

Oracle® Public Cloud Machine

Using Oracle Database Cloud Service

Release 17.1.2

E72994-04

June 2017

Copyright © 2016, 2017, 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

Preface	xi
Audience	xi
Documentation Accessibility	xi
Related Documents.....	xi
Conventions.....	xi
1 Getting Started with Database Cloud Service	
About Oracle Database Cloud Service.....	1-1
About Database Cloud Service Database Deployments	1-1
Service Level	1-2
Metering Frequency	1-2
Oracle Database Software Release	1-3
Oracle Database Software Edition	1-3
Oracle Database Type	1-4
Computing Power	1-4
Database Storage	1-5
Automatic Backup Configuration.....	1-5
Before You Begin with Database Cloud Service	1-5
About Database Cloud Service Roles and Users.....	1-5
Accessing the Oracle Database Cloud Service Console	1-6
Typical Workflow for Using Database Cloud Service.....	1-7
2 Managing the Database Cloud Service Life Cycle	
Creating a Database Deployment.....	2-1
Viewing All Database Deployments.....	2-6
Viewing Detailed Information for a Database Deployment.....	2-7
Viewing Activities for Database Deployments in an Identity Domain	2-7
Stopping, Starting and Restarting a Database Deployment	2-8
Rebooting a Compute Node.....	2-11
Scaling a Database Deployment	2-12
Deleting a Database Deployment.....	2-14

3 Administering Database Cloud Service

Adding Temporary Storage to a Database Deployment.....	3-1
Adding an SSH Public Key.....	3-5
Updating the Cloud Tooling on Database Cloud Service.....	3-5
Updating the Cloud Tooling on Deployments Hosting Single-Instance Databases	3-6
Updating the Cloud Tooling on Deployments Hosting Oracle RAC Databases	3-7
Maintaining the Manageability of Database Cloud Service	3-7
Administering Oracle REST Data Services (ORDS).....	3-7
Adding a Signed SSL Certificate to Oracle REST Data Services.....	3-8
Stopping Oracle REST Data Services.....	3-9
Starting Oracle REST Data Services.....	3-9
Loading Data into the Oracle Database on Database Cloud Service	3-10
Tuning Oracle Database Performance on Database Cloud Service.....	3-11
Monitoring and Managing Oracle Database on Database Cloud Service	3-11
Managing the Log and Diagnostic Files on Database Cloud Service	3-11

4 Accessing Database Cloud Service

Connecting to a Compute Node Through Secure Shell (SSH).....	4-1
Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms	4-1
Connecting to a Compute Node Using the PuTTY Program on Windows.....	4-2
Accessing Enterprise Manager Database Express 12c.....	4-4
Accessing Enterprise Manager 11g Database Control	4-4
Accessing the Oracle Application Express Administration Console.....	4-5
Accessing the Oracle Application Express Home Page	4-6
Connecting Remotely to the Database by Using Oracle SQL Developer	4-6
Connecting Remotely to the Database by Using Oracle Net Services	4-7
Copying Files to or from a Database Cloud Service Database Deployment.....	4-8
Generating a Secure Shell (SSH) Public/Private Key Pair.....	4-8
Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility	4-8
Generating an SSH Key Pair on Windows Using the PuTTYgen Program.....	4-9

5 Backing Up and Restoring Databases on Database Cloud Service

About Backing Up Database Deployments on Database Cloud Service.....	5-1
Creating an On-Demand Backup	5-2
Creating an On-Demand Backup by Using the bkup_api Utility	5-3
Creating an On-Demand Backup by Using the raccli Utility	5-3
Deleting a Backup	5-4
Customizing the Current Backup Configuration.....	5-4
Customizing the Current Backup Configuration on Database Deployments Hosting Single-Instance Databases	5-5

Customizing the Current Backup Configuration on Database Deployments Hosting Oracle RAC Databases	5-7
Disabling and Re-enabling Scheduled Backups.....	5-9
Restoring from the Most Recent Backup	5-10
Restoring from the Most Recent Backup by Using the dbaascli Utility	5-11
Restoring from the Most Recent Backup by Using the raccli Utility	5-12
Restoring from a Specific Backup.....	5-12
Restoring from a Specific Backup by Using the dbaascli Utility	5-13
Restoring to a Specific Point in Time	5-14
Restoring to a Specific Point in Time by Using the dbaascli Utility	5-15
Restoring to a Specific Point in Time by Using the raccli Utility	5-16
6 Patching Database Cloud Service	
Viewing Available Patches.....	6-1
Viewing Available Patches by Using the dbpatchm Subcommand	6-2
Checking Prerequisites Before Applying a Patch	6-2
Checking Patch Prerequisites by Using the dbpatchm Subcommand	6-3
Applying a Patch	6-4
Applying a Patch by Using the dbpatchm Subcommand	6-5
Applying a Patch by Using the raccli Utility.....	6-6
Rolling Back a Patch or Failed Patch.....	6-7
Rolling Back a Patch or Failed Patch by Using the dbpatchm Subcommand	6-10
Applying a Patch to a Test Deployment.....	6-12
The dbpatchm.cfg Configuration File.....	6-13
7 Configuring Database Features, Database Options, and Companion Products	
Using Oracle Real Application Clusters (RAC) in Database Cloud Service	7-1
Using Oracle Database Vault in Database Cloud Service.....	7-3
Configuring and Enabling Oracle Database Vault	7-3
Disabling Oracle Database Vault	7-4
Using Tablespace Encryption in Database Cloud Service	7-5
Creating Encrypted Tablespaces	7-5
Managing Tablespace Encryption.....	7-5
Creating and Activating a Master Encryption Key for a PDB	7-6
Using Hybrid Columnar Compression (HCC) in Database Cloud Service	7-8
Using the Demos PDB.....	7-8
Using Oracle Enterprise Manager Cloud Control with Database Cloud Service	7-9
Using Network Encryption and Integrity	7-10
8 Migrating Oracle Databases to Database Cloud Service	
Choosing a Migration Method.....	8-1
Migrating from Oracle Database 11g to Oracle Database 11g in the Cloud	8-3
Migrating from Oracle Database 11g to Oracle Database 12c in the Cloud.....	8-3

Migrating from Oracle Database 12c CDB to Oracle Database 12c in the Cloud	8-4
Migrating from Oracle Database 12c Non-CDB to Oracle Database 12c in the Cloud.....	8-6
Migration Methods	8-8
Data Pump Conventional Export/Import	8-8
Data Pump Full Transportable	8-10
Data Pump Transportable Tablespace	8-13
Remote Cloning a PDB	8-16
Remote Cloning Non-CDB.....	8-17
RMAN Cross-Platform Transportable PDB	8-18
RMAN Cross-Platform Transportable Tablespace Backup Sets.....	8-18
RMAN Transportable Tablespace with Data Pump	8-21
RMAN CONVERT Transportable Tablespace with Data Pump.....	8-24
SQL Developer and INSERT Statements to Migrate Selected Objects	8-27
SQL Developer and SQL*Loader to Migrate Selected Objects	8-28
Unplugging/Plugging a PDB.....	8-28
Unplugging/Plugging Non-CDB	8-29
9 Using Oracle DBaaS Monitor	
About Oracle DBaaS Monitor	9-1
Accessing Oracle DBaaS Monitor.....	9-2
Filtering the Display on DBaaS Monitor Pages.....	9-3
Administering the Listener	9-4
Viewing Listener Status Information	9-4
Starting the Listener	9-4
Stopping the Listener.....	9-4
Verifying that the Listener Knows of a Service.....	9-5
Starting and Stopping the Database Instance	9-5
Starting the Database Instance	9-5
Stopping the Database Instance	9-5
Viewing and Modifying Initialization Parameters	9-6
Viewing Tablespace and Segment Space Usage.....	9-6
Viewing Alert Log Entries and Checking for Errors	9-7
Viewing Real Time SQL Monitor	9-8
Administering Pluggable Databases.....	9-10
Cloning a Pluggable Database.....	9-10
Closing a Pluggable Database	9-10
Creating a Pluggable Database.....	9-11
Dropping a Pluggable Database.....	9-11
Opening a Pluggable Database	9-11
Plugging In a Pluggable Database	9-12
Unplugging a Pluggable Database	9-12
10 Frequently Asked Questions for Database Cloud Service.....	10-1

11 Troubleshooting Database Cloud Service

Problems Creating Deployments	11-1
I cannot create a deployment when I have many database deployments	11-2
I cannot create a deployment, even after waiting for an hour.....	11-2
Problems Administering Deployments.....	11-2
I am required to change the password for the oracle user when I try to connect to a compute node.....	11-2
I get a Linux error 30, Read-only file system, when trying to connect to or work in my environment	11-3
Problems with Scaling.....	11-3
My scaling operation does not start.....	11-3
My deployment is too busy to allow scaling.....	11-3
Problems with Patching and Rollback.....	11-3
I receive a message stating that the virtual machines are unhealthy	11-3
I receive a message stating that the instance is busy with another operation.....	11-3
I cannot apply a patch due to a lack of storage space.....	11-4
My attempt to roll back the January 2015 Patch Set Update (Jan 2015 PSU) fails	11-5
My attempt to roll back the April 2015 Patch Set Update (Apr 2015 PSU) fails	11-5
Problems with Backing Up and Restoring.....	11-5
There is not enough space for my backup	11-5

A Characteristics of a Newly Created Deployment

Characteristics of an Oracle Database Cloud Deployment	A-1
Linux User Accounts.....	A-2
Storage Volumes and File System Layout	A-3
Locations of Installed Software	A-4
Oracle Database Characteristics	A-5
Data Security	A-5
Location of Diagnostic and Log Files	A-6
Characteristics of an Oracle Database Cloud Service Deployment Using Oracle RAC	A-6
Linux User Accounts.....	A-7
Storage Volumes and File System Layout	A-9

B Oracle Cloud Pages for Administering Database Cloud Service

Services Page	B-1
Activity Page	B-3
SSH Access Page	B-5
Overview Page	B-6
Backup Page	B-9
Patching Page	B-9
Provision New Oracle Database Cloud Service: Service Page	B-10
Provision New Oracle Database Cloud Service: Service Details Page.....	B-12

C The dbaascli Utility

dbaascli database bounce	C-2
dbaascli database changepassword	C-3
dbaascli database start	C-3
dbaascli database status.....	C-3
dbaascli database stop.....	C-3
dbaascli dbpatchm apply.....	C-3
dbaascli dbpatchm clonedb.....	C-4
dbaascli dbpatchm list_patches	C-4
dbaascli dbpatchm prereq	C-4
dbaascli dbpatchm rollback.....	C-4
dbaascli dbpatchm switchback	C-4
dbaascli dv off	C-5
dbaascli dv on.....	C-5
dbaascli listener bounce	C-6
dbaascli listener start.....	C-6
dbaascli listener status	C-6
dbaascli listener stop	C-7
dbaascli netsec config	C-7
dbaascli netsec config encryption.....	C-10
dbaascli netsec config integrity.....	C-11
dbaascli netsec status	C-13
dbaascli orec keep list.....	C-13
dbaascli orec keep tag	C-13
dbaascli orec latest.....	C-13
dbaascli orec list	C-14
dbaascli orec pitr	C-14
dbaascli orec scn.....	C-14
dbaascli tde rotate masterkey.....	C-14
dbaascli tde status.....	C-15

D The raccli Utility

raccli apply patch.....	D-2
raccli create backup	D-2
raccli create recovery.....	D-3
raccli describe job.....	D-4
raccli describe system.....	D-5
raccli list backup	D-6
raccli list backupconfig	D-7
raccli list jobs	D-8
raccli list recovery	D-8

raccli update backupconfig	D-9
raccli update netsec	D-10
raccli update rdk	D-12
raccli update tde.....	D-13

Preface

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

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

Related Documents

For more information, see these Oracle resources:

- *Public Cloud Machine Customer Deployment Guide*
- *Public Cloud Machine Using Oracle Compute Cloud Service*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	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 Database Cloud Service

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

Topics

- [About Oracle Database Cloud Service](#)
- [About Database Cloud Service Database Deployments](#)
- [Before You Begin with Database Cloud Service](#)
- [About Database Cloud Service Roles and Users](#)
- [Accessing the Oracle Database Cloud Service Console](#)
- [Typical Workflow for Using Database Cloud Service](#)

About Oracle Database Cloud Service

Oracle Database Cloud Service provides you the ability to deploy Oracle databases in the Cloud, with each database deployment containing a single Oracle database. You have full access to the features and operations available with Oracle Database, but with Oracle providing the computing power, physical storage and (optionally) tooling to simplify routine database maintenance and management operations.

When you create database deployments, Database Cloud Service creates compute nodes to host the database, using resources provided by Oracle Compute Cloud Service.

When you create a database deployment, Oracle Database and supporting software is installed for you, an Oracle database is created using values you provide when creating the deployment, and the database is started. Additionally, you can direct Database Cloud Service to set up automatic backups. Finally, the deployment includes cloud tooling that simplifies backup, recovery, patching and upgrade operations. You have root privilege, so you can load and run software in the compute environment. You have full administrative privileges for the Oracle database. You are responsible for making any changes to the automated maintenance setup, and you are responsible for recovery operations in the event of a failure.

About Database Cloud Service Database Deployments

When you create a new database deployment on Oracle Database Cloud Service, you use the Provision New Oracle Database Cloud 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](#)
- [Computing Power](#)
- [Database Storage](#)
- [Automatic Backup Configuration](#)

Service Level

When creating a database deployment on Oracle Database Cloud Service, **Oracle Database Cloud Service** is the only service level available. With this service level, Oracle Database and supporting software is installed for you, an Oracle database is created using values you provide when creating the deployment, and the database is started. Additionally, you can direct Database Cloud Service to set up automatic backups. Finally, the deployment includes cloud tooling that simplifies backup, recovery, patching and upgrade operations. You have root privilege, so you can load and run software in the compute environment. You have full administrative privileges for the Oracle database. You are responsible for making any changes to the automated maintenance setup, and you are responsible for recovery operations in the event of a failure.

Cloud Tooling for Database Cloud Service

In addition to the capabilities of the web-based Oracle Database Cloud Service console, Database Cloud Service offers the following tools on the compute nodes associated with the Oracle Database Cloud Service service level:

- Simple Automated Backups: use the `bkup_api` utility (`raccli` on deployments that use Oracle Real Application Clusters) to perform on-demand backups and to change how automatic backups are configured. See [Backing Up and Restoring Databases on Database Cloud Service](#).
- Simple Automated Recovery: use the `orec` subcommand of the `dbaascli` utility (`raccli` on deployments that use Oracle Real Application Clusters) to restore from backups. See [Backing Up and Restoring Databases on Database Cloud Service](#).
- Simple Automated Patching: use the `dbpatchm` subcommand of the `dbaascli` utility (`raccli` on deployments that use Oracle Real Application Clusters) to apply patches. See [Patching Database Cloud Service](#).
- New DBaaS Monitor: use the Oracle DBaaS Monitor web application to monitor the Oracle database and computing resources. See [Accessing Oracle DBaaS Monitor](#). Oracle DBaaS Monitor is not available on deployments that use Oracle Real Application Clusters.

Metering Frequency

When creating a database deployment on Oracle Database Cloud Service, you choose one of the following metering frequencies:

- **Hourly**—Pay only for the number of hours used during your billing period.
You cannot switch a deployment from hourly to monthly metering frequency after it is created.
- **Monthly**—Pay one price for the full month irrespective of the number of hours used.
For deployments that are started in the middle of a month, the price is pro-rated; you pay only for the partial month from the start date.
You cannot switch a deployment from monthly to hourly metering frequency after it is created.

Oracle Database Software Release

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

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

Oracle Database Software Edition

When creating a database deployment on Oracle Database Cloud Service, you choose one of the following Oracle Database software editions. For detailed information about the included Oracle Database features, options and packs, see the [Permitted Features](#) section of *Oracle Database Licensing Information User Manual*.

- **Standard Edition**—Oracle Database Standard Edition, which delivers unprecedented ease of use, power, and performance for workgroup, department-level, and Web applications. It includes all the facilities necessary to build business-critical applications.
- **Enterprise Edition**—Oracle Database Enterprise Edition, which provides the performance, availability, scalability, and security required for mission-critical applications such as high-volume online transaction processing (OLTP) applications, query-intensive data warehouses, and demanding Internet applications.
- **Enterprise Edition - High Performance**—provides all the features of Enterprise Edition, plus many of the available Oracle Database options and management packs.
- **Enterprise Edition - Extreme Performance**—provides all the features of Enterprise Edition, plus all of the Oracle Database options and management packs appropriate for use in Oracle Database Cloud Service.

Note:

If you choose Enterprise Edition or Enterprise Edition - High Performance, all available database enterprise management packs and Enterprise Edition options are included in the database deployment. The packs and options that are not part of the software edition you chose are available to you for use on a trial basis.

Oracle Database Type

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

- **Single Instance**—A single Oracle Database instance and database data store hosted on one compute node. For more information about this type, see [Characteristics of an Oracle Database Cloud Deployment](#).
- **Database Clustering with RAC**—A two-node clustered database using Oracle Real Application Clusters technology; two compute nodes each host an Oracle Database instance, and the two instances access the same shared database data store. For more information about this type, see [Characteristics of an Oracle Database Cloud Service Deployment Using Oracle RAC](#).

Not all types are available with all combinations of service level and software edition:

- Single Instance is the only type supported by the Standard Edition software edition.
- Database Clustering with RAC is available only with Enterprise Edition - Extreme Performance software edition.

Computing Power

When creating a database deployment on Oracle Database Cloud Service, you choose the computing power for the associated compute node (or compute nodes in the case of deployments that use Oracle Real Application Clusters) from a list of supported OCPU (Oracle CPU) and processor RAM combinations. These combinations fall into two categories:

- General Purpose:
 - OC4 - 2 OCPU, 15 GB RAM
 - OC5 - 4 OCPU, 30 GB RAM
 - OC6 - 8 OCPU, 60 GB RAM
- High Memory:
 - OC1M - 1 OCPU, 15 GB RAM
 - OC2M - 2 OCPU, 30 GB RAM
 - OC3M - 4 OCPU, 60 GB RAM
 - OC4M - 8 OCPU, 120 GB RAM

Database Storage

When creating a database deployment on Oracle Database Cloud Service, you choose the amount of usable data storage you want for your database in gigabyte (GB) increments up to a maximum of 2 TB (2048 GB).

After you create the database deployment, you can add more data storage as needed. For information, see [Scaling Up the Storage for a Database Deployment](#).

By adding more storage, you can create a database of up to 12 TB (11.2 TB on a deployment hosting an Oracle RAC database with backups to remote storage only).

Automatic Backup Configuration

When creating a new database deployment on Oracle Database Cloud Service, you choose whether you want automatic backups to be configured for the database. Your choices are:

- **Remote Storage Only**—30 days' worth of backups are kept, all stored on an NFS remote location.
If you choose this option, you must provide an NFS remote backup location provided by your tenant administrator.
- **None**—automatic backups are not configured.

Before You Begin with Database Cloud Service

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

- Oracle Compute Cloud Service

Database Cloud Service compute nodes use a variety of Oracle Compute Cloud Service resources. See *Public Cloud Machine Using Oracle Compute Cloud Service* for information about instances and the disk images, compute shapes, and storage volumes that are associated with them.

Before you create a Database Cloud Service instance:

- Get connection information for the public access network the deployment is going to use.
- (Optional) Create a Secure Shell (SSH) public/private key pair to provide. The SSH keys are used to facilitate secure access to the compute nodes that support your database deployments. See [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#).
- (Optional) Get connection information for the NFS remote backup location if the deployment is going to be backed up to remote storage.

About Database Cloud Service Roles and Users

Oracle Database Cloud Service uses roles to control access to tasks and resources. A role assigned to a user gives certain privileges to the user.

The following table summarizes the responsibilities of each cloud role in the context of Database Cloud Service.

Role	Privileges
Tenant User	<ul style="list-style-type: none"> Use the Oracle Database Cloud Service console to create and manage database deployments within a specific tenant. See the next table for more details. Consume the network, compute, and storage resources allocated to a specific tenant.
Note: Tenant Users should contact their Oracle Cloud Administrator or Tenant Administrator for their user sign-in credentials and the URL to access the Oracle Database Cloud Service console.	
Tenant Administrator	<ul style="list-style-type: none"> Like a tenant user, use the Oracle Database Cloud Service console to create and manage database deployments within a specific tenant. See the next table for more details. Like a tenant user, consume the network, compute, and storage resources allocated to a specific tenant. Create and manage Tenant Users within a specific tenant.
Oracle Cloud Administrator	<ul style="list-style-type: none"> Create cloud tenants and configure the tenants for Database Cloud Service. Create users and assign them roles. Create cloud networks and assign them to tenants. Create remote file systems to use for Database Cloud Service backups.

The following table summarizes the privileges given to a **Tenant User** in Database Cloud Service.

Description of Privilege	More Information
Can create and delete database deployments	Creating a Database Deployment Deleting a Database Deployment
Can scale, patch, and back up or restore database deployments	Scaling a Database Deployment Patching Database Cloud Service Backing Up and Restoring Databases on Database Cloud Service

Accessing the Oracle Database Cloud Service Console

To access the Oracle Database Cloud Service console:

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 Cloud Service credentials.

The Oracle Database Cloud Service console opens.

3. If a Welcome page is displayed, view the list of database deployments by clicking **Services** next to “Database Cloud Service”.

For information about the details provided on the Oracle Database Cloud Service console, see [Services Page](#).

Typical Workflow for Using Database Cloud Service

To start using Oracle Database Cloud Service, 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.	Generating a Secure Shell (SSH) Public/Private Key Pair
Create a database deployment	Use a wizard to create a new database deployment.	Creating a Database Deployment
Load data into the database	Use standard Oracle Database tools to load data into your databases.	Loading Data into the Oracle Database on Database Cloud Service
Monitor database deployments	Check on the health and performance of individual database deployments.	Monitoring and Managing Oracle Database on Database Cloud Service
Patch a database deployment	Apply a patch or roll back a patch.	Patching Database Cloud Service
Back up a database deployment	Back up a database or restore a database from a backup.	Backing Up and Restoring Databases on Database Cloud Service

Managing the Database Cloud Service Life Cycle

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

Topics

- [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 a Database Deployment](#)
- [Rebooting a Compute Node](#)
- [Scaling a Database Deployment](#)
- [Deleting a Database Deployment](#)

Creating a Database Deployment

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

However, before using the Provision New Oracle Database Cloud 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 if you are creating a deployment hosting a single-instance database, as described in [After Your Database Deployment Is Created](#).

Before You Begin

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

- An SSH public/private key pair

You must associate an SSH public key with the compute infrastructure supporting the deployment. 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](#).

- A public access network specification

When creating a database deployment, you need to provide the public access network that the deployment will be available on. You get this information from your Tenant Administrator.

- 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, which you get from your Tenant Administrator, must be specified using one of the following formats:

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

Procedure

To create a database deployment on Database Cloud Service:

1. Open the Oracle Database Cloud Service console.

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

2. Click **Create Service**.

The Provision New Oracle Database Cloud Service wizard starts and the Service page is displayed.

3. On the Service page, provide a name and description for the database deployment, and provide information about its high-level characteristics. When you are finished, click **Next** to advance to the Service Details page.

Element	Description
Service Name	The name for the new database deployment. The name: <ul style="list-style-type: none">• Must not exceed 50 characters (15 characters for a deployment that uses Oracle RAC).• Must start with a letter.• Must contain only letters, numbers, or hyphens.• Must not contain any other special characters.• Must be unique within the identity domain.
Description	(Optional) A description for the new database deployment.
Service Level	The service level for the new deployment: <ul style="list-style-type: none">• Oracle Database Cloud Service See Service Level .
Metering Frequency	The metering frequency for the new deployment: <ul style="list-style-type: none">• Hourly• Monthly See Metering Frequency .

Element	Description
Software Release	<p>The release version of Oracle Database for the new deployment:</p> <ul style="list-style-type: none"> • Oracle Database 11g Release 2 • Oracle Database 12c Release 1 • Oracle Database 12c Release 2 <p>See Oracle Database Software Release.</p>
Software Edition	<p>The Oracle Database software package for the new deployment:</p> <ul style="list-style-type: none"> • Standard Edition • Enterprise Edition • Enterprise Edition - High Performance • Enterprise Edition - Extreme Performance <p>See Oracle Database Software Edition.</p>
Database Type	<p>The type of deployment to create:</p> <ul style="list-style-type: none"> • Single Instance—A single Oracle Database instance and database data store hosted on one compute node. For more information about this type, see Characteristics of an Oracle Database Cloud Deployment. • Database Clustering with RAC—A two-node clustered database using Oracle Real Application Clusters technology; two compute nodes each host an Oracle Database instance, and the two instances access the same shared database data store. For more information about this type, see Characteristics of an Oracle Database Cloud Service Deployment Using Oracle RAC. <p>Not all types are available with all combinations of service level and software edition:</p> <ul style="list-style-type: none"> • Single Instance is the only type supported by the Standard Edition software edition. • Database Clustering with RAC is available only with Enterprise Edition - Extreme Performance software edition.

4. On the Service Details page, complete the **Database Configuration** section by providing information about the Oracle Database you want to create. When you are finished, click **Next** to advance to the Confirmation page.

Element	Description
DB Name (SID)	<p>The name for the database instance. The name:</p> <ul style="list-style-type: none"> • Must not exceed 8 characters. • Must start with a letter. • Must contain only letters, numbers, or these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).

Element	Description
PDB Name	<p>(Available only for Oracle Database 12c.)</p> <p>The name for the default pluggable database (PDB). The name:</p> <ul style="list-style-type: none"> • Must not exceed 8 characters. • Must start with a letter. • Must contain only letters, numbers, or these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).
Administration Password Confirm Password	<p>The password for the following administrative users:</p> <ul style="list-style-type: none"> • Oracle Database administrative users • Oracle Application Express admin user <p>The password:</p> <ul style="list-style-type: none"> • 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 \$ (dollar sign). • Must not contain the word “oracle”.
Usable Database Storage (GB)	The amount of storage in GB for actual database data.
Total Data File Storage (GB)	The computed amount of storage in GB that will be allocated to the deployment, including space for operating system and product binaries, supporting files, database data and configuration files, and so on.
Compute Shape	The number of Oracle Compute Units (OCpus) and amount of memory (RAM) for each compute node of the new database deployment. Database Cloud Service offers several OCPU/RAM combinations, as described in Computing Power .
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 Edit to specify the public key. You can upload a file containing the public key value, paste in the value of a public key, or create a system-generated key pair.</p> <p>If you paste in the value, make sure the value does not contain line breaks or end with a line break.</p>
Advanced Settings: Listener Port	<p>The port number for the Oracle Net Listener.</p> <p>The port number must be between 1521 and 5499 (inclusive).</p>
Advanced Settings: Timezone	The time zone for the new database deployment. The default is Coordinated Universal Time (UTC).

Element	Description
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"> • 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
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>
Advanced Settings: Include "Demos" PDB	<p>(Available only for Oracle Database 12c.)</p> <p>Controls whether the "Demos" PDB is to be included in the database. This PDB contains demos for many new features of Oracle Database 12c such as in-memory and multitenant. Usable Data File Storage must be at least 25 GB to include this PDB.</p>

5. On the Service Details page, complete the **Backup and Recovery Configuration** section, choosing a backup option for the database deployment and, depending on your choice, providing information about the NFS remote backup location where backups are to be stored.

Element	Description
Backup Destination	<p>Controls how backups for the deployment are to be configured:</p> <ul style="list-style-type: none"> • Remote Storage Only—backups are configured to be created automatically and stored on an NFS remote backup location. <p>If this choice is selected, the NFS Remote Backup field is displayed.</p> <ul style="list-style-type: none"> • None—Backups are not configured for the deployment. <p>For more information about backups and backup configurations, see About Backing Up Database Deployments on Database Cloud Service.</p>
Total Estimated Monthly Storage	Storage for data files and backups.
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> <p><i>hostname:absolute-path</i> <i>ip-address:absolute-path</i></p>

6. On the Service Details page, complete the **Networks** section by using the **Public Access Network** list to choose the network in which this database deployment will be available.

7. On the Confirmation page, review the information listed. If you are satisfied with the information, click **Create**.

If you need to change the information, 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.

After Your Database Deployment Is Created

After your database deployment is created, you need to perform the following actions if you created a deployment hosting a single-instance database:

- **Update cloud tooling.**

Update the cloud tooling on the deployment, as described in [Updating the Cloud Tooling on Database Cloud Service](#).

- **Confirm correct configuration of automatic backups.**

If you specified remote backups to an NFS location, and the deployment is **not** going to be used by Oracle Java Cloud Service, check that the configuration for automatic backups is correct:

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

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

2. Start a root-user command shell:

```
$ sudo -s  
#
```

3. View the system-wide crontab file:

```
# cat /etc/crontab
```

4. Check for a line of the following form:

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

where *MM* and *HH* are the minute and hour, respectively, of the backup time (on a 24-hour clock) and *SID* is the SID of the database.

5. If you **do not** see such a line, edit the */etc/crontab* file and add a line of the following form:

```
15 0 * * * root /var/opt/oracle/bkup_api/bkup_api bkup_start --dbname=SID
```

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

```
# exit  
$ exit
```

Viewing All Database Deployments

From the Oracle Database Cloud Service Console, you can:

- View the total resources allocated across all Oracle Database Cloud Service database deployments.
- View the details for each deployment.

- Use the search field to filter the list to include only the deployments that contain a given string in their name.

To view all database deployments:

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing the 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 Cloud Service, such as description, subscription mode, and so on.
- View the total resources allocated to the deployment.
- View the details and status information for each node associated with the deployment.

To view detailed information for a database deployment:

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing the 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 Cloud Service in your identity domain. You can restrict the list of activities displayed using search filters.

To view activities for your database deployments:

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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 a Database Deployment

From the Oracle Database Cloud Service console, you can stop, start and restart the compute nodes associated with a database deployment on Oracle Database Cloud Service.

Topics

- [About Stopping, Starting and Restarting a Database Deployment](#)
- [Stopping a Database Deployment](#)
- [Starting a Stopped Database Deployment](#)
- [Restarting a Database Deployment](#)
- [Viewing Past Stop, Start and Restart Activity](#)

About Stopping, Starting and Restarting a Database Deployment

About Stopping a Database Deployment

When you stop a Database Cloud Service database deployment, no access to it is possible and you can perform no management operations on it except to start it or to delete it.

