

## **Oracle® Public Cloud Machine**

Using Oracle Database Exadata Cloud Machine

**E79023-02**

April 2017

Oracle Public Cloud Machine Using Oracle Database Exadata Cloud Machine,

E79023-02

Copyright © 2016, 2016, Oracle and/or its affiliates. All rights reserved.

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

---

---

# Contents

<b>Preface</b> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents.....	vii
Conventions.....	vii
<b>1 Getting Started with Exadata Cloud Machine</b>	
About Oracle Database Exadata Cloud Machine.....	1-1
About Exadata Cloud Machine Instances .....	1-2
Subscription Type.....	1-2
Exadata System Configuration.....	1-3
Exadata Storage Configuration .....	1-3
About Exadata Cloud Machine Database Deployments .....	1-4
Service Level .....	1-5
Metering Frequency .....	1-5
Oracle Database Software Release .....	1-5
Oracle Database Software Edition .....	1-6
Oracle Database Type .....	1-6
Automatic Backup Configuration.....	1-6
Accessing the My Services Dashboard and the Oracle Database Cloud Service Console.....	1-7
Typical Workflow for Using Exadata Cloud Machine.....	1-7
<b>2 Managing the Exadata Cloud Machine Life Cycle</b>	
Creating an Exadata Cloud Machine Instance .....	2-1
Creating a Database Deployment.....	2-3
Viewing All Database Deployments.....	2-8
Viewing Detailed Information for a Database Deployment.....	2-9
Viewing Activities for Database Deployments in an Identity Domain .....	2-9
Stopping, Starting and Restarting Compute Nodes .....	2-10
Scaling Exadata Cloud Machine .....	2-12
Deleting a Database Deployment.....	2-14
Deleting an Exadata Cloud Machine Instance.....	2-15

<b>3</b>	<b>Managing Network Access to Exadata Cloud Machine</b>	
	About Network Access to Exadata Cloud Machine .....	3-1
	Generating a Secure Shell (SSH) Public/Private Key Pair.....	3-2
	Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility .....	3-2
	Generating an SSH Key Pair on Windows Using the PuTTYgen Program.....	3-3
	Creating an SSH Tunnel to a Compute Node Port .....	3-4
	Creating an SSH Tunnel Using the ssh Utility on Linux .....	3-4
	Creating an SSH Tunnel Using the PuTTY Program on Windows .....	3-5
	Enabling Access to a Compute Node Port.....	3-7
	Controlling Network Access to Exadata Cloud Machine .....	3-7
	Defining a Custom Host Name or Domain Name for Exadata Cloud Machine .....	3-8
	Defining a Custom SCAN Host Name for Exadata Cloud Machine .....	3-8
	Using Network Encryption and Integrity .....	3-9
<b>4</b>	<b>Administering Exadata Cloud Machine</b>	
	Using Exadata I/O Resource Management .....	4-1
	Adding an SSH Public Key.....	4-2
	Removing an SSH Public Key .....	4-3
	Updating the Cloud Tooling on Exadata Cloud Machine .....	4-4
	Maintaining the Manageability of Exadata Cloud Machine .....	4-6
	Loading Data into the Oracle Database on Exadata Cloud Machine.....	4-6
	Tuning Oracle Database Performance on Exadata Cloud Machine.....	4-7
	Monitoring and Managing Oracle Database on Exadata Cloud Machine .....	4-7
<b>5</b>	<b>Accessing Exadata Cloud Machine</b>	
	Connecting to a Compute Node Through Secure Shell (SSH).....	5-1
	Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms .....	5-1
	Connecting to a Compute Node Using the PuTTY Program on Windows.....	5-2
	Accessing Enterprise Manager Database Express 12c.....	5-3
	Accessing Enterprise Manager 11g Database Control .....	5-4
	Connecting Remotely to the Database by Using Oracle Net Services .....	5-5
<b>6</b>	<b>Backing Up and Restoring Databases on Exadata Cloud Machine</b>	
	About Backing Up Database Deployments on Exadata Cloud Machine .....	6-1
	Creating an On-Demand Backup .....	6-3
	Creating an On-Demand Backup by Using the bkup_api Utility .....	6-4
	Deleting a Backup .....	6-5
	Customizing the Current Backup Configuration .....	6-6
	Disabling and Re-enabling Scheduled Backups.....	6-8
	Restoring from the Most Recent Backup .....	6-10

Restoring from a Specific Backup .....	6-10
Restoring to a Specific Point in Time .....	6-11
Manually Restoring from a Backup .....	6-12
<b>7 Patching Exadata Cloud Machine</b>	
About Patching Exadata Cloud Machine .....	7-1
Listing Available Patches.....	7-4
Listing Available Patches by Using the exadbcpatchmulti Command .....	7-4
Checking Prerequisites Before Applying a Patch .....	7-5
Checking Prerequisites Before Applying a Patch by Using the exadbcpatchmulti Command .....	7-6
Applying a Patch .....	7-7
Applying a Patch by Using the exadbcpatchmulti Command.....	7-8
Listing Applied Patches.....	7-9
Rolling Back a Patch or Failed Patch.....	7-10
Rolling Back a Patch or Failed Patch by Using the exadbcpatchmulti Command .....	7-11
<b>8 Configuring Database Features, Database Options, and Companion Products</b>	
Using Tablespace Encryption in Exadata Cloud Machine .....	8-1
Creating Encrypted Tablespaces .....	8-1
Managing Tablespace Encryption.....	8-2
Creating and Activating a Master Encryption Key for a PDB .....	8-3
Using Oracle GoldenGate Cloud Service with Exadata Cloud Machine .....	8-4
Managing Huge Pages .....	8-5
<b>9 Migrating Oracle Databases to Exadata Cloud Machine</b>	
Choosing a Migration Method.....	9-1
Migration Methods .....	9-6
Conventional RMAN Backup and Recovery .....	9-6
Conventional Data Pump Export and Import.....	9-7
Transportable Tablespaces.....	9-9
Data Pump Full Transportable Export and Import.....	9-13
Transportable Tablespaces with Cross Platform Incremental Backup.....	9-14
Transportable Database.....	9-16
Data Guard Physical Standby.....	9-17
Unplugging and Plugging a Pluggable Database .....	9-19
Plugging in a Non-CDB.....	9-21
Cloning a Remote PDB or Non-CDB.....	9-21
<b>10 Frequently Asked Questions for Exadata Cloud Machine .....</b>	<b>10-1</b>
<b>A Characteristics of a Newly Created Deployment</b>	
Linux User Accounts.....	A-1

Locations of Installed Software.....	A-2
Network Access .....	A-3
Oracle Database Characteristics .....	A-3
Location of Diagnostic and Log Files.....	A-4

**B Oracle Cloud Pages for Administering Exadata Cloud Machine**

Services Page .....	B-1
Activity Page .....	B-3
SSH Access Page .....	B-5
Overview Page .....	B-6
Backup Page .....	B-8
Patching Page .....	B-9
Create Service: Service Page.....	B-9
Create Service: Service Details Page .....	B-11
Create Service: Confirmation Page.....	B-13

---

# Preface

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

## Topics

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

## Audience

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

## 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:

- *Oracle Database Exadata Cloud Machine Deployment Guide*

## Conventions

The following text conventions are used in this document:

---

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

---



---

# Getting Started with Exadata Cloud Machine

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

## Topics

- [About Oracle Database Exadata Cloud Machine](#)
- [About Exadata Cloud Machine Instances](#)
- [About Exadata Cloud Machine Database Deployments](#)
- [Accessing the My Services Dashboard and the Oracle Database Cloud Service Console](#)
- [Typical Workflow for Using Exadata Cloud Machine](#)

## About Oracle Database Exadata Cloud Machine

Oracle Database Exadata Cloud Machine enables you to leverage the power of Exadata and the Oracle Cloud by implementing Exadata Cloud Service inside your own datacenter. 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 Machine 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:

- **Eighth Rack:** Containing 2 compute nodes and 3 Exadata Storage Servers.
- **Quarter Rack:** Containing 2 compute nodes and 3 Exadata Storage Servers. However, compared to an Eighth Rack, the Exadata Storage Servers contain double the storage capacity, while the compute nodes contain much more memory and a significant boost in the maximum number of CPU cores that can be enabled.
- **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 Machine configuration is equipped with a fixed amount of memory, storage and network resources. However, you can choose to enable additional compute node CPU cores beyond a fixed minimum for each configuration. This enables you to scale up an Exadata Cloud Machine configuration to meet

growing workload demands, and only pay for the compute node resources that you need.

The Exadata Cloud Machine compute nodes are each configured with a 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 Machine 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 Machine users seamlessly benefit from the intelligent performance and scalability of Exadata.

Subscription to Exadata Cloud Machine includes 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). Exadata Cloud Machine also comes with cloud-specific capabilities that assist with automated backup, patching and upgrade operations.

Within each Exadata configuration 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 from outside the Oracle Cloud. 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 Machine Instances

When you configure an Oracle Database Exadata Cloud Machine instance, you must make a series of choices that determine various characteristics associated with your service. These choices include:

- [Subscription Type](#)
- [Exadata System Configuration](#)
- [Exadata Storage Configuration](#)

### Subscription Type

Oracle Database Exadata Cloud Machine is only offered using a standard term-based subscription, which is also known as a non-metered subscription.

## Exadata System Configuration

Oracle Database Exadata Cloud Machine is offered in Eighth Rack, Quarter Rack, Half Rack or Full Rack configurations.

Each Exadata Cloud Machine 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.

By default, a set minimum number of cores are enabled on each configuration, and you can choose to enable additional CPU cores up to the total maximum for each configuration. This enables you to scale an Exadata Cloud Machine 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.

The following table outlines the vital statistics for each system configuration.

Statistic	Eighth Rack	Quarter Rack	Half Rack	Full Rack
Number of Compute Nodes	2	2	4	8
— Total Minimum (Default) Number of Enabled CPU Cores	16	22	44	88
— Total Maximum Number of Enabled CPU Cores	68	84	168	336
— Total RAM Capacity	480 GB	1440 GB	2880 GB	5760 GB
Number of Exadata Storage Servers	3	3	6	12
— Total Raw Flash Storage Capacity	19.2 TB	38.4 TB	76.8 TB	153.6 TB
— Total Raw Disk Storage Capacity	144 TB	288 TB	576 TB	1152 TB
— Total Usable Storage Capacity	42 TB	84 TB	168 TB	336 TB

## Exadata Storage Configuration

As part of configuring each Oracle Database Exadata Cloud Machine 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 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.

- The DBFS and ACFS disk groups are system disk groups that support various operational purposes. The DBFS disk group is primarily used to store the shared clusterware files (Oracle Cluster Registry and voting disks), while the ACFS disk groups are primarily used to store Oracle Database binaries. Compared to the DATA and RECO disk groups, the system disk groups are so small that they are typically ignored when discussing the overall storage capacity. You should not store Oracle Database data files or backups inside the system disk groups.

---

**Note:** The 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 `DATA_C2`, the RECO disk group would be named `RECO_C2`, and so on.

---

As an input to the configuration process, you must decide if you intend to perform database backups to the Exadata storage within your Exadata Cloud Machine environment. Your choice profoundly affects how storage space in the Exadata Storage Servers is allocated to the ASM disk groups.

If you choose to provision for backups on Exadata storage, approximately 40% of the available storage space is allocated to the DATA disk group and approximately 60% is allocated to the RECO disk group. If you choose not to provision for backups on Exadata storage, approximately 80% of the available storage space is allocated to the DATA disk group and approximately 20% is allocated to the RECO disk group. After the storage is configured, the only way to adjust the allocation without reconfiguring the whole environment is by lodging a Service Request with Oracle. For details see [My Oracle Support Note 2007530.1](#).

The following table outlines how the usable storage capacity is allocated to the DATA and RECO disk groups for each configuration option. The usable storage capacity is the storage that available for Oracle Database files after taking into account high-redundancy ASM mirroring (triple mirroring), which is used to provide highly resilient database storage on all Exadata Cloud Machine configurations. The usable storage capacity does not factor in the effects of Exadata compression capabilities, which can be used to increase the effective storage capacity.

Usable Storage Statistic	Eighth Rack	Quarter Rack	Half Rack	Full Rack
<b>Total Usable Storage Capacity</b>	42 TB	84 TB	168 TB	336 TB
<b>— Allocation when Database Backups on Exadata Storage are provisioned</b>	DATA: 16.8 TB RECO: 25.2 TB	DATA: 33.6 TB RECO: 50.4 TB	DATA: 67.2 TB RECO: 100.8 TB	DATA: 134.4 TB RECO: 201.6 TB
<b>— Allocation when Database Backups on Exadata Storage are not provisioned</b>	DATA: 33.6 TB RECO: 8.4 TB	DATA: 67.2 TB RECO: 16.8 TB	DATA: 134.4 TB RECO: 33.6 TB	DATA: 268.8 TB RECO: 67.2 TB

## About Exadata Cloud Machine Database Deployments

When you create a new database deployment on Oracle Database Exadata Cloud Machine, you use the Create Service wizard, which steps you through the process of

making the choices that produce a database deployment tailored to your needs. These choices include:

- [Service Level](#)
- [Metering Frequency](#)
- [Oracle Database Software Release](#)
- [Oracle Database Software Edition](#)
- [Oracle Database Type](#)
- [Automatic Backup Configuration](#)

## Service Level

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

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

## Metering Frequency

When creating a database deployment on Oracle Database Exadata Cloud Machine, monthly metering is the only available option.

