Oracle® Cloud

Administering Oracle Database Exadata Cloud Service

Release 18.4.6

E60862-47

January 2019

Oracle Cloud Administering Oracle Database Exadata Cloud Service, Release 18.4.6

E60862-47

Contents

Preface

1 Getting Started with Exadata Cloud Service

2 Managing the Exadata Cloud Service Life Cycle

C The dbaascli Utility

Preface

This document describes how to manage and monitor Oracle Database Exadata Cloud Service 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 Service.

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 Service*
- *Known Issues for Oracle Database Exadata Cloud Service*
- *REST API for Oracle Database Exadata Cloud Service*

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 Service

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

Topics

About Oracle Database Exadata Cloud Service

Oracle Database Exadata Cloud Service enables you to leverage the combined power of Exadata and Oracle Cloud. 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 Service instance** is based on an Exadata system configuration that contains a predefined number of compute nodes (database servers) and a predefined number of Exadata Storage Servers, all tied together by a high-speed, low-latency InfiniBand network and intelligent Exadata software.

The following configurations are offered:

- **Quarter Rack**: Containing 2 compute nodes and 3 Exadata Storage Servers.
- **Half Rack**: Containing 4 compute nodes and 6 Exadata Storage Servers.
- **Full Rack**: Containing 8 compute nodes and 12 Exadata Storage Servers.

Each Exadata Cloud Service 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 enables you to scale an Exadata Cloud Service configuration to meet workload demands and only pay for the compute node resources that you need.

The Exadata Cloud Service 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 are all administered by Oracle.

Exadata Cloud Service 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 Service users seamlessly benefit from the intelligent performance and scalability of Exadata.

Subscription to Exadata Cloud Service 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 Service. If you choose to use Oracle Database software licenses that are included with the Exadata Cloud Service 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 Service also comes with cloud-specific software tools that assist with automated backup, patching and upgrade operations.

Within each Exadata Cloud Service 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 set up, and you control the recovery process in the event of a database failure.

About Exadata Cloud Service Instances

- Exadata System Configuration
- Exadata Storage Configuration
- IP Networks

Exadata System Configuration

Oracle Database Exadata Cloud Service is offered in the following system configurations:

- **Quarter Rack**: Containing 2 compute nodes and 3 Exadata Storage Servers.
- **Half Rack**: Containing 4 compute nodes and 6 Exadata Storage Servers.
- **Full Rack**: Containing 8 compute nodes and 12 Exadata Storage Servers.

Exadata Cloud Service configurations were first offered on Oracle Exadata X5 systems. More recent Exadata Cloud Service configurations are based on Oracle Exadata X6 or X7 systems.

Each 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 enables you to scale an Exadata Cloud Service 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.

System Specifications

The following table outlines the technical specifications for each Exadata Cloud Service system configuration based on Oracle Exadata X5 hardware.

Specification	Quarter Rack	Half Rack	Full Rack
Number of Compute Nodes	2	4	8
— Total Maximum Number of Enabled CPU Cores	68	136	272
— Total RAM Capacity	480 GB	960 GB	1920 GB
Number of Exadata Storage Servers	3	6	12
— Total Raw Flash Storage Capacity	19.2 TB	38.4 TB	76.8 TB
— Total Raw Disk Storage Capacity	144 TB	288 TB	576 TB
— Total Usable Storage Capacity	42 TB	84 TB	168 TB

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

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

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

Specification	Quarter Rack	Half Rack	Full Rack
Number of Compute Nodes	2	4	8
— Total Maximum Number of Enabled CPU Cores	92	184	368
— Total RAM Capacity	1440 GB	2880 GB	5760 GB
Number of Exadata Storage Servers	3	6	12
— Total Raw Flash Storage Capacity	76.8 TB	153.6 TB	307.2 TB
— Total Raw Disk Storage Capacity	360 TB	720 TB	1440 TB

Specification	Quarter Rack	Half Rack	Full Rack
— Total Usable Storage Capacity	106.9 TB	213.8 TB	427.6 TB

Exadata Storage Configuration

As part of configuring each Oracle Database Exadata Cloud Service 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 Service 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 Service instances that are based on Oracle Exadata X6 hardware or Oracle Exadata X5 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 Service 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 Service 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 Service 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 Service environment. If you select this option the SPARSE disk group is created, which enables you to use Exadata Cloud Service snapshot functionality. If you do not select this option, the SPARSE disk group is not created and Exadata Cloud Service 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 Service 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 %

IP Networks

Configuring IP networks in association with Exadata Cloud Service enables you to specify the IP addresses that are used to access your Exadata Cloud Service environment.

Using IP networks allows you to create a network architecture that mirrors and extends your corporate network architecture. This provides network administrators with greater flexibility and control over the Exadata Cloud Service environment.

> **Note:**
>
> IP network configuration is only possible on environments where it is required when creating an Exadata Cloud Service instance. See Creating an Exadata Cloud Service Instance.
>
> If you are not prompted to configure IP networks when you create an Exadata Cloud Service instance, then the Exadata system supporting the service instance is not equipped for IP network configuration. In this case, the only way to get the functionality associated with IP networks is to use another Exadata Cloud Service instance that resides on an Exadata system where IP network configuration is enabled.

On Exadata Cloud Service environments where IP network configuration is enabled, you must specify IP network definitions for the following networks:

- **Client Network** — this network is primarily used for client access to the database servers. Applications typically access databases on Exadata Cloud Service through this network using Oracle Net Services in conjunction with Single Client Access Name (SCAN) and Oracle RAC Virtual IP (VIP) interfaces.
- **Backup Network** — this network is similar to the client network and is typically used to access the database servers for various purposes, including backups and bulk data transfers.

> **Note:**
>
> - You must associate the client network and the backup network with separate IP network definitions. You cannot use one IP network definition for both networks.
> - For environments where IP network configuration is enabled there is no separate administration network interface. All network communications must be done using the client network or the backup network.

An IP network definition consists of two mandatory attributes:

- **Name** — identifies the IP network definition.
- **IP Address Prefix** — specifies a range of IP addresses, in CIDR format.

> When an Exadata Cloud Service instance is associated with an IP network definition, the IP addresses for the specified network are allocated from the address range corresponding to the IP Address Prefix.

Optional attributes control how an IP network interacts with other IP networks.

Before you can associate IP network definitions with your Exadata Cloud Service instance, you must first create the required IP network definitions. To create an IP network you must use the network management Web console that is associated with Oracle Cloud Infrastructure Compute Classic. You can navigate directly to the required network management Web console by clicking Create New IP Network on the Instance Details page of the Create New Oracle Database Exadata Cloud Service Instance wizard.

See About IP Networks and Creating an IP Network in *Using Oracle Cloud Infrastructure Compute Classic*.

About Exadata Cloud Service 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 Service, 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 Service, you must have access to an Exadata Cloud Service 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 Service, 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 Grid Infrastructure Software Release

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

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 Service instance. The starter database is the very first database deployment that you create after the creation of your Exadata Cloud Service instance.

For new starter database deployments:

- If you select Oracle Database 18c as the Oracle Database software release version, then Oracle Grid Infrastructure 18c is installed.
- If you select Oracle Database 12c Release 2, or earlier, as the Oracle Database software release version, then Oracle Grid Infrastructure 12c Release 2 is installed.

After Oracle Grid Infrastructure is installed:

- On systems that are configured with Oracle Grid Infrastructure 18c, you can create database deployments that use any Oracle Database release version.
- On systems that are configured with Oracle Grid Infrastructure 12c Release 2, you can only create database deployments that use Oracle Database 12c Release 2, or earlier. Oracle Grid Infrastructure 12c Release 2 cannot support databases using Oracle Database 18c.
- On older systems that are configured with Oracle Grid Infrastructure 12c Release 1, you can only create database deployments that use Oracle Database 12c Release 1, or earlier. Oracle Grid Infrastructure 12c Release 1 cannot support databases using Oracle Database 12c Release 2, or later.

  If you want to deploy Oracle Database 12c Release 2 on a system that is already configured with Oracle Grid Infrastructure 12c Release 1, then you must manually upgrade to Oracle Grid Infrastructure 12c Release 2 and manually create the version 12.2 database deployment. For details see My Oracle Support note 2206224.1.

Oracle Database Software Edition

When creating a database deployment on Oracle Database Exadata Cloud Service, **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 Service, 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 Service 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 Service 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 Service 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.

When creating a database deployment on Exadata Cloud Service, 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 thirty 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 thirty days.
- **None** — no automatic backups are configured.

Data Security

In Oracle Database Exadata Cloud Service 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 Service 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 Service.

When a database deployment is created on Exadata Cloud Service, 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 Service 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 Service

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

- Oracle Cloud

  See *Getting Started with Oracle Cloud*.

- Oracle Cloud Infrastructure Object Storage Classic

  Exadata Cloud Service 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 Service instance:

- Procure an Exadata Cloud Service subscription. Without an active subscription, you cannot create an Exadata Cloud Service instance or database deployment.
- (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 Generating 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 Service Roles and Users

In addition to the roles and privileges described in Oracle Cloud User Roles and Privileges in *Getting Started with Oracle Cloud*, the **DBaaS Database Administrator** role is created for Oracle Database Exadata Cloud Service.

When the Exadata Cloud Service account is first set up, the service administrator is given this role. User accounts with this role must be added before anyone else can access and use Exadata Cloud Service.

The identity domain administrator can create more Exadata Cloud Service administrators by creating user accounts and assigning them the DBaaS Database Administrator role. See Managing User Accounts in *Managing and Monitoring Oracle Cloud*.

The following table summarizes the privileges given to the DBaaS Database Administrator role.

Description of Privilege	More Information
Can create and delete database deployments	Creating a Database Deployment Deleting a Database Deployment
Can scale, patch, and back up or restore database deployments	Scaling an Exadata Cloud Service Instance Patching Exadata Cloud Service Backing Up and Restoring Databases on Exadata Cloud Service
Can monitor and manage service usage in Oracle Cloud	Managing and Monitoring Oracle Cloud Services in *Managing and Monitoring Oracle Cloud*

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

To access the My Services dashboard, sign in to your Cloud Account.

See Signing in to Your Cloud Account in *Getting Started with Oracle Cloud*.

To access the Oracle Database Cloud Service console:

1. Go to the My Services dashboard.
2. Click the action menu (☰) in the Exadata Classic tile 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.