Stopping a database deployment is similar to turning off your personal computer: it has no computing capabilities because the CPU and RAM have no power, but all its other resources—disk drives and the data they contain, static IP reservations, and so on—remain and are ready to be put back into use when power is restored.

When database deployment is stopped, its CPU and RAM (an Oracle Compute Cloud Service instance) are stopped. As a consequence, it consumes no OCPU or memory resources and so metering and billing of these resources stop. However, all the other resources of the database deployment continue to exist and so continue to be metered and billed.

Additionally, when database deployment is stopped, backups of it are not performed.

About Starting a Stopped Database Deployment

When you start a stopped Database Cloud Service database deployment, access to it becomes possible again and you can perform management operations on it such as scaling and patching.

Starting a stopped database deployment is similar to turning your personal computer back on: its computing capabilities are restored because the CPU and RAM again have power, and all its other resources are put back into use.

When database deployment is started:

1. An Oracle Compute Cloud Service instance of the appropriate compute shape (OCPU and memory) is allocated to it.

2. All other Compute Cloud Service resources associated with it when it was created or as the result of a scaling operation are reattached to it.
3. The allocated Oracle Compute Cloud Service instance is started.

After these steps complete, the database deployment is running and available.

Because the started database deployment again consumes OCPU and memory resources, metering and billing of these resources resume.

About Restarting a Database Deployment

When you restart a Database Cloud Service database deployment, it is stopped and then immediately started again. Thus, the information about what happens when stopping and starting a database deployment applies to restarting a database deployment as well, just in immediate succession.

Note: Restarting a database deployment is different from rebooting a compute node of a database deployment. Rebooting a compute node, as described in [Rebooting a Compute Node](#), does not restart the database deployment. It simply reboots the compute node.

Stopping a Database Deployment

In general, you stop a Database Cloud Service database deployment for one of these reasons:

- To prohibit access to it.
- To reduce its cost of operation, especially if its billing frequency is hourly.

Before You Begin

To learn what happens when you stop a Database Cloud Service database deployment, review [About Stopping, Starting and Restarting a Database Deployment](#).

Procedure

To stop a database deployment:

1. Open the Oracle Database Cloud Service console.

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

2. In the list of deployments, click the name of the database deployment you want to stop.

The Oracle Database Cloud Service Overview page is displayed.

3. From the  menu for the database deployment's compute node, select **Stop**, and then confirm the action.

The deployment first has a status of **Maintenance** and then **Stopped** in the Oracle Database Cloud Service console. Note that you cannot scale a stopped deployment.

Starting a Stopped Database Deployment

Before You Begin

To learn what happens when you start a stopped Database Cloud Service database deployment, review [About Stopping, Starting and Restarting a Database Deployment](#).

Procedure

To start a stopped database deployment:

1. Open the Oracle Database Cloud Service console.
For detailed instructions, see [Accessing the Oracle Database Cloud Service Console](#).
2. In the list of deployments, click the name of the stopped database deployment you want to start.
The Oracle Database Cloud Service Overview page is displayed.

3. From the  menu for the database deployment's compute node, select **Start**, and then confirm the action.

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

Restarting a Database Deployment

Note: Restarting a database deployment is different from rebooting a compute node of a database deployment. Rebooting a compute node, as described in [Rebooting a Compute Node](#), does not restart the database deployment. It simply reboots the compute node.

Before You Begin

To learn what happens when you restart a Database Cloud Service database deployment, review [About Stopping, Starting and Restarting a Database Deployment](#).

Procedure

To restart a database deployment:

1. Open the Oracle Database Cloud Service console.
For detailed instructions, see [Accessing the Oracle Database Cloud Service Console](#).
2. In the list of deployments, click the name of the database deployment you want to restart.
The Oracle Database Cloud Service Overview page is displayed.

3. From the  menu for the database deployment's compute node, select **Restart**, and then confirm the action.
The deployment 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:

1. Open the Oracle Database Cloud Service console.

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

2. 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.

3. 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.

4. 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.

Rebooting a Compute Node

On occasion, you might find it necessary to reboot a compute node associated with Oracle Database Cloud Service. Follow these steps to perform the operation.

Note:

Rebooting a compute node is different from restarting a compute node. Restarting a compute node, as described in [Restarting a Database Deployment](#), stops and removes the Compute Cloud Service instance on which the compute node is running and then creates and starts a new Compute Cloud Service instance for the compute node. Rebooting a compute node uses a Linux command to restart the Compute Cloud Service instance on which the compute node is running.

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

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

2. Start a root-user command shell:

```
$ sudo -s  
#
```

3. Enter the command to reboot the compute node:

```
# reboot
```

Caution:

Do not use the `halt`, `shutdown` or `shutdown -h` commands to shut down the 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.

Your connection to the compute node is closed and the compute node reboots.

Scaling a Database Deployment

If a database deployment on Oracle Database Cloud Service is performing poorly or is running out of storage, you can scale up the environment supporting the database deployment.

Usually, the need to scale arises as the result of analyzing database performance, as described in [Tuning Oracle Database Performance on Database Cloud Service](#).

Occasionally, the need to scale arises from some change made to the database or backup configuration after it was created. For example, if the decision to use the In-Memory Database option was made after database creation, you might need to scale up the compute shape to one of the high-memory options, such as scaling from "OC4 - 2 OCPU, 15 GB RAM" to "OC2M - 2 OCPU, 30 GB RAM".

Scaling the Compute Shape for a Database Deployment

Note:

When you scale the compute shape of a database deployment on Database Cloud Service, the deployment is put into Maintenance status during the operation and it is restarted.

To scale the compute shape for a database deployment:

1. View the overview page for the database deployment:
 - a. Open the Oracle Database Cloud Service console.
For detailed instructions, see [Accessing the Oracle Database Cloud Service Console](#).
 - b. Click the name of the deployment you want to scale.
The Oracle Database Cloud Service Overview page is displayed.
2. Choose the scaling command:
 - For database deployments that use Oracle RAC, click the  next to the deployment name and choose **Scale Up/Down**.
 - For other deployments, click the  menu in the box for the compute node and choose **Scale Up/Down**.

The Scale Up/Down Service overlay is displayed. Note that the overlay includes information about the current compute shape.
3. Select a new compute shape.

Note:

When scaling down a database deployment on Oracle Public Cloud Machine, **do not** scale down to the OC3 shape. Doing so can cause an out-of-memory condition and make the compute nodes of the deployment inaccessible.

4. Click **Yes, Scale Up/Down Service** to scale the database deployment.

The scaling operation begins. The database deployment is in Maintenance status and unavailable while the scaling operation is in progress.

Scaling Up the Storage for a Database Deployment

Note:

When you scale up the storage for a database deployment on Database Cloud Service , the deployment is put into Maintenance status during the operation and it is restarted.

When you scale up the storage for a database deployment, a Compute Cloud Service storage volume is created and attached to the deployment.

This storage volume remains attached and available to the deployment even after it is restarted or is stopped and then started. Also, the storage volume exists until you delete the database deployment, at which time the storage volume is also deleted.

Note:

A compute node can have a maximum of ten storage volumes attached to it.

Five of these ten are used when the database deployment is created, leaving you only five opportunities to scale up storage. Therefore, each scale-up operation you perform can dramatically affect the maximum size your database can grow to.

To scale up the storage for a database deployment:

1. View the overview page for the database deployment:
 - a. Open the Oracle Database Cloud Service console.
For detailed instructions, see [Accessing the Oracle Database Cloud Service Console](#).
 - b. Click the name of the database deployment you want to scale.
The Oracle Database Cloud Service Overview page is displayed.
2. Choose the scaling command:
 - For database deployments that use Oracle RAC, click the  next to the deployment name and choose **Scale Up/Down**.
 - For other deployments, click the  menu in the box for the compute node and choose **Scale Up/Down**.

The Scale Up/Down Service overlay is displayed.
3. In the **Additional Storage (GB)** box, enter an amount raw storage to add to the database deployment, from 1 to 2048 GB in increments of 1 GB.

An Oracle Compute Cloud Service storage volume of the indicated size will be created. Note that a small percentage of this raw storage will be used for file system constructs and other overhead.

Note:

When adding storage to a database deployment that uses Oracle Real Application Clusters (RAC), you should specify the same size as the other storage volume or volumes already in the Oracle ASM disk group you want to scale up: Data or Backup.

4. Specify how the additional storage should be allocated in the **Add Storage to** list:
 - **Create New Storage Volume:** adds a new storage volume to the database deployment and mounts it as the next available `/u0n` mount point. This option is not available for deployments that use Oracle Real Application Clusters.
 - **Extend Data Storage Volume:** adds the storage volume to the existing Linux LVM disk group (or Oracle ASM disk group on deployments that use Oracle Real Application Clusters) for database data storage.
 - **Extend Backup Storage Volume:** adds the storage volume to the existing Linux LVM disk group (or Oracle ASM disk group on deployments that use Oracle Real Application Clusters) for backup and FRA storage.
5. Click **Yes, Scale Up/Down Service** to scale the database deployment.

The scaling operation begins. The deployment is in Maintenance status and unavailable while the scaling operation is in progress.

Deleting a Database Deployment

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

To delete a database deployment:

1. Open the Oracle Database Cloud Service console.
For detailed instructions, see [Accessing 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. Optionally, you can also select the option to delete the backups associated with the database deployment.

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

Administering Database Cloud Service

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

Topics

- [Adding Temporary Storage to a Database Deployment](#)
- [Adding an SSH Public Key](#)
- [Updating the Cloud Tooling on Database Cloud Service](#)
- [Maintaining the Manageability of Database Cloud Service](#)
- [Administering Oracle REST Data Services \(ORDS\)](#)
- [Loading Data into the Oracle Database on Database Cloud Service](#)
- [Tuning Oracle Database Performance on Database Cloud Service](#)
- [Monitoring and Managing Oracle Database on Database Cloud Service](#)
- [Managing the Log and Diagnostic Files on Database Cloud Service](#)

Adding Temporary Storage to a Database Deployment

In general, when you add storage to a database deployment on Oracle Database Cloud Service, you want the storage to be permanent; that is, to remain attached and available until the deployment is deleted. To add this kind of permanent storage, scale up the storage as described in [Scaling a Database Deployment](#).

However, you may sometimes want to add storage to a database deployment temporarily for a short period of time, after which you want to detach and delete the storage.

Topics

- [Adding Temporary Storage to a Database Deployment](#)
- [Deleting Temporary Storage from a Database Deployment](#)

Adding Temporary Storage to a Database Deployment

To add temporary storage to a database deployment, you add a storage volume to a compute node. First, you create a Compute Cloud Service storage volume and attach it to the compute node. Then, while logged into the compute node you use Linux commands to partition, format and mount the storage volume.

The storage you add by following these steps is “temporary” in that you can later unmount it from the compute node and delete it. In all other ways it is “permanent”: it remains in existence, even if you delete the database deployment to which it is attached, until you delete it.

Note:

If the database deployment to which you attach this temporary storage is restarted or is stopped and then started, the storage volume becomes detached from the compute node and you must reattach it. For instructions, see [Attaching a Storage Volume to an Instance in Public Cloud Machine Using Oracle Compute Cloud Service](#). After reattaching the storage volume, you must then connect to the compute node and remount it.

If the compute node is rebooted, such as when following the instructions in [Rebooting a Compute Node](#), the temporary storage becomes unmounted and you must remount it.

When adding a Compute Cloud Service storage volume as temporary storage, keep these points in mind:

- A compute node can have a maximum of ten storage volumes attached to it.
- You can create a storage volume from 1 GB to 2048 GB in size, in increments of 1 GB.

To add temporary storage to a database deployment:

1. Create a storage volume by following the instructions in [Creating a Storage Volume in Public Cloud Machine Using Oracle Compute Cloud Service](#).
2. Attach the storage volume to the Compute Cloud Service instance on which the compute node is running by following the instructions in [Attaching a Storage Volume to an Instance in Public Cloud Machine Using Oracle Compute Cloud Service](#).

When you attach the storage volume, assign it the next available disk number. Note down this disk number for later use.

3. Connect as the **opc** user to the compute node.

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

4. Start a root-user command shell:

```
$ sudo -s  
#
```

5. Confirm the addition of the storage volume by using the **ls** command:

```
# ls /dev/xvd*
```

In the listing that is displayed, look for an entry of the form `/dev/xvdLETTER`, where `LETTER` is the letter that is alphabetically one higher than the storage volume’s disk number.

For example, if the storage volume’s disk number is 6, you should look for the entry `/dev/xvdg` because the letter “g” is the seventh letter of the alphabet.

6. Create a single, primary partition that occupies the entire storage volume by using the `fdisk` command. For example:

```
# fdisk /dev/xvdf
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0xaa660f6f.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): c
DOS Compatibility flag is not set

Command (m for help): u
Changing display/entry units to sectors

Command (m for help): n
Command action
      e   extended
      p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (2048-20971519, default 2048): [press Enter]
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): [press
Enter]
Using default value 20971519

Command (m for help): p

Disk /dev/xvdf: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xaa660f6f

      Device Boot      Start        End      Blocks   Id  System
  /dev/xvdf1            2048    20971519    10484736   83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

In this example, note the use of the `p` command to print the partition table before writing the table and exiting. Use this command and note down the name of the new partition as displayed in the Device column.

7. Create a file system on the partition by using the `mkfs` command. For example:

```
# mkfs -t ext4 /dev/xvdf1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
```

```
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2621184 blocks
131059 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

```
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.

8. Create a directory to use as the mount point for the partition by using the `mkdir` command. For example:

```
# mkdir /u05
```

9. Mount the partition on the directory you just created by using the `mount` command. For example:

```
# mount /dev/xvdf1 /u05
```

10. Set the ownership and permissions of the mount-point directory appropriately by using the `chown` and `chmod` commands. For example:

```
# chown oracle:install /u05
# chmod 755 /u05
```

11. Exit the root-user command shell:

```
# exit
$
```

Deleting Temporary Storage from a Database Deployment

To delete temporary storage, you unmount the storage on the compute node and then detach the storage volume from the Compute Cloud Service instance and delete it.

To delete temporary storage from a database deployment:

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Display mounted filesystems and confirm the mount point of your temporary storage volume:

```
# df -hT
```

- Unmount your temporary storage volume; for example:

```
# umount /u05
```

Note:

The Linux command to unmount a volume is `umount` (with no *n*).

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

```
# exit
exit
$ exit
```

- Detach the storage volume from the Compute Cloud Service instance on which the compute node is running by following the instructions in [Detaching a Storage Volume from an Instance in Public Cloud Machine Using Oracle Compute Cloud Service](#).
- Delete the storage volume by following the instructions in [Deleting a Storage Volume in Public Cloud Machine Using Oracle Compute Cloud Service](#).

Adding an SSH Public Key

Should the need arise, you can add an SSH public key to your Oracle Database Cloud Service environment. After you add the public key, you can provide the matching private key to connect to a compute node associated with the Database Cloud Service instance as either the `opc` or the `oracle` user.

To add an SSH public key:

- Open the Oracle Database Cloud Service console.

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

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

The **Add New Key** overlay is displayed with its **Key value** field displaying the most recent SSH public key.

- Specify the new public key using one of the following methods:

- Select **Upload a new SSH Public Key value** and click **Choose File** to select a file that contains the public key.
- Select **Key value**. Delete the current key value and paste the new public key into the text area. Make sure the value does not contain line breaks or end with a line break.

- Click **Add New Key**.

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

Updating the Cloud Tooling on Database Cloud Service

How you update the cloud tooling on Oracle Database Cloud Service depends on the type of database deployment you are using:

- For database deployments hosting a single-instance database, see [Updating the Cloud Tooling on Deployments Hosting Single-Instance Databases](#).
- For database deployments hosting an Oracle Real Application Clusters (RAC) database, see [Updating the Cloud Tooling on Deployments Hosting Oracle RAC Databases](#).

Updating the Cloud Tooling on Deployments Hosting Single-Instance Databases

You can update the cloud-specific tooling included on Database Cloud Service database deployments hosting a single-instance database by applying an RPM file containing the latest version of the tools.

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. Get information about the installed cloud tooling and note down the information:

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

4. Get the file name of latest tooling RPM file available:

```
# ls /mnt/dbaas_patch/dbaastools*
```

Note that the file name may simply be `dbaastools.rpm` or it may contain version information like `dbaastools_16411opcm_161118.rpm`.

5. Get information about the cloud tooling in the latest tooling RPM file:

```
# rpm -qpi /mnt/dbaas_patch/tooling-file-name  
Name      : dbaastools                         Relocations: (not relocatable)  
Version   : version_number                     Vendor: Oracle  
Release   : release_string                     Build Date: ...  
...
```

6. If the cloud tooling in the available RPM file is newer than the installed cloud tooling, remove the installed package and install the update:

- a. Remove the installed cloud tooling:

```
# rpm -ev installed-info
```

where `installed-info` is the information you noted down about the installed cloud tooling; that is, the output from `rpm -qa|grep -i dbaastools`.

- b. Install the cloud tooling update:

```
# rpm -ivh /mnt/dbaas_patch/tooling-file-name
```

7. Run the `patch.sh` script:

```
# /var/opt/oracle/misc/patch.sh
```

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

```
# exit
$ exit
```

Updating the Cloud Tooling on Deployments Hosting Oracle RAC Databases

You use the `raccli` utility to update the cloud tooling on database deployments hosting an Oracle RAC database.

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

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

2. Execute the `raccli update rdk` subcommand:

```
$ raccli update rdk -tag tag-number
```

where `tag-number` is the version of tooling you want to update to, without the dots in the version number. For example, to update to 17.1.2 tooling you would enter `1712`.

Maintaining the Manageability of Database Cloud Service

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

To keep your Database Cloud Service instances manageable, follow these guidelines:

- Do not detach, change file access permissions for, or change the mount point of any storage volume attached to a compute node during the creation of your database deployment. In particular, do not unmount or change the file access permissions of `/u01` through `/u05`.

For details about these volumes, see [Storage Volumes and File System Layout](#).

- Do not change the compute node OS users or manually manipulate SSH key settings associated with your Database Cloud Service instance.
- Apply **only** patches that are available through Database Cloud Service. Do **not** apply patches from any other source unless directed to by Oracle Support.
- Apply the quarterly Patch Set Updates (PSUs) regularly, every quarter if possible.
- Do not change the ports for the Oracle Cloud DBaaS Monitor, Oracle Application Express, Oracle Net Listener, Enterprise Manager Database Express 12c, or Enterprise Manager 11g Database Control.

Administering Oracle REST Data Services (ORDS)

When a database deployment is created on Oracle Database Cloud Service, Oracle REST Data Services (formerly known as Oracle APEX Listener) is started.

Note:

This section does not apply to database deployments that use Oracle Real Application Clusters. Such deployments do not include Oracle REST Data Services.

Topics

- [Adding a Signed SSL Certificate to Oracle REST Data Services](#)
- [Stopping Oracle REST Data Services](#)
- [Starting Oracle REST Data Services](#)

Adding a Signed SSL Certificate to Oracle REST Data Services

You can add a signed SSL certificate to the ORDS environment on a Database Cloud Service database deployment running ORDS 3.0.5 or later.

Deployments created after early October 2016 (version 16.4.1) already have the necessary version of ORDS installed. To check your version of ORDS, connect to the deployment's compute node and enter this command:

```
$ java -jar /u01/app/oracle/product/ords/ords.war version
```

If you need to upgrade your version of ORDS, see [Updating the Cloud Tooling on Deployments Hosting Single-Instance Databases](#).

Before You Begin

To add a signed SSL certificate, you must have the following:

- The SSL certificate file from the certificate provider. This is a `.crt` file.
- The private key file you gave to the certificate provider as part of your Certificate Signing Request (CSR). This is a `.der` or `.pem` file.

Procedure

1. Copy the certificate and private key files to the database deployment's compute node.

Copy these files as the `oracle` user to the following locations:

- Certificate file: `/u01/app/oracle/product/ords/certificate.crt`
- Private key file: `/u01/app/oracle/product/ords/privkey.der` or `/u01/app/oracle/product/ords/privkey.pem`, depending on format.

For instructions on copying files to the compute node, see [Copying Files to or from a Database Cloud Service Database Deployment](#).

2. Connect as the `oracle` user to the compute node.

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

3. If your key file is in `.pem` format, convert it to `.der` format:

```
$ cd /u01/app/oracle/product/ords
$ /usr/bin/openssl pkcs8 -topk8 -inform PEM -outform DER -in privkey.pem -out
privkey.der -nocrypt
$ rm -f privkey.pem
```

4. Edit the `standalone.properties` file of your ORDS environment, adding these lines if missing:

```
ssl.cert=/u01/app/oracle/product/ords/certificate.crt
ssl.cert.key=/u01/app/oracle/product/ords/privkey.der
```

This file is located at `/u01/app/oracle/product/ords/conf/ords/standalone/standalone.properties`.

5. Set the permissions on the certificate and private key files:

```
$ chown oracle:oinstall /u01/app/oracle/product/ords/certificate.crt
$ chown oracle:oinstall /u01/app/oracle/product/ords/privkey.der
$ chmod 400 /u01/app/oracle/product/ords/certificate.crt
$ chmod 400 /u01/app/oracle/product/ords/privkey.der
```

6. Restart ORDS:

```
$ /etc/init.d/ords restart
```

7. Close your connection to the compute node:

```
$ exit
```

Stopping Oracle REST Data Services

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Stop ORDS:

```
# /u01/app/oracle/product/ords/ords stop
INFO: Stopping Oracle REST Data Services...
INFO: Oracle REST Data Services stopped
```

4. Exit the root-user command shell and close your connection to the compute node:

```
# exit
$ exit
```

Starting Oracle REST Data Services

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Start ORDS:

```
# /u01/app/oracle/product/ords/ords start
INFO: Starting Oracle REST Data Services...
INFO: Oracle REST Data Services started with PID pid
```

4. Exit the root-user command shell and close your connection to the compute node:

```
# exit
$ exit
```

Loading Data into the Oracle Database on Database Cloud Service

You load data into an Oracle database on Oracle Database Cloud Service 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 Database Cloud Service](#) for additional techniques and more specific information about migrating existing Oracle databases to Database Cloud Service.

Using SQL*Loader to Load Data into the Database

SQL*Loader is a high-speed data loading utility that loads data from external files into tables in an Oracle database. SQL*Loader accepts input data in a variety of formats, can perform filtering, and can load data into multiple Oracle database tables during the same load session. SQL*Loader provides three methods for loading data: Conventional Path Load, Direct Path Load, and External Table Load.

For information, see "SQL Loader" in *Oracle Database Utilities* for Release [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 Database Cloud Service, 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 Database Cloud Service

You tune the performance of Oracle Database on Oracle Database Cloud Service 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 12.1 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 database deployments created using the High Performance and Extreme Performance software editions. 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 Database Cloud Service to satisfy the need. See [Scaling a Database Deployment](#).

Monitoring and Managing Oracle Database on Database Cloud Service

To monitor and manage the Oracle database deployed on Oracle Database Cloud Service, 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](#).

Beyond these standard tools, Database Cloud Service provides Oracle DBaaS Monitor. This monitor, unique to Oracle Database Cloud Services, supports monitoring and management of both versions of Oracle Database and of computing resources. For more information, see [Using Oracle DBaaS Monitor](#).

Managing the Log and Diagnostic Files on Database Cloud Service

The software components in Oracle Database Cloud Service generate a variety of log and diagnostic files, and not all these files are automatically archived and purged. Thus, managing the identification and removal of these files to avoid running out of file storage space is an important administrative task.

Database deployments that host a single-instance database include the `cleandblogs` script to simplify this administrative task. This script runs weekly as a `crontab` job to archive key files and remove old log and diagnostic files. It uses a configuration file named `cleandblogs.cfg` to determine how long to retain each kind of log or diagnostic file. You can edit this file to change the default retention periods. This file is located at `/var/opt/oracle/cleandb/cleandblogs.cfg`.

The following table lists the parameters that appear in the `cleandblogs.cfg` file, providing a description and the default retention period in days for each file type.

Parameter	Description and Default Value
AlertRetention	Alert log (<code>alert_instance.log</code>) retention value in days. Default value in file: 14
ListenerRetention	Listener log (<code>listener.log</code>) retention value in days. Default value in file: 14
AuditRetentionDB	Database audit (<code>*.aud</code>) retention value in days. Default value in file: 14
CoreRetention	Core dump/files (<code>*.cmdp*</code>) retention value in days. Default value in file: 7
TraceRetention	Trace file (<code>*.tr*</code> and <code>*.prf</code>) retention value in days. Default value in file: 7
longpRetention	Data designated in the Automatic Diagnostic Repository (ADR) as having a long life (the <code>LONGP_POLICY</code> attribute). For information about ADR, see "Automatic Diagnostic Repository (ADR)" in <i>Oracle Database Administrator's Guide for Release 12.2, 12.1 or 11.2</i> . Default value in file: 14
shortpRetention	Data designated in the Automatic Diagnostic Repository (ADR) as having a short life (the <code>SHORTP_POLICY</code> attribute). For information about ADR, see "Automatic Diagnostic Repository (ADR)" in <i>Oracle Database Administrator's Guide for Release 12.2, 12.1 or 11.2</i> . Default value in file: 7
obkupLogRetention	obkup log file retention in days. Default value in file: 30
LogDirRetention	cleandblogs log file retention in days. Default value in file: 14

Archiving Alert Logs and Listener Logs

When cleaning up alert and listener logs, `cleandblogs` first archives and compresses the logs, operating as follows:

1. The current log file is copied to an archive file that ends with a date stamp.
2. The current log file is emptied.
3. The archive file is compressed using `gzip`.
4. Any existing compressed archive files older than the retention period are deleted.

Running the cleandblogs Script Manually

The `cleandblogs` script automatically runs weekly, but you can also run the script manually if the need arises.

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

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

2. Change to the directory containing the `cleandblogs` script:

```
$ cd /var/opt/oracle/cleandb
```

3. Run the `cleandblogs` script:

```
$ ./cleandblogs.pl
```

When running the script manually, you can specify an alternate configuration file to use instead of `cleandblogs.cfg` by using the `--pfile` option:

```
$ ./cleandblogs.pl --pfile config-file-name
```

4. Close your connection to the compute node:

```
$ exit
```

Accessing Database Cloud Service

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

Note:

When you create a Database Cloud Service database deployment, you provide a Secure Shell (SSH) public key value so that you can directly access command-line tools on a compute node of the deployment. For information about creating an SSH public key value, see [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#).

Topics

- [Connecting to a Compute Node Through Secure Shell \(SSH\)](#)
- [Accessing Enterprise Manager Database Express 12c](#)
- [Accessing Enterprise Manager 11g Database Control](#)
- [Accessing the Oracle Application Express Administration Console](#)
- [Accessing the Oracle Application Express Home Page](#)
- [Connecting Remotely to the Database by Using Oracle SQL Developer](#)
- [Connecting Remotely to the Database by Using Oracle Net Services](#)
- [Copying Files to or from a Database Cloud Service Database Deployment](#)
- [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#)

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 Cloud Service, you use Secure Shell (SSH) client software to establish a secure connection and log in as the user `oracle` or the user `opc`.

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

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

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

Before You Begin

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

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

Procedure

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

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

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

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

2. Run the ssh utility:

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

where:

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

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

Connecting to a Compute Node Using the PuTTY Program on Windows

PuTTY is a freely available SSH client program for Windows.

Before You Begin

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

- The IP address of the compute node

The IP address of a compute node associated with a database deployment on Oracle Database Cloud Service is listed on the Oracle Database Cloud Service Overview page. See [Viewing Detailed Information for a Database Deployment](#).

- The SSH private key file that matches the public key associated with the deployment. This private key file must be 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 Cloud Service database deployments created using Oracle Database 12c Release 1.

To access EM Express:

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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 Cloud Service database deployments created using Oracle Database 11g Release 2.

By default, access to Database Control is provided using port 1158.

To access Database Control:

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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.

Accessing the Oracle Application Express Administration Console

Database deployments on Oracle Database Cloud Service include Oracle Application Express, which you manage using the Oracle Application Express administration console. You access this administration console by going to the Oracle Application Express home page and then clicking the **Administration** link.

If you are new to Oracle Application Express, see its [Overview](#) and [Getting Started](#) pages on Oracle Technology Network to learn about its features and get started using it.

To access the administration console:

1. Direct your browser to the appropriate URL, based on which release of Oracle Database is in use:
 - For the CDB in an Oracle Database 12c Release 1 database: `https://node-ip-address/apex/`
 - For a PDB in an Oracle Database 12c Release 1 database: `https://node-ip-address/apex/lowercase-pdb-name/`
 - For an Oracle Database 11g Release 2 database: `https://node-ip-address/apex/`

where `node-ip-address` is the public IP address of the compute node hosting the administration console and `lowercase-pdb-name` is the name of the PDB, with all letters in lowercase. Obtain these values by viewing details as described in [Viewing Detailed Information for a Database Deployment](#).

After directing your browser to the appropriate URL, the Oracle Application Express Login page is displayed.

2. Enter the following information to log in as the administrator. Then, click **Login to Application Express**.
 - In the Workspace box, enter **INTERNAL**.
 - In the Username box, enter **ADMIN**.
 - In the Password box, enter the password specified during the database deployment creation process.

The Oracle Application Express administration home page is displayed.

Note:

If this is the first login to the Oracle Application Express administration console, you will be required to change your password before you can access the administration home page.

Accessing the Oracle Application Express Home Page

Database deployments on Oracle Database Cloud Service include Oracle Application Express, which you access on the Oracle Application Express home page.

If you are new to Oracle Application Express, see its [Overview](#) and [Getting Started](#) pages on Oracle Technology Network to learn about its features and get started using it.

To access the home page, direct your browser to the appropriate URL, based on which release of Oracle Database is in use:

- For the CDB in an Oracle Database 12c Release 1 database: `https://node-ip-address/apex/`
- For a PDB in an Oracle Database 12c Release 1 database: `https://node-ip-address/apex/lowercase-pdb-name/`
- For an Oracle Database 11g Release 2 database: `https://node-ip-address/apex/`

where `node-ip-address` is the public IP address of the compute node hosting Oracle Application Express and `lowercase-pdb-name` is the name of the PDB, with all letters in lowercase. Obtain these values by viewing details as described in [Viewing Detailed Information for a Database Deployment](#).

After the home page is displayed, enter **INTERNAL** as the workspace, enter **ADMIN** as the username, enter the password specified during the database deployment creation process, and then click **Login to Application Express**. Note that the initial password (that is, the password specified when the database deployment was created) must be changed the first time you log in as the **ADMIN** user.

Connecting Remotely to the Database by Using Oracle SQL Developer

You can define an Oracle SQL Developer connection to your database in the cloud to perform operations as you would with an on-premises database.

Creating a SQL Developer Connection

To create a SQL Developer connection to a database deployment:

1. Open SQL Developer. Right-click Connections and select **New Connection**.

Note:

If you are using a version of SQL Developer in which the Connections panel shows both "Connections" and "Cloud Connections", right-click **Connections**. Do not right-click Cloud Connections, which is for Oracle Database Exadata Express Cloud Service.

The New / Select Database Connection dialog appears.

2. Provide the following information and then click **Test**.
 - **Connection Name:** Create a name for this connection.

- **Username:** Name of the database user for the connection. This user must have sufficient privileges to perform the tasks that you want to perform while connected to the database, such as creating, editing, and deleting tables, views, and other objects.
- **Password:** Provide the "Administration" password that you specified when you created the database deployment.
- **Hostname:** Provide the Public IP address for the database deployment compute node you are connecting to.
- **Port:** Provide the listener port number that you specified when you created the database deployment.
- **SID or Service Name:** If you are connecting to an Oracle 11g database or an Oracle 12c container database, provide the SID. If you are connecting to an Oracle 12c pluggable database (PDB), provide the service name instead of the SID.

3. If your test results show success, click **Connect**. You have connected SQL Developer to your database deployment in Oracle Database Enterprise Cloud Service. Now you can use SQL Developer as you normally would with an on-premises database.

Connecting Remotely to the Database by Using Oracle Net Services

Note:

How you connect to an Oracle RAC database on Database Cloud Service differs from how you connect to a non-clustered database. For information, see [Creating an Oracle Net Connection to an Oracle RAC Database](#).

Oracle Database Cloud Service support access to Oracle Database on the standard Oracle Net Listener port.

To create an Oracle Net connection, you can use the easy connect method to specify a connect identifier with the following format:

node-ip-address:1521/sid-or-service-name

For example:

198.51.100.101:1521/ORCL

Creating an Oracle Net Connection to an Oracle RAC Database

On an Oracle RAC database on Database Cloud Service, the SCAN listeners on each compute node listen on the Oracle Net Services port. On the client, specify a connect descriptor that references the SCAN listeners on both compute nodes; for example:

```
alias-name = (DESCRIPTION =
  (ENABLE = BROKEN)
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = node1-ip-address)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = node2-ip-address)(PORT = 1521))
  )
)
```

```
(CONNECT_DATA = (SERVICE_NAME = service-name) )
```

where:

- *alias-name* is the name you use to identify the alias.
- *node1-ip-address* and *node2-ip-address* are the public IP addresses of the two compute nodes associated with the database deployment.
- *service-name* is the service name in the form:

pdb.identity-domain.oraclecloud.internal

For example:

PDB1.usexample5822.oraclecloud.internal

Copying Files to or from a Database Cloud Service Database Deployment

To copy files to or from a database deployment, connect to a compute node of the deployment using a secure copy utility that supports key-based, passwordless authentication. Examples of such utilities are `scp` and `sftp` on Linux and WinSCP and FileZilla on Windows.

When connecting, provide the private key file that matches the public key provided when the database deployment was created, and specify the user as `opc` or `oracle`. Do not attempt to connect as an Oracle Cloud SFTP user. The compute nodes of Database Cloud Service database deployments are not accessible to Oracle Cloud SFTP users.

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.

Backing Up and Restoring Databases on Database Cloud Service

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

Topics

- [About Backing Up Database Deployments on Database Cloud Service](#)
- [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](#)

About Backing Up Database Deployments on Database Cloud Service

By backing up your Oracle Database Cloud Service database deployments, you can protect the software, configuration and database against loss if a failure occurs. By restoring from a backup, you can restore the deployment's software, configuration, and database to their state at the time of the backup.

Database Cloud Service provides a backup feature that backs up:

- The database
- Database configuration files
- Grid Infrastructure configuration files (on deployments hosting an Oracle RAC database)
- Important system and cloud tooling files

To provide this backup feature, Database Cloud Service relies on system utilities and Oracle Database utilities, all of which are installed in the database deployment.

Default Backup Configuration

When you create a database deployment, you choose one of the following backup destinations:

- **Remote Storage Only.** Backups are configured to be created automatically and stored on an NFS remote location.
- **None.** No backup configuration is created.

The backup configuration created when you choose **Remote Storage Only** follows a set of Oracle best-practice guidelines:

- Full (level 0) backup of the database followed by rolling incremental (level 1) backups on a seven-day cycle
- Full backup of selected database configuration files
- Full backup of selected system files
- Automatic backups daily at a time between 11 PM (23:00) and 3 AM (03:00), with the specific time set during the database deployment creation process
- Retention period of 30 days
- Encryption: all backups are encrypted.

Creating an On-Demand Backup

You can create an on-demand backup of an Oracle Database Cloud Service database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to Create an On-Demand Backup](#) at the end of this topic.

Creating an On-Demand Backup by Using the Oracle Database Cloud Service Console

1. Open the Oracle Database Cloud Service console.
For detailed instructions, see [Accessing 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 on database deployments hosting a single-instance database. See [Creating an On-Demand Backup by Using the `bkup_api` Utility](#).
- You can use the `raccli` utility on database deployments hosting an Oracle Real Application Clusters (RAC) database. See [Creating an On-Demand Backup by Using the `raccli` Utility](#).

Creating an On-Demand Backup by Using the `bkup_api` Utility

You can use the `bkup_api` utility to create an on-demand backup of a database deployment hosting a single-instance database.

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. 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
```

- To create a long-term backup, enter the following `bkup_api` command:

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

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
```

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
```

Creating an On-Demand Backup by Using the `raccli` Utility

You can use the `raccli` utility to create an on-demand backup of a database deployment hosting an Oracle Real Application Clusters (RAC) database:

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

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

2. Enter the `raccli create backup` subcommand:

```
$ raccli create backup
$
```

By default, the backup is given a timestamp-based tag. To specify a custom backup tag, add the `-tag` option to the `raccli` command; for example, to create a long-term backup with the tag "monthly", enter the following command:

```
$ raccli create backup -tag monthly
```

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

```
$ raccli describe job
```

Deleting a Backup

You cannot delete backups that are part of the automatic backup configuration, whether they were created using the `bkup_api` utility, the `raccli` utility, or the Oracle Database Cloud Service console. These backups are deleted automatically based on the retention period of the automatic backup configuration.

You can, however, delete long-term backups created using the `bkup_api` utility with the `--keep` option.

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. List the available long-term backups:

```
# /var/opt/oracle/bkup_api/bkup_api recover_list --keep
```

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
```

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

How you customize the current backup configuration depends on the type of database running in the database deployment.

Topics

- [Customizing the Current Backup Configuration on Database Deployments Hosting Single-Instance Databases](#)
- [Customizing the Current Backup Configuration on Database Deployments Hosting Oracle RAC Databases](#)

Customizing the Current Backup Configuration on Database Deployments Hosting Single-Instance Databases

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

Topics

- [Customizing How the Database Is Backed Up](#)
- [Customizing Which System Files Are Backed Up](#)
- [Customizing Which Database Configuration Files Are Backed Up](#)
- [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. Start an RMAN session:

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

3. Enter the SHOW ALL command:

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

A listing of your configuration is displayed.

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

5. Exit the RMAN session:

```
RMAN> exit;
$
```

Customizing Which System Files Are Backed Up

To change which system files and directories are backed up:

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

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

2. Edit the contents of the `/home/oracle/bkup/oscfg.spec` file.

The backup feature provided by Oracle Database Cloud Service backs up the files and folders listed in this specification file.

An example of an `oscfg.spec` file with a default configuration is as follows:

```
## OS Configuration Files
#
# Doc Spec
oscfg.spec
#
# Directories
/etc/rc.d
/home/oracle/bkup
#
# Single files
/home/oracle/.bashrc
/etc/crontab
/etc/sysctl.conf
/etc/passwd
/etc/group
/etc/oraInst.loc
/etc/oratab
/etc/fstab
```

Customizing Which Database Configuration Files Are Backed Up

To change which database configuration files are backed up:

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

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

2. Edit the contents of the `/home/oracle/bkup/dbcfg.spec` file.

The backup feature provided by Oracle Database Cloud Service backs up the files and folders listed in this specification file.

An example of a `dbcfg.spec` file with a default configuration is as follows:

```
### Oracle_Home configuration files.
#
# Doc Spec
dbcfg.spec
# DB id
dbid
#
# Directories
/u01/app/oracle/product/12.1.0/dbhome_1/admin/ORCL/xdb_wallet
/u01/app/oracle/admin/ORCL/xdb_wallet
/u01/app/oracle/admin/ORCL/opc_wallet
# Note: tde_wallet must be backed up in a different location than DATA bkup.
/u01/app/oracle/admin/ORCL/tde_wallet
/u01/app/oracle/admin/ORCL/cat_wallet
#/u01/app/oracle/product/12.1.0/dbhome_1/dbs
#/u01/app/oracle/product/12.1.0/dbhome_1/network/admin
#/u01/app/oraInventory
```

```

#
# Single files
/u01/app/oracle/product/12.1.0/dbhome_1/dbs/opcORCL.ora
/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orapworcl
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/sqlnet.ora
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/tnsnames.ora
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/env_rdbms.mk
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ins_rdbms.mk

```

Customizing the Frequency of Automatic Backups

The backup feature provided by Oracle Database Cloud Service 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`:

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

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.

Customizing the Current Backup Configuration on Database Deployments Hosting Oracle RAC Databases

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

Topics

- [Customizing How the Database Is Backed Up](#)
- [Customizing Which System Files Are Backed Up](#)
- [Customizing Which Database Configuration Files Are Backed Up](#)
- [Customizing Which Grid Infrastructure Configuration Files Are Backed Up](#)
- [Customizing the Time of Automatic Daily 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 **opc** user to the compute node.

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

2. Change to the **oracle** user:

```
$ sudo -s  
# su - oracle
```

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 Which System Files Are Backed Up

To change which system files and directories are backed up:

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

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

2. Edit the contents of the `/opt/oracle/dcs/rdbaas/config/oscfg.spec` file.

The backup feature provided by Oracle Database Cloud Service backs up the files and folders listed in this specification file.

Customizing Which Database Configuration Files Are Backed Up

To change which database configuration files are backed up:

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

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

2. Edit the contents of the `/opt/oracle/dcs/rdbaas/config/dbcfg.spec` file.

The backup feature provided by Oracle Database Cloud Service backs up the files and folders listed in this specification file.

Customizing Which Grid Infrastructure Configuration Files Are Backed Up

To change which grid infrastructure configuration files are backed up:

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

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

2. Edit the contents of the `/opt/oracle/dcs/rdbaas/config/gicfg.spec` file. The backup feature provided by Oracle Database Cloud Service backs up the files and folders listed in this specification file.

Customizing the Time of Automatic Daily Backups

To change the time of day when daily automatic backups are performed:

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

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

2. Enter this `raccli` command:

```
$ raccli update backupconfig -params '{"cronDate" : "time"}'
```

where `time` is the time (using 24-hour, HH:MM format) when daily backups are to occur. For example, "02:45" is 2:45 AM, and "14:45" is 2:45 PM.

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.

Note:

Currently, disabling and re-enabling scheduled backups is not supported for Database Cloud Service database deployments that use Oracle Real Application Clusters (RAC).

Disabling Scheduled Backups

To disable scheduled backups:

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. 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 lines from the file:

- The line for database backups. This line has the following format:

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

- The line for archivelog backups. This line has one of the following formats, depending on the age of your database deployment:

```
0,30 * * * * root /var/opt/oracle/bkup_api/bkup_api bkup_archlogs --  
dbname=dbname
```

or

```
0,30 * * * * root /home/oracle/bkup/dbname/obkup -dbname=dbname -archivelog
```

In all lines, *dbname* is the name of the database that you wish to act on.

Note: You cannot comment out these lines, you must delete them.

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 Cloud Service database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to Restore from the Most Recent Backup](#) at the end of this topic.

Restoring from the Most Recent Backup by Using the Oracle Database Cloud Service Console

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database after recovery

Other Ways to Restore from the Most Recent Backup

- You can use the `dbaascli` utility on database deployments hosting a single-instance database. See [Restoring from the Most Recent Backup by Using the dbaascli Utility](#).
- You can use the `raccli` utility on database deployments hosting an Oracle Real Application Clusters (RAC) database. See [Restoring from the Most Recent Backup by Using the raccli Utility](#).

Restoring from the Most Recent Backup by Using the dbaascli Utility

You can use the `dbaascli` utility to restore from the most recent backup and perform complete recovery on a database deployment hosting a single-instance database:

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Restore the most recent backup and perform complete recovery using the `orec` subcommand of the `dbaascli` utility:

```
# dbaascli orec --args -latest
```

The restore and recover process performs these steps:

- Shuts down the database
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database instance after recovery

4. Exit the root-user command shell:

```
# exit  
$
```

Restoring from the Most Recent Backup by Using the raccli Utility

You can use the `raccli` utility to restore from the most recent backup and perform complete recovery on a database deployment hosting an Oracle Real Application Clusters (RAC) database:

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

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

2. Restore the most recent backup and perform complete recovery using the `create recovery` subcommand of the `raccli` utility:

```
$ raccli create recovery -latest
```

The restore and recover process performs these steps:

- Shuts down the database
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database instances after recovery

Restoring from a Specific Backup

Note:

Currently, restoring from a specific backup is not supported for database deployments that use Oracle Real Application Clusters (RAC).

You can restore a specific backup and perform recovery to that backup on an Oracle Database Cloud Service database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to Restore from a Specific Backup](#) at the end of this topic.

Restoring from a Specific Backup by Using the Oracle Database Cloud Service Console

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database after recovery

Other Ways to Restore from a Specific Backup

- You can use the `dbaascli` utility on database deployments hosting a single-instance database. See [Restoring from a Specific Backup by Using the dbaascli Utility](#).

Restoring from a Specific Backup by Using the dbaascli Utility

You can use the `dbaascli` utility to restore from a specific backup and perform recovery to that backup on a database deployment hosting a single-instance database.

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. List the available backups using the `orec` subcommand of the `dbaascli` utility.

- For normal backups:

```
# dbaascli orec --args -list
```

A list of available backups displays; for example:

Backup Tag	Completion Tag	Config File Location
<hr/>		
TAG20140626T191645	06/26/2014 19:16:45	
TAG20140626T195118	06/26/2014 19:51:18	.../ORCL/oscfgfiles/2014_06_26/
oscfgfiles_20140626_1951.tar.gz		.../ORCL/ohcfgfiles/2014_06_26/
ohcfgfiles_20140626_1951.tar.gz		
Config files relative path to: /u03/app/oracle/fast_recovery_area		

- For long-term backups:

```
# dbaascli orec --args -keep -list
```

A list of available long-term backups displays; for example:

```
TAG20120117T065489
TAG20110117T077324
TAG20100117T023955
```

4. Restore the specific backup you want using the `orec` subcommand:

- For normal backups:

```
# dbaascli orec --args -pitr backup-tag
```

- For long-term backups:

```
# dbaascli orec --args -keep -tag backup-tag
```

where `backup-tag` is the tag of the backup you want to restore.

The restore and recover process performs these steps:

- Shuts down the database
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database instance after recovery

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

```
# exit
$ exit
```

Restoring to a Specific Point in Time

You can restore from a backup and perform recovery to a specific point in time on an Oracle Database Cloud Service database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to Restore to a Specific Point in Time](#) at the end of this topic.

Restoring to a Specific Point in Time by Using the Oracle Database Cloud Service Console

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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.

6. Click **Recover**.

The restore and recover process performs these steps:

- Shuts down the database
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database after recovery

Other Ways to Restore to a Specific Point in Time

- You can use the `dbaascli` utility on database deployments hosting a single-instance database. See [Restoring to a Specific Point in Time by Using the dbaascli Utility](#).
- You can use the `raccli` utility on database deployments hosting an Oracle Real Application Clusters (RAC) database. See [Restoring to a Specific Point in Time by Using the raccli Utility](#).

Restoring to a Specific Point in Time by Using the dbaascli Utility

You can use the `dbaascli` utility to restore from a backup and perform recovery to a specific point in time on a database deployment hosting a single-instance database.

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Restore the most recent backup and perform complete recovery using the `orec` subcommand of the `dbaascli` utility:

```
# dbaascli orec --args -scn SCN
```

where `SCN` is the system change number (SCN) for the end point of the recovery.

The restore and recover process performs these steps:

- Shuts down the database
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database instance after recovery

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

```
# exit
$ exit
```

Restoring to a Specific Point in Time by Using the `raccli` Utility

You can use the `raccli` utility to restore from a backup and perform recovery to a specific point in time on a database deployment hosting an Oracle Real Application Clusters (RAC) database:

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

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

2. Perform recovery to a specific point in time using the `create recovery` subcommand of the `raccli` utility:

```
$ raccli create recovery -pitr -scn SCN
```

or

```
$ raccli create recovery -pitr -timestamp time
```

where `SCN` is the system change number (SCN) for the end point of the recovery and `time` is time (in the format `MM/DD/YYYY HH24:MI:SS`) for the end point of the recovery.

The restore and recover process performs these steps:

- Shuts down the database
- Extracts and restores configuration files
- Prepares for recovery
- Performs the recovery
- Restarts the database instances after recovery

Patching Database Cloud Service

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

Topics

- [Viewing Available Patches](#)
- [Checking Prerequisites Before Applying a Patch](#)
- [Applying a Patch](#)
- [Rolling Back a Patch or Failed Patch](#)
- [Applying a Patch to a Test Deployment](#)
- [The dbpatchm.cfg Configuration File](#)

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](#).

Viewing Available Patches

You can view a list of patches you can apply to a Database Cloud Service database deployment by using the Oracle Database Cloud Service console or, if desired, by using one of the ways listed in [Other Ways to 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 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 `dbpatchm` subcommand of the `dbaascli` utility on database deployments hosting a single-instance database. See [Viewing Available Patches by Using the `dbpatchm` Subcommand](#).

Viewing Available Patches by Using the `dbpatchm` Subcommand

You can use the `dbpatchm` subcommand of the `dbaascli` command to check whether any patches are available.

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

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

2. Start a root-user command shell:

```
$ sudo -s  
#
```

3. View available patches:

```
# dbaascli dbpatchm --run -list_patches
```

A patch update is available if the command response includes the "INFO: images available for patching" message. The patch ID will be displayed as part of the command response. This patch ID can be used to download and apply the patch.

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.
- Verifies that there is enough space in the `/u01` directory to apply the patch.
- Compares the patch's prerequisites to the database deployment by running `opatch prereq` commands.

Checking Prerequisites Before Applying a Patch by Using the Oracle Database Cloud Service Console

Before You Begin

Before checking patch prerequisites, make sure the database deployment has the latest cloud tooling. Some patches require a certain minimum level of cloud tooling. For more information, see [Updating the Cloud Tooling on Database Cloud Service](#).

Procedure

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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 redisplays, 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 `dbpatchm` subcommand of the `dbaascli` utility on database deployments hosting a single-instance database. See [Checking Patch Prerequisites by Using the dbpatchm Subcommand](#).

Checking Patch Prerequisites by Using the dbpatchm Subcommand

You can use the `dbpatchm` subcommand of the `dbaascli` command to check the prerequisites of a patch before you apply it to a database deployment hosting a single-instance database.

Before You Begin

Before checking patch prerequisites, make sure the database deployment has the latest cloud tooling. Some patches require a certain minimum level of cloud tooling. For more information, see [Updating the Cloud Tooling on Database Cloud Service](#).

Procedure

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

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

2. Start a root-user command shell:

```
$ sudo -s
#
```

3. Edit the `/var/opt/oracle/patch/dbpatchm.cfg` patching configuration file, setting the keys to apply the desired patch. For more information about this file and its keys, see [The dbpatchm.cfg Configuration File](#).

4. Check the the prerequisites of a patch:

```
# dbaascli dbpatchm --run -prereq
```

The prerequisites report displays.

5. 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

- Before you apply a patch, make sure the database deployment has the latest cloud tooling. Some patches require a certain minimum level of cloud tooling. For more information, see [Updating the Cloud Tooling on Database Cloud Service](#).
- Before you apply a patch, you should back up the deployment. For instructions, see [Creating an On-Demand Backup](#).

Procedure

1. Open the Oracle Database Cloud Service console.

For detailed instructions, see [Accessing 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. If you want errors to be ignored during the patching operation, select the **Force apply patch** option. Then, click **Patch**.

If the **Force apply patch** option is selected, patch conflicts or errors discovered during the precheck stage of the patching operation are ignored and the patch will

be applied (space permitting). If the option is not selected and conflicts or errors are discovered, the patch will not be applied.

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 of patches in the Available Patches. If the operation failed, the patch remains in the list. In this case, check the Details of Last Patching Activity section for information about the failure.

Note:

Patching operations are performed with a minimum of impact on the functioning of the database. However, during part of the operation the database is shut down for a period of time, thus making it inaccessible.

Other Ways to Apply a Patch

- You can use the `dbpatchm` subcommand of the `dbaascli` utility on database deployments hosting a single-instance database. See [Applying a Patch by Using the `dbpatchm` Subcommand](#).
- You can use the `raccli` utility on database deployments hosting an Oracle Real Application Clusters (RAC) database. See [Applying a Patch by Using the `raccli` Utility](#).

Applying a Patch by Using the `dbpatchm` Subcommand

You can use the `dbpatchm` subcommand of the `dbaascli` utility to apply a patch to a database deployment hosting a single-instance database.

Before you begin

- Before you apply a patch, make sure the database deployment has the latest cloud tooling. For more information, see [Updating the Cloud Tooling on Deployments Hosting Single-Instance Databases](#).
- Before you apply a patch, you should back up the deployment. For instructions, see [Creating an On-Demand Backup](#).

Procedure

To apply a patch to a database deployment by using the `dbpatchm` subcommand:

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. If you have not already done so, edit the `/var/opt/oracle/patchm/dbpatchm.cfg` patching configuration file, setting the keys to apply the desired patch. For more information about this file and its keys, see [The dbpatchm.cfg Configuration File](#).
4. Apply the patch to the deployment using the `dbpatchm` subcommand of `dbaascli`:

```
# dbaascli dbpatchm --run -apply
```

The `dbpatchm` subcommand displays progress as the patch is applied.

Note:

Patching operations are performed with a minimum of impact on the functioning of the database. However, during part of the operation the database is shut down for a period of time, thus making it inaccessible.

5. Exit the root-user command shell:

```
# exit  
$
```

Applying a Patch by Using the `raccli` Utility

You can use the `apply patch` subcommand of the `raccli` utility to apply a patch to a database deployment hosting an Oracle Real Application Clusters (RAC) database.

Before you begin

- Before you apply a patch, make sure the database deployment has the latest cloud tooling. Some patches require a certain minimum level of cloud tooling. For more information, see [Updating the Cloud Tooling on Deployments Hosting Oracle RAC Databases](#).
- Before you apply a patch, you should back up the deployment. For instructions, see [Creating an On-Demand Backup](#).

Procedure

To apply a patch to a database deployment by using the `apply patch` subcommand:

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

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

2. Apply a patch to the Oracle Database home using the `apply patch` subcommand with the `-db` option:

```
$ raccli apply patch -db -tag tag-name -local
```

Where `tag-name` is the name of the patch. If you do not specify a specific patch, the latest patch is applied.

The `local` option causes `raccli` to update the Oracle Database home only on the current (local) compute node. The local node is taken offline, patched, and then brought back online. This enables you to test the patch before applying it to the

second node. To apply it to the second node, you must run the command on that node.

If this option is omitted, the Oracle Database home is updated on all compute nodes associated with the deployment. The node from which you run the command is taken offline, patched, and then brought back online. Then the second node is taken offline, patched, and brought back online.

3. Apply a patch to the Grid Infrastructure home using the `apply patch` subcommand of `raccli`:

```
$ raccli apply patch -gi -tag tag-name -local
```

Where *tag-name* is the name of the patch. If you do not specify a specific patch, the latest patch is applied.

The `local` option causes `raccli` to update the Grid Infrastructure home only on the current (local) compute node. The local node is taken offline, patched, and then brought back online. This enables you to test the patch before applying it to the second node. To apply it to the second node, you must run the command on that node.

If this option is omitted, the Grid Infrastructure home is updated on all compute nodes associated with the deployment. The node from which you run the command is taken offline, patched, and then brought back online. Then the second node is taken offline, patched, and brought back online.

Rolling Back a Patch or Failed Patch

Note:

Currently, rolling back a patch is not supported for database deployments hosting an Oracle Real Application Clusters (RAC) database.

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.

Note:

Oracle strongly recommends against performing the rollback operation on a database deployment that has never had patches applied to it. You should only use the rollback operation on patches that you have applied to a database deployment.

Note:

Beginning with the April 2015 Patch Set Update (Apr 2015 PSU), Oracle adopted a “composite” approach to patch set updates. With this composite approach, a rollback operation restores the software release level to the previous patch set level instead of to the base software release level. For example, if you roll back the April 2015 Patch Set Update, the software is restored to the January 2015 Patch Set Update release level, not the base release level.

Rolling Back a Patch or Failed Patch by Using the Oracle Database Cloud Service Console

Note:

If you are rolling back the April 2015 Patch Set Update (Apr 2015 PSU) or the January 2015 Patch Set Update (Jan 2015 PSU), you need to include the update's overlay patch numbers in the rollback operation. To do so, you must add the overlay patch numbers to the `rollbackpatches.txt` file before rolling back the patch:

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

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

2. Navigate to the `/var/opt/oracle/patch` directory:

```
$ cd /var/opt/oracle/patch
```

3. Change permissions on the `rollbackpatches.txt` file to make it editable:

```
$ chmod +w rollbackpatches.txt
```

4. Use an editor such as `vim` to add the following lines to the end of the `rollbackpatches.txt` file, making sure to include the colon at the end of each line:

- For the Apr 2015 PSU on Oracle Database 12.1.0.2:

```
20281121:  
20415564:datapatch
```

- For the Apr 2015 PSU on Oracle Database 11.2.0.4, Standard Edition:

```
19665921:  
20406239:.../sqlpatch/20406239/postdeinstall.sql
```

- For the Apr 2015 PSU on Oracle Database 11.2.0.4, all Enterprise Editions:

```
19665921:  
20406239:.../sqlpatch/20406239/postdeinstall.sql  
19770063:.../sqlpatch/19770063/postdeinstall.sql
```

- For the Jan 2015 PSU on Oracle Database 12.1.0.2:

```
20281121:  
19877336:datapatch
```

- For the Jan 2015 PSU on Oracle Database 11.2.0.4:

```
19770063:  
19877440:.../sqlpatch/19877440/postdeinstall.sql
```

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 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 redisplays, showing a status message that your request has been submitted, the Administration tile shows the starting time of the rollback operation, and a **Rolling back...** message replaces the **Rollback** button.

Note:

Rollback operations are performed with a minimum of impact on the functioning of the database. However, during a patch rollback operation the database may be shut down for a short period of time, thus making it inaccessible.

Other Ways to Roll Back a Patch or Failed Patch

- You can use the `dbpatchm` subcommand of the `dbaascli` utility on database deployments hosting a single-instance database. See [Rolling Back a Patch or Failed Patch by Using the dbpatchm Subcommand](#).

Rolling Back a Patch or Failed Patch by Using the dbpatchm Subcommand

You can use the `dbpatchm` subcommand of the `dbaascli` utility to roll back the last patch or failed patch attempt to a database deployment hosting a single-instance database.

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

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

2. Start a root-user command shell:

```
$ sudo -s  
#
```

3. If you are rolling back the April 2015 Patch Set Update (Apr 2015 PSU) or the January 2015 Patch Set Update (Jan 2015 PSU), you need to include the update's overlay patch numbers in the rollback operation. To do so, you must add the overlay patch numbers to the `rollbackpatches.txt` file before rolling back the patch:

- a. Change to the `oracle` user:

```
# su - oracle  
$
```

- b. Navigate to the `/var/opt/oracle/patch` directory:

```
$ cd /var/opt/oracle/patch
```

c. Change permissions on the `rollbackpatches.txt` file to make it editable:

```
$ chmod +w rollbackpatches.txt
```

d. Use an editor such as vim to add the following lines to the end of the `rollbackpatches.txt` file, making sure to include the colon at the end of each line:

- For the Apr 2015 PSU on Oracle Database 12.1.0.2:

```
20281121:  
20415564:datapatch
```

- For the Apr 2015 PSU on Oracle Database 11.2.0.4, Standard Edition:

```
19665921:  
20406239:../../../../sqlpatch/20406239/postdeinstall.sql
```

- For the Apr 2015 PSU on Oracle Database 11.2.0.4, all Enterprise Editions:

```
19665921:  
20406239:../../../../sqlpatch/20406239/postdeinstall.sql  
19770063:../../../../sqlpatch/19770063/postdeinstall.sql
```

- For the Jan 2015 PSU on Oracle Database 12.1.0.2:

```
20281121:  
19877336:datapatch
```

- For the Jan 2015 PSU on Oracle Database 11.2.0.4:

```
19770063:  
19877440:../../../../sqlpatch/19877440/postdeinstall.sql
```

e. Return to being the `root` user:

```
$ exit  
#
```

4. Roll back the last patch in the deployment using the `dbpatchm` subcommand of `dbaascli`:

```
# dbaascli dbpatchm --run -rollback
```

The patch is removed and the database instance reverts to its previous state.

Note:

Rollback operations are performed with a minimum of impact on the functioning of the database. However, during part of the operation the database instance is shut down, thus making it inaccessible.

5. Exit the root-user command shell:

```
# exit  
$
```

Applying a Patch to a Test Deployment

You can use the `clonedb` option of the `dbpatchm` subcommand to apply a patch to a test deployment of Oracle Database Cloud Service before you apply it to a live, production database deployment.

Note:

Currently, applying a patch to a test deployment is not supported for database deployments that use Oracle Real Application Clusters (RAC).

To apply a patch to a test deployment:

1. Using the Provision New Oracle Database Cloud Service wizard, create a test deployment on Database Cloud Service, providing the same information as you did when creating the live deployment.
2. Connect as the `opc` user to the test deployment and then perform these steps:

- a. Start a root-user command shell:

```
$ sudo -s  
#
```

- b. Run the following commands:

```
# cd /home/oracle  
# ./deinstall.pl  
...  
# /var/opt/oracle/ocde/ocde -dump -alist=prep sda  
...  
#
```

- c. Exit the root-user command shell and disconnect from the test deployment.

3. Copy the SSH private key file for the SSH key used when creating the test deployment to the live deployment.
4. Connect as the `opc` user to the live deployment and then start a root-user command shell:

```
$ sudo -s  
#
```

5. Edit the `/var/opt/oracle/patch/dbpatchm.cfg` patching configuration file, setting the normal keys to perform a patching operation. In addition, set the following keys:
 - `cloning`: set this key to `yes`
 - `remotenode`: set this key to the IP address of the test deployment
 - `sshkey_f1`: set this key to value of the SSH private key file you copied to the live deployment
 - `upg`: set this key to `upg`

6. Apply the patch to the test deployment using the dbpatchm subcommand of dbaascli:

```
# dbaascli dbpatchm --run -clonedb
```

The dbpatchm subcommand displays progress as it copies information from the live deployment to the test deployment and then applies the patch to the test deployment.

7. Exit the root-user command shell and disconnect from the test deployment.

After applying the patch to the test deployment, confirm appropriate application of the patch on the test deployment. When satisfied, you can delete the test deployment and apply the patch to the live deployment.

The dbpatchm.cfg Configuration File

To perform a patching operation, the dbpatchm subcommand reads and acts on the content of the `/var/opt/oracle/patch/dbpatchm.cfg` patching configuration file. This file, which is created when the database deployment is created, provides information about the locations of various files that may be used in patching operations.

Before using dbpatchm to perform a patching operation, set the value of the `psunum` key in the `dbpatchm.cfg` file to the patch ID of the patch to apply. To find out what patches are available, see [Viewing Available Patches](#).

The `dbpatchm.cfg` file contains additional keys you can edit to customize the patching operation, and comment lines describing the purpose of each key and how to set its value. Here is a sample `dbpatchm.cfg` file showing these keys and comments.