---

---

**Note:** With Exadata Cloud Machine, you are billed for each Exadata system that you use, and not for each database deployment that you use.

---

---

## Oracle Database Software Release

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

---

---

**Note:**

The Oracle Database software release version that you select also determines the version of Oracle Grid Infrastructure that is configured:

- If you select Oracle Database 12c Release 2 (12.2) for your starter database, which is the very first database deployment that you create after the creation of your Exadata Cloud Machine instance, then Oracle Grid Infrastructure 12c Release 2 is installed and used to support all of your database deployments, including subsequent deployments that use an earlier Oracle Database release version.
  - If you select Oracle Database 11g Release 2 (11.2) or Oracle Database 12c Release 1 (12.1) for your starter database, then Oracle Grid Infrastructure 12c Release 1 is installed and can only be used to support version 11.2 or version 12.1 database deployments. In this case, you cannot later use the Create Service wizard to create a version 12.2 database deployment.
  - If you wish 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 Machine, 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 Machine, Database Clustering with RAC is the only available choice. This results in a clustered database that uses Oracle Real Application Clusters, with a clustered database instance on each database server in the Exadata Cloud Machine environment.

## Automatic Backup Configuration

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

- **Remote storage** — uses a remote NFS location.

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

- **Remote Storage Only** — uses remote NFS storage 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.

---

**Note:** Automatic backups cannot be configured later if you select the **None** option when you create a database deployment.

---

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

To access the My Services dashboard:


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

The Sign In page opens.

2. Sign in with your Oracle Database Exadata Cloud Machine credentials.

The My Services dashboard opens.

To access the Oracle Database Cloud Service console:

1. Click the  navigation menu in the top corner of the Exadata tile and then click **Open Service Console**.

The Oracle Database Cloud Service console opens and displays the [Services Page](#), which contains a list of database deployments.

2. If a Welcome page is displayed, click **Services** next to Database Cloud Service to display the [Services Page](#).

## Typical Workflow for Using Exadata Cloud Machine

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

Task	Description	More Information
Create an SSH key pair	Create SSH public/private key pairs to facilitate secure access to the compute nodes associated with your database deployments.	<a href="#">Generating a Secure Shell (SSH) Public/Private Key Pair</a>
Create a service instance.	Use a wizard to create a new service instance, which provisions the Exadata Database Machine that hosts your database deployments.	<a href="#">Creating an Exadata Cloud Machine Instance</a>
Create a database deployment	Use a wizard to create a new database deployment.	<a href="#">Creating a Database Deployment</a>
Load data into the database	Use standard Oracle Database tools to load data into your databases.	<a href="#">Loading Data into the Oracle Database on Exadata Cloud Machine</a>
Monitor database deployments	Check on the health and performance of individual database deployments.	<a href="#">Monitoring and Managing Oracle Database on Exadata Cloud Machine</a>
Patch a database deployment	Apply a patch or roll back a patch.	<a href="#">Patching Exadata Cloud Machine</a>

<b>Task</b>	<b>Description</b>	<b>More Information</b>
Back up a database deployment	Back up a database or restore a database from a backup.	<a href="#">Backing Up and Restoring Databases on Exadata Cloud Machine</a>



---

# Managing the Exadata Cloud Machine Life Cycle

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

## Topics

- [Creating an Exadata Cloud Machine Instance](#)
- [Creating a Database Deployment](#)
- [Viewing All Database Deployments](#)
- [Viewing Detailed Information for a Database Deployment](#)
- [Viewing Activities for Database Deployments in an Identity Domain](#)
- [Stopping, Starting and Restarting Compute Nodes](#)
- [Scaling Exadata Cloud Machine](#)
- [Deleting a Database Deployment](#)
- [Deleting an Exadata Cloud Machine Instance](#)

## Creating an Exadata Cloud Machine Instance

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

### Procedure

To create an Exadata Cloud Machine 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 select **Exadata** from the list of options.  
The Create New Oracle Database Exadata Cloud Service Instance wizard starts and the Instance Details page is displayed.
3. On the Instance Details page, configure details for your Exadata Cloud Machine instance. Then, click **Next**.
  - a. In the **Instance Details** section, specify the following attributes associated with your Exadata Cloud Machine instance.
    - **Name** — enter a name for your service instance. This name:
      - Must not exceed 25 characters.
      - Must start with a letter.
      - Must contain only lower case letters and numbers.
      - Must not contain spaces or any other special characters.
      - Must be unique within the identity domain.
    - **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 Machine only one plan is available.
    - **Rack Size** — select the rack configuration for your service instance:
      - **Eighth Rack** — configures an environment consisting of 2 compute nodes and 3 Exadata Storage Servers.
      - **Quarter Rack** — configures an environment consisting of 2 compute nodes and 3 Exadata Storage Servers, but with double the storage capacity and greater CPU capacity compared to an Eighth Rack. See [Exadata System Configuration](#).
      - **Half Rack** — configures an environment consisting of 4 compute nodes and 6 Exadata Storage Servers.
      - **Full Rack** — configures an environment consisting of 8 compute nodes and 12 Exadata Storage Servers.

Your subscription may impose limits on the available rack sizes so that you will only see the rack sizes associated with your available subscriptions.
    - **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 Machine instance.

See [Exadata System Configuration](#) for details about the minimum and maximum number of CPU cores that are available for each Exadata rack size.

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

- **Exadata System Name** — enter a name for your Exadata Database Machine environment. This name is also used as the cluster name for the Oracle Grid Infrastructure installation.
- **Database backups on Exadata Storage** — check this option to configure the storage to enable local database backups on Exadata storage.

---

**Note:** Take care when setting this option because your choice has a profound affect 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.

---

- 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 Confirmation page, review the configuration settings. If you are satisfied, click **Create Service Instance**.

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 service instance.

Clicking **Create Service Instance** 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.

## Creating a Database Deployment

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

However, before using the Create Service 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 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 Machine, the Create Service 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.

- An NFS remote backup location (Optional)

When creating a database deployment, you can choose to configure automatic backups. To do so, you must provide an NFS remote backup location. The location, must be specified using one of the following formats:

```
hostname: absolute-path  
ip-address: absolute-path
```

## Procedure

To create a database deployment on Exadata Cloud Machine:

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 Service**.  
The Create Service wizard starts.
3. On the Service page, specify basic attributes for your database deployment. Then, click **Next**.
  - **Service Name** — enter a name for your database deployment. This name:
    - Must not exceed 50 characters.
    - Must start with a letter.
    - Must contain only letters, numbers, or hyphens.
    - Must not end with a hyphen.
    - Must not contain any other special characters.
    - Must be unique within the identity domain.
  - **Description** — enter a description for your database deployment. (Optional)

- **Exadata System** — select an available Oracle Exadata Database Machine configuration. The list contains the Oracle Exadata Database Machines that are associated with your existing subscriptions.
- **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.
- **Metering Frequency** — the only valid option for use with Exadata Cloud Machine is **Monthly**.
- **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**

---

---

**Note:**

The Oracle Database software release version that you select also determines the version of Oracle Grid Infrastructure that is configured:

- If you select Oracle Database 12c Release 2 (12.2) for your starter database, which is the very first database deployment that you create after the creation of your Exadata Cloud Machine instance, then Oracle Grid Infrastructure 12c Release 2 is installed and used to support all of your database deployments, including subsequent deployments that use an earlier Oracle Database release version.
- If you select Oracle Database 11g Release 2 (11.2) or Oracle Database 12c Release 1 (12.1) for your starter database, then Oracle Grid Infrastructure 12c Release 1 is installed and can only be used to support version 11.2 or version 12.1 database deployments. In this case, you cannot later use the Create Service wizard to create a version 12.2 database deployment.
- If you wish 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](#).

---

---

**Note:**

If you proceed and attempt to deploy Oracle Database 12c Release 2 on a system running with Oracle Grid Infrastructure 12c Release 1, then the deployment will fail and an error message will be returned.

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

- **Database Type** — the only valid option for use with Exadata Cloud Machine is **Database Clustering with RAC**.
4. On the Service Details page, configure details for your database deployment. Then, click **Next**.
- a. In the **Database Configuration** section, set the administrator password, database name (SID), PDB name (for Oracle Database 12c), and other database configuration options.
- **DB Name (SID)** — enter a name for the database instances. This name:
    - Must not exceed 8 characters.
    - Must start with a letter.
    - Must contain only letters and numbers.
  - **PDB Name** (Available only for Oracle Database 12c) — enter a name for the default PDB (pluggable database). This name:
    - Must not exceed 8 characters.
    - Must start with a letter.
    - Must contain only letters and numbers.
  - **Administration Password** and **Confirm Password** — enter and then re-enter a password for the Oracle Database SYS and SYSTEM users.  
The password you enter:
    - Must be 8 to 30 characters in length.
    - Must contain at least one lowercase letter
    - Must contain at least one uppercase letter
    - Must contain at least one number
    - Must contain at least one of these symbols: \_ (underscore), # (hash sign), or – (dash or minus sign).

---

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

---

- **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 Machine instance. Subsequent database deployments are created with a standardized database configuration.

---

- **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 Machine environment already contains a previously specified SSH key.

---

- Optionally, expand **Advanced Settings** and set the following:
  - **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 Machine](#).

- 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:

- **Remote Storage Only** — uses remote NFS storage 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.

---

---

**Note:** Automatic backups cannot be configured later if you select the **None** option when you create a database deployment.

---

---

If you select **Remote Storage Only**, the NFS Remote Backup field is displayed:

- **NFS Remote Backup** — enter the path of the NFS remote backup location where backups of the database deployment are to be stored. This path has one of the following formats:

*hostname: absolute-path*  
*ip-address: absolute-path*

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, network access to the deployment is restricted to SSH connections on port 22. To open access to applications and management tools, you need to create and enable your own network security rules. See [Enabling Access to a Compute Node Port](#).

- **Update cloud tooling**

While the base images used to create Exadata Cloud Machine 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 Machine](#).

- **Apply database patches**

While the base images used to create Exadata Cloud Machine 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](#).

## Viewing All Database Deployments

From the Oracle Database Cloud Service Console, you can:

- View the total resources allocated across all Oracle Database Exadata Cloud Machine 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 [Services 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 [Services 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 Machine, 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 Machine 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 Oracle Database Cloud Service console.

For detailed instructions, see [Accessing the My Services Dashboard and the Oracle Database Cloud Service Console](#).

2. 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.

3. 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 nodes associated with a database deployment on Oracle Database Exadata Cloud Machine.

---

---

**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, when you stop and start a compute node by using the Oracle Database Cloud Service console, you destroy and recreate the virtual domain that underpins the compute node virtual machine. If you use an operating system command to stop and start the compute node, then the compute node virtual machine is restarted within the same domain. 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 nodes associated with an Exadata Cloud Machine 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 Machine databases that share the same Oracle Exadata Database Machine. If you stop all of the compute nodes associated with an Exadata Cloud Machine environment, you effectively stop all of your databases running on the Oracle Exadata Database Machine.

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  menu.

- From the  menu 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:

- Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing the My Services Dashboard and the Oracle Database Cloud Service Console](#).

- 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  menu.

- From the  menu 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, the compute node is stopped and then immediately started again.


To restart a compute node:


- Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing the My Services Dashboard and the Oracle Database Cloud Service Console](#).

- 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  menu.

- From the  menu 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 by viewing the activity log:

- Open the Oracle Database Cloud Service console.  
For detailed instructions, see [Accessing the My Services Dashboard and the Oracle Database Cloud Service Console](#).
- In the list, click the name of the database deployment whose past activity you want to view.  
The Oracle Database Cloud Service Overview page is displayed.
- 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.
- 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 Exadata Cloud Machine

Two kinds of scaling operations are supported for an Oracle Database Exadata Cloud Machine 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 a service instance requires more compute node processing power, you can scale up the number of enabled CPU cores in the Oracle Exadata Database Machine. You can temporarily modify the compute node processing power (bursting) or add compute node processing power on a more permanent basis.

The minimum and maximum number of enabled CPU cores depends on the system configuration in question, however your subscription may impose additional limits:

Statistic	Eighth Rack	Quarter Rack	Half Rack	Full Rack
Minimum Number of Enabled CPU Cores	16	22	44	88

Statistic	Eighth Rack	Quarter Rack	Half Rack	Full Rack
Maximum Number of Enabled CPU Cores	68	84	168	336


To modify the number of enabled CPU cores within an existing Exadata Cloud Machine 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  located in the Exadata 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  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, use the slider control to set the number of enabled CPU cores on each compute node, also select the type of scaling operation that you want to perform. Then, click **Next**.

- a. 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.

---

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

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

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

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

With bursting, you can quickly scale up beyond your subscription level to cater for workload peaks. You can also scale back to the subscription level at any time. CPU cores beyond your subscription level are charged separately using an hourly rate for the bursting period.

---

---

**Note:** With bursting, the maximum number of enabled CPU cores is limited to twice the number of CPU cores in the associated service subscription. For example, if your service subscription contains 8 enabled CPU cores on each compute node, then the bursting maximum is 16 CPU cores on each compute node.

