this command as the `oracle` user.

```
dbaascli database status --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to check.

Output from the command includes the open mode of the database, the software release and edition of the database deployment, and release version of other software components.

dbaascli database stop

The `database stop` subcommand of the `dbaascli` utility can be used to shut down the database.

Execute this command as the `oracle` user.

```
dbaascli database stop --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to stop.

When this subcommand is executed the database is shut down in immediate mode. No new connections or new transactions are permitted. Active transactions are rolled back and all connected users are disconnected.

dbaascli database update

The `database update` subcommand of the `dbaascli` utility enables you to perform database configuration changes.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

The following configuration changes are supported:

- To modify the allocation of CPU and memory resources to the clustered database instances, run the following command:

  ```
  dbaascli database update --dbname dbname --cpu num_cpu [--memory
  memory_size] [--precheck]
  ```

  In the above command:

  - *`dbname`* — specifies the name of the database that you want to change.
  - *`num_cpu`* — specifies the number of CPUs used by the database instance on each compute node.

  - *memory_size* — optionally specifies the amount of memory to allocate to the database instance on each compute node. The value can be expressed in megabytes (M) or gigabytes (G).

  This command modifies the allocation of CPU and memory resources to each individual clustered database instance. If you specify the --memory option, then 40% of the specified value is allocated to the Oracle Database PGA, and 60% is allocated to the SGA. If you do not specify this option, then the PGA and SGA are automatically sized according to the number of CPUs that are allocated to each instance.

  Prior to performing an update operation, you can use the --precheck option to run a series of prerequisite checks to ensure that the update can proceed. No changes are made when using the --precheck option.

- To modify the globally unique database name (DB_UNIQUE_NAME), run the following command:

  ```
  dbaascli database update --dbname dbname --db_unique_name dbname_uniquename [--
  precheck]
  ```

  In the above command:

  - *dbname* — specifies the name of the database that you want to change.
  - *uniquename* — specifies the user configurable portion of the new globally unique database name.

  This command modifies the DB_UNIQUE_NAME database parameter and related configuration entries that reference it, including entries in the Oracle Cluster Registry (OCR) and database server parameter file (SPFILE). File locations that reference the globally unique database name are also updated, including the location of the data files and keystore.

  Note that the value for the --db_unique_name option must commence with the *dbname* value followed immediately by an underscore character. The dbaascli database update command will fail with an error if this convention is not observed.

  Prior to performing an update operation, you can use the --precheck option to run a series of prerequisite checks to ensure that the update can proceed. No changes are made when using the --precheck option.

- To modify the placement of clustered database instances, run the following command:

  ```
  dbaascli database update --dbname dbname --nodelist node1[,node2, ...] [--precheck]
  ```

  In the above command:

  - *dbname* — specifies the name of the database that you want to change.
  - *node1*[, *node2*, ...] — specifies the list of compute nodes that you want to host clustered database instances.

  This command modifies the placement of clustered database instances. If you specify a compute node that does not currently host a database instance, then a database instance is started on that node. If you do not include a node that previously ran an instance of the database, then the database instance is removed from that node.

  Prior to performing an update operation, you can use the --precheck option to run a series of prerequisite checks to ensure that the update can proceed. No changes are made when using the --precheck option.

- To reconfigure the online redo log files, run the following command:

```
dbaascli database update --dbname dbname --redosize redo_size [--groups
num_groups] [--precheck]
```

In the above command:

- *dbname* — specifies the name of the database that you want to change.
- *redo_size* — specifies the size of each online redo log file in megabytes. The valid range is between 1000m and 16000m.
- *num_groups* — optionally specifies the number of online redo log groups to create. The default value is 4.

Prior to performing an update operation, you can use the `--precheck` option to run a series of prerequisite checks to ensure that the update can proceed. No changes are made when using the `--precheck` option.

dbaascli database upgrade

The `database upgrade` subcommand of the `dbaascli` utility enables you to upgrade the major Oracle Database software release version for a database.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
dbaascli database upgrade --dbname dbname --targethome oracle_home [--
executePrereqs] [--postUpgrade] [--revert]
```

In the preceding command:

- `--dbname` *dbname* — specifies the name of the database that you want to move.
- `--targethome` *oracle_home* — specifies the path to an existing Oracle Home directory location, which you want the specified database to use.
- `--executePrereqs` — optionally performs a prerequisite check. You can use this option to test the environment prior to performing an upgrade.
- `--postUpgrade` — optionally performs the post upgrade steps. You can use this option to rerun the post upgrade steps after resolving the issue that caused a previous failure in the post upgrade steps.
- `--revert` — optionally performs a version downgrade. You can use this option to revert back to a previous version.

You can use this command to upgrade a database to another major Oracle Database software release version. For example, you can upgrade from 12.2.*X* to 19.*Y*. See Upgrading to 19c Oracle Database on Gen 1 Exadata Cloud at Customer.

dbaascli dbhome create

The `dbhome create` subcommand of the `dbaascli` utility creates a new Oracle Home directory location and software installation in your Exadata Cloud at Customer environment.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
# dbaascli dbhome create --version software_version
```

In the preceding command, *software_version* specifies an Oracle Database software version. For example, 11204, 12102, 12201, or 18000. The latest available bundle patch for the specified software version is automatically used.

To see information about Oracle Database software images that are available in your Exadata Cloud at Customer environment, including software version and bundle patch details, use the dbaascli dbimage list command.

dbaascli dbhome info

The dbhome info subcommand of the dbaascli utility is used to view information about Oracle Home directory locations.