Using the Exadata Cloud Service REST APIs

You can programmatically provision and manage Oracle Database Exadata Cloud Service 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 Service 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 Service*.

Using the Oracle Cloud My Services REST APIs

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

- Create and modify an Exadata Cloud Service instance.
- Scale the number of enabled CPU cores in the Exadata Cloud Service instance by using CPU core bursting.
- Manage the self-service security firewall:
  - Create and delete security groups.
  - Add security rules to a security group, and remove security rules from a security group.
  - Manage the association between security groups and Exadata Cloud Service instances.

For details, see *Oracle Cloud My Services API*, *Managing Exadata Instances*, and *Exadata Use Cases*.

Typical Workflow for Using Exadata Cloud Service

To start using Oracle Database Exadata Cloud Service, 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 Service roles.	Adding Users and Assigning Roles in *Getting Started with Oracle Cloud*, and About Exadata Cloud Service 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.	Generating 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 Service 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 Service
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 Service
Monitor database deployments	Check on the health and performance of individual database deployments.	Monitoring and Managing Oracle Database on Exadata Cloud Service

Task	Description	More Information
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 Service
Back up a database deployment	Back up a database or restore a database from a backup.	Backing Up and Restoring Databases on Exadata Cloud Service

2

Managing the Exadata Cloud Service Life Cycle

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

Topics

Creating an Exadata Cloud Service Instance

When you create an Oracle Database Exadata Cloud Service instance, you provision the Exadata Database Machine that hosts your Exadata Cloud Service database deployments. To create an Exadata Cloud Service 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 Service instance, ensure that you have an active Exadata Cloud Service subscription in place.

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

Procedure

To create an Exadata Cloud Service 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 Service 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 Service instance. Then, click **Next**.

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

      - **Name** — enter a name for your service instance.
      - **Region** — select the region (data center) that will host your Exadata Cloud Service 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 Service 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.
      - **Additional Number of OCPU (Cores)** — enter the number of additional CPU cores that you want to enable. (Optional)

        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 amongst the compute nodes associated with the Exadata Cloud Service 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 additional limits on the number of CPU cores that you can enable.

        This option does not display if it is incompatible with your subscription.
      - **BYOL enabled** — check this option to indicate that the Exadata Cloud Service 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 Service instance.
      - **Availability Domain** — specifies the infrastructure zone to place the Exadata Cloud Service instance. (Optional)

        Use this setting to configure Exadata Cloud Service instances in different infrastructure zones in order 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.

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. On the Create Service Instance page, specify additional configuration details for your Exadata Cloud Service instance. Then, click **Create Service Instance**.

   - **Exadata System Name** — enter a name for your Exadata Database Machine environment. 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 Service 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.

   - **Client Network** — specify the client network configuration settings by selecting an available IP network definition from the list. If there are no entries in the list you must first create the required IP network definition.
   - **Backup Network** — specify the backup network configuration settings by selecting an available network definition from the list. If there are no entries in the list you must first create the required IP network definition.

Note:

- The **Client Network** and **Backup Network** settings are only available on Exadata systems that are enabled for IP network configuration. See IP Networks.
- You must specify separate IP network definitions for **Client Network** and **Backup Network**. You cannot specify the same IP network definition for both networks.
- To create an IP network you must use the network management Web console that is associated with Oracle Cloud Infrastructure Compute Classic. You can navigate directly to the required network management Web console by clicking Create New IP Network. If you navigate to the network management Web console you must restart the Create New Oracle Database Exadata Cloud Service Instance wizard and re-enter the instance details.

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 Service, 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 Service instance

  Before you can create a database deployment, you must have an active Exadata Cloud Service 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 Service.

See Creating an Exadata Cloud Service 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 Generating a Secure Shell (SSH) Public/Private Key Pair.

  When creating a database deployment on Exadata Cloud Service, 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.

- 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 Service 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 Service 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 Service.
- 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.

Procedure

To create a database deployment on Exadata Cloud Service:

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 Service 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.
   - **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 Service instance. You need to obtain a subscription and create an Exadata Cloud Service 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**

  > **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 Service instance. The starter database is the very first database deployment that you create after the creation of your Exadata Cloud Service instance.
  > - 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 Service instance. 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 Service 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 Service 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 option is available only for databases that use Oracle Database 12c, or later.
      - **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 Service 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 field 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 Service instance. 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.
  - **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`).
  - **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 Service.

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 thirty 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 thirty days.
- **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. 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.

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 Service 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 Service 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 Service 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 Service database deployment in the same identity domain. In this case, the following field is displayed:

- **Source Service Name** — specify the database deployment that is associated with the 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 Service 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.

- **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:

- **Enable network access to the deployment**

  By default, strict security restrictions limit network access to database deployments. To open access to applications and management tools, you need to create and enable your own network security rules. See Enabling Network Access to a Compute Node.

- **Update cloud tooling**

  While the base images used to create Exadata Cloud Service 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 Service.

- **Apply database patches**

  While the base images used to create Exadata Cloud Service 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 Service 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 Service.
- 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 Service 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 Service 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 Service 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 Service 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 Service 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:
   a. Log in as the oracle user.
   b. Unzip the OracleCloud_ibackup_Setup.zip file. Files are extracted into the ibackup directory.
   c. 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
      ```

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

      ```
      $ cd ibackup
      ```

   e. 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 Service 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 Service 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 Service 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 Service 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 Service 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 Service 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 Service 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 to replace the database on an Exadata Cloud Service database deployment by using `ibkup` actions with the `dbaasapi` utility.

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 Service 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 Service 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 Service database deployment, its backup was created using Transparent Data Encryption (TDE) because all cloud backups from Exadata Cloud Service use TDE as the backup encryption mode.

1. On an Exadata Cloud Service compute node that is associated with your target database deployment, create a directory to store the source database TDE wallet along with other files that you will 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 on the compute node that is associated with your target database deployment.

   If the source database is from another Exadata Cloud Service 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`.

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 except those identified as optional.

Parameter	Description
`object`	The value `"db"`.

Parameter	Description
`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 where the source database's backup is stored.
	• `oss_user`: The user name of an Oracle Cloud user who has read privileges for the container where the source database's backup is stored.
	• `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 will use 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 will be used.
	Oracle recommends that you provide both a handle and a tag to use a specific backup or provide neither a handle nor a 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 will use 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 will be used. If they do not support the specified RMAN tag, a “datafile not found” error will occur.
	Oracle recommends that you provide both a handle and a tag to use a specific backup or provide neither a handle nor a 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 Service, 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 `ibkup begin` action whose status this action will check 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 this provides additional 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. You should also ensure that all of 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 those 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 Service 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:

      - **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 thirty 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 thirty days.
      - **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. 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.

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 Service 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:

1. 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 Service, 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 Service 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 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 Service.

> **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 Service 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 Service databases that share the same compute node. If you stop all of the compute nodes associated with an Exadata Cloud Service 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 nodes that are associated with the database deployment, and each compute node entry is accompanied by a separate action menu (☰).

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

   The node first has a status of **Maintenance** and then **Stopped** in the Oracle Database Cloud Service console.

> **Caution:**
>
> Do not use the `halt`, `shutdown` or `shutdown -h` commands to shut down a compute node. Doing so will stop the compute node indefinitely and will require 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 nodes that are associated with the database deployment, and each compute node entry is accompanied by a separate action menu (☰).

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

   The node has a status of **Maintenance** in the Oracle Database Cloud Service console until it is fully started.

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 nodes that are associated with the database deployment, and each compute node entry is accompanied by a separate action menu (☰).

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

The compute node has a status of **Maintenance** in the Oracle Database Cloud Service console until it is fully restarted.

Viewing Past Stop, Start and Restart Activity

You can see information about past stop, start and restart activity for an Exadata Cloud Service 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. Click the triangle icon beside the Activity title to expand the activity log.

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

3. Click the triangle icon beside an operation to see details about that operation.

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

Scaling an Exadata Cloud Service Instance

Two kinds of scaling operations are supported for an Oracle Database Exadata Cloud Service instance:

- Scaling within an Exadata system enables you to modify compute node processing power within the confines of your existing Exadata system.
- Scaling across Exadata system configurations enables you to move to a different Exadata system configuration. For example, from a Quarter Rack to a Half Rack.

Scaling Within an Exadata System

If an Exadata Cloud Service instance requires more compute node processing power, you can scale up the number of enabled CPU cores in the corresponding Oracle Exadata Database Machine. For a non-metered service instance, you can temporarily modify the compute node processing power (bursting) or add compute node processing power on a more permanent basis. For a metered service instance, you can simply modify the number of enabled CPU cores.

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

To modify the number of enabled CPU cores within an existing Exadata Cloud Service 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 Exadata Classic tile 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, specify the type of scaling operation that you want to perform (if applicable) and use the slider control to set the number of enabled CPU cores on each compute node. Then, click **Next**.

   a. If available, specify the type of scaling operation that you want to perform by selecting the **Subscription** option or the **Burst** option:

      > **Note:**
      >
      > This option is not available for metered service instances. If this option is not available for your service instance, then any changes are charged according to the normal terms of your subscription.

      - 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.

        > **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.

   b. 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.

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

   If you need to change any of the settings, use the navigation bar or **Back** button at the top of the wizard to step back to the Instance Details page. Click **Cancel** to cancel out of the wizard without updating the service instance.

Modifying the number of enabled CPU cores is an online operation, which does not require a reboot of the affected compute nodes.

If you have explicitly set the `CPU_COUNT` database initialization parameter, that setting is not affected by modifying the number of enabled CPU cores. Consequently, if you have enabled the Oracle Database instance caging feature, the database instance will not use additional CPU cores until you alter the `CPU_COUNT` setting. If `CPU_COUNT` is set to `0` (its default setting), then Oracle Database continuously monitors the number of CPUs reported by the operating system and uses the current count.

Scaling Across Exadata System Configurations

Scaling across Exadata system configurations enables you to move to a different Exadata system configuration, such as moving from a Quarter Rack to a Half Rack for example. This is useful when a database deployment requires:

- Processing power that is beyond the capacity of the current system configuration.
- Storage capacity that is beyond the capacity of the current system configuration.
- A performance boost that can be delivered by increasing the number of available compute nodes.
- A performance boost that can be delivered by increasing the number of available Exadata Storage Servers.