---

---

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.

---

---

**Note:** Modifying the number of enabled CPU cores is an online operation, which does not require a reboot of the affected compute nodes. However, if you have explicitly set the `CPU_COUNT` 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. 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 can be achieved by expanding your existing Exadata system. Such expansion requires the installation of additional hardware in the compute nodes and the Exadata Storage Servers. Consequently, individual compute nodes and Exadata Storage Servers must be stopped and restarted to perform the expansion. However, the entire process can be achieved in a rolling manner to ensure that service availability is maintained throughout.

## Deleting a Database Deployment

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

To delete 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. Select **Delete** from the  menu corresponding with 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.

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

## Deleting an Exadata Cloud Machine Instance

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

### Procedure

To delete an Exadata Cloud Machine 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  located in the Exadata tile and choose **View Details**.

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

3. Select **Delete** from the  menu corresponding with 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.

---

---





---

## Managing Network Access to Exadata Cloud Machine

By default, network access to Exadata Cloud Machine is provided by using SSH. The SSH connection uses the SSH key specified during the process to create a database deployment. By default, port 22 is used for SSH connections. To access other ports you must perform additional configuration tasks, such as creating an SSH tunnel or enabling access to the port.

### Topics

- [About Network Access to Exadata Cloud Machine](#)
- [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#)
- [Creating an SSH Tunnel to a Compute Node Port](#)
- [Enabling Access to a Compute Node Port](#)
- [Controlling Network Access to Exadata Cloud Machine](#)
- [Defining a Custom Host Name or Domain Name for Exadata Cloud Machine](#)
- [Defining a Custom SCAN Host Name for Exadata Cloud Machine](#)
- [Using Network Encryption and Integrity](#)

### About Network Access to Exadata Cloud Machine

By default, network access to the compute nodes associated with Oracle Database Exadata Cloud Machine is provided by Secure Shell (SSH) connections on port 22. To access other network protocols and services requires 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 SSH servers.

The public key is stored in the Exadata Database Machine compute nodes associated with your Exadata Cloud Machine environment. If no public key is associated with your Exadata Cloud Machine environment you will be prompted to specify a public key when you create a database deployment. You can modify the stored keys by using the SSH Access page.

When you access any Exadata Cloud Machine 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](#).

### Access to Other Ports

To access network protocols and services on a compute node by using a port other than port 22, you must either:

- Enable network access to the port  
You can enable access to a specific compute node port from specific hosts. See [Enabling Access to a Compute Node Port](#)
- 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, you must have the SSH private key file that matches the public key specified during the database deployment creation process. See [Creating an SSH Tunnel to a Compute Node Port](#).

## 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 Machine, 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 Machine 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 Machine 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 Machine 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 you specified when creating the tunnel.

## Enabling Access to a Compute Node Port

To enable access to a specific port on the compute nodes associated with your Oracle Database Exadata Cloud Machine environment, you must submit a Service Request to Oracle Support. When you submit the Service Request, you must identify the Exadata system involved and the port that you wish to open. You will be notified through the Service Request when the port is enabled. See [How to Request Service Configuration for Oracle Database Exadata Cloud Service](#).

## Controlling Network Access to Exadata Cloud Machine

You can control network access to your Oracle Database Exadata Cloud Machine by listing network addresses that are either invited to connect, or excluded from connecting. This control is provided in different ways:

- You can define a white-list of clients that are allowed access through the firewall surrounding your Exadata Cloud Machine environment. See [Enabling Access to a Compute Node Port](#).

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 valid node checking to define a list that Oracle Net uses to allow or disallow connections from. You enable and control valid node checking by setting parameters in the `sqlnet.ora` file. Oracle Net valid node checking only controls Oracle Net connections. Connections by other means, such as SSH, are not arbitrated by Oracle Net valid node checking.

---

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

---

For Exadata Cloud Machine, your Oracle Net listeners use Oracle Grid Infrastructure software, which is typically installed under the `ORACLE_HOME` directory `/u01/app/12.1.0.2/grid` or `/u01/app/12.2.0.1/grid`, depending on which Oracle Grid Infrastructure version is in use. Therefore, to enable Oracle Net valid node checking, set `TCP.VALIDNODE_CHECKING = yes` inside `/u01/app/12.1.0.2/grid/network/admin/sqlnet.ora` or `/u01/app/12.2.0.1/grid/network/admin/sqlnet.ora`. To control Oracle Net valid node checking use the following parameters:

- `TCP.EXCLUDED_NODES` specifies clients that are denied access to the database. The parameter can be set to a list of host names or addresses and

the list may include wildcards for IPv4 addresses and CIDR (Classless Inter-Domain Routing) notation for IPv4 and IPv6 addresses. For example:

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

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

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

## Defining a Custom Host Name or Domain Name for Exadata Cloud Machine

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 Machine 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 Machine 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 Machine

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 Machine is associated with a SCAN, and in turn 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 maximise 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 Machine. 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 Machine databases, you can use native Oracle Net 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 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 Machine are configured to enable native Oracle Net encryption and integrity. Also, by default, Oracle Net clients are configured to enable native encryption and integrity when they connect to an appropriately configured server. If your Oracle Net 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 encryption and integrity as follows. For more general information about configuring native Oracle Net encryption and integrity, see "Configuring Network Data Encryption and Integrity" in *Oracle Database Security Guide* for Release 12.2 or 12.1 or in *Database Advanced Security Administrator's Guide* for Release 11.2.

### Checking your Exadata Cloud Machine environment

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

---



---

**Note:** The procedure relates to a single compute node. For Exadata Cloud Machine, you should confirm that the configuration settings are consistent across all of the compute nodes 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. Change directories to the location of the `sqlnet.ora` Oracle Net configuration file. For example:

```
$ cd /u01/app/12.1.0.2/grid/network/admin
$ ls sqlnet.ora
sqlnet.ora
```

---

**Note:** For Exadata Cloud Machine, your Oracle Net listeners use Oracle Grid Infrastructure software, which is typically installed under the `ORACLE_HOME` directory `/u01/app/12.1.0.2/grid` or `/u01/app/12.2.0.1/grid`, depending on which Oracle Grid Infrastructure version is in use.

---

3. 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 Machine.

### Checking your Oracle Net Client Configuration

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

1. In a command shell, connect to the Oracle Net client.
2. Change directories to the location of the Oracle Net configuration files `tnsnames.ora` and `sqlnet.ora`, 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 encryption and integrity are enabled in your Exadata Cloud Machine 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 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 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
```



---

# Administering Exadata Cloud Machine

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

## Topics

- [Using Exadata I/O Resource Management](#)
- [Adding an SSH Public Key](#)
- [Removing an SSH Public Key](#)
- [Updating the Cloud Tooling on Exadata Cloud Machine](#)
- [Maintaining the Manageability of Exadata Cloud Machine](#)
- [Loading Data into the Oracle Database on Exadata Cloud Machine](#)
- [Tuning Oracle Database Performance on Exadata Cloud Machine](#)
- [Monitoring and Managing Oracle Database on Exadata Cloud Machine](#)

## Using Exadata I/O Resource Management

Oracle Database Exadata Cloud Machine 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 Machine 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 proportionately 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. From the  menu for an Exadata Cloud Machine database deployment, 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 Exadata system.
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 Machine environment. After you add the public key, you can provide the matching private key to connect to a compute node associated with the Exadata Cloud Machine instance as either the `opc` or the `oracle` user.

To add an SSH public key:

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. From the  menu for a database deployment, select **SSH Access**.

The **Add New Key** overlay is displayed.

3. 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.
4. Click **Add New Key**.

You can also add SSH public keys to one or more deployments on the [SSH Access Page](#).

---

**Note:** Although you can add an SSH key using the  menu for a database deployment, every SSH key provides system-wide access to the Exadata Database Machine compute nodes associated with the Exadata Cloud Machine instance. 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 Machine environment. After you remove the public key, you can no longer use the matching private key to connect to a compute node associated with the Exadata Cloud Machine instance 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 Machine environment.

---

**Note:** The following describes the procedure for each compute node and must be repeated across all of your Exadata Cloud Machine 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 minimise 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 Machine

You can update the cloud-specific tooling included on an Exadata Cloud Machine 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 Machine environment. Therefore, the following procedure should be repeated for every Exadata Cloud Machine 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. Display information about the installed cloud tooling and note down the information:

```
# rpm -qa|grep -i dbaastools_exa  
dbaastools_exa-version_number-release_string
```

4. Check whether any cloud tooling updates are available:

```
# /var/opt/oracle/exapatch/exadbcpatch -list_tools
```

5. Examine the command response, and determine the patch ID of the available cloud tooling update.

The patch ID is listed in the `patches` group as the `patchid` value.

Cloud tooling updates are cumulative. So if multiple updates are available, you can simply install the latest update. There is no need to install all of the updates in order.

6. If the available patch is newer than the currently installed tools, download and apply the patch containing the cloud tooling update:



---

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

where *patchid* is the patch ID that you located in the previous step.

---

**Note:** 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
```

---

7. Exit the root-user command shell:

```
# exit
$
```

After you update the cloud tooling across your environment, you should also re-configure the automatic database backups to use the updated cloud tooling by using the following procedure:

1. Connect to the first compute node in your Exadata system as the `opc` user.

For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

---

**Note:**

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---

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 Machine environment that has an automatic backup configuration.

```
# /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 by examining the entries in the `/etc/crontab` file on the first compute node in your Exadata environment.

---

4. Exit the root-user command shell:

```
# exit  
$
```

## Maintaining the Manageability of Exadata Cloud Machine

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

To keep your Exadata Cloud Machine instances manageable, follow these guidelines:

- Do not change the compute node OS users or manually manipulate SSH key settings associated with your Exadata Cloud Machine instance.
- Apply **only** patches that are available through Exadata Cloud Machine. 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, Enterprise Manager Database Express 12c, or Enterprise Manager 11g Database Control.

## Loading Data into the Oracle Database on Exadata Cloud Machine

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

The following sections outline several common tools and techniques used to load data into an Oracle database. Also, see [Migrating Oracle Databases to Exadata Cloud Machine](#) for additional techniques and more specific information about migrating existing Oracle databases to Exadata Cloud Machine.

### 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 [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 [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 [12.2](#), [12.1](#) or [11.2](#).

### Using Pluggable Databases (PDBs) to Load Data into the Database

The multitenant architecture of Oracle Database 12c 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 Machine, provided that the source data is already inside a PDB on Oracle Database 12c.

For information about PDBs and how to unplug, move, and plug them, see "Overview of Managing a Multitenant Environment" in *Oracle Database Administrator's Guide* for Release [12.2](#) or [12.1](#).

## Tuning Oracle Database Performance on Exadata Cloud Machine

You tune the performance of Oracle Database on Oracle Database Exadata Cloud Machine using the same tools you would use for an Oracle database running on any system in your data center. The fact that the database is housed in the Oracle Cloud does not place any restrictions on performance tuning.

The *Oracle Database Performance Tuning Guide* for Release [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 Machine 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 Machine to satisfy the need. See [Scaling Exadata Cloud Machine](#).

## Monitoring and Managing Oracle Database on Exadata Cloud Machine

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

- 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](#).



---

## Accessing Exadata Cloud Machine

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

### Topics

- [Connecting to a Compute Node Through Secure Shell \(SSH\)](#)
- [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 Machine, 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 Machine.

### 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 Machine 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 Machine 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 of 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 12c

Enterprise Manager Database Express 12c (EM Express), a web-based tool for managing Oracle Database 12c, is available on Oracle Database Exadata Cloud Machine database deployments created using Oracle Database 12c Release 1 (12.1) or Oracle Database 12c Release 2 (12.2).

To access EM Express:

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. From the  menu for the deployment, select **Open EM Console**.

The EM Express login page is displayed.

3. Enter **SYSTEM** as the user name, enter the password specified when the database deployment was created, and then click **Login**. To connect with SYSDBA privileges, select the check box next to **as sysdba**, enter **SYS** as the user name, enter the password specified when the database deployment was created, and then click **Login**.

This option is also available from the  menu on the Oracle Database Cloud Service Instance Overview page.

Alternatively, 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.

## 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 Machine database deployments created using Oracle Database 11g Release 2.


Database Control is allocated a unique port number for each database deployment. By default, access to Database Control is provided using port 1158 for the first deployment. Subsequent deployments are allocated ports in a range starting with 5500, 5501, 5502, and so on.

---

**Note:** You can confirm the Database Control port for a database by searching for `REPOSITORY_URL` in the `$ORACLE_HOME/host_sid/sysman/config/emd.properties` file.

---

To access Database Control:

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. From the  menu for the deployment, select **Open EM Console**.  
The Database Control login page is displayed.
3. Enter **SYSTEM** as the user name, enter the password specified when the database deployment was created, and then click **Login**. To connect with SYSDBA privileges, select **SYSDBA** from the drop-down list, enter **SYS** as the user name, enter the password specified when the deployment was created, and then click **Login**.

This option is also available from the  menu on the Oracle Database Cloud Service Instance page.

Alternatively, you can access Database Control by directing your browser to the 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.



## Connecting Remotely to the Database by Using Oracle Net Services

Oracle Database Exadata Cloud Machine supports remote database access by using Oracle Net Services.

Because Exadata Cloud Machine 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. To maximise 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 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

For security reasons, the SSH port (22) is by default the only accessible network port in your Exadata Cloud Machine environment. However, by default the Oracle Net Listeners (SCAN listeners and node listeners) listen on port 1521. Therefore, before you can connect remotely to the database by using Oracle Net Services, you must enable access to the Oracle Net Listener port. For Exadata Cloud Machine you can:

- Specifically enable network access to the Oracle Net Listener port. See [Enabling Access to a Compute Node Port](#). If you specifically enable access to the Oracle Net Listener port, ensure that you always use an encrypted Oracle Net connection. See [Using Network Encryption and Integrity](#).

---

---

**Note:**

An SSH tunnel cannot be used to connect to an Exadata Cloud Machine 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 enable access to 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, 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 Machine environment; for example:

PDB1.us2.oraclecloud.com

You can determine the network domain name associated with your Exadata Cloud Machine environment by viewing details as described in [Viewing Detailed Information for a Database Deployment](#).

### Creating an Oracle Net Connection by Using SCAN

To create an Oracle Net 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 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 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, 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 in turn the connection is handled by one of the SCAN listeners. See [Defining a Custom SCAN Host Name for Exadata Cloud Machine](#).

To create a Oracle Net 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:

```
exalscan.example.com:1521/ORCL
```

or

```
exalscan.example.com:1521/PDB1.us2.oraclecloud.com
```



---

# Backing Up and Restoring Databases on Exadata Cloud Machine

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

## Topics

- [About Backing Up Database Deployments on Exadata Cloud Machine](#)
- [Creating an On-Demand Backup](#)
- [Deleting a Backup](#)
- [Customizing the Current Backup Configuration](#)
- [Disabling and Re-enabling Scheduled Backups](#)
- [Restoring from the Most Recent Backup](#)
- [Restoring from a Specific Backup](#)
- [Restoring to a Specific Point in Time](#)
- [Manually Restoring from a Backup](#)

## About Backing Up Database Deployments on Exadata Cloud Machine

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

### About Automatic Database Backups

Exadata Cloud Machine 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.

When you create a database deployment on Exadata Cloud Machine, you must choose from the following automatic backup configuration options:

- **Remote Storage Only** — uses remote NFS storage 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.

---

**Note:** Automatic backups cannot be configured later if you select the **None** option when you create a database deployment.

---

## Default Automatic Database Backup Configuration

The default automatic backup configuration follows a set of Oracle best-practice guidelines:

- Automatic backups are scheduled daily at 1:01 AM (system time).
- Backups consist of full (RMAN level 0) backups of the database, followed by daily incremental (RMAN level 1) backups, with a 7 day cycle between full backups.
- The retention period defines the period for which backups are maintained. For *Remote Storage Only* the retention period is 30 days.
- After the initial retention period, each day when an incremental backup is taken, the oldest daily incremental backup is automatically merging into the oldest full backup.
- The backup data is automatically encrypted.

You can customize some aspects of the backup configuration for your database deployment. See [Customizing the Current Backup Configuration](#).

## Viewing Available Automatic Database Backups

You can get a list of the available automatic backups and determine their location using the RMAN `list backup` command:

- To view a brief summary list of backups, which includes backup keys, timestamps and status codes, use the `list backup summary` command. For example:

```
RMAN> list backup summary;
```

```
using target database control file instead of recovery catalog
```

```
List of Backups
```

```
=====
```

```
Key      TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
```

```
-----
```

```
83      B A A SBT_TAPE 26-MAY-15 1 1 NO
```

```
DBAAS_INCR_BACKUP
```

```
84      B A A SBT_TAPE 26-MAY-15 1 1 NO
```

```
DBAAS_INCR_BACKUP
```

```
...
```

- To view a list of backups, which includes more detailed information including the location and contents of each backup piece, use the `list backup` command. For example:

```
RMAN> list backup;
```

```
List of Backup Sets
```

```
=====
```

```
BS Key Size Device Type Elapsed Time Completion Time
```

```
-----
```

```
83 180.50M SBT_TAPE 00:06:05 26-MAY-15
```

```
BP Key: 83 Status: AVAILABLE Compressed: NO Tag: DBAAS_INCR_BACKUP
```

```
Handle: cloudstorage_ORCL_6vq7tq5m_1_1 Media:
```

```
storage.us2.oraclecloud.com/v1/dbaastest-usoracle05695/dbaasdev0
```

```
List of Archived Logs in backup set 83
```

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	58	2926534	25-MAY-15	2956852	25-MAY-15
1	59	2956852	25-MAY-15	2959302	25-MAY-15
1	60	2959302	25-MAY-15	2982117	26-MAY-15
1	61	2982117	26-MAY-15	3000489	26-MAY-15

```
BS Key Size Device Type Elapsed Time Completion Time
```

```
84 302.75M SBT_TAPE 00:10:05 26-MAY-15
BP Key: 84 Status: AVAILABLE Compressed: NO Tag: DBAAS_INCR_BACKUP
Handle: cloudstorage_ORCL_6tq7tq5l_1_1 Media:
storage.us2.oraclecloud.com/v1/dbaastest-usoracle05695/dbaasdev0
```

```
List of Archived Logs in backup set 84
```

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	54	2824749	24-MAY-15	2844444	24-MAY-15
1	55	2844444	24-MAY-15	2871852	24-MAY-15
1	56	2871852	24-MAY-15	2898243	24-MAY-15
1	57	2898243	24-MAY-15	2926534	25-MAY-15

```
...
```

For information about using RMAN, see [Oracle Database Backup and Recovery User's Guide](#).

### Additional Database Backup Options

In addition to the automatic database backup capabilities provided by Exadata Cloud Machine, 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 remote storage locations as the automatic database backups provided by Exadata Cloud Machine, or they may use other storage locations. You can also create manual backups on local Exadata storage in your Exadata Cloud Machine. For this option it is recommended that you provision for database backups on Exadata storage in your Exadata Cloud Machine. For more information, see [Exadata Storage Configuration](#).

If you want to use Exadata Cloud Machine in conjunction with Oracle Zero Data Loss Recovery Appliance (ZDLRA), you can use the supplied ZDLRA library located at `/var/opt/oracle/ocde/assistants/bkup/ra_installer.zip` on the Exadata Cloud Machine compute nodes. See [Installing the Recovery Appliance Backup Module](#) for further information.

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.

## Creating an On-Demand Backup

You can create an on-demand backup of an Oracle Database Exadata Cloud Machine 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 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** and then confirm the action.  
The backup process begins.

### 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 as follows:

1. Connect to the first compute node in your Exadata system as the `opc` user.  
For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

---



---

#### Note:

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---



---

2. Start a root-user command shell:
 

```
$ sudo -s
#
```
3. You can choose to have the backup follow the current retention policy, or you can choose to create a long-term backup that persists until you delete it:
  - To create a backup that follows the current retention policy, enter 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 a long-term backup, enter the following `bkup_api` command:



```
# /var/opt/oracle/bkup_api/bkup_api bkup_start --keep --dbname=dbname
```

4. Exit the root-user command shell and disconnect from the compute node:

```
# exit
$ exit
```

By default, the 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", enter the following command:

```
# /var/opt/oracle/bkup_api/bkup_api bkup_start --keep --tag=monthly --dbname=dbname
```

After you enter a `bkup_api bkup_start` command, the `bkup_api` utility starts the backup process, which runs in the background. To check the progress of the backup process, enter the following `bkup_api` command:

```
# /var/opt/oracle/bkup_api/bkup_api bkup_status --dbname=dbname
```

## Deleting a Backup

To delete a backup of a database deployment on Oracle Database Exadata Cloud Machine, you use the `bkup_api` utility.

1. Connect to the first compute node in your Exadata system as the `opc` user.

For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

---



---

### Note:

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---



---

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. List the available backups:

```
# /var/opt/oracle/bkup_api/bkup_api recover_list --dbname=dbname
```

where ***dbname*** is the database name for the database that you want to act on.

A list of available backups is displayed.

4. Delete the backup you want:

```
# /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
$
```

## Customizing the Current Backup Configuration

You can customize many of the characteristics of the automatic backup configuration.

### Topics

- [Customizing How the Database Is Backed Up](#)
- [Customizing the Retention Period for Backups](#)
- [Customizing the Cycle Period for Backups](#)
- [Customizing the Frequency of Automatic Backups](#)

### Customizing How the Database Is Backed Up

To change how the Oracle database is backed up, you use the RMAN utility. For information about using RMAN, see *Oracle Database Backup and Recovery User's Guide* for Release [12.2](#), [12.1](#) or [11.2](#).

---

---

**Caution:**

Do not use the RMAN utility to change the retention period.

---

---

To view the current RMAN configuration, use the RMAN command `SHOW ALL`:

1. Connect as the **oracle** user to the compute node.  
For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

2. Configure your environment settings, including `$ORACLE_SID` and `$ORACLE_HOME`, for the database that you wish to customize.

3. Start an RMAN session:

```
$ rman target=/  
...  
RMAN>
```

4. Enter the `SHOW ALL` command:

```
RMAN> show all;  
...
```

A listing of your configuration is displayed.

5. Use RMAN commands to make any changes to your configuration.

6. Exit the RMAN session:

```
RMAN> exit;  
$
```

### Customizing the Retention Period for Backups

To change the retention period for backups, use the `bkup_api` utility:

---



---

**Note:** For Exadata Cloud Machine, the retention period applies to backups to cloud storage. Backups to local Exadata storage have a retention period that is equal to one cycle between full backups.

---



---

1. Connect to the first compute node in your Exadata system as the **opc** user.

For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

---



---

**Note:**

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---



---

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Enter this `bkup_api` command.

```
# /var/opt/oracle/bkup_api/bkup_api bkup_chgcfg --retention=days --dbname=dbname
```

where **days** is the number of days for which you want to retain backups, and **dbname** is the database name for the database that you want to act on..

4. Exit the root-user command shell:

```
# exit
$
```

### Customizing the Cycle Period for Backups

To change the cycle period for backups, use the `bkup_api` utility:

---



---

**Note:** For Exadata Cloud Machine, the cycle period between full backups also defines the retention period for backups to local Exadata storage.

---



---

1. Connect to the first compute node in your Exadata system as the **opc** user.

For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

---



---

**Note:**

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---



---

2. Start a root-user command shell:

```
$ sudo -s  
#
```

3. Enter this `bkup_api` command.

```
# /var/opt/oracle/bkup_api/bkup_api bkup_chgcfg --cycle=days --dbname=dbname
```

where ***days*** is the number of days you want for the cycle period, and ***dbname*** is the database name for the database that you want to act on..

4. Exit the root-user command shell:

```
# exit  
$
```

### Customizing the Frequency of Automatic Backups

The backup feature provided by Oracle Database Exadata Cloud Machine uses the Linux `cron` job scheduler to perform automatic backups.

If automatic backups are enabled, the following job entry is defined in the system-wide scheduler file, `/etc/crontab`, which is located on the first compute node in your Exadata system:

```
MM HH * * * root /var/opt/oracle/bkup_api/bkup_api bkup_start --dbname=dbname
```

---

---

**Note:**

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---

---

The `/etc/crontab` entry causes the `bkup_api` script to be run daily at `HH:MM` (on a 24-hour clock) by the `root` user. The `bkup_api` script sends log messages to the file `/home/oracle/bkup/log/obkup.log`.

To change this frequency, or to add the entry if automatic backups were not enabled when the database deployment was created, edit the `/etc/crontab` file. You must have root-user access to edit this file, so you must connect as the `opc` user and then run the command `sudo -s` to start a root-user shell.

## Disabling and Re-enabling Scheduled Backups

If some activity you want to perform requires you to temporarily disable regularly scheduled backups, you can do so by removing the scheduling information from the system-wide `/etc/crontab` file on the first compute node in your Exadata environment.

### Disabling Scheduled Backups

To disable scheduled backups:

1. Connect to the first compute node in your Exadata system as the **`opc`** user.

For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

---



---

**Note:**

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---



---

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Navigate to the `/etc` directory, which contains the system-wide `crontab` file:

```
# cd /etc
```

4. Make a copy of the `crontab` file to preserve the configuration, for example:

```
# cp crontab crontab.bak
```

5. Edit the original `crontab` file and remove the following line from the file:

```
01 01 * * * root /home/oracle/bkup/dbname/obkup -dbname=dbname
```

or

```
MM HH * * * root /var/opt/oracle/bkup_api/bkup_api bkup_start --dbname=dbname
```

where *dbname* is the name of the database that you wish to act on and *HH:MM* is the backup time (on a 24-hour clock).

---



---

**Note:** You cannot comment out the line, you must delete it. If your Exadata environment supports multiple database deployments, you may have multiple entries to delete. The format of each entry is determined by the version of the cloud tooling in your Exadata Cloud Machine environment at the time when the database deployment was created.

---



---

6. Save the file and exit from the editor.

7. Exit the root-user command shell and disconnect from the compute node:

```
# exit
$ exit
```

## Re-Enabling Scheduled Backups

To re-enable scheduled backups:

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. Copy the `crontab.bak` file you created when disabling scheduled backups to its original name, `crontab`:

```
# cp /etc/crontab.bak /etc/crontab
```

4. Exit the root-user command shell and disconnect from the compute node:

```
# exit
$ exit
```

## Restoring from the Most Recent Backup

You can restore the most recent backup and perform complete recovery on an Oracle Database Exadata Cloud Machine 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. 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 database deployment you want to restore and recover.

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 **Recover**.

The Database Recovery overlay is displayed.

5. 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 Oracle Recovery Manager (RMAN) to manually restore a database on Exadata Cloud Machine. See [Manually Restoring from a Backup](#).

## Restoring from a Specific Backup

You can restore a specific backup and perform recovery to that backup on an Oracle Database Exadata Cloud Machine 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. 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 database deployment you want to restore and recover.  
The Oracle Database Cloud Service Overview page is displayed.
3. Click the Administration tile.  
The Oracle Database Cloud Service Backup page is displayed.
4. In the list of backups, locate the backup you want to restore from.
5. In the entry for the backup you want to restore from, click the  menu, 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 Oracle Recovery Manager (RMAN) to manually restore a database on Exadata Cloud Machine. See [Manually Restoring from a Backup](#).

## 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 Machine 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. 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 database deployment you want to restore and recover.  
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 **Recover**.

The Database Recovery overlay is displayed.

5. 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 time zone setting on the compute node where the recovery is initiated. For Exadata Cloud Machine, this is the first compute node in your Exadata system.

To determine the first compute node, connect to any compute node as the `grid` user and execute the following command:

```
$ $ORACLE_HOME/bin/olsnodes -n
```

The first node has the number 1 listed beside the node name.

---

---

6. 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 Oracle Recovery Manager (RMAN) to manually restore a database on Exadata Cloud Machine. See [Manually Restoring from a Backup](#).

## Manually Restoring from a Backup

Oracle Database Exadata Cloud Machine 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 [12.2](#), [12.1](#) or [11.2](#).



---

# Patching Exadata Cloud Machine

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

## Topics

- [About Patching Exadata Cloud Machine](#)
- [Listing Available Patches](#)
- [Checking Prerequisites Before Applying a Patch](#)
- [Applying a Patch](#)
- [Listing Applied Patches](#)
- [Rolling Back a Patch or Failed 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 [12.2](#), [12.1](#) or [11.2](#).

## About Patching Exadata Cloud Machine

### The Oracle Database Cloud Service Console

You can use the Oracle Database Cloud Service console to perform patching operations for Oracle Database on Exadata Cloud Machine. To access the patching functionality in the Oracle Database Cloud Service console:

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 database deployment on which you want to perform a patching operation.  
The Oracle Database Cloud Service Overview page is displayed.
3. Click the Administration tile and then click the Patching tab.  
The Oracle Database Cloud Service Patching page is displayed.

### The `exadbcpatchmulti` Command

You can also use the `exadbcpatchmulti` utility to perform patching operations for Oracle Grid Infrastructure and Oracle Database on Exadata Cloud Machine. 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.

---

---

**Note:** The `exadbcpatchmulti` command uses the cloud-specific tooling included on your Exadata Cloud Machine 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 Machine](#).

---

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
-sshkey=sshkey_file -oh=hostname:oracle_home
```

- To check prerequisites before applying a patch:

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

- To apply a patch:

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

- To report the status of a patching operation for an Oracle home directory:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -get_status patchtxn
-sshkey=sshkey_file -oh=hostname:oracle_home
```

- To rollback a previously applied patch or a failed patch:

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

The following table describes the arguments shown in the syntax for the `exadbcpatchmulti` command.

Argument	Description
<i>patchid</i>	<p>Identifies the patch to be pre-checked, applied or rolled back.</p> <p>To list the applicable patch identifiers for an Oracle home directory execute the <code>exadbcpatchmulti</code> command with the <code>-list_patches</code> action.</p>
<code>-sshkey = sshkey_file</code>	<p>Specifies an SSH private key associated with the <code>opc</code> user, which is used to connect to compute nodes in the cluster.</p> <p>Typically this file is located at <code>/home/opc/.ssh/id_rsa</code>.</p>
<code>-instanceN =</code> <code>hostname:oracle_home1</code> <code>[,oracle_home2 ...]</code>	<p>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.</p> <p>You must 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.</p>
<code>-run_datasql = 1</code>	<p>Use this argument to execute the SQL commands associated with a patch or rollback operation.</p> <p>This operation should only be performed after all of the compute nodes are patched or rolled back. Therefore, 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.</p> <p>This argument can only be specified in conjunction with a patching or rollback operation acting on a set of compute nodes. Therefore, 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. Refer to the patch documentation for further details.</p>
<code>-oh =</code> <code>hostname:oracle_home</code>	<p>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.</p>
<i>patchtxn</i>	<p>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.</p> <p>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.</p>

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 you can apply to an Exadata Cloud Machine database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to View Available Patches](#) at the end of this topic.

### Viewing Available Patches by Using the Oracle Database Cloud Service Console

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 database deployment on which you want to check patching.  
The Oracle Database Cloud Service Overview page is displayed.
3. 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.

### Other Ways to View Available 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 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
-sshkey=sshkey_file -oh=hostname:oracle_home
```

where:

- `-sshkey` specifies the location of the SSH private key of the `opc` user, which is used to connect to compute nodes in the cluster.
- `-oh` specifies a compute node and Oracle home directory for which you want to list the available patches. In this context, an Oracle home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.

For example:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -list_patches
-sshkey=/home/opc/.ssh/id_rsa
-oh=hostname1:/u01/app/oracle/product/12.1.0.2/dbhome_1
```

---

**Note:** The list of available patches is determined by interrogating the database to establish the patches that have already been applied. When a patch is applied, the corresponding database entry is made as part of the SQL patching operation, which is executed at the end of the patch workflow. Therefore, the list of available patches may include partially applied patches along with patches that are currently being applied.

---

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 by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to Check Prerequisites Before Applying a Patch](#) at the end of this topic.

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 the `opatch prereq` command to validate that 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 Machine 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 Machine](#).

#### 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. Click the database deployment on which you want to check patching.  
The Oracle Database Cloud Service Overview page is displayed.
3. 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.

4. In the entry for the patch whose prerequisites you want to check, click the  menu and then select **Precheck**.

If you have previously checked prerequisites for the selected patch, the Patch Precheck Service window displays, showing the results of the previous check and asking you to perform another set of prerequisite checks. In this case, click **Precheck** to continue.

The Patching page redisplay, showing a status message indicating that prerequisite checks are in progress.

5. Refresh the Patching page occasionally to update the status message.  
Note that prerequisite checking can take several minutes to complete.
6. 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

- 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 the prerequisites-checking 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 `-precheck_async` action:

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

where:

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

---

**Note:** For details about how to find the available patch identifiers, see [Listing Available Patches](#).

---

- `-sshkey` specifies the location of the SSH private key of the `opc` user, which is used to connect to compute nodes in the cluster.
- `-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.

For example:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -precheck_async 12345678
-sshkey=/home/opc/.ssh/id_rsa
-instance1=hostname1:/u01/app/12.1.0.2/grid,/u01/app/oracle/product/12.1.0.2/
dbhome_1
```

4. Exit the root-user command shell:

```
# exit
$
```

## Applying a Patch

You can apply a patch to a database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to Apply a Patch](#) at the end of this topic.

### 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 Machine 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 Machine](#).

#### 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. Click the database deployment to which you want to apply a patch.  
The Oracle Database Cloud Service Overview page is displayed.
3. 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.
4. In the entry for the patch you want to apply, click the  menu and then select **Patch**.  
The Patch Service window displays.
5. Optionally, 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.

---

---

### Other Ways to Apply a Patch

- 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.
- 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:

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

where:

- `patchid` identifies the patch to be applied.



---

**Note:** For details about how to find the available patch identifiers, see [Listing Available Patches](#).

---

- `-sshkey` specifies the location of the SSH private key of the `opc` user, which is used to connect to compute nodes in the cluster.
- `-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.
- `-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. Therefore, take care not to specify this argument if you are patching a subset of nodes and further nodes remain to be patched.

---



---

**Note:** This argument can only be specified in conjunction with a patching operation on a set of compute nodes. Therefore, 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. Refer to the patch documentation for further details.

---

For example:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -apply_async 23456789
-sshkey=/home/opc/.ssh/id_rsa
-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
```

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 by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to Roll Back a Patch or Failed Patch](#) at the end of this topic.

### 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. 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 database deployment on which you want to roll back a patch.

The Oracle Database Cloud Service Overview page is displayed.

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

The Oracle Database Cloud Service Patching page is displayed.

4. Click **Rollback**.

The Patching page redisplay, 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

- 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 rollback operation:

- Can be used to roll back a patch on some or all of your compute nodes using one command.
- Coordinates multi-node operations in a rolling manner.
- Can execute rollback-related SQL after rolling back the patch on all the compute nodes in the cluster.

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

1. Connect to the compute node as the `opc` user.

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:

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

where:

- `patchid` identifies the patch to be rolled back.
- `-sshkey` specifies the location of the SSH private key of the `opc` user, which is used to connect to compute nodes in the cluster.
- `-instanceN` specifies a compute node and one or more Oracle home directories that are subject to the rollback operation. In this context, an Oracle home directory may be an Oracle Database home directory or the Oracle Grid Infrastructure home directory.
- `-run_datasql=1` instructs the `exadbcpatchmulti` command to execute rollback-related SQL commands.

---

**Note:** Rollback-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.

---

---

---

**Note:** This argument can only be specified in conjunction with a rollback operation on a set of compute nodes. Therefore, if you have rolled back all of your nodes and you did not specify this argument, you will need to manually execute the SQL commands associated with the rollback operation. Refer to the patch documentation for further details.

---

---

For example:

```
# /var/opt/oracle/exapatch/exadbcpatchmulti -rollback_async 34567890
-sshkey=/home/opc/.ssh/id_rsa
-instance1=hostname1:/u01/app/12.1.0.2/grid
-instance2=hostname2:/u01/app/12.1.0.2/grid
-run_datasql=1
```

4. Exit the root-user command shell:

```
# exit
$
```

---

# Configuring Database Features, Database Options, and Companion Products

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

## Topics

- [Using Tablespace Encryption in Exadata Cloud Machine](#)
- [Creating and Activating a Master Encryption Key for a PDB](#)
- [Using Oracle GoldenGate Cloud Service with Exadata Cloud Machine](#)
- [Managing Huge Pages](#)

## Using Tablespace Encryption in Exadata Cloud Machine

All new tablespaces that you create in an Exadata Cloud Machine database are encrypted by default.

However, the tablespaces that are created in conjunction with the database deployment may not be encrypted by default:

- For Oracle Database 11g databases, none of the tablespaces created in conjunction with the database deployment are encrypted.
- For Oracle Database 12c Release 1 databases (12.1.0.2), none of the tablespaces created in conjunction with the database deployment are encrypted. This includes the tablespaces in the root container (CDB\$ROOT), the seed pluggable database (PDB\$SEED), and the first user-created pluggable database.
- For Oracle Database 12c Release 2 databases (12.2.0.1), only the USERS tablespaces 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), and the first user-created pluggable database.

## Topics

- [Creating Encrypted Tablespaces](#)
- [Managing Tablespace Encryption](#)

## Creating Encrypted Tablespaces

User-created tablespaces are encrypted by default.

By default, any new tablespaces you create by using the SQL `CREATE TABLESPACE` command, or any tool executing the `CREATE TABLESPACE` command, will be 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 on the `CREATE TABLESPACE` command. Supported algorithms for Oracle Database 11g and Oracle Database 12c are AES256, AES192, AES128, and 3DES168.

## Managing Tablespace Encryption

You can manage the software keystore (known as an Oracle wallet in Oracle Database 11g), 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 Machine, 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 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 Machine databases, this parameter is set to `CLOUD_ONLY`.

Values of this parameter are as follows.

---

Value	Description
ALWAYS	Any tablespace created will be transparently encrypted with the AES128 algorithm unless a different algorithm is specified on the <code>ENCRYPTION</code> clause.

---

Value	Description
CLOUD_ONLY	Tablespaces created in a Database Cloud Service database will be transparently encrypted with the AES128 algorithm unless a different algorithm is specified on the ENCRYPTION clause. For non-Database Cloud Service databases, tablespaces will only be encrypted if the ENCRYPTION clause is specified. This is the default value.
DDL	Tablespaces are not transparently encrypted and are only encrypted if the ENCRYPTION clause is specified.

## Creating and Activating a Master Encryption Key for a PDB

You must create and activate a master encryption key for any PDBs you create.

After creating or plugging in a new 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. Set the container to the PDB.

```
ALTER SESSION SET CONTAINER = pdb;
```

2. Query V\$ENCRYPTION\_WALLET as follows:

```
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. In the root container, query V\$ENCRYPTION\_WALLET and take note of the current value of the WALLET\_TYPE column:

```
SELECT wallet_type FROM v$encryption_wallet;
```

2. Close the keystore. How you close the keystore depends on the current wallet type observed in the previous step.

- If the current wallet type is AUTOLOGIN, use the following command:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE close;
```

- If the current wallet type is PASSWORD, use the following command:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE close IDENTIFIED BY keystore-password;
```

3. Reopen the keystore by executing the following command:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE open IDENTIFIED BY keystore-password  
CONTAINER=all;
```

Specifying CONTAINER=all opens the keystore in the root and in all PDBs. See "ADMINISTER KEY MANAGEMENT" in *Oracle Database SQL Language Reference* for Release [12.2](#) or [12.1](#).

If the command generates an ORA-28354 error, see [TDE Wallet Problem in 12c: Cannot do a Set Key operation when an auto-login wallet is present](#).

4. Set the container to the PDB.