Connect to the compute node as the opc user and execute this command as the root user.

```
dbaascli dbhome info
```

When prompted, specify an Oracle Home name to view information about that Oracle Home or press Enter to view information about all Oracle Homes registered in your Exadata Cloud at Customer environment.

dbaascli dbhome purge

The dbhome purge subcommand of the dbaascli utility is used to delete an unused Oracle Home directory location.

Connect to the compute node as the opc user and execute this command as the root user.

```
dbaascli dbhome purge
```

When first prompted, enter:

- 1 — if you want to specify the Oracle Home name for the location being purged.
- 2 — if you want to specify the Oracle Home directory path for the location being purged.

When next prompted, enter the Oracle Home name or directory path for the location being purged.

If your entries are valid and the Oracle Home is not associated with a database deployment, then the Oracle binaries are removed from the Oracle Home directory location and the associated metadata is removed from the system.

dbaascli dbimage activateBP

The dbimage activateBP subcommand of the dbaascli utility activates the specified software image, making it the default software image for the corresponding software release in your Exadata Cloud at Customer environment.

Connect to the compute node as the opc user and execute this command as the root user.

```
# dbaascli dbimage activateBP --version software_version --bp software_bp [--cdb ( yes
| no )]
```

In the preceding command:

- *software_version* — specifies the Oracle Database software version. For example, 11204, 12102, 12201, 18000, 19000.
- *software_bp* — identifies the bundle patch release. For example, APR2018, JAN2019, OCT2019, and so on.
- --cdb — optionally specifies whether to activate a software image that supports the Oracle multitenant architecture. Default is yes. If you specify --cdb no, then the command acts on the software image that contains binaries to support non-container databases (non-CDB).

The command fails and outputs an error message if the specified software image is not already downloaded to your Exadata Cloud at Customer environment.

dbaascli dbimage list

The dbimage list subcommand of the dbaascli utility displays information about Oracle Database software images that are downloaded to your Exadata Cloud at Customer environment.

Connect to the compute node as the opc user and execute this command as the root user.

```
# dbaascli dbimage list
```

The command displays a list of software images that are downloaded to your Exadata Cloud at Customer environment, including version and bundle patch information.

dbaascli dbimage purge

The dbimage purge subcommand of the dbaascli utility removes the specified software image from your Exadata Cloud at Customer environment.

Connect to the compute node as the opc user and execute this command as the root user.

```
# dbaascli dbimage purge --version software_version --bp software_bp [--cdb
( yes | no )]
```

In the preceding command:

- *software_version* — specifies the Oracle Database software version. For example, 11204, 12102, 12201, 18000, 19000.
- *software_bp* — identifies the bundle patch release. For example, APR2018, JAN2019, OCT2019, and so on.
- --cdb — optionally specifies whether to remove the software image that supports the Oracle multitenant architecture. Default is yes. If you specify --cdb no, then the software image that contains binaries to support non-container databases (non-CDB) is removed.

If the command will remove a software image that is not currently available in the software image library, and therefore cannot be downloaded again, then the command pauses and prompts for confirmation.

You cannot remove the current default software image for any software version. To avoid this restriction, you must make another software image the current default.

dbaascli grid upgrade

The `grid upgrade` subcommand of the `dbaascli` utility enables you to upgrade the Oracle Grid Infrastructure software release.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
dbaascli grid upgrade --version grid_version [--targethome oracle_home] [--
containerUrl container] [--executePrereqs] [--skipPrereqs] [--
ignorePrereqFailure] [--resume] [--revert]
```

In the preceding command:

- `--version` *`grid_version`* — specifies the Oracle Grid Infrastructure version that you want to upgrade to.
- `--targethome` *`oracle_home`* — optionally specifies the path to the existing Oracle Grid Infrastructure directory location.
- `--containerUrl` *`container`* — optionally specifies the Oracle Grid Infrastructure image files to use for the upgrade. The location is specified as a URL to a cloud storage container that contains the software images.

  Normally, the `--containerUrl` option is not required because the location of the software image library is automatically derived from configuration information in the Exadata Cloud at Customer environment. However, if you experience any difficulties with the automatic location, you can lodge a service request (SR) with Oracle Support and they may instruct you to use this option to specify an alternate location.
- `--executePrereqs` — optionally performs a prerequisite check. You can use this option to test the environment prior to performing an upgrade.
- `--skipPrereqs` — optionally skips prerequisite checks.
- `--ignorePrereqFailure` — optionally ignores failure of a prerequisite check. You can use this option to perform the upgrade even if a prerequisite check fails.
- `--resume` — optionally resumes an upgrade. You can use this option to resume an upgrade after resolving the issue that caused a previous upgrade failure.
- `--revert` — optionally performs a version downgrade. Use this option only after an upgrade failure and with the guidance of Oracle Support.

You can use this command to upgrade Oracle Grid Infrastructure to another major software release version. For example, you can upgrade from 12.2.0.1 to 19.0.0.0. See Upgrading to 19c Oracle Grid Infrastructure on Gen 1 Exadata Cloud at Customer.

dbaascli listener bounce

The `listener bounce` subcommand of the `dbaascli` utility is used to stop and restart the listener.

Execute this command as the `oracle` user.

```
dbaascli listener bounce --dbname dbname
```

In the above command, *dbname* specifies the name of the database whose listener you want to bounce.

This command causes the listener to be stopped and then restarted.

dbaascli listener start

The `listener start` subcommand of the `dbaascli` utility is used to start the listener.

Execute this command as the `oracle` user.