Scaling across Exadata system configurations requires that the data associated with your database deployment is backed up and restored on a different Exadata Database Machine, which requires an amount of planning and coordination between you and Oracle. To commence the process, submit a service request to Oracle.

Creating and Managing Snapshots of a Database Deployment

Oracle Database Exadata Cloud Service 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 Service instance must be configured with the SPARSE disk group. See Creating an Exadata Cloud Service Instance.
> - Exadata Cloud Service supports snapshots only in conjunction 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 Service 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 Service.

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 Service, 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 Service 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 Service Instance

When you delete an Oracle Database Exadata Cloud Service 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 Service instance may fail when existing snapshot clone database deployments are present. To avoid this problem ensure that you delete all snapshot clone database deployments before you attempt to delete an Exadata Cloud Service instance. See Deleting a Database Deployment.

To delete an Exadata Cloud Service 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 Exadata Classic tile 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 Service

By default, strict security restrictions limit network access to database deployments. To open access to applications and management tools, you may need to perform additional configuration tasks, such as enabling access to a network port or creating an SSH tunnel.

Topics

- About Network Access to Exadata Cloud Service
- Generating a Secure Shell (SSH) Public/Private Key Pair
- Creating an SSH Tunnel to a Compute Node Port
- Enabling Network Access to a Compute Node
- Enabling IPSec VPN Access to Exadata Cloud Service
- Enabling Access to Exadata Cloud Service Using FastConnect Classic
- Enabling Access to Your Network From Exadata Cloud Service
- Controlling Network Access to Exadata Cloud Service
- Defining a Custom Host Name or Domain Name for Exadata Cloud Service
- Defining a Custom SCAN Host Name for Exadata Cloud Service
- Using Network Encryption and Integrity

About Network Access to Exadata Cloud Service

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

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 Service environment. If no public key is associated with your Exadata Cloud Service 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 Service 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 Service 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 Generating a Secure Shell (SSH) Public/Private Key Pair.

Port 22 must be open to access the compute nodes associated with your Exadata Cloud Service environment using SSH. The default configuration of port 22 depends on the firewall configuration that is associated with yourExadata Cloud Service environment. If your Exadata Cloud Service environment uses an Oracle-managed firewall, then port 22 is open by default. Otherwise, port 22 is closed by default and you must create your own security rules to manage access via SSH. See Enabling Network Access to a Compute Node.

Other Network Access Options

Additional configuration is required to access network protocols and services on a compute node other than by using SSH on port 22. You may:

- Enable network access to the port

  You can enable access to a specific compute node port from specific hosts. See Enabling Network Access to a Compute Node

- Create an SSH tunnel to the port

  Creating an SSH tunnel enables you to access a specific compute node port by using an SSH connection as the transport mechanism. To create the tunnel, port 22 must be open (unblocked) in your Exadata Cloud Service environment and you must have the SSH private key file that matches a public key associated with your environment. See Creating an SSH Tunnel to a Compute Node Port.

- Configure an IPSec VPN

  Exadata Cloud Service supports virtual private network (VPN) under the IPSec protocol. This enables secure connectivity between a customer network and Oracle Cloud over the Internet. See Enabling IPSec VPN Access to Exadata Cloud Service.

- Use Oracle Cloud Infrastructure FastConnect Classic

  You can use Oracle FastConnect Classic to create a dedicated private high-speed low-latency network connection between your network and Exadata Cloud Service on Oracle Cloud. See Enabling Access to Exadata Cloud Service Using FastConnect Classic.

Accessing Your Network from Exadata Cloud Service

Additional configuration of your corporate network may be required to support connections originating from Exadata Cloud Service. See Enabling Access to Your Network From Exadata Cloud Service.

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

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

Generating 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.

Generating 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 Service, 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 Service 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 Service 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 Service 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.

Enabling Network Access to a Compute Node

Oracle Database Exadata Cloud Service provides mechanisms to control network access to your Exadata environment. How you control network access depends on the configuration of your Exadata Cloud Service instance:

- If the Exadata Cloud Service instance is configured to use IP Networks, then you must use the network management interfaces that are associated with Oracle Cloud Infrastructure Compute Classic to control network access.

  To control network traffic using IP networks, including enabling access to Exadata Cloud Service compute node ports, see Creating a Security Rule for IP Networks in *Using Oracle Cloud Infrastructure Compute Classic*.

- If the Exadata Cloud Service instance is not configured to use IP networks, but the instance-level action menu (☰) in the Service Details page contains the Manage Security Groups option, then it is configured to use self-service firewall functionality that is native to Exadata Cloud Service. See Using the Exadata Cloud Service Self-Service Firewall.

- Otherwise, the instance is configured to use the Oracle-managed firewall. In that case, to enable access to a specific port on the compute nodes associated with your Exadata Cloud Service environment, you must submit a Service Request to Oracle Support. See How to Request Service Configuration for Oracle Database Exadata Cloud Service.

Using the Exadata Cloud Service Self-Service Firewall

You can use the Exadata Cloud Service self-service firewall if you can access the Manage Security Groups option in the action menu (☰) that is associated with your service instance, which is located on the Service Details page that is accessible from the My Services dashboard.

You can use the self-service firewall to configure security rules and associate them with your Exadata Cloud Service instance. The security rules effectively define a white-list of allowed network access points.

The firewall provides a system of rules and groups. By default, the firewall denies network access to the Exadata Cloud Service instance. When you enable a security rule you enable access to the Exadata Cloud Service instance. To enable access you must:

1. Create a security group.
2. Within the security group, create security rules that define specific network access allowances.
3. Associate the security group with your Exadata Cloud Service instance.

You can define numerous security groups, and each security group can contain numerous security rules. You can associate numerous security groups with each Exadata Cloud Service instance, and each security group can be associated with numerous Exadata Cloud Service instances. You can dynamically enable and disable different security rules by modifying the security groups that are associated with each Exadata Cloud Service instance.

Creating Security Groups

To create a security group:

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 Exadata Classic tile 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 **Manage Security Groups**.

   The Security Groups and Security Rules management page is displayed.

4. Click **Create Group**.

   The Create Security Group dialog is displayed.

5. Specify a **Name** and a **Description** for the new security group. Then, click **Create**.

Creating Security Rules

To create a security rule within a security group:

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 Exadata Classic tile 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 **Manage Security Groups**.

   The Security Groups and Security Rules management page is displayed.

4. Select the desired security group from the list of security groups.

   The selected Security Group is highlighted.

5. Click **Create Rule**.

   The Add Security Rule dialog is displayed.

6. Specify the following attributes for the new security rule. Then, click **Add**.

   - **Direction** — select the direction of the network communications that are subject to this rule:

     - **Inbound** — configures the rule to allow network communications to be received from the location specified in the rule.

     - **Outbound** — configures the rule to allow network communications to be sent to the location specified in the rule.

   - **Protocol** — select the protocol of the network traffic that is subject to this rule:

     - **TCP** — configures the rule to allow TCP/IP network communications.

     - **UDP** — configures the rule to allow UDP network communications.

   - **Interface** — select the network interface that is subject to this rule:

     - **Admin** — specifies that the rule applies to network communications over the administration network interface. The administration network is typically used to support administration tasks by using terminal sessions, monitoring agents, and so on.

     - **Client** — specifies that the rule applies to network communications over the client access network interface, which is typically used by Oracle Net Services connections.

     - **Backup** — specifies that the rule applies to network communications over the backup network interface, which is typically used to transport backup information to and from network-based storage that is separate from Exadata Cloud Service.

   - **Port** — determines whether the rule applies to a specific network port or to a range of ports:

    - **With Range** — specifies that the rule applies to the range of port numbers bounded by **Start Port** and **End Port**. If you select this option you must also specify values for **Start Port** and **End Port**.
    - **Without Range** — specifies that the rule applies to the port number specified by **Port Value**. If you select this option you must also specify a value for **Port Value**.
  - **Start Port**, **End Port**, and **Port Value** — specify the network ports that are subject to this rule. You must enter a valid port number within the range `0` — `65535`.
  - **IP Subnet** — specifies the IP addresses that are subject to this rule. You must enter a single IP address, or specify a range of IP addresses using Classless Inter-Domain Routing (CIDR) notation.

  > **Note:**
  >
  > You can repeat steps 5 and 6 to create multiple rules within a security group.

7. To save and apply the newly created security rules, click **Apply** on the Security Groups and Security Rules management page and then click **Apply** in the Apply Rule dialog.

   > **Note:**
   >
   > Click **Cancel** on the Security Groups and Security Rules management page to remove unapplied security rules. Also, unapplied security rules are automatically removed if you navigate away from the page before they are applied.

Associating Security Groups with an Exadata Cloud Service Instance

To associate security groups with an Exadata Cloud Service 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 Exadata Classic tile 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 **Associate Security Groups**.

   The Associate Security Groups dialog is displayed.

4. Use the dialog controls to specify the desired list of security groups in the **Associated Security Groups** list. Then, click **Add**.

   The security groups, and their corresponding security rules, are enabled immediately.

> **Note:**
>
> You can also use the Associate Security Groups dialog to disable access by removing security groups from the **Associated Security Groups** list.

Enabling IPSec VPN Access to Exadata Cloud Service

Oracle Cloud can provide add-on VPN services, which are available for an additional subscription fee. Using these services, you can create a secure virtual private network (VPN) tunnel over the Internet that connects your corporate network to Oracle Cloud services, such as Oracle Database Exadata Cloud Service. Oracle Cloud VPN services use IPsec, which is a suite of protocols designed to authenticate and encrypt all IP traffic between two locations.

To use this facility, you must have a VPN gateway device that uses current IPSec standards to establish a secure tunnel between your network and Oracle Cloud. Specifically, the device must support:

- IPv4 traffic with support for ICMP, TCP and UDP. Multicast traffic is not supported.
- Tunnel mode sessions: Tunnel mode is used to create a virtual private network between your network and Oracle Cloud, rather than between a specific set of hosts. It is used to protect all communications between both networks.
- Pre-shared key authentication: The supported authentication method for enabling IPSec VPN access to Exadata Cloud Service uses pre-shared keys. With pre-shared keys, the same pre-shared key is configured on each IPSec VPN gateway device.
- Dynamic rekeying: IPsec uses a method called dynamic rekeying to control how often a new key is generated during communication. Communication is sent in blocks and each block of data is secured with a different key.