```
ALTER SESSION SET CONTAINER = pdb;
```

5. Create and activate a master encryption key in the PDB by executing the following command:

```
ADMINISTER KEY MANAGEMENT SET KEY USING TAG 'tag' IDENTIFIED BY keystore-password
WITH BACKUP USING 'backup_identifier';
```

You can use the optional USING TAG clause to associate a tag with the new master encryption key. Specify the WITH BACKUP clause, and optionally the USING 'backup\_identifier' clause, to create a backup of the keystore before the new master encryption key is created. See "ADMINISTER KEY MANAGEMENT" in *Oracle Database SQL Language Reference* for Release 12.2 or 12.1.

If the command generates an ORA-28354 error, see [TDE Wallet Problem in 12c: Cannot do a Set Key operation when an auto-login wallet is present](#).

6. Query V\$ENCRYPTION\_WALLET again to verify that the STATUS column is set to OPEN.

```
SELECT wrl_parameter, status, wallet_type FROM v$encryption_wallet;
```

## Using Oracle GoldenGate Cloud Service with Exadata Cloud Machine

Oracle GoldenGate Cloud Service is a high performance, secure public cloud data integration and replication solution that can replicate data in real time from on-premises databases to databases in Oracle Database Exadata Cloud Machine.

You must create an Exadata Cloud Machine 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 Machine database deployment for use as a replication target:

- You must configure the database deployment to use characteristics (like service level, database release, database edition and so on) that are supported by Oracle GoldenGate Cloud Service.

See *Before You Begin with Oracle GoldenGate Cloud Service* in *Using Oracle GoldenGate Cloud Service* for information about the Exadata Cloud Machine characteristics that Oracle GoldenGate Cloud Service supports.

- 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 Service Details page of the Create Service wizard.

- The target database must be network accessible on the listener port.

For Exadata Cloud Machine you can:

- Specifically enable network access to the Oracle Net Listener port. See [Enabling Access to a Compute Node Port](#). If you specifically enable access to



the Oracle Net Listener port, ensure that you always use an encrypted Oracle Net connection. See [Using Network Encryption and Integrity](#).

Once you have created and properly configured an Exadata Cloud Machine 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*.

## Managing Huge Pages

Huge Pages provide considerable performance benefits for Oracle Database on systems with large amounts of memory. Oracle Database Exadata Cloud Machine 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 Machine environments, a standard page is 4 KB, while a Huge Page is 2 MB by default. Therefore, an Oracle Database on Exadata Cloud Machine 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.

### Default Configuration of Huge Pages

By default, 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 Machine 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 Machine 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](#).

---

# Migrating Oracle Databases to Exadata Cloud Machine

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

## Topics

- [Choosing a Migration Method](#)
- [Migration Methods](#)

## Choosing a Migration Method

Various different migration methods exist, and each migration method is associated with different benefits, opportunities, requirements and limitations.

### Migration Considerations

Many of migration methods apply only if specific characteristics of the source (on-premises) and target (Exadata Cloud Machine) databases match or are compatible. Moreover, additional factors can affect which method you choose for your migration from among the methods that are technically applicable to your migration scenario.

For example, Exadata Cloud Machine 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 use of specific database features, such as materialized views or object data types, may impose restrictions on some migration methods.

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 or compromised in some way. 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 Machine. However, the physical organization of the data in the Transportable Tablespaces may not be ideal for Database Machine, 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 Machine. For smaller data sets you can transfer the data across a network link between your source system and Exadata Cloud Machine. However, for larger data sets this is not feasible because it would simply take too long. To accommodate these situations, you can use Oracle Public Cloud Data Transfer Services to physically send large data sets to Oracle Public Cloud. When you engage Oracle Public Cloud Data Transfer Services, Oracle ships a ZFS storage array to your data center. After your data is loading on to the storage array and it is shipped back, Oracle transfers the data to an Oracle Storage Cloud Service container that is accessible from your Exadata Cloud Machine environment. See [Loading Data into the Oracle Database on Exadata Cloud Machine](#).

Finally, Oracle can offer professional services to assist with all aspects of data migration to Exadata Cloud Machine. 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. Database version of your source database:
  - Oracle Database 11g Release 2 version lower than 11.2.0.3
  - Oracle Database 11g Release 2 version 11.2.0.3 or higher
  - Oracle Database 12c Release 1 version lower than 12.1.0.2

- Oracle Database 12c Release 1 version 12.1.0.2 or higher
2. For Oracle Database 12c Release 1 source databases, the architecture of the database:
    - 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 Machine 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 Machine.
  5. The target database version that you are migrating to on Exadata Cloud Machine:
    - Oracle Database 11g Release 2
    - Oracle Database 12c Release 1

With Exadata Cloud Machine, Oracle Database 12c Release 1 databases are configured to use the CDB architecture.

After gathering this information, use the following table as a guide to see which migration methods might apply to your migration scenario. Then consider the information about applicable migration methods 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 Method	11g source to 11g target	11g source to 12c target	12c CDB source to 12c target	12c Non-CDB source to 12c target	Source Platform	Source Database Character Set
<a href="#">Conventional RMAN Backup and Recovery</a>	Yes		Yes		Linux x86-64	Any, but matching the target database character set or using a compatible subset is recommended.
<a href="#">Conventional Data Pump Export and Import</a>	Yes	Yes	Yes	Yes	Any	Any
<a href="#">Transportable Tablespaces</a>	Yes	Yes	Yes	Yes	Most	Must match target database character set or be a compatible subset.
<a href="#">Data Pump Full Transportable Export and Import</a>		Yes	Yes	Yes	Most	Must match target database character set or be a compatible subset.
<a href="#">Transportable Tablespaces with Cross Platform Incremental Backup</a>	Yes	Yes	Yes	Yes	Most	Must match target database character set or be a compatible subset.

Migration Method	11g source to 11g target	11g source to 12c target	12c CDB source to 12c target	12c Non-CDB source to 12c target	Source Platform	Source Database Character Set
<a href="#">Transportable Database</a>	Yes		Yes		Little-endian	Any, but matching the target database character set or using a compatible subset is recommended.
<a href="#">Data Guard Physical Standby</a>	Yes		Yes		Linux x86-64 and compatible little-endian platforms	Any, but matching the target database character set or using a compatible subset is recommended.
<a href="#">Unplugging and Plugging a PDB</a>			Yes		Little-endian	Must match target database character set or be a compatible subset.
<a href="#">Plugging in a Non-CDB</a>				Yes	Little-endian	Must match target database character set or be a compatible subset.

Migration Method	11g source to 11g target	11g source to 12c target	12c CDB source to 12c target	12c Non-CDB source to 12c target	Source Platform	Source Database Character Set
<a href="#">Cloning a Remote PDB or Non-CDB</a>			Yes	Yes	Little-endian	Must match target database character set or be a compatible subset.

## Migration Methods

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

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](#)
- [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 Machine 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 Machine simply by transferring a backup of your source database to Exadata Cloud Machine



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 Machine), and it uses Oracle Database 11g Release 2 or Oracle Database 12c Release 1, you can use RMAN to restore a backup of your source database on Exadata Cloud Machine.

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](#).

---

---

**Note:**

RMAN provides other options if your source database platform differs from Exadata Cloud Machine:

- If your source database resides on another little-endian platform, you can use RMAN to transport the entire database to Exadata Cloud Machine. 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](#).

---

---

## 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, it's 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-organise 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 databases 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 Machine 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 Machine compute node.
3. On the Exadata Cloud Machine 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 [Part I: Oracle Data Pump](#) in *Oracle Database Utilities*.

### 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 Machine.

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 Machine compute node.

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

- a. On the target Exadata Cloud Machine 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 Machine compute node is available on your source host. For more information about SSH keys, see [About Network Access to Exadata Cloud Machine](#).

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

- a. On the Exadata Cloud Machine 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 Machine 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 Machine 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 Machine (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](#).

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

### 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 Machine.

This example performs a migration of the FSDATA and FSINDEX tablespaces, which contain objects owner 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 the 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 Machine compute node.

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

- a. On the target Exadata Cloud Machine 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 Machine compute node is available on your source host. For more information about SSH keys, see [About Network Access to Exadata Cloud Machine](#).

- c. On the source database host, use the SCP utility to transfer the dump file and tablespace data files to the target Exadata Cloud Machine 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 Machine 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 conversion 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 Machine compute node, use Data Pump Import to load the metadata associated with the tablespace set.

- a. Invoke SQL\*Plus and log in to the target database as the SYSTEM user.
- b. Create a directory object in the target database that points to the operating system directory containing the Data Pump dump file.

```

SQL> CREATE DIRECTORY dp_from_source AS '/u01/app/oracle/admin/ORCL/dpdump/
from_source';

```

- c. If they do not already exist, create user accounts for the owners of the objects that will be imported into the target database.

```

SQL> CREATE USER fsowner
2 PROFILE default
3 IDENTIFIED BY fspass
4 TEMPORARY TABLESPACE temp
5 ACCOUNT UNLOCK;

```

- d. Invoke Data Pump Import and import the tablespace metadata into the target database. Use the TRANSPORT\_DATAFILES option and specify the file names for the data files that are converted and loaded into ASM.

```

$ impdp system DIRECTORY=dp_from_source \
TRANSPORT_DATAFILES='+DATA_SYSNAME/ORCL/datafile/fsdata01.277.821069105', \
'+DATA_SYSNAME/ORCL/datafile/fsindex01.278.419052810'

```

7. On the target database, set the FSDATA and FSINDEX tablespaces to READ WRITE mode.