```
# sample config file

# oss storage container url with public access, normally should not be changed
# there is a default for this now, that is set to production container
oss_container_url=""

# change this following golden image zip/psu zip file to be used in patching
# this will be used only if you run dbaascli/dbpatchm directly
# keep these files in different location from temporary_space given below & /u01/psu
gold_img_loc="</tmp/db11.2.0.4.0_EE_PSU.tar.gz>"
psu_zip_loc="</tmp/p19121551_112040_Linux-x86-64.zip>

# location where the temporary files will be kept - should have 15GB space minimum
# change this location if needed but do not use /u01/psu - used for conflict check
temporary_space="/u01/download";

# turn this ignore_patch_conflict to 1 to let patching ignore conflicts
ignore_patch_conflict=0

# turn this ignore_space_less_than_15g to 1 to let patching proceed if discspace <
15g
ignore_space_less_than_15g=0

# create /var/opt/oracle/patch/files_to_save.ora with full path of directory or
# files to preserve any special files you may have in your /u01/app directory.
# set this to yes, if you have files_to_save.ora
special_files="no"

# type could be "psu", "upg"
```

```
type="psu"
# psunum is the bugid for PSU's - needed to rollback the psu patches
psunum="

# patching method could be cloning (for temporary instance validation) or psu way
# when cloning is yes, remote node IP needs to be provided and also sys password
cloning="no"
remotenode=""
syspasswd="

# ssh private key needed for cloning that needs a 2nd VM connectivity
# "/root/.ssh/pat.key"
sshkey_fl="

# oracle recovery manager catalog connect string - if configured
rcatconnect="

# data guard patching - need to provide primary and standby ip's and
# private sshkeys to connect to the same
# if dg_inst="yes", then all dg_ parameters need to be provided
dg_inst="no"
dg_primary=""
dg_primary_sshkey_fl=""
dg_standby=""
dg_standby_sshkey_fl=""
```

Configuring Database Features, Database Options, and Companion Products

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

Topics

- [Using Oracle Real Application Clusters \(RAC\) in Database Cloud Service](#)
- [Using Oracle Database Vault in Database Cloud Service](#)
- [Using Tablespace Encryption in Database Cloud Service](#)
- [Creating and Activating a Master Encryption Key for a PDB](#)
- [Using Hybrid Columnar Compression \(HCC\) in Database Cloud Service](#)
- [Using the Demos PDB](#)
- [Using Oracle Enterprise Manager Cloud Control with Database Cloud Service](#)
- [Using Network Encryption and Integrity](#)

Using Oracle Real Application Clusters (RAC) in Database Cloud Service

You can create an Oracle RAC database when creating a database deployment on Oracle Database Cloud Service.

Oracle RAC enhances Oracle Database capabilities so that you can concurrently use multiple database instances on different compute nodes. This allows you to scale workload across multiple database instances in order to efficiently store, update, and retrieve data.

Oracle RAC provides the software that manages multiple servers and database instances as a single set of servers, called a cluster. The data files that comprise the database reside on shared storage that is accessible from all servers that are part of the cluster. Each server in the cluster runs the Oracle RAC software.

Unlike a single-instance Oracle database, which has a one-to-one relationship between data files and the database instance, Oracle RAC databases have a one-to-many relationship between data files and database instances. This means that in an Oracle RAC database multiple database instances access a single set of database files concurrently, allowing you to access the data from any database instance in the database cluster.

This allows you to use horizontal scalability beyond the scope of one compute node, in case this compute node is insufficient to run the desired workload. It also increases availability of the database and the data in case a database instance or compute node fails. The remaining database instance can be used to continue operations while the

failed database instance or compute node is being restarted. Having more than one database instance also allows you to perform rolling patch upgrades.

To create an Oracle RAC database in Database Cloud Service, make the following choices in the Provision New Oracle Database Cloud Service wizard:

- For Service Level, choose **Oracle Database Cloud Service**.
- For Software Edition, choose **Enterprise Edition - Extreme Performance**.
- For Database Type, choose **Database Clustering with RAC**.
- For Compute Shape (on the Service Details page), choose a shape with two OCPUs for trial accounts or a shape with four or more OCPUs for subscription accounts.

When you make these choices, Database Cloud Service creates a two-node Oracle RAC database, hosting the database on two independent compute nodes that share data, fast recovery area, and redo log storage. It creates these compute nodes using computing, storage and networking resources provided by Oracle Compute Cloud service.

Managing a Deployment that Uses Oracle RAC Compared to a Deployment that Doesn't

Because a Database Cloud Service deployment that uses Oracle RAC comprises two compute nodes that each host a RAC database instance, you manage the deployment in slightly different ways:

- Cloud tooling: you use `raccli` instead of `bkup_api` or `dbaascli`. For more information, see [The raccli Utility](#).
- You can stop and start the database instances and even the compute nodes independently of each other. Thus, the database can remain available even when you need to perform maintenance that requires you to stop a database instance or compute node.

More About the Oracle RAC Configuration on Database Cloud Service

- Cluster size: currently, the Oracle RAC database on a Database Cloud Service deployment is limited to a two-node cluster.
- Cloud tooling: Oracle Cloud tooling is provided for the common administrative tasks of scaling, backing up and recovering, and patching. For more information, see [Scaling a Database Deployment](#) and [The raccli Utility](#).
- Networking for client access: to make client connections to the Oracle RAC database, you include particular options in the connection's entry in the client's `tnsnames.ora` file. For more information, see [Connecting Remotely to the Database by Using Oracle Net Services](#).
- Database file storage: storage for database data files, the fast recovery area, and the redo logs is created and managed using Oracle Automatic Storage Management (ASM) and Oracle Automatic Storage Management Cluster File System (ACFS) instead of Linux LVM.

- Included software: Oracle Grid Infrastructure, Oracle ASM and Oracle ACFS are included; Oracle Application Express, Oracle REST Data Services, and Oracle DBaaS Monitor are not currently included.

Using Oracle Database Vault in Database Cloud Service

You can use Oracle Database Vault in an Oracle Database Cloud Service database deployment.

Oracle Database Vault provides powerful security controls to help protect application data from unauthorized access, and comply with privacy and regulatory requirements.

You can deploy controls to block privileged account access to application data and control sensitive operations inside the database. Trusted paths can be used to add additional security controls to authorized data access and database changes. Through the runtime analysis of privileges and roles, you can increase the security of existing applications by implementing least privileges and reducing the attack profile of your database accounts. Oracle Database Vault secures existing database environments transparently, eliminating costly and time consuming application changes.

The information in this document tells you about enabling and disabling Oracle Database Vault in an Oracle Database Cloud Service database deployment, but does not provide detail on using the features of Oracle Database Vault. Be sure to refer to *Oracle Database Vault Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#) for detailed information on implementing Oracle Database Vault features.

Topics

- [Configuring and Enabling Oracle Database Vault](#)
- [Disabling Oracle Database Vault](#)

Configuring and Enabling Oracle Database Vault

You can use the `dv on` subcommand of the `dbaascli` utility to configure and enable Database Vault with your database.

Oracle Database includes Database Vault, but you must configure and enable it before you can use it.

The `dbaascli` utility provides an easy-to-use interface for configuring and enabling Database Vault. As an alternative to using `dbaascli`, you can follow the steps in "Getting Started with Oracle Database Vault" in *Oracle Database Vault Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#).

Be sure to review "What to Expect After You Enable Oracle Database Vault" in *Oracle Database Vault Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#) to gain an understanding of the impact of enabling and configuring Database Vault.

As part of the configuration process, Database Vault administrative accounts are created. Oracle strongly recommends that you create two accounts for each role. One account, the primary account, will be used on a day-to-day basis and the other account will be used as a backup account in case the password of the primary account is lost and must be reset.

Refer to [dbaascli dv on](#) for additional information about the `dv on` subcommand, including options that can be used to enable Database Vault only for the root container

(CDB) or a specified pluggable database (PDB) in an Oracle Database 12c database deployment.

To enable and configure Database Vault by using the `dv on` subcommand:

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

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

2. Enable and configure Database Vault:

```
$ dbaascli dv on
...
Enter DV owner username: DVownerusername
Enter DV owner password: DVownerpassword
Re-enter DV owner password: DVownerpassword
Enter DV manager username: DVmanagerusername
Enter DV manager password: DVmanagerpassword
Re-enter DV manager password: DVmanagerpassword
...
Successfully configured DV
$
```

Enter a user name and password for the Database Vault Owner and Database Vault Account Manager when prompted. In an Oracle Database 12c database deployment, the Database Vault Owner and Account Manager user names must begin with `c##`.

3. Disconnect from the compute node.

Disabling Oracle Database Vault

You can use the `dv off` subcommand of the `dbaascli` utility to disable Database Vault in your database.

The `dbaascli` utility provides an easy-to-use interface for disabling Database Vault. As an alternative to using `dbaascli`, you can follow the steps in "Disabling and Enabling Oracle Database Vault" in *Oracle Database Vault Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#).

When you install Oracle Database Vault, it revokes a set of privileges from several Oracle Database-supplied users and roles. Be aware that if you disable Oracle Database Vault, these privileges remain revoked. See "Privileges That Are Revoked from Existing Users and Roles" in *Oracle Database Vault Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#) for additional information.

Refer to [dbaascli dv off](#) for additional information about the `dv off` subcommand, including options to disable Database Vault for only the root container (CDB) or a specific pluggable database (PDB) in an Oracle Database 12c database deployment.

To enable and configure Database Vault by using the `dv off` subcommand:

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

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

2. Disable Database Vault:

```
$ dbaascli dv off
...
```

```

Enter DV owner username: DVownerusername
Enter DV owner password: DVownerpassword
...
Successfully configured DV
$
```

Enter the user name and password for the Database Vault Owner when prompted.

3. Disconnect from the compute node.

Using Tablespace Encryption in Database Cloud Service

All new tablespaces you create in a Database Cloud Service database are encrypted by default.

However, not all of the tablespaces created when you create a database deployment are encrypted:

- In an Oracle Database 11g database, none of the tablespaces created when you create a database deployment are encrypted.
- In an Oracle Database 12c Release 1 database, none of the tablespaces created when you create a database deployment are encrypted. This includes the tablespaces in the root (CDB\$ROOT), the seed (PDB\$SEED), and the PDB created when you create a database deployment.
- In an Oracle Database 12c Release 2 database, only the USERS tablespaces created when you create a database deployment are encrypted. None of the other tablespaces are encrypted. This includes the tablespaces in the root (CDB\$ROOT), the seed (PDB\$SEED), and the PDB created when you create a database deployment.

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 Software Keystore and 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 the database deployment is created on Database Cloud Service, a local auto-login software keystore is created. The keystore is local to the compute node and is protected by a system-generated password. The auto-login software keystore is automatically opened when accessed.

You can change (rotate) the master encryption key by using the `tde rotate masterkey` subcommand of the `dbaascli` utility. When you execute this subcommand you will be prompted for the keystore password. Enter the password specified during the database deployment creation process. For example:

```
DBAAS>tde rotate masterkey
Executing command tde rotate masterkey
Enter keystore password:
Successfully rotated TDE masterkey
```

For more information about changing the master encryption key, 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 Database Cloud Service databases, this parameter is set to `CLOUD_ONLY`. See [Viewing and Modifying Initialization Parameters](#) for additional information.

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.
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 <code>ENCRYPTION</code> clause. For non-Database Cloud Service databases, tablespaces will only be encrypted if the <code>ENCRYPTION</code> clause is specified. This is the default value.
DDL	Tablespaces are not transparently encrypted and are only encrypted if the <code>ENCRYPTION</code> 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 Hybrid Columnar Compression (HCC) in Database Cloud Service

Hybrid Columnar Compression (HCC) is a storage-related Oracle Database feature that causes the database to store the same column for a group of rows together.

Storing column data together in this way can dramatically increase the storage savings achieved from compression. Because database operations work transparently against compressed objects, no application changes are required.

The HCC feature is available on database deployments created on Oracle Database Cloud Service.

For more information about HCC, see:

- "Hybrid Columnar Compression" in *Oracle Database Concepts* for Release [12.2](#) or [12.1](#)
- "Consider Using Table Compression" in *Oracle Database Administrator's Guide* for Release [12.2](#) or [12.1](#)
- "*table_compression*" in "CREATE TABLE" in *Oracle Database SQL Language Reference* for Release [12.2](#) or [12.1](#)

Using the Demos PDB

The Demos PDB (pluggable database) contains demos that highlight some of the more popular features of Oracle Database 12c. You can have this PDB included when you create a Database Cloud Service database deployment hosting an Oracle Database 12c single-instance database.

Some of these popular features include JSON in the database, XMLDB, APEX development, the In-Memory option and Data Mining. There is also a web page installed for you that provides access to labs and tutorials to be used with this PDB.

To install the Demos PDB, simply check the **Include "Demos" PDB option** on the Details page of the Provision New Oracle Database Cloud Service wizard when creating a database deployment.

Accessing the Labs and Tutorials

You can access the web page to the Demos PDB's labs and tutorials by directing your browser to the URL `https://node-ip-address/jet/home.html`. In this URL *node-ip-address* is the IP address of the deployment's compute node; it is listed on the deployment's Oracle Database Cloud Service Overview page. To display this page, see [Viewing Detailed Information for a Database Deployment](#).

Adding the Demos PDB to an Existing Database Deployment

If you didn't install the Demos PDB during the creation of your database deployment, you can add it later by following these steps:

1. Connect to the deployment's compute node as the **oracle** user.

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

2. Download the script to create and enable the Demos PDB:

```
$ wget https://storage.us2.oraclecloud.com/v1/dbcsswlibp-usoracle29538/pdb_demo/demo.pl
```

3. Set the file permissions on the script to make it executable:

```
$ chmod a+x demo.pl
```

4. Execute the patching script:

```
$ ./demo.pl
```

5. Disconnect from the compute node:

```
$ exit
```

Using Oracle Enterprise Manager Cloud Control with Database Cloud Service

Oracle Enterprise Manager Cloud Control 12c Release 5 (12.1.0.5) includes Hybrid Cloud Management, which you can use to manage the Oracle Databases on Oracle Database Cloud Service from the same management console that you use for your on-premises databases.

Oracle Enterprise Manager is Oracle's integrated enterprise IT management product line, which provides the industry's only complete, integrated and business-driven enterprise cloud management solution. Oracle Enterprise Manager 12c delivers comprehensive capabilities to manage, migrate, test, and deploy applications and workloads across hybrid clouds. See [EM Hybrid Cloud Management](#) for information about its hybrid cloud features.

To use the Hybrid Cloud Management feature to manage the Oracle Databases on Database Cloud Service, you perform these high-level tasks:

1. Ensure that your on-premises Enterprise Manager Cloud Control Oracle Management Service (OMS) is of version 12.1.0.5, and that at least one 12.1.0.5 Management Agent exists in your enterprise.
2. Configure one or more version 12.1.0.5 Management Agents within your enterprise to act as Hybrid Cloud Gateway Agents, which provide an SSH-based communication channel between Database Cloud Service compute nodes and the on-premises OMS.
3. Ensure that the Hybrid Cloud Gateway Agents and the on-premises OMS can communicate with the Database Cloud Service compute nodes.
4. Deploy Management Agents to Database Cloud Service compute nodes using the Add Host Targets Wizard or Enterprise Manager Command Line Interface.

For the detailed steps to perform these tasks, see [Enabling Hybrid Cloud Management](#) in *Oracle Enterprise Manager Cloud Control Administrator's Guide*, 12c Release 5 (12.1.0.5).

Preserving the Hybrid Cloud Agent Home When Patching a Database Cloud Service Deployment

If you installed the Oracle Enterprise Manager Cloud Control agent under /u01/app/oracle (the Oracle Base directory) the agent home will be moved

to `/u01/app.ORG/oracle` when a database patch is applied to the database deployment using the cloud tooling.

If the agent home has already been moved to `/u01/app.ORG/oracle`, you can copy it back to `/u01/app/oracle` to restore it.

Perform the following steps to configure the patching tools so that they do not move the agent home to `/u01/app.ORG/oracle`:

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

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

2. Create a file named `/var/opt/oracle/patch/files_to_save.ora`.

3. Add the full path of the agent directory to the `/var/opt/oracle/patch/files_to_save.ora` file.

4. Disconnect from the compute node, and then reconnect as the `opc` user.

5. Start a root-user command shell:

```
$ sudo -s  
#
```

6. Edit the patching configuration file, `/var/opt/oracle/patch/dbpatchm.cfg`, and search for the following lines:

```
# create /var/opt/oracle/patch/files_to_save.ora with full path of directory or  
# files to preserve any special files you may have in your /u01/app directory.  
# set this to yes, if you have files_to_save.ora  
special_files="no"
```

7. Change `special_files="no"` to `special_files="yes"`.

8. Save and close the file.

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

After performing these steps, the agent home will be preserved in its original location whenever the database deployment is patched.

Using Network Encryption and Integrity

To secure connections to your Oracle Database Cloud Service 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 Database Cloud Service 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 Database Cloud Service environment

The following procedure outlines the basic steps required to confirm that native Oracle Net encryption and integrity are enabled in your Database Cloud Service 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 $ORACLE_HOME/network/admin
$ ls sqlnet.ora
sqlnet.ora
```

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 Database Cloud Service.

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 Database Cloud Service environment by default, any parameter setting other than rejected, or no setting at all, would result in the use of native encryption and integrity.

Verifying the use of Native Encryption and Integrity

You can verify the use of native Oracle Net 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
```

Migrating Oracle Databases to Database Cloud Service

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

Topics

- [Choosing a Migration Method](#)
- [Migrating from Oracle Database 11g to Oracle Database 11g in the Cloud](#)
- [Migrating from Oracle Database 11g to Oracle Database 12c in the Cloud](#)
- [Migrating from Oracle Database 12c CDB to Oracle Database 12c in the Cloud](#)
- [Migrating from Oracle Database 12c Non-CDB to Oracle Database 12c in the Cloud](#)
- [Migration Methods](#)

Choosing a Migration Method

You can migrate your on-premises Oracle Database database to an Oracle Database Cloud database using a number of different methods that use several different tools.

Not all migration methods apply to all migration scenarios. Many of the migration methods apply only if specific characteristics of the source and destination 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.

Some of the characteristics and factors to consider when choosing a migration method are:

- On-premises database version
- Oracle Database Cloud database version
- On-premises host operating system and version
- On-premises database character set
- Quantity of data, including indexes
- Data types used in the on-premises database
- Storage for data staging
- Acceptable length of system outage

- Network bandwidth

To determine which migration methods are applicable to your migration scenario, gather the following information.

1. Database version of your on-premises 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 on-premises Oracle Database 12c Release 1 databases, the architecture of the database:
 - Multitenant container database (CDB)
 - Non-CDB
3. Endian format (byte ordering) of your on-premises database's host platform

Some platforms are little endian and others are big endian. Query `V$TRANSPORTABLE_PLATFORM` to identify the endian format, and to determine whether cross-platform tablespace transport is supported.

Oracle Database Cloud uses the Linux platform, which is little endian.

4. Database character set of your on-premises database and Oracle Database Cloud Service database
5. Database version of the Oracle Database Cloud database you are migrating to
 - Oracle Database 11g Release 2
 - Oracle Database 12c Release 1

Oracle Database 12c Release 1 databases created on Oracle Database Cloud use CDB architecture. Databases created using the Enterprise Edition software edition are single-tenant, and databases created using the High Performance or Extreme Performance software editions are multitenant.

After gathering this information, use the "source" and "destination" database versions as your guide to see which migration methods apply to your migration scenario:

- [Migrating from Oracle Database 11g to Oracle Database 11g in the Cloud](#)
- [Migrating from Oracle Database 11g to Oracle Database 12c in the Cloud](#)
- [Migrating from Oracle Database 12c CDB to Oracle Database 12c in the Cloud](#)
- [Migrating from Oracle Database 12c Non-CDB to Oracle Database 12c in the Cloud](#)

Migrating from Oracle Database 11g to Oracle Database 11g in the Cloud

You can migrate Oracle Database 11g databases from on-premises to Oracle Database 11g databases in Oracle Database Cloud using several different methods.

The applicability of some of the migration methods depends on the on-premises database's database character set and platform endian format.

If you have not already done so, determine the database character set of your on-premises database, and determine the endian format of the platform your on-premises database resides on. Use this information to help you choose an appropriate method.

- Data Pump Conventional Export/Import

This method can be used regardless of the endian format and database character set of the on-premises database.

For the steps this method entails, see [Data Pump Conventional Export/Import](#).

- Data Pump Transportable Tablespace

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [Data Pump Transportable Tablespace](#).

- RMAN Transportable Tablespace with Data Pump

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [RMAN Transportable Tablespace with Data Pump](#).

- RMAN CONVERT Transportable Tablespace with Data Pump

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

This method is similar to the Data Pump Transportable Tablespace method, with the addition of the RMAN CONVERT command to enable transport between platforms with different endianness. Query V\$TRANSPORTABLE_PLATFORM to determine if the on-premises database platform supports cross-platform tablespace transport and to determine the endian format of the platform. The Oracle Database Cloud platform is little-endian format.

For the steps this method entails, see [RMAN CONVERT Transportable Tablespace with Data Pump](#).

Migrating from Oracle Database 11g to Oracle Database 12c in the Cloud

You can migrate Oracle Database 11g databases from on-premises to Oracle Database 12c databases in Oracle Database Cloud using several different methods.

The applicability of some of the migration methods depends on the on-premises database's database version, database character set and platform endian format.

If you have not already done so, determine the database version and database character set of your on-premises database, and determine the endian format of the

platform your on-premises database resides on. Use this information to help you choose an appropriate method.

- Data Pump Conventional Export/Import

This method can be used regardless of the endian format and database character set of the on-premises database.

For the steps this method entails, see [Data Pump Conventional Export/Import](#).

- Data Pump Transportable Tablespace

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [Data Pump Transportable Tablespace](#).

- RMAN Transportable Tablespace with Data Pump

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [RMAN Transportable Tablespace with Data Pump](#).

- RMAN CONVERT Transportable Tablespace with Data Pump

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

This method is similar to the Data Pump Transportable Tablespace method, with the addition of the RMAN CONVERT command to enable transport between platforms with different endianness. Query V\$TRANSPORTABLE_PLATFORM to determine if the on-premises database platform supports cross-platform tablespace transport and to determine the endian format of the platform. The Oracle Database Cloud platform is little-endian format.

For the steps this method entails, see [RMAN CONVERT Transportable Tablespace with Data Pump](#).

- Data Pump Full Transportable

This method can be used only if the source database release version is 11.2.0.3 or later, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [Data Pump Full Transportable](#).

Migrating from Oracle Database 12c CDB to Oracle Database 12c in the Cloud

You can migrate Oracle Database 12c CDB databases from on-premises to Oracle Database 12c databases in Oracle Database Cloud using several different methods.

The applicability of some of the migration methods depends on the on-premises database's database character set and platform endian format.

If you have not already done so, determine the database character set of your on-premises database, and determine the endian format of the platform your on-premises database resides on. Use this information to help you choose an appropriate method.

- Data Pump Conventional Export/Import

This method can be used regardless of the endian format and database character set of the on-premises database.

For the steps this method entails, see [Data Pump Conventional Export/Import](#).

- Data Pump Transportable Tablespace

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [Data Pump Transportable Tablespace](#).

- RMAN Transportable Tablespace with Data Pump

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [RMAN Transportable Tablespace with Data Pump](#).

- RMAN CONVERT Transportable Tablespace with Data Pump

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

This method is similar to the Data Pump Transportable Tablespace method, with the addition of the RMAN CONVERT command to enable transport between platforms with different endianness. Query V\$TRANSPORTABLE_PLATFORM to determine if the on-premises database platform supports cross-platform tablespace transport and to determine the endian format of the platform. The Oracle Database Cloud platform is little-endian format.

For the steps this method entails, see [RMAN CONVERT Transportable Tablespace with Data Pump](#).

- RMAN Cross-Platform Transportable Tablespace Backup Sets

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [RMAN Cross-Platform Transportable Tablespace Backup Sets](#).

- Data Pump Full Transportable

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [Data Pump Full Transportable](#).

- Unplugging/Plugging (CDB)

This method can be used only if the on-premises platform is little endian, and the on-premises database and Database Cloud Service database have compatible database character sets and national character sets.

For the steps this method entails, see [Unplugging/Plugging a PDB](#).

- Remote Cloning (CDB)

This method can be used only if the on-premises platform is little endian, the on-premises database release is 12.1.0.2 or higher, and the on-premises database and Database Cloud Service database have compatible database character sets and national character sets.

For the steps this method entails, see [Remote Cloning a PDB](#).

- RMAN Cross-Platform Transportable PDB

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [RMAN Cross-Platform Transportable PDB](#).

- SQL Developer and SQL*Loader to Migrate Selected Objects

You can use SQL Developer to create a cart into which you add selected objects to be loaded into your Oracle Database 12c database on the cloud. In this method, you use SQL*Loader to load the data into your cloud database.

For the steps this method entails, see [SQL Developer and SQL*Loader to Migrate Selected Objects](#).

- SQL Developer and `INSERT` Statements to Migrate Selected Objects

You can use SQL Developer to create a cart into which you add selected objects to be loaded into your Oracle Database 12c database on the cloud. In this method, you use SQL `INSERT` statements to load the data into your cloud database.

For the steps this method entails, see [SQL Developer and `INSERT` Statements to Migrate Selected Objects](#).

Migrating from Oracle Database 12c Non-CDB to Oracle Database 12c in the Cloud

You can migrate Oracle Database 12c non-CDB databases from on-premises to Oracle Database 12c databases in Oracle Database Cloud using several different methods.

The applicability of some of the migration methods depends on the on-premises database's database character set and platform endian format.

If you have not already done so, determine the database character set of your on-premises database, and determine the endian format of the platform your on-premises database resides on. Use this information to help you choose an appropriate method.

- Data Pump Conventional Export/Import

This method can be used regardless of the endian format and database character set of the on-premises database.

For the steps this method entails, see [Data Pump Conventional Export/Import](#).

- Data Pump Transportable Tablespace

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [Data Pump Transportable Tablespace](#).

- RMAN Transportable Tablespace with Data Pump

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [RMAN Transportable Tablespace with Data Pump](#).

- RMAN CONVERT Transportable Tablespace with Data Pump

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

This method is similar to the Data Pump Transportable Tablespace method, with the addition of the RMAN CONVERT command to enable transport between platforms with different endianness. Query V\$TRANSPORTABLE_PLATFORM to determine if the on-premises database platform supports cross-platform tablespace transport and to determine the endian format of the platform. The Oracle Database Cloud platform is little-endian format.

For the steps this method entails, see [RMAN CONVERT Transportable Tablespace with Data Pump](#).

- RMAN Cross-Platform Transportable Tablespace Backup Sets

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [RMAN Cross-Platform Transportable Tablespace Backup Sets](#).

- Data Pump Full Transportable

This method can be used only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

For the steps this method entails, see [Data Pump Full Transportable](#).

- Unplugging/Plugging (non-CDB)

This method can be used only if the on-premises platform is little endian, and the on-premises database and Database Cloud Service database have compatible database character sets and national character sets.

You can use the unplug/plug method to migrate an Oracle Database 12c non-CDB database to Oracle Database 12c in the cloud. This method provides a way to consolidate several non-CDB databases into a single Oracle Database 12c CDB on the cloud.

For the steps this method entails, see [Unplugging/Plugging Non-CDB](#).

- Remote Cloning (non-CDB)

This method can be used only if the on-premises platform is little endian, the on-premises database release is 12.1.0.2 or higher, and the on-premises database and Database Cloud Service database have compatible database character sets and national character sets.

You can use the remote cloning method to copy an Oracle Database 12c non-CDB on-premises database to your Oracle Database 12c database in the cloud.

For the steps this method entails, see [Remote Cloning Non-CDB](#).

- SQL Developer and SQL*Loader to Migrate Selected Objects

You can use SQL Developer to create a cart into which you add selected objects to be loaded into your Oracle Database 12c database on the cloud. In this method, you use SQL*Loader to load the data into your cloud database.

For the steps this method entails, see [SQL Developer and SQL*Loader to Migrate Selected Objects](#).

- SQL Developer and `INSERT` Statements to Migrate Selected Objects

You can use SQL Developer to create a cart into which you add selected objects to be loaded into your Oracle Database 12c database on the cloud. In this method, you use SQL `INSERT` statements to load the data into your cloud database.

For the steps this method entails, see [SQL Developer and INSERT Statements to Migrate Selected Objects](#).

Migration Methods

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

Which of these methods apply to a given migration scenario depends on several factors, including the version, character set, and platform endian format of the source and target databases.

Topics

- [Data Pump Conventional Export/Import](#)
- [Data Pump Full Transportable](#)
- [Data Pump Transportable Tablespace](#)
- [Remote Cloning a PDB](#)
- [Remote Cloning Non-CDB](#)
- [RMAN Cross-Platform Transportable PDB](#)
- [RMAN Cross-Platform Transportable Tablespace Backup Sets](#)
- [RMAN Transportable Tablespace with Data Pump](#)
- [RMAN CONVERT Transportable Tablespace with Data Pump](#)
- [SQL Developer and INSERT Statements to Migrate Selected Objects](#)
- [SQL Developer and SQL*Loader to Migrate Selected Objects](#)
- [Unplugging/Plugging a PDB](#)
- [Unplugging/Plugging Non-CDB](#)

Data Pump Conventional Export/Import

You can use this method regardless of the endian format and database character set of the on-premises database.

To migrate an on-premises source database, tablespace, schema, or table to the database on an Oracle Database Cloud Service database deployment using Data Pump Export and Import, you perform these tasks:

1. On the on-premises database host, invoke Data Pump Export and export the on-premises database.
2. Use a secure copy utility to transfer the dump file to the Database Cloud Service compute node.
3. On the Database Cloud Service compute node, invoke Data Pump Import and import the data into the database.
4. After verifying that the data has been imported successfully, you can delete the dump file.

For information about Data Pump Import and Export, see these topics:

- "Data Pump Export Modes" in *Oracle Database Utilities for Release 12.2, 12.1 or 11.2*.
- "Data Pump Import Modes" in *Oracle Database Utilities for Release 12.2, 12.1 or 11.2*.

Data Pump Conventional Export/Import: Example

This example provides a step-by-step demonstration of the tasks required to migrate a schema from an on-premises Oracle database to an Oracle Database Cloud Service database.

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 on-premises database is on a Linux host.

1. On the on-premises database host, invoke Data Pump Export to export the schemas.
 - a. On the on-premises database host, create an operating system directory to use for the on-premises database export files.