> **Note:**
>
> For information on IPSec standards, see the Internet Engineering Task Force (IETF) Request for Comments (RFC) 6071: *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*.

In order to avoid any IP address conflict with your client network, Oracle Cloud implements a registered public but non-routable network segment dedicated to act as the destination subnet. This ensures a unique routing target for your clients. Additionally, Oracle requires that you mask your internal systems with a public or non-RFC 1918 address range, which makes an IP address conflict practically impossible in the end-to-end network.

The VPN provisioning process is a collaborative effort between Oracle Cloud network engineers and your corporate network administrators. Key steps in the provisioning process include:

1. An order for Oracle Cloud VPN services is placed. This can be a separate order, or it can be in conjunction with an order for Exadata Cloud Service.

2. You are sent an Oracle Cloud Network VPN Form. This is a pre-filled form based on the service type and hosting location of your Oracle Cloud services. The form requests information required to provision the VPN connection.
3. Oracle receives the completed form and checks that all the prerequisites are met.
4. Oracle provisions the VPN service in conjunction with your network engineers during an agreed maintenance window.
5. Oracle runs through a post-configuration checklist with you to ensure that the VPN is working and that the setup is completed.

See How to Request Service Configuration for Oracle Database Exadata Cloud Service.

Enabling Access to Exadata Cloud Service Using FastConnect Classic

You can use Oracle Cloud Infrastructure FastConnect Classic to access Oracle Database Exadata Cloud Service using a reliable, private and direct connection from your corporate network. When you use FastConnect Classic, your network traffic is routed over a direct and deterministic path that is separate from the public Internet. Consequently, FastConnect Classic provides guaranteed bandwidth and delivers consistent performance and latency.

FastConnect Classic is an add-on networking service, which is available for an additional subscription fee. FastConnect Classic is offered in increments of 1 Gbps and 10 Gbps, and you can combine 1 Gbps and 10 Gbps ports to meet your required network bandwidth.

The configuration of FastConnect Classic depends on the network configuration of the Exadata Cloud Service instance:

- If the Exadata Cloud Service instance is configured to use IP networks, FastConnect private peering is used. In this case, your corporate network RFC1918 IP address space is advertised to the Oracle FastConnect router and Oracle advertises the RFC1918 IP address spaces of the IP networks.
- If the Exadata Cloud Service instance is not configured to use IP networks, FastConnect public peering is used. In this case, your corporate network public IP addresses are advertised to the Oracle FastConnect router. If your corporate network used private IP addresses internally, then you must use a Network Address Translation (NAT) to define the required public addresses. Oracle advertises all of the public IP addresses assigned to your Exadata Cloud Service instance.

The FastConnect provisioning process is a collaborative effort between Oracle Cloud network engineers and your corporate network administrators, which starts when you make a service request to configure FastConnect Classic in conjunction with Exadata Cloud Service.

See How to Request Service Configuration for Oracle Database Exadata Cloud Service.

Enabling Access to Your Network From Exadata Cloud Service

At times you may need to provide access from Exadata Cloud Service to your corporate network. For example, a data loading or migration function running on your Exadata Cloud Service compute nodes may need to pull data from an application or service residing in your corporate network.

If you are not using an IPSec VPN or FastConnect Classic, then you can enable access to your corporate network from Exadata Cloud Service by exposing your corporate network services through a publicly routable IP address. This can be done in your corporate network environment by using a reverse proxy or specific Network Address Translation (NAT) configuration. If you pursue such a configuration:

- Ensure that your corporate network allows publishing of network resources through a publicly routable IP address.
- Ensure that the network traffic is appropriately protected at the application level by using strong encryption and integrity algorithms.
- While considering the security requirements and constraints of your corporate systems and network, ensure that you mitigate any risks associated with publishing the network resource through a publicly routable IP address.

If you are using an IPSec VPN or FastConnect Classic, then you can enable access to your corporate network from Exadata Cloud Service by appropriately configuring the IPSec VPN or FastConnect Classic connection. In this case, ensure that specify your requirements during the network provisioning process.

Controlling Network Access to Exadata Cloud Service

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

- You can define a white-list of clients that are allowed access through the firewall surrounding your Exadata Cloud Service environment. See Enabling Network Access to a Compute Node. After a white-list is defined, the firewall rejects all network traffic that does not conform to the white-list. All network protocols are affected using this mechanism.
- 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)
```

> **Note:**
>
> Regardless of whether you enable Oracle Net Services valid node checking, to enable any Oracle Net Services connections you must enable access to the Oracle Net Listener port (typically port 1521) on your Exadata Cloud Service compute nodes. See Connecting Remotely to the Database by Using Oracle Net Services.

Defining a Custom Host Name or Domain Name for Exadata Cloud Service

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 Service 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 Service 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 Service

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 Service 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 Service. 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 Service 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 Service 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 Service 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 Service 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 Service , 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 Service.

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 Service 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 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 Service

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

Topics

- Connecting to a Compute Node Through Secure Shell (SSH)
- Accessing Enterprise Manager Database Express 18c
- 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)

To gain local access the tools, utilities and other resources on a compute node associated with Oracle Database Exadata Cloud Service, you use Secure Shell (SSH) client software to establish a secure connection and log in as the user `oracle` or the user `opc`.

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 Service.

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 Service 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 Service 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

Enterprise Manager Database Express (EM Express), a web-based tool for managing Oracle Database 18c, is available on Oracle Database Exadata Cloud Service database deployments created using Oracle Database 18c.

You can access EM Express in the following ways:

- Using the Open EM Console menu item
- Using a direct URL
- Using an SSH tunnel

> **Note:**
>
> On Exadata Cloud Service, the EM Express network port is blocked by default. To access EM Express by using the Open EM Console menu item or by using a direct URL, you must first gain access to the network port. See Finding the EM Express Network Port and Enabling Network Access to a Compute Node.

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 Service 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 Service 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 Service 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 Service 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 Service 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
> ```

After you determine the EM Express port for the CDB or PDB that you want to manage, you must choose one of the following options to access EM Express:

- **Unblock the port.** You can unblock the port by Enabling Network Access to a Compute Node.

  After unblocking the port, you can access EM Express on that port as described in Accessing EM Express Using the EM Express Port.
- **Leave the port blocked.** If your security requirements demand that you leave the port blocked, you can still access EM Express by connecting to it through an SSH tunnel, as described in Accessing EM Express Using an SSH Tunnel.

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 Service 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

> **Note:**
>
> On Exadata Cloud Service, the Database Control network port is blocked by default. To access Database Control by using the Open EM Console menu item or by using a direct URL, you must first gain access to the network port. See Finding the Database Control Network Port and Enabling Network Access to a Compute Node.

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 Service 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 theDBA 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 Service 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 theDBA 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 theDBA 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 Service 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 Service supports remote database access by using Oracle Net Services.

Because Exadata Cloud Service leverages 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 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 a node shutdown or failure. 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. The selection of which node listener receives each connection is determined by a combination of factors including listener availability, database instance placement and workload distribution.

Before You Can Connect

By default, the Oracle Net Listeners (SCAN listeners and node listeners) use port 1521, and for security reasons network access to the Oracle Net Listener port (1521) is restricted. Therefore, before you can connect remotely to the database by using Oracle Net Services, you must enable access to the Oracle Net Listener port. See Using Network Encryption and Integrity.

> **Note:**
>
> An SSH tunnel cannot be used to connect to an Exadata Cloud Service 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.

After you are able to access the Oracle Net Listener port, you require two additional pieces of information in order to make a remote database connection by using Oracle Net Services:

- 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 Service environment; for example:

  ```
  PDB1.us2.oraclecloud.com
  ```

  You can determine the network domain name associated with your Exadata Cloud Service 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 in order 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 connection is handled by one of the SCAN listeners. See Defining a Custom SCAN Host Name for Exadata Cloud Service.

  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 Service

This section describes tasks for administering your Oracle Database Exadata Cloud Service environment and the Oracle databases contained therein.

Topics

Using Exadata I/O Resource Management

Oracle Database Exadata Cloud Service 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 Service 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 Service instance.

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 Service 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 Service 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 Service 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 Service

You can update the cloud-specific tooling included on an Exadata Cloud Service compute node by downloading and applying an RPM file containing the latest version of the tools.

Note:

It is highly recommended to maintain the same version of cloud tooling across your Exadata Cloud Service environment. Therefore, the following procedure should be repeated for every compute node.

To update the cloud-specific tooling 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. Download and apply the patch containing the latest cloud tooling update:

   ```
   # /var/opt/oracle/exapatch/exadbcpatch -toolsinst -rpmversion=LATEST
   ```

> **Note:**
>
> If the command fails with an error indicating that LATEST is an invalid RPM version, then proceed as follows:
>
> a. List the available cloud tooling updates:
>
> ```
> # /var/opt/oracle/exapatch/exadbcpatch -list_tools
> ```
>
> b. Examine the command response, and determine the patch ID of the latest cloud tooling update.
>
> The patch ID is listed in the `patches` group as the `patchid` value.
>
> c. Download and apply the patch containing the latest cloud tooling update:
>
> ```
> # /var/opt/oracle/exapatch/exadbcpatch -toolsinst -
> rpmversion=patchid
> ```
>
> where *`patchid`* is the patch ID that you located in the previous step.

The `exadbcpatch` utility runs as a foreground process and does not return control to the user until it completes. Alternatively, you can use `exadbcpatchsm`, which executes as a background process. Both utilities accept the same arguments and perform the same operations. However, when you use `exadbcpatchsm` the utility outputs a transaction ID and immediately returns control to the user. Command output is written to a log file. You can monitor the progress of operations by executing:

```
# /var/opt/oracle/exapatch/exadbcpatchsm -get_status transactionid
```

4. Exit the root-user command shell:

```
# exit
$
```

After you update the cloud tooling across your environment, you should also reconfigure the automatic database backups to use the updated cloud tooling. Use the following procedure on every 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. Re-configure automatic backups to use the updated cloud tooling.

   Execute the following command for every database in your Exadata Cloud Service environment that has an automatic backup configuration that is driven from the current compute node.