```
dbaascli listener start --dbname dbname
```

In the above command, *dbname* specifies the name of the database whose listener you want to start.

dbaascli listener status

The `listener status` subcommand of the `dbaascli` utility is used to obtain information about the status of the listener.

Execute this command as the `oracle` user.

```
dbaascli listener status --dbname dbname
```

In the above command, *dbname* specifies the name of the database whose listener you want to check.

Basic status information about the listener, including a summary of listener configuration settings, listening protocol addresses, and a summary of services registered with the listener is displayed.

dbaascli listener stop

The `listener stop` subcommand of the `dbaascli` utility is used to stop the listener.

Execute this command as the `oracle` user.

```
dbaascli listener stop --dbname dbname
```

In the above command, *dbname* specifies the name of the database whose listener you want to stop.

dbaascli orec keep list

The `orec latest` subcommand of the `dbaascli` utility is used to list the available long-term backups.

You must execute this command as the `root` user.

```
dbaascli orec --args -keep -list --dbname dbname
```

In the above command, *dbname* specifies the name of the database that you want to list.

dbaascli orec keep tag

The `orec keep tag` subcommand of the `dbaascli` utility is used to restore a specific long-term backup and perform recovery.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
dbaascli orec --args -keep -tag backup-tag --dbname dbname
```

In the above command:

- *`backup-tag`* — specifies the backup tag of the long-term backup that you want to use for the recovery operation.
- *`dbname`* — specifies the name of the database that you want to recover.

dbaascli orec latest

The `orec latest` subcommand of the `dbaascli` utility is used to restore the most recent backup and perform complete recovery.

You must execute this command as the `root` user.

```
dbaascli orec --args -latest [-cfgfiles] --dbname dbname
```

In the above command:

- `-cfgfiles` — optionally directs that any system and database configuration files included in the backup are also restored. Without this option, only the database is restored.
- *`dbname`* — specifies the name of the database that you want to recover.

dbaascli orec list

The `orec latest` subcommand of the `dbaascli` utility is used to list the available normal backups.

You must execute this command as the `root` user.

```
dbaascli orec --args -list --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to list.

dbaascli orec pitr

The `orec pitr` subcommand of the `dbaascli` utility is used to restore a specific normal backup and perform recovery.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
dbaascli orec --args -pitr backup-tag [-cfgfiles] --dbname dbname
```

In the above command:

- *`backup-tag`* — specifies the backup tag of the backup that you want to use for the recovery operation.

- `-cfgfiles` — optionally directs that any system and database configuration files included in the backup are also restored. Without this option, only the database is restored.
- *`dbname`* — specifies the name of the database that you want to recover.

dbaascli orec scn

The `orec scn` subcommand of the `dbaascli` utility is used to restore the most recent backup and perform recovery through the specified system change number (SCN).

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
dbaascli orec --args -scn SCN [-cfgfiles] --dbname dbname
```

In the above command:

- *`SCN`* — specifies the system change number (SCN) for the end point of the recovery operation.
- `-cfgfiles` — optionally directs that any system and database configuration files included in the backup are also restored. Without this option, only the database is restored.
- *`dbname`* — specifies the name of the database that you want to recover.

dbaascli patch db apply

The `patch db apply` subcommand of the `dbaascli` utility is used to apply an Oracle Database or Oracle Grid Infrastructure patch.
Connect to a compute node as the `opc` user and run the command as the `root` user.

- To apply a patch to specific instances, use the following command:

  ```
  # dbaascli patch db apply --patchid patchid
  --instance1 hostname:oracle_home
  [--dbnames dbname[,dbname2 ...]] [--run_datasql (0|1)]
  ```

- To apply a patch by specifying only database names, use the following command:

  ```
  # dbaascli patch db apply --patchid patchid
  --dbnames dbname[,dbname2 ...] [--run_datasql (0|1)] [-alldbs]
  ```

In the preceding commands:

- *`patchid`* identifies the patch to be applied.

> **Note:**
>
> For details about how to find the available patch identifiers, see Listing Available Patches.

- `--instance1` specifies a compute node and Oracle Home directory that is subject to the patching operation. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

  If you use this argument to specify a shared Oracle Home directory and you do not specify the `--dbnames` argument, then all of the databases that share the specified Oracle Home are patched.
- `--dbnames` specifies the database names for the databases that are the target of the patching operation.

  If you use this option to patch a database that uses a shared Oracle Home, and you do not specify the `-alldbs` option, then the database is moved to another Oracle Home that contains the patched Oracle Database binaries.
- `-alldbs` patches all of the databases that share the same Oracle Database binaries (Oracle Home) as the databases specified in the `--dbnames` argument.
- `--run_datasql` controls execution of the SQL commands associated with the patch:
  - `--run_datasql 1` executes the SQL commands associated with the patch.
  - `--run_datasql 0` does not execute the SQL commands associated with the patch.

  If the argument is not specified, the patch-related SQL is executed by default; equivalent to `--run_datasql 1`.

  Patch-related SQL should only be executed after all of the compute nodes are patched. Ensure that you specify `--run_datasql 0` if you are patching a node and further nodes remain to be patched.

  This argument can only be specified along with a patching operation on a compute node. If you have patched all of your nodes and you did not specify this argument, you need to manually execute the SQL commands associated with the patch, which typically involves running the `catbundle.sql` script for Oracle Database 11g or the `datapatch` utility for Oracle Database 12c, or later. Refer to the patch documentation for full details.

> **Note:**
>
> To save time and storage space, the patching operation may move a database to another Oracle Home that already contains the patched Oracle Database binaries.

dbaascli patch db list

The `patch db list` subcommand of the `dbaascli` utility is used to check whether any Oracle Database or Oracle Grid Infrastructure patches are available.
Connect to a compute node as the `opc` user and run the command as the `root` user.