```
$ mkdir /u01/app/oracle/admin/orcl/dpdump/for_cloud
```
 - b. On the on-premises database host, invoke SQL*Plus and log in to the on-premises database as the **SYSTEM** user.


```
$ sqlplus system
Enter password: <enter the password for the SYSTEM user>
```
 - c. Create a directory object in the on-premises 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 on-premises database host, invoke Data Pump Export as the **SYSTEM** user or another user with the **DATAPUMP_EXP_FULL_DATABASE** role and export the on-premises schemas. Provide the password for the user when prompted.


```
$ expdp system SCHEMAS=fsowner DIRECTORY=dp_for_cloud
```
2. Use a secure copy utility to transfer the dump file to the Database Cloud Service compute node.

In this example the dump file is copied to the /u01 directory. Choose the appropriate location based on the size of the file that will be transferred.

- a. On the Database Cloud Service compute node, create a directory for the dump file.

```
$ mkdir /u01/app/oracle/admin/ORCL/dpdump/from_onprem
```

- b. Before using the `scp` command to copy the export dump file, make sure the SSH private key that provides access to the Database Cloud Service compute node is available on your on-premises host.

- c. On the on-premises database host, use the SCP utility to transfer the dump file to the Database Cloud Service compute node.

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/dpdump/for_cloud/expdat.dmp \
oracle@IP_address_DBaaS_VM:/u01/app/oracle/admin/ORCL/dpdump/from_onprem
```

3. On the Database Cloud Service compute node, invoke Data Pump Import and import the data into the database.

- a. On the Database Cloud Service compute node, invoke SQL*Plus and log in to the database as the `SYSTEM` user.

```
$ sqlplus system
Enter password: <enter the password for the SYSTEM user>
```

- b. Create a directory object in the Database Cloud Service database.

```
SQL> CREATE DIRECTORY dp_from_onprem AS '/u01/app/oracle/admin/ORCL/dpdump/
from_onprem';
```

- 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 Database Cloud Service compute node, invoke Data Pump Import and connect to the database. Import the data into the database.

```
$ impdp system SCHEMAS=fsowner DIRECTORY=dp_from_onprem
```

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

Data Pump Full Transportable

You can use this method only if the source database release version is 11.2.0.3 or later, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

You can use the Data Pump full transportable method to copy an entire database from your on-premises host to the database on an Oracle Database Cloud Service database deployment.

To migrate an Oracle Database 11g on-premises database to the Oracle Database 12c database on a Database Cloud Service database deployment using the Data Pump full transportable method, you perform these tasks:

1. On the on-premises database host, prepare the database for the Data Pump full transportable export by placing the user-defined tablespaces in READ ONLY mode.
2. On the on-premises database host, invoke Data Pump Export to perform the full transportable export.
3. Use a secure copy utility to transfer the Data Pump Export dump file and the datafiles for all of the user-defined tablespaces to the Database Cloud Service compute node.
4. Set the on-premises tablespaces back to READ WRITE.
5. On the Database Cloud Service compute node, prepare the database for the tablespace import.
6. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the database.
7. After verifying that the data has been imported successfully, you can delete the dump file.

Data Pump Full Transportable: Example

This example provides a step-by-step demonstration of the tasks required to migrate an Oracle Database 11g database to an Oracle Database Cloud Service 12c database.

In this example, the source database is on a Linux host.

1. On the source database host, prepare the database for the Data Pump full transportable export.
 - a. On the source database host, create a directory in the operating system to use for the source export.


```
$ 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. Determine the name(s) of the tablespaces and data files that belong to the user-defined tablespaces by querying DBA_DATA_FILES. These files will also be listed in the export output.


```
SQL> SELECT tablespace_name, file_name FROM dba_data_files;
TABLESPACE_NAME   FILE_NAME
----- -----
USERS           /u01/app/oracle/oradata/orcl/users01.dbf
UNDOTBS1        /u01/app/oracle/oradata/orcl/undotbs01.dbf
SYSAUX          /u01/app/oracle/oradata/orcl/sysaux01.dbf
SYSTEM           /u01/app/oracle/oradata/orcl/system01.dbf
EXAMPLE          /u01/app/oracle/oradata/orcl/example01.dbf
FSDATA           /u01/app/oracle/oradata/orcl/fsdata01.dbf
```

```
SQL> SELECT tablespace_name, file_name FROM dba_data_files;
TABLESPACE_NAME   FILE_NAME
----- -----
USERS           /u01/app/oracle/oradata/orcl/users01.dbf
UNDOTBS1        /u01/app/oracle/oradata/orcl/undotbs01.dbf
SYSAUX          /u01/app/oracle/oradata/orcl/sysaux01.dbf
SYSTEM           /u01/app/oracle/oradata/orcl/system01.dbf
EXAMPLE          /u01/app/oracle/oradata/orcl/example01.dbf
FSDATA           /u01/app/oracle/oradata/orcl/fsdata01.dbf
```

```
FSINDEX          /u01/app/oracle/oradata/orcl/fsindex01.dbf
SQL>
```

- e. On the source database host, set all tablespaces that will be transported (the transportable set) to READ ONLY mode.

```
SQL> ALTER TABLESPACE example READ ONLY;
Tablespace altered.
SQL> ALTER TABLESPACE fsindex READ ONLY;
Tablespace altered.
SQL> ALTER TABLESPACE fsdata READ ONLY;
Tablespace altered.
SQL> ALTER TABLESPACE users READ ONLY;
Tablespace altered.
SQL>
```

- f. Exit from SQL*Plus.

2. On the source database host, invoke Data Pump Export to perform the full transportable export. Specify FULL=y and TRANSPORTABLE=always. Because this is an Oracle Database 11g database and full transportable is an Oracle Database 12c feature, specify VERSION=12. Provide the password for the SYSTEM user when prompted.

```
$ expdp system FULL=y TRANSPORTABLE=always VERSION=12 DUMPFILE=expdat.dmp
DIRECTORY=dp_for_cloud
```

3. Use a secure copy utility to transfer the Data Pump Export dump file and the datafiles for all of the user-defined tablespaces to the Database Cloud Service compute node.

In this example the dump file is copied to the /u01 directory. Choose the appropriate location based on the size of the file that will be transferred.

- a. On the Database Cloud Service compute node, create a directory for the dump file.

```
$ mkdir /u01/app/oracle/admin/ORCL/dpdump/from_source
```

- b. Before using the scp utility to copy files, make sure the SSH private key that provides access to the Database Cloud Service compute node is available on your source host.

- c. On the source database host, use the scp utility to transfer the dump file and all datafiles of the transportable set to the Database Cloud Service compute node.

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/dpdump/for_cloud/expdat.dmp \
oracle@compute_node_IP_address:/u01/app/oracle/admin/ORCL/dpdump/from_source

$ scp -i private_key_file \
/u01/app/oracle/oradata/orcl/example01.dbf \
oracle@compute_node_IP_address:/u02/app/oracle/oradata/ORCL/PDB2

$ scp -i private_key_file \
/u01/app/oracle/oradata/orcl/fsdata01.dbf \
oracle@compute_node_IP_address:/u02/app/oracle/oradata/ORCL/PDB2

$ scp -i private_key_file \
/u01/app/oracle/oradata/orcl/fsindex01.dbf \
```

```

oracle@compute_node_IP_address:/u02/app/oracle/oradata/ORCL/PDB2

$ scp -i private_key_file \
/u01/app/oracle/oradata/orcl/users01.dbf \
oracle@compute_node_IP_address:/u02/app/oracle/oradata/ORCL/PDB2

```

4. Set the source tablespaces back to READ WRITE.
 - a. Invoke SQL*Plus and log in as the SYSTEM user.
 - b. Set the user-defined tablespaces back to READ WRITE mode.

```

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

```
 - c. Exit from SQL*Plus.
5. On the Database Cloud Service compute node, prepare the PDB for the tablespace import.
 - a. On the Database Cloud Service compute node, invoke SQL*Plus and log in to the PDB as the SYSTEM user.
 - b. Create a directory object in the PDB.

```

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

```
6. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the PDB.

Import the data into the database using the TRANSPORT_DATAFILES option.

```

$ impdp system@PDB2 FULL=y DIRECTORY=dp_from_source \
TRANSPORT_DATAFILES='/u02/app/oracle/oradata/ORCL/PDB2/example01.dbf', \
'/u02/app/oracle/oradata/ORCL/PDB2/fsdata01.dbf', \
'/u02/app/oracle/oradata/ORCL/PDB2/fsindex01.dbf', \
'/u02/app/oracle/oradata/ORCL/PDB2/users01.dbf'

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

Data Pump Transportable Tablespace

You can use this method only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

The Transportable Tablespace method is generally much faster than a conventional export/import of the same data because the data files containing all of the actual data are simply copied to the destination location. You use Data Pump to transfer only the metadata of the tablespace objects to the new database.

To migrate an on-premises source database to the database deployment on Oracle Database Cloud Service using the Data Pump Transportable Tablespace method, you perform these tasks:

1. On the on-premises database host, prepare the database for the Data Pump transportable tablespace export.
2. On the on-premises database host, invoke Data Pump Export to perform the transportable tablespace export.
3. Use a secure copy utility to transfer the Data Pump Export dump file and the tablespace datafiles to the Database Cloud Service compute node.
4. Set the on-premises tablespaces back to READ WRITE.
5. On the Database Cloud Service compute node, prepare the database for the tablespace import.
6. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the database.
7. Set the tablespaces on the Database Cloud Service database to READ WRITE mode.
8. After verifying that the data has been imported successfully, you can delete the dump file.

Data Pump Transportable Tablespace: Example

This example provides a step-by-step demonstration of the tasks required to migrate tablespaces in an on-premises Oracle database to an Oracle Database Cloud Service database.

This example performs a migration of the FSDATA and FSINDEX tablespaces.

In this example, the on-premises database is on a Linux host.

1. On the on-premises database host, prepare the database for the Data Pump transportable tablespace export.
 - a. On the on-premises database host, create a directory in the operating system to use for the on-premises export.

```
$ mkdir /u01/app/oracle/admin/orcl/dpdump/for_cloud
```
 - b. On the on-premises database host, invoke SQL*Plus and log in to the on-premises database as the SYSTEM user.

```
$ sqlplus system
Enter password: <enter the password for the SYSTEM user>
```
 - c. Create a directory object in the on-premises database to reference the operating system directory.

```
SQL> CREATE DIRECTORY dp_for_cloud AS '/u01/app/oracle/admin/orcl/dpdump/
for_cloud';
```
 - d. Determine the name(s) of the datafiles 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 = 'FSDATA';
```

```
FILE_NAME
-----
```

```

/u01/app/oracle/oradata/orcl/fsdata01.dbf

SQL> SELECT file_name FROM dba_data_files
  2 WHERE tablespace_name = 'FSINDEX';

FILE_NAME
-----
/u01/app/oracle/oradata/orcl/fsindex01.dbf

```

- e. On the on-premises database host, set all tablespaces that will be transported (the transportable set) to READ ONLY mode.

```

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

```

- f. Exit from SQL*Plus.
2. On the on-premises database host, invoke Data Pump Export to perform the transportable tablespace export.

On the on-premises database host, invoke Data Pump Export and connect to the on-premises database. Export the on-premises 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

```

3. Use a secure copy utility to transfer the Data Pump Export dump file and the tablespace datafiles to the Database Cloud Service compute node.

In this example the dump file is copied to the /u01 directory. Choose the appropriate location based on the size of the file that will be transferred.

- a. On the Database Cloud Service compute node, create a directory for the dump file.
- b. Before using the scp utility to copy files, make sure the SSH private key that provides access to the Database Cloud Service compute node is available on your on-premises host.
- c. On the on-premises database host, use the scp utility to transfer the dump file and all datafiles of the transportable set to the Database Cloud Service compute node.

```

$ mkdir /u01/app/oracle/admin/ORCL/dpdump/from_onprem

$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/dpdump/for_cloud/expdat.dmp \
oracle@IP_address_DBaaS_VM:/u01/app/oracle/admin/ORCL/dpdump/from_onprem

$ scp -i private_key_file \
/u01/app/oracle/oradata/orcl/fsdata01.dbf \
oracle@IP_address_DBaaS_VM:/u02/app/oracle/oradata/ORCL

$ scp -i private_key_file \
/u01/app/oracle/oradata/orcl/fsindex01.dbf \
oracle@IP_address_DBaaS_VM:/u02/app/oracle/oradata/ORCL

```

4. Set the on-premises tablespaces back to READ WRITE.

- a. Invoke SQL*Plus and log in as the SYSTEM user.
 - b. Set the FSDATA and FSINDEX tablespaces back to READ WRITE mode.

```
SQL> ALTER TABLESPACE fsdata READ WRITE;
Tablespace altered.
SQL> ALTER TABLESPACE fsindex READ WRITE;
Tablespace altered.
```
 - c. Exit from SQL*Plus.
5. On the Database Cloud Service compute node, prepare the database for the tablespace import.
 - a. On the Database Cloud Service compute node, invoke SQL*Plus and log in to the database as the SYSTEM user.
 - b. Create a directory object in the Database Cloud Service database.

```
SQL> CREATE DIRECTORY dp_from_onprem AS '/u01/app/oracle/admin/ORCL/dpdump/
from_onprem';
```
 - c. If the owners of the objects that will be imported do not exist in the database, create them before performing the import. The transportable tablespace mode of import does not create the users.

```
SQL> CREATE USER fsowner
  2  PROFILE default
  3  IDENTIFIED BY fspass
  4  TEMPORARY TABLESPACE temp
  5  ACCOUNT UNLOCK;
```
6. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the database.

Import the data into the database using the TRANSPORT_DATAFILES option.

```
$ impdp system DIRECTORY=dp_from_onprem \
TRANSPORT_DATAFILES='/u02/app/oracle/oradata/ORCL/fsdata01.dbf', \
'/u02/app/oracle/oradata/ORCL/fsindex01.dbf'
```
7. Set the tablespaces on the Database Cloud Service database to READ WRITE mode.
 - a. Invoke SQL*Plus and log in as the SYSTEM user.
 - b. 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.
```
 - c. Exit from SQL*Plus.
8. After verifying that the data has been imported successfully, you can delete the expdat .dmp dump file.

Remote Cloning a PDB

You can use this method only if the on-premises platform is little endian, the on-premises database release is 12.1.0.2 or higher, and the on-premises database and

Database Cloud Service database have compatible database character sets and national character sets.

You can use the remote cloning method to copy a PDB from your on-premises Oracle Database 12c database to a PDB in an Oracle Database 12c database on Oracle Database Cloud Service.

To migrate an Oracle Database 12c PDB to a PDB in a Database Cloud Service database deployment using the remote cloning method, you perform these tasks:

1. On the on-premises database host, invoke SQL*Plus and close the on-premises PDB and then reopen it in READ ONLY mode.
2. On the Database Cloud Service compute node, invoke SQL*Plus and create a database link that enables a connection to the on-premises database.
3. On the Database Cloud Service compute node, execute the `CREATE PLUGGABLE DATABASE` command to clone the on-premises PDB.
4. On the Database Cloud Service compute node, open the new PDB by executing the `ALTER PLUGGABLE DATABASE OPEN` command.
5. Optionally, on the on-premises database host invoke SQL*Plus and set the on-premises PDB back to READ WRITE mode.

For more information, see "Cloning a Remote PDB or Non-CDB" in *Oracle Database Administrator's Guide* for Release [12.2](#) or [12.1](#).

Remote Cloning Non-CDB

You can use this method only if the on-premises platform is little endian, the on-premises database release is 12.1.0.2 or higher, and the on-premises database and Database Cloud Service database have compatible database character sets and national character sets.

You can use the remote cloning method to copy an Oracle Database 12c non-CDB on-premises database to a PDB in an Oracle Database 12c database on Oracle Database Cloud Service.

To migrate an Oracle Database 12c non-CDB database to a Database Cloud Service database deployment using the remote cloning method, you perform these tasks:

1. On the on-premises database host, invoke SQL*Plus and set the on-premises database to READ ONLY mode.
2. On the Database Cloud Service compute node, invoke SQL*Plus and create a database link that enables a connection to the on-premises database.
3. On the Database Cloud Service compute node, execute the `CREATE PLUGGABLE DATABASE` command to clone the on-premises non-CDB database.
4. On the Database Cloud Service compute node, execute the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` script.
5. On the Database Cloud Service compute node, open the new PDB by executing the `ALTER PLUGGABLE DATABASE OPEN` command.
6. Optionally, on the on-premises database host invoke SQL*Plus and set the on-premises database back to READ WRITE mode.

For more information, see "Cloning a Remote PDB or Non-CDB" in *Oracle Database Administrator's Guide* for Release [12.2](#) or [12.1](#).

RMAN Cross-Platform Transportable PDB

This method can be used only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

To migrate an Oracle Database 12c PDB to a PDB in an Oracle Database 12c database on an Oracle Database Cloud Service deployment using the RMAN cross-platform transportable PDB method, you perform these tasks:

1. On the on-premises database host, invoke SQL*Plus and close the on-premises PDB.
2. On the on-premises database host, execute the `ALTER PLUGGABLE DATABASE UNPLUG` command to generate an XML file containing the list of datafiles that will be plugged in on the cloud database.
3. On the on-premises database host, invoke RMAN and connect to the root. Execute the `BACKUP FOR TRANSPORT PLUGGABLE DATABASE` command.
4. Use a secure copy utility to transfer the XML file and the backup set to the Database Cloud Service compute node.
5. On the Database Cloud Service compute node, invoke RMAN and connect to the root. Execute the `RESTORE ALL FOREIGN DATAFILES` command.
6. On the Database Cloud Service compute node, invoke SQL*Plus and connect to the root. Execute the `CREATE PLUGGABLE DATABASE` command.
7. On the Database Cloud Service compute node, execute the `ALTER PLUGGABLE DATABASE OPEN` command.

For more information, see "Performing Cross-Platform Data Transport in CDBs and PDBs" in *Oracle Database Backup and Recovery User's Guide* for Release [12.2](#) or [12.1](#).

RMAN Cross-Platform Transportable Tablespace Backup Sets

You can use this method only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

Note:

See *Oracle Database 12c Backup and Recovery User's Guide* for Release [12.2](#) or [12.1](#) for detailed information on a similar method that enables you to perform a cross-platform transport of an entire database. When you transport an entire database to a different platform, the source platform and the destination platform must use the same endian format.

To migrate Oracle Database 12c on-premises tablespaces to an Oracle Database 12c database on an Oracle Database Cloud Service deployment using the RMAN cross-platform transportable backup sets method, you perform these tasks:

1. On the on-premises database host, prepare the database by placing the user-defined tablespaces that you intend to transport in `READ ONLY` mode.

2. On the on-premises database host, invoke RMAN and use the BACKUP command with the TO PLATFORM or FOR TRANSPORT clause and the DATAPUMP clause to create a backup set for cross-platform transport. See in "BACKUP" in *Oracle Database Backup and Recovery Reference* for Release 12.2 or 12.1 for more information on the BACKUP command.
3. Use a secure copy utility to transfer the backup sets, including the Data Pump export dump file, to the Database Cloud Service compute node.
4. Set the on-premises tablespaces back to READ WRITE.
5. On the Database Cloud Service compute node, prepare the database by creating the required schemas.
6. On the Database Cloud Service compute node, invoke RMAN and use the RESTORE command with the *foreignFileSpec* subclause to restore the cross-platform backup.
7. On the Database Cloud Service compute node, set the tablespaces on the database to READ WRITE mode.

For more information, see "Overview of Cross-Platform Data Transport Using Backup Sets" in *Oracle Database Backup and Recovery User's Guide* for Release 12.2 or 12.1.

RMAN Cross-Platform Transportable Tablespace Backup Sets: Example

This example provides a step-by-step demonstration of the tasks required to migrate tablespaces in an Oracle Database PDB to an Oracle Database Cloud Service database.

This example performs a migration of the FSDATA and FSINDEX tablespaces.

In this example, the on-premises database is on a Linux host.

1. On the on-premises database host, prepare the database by creating a directory for the export dump file and placing the user-defined tablespaces that you intend to transport in READ ONLY mode..
 - a. On the on-premises database host, create a directory in the operating system to use for the export dump.


```
$ mkdir /u01/app/oracle/admin/orcl/dpdump/for_cloud
```
 - b. On the on-premises data host, invoke SQL*Plus and log in to the PDB as the SYSTEM user..


```
$ sqlplus system@pdb_servicename
Enter password: enter the password for the SYSTEM user
```
 - c. Create a directory object in the on-premises database to reference the operating system directory.


```
SQL> CREATE DIRECTORY dp_for_cloud AS '/u01/app/oracle/admin/orcl/dpdump/for_cloud';
```
 - d. On the on-premises database host, set all tablespaces that will be transported (the transportable set) to READ ONLY mode.


```
SQL> ALTER TABLESPACE fsindex READ ONLY;
SQL> ALTER TABLESPACE fsdata READ ONLY;
```
 - e. Exit from SQL*Plus.

2. On the on-premises database host, invoke RMAN and use the BACKUP command with the TO PLATFORM or FOR TRANSPORT clause and the DATAPUMP clause to create a backup set for cross-platform transport.

- a. On the on-premises database host, create an operating system directory for the datafiles.

```
$ mkdir /u01/app/oracle/admin/orcl/rman_transdest
```

- b. Invoke RMAN and log in as a user that has been granted the SYSDBA or SYSBACKUP privilege.

```
$ rman target username@pdb_servicename
```

- c. Execute the BACKUP command.

```
RMAN> BACKUP FOR TRANSPORT
2> FORMAT '/u01/app/oracle/admin/orcl/rman_transdest/fs_tbs.bck'
3> TABLESPACE fsdata,fsindex
4> DATAPUMP FORMAT '/u01/app/oracle/admin/orcl/rman_transdest/fs_tbs.dmp' ;
```

- d. Log out of RMAN.

- e. Optionally, navigate to the directory you specified in the BACKUP command to view the files that were created.

```
$ cd /u01/app/oracle/admin/orcl/rman_transdest
$ ls
fs_tbs.bck  fs_tbs.dmp
```

3. Use a secure copy utility to transfer the backup set, including the Data Pump export dump file, to the Database Cloud Service compute node.

- a. On the Database Cloud Service compute node, create a directory for the backup set and dump file.

```
$ mkdir /tmp/from_onprem
```

- b. Before using the scp command to copy files, make sure the SSH private key that provides access to the Database Cloud Service compute node is available on your on-premises host.

- c. On the on-premises database host, use the SCP utility to transfer the backup set and the dump file to the Database Cloud Service compute node.

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/rman_transdest/fs_tbs.bck \
oracle@IP_address_DBaaS_VM:/tmp/from_onprem
```

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/rman_transdest/fs_tbs.dmp \
oracle@IP_address_DBaaS_VM:/tmp/from_onprem
```

```
$
```

4. Set the on-premises tablespaces back to READ WRITE.

- a. Invoke SQL*Plus and log in to the PDB as the SYSTEM user.

- b. Set the FSDATA and FSINDEX tablespaces back to READ WRITE mode.

```
SQL> ALTER TABLESPACE fsdata READ WRITE;
SQL> ALTER TABLESPACE fsindex READ WRITE;
```

- c. Exit from SQL*Plus.

5. On the Database Cloud Service compute node, prepare the database by creating the required schemas.

- a. On the Database Cloud Service compute node, invoke SQL*Plus and log in to the PDB as the SYSTEM user.
- b. If the owners of the objects that will be imported do not exist in the database, create them before performing the RESTORE.

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

6. On the Database Cloud Service compute node, invoke RMAN and use the RESTORE command with the *foreignFileSpec* subclause to restore the cross-platform backup.

- a. Create an operating system directory for the Data Pump Dump file.

```
$ mkdir /tmp/from_onprem
```

- b. Invoke RMAN and log in to the PDB as a user that has been granted the SYSDBA or SYSBACKUP privilege.

```
$ rman target username@pdb_servicename
```

- c. Execute the RESTORE command.

```
RMAN> RESTORE FOREIGN TABLESPACE fsdata,fsindex TO NEW
  2> FROM BACKUPSET '/tmp/from_onprem/fs_tbs.bck'
  3> DUMP FILE DATAPUMP DESTINATION '/tmp/datapump'
  4> FROM BACKUPSET '/tmp/from_onprem/fs_tbs.dmp';
```

- d. Exit from RMAN.

7. On the Database Cloud Service compute node, set the tablespaces to READ WRITE mode.

- a. Invoke SQL*Plus and log in to the PDB as the SYSTEM user.
- b. Set the FSDATA and FSINDEX tablespaces to READ WRITE.

```
SQL> ALTER TABLESPACE fsdata READ WRITE;
SQL> ALTER TABLESPACE fsindex READ WRITE;
```

- c. Exit from SQL*Plus.

8. After verifying that the data has been imported successfully, you can delete the backup set files that were transported from the on-premises host.

RMAN Transportable Tablespace with Data Pump

You can use this method only if the on-premises platform is little endian, and the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

You can use this method to eliminate placing the tablespaces in READ ONLY mode, as required by the Data Pump Transportable Tablespace method.

To migrate an on-premises source database to a database deployment on Oracle Database Cloud Service using the RMAN Transportable Tablespace with Data Pump method, you perform these tasks:

1. On the on-premises database host, invoke RMAN and create the transportable tablespace set.
2. Use a secure copy utility to transfer the Data Pump Export dump file and the tablespace datafiles to the Database Cloud Service compute node.
3. On the Database Cloud Service compute node, prepare the database for the tablespace import.
4. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the database. Import the data into the database using the `TRANSPORT_DATAFILES` option.
5. After verifying that the data has been imported successfully, you can delete the dump file.

RMAN Transportable Tablespace with Data Pump: Example

This example provides a step-by-step demonstration of the tasks required to migrate tablespaces in an on-premises Oracle database to an Oracle Database Cloud Service database.

This example performs a migration of the `FSDATA` and `FSINDEX` tablespaces.

In this example, the on-premises database is on a Linux host.

1. On the on-premises database host, invoke RMAN and create the transportable tablespace set.

- a. On the on-premises database host, create an operating system directory for the datafiles.

```
$ mkdir /u01/app/oracle/admin/orcl/rman_transdest
```

- b. On the on-premises data host, create an operating system directory for the RMAN auxiliary instance files.

```
$ mkdir /u01/app/oracle/admin/orcl/rman_auxdest
```

- c. Invoke RMAN and log in as the `SYSTEM` user. Enter the password for the `SYSTEM` user when prompted.

```
$ rman target system
```

- d. Execute the `TRANSPORT TABLESPACE` command.

```
RMAN> TRANSPORT TABLESPACE fsdata, fsindex
2> TABLESPACE DESTINATION '/u01/app/oracle/admin/orcl/rman_transdest'
3> AUXILIARY DESTINATION '/u01/app/oracle/admin/orcl/rman_auxdest';
```

- e. Log out of RMAN.

- f. Optionally, navigate to the directory you specified for the `TABLESPACE DESTINATION` and view the files that were created by the `TRANSPORT TABLESPACE` operation.

```
$ cd /u01/app/oracle/admin/orcl/rman_transdest
$ ls
dmpfile.dmp  fsdata01.dbf  fsindex01.dbf  impscrpt.sql
```

2. Use a secure copy utility to transfer the Data Pump Export dump file and the tablespace datafiles to the Database Cloud Service compute node.

In this example the dump file is copied to the /u01 directory. Choose the appropriate location based on the size of the file that will be transferred.

- a. On the Database Cloud Service compute node, create a directory for the dump file.

```
$ mkdir /u01/app/oracle/admin/ORCL/dpdump/from_onprem
```

- b. Before using the `scp` command to copy files, make sure the SSH private key that provides access to the Database Cloud Service compute node is available on your on-premises host.

- c. On the on-premises database host, use the SCP utility to transfer the dump file and all datafiles of the transportable set to the Database Cloud Service compute node.

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/rman_transdest/dmpfile.dmp \
oracle@IP_address_DBaaS_VM:/u01/app/oracle/admin/ORCL/dpdump/from_onprem
```

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/rman_transdest/fsdata01.dbf \
oracle@IP_address_DBaaS_VM:/u02/app/oracle/oradata/ORCL
```

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/rman_transdest/fsindex01.dbf \
oracle@IP_address_DBaaS_VM:/u02/app/oracle/oradata/ORCL
```

3. On the Database Cloud Service compute node, prepare the database for the tablespace import.
- a. On the Database Cloud Service compute node, invoke SQL*Plus and log in to the database as the `SYSTEM` user.
- b. Create a directory object in the Database Cloud Service database.