```
# /var/opt/oracle/ocde/assistants/bkup/bkup -dbname=dbname
```

   where *`dbname`* is the name of the database that you wish to act on.

> **Note:**
>
> You can determine the databases that have automatic backup configurations that are driven from the current compute node by examining the entries in the `/etc/crontab` file.

4. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Administering Oracle Homes

An Oracle Home is a directory location on the compute nodes that contains Oracle Database binaries. Oracle Database Exadata Cloud Service 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
- Deleting an Oracle Home

> **Note:**
>
> Oracle Home directory locations are created when you create a database deployment. See Creating a Database Deployment.

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 Service 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 above command, *dbname* specifies the name of the database that you wish 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 above command:

- *dbname* — specifies the name of the database that you wish to move.
- *oracle_home* — specifies the path to an existing Oracle Home directory location, which you want the specified database to use.

A move is only feasible if the specified database is at the same patch level as the specified Oracle Home. If the database and Oracle Home are not compatible, then the command fails and returns an error.

5. 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 Service 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 Service instance. Over time, the software images in your Exadata Cloud Service 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.

Topics

- Viewing Information About Downloaded Software Images
- Viewing Information About Available Software Images
- Downloading 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 Service 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 Service 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 Service 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
   ```

   The command displays a list of available software images, including version and bundle patch information that you can use to download the software image.

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 Service 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]
   ```

   In the above command:

   - *`software_version`* — optionally specifies an Oracle Database software version. For example, `11204`, `12102`, or `12201`.
   - *`software_bp`* — optionally identifies a bundle patch release. For example, `APR2018`, `JAN2018`, or `OCT2017`.

   Without the use of any optional arguments, the `dbaascli cswlib download` command downloads the latest available software image for all available Oracle Database software versions.

4. Exit the root-user command shell:

   ```
   # exit
   $
   ```

Administering a Data Guard Configuration

Oracle Database Exadata Cloud Service 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 Service 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 Service 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 on Exadata Cloud Service. The APIs provide a simple mechanism for performing PDB lifecycle management operations that can be used interactively or programmatically.

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
Start the Oracle Database service that is associated with a PDB.	dbaascli pdb start_service
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

Maintaining the Manageability of Exadata Cloud Service

The following best practices will ensure that your Oracle Database Exadata Cloud Service instances stay manageable.

To keep your Exadata Cloud Service 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 users or manually manipulate SSH key settings associated with your Exadata Cloud Service instance.
- Apply **only** patches that are available through Exadata Cloud Service. Do **not** apply patches from any other source unless directed to by Oracle Support.
- 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 Service

You load data into an Oracle database on Oracle Database Exadata Cloud Service using the same tools you would use for an Oracle database on another system.

The location of the database in an Oracle data center does not place any special restrictions on data loading. However, transmission speeds across the Internet tend to be slower, sometimes much slower, than on internal networks, and you should factor this in when choosing any data loading approach.

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 Service for additional techniques and more specific information about migrating existing Oracle databases to Exadata Cloud Service.

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 Service, 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.

Using Oracle Data Transfer Service to Move Large Data Sets

Regardless of how you load data into your databases, it typically makes sense to use a bulk data transfer mechanism to move data close to your Exadata Cloud Service instance before performing a data loading operation.

For smaller data sets, the data can easily be copied over the Internet. However, when the data set is large this may not be feasible. To accommodate these situations, you can use Oracle Data Transfer Service to physically send large data sets to Oracle Cloud.

When you engage Oracle Data Transfer Service:

1. Oracle sends a storage appliance to your data center.
2. You install and configure the appliance into your environment and copy data to the appliance using NFS.
3. Before shipping the appliance back to Oracle, you remove and retain the encryption key from the appliance to ensure in-transit data security.
4. Oracle picks up the appliance and ships it back to the Oracle Cloud data center.
5. You transmit the appliance encryption key to Oracle using a secure communication channel.
6. Oracle copies your data to an Oracle Storage Cloud Service container and provides you with logs to verify the data transfer.
7. You load the data into an Exadata Cloud Service database.
8. Oracle scrubs the storage appliance to remove all traces of the data.

Oracle can provide additional consulting and advanced support services to assist with:

- Installing and configuring the storage appliance in your data center.
- Copying data to the storage appliance.
- Loading data into an Exadata Cloud Service database.

Tuning Oracle Database Performance on Exadata Cloud Service

You tune the performance of Oracle Database on Oracle Database Exadata Cloud Service using the same tools you would use for an Oracle database running on any system in your data center. Exadata Cloud Service 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 Service 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 Service to satisfy the need. See Scaling an Exadata Cloud Service Instance.

Monitoring and Managing Oracle Database on Exadata Cloud Service

To monitor and manage the Oracle database deployed on Oracle Database Exadata Cloud Service, you can use the standard management tool provided with the version of the database:

- For Oracle Database 18c, use Enterprise Manager Database Express 18c. See Accessing Enterprise Manager Database Express 18c.
- 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 Service .

6

Backing Up and Restoring Databases on Exadata Cloud Service

This section explains how to back up and restore Oracle databases on Oracle Database Exadata Cloud Service.

Topics

About Backing Up Database Deployments on Exadata Cloud Service

By backing up your Oracle Database Exadata Cloud Service database deployments, you can protect against data loss if a failure occurs.

About Automatic Database Backups

Exadata Cloud Service 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 Service, 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 thirty 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 thirty days.
- **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.
- 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.
- 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 Service 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 Service 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 Service, 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 Service 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 ouput.

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.

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 a 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 will be 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`	Time of the automatic daily backup using 24 hour time expressed as `hh:mm`.
`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 in 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 in number of days between 1 and 30. Only applicable when `bkup_oss` is set to `yes`. 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`.

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 set config command are not reflected in the Oracle Database Exadata Cloud Service console.

If your backup configuration includes bkup_cfg_files=yes, then each backup includes system and database configuration files and directories specified in the oscfg.spec file and the dbcfg.spec file. Both files are located under /var/opt/oracle/dbaas_acfs/bkup/*dbname*, where *dbname* is the name of the database that is associated with the backup configuration.

To change which system and database configuration files and directories are backed up, you can edit the oscfg.spec file and the dbcfg.spec file.

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
```

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/12.2.0/dbhome_3/admin/db12c/xdb_wallet
/u02/app/oracle/admin/db12c/xdb_wallet
/u02/app/oracle/admin/db12c/db_wallet
# Note: tde_wallet must be backed up in a different location than DATA
bkup.
/u02/app/oracle/admin/db12c/tde_wallet
/u02/app/oracle/admin/db12c/cat_wallet
#/u01/app/oraInventory
#
# Single files
/var/opt/oracle/dbaas_acfs/db12c/opc/opcdb12c.ora
/u02/app/oracle/product/12.2.0/dbhome_3/dbs/opcdb12c.ora
/u02/app/oracle/product/12.2.0/dbhome_3/dbs/orapwdb12c1
/u02/app/oracle/product/12.2.0/dbhome_3/network/admin/listener.ora
/u02/app/oracle/product/12.2.0/dbhome_3/network/admin/sqlnet.ora
/u02/app/oracle/product/12.2.0/dbhome_3/network/admin/tnsnames.ora
/u02/app/oracle/product/12.2.0/dbhome_3/rdbms/lib/env_rdbms.mk
/u02/app/oracle/product/12.2.0/dbhome_3/rdbms/lib/ins_rdbms.mk
#
# Creg
/var/opt/oracle/creg/db12c1.ini
#
```

Creating an On-Demand Backup

You can create an on-demand backup of an Oracle Database Exadata Cloud Service 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 Service:

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:
   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 name of the database deployment whose backup credentials you want to update.

      The Overview page for the deployment is displayed.

   c. 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 Service, 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 Service.

> **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 Service 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 in 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 in number of days between 1 and 30. Only applicable when bkup_oss is set to yes. 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
   $
   ```

Restoring from the Most Recent Backup

You can restore the most recent backup and perform complete recovery on an Oracle Database Exadata Cloud Service 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 restore and recover process performs these steps:
   - Shuts down the database
   - Prepares for recovery
   - Performs the recovery
   - Restarts 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 Service. 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.

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 Service 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 restore and recover process performs these steps:

   - Shuts down the database
   - Prepares for recovery
   - Performs the recovery
   - Restarts 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 Service. 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.

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 Service 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 restore and recover process performs these steps:

   - Shuts down the database
   - Prepares for recovery
   - Performs the recovery
   - Restarts 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 Service. 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
```

- 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.

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
   ```

Manually Restoring from a Backup

Oracle Database Exadata Cloud Service 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.

7 Patching Exadata Cloud Service

This section explains how to apply a patch to Oracle Database Exadata Cloud Service, and roll back the patch as necessary.

Topics

About Patching Exadata Cloud Service

Routine Patching Facilities in Exadata Cloud Service

Oracle Database Exadata Cloud Service provides facilities to manage routine patching of the Oracle Database and Oracle Grid Infrastructure software. These facilities are provided by the Oracle Database Cloud Service console and also by means of using the `exadbcpatchmulti` utility. You are responsible for managing patches and updates to the Oracle Database and Oracle Grid Infrastructure software on the compute nodes.

This document focuses on using the facilities provided by Exadata Cloud Service 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 Service to perform routine patching of Oracle Database and Oracle Grid Infrastructure software. However, you may need to manually patch the Oracle Database or Oracle Grid Infrastructure software in the following circumstances:

- **OJVM Patching** — Patches for the Oracle Java Virtual Machine (OJVM) component of Oracle Database cannot generally be applied in a rolling fashion, and therefore are not included in the routine patch sets provided through Exadata

Cloud Service. If you need to apply patches to the OJVM component of Oracle Database you must do so manually. See Oracle JavaVM Component Database PSU and RU" (OJVM PSU and OJVM RU) Patches.

- **DST Patching** — Patches for the Daylight Savings Time (DST) definitions in Oracle Database cannot generally be applied in a rolling fashion, and therefore are not included in the routine patch sets provided through Exadata Cloud Service. If you need to apply patches to the DST definitions in Oracle Database you must do so manually. See Updated DST Transitions and New Time Zones in Oracle RDBMS and OJVM Time Zone File Patches.
- **Non-routine Patching** — If you encounter a problem that requires a patch which is not included in any routine patch set, 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.

Patching the Compute Node Operating System

You are responsible for managing patches and updates to the operating system environment on the compute nodes. Updating operating system components on the compute nodes is achieved using standard Exadata tools and techniques. See Updating Oracle Linux Database Servers in the *Oracle Exadata Database Machine Maintenance Guide*.

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.