```
# dbaascli patch db list --oh hostname:oracle_home
```

In the preceding command, `--oh` specifies a compute node and Oracle Home directory for which you want to list the available patches. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

dbaascli patch db prereq

The `dbpatchm prereq` subcommand of the `dbaascli` utility is used to check the prerequisites for an Oracle Database or Oracle Grid Infrastructure patch.
Connect to a compute node as the `opc` user and run the command as the `root` user.

- To check patch prerequisites on specific instances, use the following command:

```
# dbaascli patch db prereq --patchid patchid
--instance1 hostname:oracle_home [--dbnames dbname[,dbname2 ...]]
```

- To check patch prerequisites by specifying only database names, use the following command:

```
# dbaascli patch db prereq --patchid patchid
--dbnames dbname[,dbname2 ...] [-alldbs]
```

In the preceding commands:

- *`patchid`* identifies the patch to be pre-checked.

> **Note:**
>
> For details about how to find the available patch identifiers, see Listing Available Patches.

- `--instance1` specifies a compute node and Oracle Home directory that is subject to the pre-check operation. In this context, the Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.
- `--dbnames` specifies the database names for one or more databases that are the target of the pre-check operation.
- `-alldbs` specifies that you want to pre-check all of the databases that share the same Oracle Database binaries (Oracle Home) as the specified databases.

dbaascli patch db switchback

The `patch db switchback` subcommand of the `dbaascli` utility is used to roll back an Oracle Database or Oracle Grid Infrastructure patch.
Connect to a compute node as the `opc` user and run the command as the `root` user.

- To roll back a patch on specific instances, use the following command:

```
# dbaascli patch db switchback --patchid patchid
--instance1 hostname:oracle_home
[--dbnames dbname[,dbname2 ...]] [--run_datasql (0|1)]
```

- To roll back a patch by specifying only database names, use the following command:

  ```
  # dbaascli patch db switchback --patchid patchid
  --dbnames dbname[,dbname2 ...] [--run_datasql (0|1)] [-alldbs]
  ```

In the preceding commands:

- `patchid` identifies the patch to be rolled back.
- `--instance1` specifies a compute node and Oracle Home directory that is subject to the rollback operation. In this context, an Oracle Home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

  If you use this argument to specify a shared Oracle Home directory and you do not specify the `--dbnames` argument, then all of the databases that share the specified Oracle Home are rolled back.
- `--dbnames` specifies the database names for the databases that are the target of the rollback operation.
- `-alldbs` specifies that you want to roll back all of the databases that share the same Oracle Database binaries (Oracle Home) as the databases specified in the `--dbnames` argument.
- `--run_datasql` controls execution of the rollback-related SQL commands associated with the patch:
  - `--run_datasql 1` executes the rollback-related SQL commands associated with the patch.
  - `--run_datasql 0` does not execute the rollback-related SQL commands associated with the patch.

  If the argument is not specified, the rollback-related SQL is executed by default; equivalent to `--run_datasql 1`.

> **Note:**
>
> - Rollback-related SQL should only be executed after all of the compute nodes are rolled back. Therefore, ensure that you specify `--run_datasql 0` if you are rolling back a node and further nodes remain to be rolled back.
> - This argument can only be specified along with a rollback operation on a compute node. Therefore, if you have rolled back all of your nodes and you did not specify this argument, you need to manually execute the SQL commands associated with the rollback operation. Refer to the patch documentation for further details.

dbaascli patch tools apply

The `patch tools apply` subcommand of the `dbaascli` utility is used to download and apply a patch containing the cloud tooling update.
Connect to a compute node as the `opc` user and execute the command as the `root` user:

- To update to the latest available cloud tooling, use the following command:

  ```
  # dbaascli patch tools apply --patchid LATEST
  ```

- To update to a specific cloud tooling release, use the following command:

  ```
  # dbaascli patch tools apply --patchid patchid
  ```

  In the preceding command, *patchid* is a cloud tooling patch identifier, as reported in the output of the dbaascli patch tools list command.

The cloud tooling update is applied to all nodes in the VM cluster.

dbaascli patch tools list

The `patch tools list` subcommand of the `dbaascli` utility is used to check whether any cloud tooling updates are available.
Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
dbaascli patch tools list
```

The command output displays:

- The version of the cloud tooling that is installed on the compute node.
- The list of available updates.
- Notification of the cloud tooling version that is installed on the other compute nodes in the VM cluster.

dbaascli pdb checkdb

The `pdb checkdb` subcommand of the `dbaascli` utility lists information about a container database (CDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb checkdb --dbname dbname
```

In the above command *dbname* specifies the name of the CDB for which you want display information. The information returned by this command includes the number of instances and the CPU count that are associated with the CDB.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb checknode

The `pdb checknode` subcommand of the `dbaascli` utility lists status information about pluggable databases (PDBs) that are associated with a specific container database (CDB) and a specific compute node in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb checknode --node nodenum --dbname dbname
```

In the above command:

- *nodenum* — specifies the node number for a compute node in the Exadata Cloud at Customer environment.
- *dbname* — specifies the name of the CDB.