```

SQL> ALTER TABLESPACE fsdata READ WRITE;
Tablespace altered.
SQL> ALTER TABLESPACE fsindex READ WRITE;
Tablespace altered.

```

8. After verifying that the data has been imported successfully, you can delete the expdat.dmp dump file.

## Data Pump Full Transportable Export and Import

Like transportable tablespaces, this method provides broad cross-platform migration support, limited support for source and destination databases with different character sets, and it can be used to migrate data to a later version of Oracle Database. It

simplifies the process of migrating complete databases and leverages the transportable tablespace feature where possible.

Data Pump full transportable export and import is an extension of basic transportable tablespaces, which can be used to migrate the entire contents of your source database to Exadata Cloud Machine.

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](#).

To migrate your source database to Exadata Cloud Machine 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 Machine compute node.
4. On the Exadata Cloud Machine 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 Machine 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 Machine 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 is accomplished in three phases:

**1. Preparation.**

- a.** Use RMAN to backup your source tablespaces.
- b.** Transfer the backups to an Exadata Cloud Machine compute node.
- c.** On the Exadata Cloud Machine 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 Machine compute node.
- c.** On the Exadata Cloud Machine 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 Machine compute node.
- e.** On the Exadata Cloud Machine 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 Machine compute node, use Data Pump Import to load the metadata associated with the tablespace set.

- g.** Set the tablespaces on the Exadata Cloud Machine 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 option see [Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup](#).

## 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 is 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 Machine.

When you use the transportable database method, the result is a block-for-block copy of the source database, and consequently 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 is suitable for use in conjunction with Exadata Cloud Machine 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 Machine.
  - The target system. You might select this option to minimize any migration impact on the source system.

See [Transporting Data Across Platforms](#).

## 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 Machine. Afterwards 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 Machine, your source database must reside on Linux x86–64 (the same as Exadata Cloud Machine) 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 Machine 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 consequently 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 Machine 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 Machine.

You must create an Exadata Cloud Machine deployment that will eventually incorporate your migrated database.

2. Manually delete the Exadata Cloud Machine database.

When you create a database deployment on Exadata Cloud Machine, a default database is also created. This database cannot be used as a Data Guard standby

database and must be manually deleted. You can manually delete the database by using the Database Configuration Assistant (DBCA) and specifying the `deleteDatabase` option.

**3. Configure the network.**

You must configure a secure and reliable network link between your source database environment and Exadata Cloud Machine. Steps include:

- a.** Configure access to Exadata Cloud Machine. This may include enabling access to the Oracle Net listener port, or configuring an IPSec VPN. See [Managing Network Access to Exadata Cloud Machine](#)
- b.** If you are not using IPSec VPN to encrypt all network traffic between your network and Exadata Cloud Machine, configure Oracle Net encryption and integrity. By default, Oracle Net encryption and integrity is configured on Exadata Cloud Machine.

The easiest way to configure your source environment is to copy the `SQLNET.ENCRYPTION` and `SQLNET.CRYPTO` parameter settings from the `sqlnet.ora` file on an Exadata Cloud Machine compute node. The `sqlnet.ora` file is located under `$ORACLE_HOME/network/admin/sqlnet.ora`, where `ORACLE_HOME` is typically `/u01/app/12.1.0.2/grid` or `/u01/app/12.2.0.1/grid`, depending on which Oracle Grid Infrastructure version is in use.

- c.** Configure your source network. This may include:
  - i.** Configure a naming service to resolve Exadata Cloud Machine addresses from your source network.
  - ii.** Configure prompt-less SSH connectivity in both directions between your source database environment and Exadata Cloud Machine.
  - iii.** Configure your source network firewall to allow network connectivity from Exadata Cloud Machine.

**4. Create the Data Guard physical standby database.**

To create a Data Guard physical standby database you must:

- a.** Prepare the source database.

Tasks include:

- Enable forced logging.
- Create standby log files.
- Configure redo transport authentication.
- Configure the primary database to receive redo data.
- Set the required primary database initialization parameters.
- Enable archiving

- b.** Duplicate the source database on to the standby system.

To duplicate the source database, you have two options:

- You can use RMAN to perform active duplication. In this case, RMAN uses a network link to copy the data files directly from the primary database to the standby database. 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.
  - Backup-based duplication enables you to transfer a backup of the source database by any means available. The backup can be an RMAN backup set or it can be manually created. You can also use backups stored in Oracle Database Backup Cloud Service as the source for backup-based duplication.
- c. Complete configuration and start the Data Guard redo apply process.
- Tasks include:
- Configure redo transport authentication for the standby database.
  - Create Oracle Net service names for both databases.
  - Copy the primary database encryption wallet to the standby database system if required.
  - Set the required standby database initialization parameters.
  - Start the standby database.
  - Start Data Guard redo apply process.
  - Optionally, configure Data Guard broker.

See [Creating a Physical Standby Database](#) and [Creating a Standby Database with Recovery Manager](#) in *Oracle Data Guard Concepts and Administration*.

5. Optionally, enable redo transport compression.

Redo transport compression may be required if the uncompressed redo volume exceeds the available network bandwidth.

6. Perform a switchover.

After the primary and standby databases are fully synchronized, you can perform a switchover to swap the role of each database. After the switchover, the original standby database, hosted in Exadata Cloud Machine, becomes the primary database. At this point you can direct all of your application traffic to Exadata Cloud Machine. The original source database now assumes the standby database role.

7. Decouple the databases.

After Exadata Cloud Machine becomes the host of your primary database, you can decouple the databases by stopping the Data Guard redo apply services and removing the initialization parameter settings for Data Guard. Finally, you can decommission your original source database.

## 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 to enable easy migration of pluggable databases (PDBs).

You can migrate an Oracle Database 12c PDB to Oracle Database Exadata Cloud Machine by unplugging the PDB from the source container database (CDB) and plugging it into a CDB on Exadata Cloud Machine.

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 Machine.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud Machine. Alternatively, the PDB character set must be 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 an Oracle Database 12c PDB to Exadata Cloud Machine 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 Machine compute node.
3. On the Exadata Cloud Machine 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 Machine.
  - 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*.

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*.

## 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 and provides a way to consolidate several non-CDBs into a multitenant database on Exadata Cloud Machine.

You can migrate an Oracle Database 12c non-CDB to Oracle Database Exadata Cloud Machine by plugging the non-CDB into a CDB on Exadata Cloud Machine. This method is similar to unplugging and plugging a PDB, and has similar requirements and restrictions:

- The source database must be version 12.1 or later.
- The source and target platform must have the same endian format, which is little-endian for Exadata Cloud Machine.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud Machine. Alternatively, the PDB character set must be 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 an Oracle Database 12c non-CDB to Exadata Cloud Machine 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 Machine compute node.
3. On the Exadata Cloud Machine 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*.

## 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 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 12.1 or later.
- The source and target platform must have the same endian format, which is little-endian for Exadata Cloud Machine.
- Ideally, the source and target CDBs must use the same character set, which is AL32UTF8 for Exadata Cloud Machine. Alternatively, the PDB character set must be 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 an Oracle Database 12c PDB or Non-CDB to Exadata Cloud Machine 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*.



---

## Frequently Asked Questions for Exadata Cloud Machine

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

- [Who is the service right for?](#)
- [Does the Exadata Cloud Machine support external Oracle Net \(SQL\\*Net\) connections?](#)
- [How is storage allocated?](#)
- [How are users defined?](#)
- [How can I secure my data?](#)
- [Can I load additional third-party software?](#)
- [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 Machine 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 Machine support external Oracle Net (SQL\*Net) connections?**

Yes. Exadata Cloud Machine supports direct external connections using Oracle Net. See [Connecting Remotely to the Database by Using Oracle Net Services](#).

### **How is storage allocated?**

The amount of storage space allocated to Exadata Cloud Machine 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 Machine 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 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*.

### **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 Machine 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).

---

---

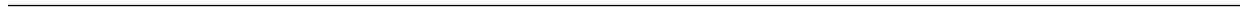
**Note:** Some options are dependant on the Oracle Database version in use. For example, Oracle Database In-Memory can only be used with Oracle Database software version 12.1.0.2, or later.

---

---

**Is this service enabled to use Application Express?**

No, by default Oracle Application Express is not enabled on Exadata Cloud Machine deployments. However, you may manually customize your databases to configure and enable Oracle Application Express.



---

## 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 Machine.

### Topics

- [Linux User Accounts](#)
- [Locations of Installed Software](#)
- [Network Access](#)
- [Oracle Database Characteristics](#)
- [Location of Diagnostic and Log Files](#)

### Linux User Accounts

This section provides information about Linux user accounts that are provisioned on Oracle Database Exadata Cloud Machine.

Every Exadata Cloud Machine compute node is provisioned with the following operating system user accounts.

User	Description
<code>opc</code>	The system administrator account you use in conjunction with the <code>sudo</code> command to gain <code>root</code> user access to your compute nodes.
<code>oracle</code>	The Oracle Database administrator account you use to access the system and perform database administration tasks. A home directory, <code>/home/oracle</code> , is created for this user. This user cannot use the <code>sudo</code> command to perform operations that require <code>root</code> user access.
<code>root</code>	The root administrator for the system. You do not have direct access to this account. To perform operations that require <code>root</code> user access, execute <code>sudo -s</code> as the <code>opc</code> user.
<code>grid</code>	The Oracle Grid Infrastructure administrator account you use to perform ASM and clusterware administration tasks. A home directory, <code>/home/grid</code> , is created for this user. This user cannot use the <code>sudo</code> command to perform operations that require <code>root</code> user access. You do not have direct access to this account. To perform operations that require <code>grid</code> user access, execute <code>sudo -s</code> as the <code>opc</code> user to get <code>root</code> access, and then execute <code>su - grid</code> to become the <code>grid</code> 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 <code>/home/opc</code> , <code>/home/oracle</code> or <code>/home/grid</code> .
HOSTNAME	The host name of the compute node.
LANG	The system language, <code>en_US.UTF-8</code> .
PATH	The paths to search for executables; set to include: <ul style="list-style-type: none"><li>• <code>/sbin</code></li><li>• <code>/usr/sbin</code></li><li>• <code>/bin</code></li><li>• <code>/usr/bin</code></li><li>• <code>\$HOME</code></li></ul>
SHELL	The default shell, <code>/bin/bash</code> .
USER	The user name, either <code>opc</code> , <code>oracle</code> or <code>grid</code> .

---

In addition, the following environment variable settings are created for the `grid` user only.

---

Variable	Description
ORACLE_HOME	The Oracle Grid Infrastructure home directory: <code>/u01/app/12.1.0.2/grid</code> or <code>/u01/app/12.2.0.1/grid</code>
ORACLE_SID	The ASM system identifier (SID) associated with the ASM instance on the compute node: <code>+ASM<math>N</math></code> , where $N$ is a unique number (1, 2, 3, and so on).
PATH	Additional paths to search for executables: <ul style="list-style-type: none"><li>• <code>\$ORACLE_HOME/bin</code></li><li>• <code>\$ORACLE_HOME/OPatch</code></li></ul>

---

## Locations of Installed Software

This section provides information about the locations of installed software on a newly created Oracle Database Exadata Cloud Machine database deployment.

When a database deployment is created on Exadata Cloud Machine, software is installed in the following locations.

Software	Installation Location
Oracle Database	<p>\$ORACLE_HOME:</p> <ul style="list-style-type: none"> <li>Oracle Database 12c Release 2: The binaries associated with the first database deployment are located at /u01/app/oracle/product/12.2.0.1/dbhome_1. Binaries associated with subsequent deployments are located at /u02/app/oracle/product/12.2.0/dbhome_N, where N is a unique number (1, 2, 3, and so on).</li> <li>Oracle Database 12c Release 1: The binaries associated with the first database deployment are located at /u01/app/oracle/product/12.1.0.2/dbhome_1. Binaries associated with subsequent deployments are located at /u02/app/oracle/product/12.1.0/dbhome_N, where N is a unique number (1, 2, 3, and so on).</li> <li>Oracle Database 11g Release 2: The binaries associated with the first deployment are located at /u01/app/oracle/product/11.2.0.4/dbhome_1. Binaries associated with subsequent deployments are located at /u02/app/oracle/product/11.2.0/dbhome_N, where N is a unique number (1, 2, 3, and so on).</li> <li>Oracle Grid Infrastructure: /u01/app/12.1.0.2/grid or /u01/app/12.2.0.1/grid</li> </ul>

## Network Access

This section provides information about network access to Oracle Database Exadata Cloud Machine.

With Exadata Cloud Machine, compute node network access is limited to Secure Shell (SSH) connections on port 22 by default. This access restriction ensures that the environment is secure by default. To access other ports, you can enable network access to the port or create an SSH tunnel. For more information, see:

- [Enabling Access to a Compute Node Port](#)
- [Creating an SSH Tunnel to a Compute Node Port](#)

## Oracle Database Characteristics

When a database deployment is created on Oracle Database Exadata Cloud Machine, an Oracle database is created using information provided in the Create Service wizard:

Wizard Page and Field	How Used When Creating the Database
Software Release on the Service page	Determines which version of Oracle Database is used, 12c Release 2, 12c Release 1 or 11g Release 2.
DB Name (SID) on the Service Details page	The database system identifier (SID) of the database.
PDB Name on the Service Details page (for Oracle Database 12c only)	The name of the default pluggable database (PDB) created in the container database.

Wizard Page and Field	How Used When Creating the Database
Administrator Password on the Service Details page	The password used for the SYS and SYSTEM database users.
Application Type on the Service Details page (for the first database deployment only)	Adjusts Oracle Database parameter settings: <ul style="list-style-type: none"><li>• <b>Transactional (OLTP)</b> — configures the database for a transactional workload, with a bias towards high volumes of random data access.</li><li>• <b>Decision Support or Data Warehouse</b> — configures the database for a decision support or data warehouse workload, with a bias towards large data scanning operations.</li></ul>
Character Set on the Service Details page	The database character set.
National Character Set on the Service Details page	The database national character set.

---

## Location of Diagnostic and Log Files

When a database deployment is created on Oracle Database Exadata Cloud Machine, 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 `/u01/app/oracle/diag`.