For further information about the standard update policies and practices that apply to Exadata Cloud Service see Oracle Database Cloud Exadata Service Supported Software Versions and Planning for Updates.

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 Service 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 Service. 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 Service 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 Service.

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=hostname:oracle_home1[,oracle_home2 ...]
    [-instance2=hostname:oracle_home1[,oracle_home2 ...] ...]
    -dbname=dbname [-sshkey=sshkey_file]
    ```

  - By specifying a database name:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async patchid
    -dbname=dbname [-alldbs] [-sshkey=sshkey_file]
    ```

- To apply a patch:
  - On specific instances:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
    -instance1=hostname:oracle_home1[,oracle_home2 ...]
    [-instance2=hostname:oracle_home1[,oracle_home2 ...] ...]
    [-dbname=dbname] [-run_datasql=1] [-sshkey=sshkey_file]
    ```

  - By specifying a database name:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
    -dbname=dbname [-alldbs] [-run_datasql=1] [-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 rollback a previously applied patch or a failed patch:
  - On specific instances:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async patchid
    -instance1=hostname:oracle_home1[,oracle_home2 ...]
    [-instance2=hostname:oracle_home1[,oracle_home2 ...] ...]
    [-dbname=dbname] [-run_datasql=1] [-sshkey=sshkey_file]
    ```

  - By specifying a database name:

    ```
    # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async patchid
    -dbname=dbname [-alldbs] [-run_datasql=1] [-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
`-instanceN = hostname:oracle_home1 [,oracle_home2 ...]`	Specifies a compute node and a list of Oracle Home directories that are 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 at a later time. If you use this argument to specify a shared Oracle Home directory and you do not specify the `-dbname` argument, then the patching action is applied to all of the databases that share the specified Oracle Home.
`-dbname = dbname`	Specifies the database name for the database that is the target of the specified patching action. If you use this argument to patch a database that uses 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 database is moved to the new Oracle Home.
`-alldbs`	Specifies that you wish to apply the specified patching operation to all of the databases that share the same Oracle Database binaries (Oracle Home) as the database specified in the `-dbname` argument.
`-run_datasql = 1`	Use this argument to execute the SQL commands associated with a patch or rollback operation. This operation should only be performed after all of the compute nodes are patched or rolled back. Take care not to specify this argument 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 in conjunction 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 will 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.
`-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`.

Argument	Description
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

You can view a list of patches that are associated with an Exadata Cloud Service database deployment.

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 two most recent patches that are associated with the database deployment for each patching category; that is, database patches or grid infrastructure patches. If you wish to access older patches, then you must use the `exadbcpatchmulti` utility.

Other Ways to View Available Patches

- For Oracle Database and Oracle Grid Infrastructure patches, you can use the `exadbcpatchmulti` utility. See Listing Available Patches by Using the exadbcpatchmulti 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.

   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 `exadbcpatchmulti` command with the `-list_patches` action:

   ```
   # /var/opt/oracle/exapatch/exadbcpatchmulti -list_patches
    -oh=hostname:oracle_home [-sshkey=sshkey_file]
   ```

   where:

   - `-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` 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:/u01/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.

4. Exit the root-user command shell:

   ```
   # exit
   $
   ```

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 Service 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 Service.

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 two most recent patches that are associated with the database deployment for each patching category; that is, database patches or grid infrastructure patches. If you wish to access older patches, then you must use the `exadbcpatchmulti` utility.

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` utility. See Checking Prerequisites Before Applying a Patch by Using the exadbcpatchmulti 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.

   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 `exadbcpatchmulti` command with the `-precheck_async` action:

   - On specific instances:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async patchid
     -instance1=hostname:oracle_home1[,oracle_home2 ...]
     [-instance2=hostname:oracle_home1[,oracle_home2 ...] ...]
     -dbname=dbname [-sshkey=sshkey_file]
     ```

   - By specifying a database name:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async patchid
     -dbname=dbname [-alldbs] [-sshkey=sshkey_file]
     ```

   where:

   - *`patchid`* identifies the patch to be pre-checked.

     > **Note:**
     >
     > For details about how to find the available patch identifiers, see Listing Available Patches.

   - `-instanceN` specifies a compute node and one or more Oracle Home directories that are 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.

- -dbname specifies the database name for the database that is the target of the pre-check operation.
- -alldbs specifies that you wish to pre-check all of the databases that share the same Oracle Database binaries (Oracle Home) as the specified database.
- -sshkey 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:/u01/app/12.1.0.2/grid,/u01/app/oracle/product/12.1.0.2/
dbhome_1
-sshkey=/home/opc/.ssh/id_rsa
```

4. Exit the root-user command shell:

```
# exit
$
```

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 Service 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 Service.
- If you use the Oracle Database Cloud Service console to apply an Oracle Database patch to a database deployment that uses a shared Oracle Home, then the patch is only applied to the selected database deployment. If you wish to patch all of the databases that share an Oracle Home in one operation, then you must use the exadbcpatchmulti utility. See Applying a Patch by Using the exadbcpatchmulti Command.

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 two most recent patches that are associated with the database deployment for each patching category; that is, database patches or grid infrastructure patches. If you wish to access older patches, then you must use the `exadbcpatchmulti` utility.

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. 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.
   > - When you apply a patch to a database that uses a shared Oracle Home, a new patched Oracle Home is created and the database is configured to use the new Oracle Home.

Other Ways to Apply a Patch

- For Oracle Database and Oracle Grid Infrastructure patches, you can use the `exadbcpatchmulti` utility. See Applying a Patch by Using the exadbcpatchmulti Command.

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 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. Execute the `exadbcpatchmulti` command with the `-apply_async` action:

   - On specific instances:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
     -instance1=hostname:oracle_home1[,oracle_home2 ...]
     [-instance2=hostname:oracle_home1[,oracle_home2 ...] ...]
     [-dbname=dbname] [-run_datasql=1] [-sshkey=sshkey_file]
     ```

   - By specifying a database name:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async patchid
     -dbname=dbname [-alldbs] [-run_datasql=1] [-sshkey=sshkey_file]
     ```

   where:

   - *`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 one or more Oracle Home directories that are 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. After the operation, the Oracle Home directory location remains unchanged; however, the patch level information embedded in the Oracle Home name is adjusted to reflect the patching operation.

   - `-dbname` specifies the database name for the database that is the target of the patching operation.

     If you use this argument to patch a database that uses 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 database is moved to the new Oracle Home.

- `-alldbs` patches all of the databases that share the same Oracle Database binaries (Oracle Home) as the database specified in the `-dbname` argument.

  After the operation, the Oracle Home directory location remains unchanged; however, the patch level information embedded in the Oracle Home name is adjusted to reflect the patching operation.

- `-run_datasql=1` instructs the `exadbcpatchmulti` command to execute patch-related SQL commands.

  > **Note:**
  >
  > – Patch-related SQL should only be executed after all of the compute nodes are patched. Take care not to specify this argument if you are patching a subset of nodes and further nodes remain to be patched.
  >
  > – This argument can only be specified in conjunction with a patching operation on a set of compute nodes. If you have patched all of your nodes and you did not specify this argument, you will 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` 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:/u01/app/oracle/product/12.1.0.2/dbhome_1
-instance2=hostname2:/u01/app/oracle/product/12.1.0.2/dbhome_1
-run_datasql=1 -sshkey=/home/opc/.ssh/id_rsa
```

4. Exit the root-user command shell:

```
# exit
$
```

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

You can roll back a patch or failed patch attempt on a database deployment.

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` utility. See Rolling Back a Patch or Failed Patch by Using the exadbcpatchmulti Command.

Rolling Back a Patch or Failed Patch by Using the exadbcpatchmulti Command

You can roll back a patch or failed patch attempt on a by using the `exadbcpatchmulti` command.

The patch roll back operation:

- 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 roll back-related SQL after rolling back the patch on all the compute nodes in the cluster.

You can perform a patch roll back operation using the `exadbcpatchmulti` command as follows:

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. Execute the `exadbcpatchmulti` command with the `-rollback_async` action:

   - On specific instances:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async patchid
     -instance1=hostname:oracle_home1[,oracle_home2 ...]
     [-instance2=hostname:oracle_home1[,oracle_home2 ...] ...]
     [-dbname=dbname] [-run_datasql=1] [-sshkey=sshkey_file]
     ```

   - By specifying a database name:

     ```
     # /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async patchid
     -dbname=dbname [-alldbs] [-run_datasql=1] [-sshkey=sshkey_file]
     ```

where:

- *patchid* identifies the patch to be rolled back.
- -instance*N* specifies a compute node and one or more Oracle Home directories that are subject to the roll back 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 rolled back.
- -dbname specifies the database name for the database that is the target of the roll back operation.
- -alldbs specifies that you wish to roll back all of the databases that share the same Oracle Database binaries (Oracle Home) as the database specified in the -dbname argument.
- -run_datasql=1 instructs the exadbcpatchmulti command to execute roll back-related SQL commands.

  > **Note:**
  >
  > – Roll back-related SQL should only be executed after all of the compute nodes are rolled back. Therefore, take care not to specify this argument if you are rolling back a subset of nodes and further nodes remain to be rolled back.
  >
  > – This argument can only be specified in conjunction with a roll back 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 will need to manually execute the SQL commands associated with the roll back operation. Refer to the patch documentation for further details.