This command displays status information for all PDBs that are associated with the specified compute node and CDB, including the open mode of each PDB. It is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb checkpdb

The `pdb checkpdb` subcommand of the `dbaascli` utility lists status information about a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb checkpdb --pdbname pdbname --dbname dbname
```

In the above command:

- *pdbname* — specifies the name of the PDB.
- *dbname* — specifies the name of the container database that hosts the PDB.

This command displays status information for the specified PDB, including the open mode and restricted status. It is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb close

The `pdb close` subcommand of the `dbaascli` utility closes a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb close --pdbname pdbname --dbname dbname
```

In the above command:

- *pdbname* — specifies the name of the PDB that you want to close.
- *dbname* — specifies the name of the container database that hosts the PDB.

Upon successful completion, the PDB is closed on all of the container database instances.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb connect_info

The `pdb connect_info` subcommand of the `dbaascli` utility returns network connection information for a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb connect_info --pdbname pdbname --dbname dbname
```

In the above command:

- *pdbname* — specifies the name of the PDB for which you want to return connection information.
- *dbname* — specifies the name of the container database that hosts the PDB.

This command outputs a zip file that contains `tnsnames.ora`, `sqlnet.ora` and `ojdbcs` properties for the PDB. It is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb connect_string

The `pdb connect_string` subcommand of the `dbaascli` utility displays Oracle Net connect string information for a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb connect_string --pdbname pdbname --dbname dbname
```

In the above command:

- *pdbname* — specifies the name of the PDB for which you want to display connect string information.
- *dbname* — specifies the name of the container database that hosts the PDB.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb create

The `pdb create` subcommand of the `dbaascli` utility creates a new pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb create --pdbname pdbname --dbname dbname [--maxsize maxsize] [--
maxcpu maxcpu]
```

In the above command:

- *pdbname* — specifies the name of the new PDB that you want to create.
- *dbname* — specifies the name of the container database in which you want create the new PDB.
- *maxsize* — optionally specifies the maximum total size of data files and temporary files for tablespaces belonging to the PDB. Setting this option is effectively the same as setting the `MAXSIZE` PDB storage clause in the `CREATE PLUGGABLE DATABASE` SQL command. You can impose a limit by specifying an integer followed by a size unit (`K`, `M`, `G`, or `T`), or you can specify `UNLIMITED` to explicitly enforce no limit.
- *maxcpu* — optionally specifies the maximum number of CPUs that are available to the PDB. Setting this option is effectively the same as setting the `CPU_COUNT` parameter in the PDB.

During the PDB creation process you are prompted to specify the administration password for the new PDB.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb delete

The `pdb delete` subcommand of the `dbaascli` utility deletes a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb delete --pdbname pdbname --dbname dbname
```

In the above command:

- *pdbname* — specifies the name of the PDB that you want to delete.
- *dbname* — specifies the name of the container database that hosts the PDB.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb info

The `pdb info` subcommand of the `dbaascli` utility displays more detailed information about a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb info --pdbname pdbname --dbname dbname [--detailed]
```

In the above command:

- *pdbname* — specifies the name of the PDB for which you want to display information.
- *dbname* — specifies the name of the container database that hosts the PDB.

The command displays information about the specified PDB, including attributes such as the CPU count and storage usage that is associated with the PDB. You can add the optional `--detailed` argument to display additional detailed information, including the list of compute nodes where the PDB is currently open in read-write mode.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb local_clone

The `pdb local_clone` subcommand of the `dbaascli` utility creates a new pluggable database (PDB) as a clone of an existing PDB in the same container database (CDB).

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb local_clone --pdbname sourcepdbname --target_pdbname targetpdbname --
dbname dbname
```

In the above command:

- *sourcepdbname* — specifies the name of the PDB that you want to clone.
- *targetpdbname* — specifies the name of the new PDB that you want to create.
- *dbname* — specifies the name of the CDB that hosts the PDBs.

The newly cloned PDB inherits administration passwords from the source PDB.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb open

The `pdb open` subcommand of the `dbaascli` utility opens a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb open --pdbname pdbname --dbname dbname
```

In the above command:

- *pdbname* — specifies the name of the PDB that you want to open.
- *dbname* — specifies the name of the container database that hosts the PDB.

Upon successful completion, the PDB is opened on all of the container database instances.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb remote_clone

The `pdb remote_clone` subcommand of the `dbaascli` utility creates a new pluggable database (PDB) as a clone of an existing PDB in another container database (CDB).

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb remote_clone --pdbname sourcepdbname --source_db sourcedbname --
source_db_scan sourcedbscan --dbname dbname
```

In the above command:

- *pdbname* — specifies the name of the source PDB that you want to clone.
- *sourcedbname* — specifies the name (`DB_UNIQUE_NAME`) of the CDB that hosts the source PDB.
- *sourcedbscan* — specifies the Single Client Access Name (SCAN) that is used to connect to the source database.
- *dbname* — specifies the name (`DB_NAME`) of the CDB that hosts the newly cloned PDB.

When promoted, you must supply the SYS user password for the source PDB.

The newly cloned PDB inherits administration passwords from the source PDB. The cloned PDB is named using the following format: *dbname_sourcepdbname*

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb rename

The `pdb rename` subcommand of the `dbaascli` utility renames a pluggable database (PDB) in your Exadata Cloud at Customer environment.