Oracle Grid Infrastructure trace files and log files are stored in subdirectories of `/u01/app/grid/diag`.



---

## Oracle Cloud Pages for Administering Exadata Cloud Machine

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 Machine.

### Topics

- [Services Page](#)
- [Activity Page](#)
- [SSH Access Page](#)
- [Overview Page](#)
- [Backup Page](#)
- [Patching Page](#)
- [Create Service: Service Page](#)
- [Create Service: Service Details Page](#)
- [Create Service: Confirmation Page](#)

### Services Page

The Oracle Database Cloud Service Services page displays all deployments on Oracle Database Exadata Cloud Machine.

### Topics

- [What You Can Do From the Oracle Database Cloud Service Services Page](#)
- [What You See on the Oracle Database Cloud Service Services Page](#)






### What You Can Do From the Oracle Database Cloud Service Services Page


Use the Oracle Database Cloud Service Services 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 Services Page

The following table describes the key information shown on the Oracle Database Cloud Service Services page.

Element	Description
 navigation menu	Navigation menu providing access other Oracle Cloud services in the identity domain.
	User menu providing access to help, accessibility options, console version information and sign-out.
 menu next to "Oracle Database Cloud Service"	Menu that provides access to Platform Services.
<b>Activity</b>	Click to go to the <a href="#">Activity Page</a> .
<b>SSH Access</b>	Click to go to the <a href="#">SSH Access Page</a> .
<b>Welcome!</b>	Click to go to the Oracle Database Cloud Service console Welcome page.
<b>REST APIs</b>	Click to go to the API Catalog Cloud Service.
<b>Services, OCPUs, Memory, Storage and Public IPs</b>	Summary of resources being used: <ul style="list-style-type: none"> <li>• <b>Services</b> — Total number of configured deployments.</li> <li>• <b>OCPUs</b> — Total number of Oracle CPUs allocated across all deployments.</li> <li>• <b>Memory</b> — Total amount of compute node memory allocated across all deployments.</li> <li>• <b>Storage</b> — Total amount of storage allocated across all deployments.</li> <li>• <b>Public IPs</b> — Number of public IP addresses allocated across all deployments.</li> </ul>
<input type="text" value="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.
<b>Create Service</b>	Click to create a new database deployment on Exadata Cloud Machine. See <a href="#">Creating a Database Deployment</a> .
	Click to view details for the database deployment or clone deployment.
<b>Status</b>	Status of the deployment if it is not running. Status values include "In Progress", "Maintenance", "Stopped", and "Terminating".
<b>Version</b>	Version of Oracle Database configured on the deployment. For example: 12.1.0.2 or 11.2.0.4.
<b>Edition</b>	Software edition of Oracle Database configured on the deployment.

Element	Description
<b>Created On or Submitted On</b>	Date when the deployment was created. During the creation process, the date when the creation request was submitted.
<b>Exadata System</b>	Name of the Exadata Cloud Machine instance.
<b>OCPUs</b>	Number of Oracle CPUs associated with the deployment.
<b>Memory</b>	Amount of compute node memory in GBs associated with the deployment.
<b>Storage</b>	Amount of storage in GBs associated with the deployment.
 menu for each deployment	<p>Menu that provides the following options:</p> <ul style="list-style-type: none"> <li>• <b>Open EM Console</b> — Open the database console, either Enterprise Manager Database Express 12c or Enterprise Manager 11g Database Control.</li> <li>• <b>SSH Access</b> — Add an SSH public key. See <a href="#">Adding an SSH Public Key</a>.</li> <li>• <b>Update Exadata IORM</b> — Update settings for Exadata I/O resource management (IORM). See <a href="#">Using Exadata I/O Resource Management</a>.</li> <li>• <b>Delete</b> — Delete the deployment. See <a href="#">Deleting a Database Deployment</a>.</li> </ul>
<b>Service create and delete history</b>	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 Machine 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 Machine 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: <ul style="list-style-type: none"> <li>• All</li> <li>• Scheduled</li> <li>• Running</li> <li>• Succeeded</li> <li>• Failed</li> </ul> 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: <i>service_instance:identity_domain</i> 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.

Element	Description
Operation	Shows the type of operation performed on the service instance. You can sort the column in ascending or descending order.
Status	Shows the status of the operation performed on the service instance. You can sort the column in ascending or descending order.
Start Time	Shows the time the operation started. You can sort the column in ascending or descending order.
End Time	Shows the time the operation ended, if the operation is complete. You can sort the column in ascending or descending order.
Initiated By	Shows the user that initiated the operation. The user can be any user in the identity domain who initiated the operation or, for certain operations such as automated backup, System. You can sort the column in ascending or descending order.

## SSH Access Page

The SSH Access page enables you to view and add SSH public keys to Oracle Database Exadata Cloud Machine 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 Machine 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.

Element	Description
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.
<b>Search</b>	Searches for SSH keys by applying the filters specified by the Service Name and Service Type fields, and displays the results in the table.
Results per page	Specifies the number of results you want to view per page. The default value is 10.
	Displays a description of an item in the results table. Clicking on the resulting downward arrow hides the description.
Service Name	Shows the name of the deployment.
Service Type	Shows the type of cloud service for this deployment.
Last Update	Shows the most recent time the SSH keys for this deployment were updated.
Actions	<p>Click the <b>Add New Key</b> button to add a new SSH public key to this deployment.</p> <p>The <b>Add New Key</b> overlay is displayed with its <b>Key value</b> field displaying the deployment's most recent SSH public key.</p> <p>Specify the new public key using one of the following methods:</p> <ul style="list-style-type: none"> <li>• Select <b>Upload a new SSH Public Key value</b> and click <b>Choose File</b> to select a file that contains the public key.</li> <li>• Select <b>Key value</b>. 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.</li> </ul>

## Overview Page



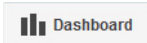

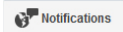


The Oracle Database Cloud Service Overview page displays overview information for an Oracle Database Exadata Cloud Machine database deployment.

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.
 ▼	User menu providing access to help, accessibility options, console version information and sign-out.
	Click to go to the My Services Dashboard page.
	Click to go to the My Services Users page.
	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.
<b>Oracle Database Cloud Service</b> link	Click to return to the <a href="#">Services Page</a> .
 (next to the deployment’s name)	Deployment menu that provides the following options: <ul style="list-style-type: none"> <li>• <b>Open EM Console</b> — Open the database console for the deployment, either Enterprise Manager Database Express 12c or Enterprise Manager 11g Database Control.</li> <li>• <b>SSH Access</b> — Add an SSH public key to the deployment. See <a href="#">Adding an SSH Public Key</a>.</li> <li>• <b>View Activity</b> — Go to the <a href="#">Activity Page</a> to view activities performed on this deployment.</li> </ul>



### 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
<b>Overview</b> tile	The current tile, highlighted to indicate that you are viewing the Overview page.
<b>Administration</b> tile	Click to access these pages for the deployment: <ul style="list-style-type: none"> <li>• <a href="#">Backup Page</a></li> <li>• <a href="#">Patching Page</a></li> </ul>

### 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.
<b>Service Overview</b> section	<p>Displays a summary box followed by information about the deployment.</p> <p>The summary box shows high-level information about the Exadata Cloud Service instance hosting the deployment: compute nodes, OCPUs, memory, and storage.</p> <p>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 <b>Show more...</b> link to see even more information about the deployment.</p>
<b>Resources</b> 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.
 (for each compute node)	<p>Compute node menu that provides the following options:</p> <ul style="list-style-type: none"> <li>• <b>Start</b>—Start a stopped compute node. See <a href="#">Stopping, Starting and Restarting Compute Nodes</a></li> <li>• <b>Stop</b>—Stop a compute node. See <a href="#">Stopping, Starting and Restarting Compute Nodes</a></li> <li>• <b>Restart</b>—Restart a compute node. See <a href="#">Stopping, Starting and Restarting Compute Nodes</a></li> </ul>
<b>Network Information</b>	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
<b>Backup Now</b>	Click to create a full backup of the database deployment.
<b>Recover</b>	Click to recover the database deployment to the latest backup or to a specific point in time.
<b>Configure Backups</b>	Click to update the credentials for backing up to cloud storage.




Element	Description
 (for each available backup)	Menu that provides the <b>Recover</b> option. Choose this option to recover to the given backup.
<b>Recovery History</b>	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
<b>Available Patches</b>	A list of patches you can apply to the deployment.
 (for each listed patch)	<b>Menu</b> icon provides the following options for the patch: <ul style="list-style-type: none"> <li><b>Precheck</b> — Check whether the patch can be successfully applied to the deployment.</li> <li><b>Patch</b> — Apply the patch to the deployment.</li> </ul>
<b>Details of Last Patching Activity</b>	Expand to see a description of the actions taken during the last patching operation.
<b>Rollback</b>	Click to roll back the last patching operation. See <a href="#">Rolling Back a Patch or Failed Patch by Using the Oracle Database Cloud Service Console</a> .

## Create Service: Service Page

Create Service: Service 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
<b>Cancel</b>	Click to cancel the Create Service wizard without creating a new database deployment.
<b>Next&gt;</b>	Click to advance to the Create Service: Service Details page.

### What You See in the Page Content Area

The following table describes the key information shown on the Create Service: Service page.

Element	Description
Service Name	The name for the new database deployment. The name: <ul style="list-style-type: none"> <li>• Must not exceed 50 characters.</li> <li>• Must start with a letter.</li> <li>• Must contain only letters, numbers, or hyphens.</li> <li>• Must not contain any other special characters.</li> <li>• Must be unique within the identity domain.</li> </ul>
Description	(Optional) A description for the new database deployment.
Exadata System	This list contains the Oracle Exadata Database Machines that are associated with your existing subscriptions. Exadata Cloud Machine offers several configurations, as described in <a href="#">Exadata System Configuration</a> .
Service Level	The service level for the new deployment: <ul style="list-style-type: none"> <li>• <b>Oracle Database Exadata Cloud Service</b> — is the only Service Level setting compatible with Exadata Cloud Machine. The other service level options relate to Oracle Database Cloud Service, which does not use Exadata.</li> <li>• <b>Oracle Database Cloud Service</b></li> </ul> See <a href="#">Service Level</a> .
Metering Frequency	The metering frequency for the new deployment: <ul style="list-style-type: none"> <li>• <b>Monthly</b></li> </ul> See <a href="#">Metering Frequency</a> .
Software Release	The release version of Oracle Database for the new deployment: <ul style="list-style-type: none"> <li>• <b>Oracle Database 11g Release 2</b></li> <li>• <b>Oracle Database 12c Release 1</b></li> </ul> See <a href="#">Oracle Database Software Release</a> .
Software Edition	The Oracle Database software package for the new deployment: <ul style="list-style-type: none"> <li>• <b>Enterprise Edition - Extreme Performance</b></li> </ul> See <a href="#">Oracle Database Software Edition</a> .
Database Type	The type of deployment to create: <ul style="list-style-type: none"> <li>• <b>Single Instance</b></li> <li>• <b>Database Clustering with RAC</b> — is the only Database Type setting compatible with Exadata Cloud Machine. This setting results in a clustered database that uses Oracle Real Application Clusters, with a clustered database instance on each database server in the Exadata Cloud Machine environment.</li> <li>• <b>Single Instance with Data Guard Standby</b></li> <li>• <b>Database Clustering with RAC and Data Guard Standby</b></li> </ul> See <a href="#">Oracle Database Type</a> .

## Create Service: Service Details Page

Create Service: Service Details is a page in the Create Service 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 Service: Service 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 Navigation Area

Element	Description
<Previous	Click to return to the Create Service: Service page.
Cancel	Click to cancel the Create Service wizard without creating a new database deployment.
Next>	Click to advance to the Create Service: Confirmation page.

### What You See in the Database Configuration Section

Element	Description
DB Name (SID)	The name for the database instance. The name: <ul style="list-style-type: none"> <li>• Must not exceed 8 characters.</li> <li>• Must start with a letter.</li> <li>• Must contain only letters, numbers, or these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).</li> </ul>
PDB Name	(Available only for Oracle Database 12c.) The name for the default pluggable database (PDB). The name: <ul style="list-style-type: none"> <li>• Must not exceed 8 characters.</li> <li>• Must start with a letter.</li> <li>• Must contain only letters, numbers, or these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).</li> </ul>
Administration Password Confirm Password	The password for the Oracle Database administrative users. The password: <ul style="list-style-type: none"> <li>• Must be 8 to 30 characters in length.</li> <li>• Must contain at least one lowercase letter</li> <li>• Must contain at least one uppercase letter</li> <li>• Must contain at least one number</li> <li>• Must contain at least one of these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).</li> </ul>

Element	Description
Application Type	<p>Specifies how the database deployment is configured:</p> <ul style="list-style-type: none"> <li>• <b>Transactional (OLTP)</b> — configures the database for a transactional workload, with a bias towards high volumes of random data access.</li> <li>• <b>Decision Support or Data Warehouse</b> — configures the database for a decision support or data warehouse workload, with a bias towards large data scanning operations.</li> </ul> <hr/> <p><b>Note:</b> 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.</p> <hr/>
SSH Public Key Edit	<p>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.</p> <p>Click <b>Edit</b> 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.</p> <p>If you paste in the value, make sure the value does not contain line breaks or end with a line break.</p> <hr/> <p><b>Note:</b> The SSH Public Key field will not be displayed if the selected Exadata Cloud Machine environment already contains a previously specified SSH key.</p> <hr/>
Advanced Settings: Character Set	<p>The database character set for the database. The database character set is used for:</p> <ul style="list-style-type: none"> <li>• Data stored in SQL CHAR data types (CHAR, VARCHAR2, CLOB, and LONG)</li> <li>• Identifiers such as table names, column names, and PL/SQL variables</li> <li>• Entering and storing SQL and PL/SQL source code</li> </ul>
Advanced Settings: National Character Set	<p>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).</p>

## What You See in the Backup and Recovery Configuration Section

Element	Description
Backup Destination	<p>Controls the destination and configuration of automatic backups:</p> <ul style="list-style-type: none"> <li><b>Remote Storage Only</b> — uses remote NFS storage 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.</li> <li><b>None</b> — no automatic backups are configured.</li> </ul> <hr/> <p><b>Note:</b> Automatic backups cannot be configured later if you select the <b>None</b> option when you create a database deployment.</p> <hr/> <p>For more information about backups and backup configurations, see <a href="#">About Backing Up Database Deployments on Exadata Cloud Machine</a>.</p>
NFS Remote Backup	<p>The path of the NFS remote backup location where backups of the database deployment are to be stored. This path has one of the following formats:</p> <pre>hostname: absolute-path ip-address: absolute-path</pre> <p>This field is only displayed if remote storage is included in your Backup Destination choice.</p>

## Create Service: Confirmation Page

Create Service: Confirmation is the final page in the Create Service wizard you use to create a new database deployment. For more information, see [Creating a Database Deployment](#).

### What You See on the Create Service: Confirmation Page

The Create Service: Confirmation page presents a summary list of all the choices you made on the preceding pages of the Create Service wizard. In addition, it provides the controls described in the following table.

Element	Description
<Previous	Click to return to the Create Service: Details page.
Cancel	Click to cancel the Create Service wizard without creating a new deployment.
Create>	<p>Click to begin the process of creating an Exadata Cloud Machine deployment.</p> <p>The Create Service wizard closes and the Oracle Database Cloud Service console is displayed, showing the new deployment with a status of In progress.</p>