- -sshkey 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 -rollback_async 34567890
-instance1=hostname1:/u01/app/12.1.0.2/grid
-instance2=hostname2:/u01/app/12.1.0.2/grid
-run_datasql=1 -sshkey=/home/opc/.ssh/id_rsa
```

4. Exit the root-user command shell:

```
# exit
$
```

8

Configuring Database Features, Database Options, and Companion Products

Oracle Database Exadata Cloud Service provides special capabilities for certain Oracle Database features and options and for certain companion products.

Topics

Using Oracle Data Guard in Exadata Cloud Service

When creating an Oracle Database Exadata Cloud Service 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 Service

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 Service 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 Service, 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 Service 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).

How you specifically enable the network communication that is required for Oracle Data Guard depends on the configuration of the primary and the standby service instances:

- If the Exadata Cloud Service instances use IP Networks, then you must specifically configure your IP network definitions to enable an Oracle Data Guard configuration. See Configuring IP Networks for Oracle Data Guard.
- If the service instances are not configured to use IP networks, but the instance-level action menu (☰) in the Service Details page for each service instance contains the Manage Security Groups option, then the service instances are configured to use self-service firewall functionality that is native to Exadata Cloud Service. If the service instances use self-service firewall functionality, then you must create the security rules to enable cross-site communications. See Using the Exadata Cloud Service Self-Service Firewall.
- Otherwise, the service instances use the Oracle-managed firewall, and you must submit a Service Request to Oracle Support to enable the network communication that is required for Oracle Data Guard. See How to Request Service Configuration for Oracle Database Exadata Cloud Service.

Creating an Oracle Data Guard Configuration

To create an Oracle Data Guard configuration in Exadata Cloud Service, 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 Service 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.

- **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 Service 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 Service, 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.

Configuring IP Networks for Oracle Data Guard

You must configure your IP network definitions to enable the network communication that is required for Oracle Data Guard.

With Exadata Cloud Service, the client network is used by default to facilitate network communications between the primary and standby instances in an Oracle Data Guard configuration.

The required configuration of the client IP networks on your primary and standby instances depends on whether they share the same client IP network. The following scenarios are supported:

- Instances Sharing the Same Client Network
- Instances with Different Client Networks

Instances Sharing the Same Client Network

If your Exadata Cloud Service instances are in the same region and use the same client IP network, then you must create two security rules to enable the network communication that is required for Oracle Data Guard:

- Create an ingress rule for the client network of the primary instance to accept network traffic from the standby instance by specifying the following properties:
  - **Name** — specify a name of your choosing to identify the security rule.
  - **Status** — select `Enabled`.
  - **Type** — select `Ingress`.
  - **Access Control List** — select the access control list that is associated with the client IP network on the primary instance.
  - **Source vNICset** — select the virtual NIC set (vNICset) that is associated with the client IP network on the standby instance.
  - **Destination vNICset** — select the virtual NIC set that is associated with the client IP network on your the instance.
- Create an ingress rule for the client network of the standby instance to accept network traffic from the primary instance by specifying the following properties:
  - **Name** — specify a name of your choosing to identify the security rule.
  - **Status** — select `Enabled`.
  - **Type** — select `Ingress`.
  - **Access Control List** — select the access control list that is associated with the client IP network on the standby instance.
  - **Source vNICset** — select the virtual NIC set (vNICset) that is associated with the client IP network on the primary instance.
  - **Destination vNICset** — select the virtual NIC set that is associated with the client IP network on the standby instance.

See Creating a Security Rule for IP Networks.

Instances with Different Client Networks

If your Exadata Cloud Service instances are in the same region but use different client IP networks, then you must perform the following configuration steps to enable the network communication that is required for Oracle Data Guard:

1. Create an IP network exchange.

   See Creating an IP Network Exchange.

2. Modify the primary and standby client IP networks to use the newly created IP network exchange.

   See Updating an IP Network.

3. Create two security rules to enable the network communication that is required for Oracle Data Guard:

   - Create an ingress rule for the client network of the primary instance to accept network traffic from the standby instance by specifying the following properties:

     - **Name** — specify a name of your choosing to identify the security rule.
     - **Status** — select `Enabled`.
     - **Type** — select `Ingress`.
     - **Access Control List** — select the access control list that is associated with the client IP network on the primary instance.
     - **Source vNICset** — select the virtual NIC set (vNICset) that is associated with the client IP network on the standby instance.
     - **Destination vNICset** — select the virtual NIC set that is associated with the client IP network on the primary instance.

   - Create an ingress rule for the client network of the standby instance to accept network traffic from the primary instance by specifying the following properties:

     - **Name** — specify a name of your choosing to identify the security rule.
     - **Status** — select `Enabled`.
     - **Type** — select `Ingress`.
     - **Access Control List** — select the access control list that is associated with the client IP network on the standby instance.
     - **Source vNICset** — select the virtual NIC set (vNICset) that is associated with the client IP network on the primary instance.
     - **Destination vNICset** — select the virtual NIC set that is associated with the client IP network on the standby instance.

   See Creating a Security Rule for IP Networks.

Using Oracle Multitenant in Exadata Cloud Service

When you create an Oracle Database Exadata Cloud Service database deployment that uses Oracle Database 12c or later, an Oracle Multitenant environment is created.

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

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 in to 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 Service

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 Service.
You must create an Exadata Cloud Service 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 Service 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 can specifically enable network access to the Oracle Net Listener port. See Enabling Network Access to a Compute Node. If you specifically enable access to the Oracle Net Listener port, 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 Service 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 Service 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 Service , 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 Service, 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 default encryption of new tablespaces. In Exadata Cloud Service 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 Service 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 Service. 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 Service 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 Service environments, a standard page is 4 KB, while a Huge Page is 2 MB by default. Therefore, an Oracle Database on Exadata Cloud Service 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 Service instance was created. Two configurations exist:

- Default Configuration of Huge Pages — After Exadata Cloud Service release 17.1.5
- Default Configuration of Huge Pages — Prior to Exadata Cloud Service release 17.1.5

Default Configuration of Huge Pages — After Exadata Cloud Service release 17.1.5

For Exadata Cloud Service 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 Service release 17.1.5

For Exadata Cloud Service 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 Service 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` entry in the `/etc/sysctl.conf` file. This setting is made on each compute node in the environment and it is strongly recommended that the setting is consistent across all of the compute nodes. To alter the Huge Page allocation you can execute the following command on each compute node as the root user:

  ```
  # sysctl -w vm.nr_hugepages=value
  ```

  where `value` is the number of Huge Pages that you want to allocate.

On Exadata Cloud Service 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
```

- 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 Service

You can migrate your on-premises Oracle databases to Oracle Database Exadata Cloud Service using various different approaches based on different tools and technologies.

Topics

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 Service) 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 Service 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 Service. 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.

As part of determining your migration approach, you also need to consider how you physically transport your data to Exadata Cloud Service. For smaller data sets you can transfer the data across a network link between your source system and Exadata Cloud Service. However, for larger data sets this is not feasible because it would simply take too long. To accommodate these situations, you can use Oracle Data Transfer Service to physically send large data sets to Oracle Cloud. When you engage Oracle Data Transfer Service, Oracle ships a storage appliance to your data center. After your data is loading on to the storage and shipped back, Oracle transfers the data to an Oracle Storage Cloud Service container that is accessible from your Exadata Cloud Service environment. See Loading Data into the Oracle Database on Exadata Cloud Service.

Finally, Oracle can offer professional services to assist with all aspects of data migration to Exadata Cloud Service. 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 Service 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 Service.

5. The target database version that you are migrating to on Exadata Cloud Service:

   With Exadata Cloud Service, 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.

> **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.

Migration Methods

Many methods exist to migrate Oracle databases to Oracle Database Exadata Cloud Service.

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.

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 Service 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 Service simply by transferring a backup of your source database to Exadata Cloud Service 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 Service), 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 Service.

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 Service:
>
> - If your source database resides on another little-endian platform, you can use RMAN to transport the entire database to Exadata Cloud Service. See Transportable Database.
> - If your source database resides on a big-endian platform, then you can only use RMAN in conjunction 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.

Using RMAN and Data Transfer Service

You can use RMAN in conjunction with Oracle Data Transfer Service to migrate large databases to Oracle Database Exadata Cloud Service.

When you engage Oracle Data Transfer Service, Oracle ships a storage appliance to your data center. After your data is loading on to the storage and shipped back, Oracle transfers the data to an Oracle Storage Cloud Service container that is accessible from your Exadata Cloud Service environment. See Loading Data into the Oracle Database on Exadata Cloud Service.

RMAN on Exadata Cloud Service uses an SBT (System Backup to Tape) module to access data in an Oracle Storage Cloud Service container. However, RMAN backups in your data center will not use the SBT module, so there are some additional steps that are required to make your backups accessible to Exadata Cloud Service when you use Data Transfer Service to transport RMAN backups from your data center to Exadata Cloud Service.

To use RMAN in conjunction with Data Transfer Service:

- Perform the RMAN backup of your source database in line with the following recommendations:
    - Perform `backupset` backups to the NFS mount point of the storage appliance.
    - Use RMAN backup encryption.
    - Do not use the `maxpiecesize` option to facilitate faster cataloging of backup pieces.
    - Document all of the backup names because these are needed during the restore process.
- Perform the following tasks before you restore the backup to Exadata Cloud Service.
    - Use the RMAN `SEND` command to generate metadata so that the backup pieces can be accessed through the SBT module on Exadata Cloud Service. You can specify one or more backup pieces when you run the `SEND` command. For example:

      ```
      RMAN> run {
                     allocate channel t1 device type sbt
      parms='SBT_LIBRARY=libopc.so';
                     send channel t1 '
                          export backuppiece backup_piece_file_name_1;
                          export backuppiece backup_piece_file_name_2;
                          ...
                          export backuppiece backup_piece_file_name_N;
                     ';
                }
      ```

      where *`backup_piece_file_name_N`* specifies the file name for each backup piece. Specify each file name as reported by RMAN when the backup is taken.
    - Use the RMAN `CATALOG` command to ensure that the backup pieces are known to your database on Exadata Cloud Service. You can specify one or more backup pieces when you run the `CATALOG` command. For example:

      ```
      RMAN> catalog device type sbt backuppiece
                 'backup_piece_file_name_1',
                 'backup_piece_file_name_2',
                 ...
                 'backup_piece_file_name_N';
      ```

      where *`backup_piece_file_name_N`* specifies the file name for each backup piece. Specify the file names as reported by RMAN when the backup is taken.

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 Service.

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 Service 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 Service compute node.
4. On the Exadata Cloud Service 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 Service 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 Service 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 Service compute node.
   c. On the Exadata Cloud Service 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 Service compute node.
   c. On the Exadata Cloud Service 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 Service compute node.
   e. On the Exadata Cloud Service 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 Service compute node, use Data Pump Import to load the metadata associated with the tablespace set.
   g. Set the tablespaces on the Exadata Cloud Service 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 facilitate transport to Exadata Cloud Service.

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 Service 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 Service.

- 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 Service. 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 Service, your source database must reside on Linux x86–64 (the same as Exadata Cloud Service) 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 Service 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 Service 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 Service that will eventually incorporate your migrated database.
2. Manually delete the Exadata Cloud Service database that is created in conjunction with the database deployment in step 1.
3. Create the standby database on Exadata Cloud Service 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 Service 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 Service by unplugging the PDB from the source container database (CDB) and plugging it into a CDB on Exadata Cloud Service.