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb rename --pdbname oldname --newname newname --dbname dbname
```

In the above command:

- *`oldname`* — specifies the old name of the PDB that you want to rename.
- *`newname`* — specifies the new name of the PDB that you want to rename.
- *`dbname`* — specifies the name of the container database that hosts the PDB.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli pdb resize

The `pdb resize` subcommand of the `dbaascli` utility modifies the size limits for a pluggable database (PDB).

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb resize --pdbname pdbname --dbname dbname [--maxsize maxsize] [--maxcpu 
maxcpu]
```

In the above command:

- *`pdbname`* — specifies the name of the PDB that you want to modify.
- *`dbname`* — specifies the name of the container database that hosts the PDB.
- *`maxsize`* — optionally specifies the maximum total size of data files and temporary files for tablespaces belonging to the PDB. Setting this option is effectively the same as setting the `MAXSIZE` PDB storage clause in the `ALTER PLUGGABLE DATABASE` SQL command. You can impose a new limit by specifying an integer followed by a size unit (`K`, `M`, `G`, or `T`), or you can specify `UNLIMITED` to remove a previous limit.
- *`maxcpu`* — optionally specifies the maximum number of CPUs that are available to the PDB. Setting this option is effectively the same as modifying the `CPU_COUNT` parameter in the PDB.

When you run the command you must specify at least one of optional attributes, `--maxsize` or `--maxcpu`. You can specify both optional attributes in a single command.

This command is supported only for databases that are not in a Data Guard configuration and use Oracle Database version 12.2.0.1, or later.

dbaascli registerdb begin

The `registerdb begin` subcommand of the `dbaascli` utility registers a database with the Exadata Cloud at Customer tooling.

Connect to a compute node as the `opc` user and run this command as the `root` user.

```
# dbaascli registerdb begin --dbname dbname [ --db_unique_name dbuniquename ]
```

In the preceding command:

- *dbname* — specifies the database name (`DB_NAME`) of the database that you are registering.
- *dbuniquename* — optionally specifies the globally unique name (`DB_UNIQUE_NAME`) of the database. You only need to specify this parameter of the unique name differs from the database name.

dbaascli registerdb prereqs

The `registerdb prereqs` subcommand of the `dbaascli` utility checks the prerequisites for registering a database with the Exadata Cloud at Customer tooling.

Connect to a compute node as the `opc` user and run this command as the `root` user.

```
# dbaascli registerdb prereqs --dbname dbname [ --db_unique_name dbuniquename ]
```

In the preceding command:

- *dbname* — specifies the database name (`DB_NAME`) of the database that you are registering.
- *dbuniquename* — optionally specifies the globally unique name (`DB_UNIQUE_NAME`) of the database. You only need to specify this parameter of the unique name differs from the database name.

dbaascli registerdb registry

The `registerdb registry` subcommand of the `dbaascli` utility generates the cloud registration file (`creg.ini`) that is used for registering a database.

Connect to a compute node as the `opc` user and run this command as the `root` user.

```
# dbaascli registerdb registry --action ( initialize | report ) --db dbname [ --
udb dbuniquename ]
```

In the preceding command:

- `--action` — specifies the action that you want to perform:
  - `initialize` generates the cloud registration file and distributes it to the required cloud registry location (under `/var/opt/oracle/creg`) on all of the cluster nodes.

  - `report` generates the cloud registration file and places it in a log file location (under `/var/opt/oracle/log/registerdb/creg_report`) so that you can review the contents.
- *dbname* — specifies the database name (`DB_NAME`) of the database that you are registering.
- *dbuniquename* — optionally specifies the globally unique name (`DB_UNIQUE_NAME`) of the database. You only need to specify this parameter of the unique name differs from the database name.

dbaascli tde rotate masterkey

The `tde rotate masterkey` subcommand of the `dbaascli` utility is used to change (rotate) the master encryption key.

Execute this command as the `oracle` user.

```
dbaascli tde rotate masterkey --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to affect.

Enter the keystore password when prompted. The keystore password is initially set to the administration password that you specified when you created the database deployment.

dbaascli tde status

The `tde status` subcommand of the `dbaascli` utility is used to view information about the software keystore used in tablespace encryption.

Execute this command as the `oracle` user.