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 Service.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud Service. 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 Service 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 Service compute node.
3. On the Exadata Cloud Service 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 Service.
   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 Service.

You can migrate a non-CDB to Oracle Database Exadata Cloud Service by plugging the non-CDB into a CDB on Exadata Cloud Service. 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 Service.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud Service. 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 Service 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 Service compute node.
3. On the Exadata Cloud Service 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 Service.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud Service. 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 Service 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.

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 Service 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 Service compute node.
3. On the Exadata Cloud Service 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 Service.

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 Service compute node.

   In this example, the dump file is copied across the network by using the SCP utility.

   a. On the target Exadata Cloud Service 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 Service compute node is available on your source host. For more information about SSH keys, see About Network Access to Exadata Cloud Service.

   c. On the source database host, use the SCP utility to transfer the dump file to the target Exadata Cloud Service 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 Service compute node, invoke Data Pump Import and import the data into the database.

   a. On the Exadata Cloud Service 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 Service 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 Service 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 Service (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 Service compute node.
5. On the Exadata Cloud Service 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 Service compute node, use Data Pump Import to load the metadata associated with the tablespace set.
7. Set the tablespaces on the Exadata Cloud Service 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 Service.

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 Service compute node.

   In this example, the files are copied across the network by using the SCP utility.

   a. On the target Exadata Cloud Service 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 Service compute node is available on your source host. For more information about SSH keys, see About Network Access to Exadata Cloud Service.

   c. On the source database host, use the SCP utility to transfer the dump file and tablespace data files to the target Exadata Cloud Service 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 Service 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 Service 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.

10 Frequently Asked Questions for Exadata Cloud Service

This section provides answers to frequently asked questions (FAQs) for Oracle Database Exadata Cloud Service.

- Who is the service right for?
- Does the Exadata Cloud Service 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?

Who is the service right for?

Exadata Cloud Service 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 Service support external Oracle Net Services (SQL*Net) connections?

Yes. Exadata Cloud Service 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 Service 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 Service 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 Service 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 Service 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 Service deployments. However, you may manually customize your databases to configure and enable Oracle Application Express.

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 Service.

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 Service.

Every Exadata Cloud Service 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.
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 Service database deployment.

When a database deployment is created on Exadata Cloud Service, 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 Service, 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 Service, 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 Service deployment includes a primary database and a single physical standby database. The Oracle Data Guard configuration in an Oracle Database Exadata Cloud Service 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 Service

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 Service.

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 Service.

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 Service. 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 Service 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. **Note:** By default, the port required to access the Enterprise Manager console is initially blocked. To use the console, you must first enable network access to the console's port or create an SSH tunnel to the console's port. See Accessing Exadata Cloud Service • **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 Service 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 Service 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.

Element	Description
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 Service 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 Service 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.

Element	Description
Last Update	Shows the most recent time the SSH keys for this deployment were updated.
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 Service 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. **Note:** By default, the port required to access the Enterprise Manager console is initially blocked. To use the console, you must first enable network access to the console’s port or create an SSH tunnel to the console’s port. See Accessing Exadata Cloud Service - **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.

Element	Description
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.
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.

Element	Description
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 Service 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 Service. 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** 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 Service.

Element	Description
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
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.

Element	Description
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.
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 Service 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.

Element	Description
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.
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 Service.

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 thirty 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 thirty days. • **None** — no automatic backups are configured. For more information about backups and backup configurations, see About Backing Up Database Deployments on Exadata Cloud Service.

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.

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.
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 Service 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 Service 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;`

Element	Description
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 Service 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 Service for more information.

Element	Description
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 Service instances.
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 Service 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 a variety of life-cycle and administration operations on Oracle Database Exadata Cloud Service database deployments.

Using the `dbaascli` utility, you can:

- Change the password of a database user
- Start and stop a database
- Start and stop the Oracle Net listener
- View information about Oracle Homes
- Move a database to another Oracle Home
- Delete an unused Oracle Home
- Perform database configuration changes
- Manage Oracle Database software images
- Manage pluggable databases (PDBs)
- Perform database recovery
- Rotate the master encryption key

To use the `dbaascli` utility:

1. Connect to a compute node associated with the Exadata Cloud Service deployment.

   Commands using the `dbhome`, `dbimage`, `cswlib`, or `orec` subcommands must be run with `root` administrator privileges. In this case, first connect as the `opc` user and then start a root-user command shell by executing the `sudo –s` command.

   Otherwise, connect as the `oracle` user.

   For instructions, see Connecting to a Compute Node Through Secure Shell (SSH).

2. Run the `dbaascli` utility using a command of the form:

   ```
   # dbaascli subcommand subcommand-options
   ```

On Exadata Cloud Service, the `dbaascli` utility supports these subcommands:

Subcommand	Subcommand Options
`cswlib`	`download` – downloads available software images and makes them available in your Exadata Cloud Service environment. `list` – displays information about Oracle Database software images that are available to download to your Exadata Cloud Service environment.

Subcommand	Subcommand Options
`database`	`bounce` – shuts down and then restarts the database instance. `changepassword` – changes the password of the specified user. `move` – moves a database to another Oracle Home. `start`– starts the database instance and opens the database. `status` – displays the open mode of the database and additional information about the database deployment. `stop` – shuts down the database instance. `update` – performs database configuration changes.
`dbhome`	`info` – displays information about Oracle Homes. `purge` – deletes an unused Oracle Home.
`dbimage`	`list` – displays information about Oracle Database software images that are downloaded to your Exadata Cloud Service environment.
`listener`	`bounce` – stops and restarts the listener. `start` – starts the listener. `status` – displays the status of the listener, including a summary of listener configuration settings, listening protocol addresses, and a summary of services registered with the listener. `stop` – stops the listener.
`orec`	`latest` – restores the most recent backup and performs complete recovery. `list` – lists the available normal backups. `pitr` – restores a specific normal backup and performs recovery. `scn` – restores the most recent backup and performs recovery through the specified SCN.

Subcommand	Subcommand Options
pdb	checkdb – lists information about a container database. checknode – lists status information about pluggable databases that are associated with a specific container database and a specific compute node. checkpdb – lists status information about a pluggable database. close – closes a pluggable database. connect_info – returns network connection information for a pluggable database. connect_string – displays Oracle Net connect string information for a pluggable database. create – creates a new pluggable database. delete – deletes a pluggable database. info – displays more detailed information about a pluggable database. local_close – creates a new pluggable database as a clone of an existing PDB in the same container database. open – opens a pluggable database. remote_clone – creates a new pluggable database as a clone of an existing PDB in another container database. rename – renames a pluggable database. resize – modifies the size limits for a pluggable database. start_service – starts the Oracle Database service that is associated with a pluggable database.
tde	rotate masterkey – changes (rotates) the master encryption key. 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 Service 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]
```

In the above command:

- *software_version* — optionally specifies an Oracle Database software version. For example, `11204`, `12102`, or `12201`.
- *software_bp* — optionally identifies a bundle patch release. For example, `APR2018`, `JAN2018`, or `OCT2017`.

Without the use of any optional arguments, the `dbaascli cswlib download` command downloads the latest available software image for all available Oracle Database software versions.

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 Service environment.

Connect to the compute node as the `opc` user and execute this command as the `root` user.

```
# dbaascli cswlib list
```

The command displays a list of available software images, including version and bundle patch information that you can use to download the software image.

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 `oracle` 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
```

In the above 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.

Prior to performing a move operation, ensure that all of the database instances associated with the database deployment are up and running.

A move is only feasible if the specified database is at the same patch level as the specified Oracle Home. If the database and Oracle Home are not compatible, then the command fails and returns an error.

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.

Execute the `dbaascli database update` command as the `oracle` user.

The following configuration changes are supported:

- 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 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 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 Service 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 list

The dbimage list subcommand of the dbaascli utility displays information about Oracle Database software images that are downloaded to your Exadata Cloud Service 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 Service environment, including version and bundle patch information.

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 pdb checkdb

The `pdb checkdb` subcommand of the `dbaascli` utility lists information about a container database (CDB) in your Exadata Cloud Service 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.

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 Service 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 Service 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.

dbaascli pdb checkpdb

The `pdb checkpdb` subcommand of the `dbaascli` utility lists status information about a pluggable database (PDB) in your Exadata Cloud Service 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.

dbaascli pdb close

The `pdb close` subcommand of the `dbaascli` utility closes a pluggable database (PDB) in your Exadata Cloud Service 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.

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 Service 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.

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 Service 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.

dbaascli pdb create

The `pdb create` subcommand of the `dbaascli` utility creates a new pluggable database (PDB) in your Exadata Cloud Service 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.

dbaascli pdb delete

The `pdb delete` subcommand of the `dbaascli` utility deletes a pluggable database (PDB) in your Exadata Cloud Service 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.

dbaascli pdb info

The `pdb info` subcommand of the `dbaascli` utility displays more detailed information about a pluggable database (PDB) in your Exadata Cloud Service 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.

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.

dbaascli pdb open

The `pdb open` subcommand of the `dbaascli` utility opens a pluggable database (PDB) in your Exadata Cloud Service 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.

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 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 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*

dbaascli pdb rename

The `pdb rename` subcommand of the `dbaascli` utility renames a pluggable database (PDB) in your Exadata Cloud Service 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.

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 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 `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.

For each command execution, you must specify at least one of optional attributes, `--maxsize` or `--maxcpu`. You can specify both optional attributes in a single command execution.

dbaascli pdb start_service

The `pdb start_service` subcommand of the `dbaascli` utility starts the Oracle Database service that is associated with a pluggable database (PDB).

Connect to a compute node as the `oracle` user and execute this command.

```
$ dbaascli pdb start_service --pdbname pdbname --dbname dbname
```

In the above command:

- *`pdbname`* — specifies the name of the PDB that is associated with the database service that you want to start.
- *`dbname`* — specifies the name of the container database that hosts the PDB.

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 --dbname dbname
```

In the above command, *`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 --dbname dbname
```

In the above command:

- *backup-tag* — specifies the backup tag of the backup that you want to use for the recovery operation.
- *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 --dbname dbname
```

In the above command:

- *SCN* — specifies the system change number (SCN) for the end point of the recovery operation.
- *dbname* — specifies the name of the database that you want to recover.

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 password specified during the database deployment creation process when prompted for the keystore password.

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.