```
dbaascli tde status --dbname dbname
```

In the above command, *`dbname`* specifies the name of the database that you want to check.

Output from the command includes the type of keystore and the status of the keystore.

D
ExaCLI Commands

You can use the ExaCLI utility to execute specific CellCLI commands, which perform monitoring and management functions on the Exadata Storage Servers that are associated with your Oracle Database Exadata Cloud at Customer instance.

You can use ExaCLI on Exadata Cloud at Customer to view information about the following Exadata Storage Server objects by using the `LIST` command:

- `activerequest`
- `alertdefinition`
- `alerthistory`
- `cell`
- `celldisk`
- `database`
- `flashcache`
- `flashcachecontent`
- `flashlog`
- `griddisk`
- `ibport`
- `iormprofile`
- `lun`
- `metriccurrent`
- `metricdefinition`
- `metrichistory`
- `offloadgroup`
- `physicaldisk`
- `pluggabledatabase`

You can use ExaCLI on Exadata Cloud at Customer to act on the following Exadata Storage Server objects by using the `CREATE`, `ALTER`, `DROP`, and `LIST` commands:

- `diagpack`
- `iormplan`
- `quarantine`

You can use ExaCLI on Exadata Cloud at Customer to alter the following Exadata Storage Server cell attributes by using the `ALTER CELL` command:

- `dbPerfDataSuppress`

- `diagHistoryDays`
- `events`
- `metricCollection`
- `metricHistoryDays`
- `offloadGroupEvents`
- `securityCert`
- `securityPrivKey`
- `securityPrivKeyPW`
- `securityPubKey`
- `traceLevel`

E

Cloud Notification Service Events

Oracle Database Exadata Cloud at Customer is integrated with Cloud Notification Service (CNS), which provides notification for events that affect the status of system components.

CNS Events in Exadata Cloud at Customer

Events that affect the status of key system components are detected on Exadata Cloud at Customer. The following table lists the events that are defined for Exadata Cloud at Customer.

Event Name	Description
`Listener Up`	Generated when the Oracle Net listener starts up.
`Listener Down`	Generated when the Oracle Net listener shuts down.
`Scan Listener Up`	Generated when the SCAN listener starts up.
`Scan Listener Down`	Generated when the SCAN listener shuts down.
`Database Up`	Generated when the associated database starts up.
`Database Down`	Generated when the associated database shuts down.
`Grid Space Full`	Generated when used space in the file system containing the Oracle Grid Infrastructure home directory exceeds the critical threshold.
`Recovery Area Full`	Generated when free space in the RECO ASM disk group falls below a warning or critical threshold.
`Data Space Full`	Generated when free space in the DATA ASM disk group falls below a warning or critical threshold.
`Oracle Home Full`	Generated when free space in the file system containing the Oracle Database home directory falls below the critical threshold.
`Backup Started`	Generated when a database backup is started using cloud tooling.
`Backup Complete`	Generated when a database backup completes, which was started using cloud tooling.
`Backup Failed`	Generated when a database backup fails, which was started using cloud tooling.

Event Generation and Scope

Events are generated in real time or by periodic polling. Polled events are collected at regular intervals by a CNS event monitor process, which runs on one of the cluster nodes. Other events are collected in real time when specific actions are performed by the cloud tooling on Exadata Cloud at Customer.

Events are subject to different regeneration policies. After some events are generated, they are not generated again until they are cleared. For example, after a `Database Up` event is generated, another `Database Up` event is not generated until the first event is cleared by a corresponding `Database Down` event. However, some events are automatically regenerated after 24 hours if the corresponding system status remains unchanged.

Events are also subject to different scopes. Some events are associated with the VM cluster, while other events are associated with a specific database.

The table lists the scope, generation method, and regeneration policy for events that are associated with Exadata Cloud at Customer.

Event Name	Description	Scope	Generation Method	Regeneration Policy
`Listener Up`	Generated when the Oracle Net listener starts up.	Cluster	Periodic polling	Not regenerated
`Listener Down`	Generated when the Oracle Net listener shuts down.	Cluster	Periodic polling	Not regenerated
`Scan Listener Up`	Generated when the SCAN listener starts up.	Cluster	Periodic polling	Not regenerated
`Scan Listener Down`	Generated when the SCAN listener shuts down.	Cluster	Periodic polling	Not regenerated
`Database Up`	Generated when the associated database starts up.	Database	Periodic polling	Not regenerated
`Database Down`	Generated when the associated database shuts down.	Database	Periodic polling	Not regenerated
`Grid Space Full`	Generated when used space in the file system containing the Oracle Grid Infrastructure home directory exceeds the critical threshold.	Cluster	Periodic polling	Automatically regenerated after 24 hours
`Recovery Area Full`	Generated when free space in the RECO ASM disk group falls below a warning or critical threshold.	Cluster	Periodic polling	Automatically regenerated after 24 hours
`Data Space Full`	Generated when free space in the DATA ASM disk group falls below a warning or critical threshold.	Cluster	Periodic polling	Automatically regenerated after 24 hours
`Oracle Home Full`	Generated when free space in the file system containing the Oracle Database home directory falls below the critical threshold.	Cluster	Periodic polling	Automatically regenerated after 24 hours
`Backup Started`	Generated when a database backup is started using cloud tooling.	Database	Real-time	Not regenerated
`Backup Complete`	Generated when a database backup completes, which was started using cloud tooling.	Database	Real-time	Not regenerated
`Backup Failed`	Generated when a database backup fails, which was started using cloud tooling.	Database	Real-time	Not regenerated

Event Topics

Every event is posted to a CNS topic, and notifications are generated for all subscriptions to the topic. Each CNS topic is identified by a fully qualified topic name. The CNS topics associated with Exadata Cloud at Customer have fully qualified names that are formatted as follows:

- For database-specific events, the fully qualified CNS topic name has the format: `UUID.event-category.cluster-name.database.dbname`
- For cluster-related events, the fully qualified CNS topic name has the format: `UUID.event-category.cluster-name.database`

In the CNS topic names:

- *`UUID`* — specifies a unique identifier that is associated with the Exadata Cloud at Customer service entitlement.
- *`event-category`* — specifies the event category, which is one of: `critical`, `warning`, or `informational`.
- *`cluster-name`* — specifies the name of the Exadata Cloud at Customer VM cluster.
- *`dbname`* — specifies the database name that is associated with the Exadata Cloud at Customer database deployment. This is the same as the Container Name that is associated with the database deployment in the Oracle Database Cloud Service console.

The table lists the CNS topics that are associated with Exadata Cloud at Customer.

Event Name	Description	CNS Topics
`Listener Up`	Generated when the Oracle Net listener starts up.	`UUID.informational.cluster-name.database`
`Listener Down`	Generated when the Oracle Net listener shuts down.	`UUID.critical.cluster-name.database`
`Scan Listener Up`	Generated when the SCAN listener starts up.	`UUID.informational.cluster-name.database`
`Scan Listener Down`	Generated when the SCAN listener shuts down.	`UUID.critical.cluster-name.database`
`Database Up`	Generated when the associated database starts up.	`UUID.informational.cluster-name.database.dbname`
`Database Down`	Generated when the associated database shuts down.	`UUID.critical.cluster-name.database.dbname`
`Grid Space Full`	Generated when used space in the file system containing the Oracle Grid Infrastructure home directory exceeds the critical threshold.	`UUID.critical.cluster-name.database`
`Recovery Area Full`	Generated when free space in the RECO ASM disk group falls below a warning or critical threshold.	`UUID.warning.cluster-name.database` `UUID.critical.cluster-name.database`

Event Name	Description	CNS Topics
`Data Space Full`	Generated when free space in the DATA ASM disk group falls below a warning or critical threshold.	*UUID*.warning.*cluster-name*.database *UUID*.critical.*cluster-name*.database
`Oracle Home Full`	Generated when free space in the file system containing the Oracle Database home directory falls below the critical threshold.	*UUID*.critical.*cluster-name*.database
`Backup Started`	Generated when a database backup is started using cloud tooling.	*UUID*.informational.*cluster-name*.database.*dbname*
`Backup Complete`	Generated when a database backup completes, which was started using cloud tooling.	*UUID*.informational.*cluster-name*.database.*dbname*
`Backup Failed`	Generated when a database backup fails, which was started using cloud tooling.	*UUID*.critical.*cluster-name*.database.*dbname*

Event Thresholds

Some events are generated when the value for a system attribute crosses a defined threshold. The following table lists the default threshold values.

Event Name	Description	Default Thresholds
`Grid Space Full`	Generated when used space in the file system containing the Oracle Grid Infrastructure home directory exceeds the critical threshold.	Critical Threshold: 98%
`Recovery Area Full`	Generated when free space in the RECO ASM disk group falls below a warning or critical threshold.	Warning Threshold: 10 GB Critical Threshold: 5 GB
`Data Space Full`	Generated when free space in the DATA ASM disk group falls below a warning or critical threshold.	Warning Threshold: 100 GB Critical Threshold: 50 GB
`Oracle Home Full`	Generated when free space in the file system containing the Oracle Database home directory falls below the critical threshold.	Critical Threshold: 20 GB

Event Notification

Notification of events can be accessed through the My Services application and the Oracle Database Cloud Service console. You can also consume events programmatically by subscribing to specific CNS topics that relate to Exadata Cloud at Customer. You can subscribe to the CNS topics by using the CNS Web console or REST API.

Event Configuration

You can control the generation of CNS events on Exadata Cloud at Customer by using the `dbaascli` utility.

By default, event generation is enabled for cluster events and database events. Event generation can be disabled for a specific database or the entire cluster. When events are disabled for the cluster, database-specific events are also disabled. You can control event generation for cluster events and database events by using the following commands:

- To enable event generation on the VM cluster:

  ```
  # dbaascli cns enable
  ```

- To disable event generation on the VM cluster:

  ```
  # dbaascli cns disable
  ```

- To check the status of event generation on the VM cluster:

  ```
  # dbaascli cns status
  ```

- To enable event generation for a database:

  ```
  # dbaascli cns dbenable --dbname dbname
  ```

- To disable event generation for a database:

  ```
  # dbaascli cns dbdisable --dbname dbname
  ```

- To check the status of event generation for a database:

  ```
  # dbaascli cns dbstatus --dbname dbname
  ```

In the above commands, *`dbname`* specifies the name of the database that you want to control.

You can also view and set the default thresholds to threshold-based events by using the following commands:

- To view the current threshold value:

  ```
  # dbaascli cns getprop --propname property
  ```

- To set a threshold value:

  ```
  # dbaascli cns setprop --propname property --value property-value
  ```

In the above commands:

- *`property`* specifies one of the following:
  - `space_reco_warn` defines the threshold that is used to generate the `Recovery Area Full` warning event. The value of this threshold is the amount of free space expressed in gigabytes (GB).
  - `space_reco_crit` defines the threshold that is used to generate the `Recovery Area Full` critical event. The value of this threshold is the amount of free space expressed in gigabytes (GB).

  - `space_data_warn` defines the threshold that is used to generate the `Data Space Full` warning event. The value of this threshold is the amount of free space expressed in gigabytes (GB).
  - `space_data_crit` defines the threshold that is used to generate the `Data Space Full` critical event. The value of this threshold is the amount of free space expressed in gigabytes (GB).
  - `space_ohome_crit` defines the threshold that is used to generate the `Oracle Home Full` critical event. The value of this threshold is the amount of free space expressed in gigabytes (GB).
  - `space_grid_crit` defines the threshold that is used to generate the `Grid Space Full` critical event. The value of this threshold is the percentage of used space.
- *`property-value`* specifies the integer value for the threshold setting.

For example, to view the current warning threshold for the DATA ASM disk group you can run:

```
# dbaascli cns getprop --propname space_data_warn
```

To set the threshold so that a warning is generated when the free space in the DATA ASM disk group falls below 120GB you can run:

```
# dbaascli cns setprop --propname space_data_warn --value 120
```