```
SQL> CREATE DIRECTORY dp_from_onprem AS '/u01/app/oracle/admin/ORCL/dpdump/
from_onprem';
```

- c. If the owners of the objects that will be imported do not exist in the database, create them before performing the import. The transportable tablespace mode of import does not create the users.

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

4. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the database.

Import the data into the database using the `TRANSPORT_DATAFILES` option.

```
$ impdp system DIRECTORY=dp_from_onprem DUMPFILE='dmpfile.dmp' \
TRANSPORT_DATAFILES='/u02/app/oracle/oradata/ORCL/fsdata01.dbf', \
'/u02/app/oracle/oradata/ORCL/fsindex01.dbf'
```

5. After verifying that the data has been imported successfully, you can delete the `dmpfile.dmp` dump file.

RMAN CONVERT Transportable Tablespace with Data Pump

You can use this method only if the database character sets of your on-premises database and Oracle Database Cloud Service database are compatible.

This method is similar to the Data Pump Transportable Tablespace method, with the addition of the RMAN `CONVERT` command to enable transport between platforms with different endianness. Query `V$TRANSPORTABLE_PLATFORM` to determine if the on-premises database platform supports cross-platform tablespace transport and to determine the endian format of the platform. The Oracle Database Cloud Service platform is little-endian format.

To migrate tablespaces from your on-premises Oracle database to a database deployment on Database Cloud Service using RMAN, you perform these tasks:

1. On the on-premises database host, prepare the database for the Data Pump transportable tablespace export.
2. On the on-premises database host, invoke Data Pump Export to perform the transportable tablespace export.
3. On the on-premises database host, invoke RMAN and use the `CONVERT TABLESPACE` command to convert the tablespace datafile to the Oracle Database Cloud platform format. Refer to the *Oracle Database Backup and Recovery Reference* for more information on the `CONVERT` command.
4. Use a secure copy utility to transfer the Data Pump Export dump file and the converted tablespace datafiles to the Database Cloud Service compute node.
5. Set the on-premises tablespaces back to `READ WRITE`.
6. On the Database Cloud Service compute node, prepare the database for the tablespace import.
7. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the database.
8. On the Database Cloud Service compute node, set the tablespaces in the database to `READ WRITE` mode.
9. After verifying that the data has been imported successfully, you can delete the dump file.

RMAN CONVERT Transportable Tablespace with Data Pump: Example

This example provides a step-by-step demonstration of the tasks required to migrate tablespaces in an on-premises Oracle database to an Oracle Database Cloud Service database.

In this example, the on-premises database is on a Linux host.

1. On the on-premises database host, prepare the database for the Data Pump transportable tablespace export.

- a. On the on-premises database host, create a directory in the operating system to use for the on-premises export.


```
$ mkdir /u01/app/oracle/admin/orcl/dpdump/for_cloud
```
- b. On the on-premises database host, invoke SQL*Plus and log in to the on-premises database as the SYSTEM user.


```
$ sqlplus system
Enter password: <enter the password for the SYSTEM user>
```
- c. Create a directory object in the on-premises database to reference the operating system directory.


```
SQL> CREATE DIRECTORY dp_for_cloud AS '/u01/app/oracle/admin/orcl/dpdump/for_cloud';
```
- d. On the on-premises database host, set all tablespaces that will be transported (the transportable set) to READ ONLY mode.


```
SQL> ALTER TABLESPACE fsindex READ ONLY;
Tablespace altered.
SQL> ALTER TABLESPACE fsdata READ ONLY;
Tablespace altered.
```
- e. Exit from SQL*Plus.

2. On the on-premises database host, invoke Data Pump Export to perform the transportable tablespace export.

On the on-premises database host, invoke Data Pump Export and connect to the on-premises database. Export the on-premises 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
```
3. On the on-premises database host, invoke RMAN and use the CONVERT TABLESPACE command to convert the tablespace datafile to the Oracle Database Cloud platform format.
 - a. Invoke RMAN.


```
$ rman target /
```
 - b. Execute the RMAN CONVERT TABLESPACE command to convert the datafiles and store the converted files in a temporary location on the on-premises database host.


```
RMAN> CONVERT TABLESPACE fsdata, fsindex
2> TO PLATFORM 'Linux x86 64-bit'
3> FORMAT '/tmp/%U';
...
input datafile file number=00006 name=/u01/app/oracle/oradata/orcl/
fsdata01.dbf
converted datafile=/tmp/data_D-ORCL_I-1410251631_TS-FSDATA_FNO-6_0a9c9un3
...
input datafile file number=00007 name=/u01/app/oracle/oradata/orcl/
fsindex01.dbf
converted datafile=/tmp/data_D-ORCL_I-1410251631_TS-FSINDEX_FNO-7_0b9c9un6
...
```

- c. Take note of the names of the converted files. You will copy these files to the Database Cloud Service compute node in the next step.
 - d. Exit RMAN.
4. Use a secure copy utility to transfer the Data Pump Export dump file and the converted tablespace datafiles to the Database Cloud Service compute node.

In this example the dump file is copied to the /u01 directory. Choose the appropriate location based on the size of the file that will be transferred.

- a. On the Database Cloud Service compute node, create a directory for the dump file.

```
$ mkdir /u01/app/oracle/admin/ORCL/dpdump/from_onprem
```

- b. Before using the scp command to copy files, make sure the SSH private key that provides access to the Database Cloud Service compute node is available on your on-premises host.

- c. On the on-premises database host, use the scp utility to transfer the dump file and all datafiles of the transportable set to the Database Cloud Service compute node.

```
$ scp -i private_key_file \
/u01/app/oracle/admin/orcl/dpdump/for_cloud/expdat.dmp \
oracle@IP_address_DBaaS_VM:/u01/app/oracle/admin/ORCL/dpdump/from_onprem

$ scp -i private_key_file \
/tmp/data_D-ORCL_I-1410251631_TS-FSDATA_FNO-6_0aqc9un3 \
oracle@IP_address_DBaaS_VM:/u02/app/oracle/oradata/ORCL/fsdata01.dbf

$ scp -i private_key_file \
/tmp/data_D-ORCL_I-1410251631_TS-FSINDEX_FNO-7_0bqc9un6 \
oracle@IP_address_DBaaS_VM:/u02/app/oracle/oradata/ORCL/fsindex01.dbf
```

5. Set the on-premises tablespaces back to READ WRITE.

- a. Invoke SQL*Plus and log in as the SYSTEM user.
- b. Set the FSDATA and FSINDEX tablespaces back to READ WRITE mode.

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

- c. Exit from SQL*Plus.

6. On the Database Cloud Service compute node, prepare the database for the tablespace import.
- a. On the Database Cloud Service compute node, invoke SQL*Plus and log in to the database as the SYSTEM user.
- b. Create a directory object in the Database Cloud Service database.

```
SQL> CREATE DIRECTORY dp_from_onprem AS '/u01/app/oracle/admin/ORCL/dpdump/
from_onprem';
```

- c. If the owners of the objects that will be imported do not exist in the database, create them before performing the import. The transportable tablespace mode of import does not create the users.

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

- 7. On the Database Cloud Service compute node, invoke Data Pump Import and connect to the database.

Import the data into the DBaaS database using the TRANSPORT_DATAFILES option

```
$ impdp system DIRECTORY=dp_from_onprem \
TRANSPORT_DATAFILES='/u02/app/oracle/oradata/ORCL/fsdata01.dbf', \
'/u02/app/oracle/oradata/ORCL/fsindex01.dbf'
```

- 8. On the Database Cloud Service compute node, set the tablespaces in the database to READ WRITE mode.
 - a. Invoke SQL*Plus and log in as the SYSTEM user.
 - b. 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.
```

- c. Exit from SQL*Plus.

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

SQL Developer and INSERT Statements to Migrate Selected Objects

You can use SQL Developer to create a cart into which you add selected objects to be loaded into an Oracle Database 12c database on Oracle Database Cloud Service.

In this method, you use SQL INSERT statements to load the data into your cloud database.

To migrate selected objects to an Oracle Database 12c database on a Database Cloud Service deployment using SQL Developer and INSERT statements, you perform these tasks:

1. Launch SQL Developer, connect to your on-premises database and create a cart containing the objects you want to migrate.
2. In SQL Developer, click the Export Cart icon and select “Insert” in the Format menu.
3. In SQL Developer, open a connection to the Oracle Database 12c database on Database Cloud Service and execute the generated script to create the database objects.

4. In SQL Developer, open a connection to the Oracle Database 12c database on Database Cloud Service and run the generated script to create the objects and load the data.

SQL Developer and SQL*Loader to Migrate Selected Objects

You can use SQL Developer to create a cart into which you add selected objects to be loaded into an Oracle Database 12c database on Oracle Database Cloud Service.

In this method, you use SQL*Loader to load the data into your cloud database.

To migrate selected objects to an Oracle Database 12c database on a Database Cloud Service deployment using SQL Developer and SQL*Loader, you perform these tasks:

1. Launch SQL Developer, connect to your on-premises database and create a cart containing the objects you want to load into your cloud database.
2. In SQL Developer, click the Export Cart icon and select “loader” in the Format menu.
3. In SQL Developer, open a connection to the Oracle Database 12c database on Database Cloud Service and execute the generated script to create the database objects.
4. Use a secure copy utility to transfer the SQL*Loader control files and the SQL*Loader data files to the Database Cloud Service compute node.
5. On the Database Cloud Service compute node, invoke SQL*Loader to load the data using the SQL*Loader control files and data files for each object.

Unplugging/Plugging a PDB

You can use this method only if the on-premises platform is little endian, and the on-premises database and Database Cloud Service database have compatible database character sets and national character sets.

You can use the unplug/plug method to migrate an Oracle Database 12c PDB to a PDB in an Oracle Database 12c database on an Oracle Database Cloud Service database deployment.

To migrate an Oracle Database 12c PDB to a PDB in the Oracle Database 12c database on a Database Cloud Service database deployment using the plug/unplug method, you perform these tasks:

1. On the on-premises database host, invoke SQL*Plus and close the on-premises PDB.
2. On the on-premises database host, execute the `ALTER PLUGGABLE DATABASE UNPLUG` command to generate an XML file containing the list of datafiles that will be plugged in to the database on Database Cloud Service.
3. Use a secure copy utility to transfer the XML file and the datafiles to the Database Cloud Service compute node.
4. On the Database Cloud Service compute node, invoke SQL*Plus and execute the `CREATE PLUGGABLE DATABASE` command to plug the database into the CDB.
5. On the Database Cloud Service compute node, open the new PDB by executing the `ALTER PLUGGABLE DATABASE OPEN` command.

For more information, see "Creating a PDB by Plugging an Unplugged PDB into a CDB" in *Oracle Database Administrator's Guide* for Release [12.2](#) or [12.1](#).

Unplugging/Plugging Non-CDB

You can use this method only if the on-premises platform is little endian, and the on-premises database and Database Cloud Service database have compatible database character sets and national character sets.

You can use the unplug/plug method to migrate an Oracle Database 12c non-CDB database to a PDB in an Oracle Database 12c database on an Oracle Database Cloud Service database deployment. This method provides a way to consolidate several non-CDB databases into a single Oracle Database 12c multitenant database on Database Cloud Service.

To migrate an Oracle Database 12c non-CDB database to the Oracle Database 12c database on a Database Cloud Service database deployment using the plug/unplug method, you perform these tasks:

1. On the on-premises database host, invoke SQL*Plus and set the on-premises database to READ ONLY mode.
2. On the on-premises database host, execute the `DBMS_PDB.DESCRIBE` procedure to generate an XML file containing the list of datafiles that will be plugged in on the cloud database.
3. Use a secure copy utility to transfer the XML file and the datafiles to the Database Cloud Service compute node.
4. On the Database Cloud Service compute node, invoke SQL*Plus and execute the `CREATE PLUGGABLE DATABASE` command to plug the database into the CDB.
5. On the Database Cloud Service compute node, execute the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` script to delete unnecessary metadata from the `SYSTEM` tablespace of the new PDB.
6. On the Database Cloud Service compute node, open the new PDB by executing the `ALTER PLUGGABLE DATABASE OPEN` command.
7. Optionally, on the on-premises database host invoke SQL*Plus and set the on-premises database back to READ WRITE mode.

For more information, see "Creating a PDB Using a Non-CDB" in *Oracle Database Administrator's Guide* for Release [12.2](#) or [12.1](#).

Using Oracle DBaaS Monitor

Oracle DBaaS Monitor provides monitoring and management of the Oracle database and listener on Oracle Database Cloud Service.

Note:

This section does not apply to database deployments that use Oracle Real Application Clusters. Such deployments do not currently include Oracle DBaaS Monitor.

Topics

- [About Oracle DBaaS Monitor](#)
- [Accessing Oracle DBaaS Monitor](#)
- [Filtering the Display on DBaaS Monitor Pages](#)
- [Administering the Listener](#)
- [Starting and Stopping the Database Instance](#)
- [Viewing and Modifying Initialization Parameters](#)
- [Viewing Tablespace and Segment Space Usage](#)
- [Viewing Alert Log Entries and Checking for Errors](#)
- [Viewing Real Time SQL Monitor](#)
- [Administering Pluggable Databases](#)

About Oracle DBaaS Monitor

Oracle DBaaS Monitor provides monitoring and management of the Oracle database and listener on Oracle Database Cloud Service.

DBaaS Monitor provides quick and easy access to a variety of information about the database instance running on a database deployment:

- Overall, how much storage is allocated to tablespaces, and how much of that storage is used
- For each tablespace, how much storage is allocated and how much of that storage is used, with additional drill-down capabilities to view segments
- A real-time graph showing wait events across several selectable categories

- The alert log, with log searching capabilities
- A list of open user sessions, with drill-down capabilities to view session details such as the last SQL statement, explain plan, waits, contention, and so on
- A list of initialization parameters, with the ability to change parameter values, both in memory and in the SPFILE.
- Indication of whether certain database options are enabled
- Monitoring of current and past SQL Developer PDB uploads
- A list of the SQL statements that are being monitored in the database, with real time display of details such as the status, duration, degree of parallelism, and so on

You can use DBaaS Monitor to view information about the compute node:

- CPU utilization information in an interactive table format, with automatic refresh intervals
- OS process information, with filtering and automatic refresh capabilities

DBaaS Monitor also provides the following management capabilities:

- Start up and shut down the database instance
- Open and close a pluggable database
- Create and drop a pluggable database
- Plug in and unplug a pluggable database
- Clone a pluggable database
- Start and stop the listener

Accessing Oracle DBaaS Monitor

Database deployments on Oracle Database Cloud Service include Oracle DBaaS Monitor, a built-in monitor that provides a wide spectrum of information about Oracle Database status and resource usage.

To access Oracle DBaaS Monitor:

1. Open the Oracle Database Cloud Service console.
For detailed instructions, see [Accessing the Oracle Database Cloud Service Console](#).
2. From the  menu for the deployment, select **Open DBaaS Monitor Console**.
A window prompting you for your user name and password is displayed.
3. Enter **dbaas_monitor** as the user name and the password specified during the database deployment creation process, and then click **OK**.

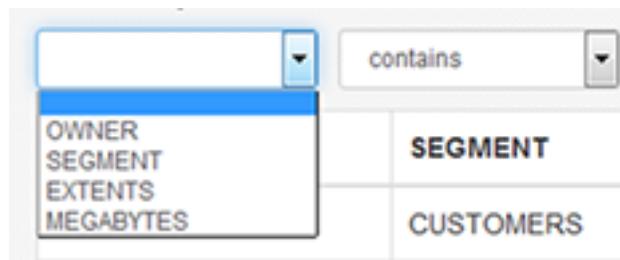
Filtering the Display on DBaaS Monitor Pages

On DBaaS Monitor pages, the PDB selector enables you to limit the display of information in Oracle 12c databases. You can also filter what items display in tabular lists.

For Oracle 12c databases, you can use the PDB selector to limit displayed information. The default selection, Overall, results in no filtering; information for the container database and all PDBs is shown. The CDB\$ROOT selection shows only tablespace information for the tablespaces in the container database root. The PDB\$SEED selection shows only tablespace information for the tablespaces in the pluggable database seed. The PDB selector also lets you limit displayed information to a specific pluggable database.

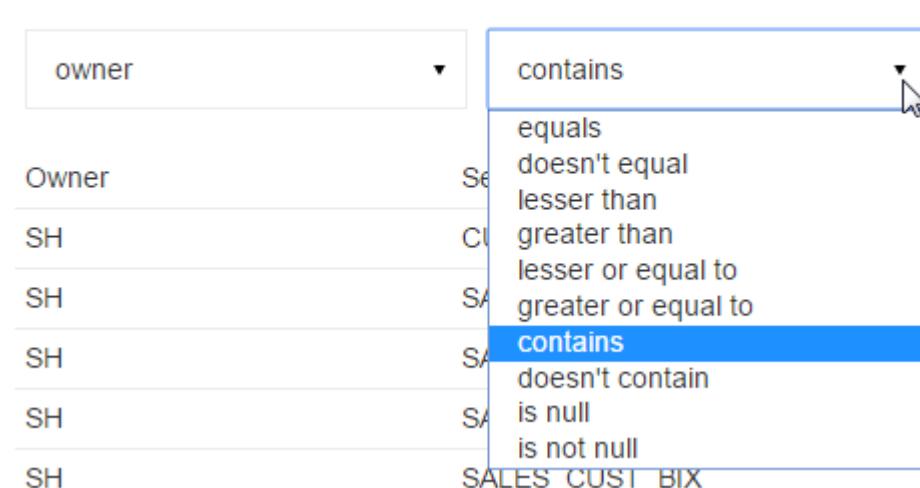
On DBaaS Monitor pages that show tabular lists, you can filter what items the table displays.

1. Expand the first drop-down menu and select the column you want to use as the filter.



2. In the second drop-down menu, select the operator.

Segments



3. In the box, enter the value and click the search icon.



Administering the Listener

You can use DBaaS Monitor to administer the listener, including troubleshooting unknown service name errors.

Topics

- [Viewing Listener Status Information](#)
- [Starting the Listener](#)
- [Stopping the Listener](#)
- [Verifying that the Listener Knows of a Service](#)

Viewing Listener Status Information

The Oracle Net Listener (the listener) is a process that resides on the server. It listens for incoming client connection requests and manages traffic to the server.

To review listener status information:

- On the DBaaS Monitor home page, click **Listener**. Or in the Database drop-down menu, click **Listener**.
The RDBMS Listener page displays.

The RDBMS Listener page shows the following information:

- Status of the listener, including start time
- Protocol addresses the listener is configured to listen on
- Summary of the database services registered with the listener and the service handlers allocated to each service
- Registered database services (service name), database instance associated with the service, and connection status

Starting the Listener

By default, an Oracle Net listener is automatically started in your database deployment. If you have shut it down or it has crashed, you can use DBaaS Monitor to start the listener by performing the following steps:

1. On the DBaaS Monitor home page, click **Listener**. Or in the Database drop-down menu, click **Listener**.
2. In the Listener page menu, click **Turn on** to start the listener.

Stopping the Listener

To stop the listener:

1. On the DBaaS Monitor home page, click **Listener**. Or in the Database drop-down menu, click **Listener**.
2. Click **Turn off** to stop the listener.

Verifying that the Listener Knows of a Service

An "ORA-12514: Listener Does Not Currently Know of Service Requested in Connect Descriptor" error indicates that the specified service name is unknown by the Oracle Net Listener process. You can verify that the specified service name is configured with the listener by performing the following steps:

1. On the DBaaS Monitor home page, click **Listener**. Or in the Database drop-down menu, click **Listener**.

The RDBMS Listener page displays.

2. Review the information displayed to determine whether the requested service name is listed.

Starting and Stopping the Database Instance

You can use DBaaS Monitor to start up and shut down your database instance.

Topics

- [Starting the Database Instance](#)
- [Stopping the Database Instance](#)

Starting the Database Instance

You can use DBaaS Monitor to start the database instance in OPEN mode. In this mode the instance is started, the database is mounted and then opened.

If you need to start the database instance in any other mode (NOMOUNT, MOUNT, or FORCE), use SQL*Plus instead of DBaaS Monitor. For more information, see "Starting Up and Shutting Down" in *Oracle Database Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#).

To start the database instance in OPEN mode:

1. On the DBaaS Monitor home page, click Database Status. Or, in the Database pull-down menu, click **Manage**.
2. In the menu for your database, click **Start database**.

The database instance is started.

Stopping the Database Instance

You can use DBaaS Monitor to shut down the database instance in IMMEDIATE mode. In this mode, no new connections are allowed. No new transactions are allowed to be started and any uncommitted transactions are rolled back.

If you need to shut down the database instance in any other mode (ABORT, NORMAL, or TRANSACTIONAL), use SQL*Plus instead of DBaaS Monitor. For more information, see "Starting Up and Shutting Down" in *Oracle Database Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#).

To shut down the database instance in IMMEDIATE mode:

1. On the DBaaS Monitor home page, click Database Status. Or, in the Database pull-down menu, click **Manage**.
2. In the menu for your database, click **Stop database**.

The database instance is shut down.

Viewing and Modifying Initialization Parameters

Initialization parameters configure the database instance, including memory structures, and define locations for database files.

Values for initialization parameters are stored in a text-based initialization parameter file (PFILE) or binary server parameter file (SPFILE). The initialization parameter file is read at database instance startup. For more information, see "Initialization Parameters" in *Oracle Database Reference* for Release 12.2, 12.1 or 11.2.

To view and modify the value of an initialization parameter:

1. From any DBaaS Monitor page, click Parameters from the Database drop-down menu.
2. On the Parameters page, use the arrow buttons in the lower right corner to navigate the multiple pages of parameters, or use the search fields at the top of the page.
3. To perform a search, enter values in the search criteria columns and click the search icon to locate the initialization parameter.
4. Enter the new value in the Value field.
5. Expand the Save changes menu and select the appropriate value:
 - To Memory: Updates the value for the existing database instance, but does not save it to the SPFILE.
 - To SPFILE: Updates the value in the SPFILE, but does not change it in the existing instance. The new value will take effect when the instance is restarted.
 - To both: Updates the values for the existing database instance and updates the value in the SPFILE.
6. Click **Yes** to confirm your change.
7. Click **OK** to close the Results page.

Viewing Tablespace and Segment Space Usage

You can use DBaaS Monitor to view tablespace and segment space usage.

A tablespace is a database storage unit that groups related logical structures together. A tablespace is comprised of datafiles. A segment is a set of extents allocated from a tablespace for a specific database object such as a table or index.

To view space usage information

1. On the DBaaS Monitor home page, click **Online Database Storage**. Alternatively, from any DBaaS Monitor page, click **Storage** from the Database drop-down menu.

The Storage page displays. For an Oracle Database 12c database, the Storage page shows the used and allocated storage space for all tablespaces in the container in

the root, and the used and allocated storage space for tablespaces in any pluggable databases. For an Oracle Database 11g database, the Storage page shows the used and allocated space for the entire database.

2. For an Oracle Database 12c database, you can click **show tablespaces** for the container database, or click **show tablespaces** for each pluggable database.
3. When you click **show tablespaces**, a list of tablespaces appears. You can click a tablespace to view its storage information. An interactive report appears, showing the segments that exist within the tablespace. Most segments are user objects, and they include tables, LOBs, and indexes.
4. On the Segments page, you can refine the list of segments shown by using the filter feature.

For example, you can search for all the segments for a specific owner (schema) by selecting OWNER from the first drop-down list, entering the owner (schema) name in the box, and clicking the search icon.

Viewing Alert Log Entries and Checking for Errors

You can use DBaaS Monitor to review the alert log periodically to verify that your database system is operating normally.

The alert log is a chronological log of messages including the following:

- Nondefault initialization parameters used at startup
- Administrative operations, such as STARTUP, SHUTDOWN, ARCHIVE LOG, RECOVER, and CREATE/ALTER/ DROP DATABASE/TABLESPACE
- Messages and errors relating to the functions of certain background processes, such as LGWR
- Internal errors (ORA-600), block corruption errors (ORA-1578), and deadlock errors (ORA-60)

To view alert log entries:

1. On any DBaaS Monitor page, click **Alerts** from the Database drop-down menu.
2. Search for a specific value by selecting the display column in the first drop-down list, selecting the condition in the second drop-down list, and entering your search value in the box. Then, click the search icon.

The Alerts box on the DBaaS Monitor Home page has two sections, each of which can be clicked for further detail. One section indicates the total number of all messages. The other section indicates how many errors have been recorded in the alert log in the last 7 days. To view errors in the alert log:

1. On the DBaaS Monitor home page, click **ERRORS** in the Alerts box.
2. On the Alerts page select **Type** from the first drop-down list and enter your search criteria in the third field. Then, click the search icon.

Errors recorded in the alert log are displayed.

Viewing Real Time SQL Monitor

This page shows, in real time, the SQL statements that are being monitored in the database.

To review the SQL statements being monitored:

- On the DBaaS Monitor home page, in the Database drop-down menu, click **Real Time SQL Monitor**.

The Real Time SQL Monitor page displays.

This tool helps identify run-time issues for SQL statements and monitor their behavior, by providing two major functions:

- General view of monitored statements
- View of SQL execution details

General View of Monitored Statements

The page contains a table of SQL statements currently running. This table shows the following information:

- **Status** — Current state of the SQL statement execution. For example, a SQL statement that has already finished its execution will show a status of "DONE".
- **Duration** — This is the amount of time a SQL statement is taking, or has taken, to execute.
- **SQL ID** — SQL identifier of the statement being monitored.
- **Session ID** — Session identifier that is executing, or has executed, the SQL statement.
- **Session Serial Number** — Used to uniquely identify a session's objects.
- **Instance Degree of Parallelism** — This Degree of Parallelism (DOP) column shows how many instances and parallel execution servers are allocated. It is shown in the form of "number of instances" | "number of parallel servers"
- **CPU Time** — This is the CPU time consumed by the execution of the query.
- **I/O Time** — This is the I/O time consumed by the execution of the query.
- **Start Time** — This is the time in which the execution of the SQL statement started.
- **SQL Statement** — This is the SQL statement being monitored.

For more information, see "Monitoring the Database" in *Oracle Database Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#).

View of SQL Execution Details

When a SQL statement is drilled down from the main monitor table, a detailed view is shown. The SQL ID, Start Time and the SQL Execution ID represent the execution key that uniquely identify this SQL statement. A detail view consists of the general characteristics that integrate the execution of a SQL statement.

General information about the query execution is provided:

- **Execution Plan** — Degree of Parallelism of the SQL statement
- **Execution Started** — Time that the SQL statement execution started
- **Last Refresh Time:** — Last update time of the SQL monitor registry for the SQL statement
- **Execution ID** — Execution identifier
- **User** — User in the format USER@CONTAINER
- **Fetch Calls** — Number of fetch calls done by the SQL statement

General statistics of the SQL statement are provided: total duration of execution, the number of buffered gets, number of Input/Output requests and bytes.

A duration breakdown shows the percentage of the total duration of the execution of the SQL statement, divided into two types of times:

- **Database Time** — Any time of execution related to the database needs; for example, CPU Time
- **Wait Time** — The waiting time that the statement goes through to complete an execution

Each duration time can be drilled down by clicking on the "Show Detail" element, where a gauge graph shows the percentage of time that integrates each type of duration time.

PLSQL and Java Time are provided. These time measurements are outside of the duration of the SQL statement.

Detailed information of the statement — This space holds the information corresponding to the explain plan, parallel behaviour and CPU activity involved in the execution of the statement:

- **SQL Text** — SQL statement that was or is being executed.
- **Plan Statistics** — Explain plan of the execution of the SQL statement in the form of a table. Each row is a different operation involved in the execution of the SQL statement and it shows hierarchy dependency by adding a space at the beginning of the text in the Operation column.
 - Operation, Name, Estimated Rows, Cost, Actual Rows, Memory, Temp(Max), IO Requests, IO Bytes
- **Graphic view of the Plan Statistics** — The plan statistics table in a graphic representation; the hierarchy is presented as a collapsible tree map in which each node represent an operation.
- **Parallelism Details for the SQL statement** — Each execution consists of a parallel coordinator and one or more parallel sets. Each set can have one or more processes. When a row has dependents, each of its columns will be the sum of the values of its dependants. When this happens, a sigma symbol will appear to show that a value consists of the sum of others. The columns shown are the following:
 - Process Name, Buffer Gets, CPU Time, Elapsed Time, Other Wait Time, Server Set, Read Requests, Read Bytes
- **Activity Line Chart for the CPU Usage** — Line chart showing the number of different types of CPU activities registered in the execution of the SQL statement.

The Y axis represents the number of CPU activities and the X axis represents the time registered for that activity. Each activity is represented by a custom set of colors depending on the activity.

Administering Pluggable Databases

You can use DBaaS Monitor to create and drop a pluggable database, open and close a pluggable database, plug in and unplug a pluggable database, and clone a pluggable database.

Topics

- [Cloning a Pluggable Database](#)
- [Closing a Pluggable Database](#)
- [Creating a Pluggable Database](#)
- [Dropping a Pluggable Database](#)
- [Opening a Pluggable Database](#)
- [Plugging In a Pluggable Database](#)
- [Unplugging a Pluggable Database](#)

Cloning a Pluggable Database

You can use DBaaS Monitor to clone a pluggable database.

1. On the DBaaS Monitor home page, click **Database Status**. Or, in the Database pull-down menu, click **Manage**.
2. In the menu for your pluggable database, click **Clone**.
3. Complete the Clone PDB dialog and click **OK**.
 - **New PDB Name**—Provide a name for the new PDB.
 - **Source PDB**—This field is pre-filled with the name of the PDB you chose as the source for cloning.
 - **File Name Conversions**— You can optionally provide custom names and expressions for the PDB datafiles.
 - **Unlimited Storage and Reuse Temp File**— By default, unlimited storage and reuse temp file are selected for the PDB.
 - **Clone TDE Key**—TDE Key cloning is selected by default.
 - **Keystore Password**—If Clone TDE Key is selected, the Keystore Password field is made visible. Use the password that was specified during the database deployment creation process.

Closing a Pluggable Database

You can use DBaaS Monitor to close a pluggable database.

1. On the DBaaS Monitor home page, click **Database Status**. Or, in the Database pull-down menu, click **Manage**.

2. In the menu for your pluggable database, click **Modify state**.
3. The Modify PDB dialog appears. In the State field, click **CLOSE**.
4. In the Option field, the default is IMMEDIATE. Choose IMMEDIATE or **NORMAL**.
5. Click **OK**.

Creating a Pluggable Database

You can use DBaaS Monitor to create a pluggable database.

1. On the DBaaS Monitor home page, click **Database Status**. Or, in the Database pull-down menu, click **Manage**.
2. Click **Create PDB**.
3. Complete the Create PDB dialog and click **OK**.
 - **New PDB Name**—Provide a name for the new pluggable database.
 - **Admin Username** and **Admin Password**—Provide a user name and password for the PDB administrator. You need not use the same admin password that you used during the database deployment creation process, unless you wish.
 - **File Name Conversions**—You can optionally provide custom names and expressions for the PDB datafiles.
 - **Unlimited Storage** and **Reuse Temp File**—By default, unlimited storage and reuse temp file are selected for the PDB.
 - **Create TDE Key**—TDE Key creation is selected by default.
 - **Keystore Password**—If Create TDE Key is selected, the Keystore Password field is made visible. Use the password that was specified during the database deployment creation process.

Dropping a Pluggable Database

You can use DBaaS Monitor to drop a pluggable database.

1. On the DBaaS Monitor home page, click **Database Status**. Or, in the Database pull-down menu, click **Manage**.
2. In the menu for your PDB, click **Drop**.
3. In the Drop PDB dialog, choose whether you want to keep or delete the PDB's datafiles. Then click **OK**.

Opening a Pluggable Database

You can use DBaaS Monitor to open a pluggable database.

1. On the DBaaS Monitor home page, click **Database Status**. Or, in the Database pull-down menu, click **Manage**.
2. In the menu for your pluggable database, click **Modify state**.

3. The Modify PDB dialog appears. In the State field, click **OPEN**.
4. In the Option field, the default is **READ WRITE**. Choose **READ WRITE**, **READ ONLY** or **RESTRICTED**.
5. Click **OK**.

Plugging In a Pluggable Database

You can use DBaaS Monitor to plug in a pluggable database.

1. On the DBaaS Monitor home page, click **Database Status**. Or, in the Database pull-down menu, click **Manage**.
2. Click **Plug in PDB**.
3. Complete the Plug PDB dialog and click **OK**.
 - **New PDB Name**—Provide a name for the pluggable database.
 - **Plug as Clone**—This field is selected by default. If you are plugging in a PDB as a clone, fill the XML Filename field.
 - **Source File Name Conversion**—You can optionally provide custom expressions for the PDB datafiles.
 - **Copy Action**—The default value is **Don't Copy**. Choose **Don't Copy**, **Copy**, or **Move**.
 - **Unlimited Storage and Reuse Temp File**—By default, unlimited storage and reuse temp file are selected for the PDB.
 - **Import TDE Key**—TDE Key import is selected by default.
 - **Keystore Password, Import From, Key Secret**—If Import TDE Key is selected, the Keystore Password, Import From, and Key Secret fields are made visible. For Keystore Password, use the password that was specified during the database deployment creation process.

Unplugging a Pluggable Database

You can use DBaaS Monitor to unplug a pluggable database.

1. On the DBaaS Monitor home page, click **Database Status**. Or, in the Database pull-down menu, click **Manage**.
2. In the menu for your pluggable database, click **Unplug**.
3. Complete the Unplug PDB dialog and click **OK**.
 - **PDB**—This field is pre-filled with the name of the pluggable database you selected.
 - **XML Filename**—This field is pre-filled with the path to the PDB's XML file.
 - **Export TDE Key**—This field is selected by default.
 - **Keystore Password, Export To, Key Secret**—If Export TDE Key is selected, the Keystore Password, Export To, and Key Secret fields are made visible. For

Keystore Password, use the password that was specified during the database deployment creation process.

Frequently Asked Questions for Database Cloud Service

To see a list of frequently asked questions for Oracle Database Cloud Service, see the [FAQ](#) page for Oracle Database Cloud at cloud.oracle.com.

Troubleshooting Database Cloud Service

This section describes common problems that you might encounter when using Oracle Database Cloud Service and explains how to solve them.

Topics

- [Problems Creating Deployments](#)
 - I cannot create a deployment when I have many database deployments
 - I cannot create a deployment, even after waiting for an hour
- [Problems Administering Deployments](#)
 - I am required to change the password for the oracle user when I try to connect to a compute node
 - I get a Linux error 30, Read-only file system, when trying to connect to or work in my environment
- [Problems with Scaling](#)
 - My scaling operation does not start
 - My deployment is too busy to allow scaling
- [Problems with Patching and Rollback](#)
 - I receive a message stating that the virtual machines are unhealthy
 - I receive a message stating that the instance is busy with another operation
 - I cannot apply a patch due to a lack of storage space
 - My attempt to roll back the January 2015 Patch Set Update (Jan 2015 PSU) fails
 - My attempt to roll back the April 2015 Patch Set Update (Apr 2015 PSU) fails
- [Problems with Backing Up and Restoring](#)
 - There is not enough space for my backup

Problems Creating Deployments

The following solutions apply to problems with creating database deployments on Oracle Database Cloud Service.

- [I cannot create a deployment when I have many database deployments](#)

- [I cannot create a deployment, even after waiting for an hour](#)

I cannot create a deployment when I have many database deployments

Your account might not have enough compute quota to create the deployment.

If you have database deployments you do not need, delete them. If you need all your database deployments, contact Oracle Sales and Services to buy more quota for your account.

I cannot create a deployment, even after waiting for an hour

If deployment creation fails after one hour, the system might be experiencing a heavy load, and resources are not yet available.

Wait before you try again to create the deployment. If this doesn't work, contact Oracle Support.

Problems Administering Deployments

The following solutions apply to problems with administering database deployments on Oracle Database Cloud Service.

- [I am required to change the password for the oracle user when I try to connect to a compute node](#)
- [I get a Linux error 30, Read-only file system, when trying to connect to or work in my environment](#)

I am required to change the password for the oracle user when I try to connect to a compute node

You cannot change the password as required because the `oracle` user does not have a password. Instead, change the properties of the `oracle` user so that its password does not expire:

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. Change the expiration properties of the `oracle` user by using the `chage` command:

```
# /usr/bin/chage -I -1 -m 0 -M 99999 -E -1 oracle
```

4. Confirm that the expiration properties have been changed by using the `chage` command again:

```
# /usr/bin/chage -l oracle  
Last password change : date  
Password expires : never  
Password inactive : never  
Account expires : never  
Minimum number of days between password change : 0
```

```
Maximum number of days between password change      : 99999
Number of days of warning before password expires : 7
```

5. Close your connection to the compute node.

I get a Linux error 30, Read-only file system, when trying to connect to or work in my environment

In certain rare cases, Oracle Compute Cloud Service sets the access of storage volumes attached to a Database Cloud Service deployment to read-only. When this situation arises, you can restore read-write access by restarting the compute node, as described in [Rebooting a Compute Node](#).

Problems with Scaling

The following solutions apply to problems with scaling an Oracle Database Cloud Service environment.

- [My scaling operation does not start](#)
- [My deployment is too busy to allow scaling](#)

My scaling operation does not start

The system is overloaded with requests.

Wait before you try to scale again. If that doesn't work, contact Oracle Support.

My deployment is too busy to allow scaling

Your database deployment has a pending maintenance operation such as backup or patching.

Wait until maintenance has completed before you try scaling again.

Problems with Patching and Rollback

The following solutions apply to problems with patching and rollback operations on Oracle Database Cloud Service.

- [I receive a message stating that the virtual machines are unhealthy](#)
- [I receive a message stating that the instance is busy with another operation](#)
- [I cannot apply a patch due to a lack of storage space](#)
- [My attempt to roll back the January 2015 Patch Set Update \(Jan 2015 PSU\) fails](#)
- [My attempt to roll back the April 2015 Patch Set Update \(Apr 2015 PSU\) fails](#)

I receive a message stating that the virtual machines are unhealthy

You cannot apply a patch if the compute nodes are not in a healthy state.

Restore the deployment using a backup and try patching again.

I receive a message stating that the instance is busy with another operation

You cannot apply a patch when the deployment is under maintenance, for example, scaling or backup.

Wait until the deployment is no longer under maintenance and try patching again.

I cannot apply a patch due to a lack of storage space

When you apply a patch, storage space is required for temporary files that are created and used during the patching operation. If you receive a message indicating that you don't have sufficient space to patch, either when applying the patch or checking its prerequisites, take these steps:

1. Check whether you might actually have enough space.

The patching tools check whether you have 15 GB free space on /u01, but the space actually required for temporary use during patching is somewhat less than 15 GB. To check whether you might have enough space:

- a. Connect as the **opc** user to the compute node .

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

- b. Display the mounted filesystems:

```
$ df -hT
```

- c. Locate the row for /u01 and check its space available, as reported in the **Avail** column.

- d. If the space available is 13 GB or greater, you have enough space to patch and you can continue to Step e. If you do not have enough space, close your connection and go to Step 2.

- e. Start a root-user command shell:

```
$ sudo -s
#
```

- f. Navigate to the /var/opt/oracle/patch directory:

```
# cd /var/opt/oracle/patch
```

- g. Use an editor such as vim to change the value of the **ignore_space_less_than_15g** key in the **dbpatchm.cfg** file from 0 (zero) to 1 (one):

```
ignore_space_less_than_15g=1
```

- h. Apply the patch according to the instructions in [Applying a Patch](#).

2. Add temporary storage for the temporary files.

If you do not have enough space on /u01, you can add temporary storage to the compute node, apply the patch, and then remove the temporary storage:

- a. Add 20 GB of temporary storage to the compute node by following these instructions: [Adding Temporary Storage to a Database Deployment](#). When following these instructions, use /patchtemp as the mount point directory.

- b. Connect to the compute node as the **opc** user.

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

c. Start a root-user command shell:

```
$ sudo -s
#
```

d. Navigate to the `/var/opt/oracle/patch` directory:

```
# cd /var/opt/oracle/patch
```

e. Use an editor such as vim to edit the `dbpatchm.cfg` file:

- Change the value of the `temporary_space` key from `"/u01/download"` to `"/patchtemp/download"`
- Change the value of the `ignore_space_less_than_15g` key from 0 (zero) to 1 (one)

For example:

```
temporary_space="/patchtemp/download";
...
ignore_space_less_than_15g=1
```

f. Apply the patch according to the instructions in [Applying a Patch](#).

g. Remove the 20 GB of temporary storage you added to the compute node by following these instructions: [Deleting Temporary Storage from a Database Deployment](#)

My attempt to roll back the January 2015 Patch Set Update (Jan 2015 PSU) fails

The January 2015 Patch Set Update includes overlay patches and so requires you to take special steps in order to roll it back. To roll back this patch set update, follow the instructions in [Rolling Back a Patch or Failed Patch by Using the dbpatchm Subcommand](#), which includes these special steps.

My attempt to roll back the April 2015 Patch Set Update (Apr 2015 PSU) fails

The April 2015 Patch Set Update includes overlay patches and so requires you to take special steps in order to roll it back. To roll back this patch set update, follow the instructions in [Rolling Back a Patch or Failed Patch by Using the dbpatchm Subcommand](#), which includes these special steps.

Problems with Backing Up and Restoring

The following solutions apply to problems with backup and restore operations on Oracle Database Cloud Service.

- [There is not enough space for my backup](#)

There is not enough space for my backup

The backup storage area does not have enough space for the backup operation to create the archive.

Do one of the following:

- Delete any unwanted backups.
- Archive one or more backups to another location.

Characteristics of a Newly Created Deployment

This section provides information about the content and configuration of a newly created database deployment on Oracle Database Cloud Service.

Topics

- [Characteristics of an Oracle Database Cloud Deployment](#)
- [Characteristics of an Oracle Database Cloud Service Deployment Using Oracle RAC](#)

Characteristics of an Oracle Database Cloud Deployment

This section provides information about the content and configuration of a newly created database deployment on Oracle Database Cloud Service using the Oracle Database Cloud Service service level.

When you create a database deployment using the Oracle Database Cloud Service service level and choose the Single Instance database type, Database Cloud Service creates a single-instance Oracle database. The database is housed on a compute node that Database Cloud Service creates using Oracle Compute Cloud Service resources. In brief, Database Cloud Service:

- Creates a compute node that has its own IP address with a name of this form:
deployment-name db 1
This Compute Cloud Service instance uses the compute shape specified during the database deployment creation process.
- Installs Oracle Linux 6.6, Oracle Database 12.2.0.1, 12.1.0.2 or 11.2.0.4 (depending on which version was selected), and cloud tooling software on the compute node.
- Creates storage for database data, the fast recovery area, and the redo logs.
- Creates and starts an Oracle database on the compute node and starts the network listener for the node.

Topics

- [Linux User Accounts](#)
- [Storage Volumes and File System Layout](#)
- [Locations of Installed Software](#)
- [Oracle Database Characteristics](#)

- [Data Security](#)
- [Location of Diagnostic and Log Files](#)

Linux User Accounts

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

Every Database Cloud Service compute node is provisioned with the following operating system user accounts.

User	Description
opc	The system administrator account you use with the <code>sudo</code> command to perform operations that require root-user access.
oracle	The Oracle Database administrator account you use to access the system and perform non-root 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 root-user access.
root	The root administrator for the system. You do not have direct access to this account. To perform operations that require root-user access, use the <code>sudo</code> command as the <code>opc</code> user.

The following environment variables are created for the `opc` and `oracle` users.

Variable	Description
HOME	The home directory of the user, either <code>/home/opc</code> or <code>/home/oracle</code> .
HOSTNAME	The host name of the compute node.
LANG	The system language, <code>en_US.UTF-8</code> .
OH	Short form for <code>ORACLE_HOME</code> .
ORACLE_HOME	The Oracle Database home directory:
ME	<ul style="list-style-type: none">• Oracle Database 12c Release 2: <code>/u01/app/oracle/product/12.2.0/dbhome_1</code>• Oracle Database 12c Release 1: <code>/u01/app/oracle/product/12.1.0/dbhome_1</code>• Oracle Database 11g Release 2: <code>/u01/app/oracle/product/11.2.0/dbhome_1</code>
ORACLE_SID	The database system identifier (SID) provided during the database deployment creation process.

Variable	Description
PATH	The paths to search for executables; set to include: <ul style="list-style-type: none"> • /sbin • /usr/sbin • /bin • /usr/bin • \$ORACLE_HOME/bin • \$ORACLE_HOME/OPatch • \$HOME
SHELL	The default shell, /bin/bash.
USER	The user name, either <code>opc</code> or <code>oracle</code> .

Storage Volumes and File System Layout

This section provides information about the storage volumes and file system layout of a newly created database deployment on Oracle Database Cloud Service.

Compute Cloud Service Storage Volumes

When a Database Cloud Service deployment is created using the Oracle Database Cloud Service service level, the following storage volumes are created.

Storage Volume	Description
bits	30 GB volume completely allocated to /u01 on the virtual machine.
boot	31 GB volume allocated to the following file system mounts on the virtual machine: <ul style="list-style-type: none"> • / (root) • /boot • /opt/emagent_instance • swap space
data	GB size equal to the value provided in the Usable Data Storage field during the database deployment creation process, with a minimum of 15 GB. This volume is completely allocated to /u02 on the virtual machine.
fra	GB size depends on the choice of Backup Destination during the database deployment creation process: <ul style="list-style-type: none"> • Remote Storage Only: GB size equal to 1.7 times the size of the data volume. • None: GB size equal to 0.7 times the size of the data volume, with a minimum of 7 GB This volume is completely allocated to /u03 on the virtual machine.
redo	10 GB volume completely allocated to /u04 on the virtual machine.

File System Layout

When a database deployment is created using the Oracle Database Cloud Service service level, Oracle Cloud Service storage volumes are created and allocated as follows.

File System Mount	Description
swap	Swap space; 4 GB allocated from the boot Compute Cloud storage volume.
/ (root)	Operating system files; 24 GB allocated from the boot Compute Cloud storage volume.
/boot	Operating system kernel; 250 MB allocated from the boot Compute Cloud storage volume.
/opt/ emagent_instance	Oracle Enterprise Manager agent software; 2 GB allocated from the boot Compute Cloud storage volume.
/u01	Oracle product software; the entire bits Compute Cloud storage volume.
/u02	Oracle Database data storage; the entire data Compute Cloud storage volume.
/u03	Database backup storage; the entire fra Compute Cloud storage volume.
/u04	Database redo logs; the entire redo Compute Cloud storage volume.

Locations of Installed Software

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

When a database deployment is created using the Oracle Database Cloud Service service level, software is installed in the following locations.

Software	Installation Location
Oracle Database	\$ORACLE_HOME: <ul style="list-style-type: none"> Oracle Database 12c Release 2: /u01/app/oracle/product/12.2.0/dbhome_1 Oracle Database 12c Release 1: /u01/app/oracle/product/12.1.0/dbhome_1 Oracle Database 11g Release 2: /u01/app/oracle/product/11.2.0/dbhome_1
Oracle REST Data Services	/u01/app/oracle/product/apex_listener
dbaascli utility	/var/opt/oracle/dbaascli
bkup_api utility	/var/opt/oracle/bkup_api

Oracle Database Characteristics

When a database deployment is created on Oracle Database Cloud Service, an Oracle database is created using information provided in the Provision New Oracle Database Cloud 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.
Software Edition on the Service page	Determines which database edition is used. The edition determines what database features and options are available. For more information, see Oracle Database Edition .
Usable Data Storage (GB) on the Service Details page	The amount of data storage for the database data files. A storage volume of this size is created and mounted on /u02.
Administrator Password on the Service Details page	The password used for the SYS and SYSTEM database users.
DB Name (SID) on the Service Details page	The database system identifier (SID) of the database.
PDB Name on the Service Details page	(Oracle Database 12c only) The name of the default pluggable database (PDB) created in the database.

Data Security

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

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

Topics

- [Security of Data at Rest](#)
- [Security of Data in Transit](#)

Security of Data at Rest

Oracle Database Cloud Service databases include a key management framework that stores and manages keys and credentials used to encrypt data in the database data files and in backups.

The key management framework includes the keystore (referred to as a wallet in Oracle Database 11g and previous releases) to securely store Transparent Data Encryption (TDE) master encryption keys, and the management framework to securely and efficiently manage keystore and key operations for various database components. TDE is the underlying mechanism used for default tablespace encryption and encrypted backups.

TDE uses a two-tiered, key-based architecture to transparently encrypt and decrypt data. The master encryption key is stored in the software keystore. For tablespace encryption, this master encryption key is used to encrypt the tablespace encryption key, which in turn is used to encrypt and decrypt data in the tablespace. Refer to [Using Tablespace Encryption in Database Cloud Service](#) for details on the implementation of tablespace encryption by default in Oracle Database Cloud Service

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

The keystore location is specified in the `ENCRYPTION_WALLET_LOCATION` parameter in the `$ORACLE_HOME/network/admin/sqlnet.ora` file.

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

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

Security of Data in Transit

Oracle Database Cloud Service uses native Oracle Net encryption and integrity capabilities to secure connections to the database.

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

Location of Diagnostic and Log Files

When a database deployment is created on Oracle Database Cloud Service, log files from the creation operation are stored in subdirectories of `/var/opt/oracle/log`.

By default, Oracle Database trace files and log files are stored in subdirectories of `/u01/app/oracle/diag`.

Characteristics of an Oracle Database Cloud Service Deployment Using Oracle RAC

This section provides information about the content and configuration of a newly created database deployment on Oracle Database Cloud Service that hosts an Oracle Real Application Clusters (Oracle RAC) database.

When you create a database deployment using the Oracle Database Cloud Service service level and choose the RAC Database option, Database Cloud Service creates a two-node cluster database using Oracle RAC. Each node of the database is housed on a compute node that Database Cloud Service creates using Oracle Compute Cloud Service resources. In brief, Database Cloud Service:

- Creates two compute nodes that are alike in all respects except that each one has its own IP address.
- Installs Oracle Linux 6.6, Oracle Grid Infrastructure 12.1.0.2, Oracle Database 12.1.0.2 or 11.2.0.4 (depending on which version was selected), and cloud tooling software on each of the compute nodes.

- Creates three Oracle Automatic Storage Management (ASM) disk groups to provide shared storage for database data, the fast recovery area, and the redo logs, and mounts the disk groups as shared file systems on the two compute nodes using Oracle ASM Cluster File System (ACFS).
- Creates and starts a two-node Oracle RAC database on the compute nodes and starts the network listeners for the nodes.

The following topics provide more detail about this configuration:

Topics

- [Linux User Accounts](#)
- [Storage Volumes and File System Layout](#)

Linux User Accounts

This section provides information about Linux user accounts that are provisioned on an Oracle Database Cloud Service deployment that hosts an Oracle Real Application Clusters (Oracle RAC) database.

Both compute nodes are provisioned with the following operating system user accounts.

User	Description
opc	The system administrator account you use to connect to the compute node using SSH. This user can use the <code>sudo</code> command to perform operations that require root-user access.
oracle	The Oracle Database administrator account you use to access the system and perform non-root 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 root-user access. Additionally, by default you cannot connect as this user to the compute node using SSH. You can add the public key to the user's <code>\$HOME/.ssh/authorized_keys</code> file to grant persistent SSH access, or you can connect as the <code>opc</code> user and then use the <code>sudo -s</code> command to start a root-user command shell, followed by an <code>su - oracle</code> command to switch to the <code>oracle</code> user.
grid	The Oracle Grid Infrastructure administrator account you use to perform ASM, ACFS, 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 root-user access. Additionally, by default you cannot connect as this user to the compute node using SSH. You can add the public key to the user's <code>\$HOME/.ssh/authorized_keys</code> file to grant persistent SSH access, or you can connect as the <code>opc</code> user and then use the <code>sudo -s</code> command to start a root-user command shell, followed by an <code>su - grid</code> command to switch to the <code>grid</code> user.
root	The root administrator for the system. You do not have direct access to this account. To perform operations that require root-user access, use the <code>sudo</code> command as the <code>opc</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 /home/opc, /home/oracle or /home/grid.
HOSTNAME	The host name of the compute node: <ul style="list-style-type: none">• <i>deployment-name1</i> for the first compute node• <i>deployment-name2</i> for the second compute node
LANG	The system language, en_US.UTF-8.
SHELL	The default shell, /bin/bash.
USER	The user name, either opc, oracle or grid.

In addition, the PATH variable is also created for all three users, but its value differs (line breaks added to improve clarity):

- For the **opc** user:

```
/opt/oracle/dcs/client/bin:/usr/java/jdk1.7.0_72/bin:  
/usr/lib64/qt-3.3/bin:  
/usr/local/bin:/bin:/usr/bin:  
/usr/local/sbin:/usr/sbin:/sbin:  
/home/opc/bin
```

- For the **oracle** user:

```
/usr/lib64/qt-3.3/bin:  
/usr/local/bin:/bin:/usr/bin:  
/usr/local/sbin:/usr/sbin:/sbin:  
/u01/app/oracle/product/db-version/dbhome_1/bin:  
/home/oracle/bin
```

where *db-version* is either 12.1.0.2 or 11.2.0.4, depending on which version of Oracle Database was installed.

- For the **grid** user:

```
/usr/lib64/qt-3.3/bin:  
/usr/local/bin:/bin:/usr/bin:  
/usr/local/sbin:/usr/sbin:/sbin:  
/u01/app/12.1.0.2/grid/bin:  
/home/grid/bin
```

In addition, the following environment variable settings are created for the **oracle** user.

Variable	Description
LD_LIBRARY_PATH	The Oracle Database library directory: <ul style="list-style-type: none">• For Oracle Database 12c, /u01/app/oracle/product/12.1.0.2/dbhome_1/lib• For Oracle Database 11g, /u01/app/oracle/product/11.2.0.4/dbhome_1/lib
Y_PATH	

Variable	Description
ORACLE_HO ME	The Oracle Database home directory: • For Oracle Database 12c, /u01/app/oracle/product/12.1.0.2/dbhome_1 • For Oracle Database 11g, /u01/app/oracle/product/11.2.0.4/dbhome_1
ORACLE_SI D	The database system identifier (SID) for the database instance on the compute node: • <i>db-sid1</i> for the first compute node • <i>db-sid2</i> for the second compute node where <i>db-sid</i> is the database system identifier (SID) provided as the DB Name (SID) value during the database deployment creation process.
ORACLE_UN QNAME	The database system identifier (SID) provided as the DB Name (SID) value during the database deployment creation process.

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

Variable	Description
LD_LIBRARY_PATH	Set to /u01/app/12.1.0.2/grid/lib.
ORACLE_HO ME	The Oracle Grid Infrastructure home directory: /u01/app/12.1.0.2/grid.

Storage Volumes and File System Layout

This section provides information about the storage volumes and file system layout of a newly created database deployment on Oracle Database Cloud Service that hosts an Oracle Real Application Clusters (Oracle RAC) database.

When the database deployment is created, the following Oracle Compute Cloud Service storage volumes are created and attached to the two compute nodes.

Storage Volume	Description
boot (two volumes, one for each compute node)	31 GB volume for operating system files, user directories and swap space.
bits (two volumes, one for each compute node)	70 GB volume for Oracle Database and Oracle Grid Infrastructure software.
data (one volume accessed by both compute nodes)	Shared storage for database files. GB size equal to two plus the value provided in the Usable Data Storage field during the database deployment creation process, with a minimum of 15 GB.

Storage Volume	Description
fra (one volume accessed by both compute nodes)	Shared storage for the fast recovery area. GB size depends on the choice of Backup Destination during the database deployment creation process: <ul style="list-style-type: none"> • Remote Storage Only: GB size equal to 1.7 times the size of the data volume. • None: GB size equal to 0.7 times the size of the data volume, with a minimum of 7 GB
redo (one volume accessed by both compute nodes)	20 GB shared storage volume for redo logs.

These storage volumes are mounted on the compute nodes as follows.

File System Mount	Description
swap	Swap space; 4 GB allocated from the boot Compute Cloud storage volume.
/ (root)	Operating system files; 16 GB allocated from the boot Compute Cloud storage volume.
/boot	Operating system kernel; 500 MB allocated from the boot Compute Cloud storage volume.
/u01	Oracle product software; the entire bits Compute Cloud storage volume.
/u02	Oracle Database data storage; the entire data Compute Cloud storage volume. An Oracle ASM diskgroup named DATA is created on the storage volume and Oracle ACFS is used to mount it.
/u03	Database backup storage; the entire fra Compute Cloud storage volume. An Oracle ASM diskgroup named FRA is created on the storage volume and Oracle ACFS is used to mount it.
/u04	Database redo logs; the entire redo Compute Cloud storage volume. An Oracle ASM diskgroup named REDO is created on the storage volume and Oracle ACFS is used to mount it.

Oracle Cloud Pages for Administering Database Cloud Service

This section provides information about what you can do and what you see on each of the Oracle Cloud pages for administering Oracle Database Cloud Service.

Topics

- [Services Page](#)
- [Activity Page](#)
- [SSH Access Page](#)
- [Overview Page](#)
- [Backup Page](#)
- [Patching Page](#)
- [Provision New Oracle Database Cloud Service: Service Page](#)
- [Provision New Oracle Database Cloud Service: Service Details Page](#)
- [Provision New Oracle Database Cloud Service: Confirmation Page](#)

Services Page

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

Topics

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

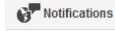
What You Can Do From the Oracle Database Cloud Service Services Page

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 to My Services and other Oracle Cloud services in the identity domain.
 username 	User menu providing access to help, accessibility options, console version information and sign-out.
 Dashboard	Click to go to the My Services Dashboard page.
 Users	Click to go to the My Services Users page.
 Notifications	Click to go to the My Services Notifications page.
 menu next to “Oracle Database Cloud Service”	Menu that provides access to Platform Services.
Activity	Click to go to the Activity Page .
SSH Access	Click to go to the SSH Access Page .
Welcome!	Click to go to the Oracle Database Cloud Service console Welcome page.
REST APIs	Click to go to the API Catalog Cloud Service.
Services, OCPUs, Memory, Storage and Public IPs	<p>Summary of resources being used:</p> <ul style="list-style-type: none"> Services — Total number of configured deployments. OCPUs — Total number of Oracle CPUs allocated across all deployments. Memory — Total amount of compute node memory allocated across all deployments. Storage — Total amount of Oracle Compute Cloud Service storage allocated across all deployments. Public IPs — Number of public IP addresses allocated across all deployments.
<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.
Create Service	Click to create a new database deployment on Database Cloud Service. See Creating a Database Deployment .
	Click to view details for the deployment.

Element	Description
Status	Status of the deployment if it is not running. Status values include “In Progress”, “Maintenance”, “Stopped”, and “Terminating”.
Version	Version of Oracle Database configured on the deployment. For example: 12.1.0.2 or 11.2.0.4.
Edition	Software edition of Oracle Database configured on the deployment. For example: Enterprise Edition or Standard Edition.
Created On or Submitted On	Date when the deployment was created. During the creation process, the date when the creation request was submitted.
OCPUs	Number of Oracle CPUs associated with the deployment.
Memory	Amount of compute node memory in GBs associated with the deployment.
Storage	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"> • Open DBaaS Monitor Console — Open the Oracle Cloud Database Monitor for the deployment. • Open Application Express Console — Open the Oracle Application Express home page for the deployment. • Open EM Console — Open the database console, either Enterprise Manager Database Express 12c or Enterprise Manager 11g Database Control. • SSH Access — Add an SSH public key. See Adding an SSH Public Key. • Delete — Delete the deployment. See Deleting a Database Deployment.
Service create and delete history	Listing of attempts to create or delete a deployment. Click the triangle icon next to the title to view the history listing.

Activity Page

The Activity page displays activities for all Oracle Database Cloud Service deployments in your identity domain. You can restrict the list of activities displayed using search filters.

Topics

- [What You Can Do From the Activity Page](#)
- [What You See on the Activity Page](#)

What You Can Do From the Activity Page

Use the Activity page to view operations for all Database Cloud Service deployments in your identity domain.

You can use the page's Search Activity Log section to filter the list of displayed operations based on:

- The time the operation was started
- The status of the operation
- The name of the deployment on which the operation was performed
- The type of the operation

In the table of results, you can:

- Click any column heading to sort the table by that column.
- Click the triangle at the start of an operation's row to see more details about that operation.

What You See on the Activity Page

The following table describes the key information shown on the Activity page.

Element	Description
Start Time Range	Filters activity results to include only operations started within a specified time range. The range defaults to the previous 24 hours.
Status	Filters operations by status of the operation: <ul style="list-style-type: none">• All• Scheduled• Running• Succeeded• Failed You can select any subset of status types. The default value is All.
Service Name	Filters the activity results to include operations only for the specified service instance. You can enter a full or partial service instance name.
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.

Element	Description
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.
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 Cloud Service deployments in your identity domain. You can restrict the list of deployments displayed using search filters.

Topics

- [What You Can Do From the Activity Page](#)
- [What You See on the Activity Page](#)

What You Can Do From the SSH Access Page

Use the SSH Access page to view and add SSH public keys to Database Cloud Service deployments in your identity domain.

You can use the page's Search section to filter the list of displayed deployments based on deployment name.

In the table of results, you can:

- Click any column heading to sort the table by that column.
- Click the triangle at the start of a deployment's row to see more details.

What You See on the SSH Access Page

The following table describes the key information shown on the SSH Access page.

Element	Description
Service Name	Filters the results to include SSH keys only for the specified deployment. You can enter a full or partial deployment name.
Service Type	Filters the results to include SSH keys only for deployments of the specified service type. The default value is the current cloud service.
Search	Searches for SSH keys by applying the filters specified by the Service Name and Service Type fields, and displays the results in the table.
Results per page	Specifies the number of results you want to view per page. The default value is 10.
	Displays a description of an item in the results table. Clicking on the resulting downward arrow hides the description.
Service Name	Shows the name of the deployment.
Service Type	Shows the type of cloud service for this deployment.
Last Update	Shows the most recent time the SSH keys for this deployment were updated.
Actions	<p>Click the Add New Key button to add a new SSH public key to this deployment.</p> <p>The Add New Key overlay is displayed with its Key value 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"> Select Upload a new SSH Public Key value and click Choose File to select a file that contains the public key. Select Key value. Delete the current key value and paste the new public key into the text area. Make sure the value does not contain line breaks or end with a line break.

Overview Page

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

The following tables describe the elements and options available in the various areas of the Overview page:

- [What You See in the Banner Area](#)
- [What You See in the Tiles Area](#)
- [What You See in the Page Content Area](#)

What You See in the Banner Area

The following table describes the elements and options available in the banner area at the top of the page.

Element	Description
 menu	Navigation menu providing access to My Services and other Oracle Cloud services in the identity domain.
username 	User menu providing access to help, accessibility options, console version information and sign-out.
 Dashboard	Click to go to the My Services Dashboard page.
 Users	Click to go to the My Services Users page.
 Notifications	Click to go to the My Services Notifications page.
 (next to the "Oracle Database Cloud Service" link)	Click to see details about the database deployment: description, identity domain, subscription type, user who created the deployment, and when the deployment was created.
Oracle Database Cloud Service link	Click to return to the Services Page .
 (next to the deployment's name)	Deployment menu that provides the following options: <ul style="list-style-type: none"> • Open DBaaS Monitor Console — Open the Oracle Cloud Database Monitor for the deployment. • Open Application Express Console — Open the Oracle Application Express home page for the deployment. • Open EM Console — Open the database console for the deployment, either Enterprise Manager Database Express 12c or Enterprise Manager 11g Database Control. • Start — Start a stopped deployment. See Stopping, Starting and Restarting a Database Deployment. • Stop — Stop a deployment. See Stopping, Starting and Restarting a Database Deployment. • Restart — Restart a deployment. See Stopping, Starting and Restarting a Database Deployment. • Scale Up/Down — Scale the compute shape or storage of the deployment. For information, see Scaling a Database Deployment. • SSH Access — Add an SSH public key to the deployment. See Adding an SSH Public Key. • View Activity — Go to the Activity Page to view activities performed on this deployment.
	Click to start a stopped deployment.
	Click to stop a running deployment.

Element	Description
	Click to restart a running deployment.
	Click to poll the status of the deployment's compute nodes and display the results on this page.

What You See in the Tiles Area

The following table describes the elements and options available in the tiles area at the side of the page.

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

What You See in the Page Content Area

The following table describes the elements and options available in the main content area of the page.

Element	Description
	Click to refresh the page.
Service Overview section	Displays a summary box followed by information about the deployment. The summary box shows high-level information about the deployment: compute nodes, OCPUs, memory, and local storage. Following the summary box is a listing of information about the deployment, including Oracle Database version, Software edition, backup destination, overall status, and so on. Click the Show more... link to see even more information about the deployment.
Resources section	Contains an entry for each compute node of the deployment. Each entry displays information about the compute node and provides a menu to perform actions on the compute node.

Element	Description
 (for each compute node)	Compute node menu that provides the following options: <ul style="list-style-type: none"> • Start—Start a stopped compute node. See Stopping, Starting and Restarting a Database Deployment • Stop—Stop a compute node. See Stopping, Starting and Restarting a Database Deployment • Restart—Restart a compute node. See Stopping, Starting and Restarting a Database Deployment • Scale Up/Down—Scale the compute shape or storage of a compute node. For information, see Scaling a Database Deployment.

Backup Page

You use the Backup page to manage backup and recovery of a particular database deployment.

What You See on the Oracle Database Cloud Service Backup Page

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

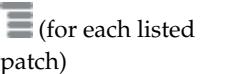
Element	Description
Backup Now	Click to create a full backup of the database deployment.
Recover	Click to recover the database deployment to the latest backup or to a specific point in time.
Configure Backups	Click to update the credentials for backing up to cloud storage.
 (for each available backup)	Menu that provides the Recover option. Choose this option to recover to the given backup.
Recovery History	Listing of recovery operations on the database deployment. Click the triangle icon next to the title to view the listing.

Patching Page

You use the Patching page to view available patches, initiate a patching process, and view details of the last patching process for a particular database deployment.

What You See on the Oracle Database Cloud Service Patching Page

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

Element	Description
 Services	Click the Services home icon from any page to return to the Oracle Database Cloud Service Services page.
Oracle Database Version	Version of Oracle Database configured for the deployment. For example: 12.1.0.2 or 11.2.0.4.
Description	Description of the deployment.
 (for deployment)	<p>Menu icon provides the following options:</p> <ul style="list-style-type: none"> • Open DBaaS Monitor Console—Open the Oracle Cloud Database Monitor for the deployment. • Open Application Express Console—Open the Oracle Application Express home page for the deployment. • Open EM Console—Open the database console for the deployment, either Enterprise Manager Database Express 12c or Enterprise Manager 11g Database Control. • SSH Access—Add an SSH public key to the deployment. See Adding an SSH Public Key. • View Activity—Go to the Activity Page to view activities performed on this deployment.
Available Patches	A list of patches you can apply to the deployment.
 (for each listed patch)	<p>Menu icon provides the following options for the patch:</p> <ul style="list-style-type: none"> • Precheck—Check whether the patch can be successfully applied to the deployment. • Patch—Apply the patch to the deployment.
Details of Last Patching Activity	Expand to see a description of the actions taken during the last patching operation.
Rollback	Click to roll back the last patching operation. See Rolling Back a Patch or Failed Patch by Using the Oracle Database Cloud Service Console .

Provision New Oracle Database Cloud Service: Service Page

Provision New Oracle Database Cloud 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
Cancel	Click to cancel the Provision New Oracle Database Cloud Service wizard without creating a new database deployment.
Next>	Click to advance to the Provision New Oracle Database Cloud Service: Service Details page.

What You See in the Page Content Area

The following table describes the key information shown on the Provision New Oracle Database Cloud Service: Service page.

Element	Description
Service Name	<p>The name for the new database deployment. The name:</p> <ul style="list-style-type: none"> • Must not exceed 50 characters (15 characters for a deployment that uses Oracle RAC). • Must start with a letter. • Must contain only letters, numbers, or hyphens. • Must not contain any other special characters. • Must be unique within the identity domain.
Description	(Optional) A description for the new database deployment.
Service Level	<p>The service level for the new deployment:</p> <ul style="list-style-type: none"> • Oracle Database Cloud Service See Service Level.
Metering Frequency	<p>The metering frequency for the new deployment:</p> <ul style="list-style-type: none"> • Hourly • Monthly <p>See Metering Frequency.</p>
Software Release	<p>The release version of Oracle Database for the new deployment:</p> <ul style="list-style-type: none"> • Oracle Database 11g Release 2 • Oracle Database 12c Release 1 • Oracle Database 12c Release 2 <p>See Oracle Database Software Release.</p>
Software Edition	<p>The Oracle Database software package for the new deployment:</p> <ul style="list-style-type: none"> • Standard Edition • Enterprise Edition • Enterprise Edition - High Performance • Enterprise Edition - Extreme Performance <p>See Oracle Database Software Edition.</p>

Element	Description
Database Type	<p>The type of deployment to create:</p> <ul style="list-style-type: none"> • Single Instance—A single Oracle Database instance and database data store hosted on one compute node. For more information about this type, see Characteristics of an Oracle Database Cloud Deployment. • Database Clustering with RAC—A two-node clustered database using Oracle Real Application Clusters technology; two compute nodes each host an Oracle Database instance, and the two instances access the same shared database data store. For more information about this type, see Characteristics of an Oracle Database Cloud Service Deployment Using Oracle RAC. <p>Not all types are available with all combinations of service level and software edition:</p> <ul style="list-style-type: none"> • Single Instance is the only type supported by the Standard Edition software edition. • Database Clustering with RAC is available only with Enterprise Edition - Extreme Performance software edition.

Provision New Oracle Database Cloud Service: Service Details Page

Provision New Oracle Database Cloud Service: Service Details is a page in the Provision New Oracle Database Cloud 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 Provision New Oracle Database Cloud 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 Networks Section](#)

What You See in the Navigation Area

Element	Description
<Previous	Click to return to the Provision New Oracle Database Cloud Service: Service page.
Cancel	Click to cancel the Provision New Oracle Database Cloud Service wizard without creating a new database deployment.
Next>	Click to advance to the Provision New Oracle Database Cloud 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"> • Must not exceed 8 characters. • Must start with a letter. • Must contain only letters, numbers, or these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).
PDB Name	(Available only for Oracle Database 12c.) The name for the default pluggable database (PDB). The name: <ul style="list-style-type: none"> • Must not exceed 8 characters. • Must start with a letter. • Must contain only letters, numbers, or these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).
Administration Password	The password for the following administrative users:
Confirm Password	<ul style="list-style-type: none"> • Oracle Database administrative users • Oracle Application Express admin user The password: <ul style="list-style-type: none"> • 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 \$ (dollar sign). • Must not contain the word “oracle”.
Usable Database Storage (GB)	The amount of storage in GB for actual database data.
Total Data File Storage (GB)	The computed amount of storage in GB that will be allocated to the deployment, including space for operating system and product binaries, supporting files, database data and configuration files, and so on.
Compute Shape	The number of Oracle Compute Units (OCpus) and amount of memory (RAM) for each compute node of the new database deployment. Database Cloud Service offers several OCPU/RAM combinations, as described in Computing Power .
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 Edit to specify the public key. You can upload a file containing the public key value, paste in the value of a public key, or create a system-generated key pair.</p> <p>If you paste in the value, make sure the value does not contain line breaks or end with a line break.</p>

Element	Description
Advanced Settings: Listener Port	The port number for the Oracle Net Listener. The port number must be between 1521 and 5499 (inclusive).
Advanced Settings: Timezone	The time zone for the new database deployment. The default is Coordinated Universal Time (UTC).
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"> • 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
Advanced Settings: National Character Set	The national character set for the database. The national character set is used for data stored in SQL NCHAR data types (NCHAR, NCLOB, and NVARCHAR2).
Advanced Settings: Include "Demos" PDB	<p>(Available only for Oracle Database 12c.)</p> <p>Controls whether the "Demos" PDB is to be included in the database. This PDB contains demos for many new features of Oracle Database 12c such as in-memory and multitenant. Usable Data File Storage must be at least 25 GB to include this PDB.</p>

What You See in the Backup and Recovery Configuration Section

Element	Description
Backup Destination	<p>Controls how backups for the deployment are to be configured:</p> <ul style="list-style-type: none"> • Remote Storage Only—backups are configured to be created automatically and stored on an NFS remote backup location. If this choice is selected, the NFS Remote Backup field is displayed. • None—Backups are not configured for the deployment. <p>For more information about backups and backup configurations, see About Backing Up Database Deployments on Database Cloud Service.</p>
Total Estimated Monthly Storage	Storage for data files and backups.
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> <p><i>hostname:absolute-path</i> <i>ip-address:absolute-path</i></p>

What You See in the Networks Section

Element	Description
Public Access Network	The network in which this database deployment will be available.

Provision New Oracle Database Cloud Service: Confirmation Page

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

What You See on the Provision New Oracle Database Cloud Service: Confirmation Page

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

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

The dbaascli Utility

The dbaascli utility is provided on Oracle Database Cloud Service deployments to perform a variety of life-cycle and administration operations.

Using the dbaascli utility, you can perform operations like:

- Changing the password of the `SYS` user
- Patching the database deployment
- Database recovery
- Rotating the master encryption key

To use the dbaascli utility:

1. Connect to a compute node associated with the Database Cloud Service deployment. For most dbaascli subcommands, connect as the `oracle` user. For the `dbpatchm` and `orec` subcommands, connect as the `opc` user.

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

2. If you connected as the `opc` user, start a root-user command shell:

```
$ sudo -s  
#
```

3. Run the dbaascli utility using a command of the form:

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

4. When you are finished using the dbaascli utility, exit the root-user command shell if necessary and disconnect from the compute node:

```
# exit  
$ exit
```

The dbaascli utility supports these subcommands:

Subcommand	Subcommand Options
database	<code>bounce</code> – shuts down and then restarts the database instance. <code>changepassword</code> – changes the password of the specified user. <code>start</code> – starts the database instance and opens the database. <code>status</code> – displays the open mode of the database and additional information about the database deployment. <code>stop</code> – shuts down the database instance.

Subcommand	Subcommand Options
dbpatchm	<code>apply</code> – applies the patch. <code>clonedb</code> – applies a patch to a test deployment. <code>list_patches</code> – displays a list of available patches. <code>prereq</code> – checks the prerequisites of a patch. <code>rollback</code> – rolls back the last deployment patch. <code>switchback</code> – restores database software to a prior state.
dv	<code>off</code> – disables Oracle Database Vault. <code>on</code> – enables Oracle Database Vault.
listener	<code>bounce</code> – stops and restarts the listener. <code>start</code> – starts the listener. <code>status</code> – displays the status of the listener, including a summary of listener configuration settings, listening protocol addresses, and a summary of services registered with the listener. <code>stop</code> – stops the listener.
netsec	<code>config</code> – configures network encryption and network integrity. <code>config encryption</code> – configures network encryption. <code>config integrity</code> – configures network integrity. <code>status</code> – displays network encryption and network integrity configuration information.
orec	<code>keep list</code> – lists the available long-term backups. <code>keep tag</code> – restores a specific long-term backup and performs recovery. <code>latest</code> – restores the most recent backup and performs complete recovery. <code>list</code> – lists the available normal backups. <code>pitr</code> – restores a specific normal backup and performs recovery. <code>scn</code> – restores the most recent backup and performs recovery through the specified SCN.
tde	<code>rotate masterkey</code> – changes (rotates) the master encryption key. <code>status</code> – displays information about the software keystore, including the type and status.

dbaascli database bounce

The database bounce subcommand of the dbaascli utility can be used to shut down and restart the database.

Execute this command as the `oracle` user.

```
dbaascli database bounce
```

When this subcommand is executed the database is shut down in immediate mode. The database instance is then restarted and the database is opened. In an Oracle Database 12c database, all PDBs are opened.

dbaascli database changepassword

The database changepassword subcommand of the dbaascli utility is used to change the password of a database user.

Execute this command as the oracle user.

```
dbaascli database changepassword
```

Enter the user name and new password when prompted.

dbaascli database start

The database start subcommand of the dbaascli utility can be used to start the database instance and open the database.

Execute this command as the oracle user.

```
dbaascli database start
```

When this subcommand is executed the database instance is started and the database is opened. In an Oracle Database 12c database, all PDBs are opened.

dbaascli database status

The database status subcommand of the dbaascli utility can be used to check the status of the database in your database deployment.

Execute this command as the oracle user.

```
dbaascli database status
```

Output from the command includes the open mode of the database, the software release and edition of the database deployment, and release version of other software components.

dbaascli database stop

The database stop subcommand of the dbaascli utility can be used to shut down the database.

Execute this command as the oracle user.

```
dbaascli database stop
```

When this subcommand is executed the database is shut down in immediate mode. No new connections or new transactions are permitted. Active transactions are rolled back and all connected users are disconnected.

dbaascli dbpatchm apply

The dbpatchm apply subcommand of the dbaascli utility is used to apply a patch.

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

```
dbaascli dbpatchm --run -apply
```

Before executing the apply subcommand you must edit the /var/opt/oracle/patch/dbpatchm.cfg patching configuration file, setting the keys for the desired patch. For more information about this file and its keys, see [The dbpatchm.cfg Configuration File](#).

The dbpatchm apply subcommand displays progress as the patch is applied.

dbaascli dbpatchm clonedb

The dbpatchm clonedb subcommand of the dbaascli utility is used to apply a patch to a test deployment before you apply it to a live, production database deployment.

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

```
dbaascli dbpatchm --run -clonedb
```

The dbpatchm clonedb subcommand displays progress information as it copies information from the live deployment to the test deployment and then applies the patch to the test deployment.

dbaascli dbpatchm list_patches

The dbpatchm list_patches subcommand of the dbaascli utility is used to check whether any patches are available.

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

```
dbaascli dbpatchm --run -list_patches
```

A patch update is available if the command response includes the "INFO: images available for patching" message. The patch ID will be displayed as part of the command response. This patch ID can be used to download and apply the patch. See [dbaascli dbpatchm apply](#) for detail on applying a patch.

dbaascli dbpatchm prereq

The dbpatchm prereq subcommand of the dbaascli utility is used to check the prerequisites of a patch.

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

```
dbaascli dbpatchm --run -prereq
```

Before executing the prereq subcommand you must edit the `/var/opt/oracle/patch/dbpatchm.cfg` patching configuration file, setting the keys for the desired patch. For more information about this file and its keys, see [The dbpatchm.cfg Configuration File](#).

dbaascli dbpatchm rollback

The dbpatchm rollback subcommand of the dbaascli utility is used to roll back the last patch applied to a database deployment.

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

```
dbaascli dbpatchm --run -rollback
```

dbaascli dbpatchm switchback

The dbpatchm switchback subcommand of the dbaascli utility is used to revert back to the state of the database deployment before a patch was applied.

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

```
dbaascli dbpatchm --run -switchback
```

dbaascli dv off

The `dv off` subcommand of the `dbaascli` utility is used to disable Oracle Database Vault in a database deployment.

Note: this command shuts down and then restarts the database. Therefore, you should make sure all database connections are closed before you use this command.

```
dbaascli dv off [cdb|pdb]
```

The utility prompts the user for the following:

- Database Vault Owner user name
- Database Vault Owner password

If no options are specified, the utility will disable Database Vault on the root container (CDB) and all existing pluggable databases (PDBs) in an Oracle Database 12c database deployment. The utility assumes that Database Vault credentials are the same in the CDB and all of the PDBs.

When you install Oracle Database Vault, it revokes a set of privileges from several Oracle Database-supplied users and roles. Be aware that if you disable Oracle Database Vault, these privileges remain revoked. See "Privileges That Are Revoked from Existing Users and Roles" in *Oracle Database Vault Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#) for additional information.

Options of this subcommand are as follows.

Option	Description
<code>cdb</code>	The <code>cdb</code> option is used to disable Database Vault on the root container (CDB) only in an Oracle Database 12c database deployment.
<code>pdb</code>	The <code>pdb</code> option is used to disable Database Vault on a specific PDB in an Oracle Database 12c database deployment. When this option is used, the utility also prompts the user for the PDB name.

dbaascli dv on

The `dv on` subcommand of the `dbaascli` utility is used to configure and enable Oracle Database Vault in a database deployment.

```
dbaascli dv on [cdb|pdb]
```

The utility prompts the user for the following:

- Database Vault Owner user name
- Database Vault Owner password
- Database Vault Account Manager user name

- Database Vault Account Manager password
- PDB name, if the `pdb` option is specified

If no options are specified, the utility will configure and enable Database Vault on the root container (CDB) and all existing PDBs of an Oracle Database 12c database deployment. It will set the same Database Vault credentials in the root container and the PDBs.

If you want to enable Database Vault on PDBs separately, you must first enable Database Vault on the CDB.

In an Oracle Database 12c database deployment, the Database Vault Owner and Account Manager user names must begin with `c##`.

Review "What to Expect After You Enable Oracle Database Vault" in *Oracle Database Vault Administrator's Guide* for Release [12.2](#), [12.1](#) or [11.2](#) to gain an understanding of the impact of enabling and configuring Database Vault.

Options of this subcommand are as follows.

Option	Description
<code>cdb</code>	The <code>cdb</code> option is used to configure and enable Database Vault on the root container (CDB) only in an Oracle Database 12c database deployment.
<code>pdb</code>	The <code>pdb</code> option is used to configure and enable Database Vault on a specific PDB in an Oracle Database 12c database deployment. When this option is used, the utility also prompts the user for the PDB name. Database Vault must be enabled on the CDB before you can enable it on any PDBs. Provide the same credentials for the PDB as you provided for the CDB.

dbaascli listener bounce

The `listener bounce` subcommand of the `dbaascli` utility is used to stop and restart the listener.

Execute this command as the `oracle` user.

```
dbaascli listener bounce
```

This command causes the listener to be stopped and then restarted.

dbaascli listener start

The `listener start` subcommand of the `dbaascli` utility is used to start the listener.

Execute this command as the `oracle` user.

```
dbaascli listener start
```

dbaascli listener status

The `listener status` subcommand of the `dbaascli` utility is used to obtain information about the status of the listener.

Execute this command as the `oracle` user.

```
dbaascli listener status
```

Basic status information about the listener, including a summary of listener configuration settings, listening protocol addresses, and a summary of services registered with the listener is displayed.

dbaascli listener stop

The `listener stop` subcommand of the `dbaascli` utility is used to stop the listener.

Execute this command as the `oracle` user.

```
dbaascli listener stop
```

dbaascli netsec config

The `netsec config` subcommand of the `dbaascli` utility is used to configure Oracle Net encryption and integrity settings.

By default, database deployments on Database Cloud Service are configured to enable native Oracle Net encryption and integrity. You can use the `netsec config` subcommand to change Oracle Net encryption and integrity settings. For detailed information on Oracle Net encryption and integrity, see "Configuring Network Data Encryption and Integrity" in *Oracle Database Security Guide* for Release 12.2, 12.1 or 11.2.

If you only want to configure encryption settings, see [dbaascli netsec config encryption](#). If you only want to configure integrity settings, see [dbaascli netsec config integrity](#).

Execute this command as the `oracle` user.

```
dbaascli netsec config
  --encryption_methods algorithm[,algorithm]...
  --encryption_target client|server
  --encryption_type accepted|rejected|requested|required
  --integrity_clevel accepted|rejected|requested|required
  --integrity_methods algorithm[,algorithm]...
  --integrity_target client|server
```

Options of this subcommand are as follows.

Option	Description
<code>encryption_methods</code> <code>algorithm[,algorithm]...</code>	The <code>encryption_methods</code> option is used to specify the encryption algorithm(s). Valid values are: AES128, AES192, and AES256.
<code>encryption_target</code> <code>client server</code>	The <code>encryption_target</code> option is used to specify whether the encryption setting applies to the client or server. Use <code>server</code> .

Option	Description
encryption_type accepted rejected requested required	<p>The <code>encryption_type</code> option is used to specify the action to take when negotiating encryption.</p> <ul style="list-style-type: none"> • <code>rejected</code>—Enter this value if you do not elect to enable data encryption, even if required by the client. <p>In this scenario, this side of the connection specifies that data encryption is not permitted. If the client side is set to <code>required</code>, the connection terminates with error message ORA-12650. If the client side is set to <code>requested</code>, <code>accepted</code> or <code>rejected</code>, the connection continues without error and without data encryption enabled.</p> <ul style="list-style-type: none"> • <code>accepted</code>—Select this value to enable data encryption if required or requested by the client. <p>In this scenario, this side of the connection does not require data encryption, but it is enabled if the client side is set to <code>required</code> or <code>requested</code>. If the client side is set to <code>required</code> or <code>requested</code>, and an encryption algorithm match is found, the connection continues without error and with data encryption enabled. If the client side is set to <code>required</code> and no algorithm match is found, the connection terminates with error message ORA-12650.</p> <p>If the client side is set to <code>requested</code> and no algorithm match is found, or if the client side is set to <code>accepted</code> or <code>rejected</code>, the connection continues without error and without data encryption enabled.</p> <ul style="list-style-type: none"> • <code>requested</code>—Select this value to enable data encryption if the client permits it. <p>In this scenario, this side of the connection specifies that data encryption is desired but not required. Data encryption is enabled if the client side specifies <code>accepted</code>, <code>requested</code>, or <code>required</code>. There must be a matching algorithm available, otherwise data encryption is not enabled. If the client side specifies <code>required</code> and there is no matching algorithm, the connection fails.</p> <ul style="list-style-type: none"> • <code>required</code>—Select this value to enable data encryption or preclude the connection. <p>In this scenario, this side of the connection specifies that data encryption must be enabled. The connection fails if the client side specifies <code>rejected</code> or if there is no compatible algorithm.</p>

Option	Description
integrity_level accepted rejected requested required	<p>The <code>integrity_level</code> option is used to specify the checksum level.</p> <ul style="list-style-type: none"> • <code>rejected</code>—Enter this value if you do not elect to enable data integrity, even if required by the client. <p>In this scenario, this side of the connection specifies that data integrity is not permitted. If the client side is set to <code>required</code>, the connection terminates with error message ORA-12650. If the client side is set to <code>requested</code>, <code>accepted</code> or <code>rejected</code>, the connection continues without error and without data integrity enabled.</p> <ul style="list-style-type: none"> • <code>accepted</code>—Select this value to enable data integrity if required or requested by the client. <p>In this scenario, this side of the connection does not require data integrity, but it is enabled if the client side is set to <code>required</code> or <code>requested</code>. If the client side is set to <code>required</code> or <code>requested</code>, and an integrity algorithm match is found, the connection continues without error and with data integrity enabled. If the client side is set to <code>required</code> and no algorithm match is found, the connection terminates with error message ORA-12650.</p> <p>If the client side is set to <code>requested</code> and no algorithm match is found, or if the client side is set to <code>accepted</code> or <code>rejected</code>, the connection continues without error and without data integrity enabled.</p> <ul style="list-style-type: none"> • <code>requested</code>—Select this value to enable data integrity if the client permits it. <p>In this scenario, this side of the connection specifies that data integrity is desired but not required. Data integrity is enabled if the client side specifies <code>accepted</code>, <code>requested</code>, or <code>required</code>. There must be a matching algorithm available, otherwise data integrity is not enabled. If the client side specifies <code>required</code> and there is no matching algorithm, the connection fails.</p> <ul style="list-style-type: none"> • <code>required</code>—Select this value to enable data integrity or preclude the connection. <p>In this scenario, this side of the connection specifies that data integrity must be enabled. The connection fails if the client side specifies <code>rejected</code> or if there is no compatible algorithm.</p>
integrity_methods <code>algorithm[,algorithm...]</code>	The <code>integrity_methods</code> option is used to specify the integrity algorithm. Valid values are: <code>SHA1</code> , <code>SHA256</code> , <code>SHA384</code> , and <code>SHA512</code> . <code>SHA1</code> is the only algorithm supported by Oracle Database 11g.
integrity_target client server	The <code>integrity_target</code> option is used to specify whether the integrity setting applies to the client or server. Use <code>server</code> .

dbaascli netsec config encryption

The `netsec config encryption` subcommand of the `dbaascli` utility is used to configure Oracle Net encryption settings.

By default, database deployments on Database Cloud Service are configured to enable native Oracle Net encryption and integrity. You can use the `netsec config encryption` subcommand to change Oracle Net encryption settings. See "Configuring Network Data Encryption and Integrity" in *Oracle Database Security Guide* for Release [12.2](#), [12.1](#) or [11.2](#) for detailed information on Oracle Net encryption.

Execute this command as the `oracle` user.

```
dbaascli netsec config encryption
  --methods algorithm[,algorithm]...
  --target client|server
  --type accepted|rejected|requested|required
```

Options of this subcommand are as follows.

Option	Description
<code>methods</code> <code>algorithm[,algorithm]...</code>	The <code>methods</code> option is used to specify the encryption algorithm(s). Valid values are: AES128, AES192, and AES256.
<code>target</code> <code>client server</code>	The <code>target</code> option is used to specify whether the encryption setting applies to the client or server. Use <code>server</code> .

Option	Description
type accepted rejected requested required	<p>The type option is used to specify the action to take when negotiating encryption.</p>
	<ul style="list-style-type: none"> • rejected—Enter this value if you do not elect to enable data encryption, even if required by the client.
	<p>In this scenario, this side of the connection specifies that data encryption is not permitted. If the client side is set to required, the connection terminates with error message ORA-12650. If the client side is set to requested, accepted or rejected, the connection continues without error and without data encryption enabled.</p>
	<ul style="list-style-type: none"> • accepted—Select this value to enable data encryption if required or requested by the client.
	<p>In this scenario, this side of the connection does not require data encryption, but it is enabled if the client side is set to required or requested. If the client side is set to required or requested, and an encryption algorithm match is found, the connection continues without error and with data encryption enabled. If the client side is set to required and no algorithm match is found, the connection terminates with error message ORA-12650.</p>
	<p>If the client side is set to requested and no algorithm match is found, or if the client side is set to accepted or rejected, the connection continues without error and without data encryption enabled.</p>
	<ul style="list-style-type: none"> • requested—Select this value to enable data encryption if the client permits it.
	<p>In this scenario, this side of the connection specifies that data encryption is desired but not required. Data encryption is enabled if the client side specifies accepted, requested, or required. There must be a matching algorithm available, otherwise data encryption is not enabled. If the client side specifies required and there is no matching algorithm, the connection fails.</p>
	<ul style="list-style-type: none"> • required—Select this value to enable data encryption or preclude the connection.
	<p>In this scenario, this side of the connection specifies that data encryption must be enabled. The connection fails if the client side specifies rejected or if there is no compatible algorithm.</p>

dbaascli netsec config integrity

The `netsec config integrity` subcommand of the `dbaascli` utility is used to configure Oracle Net integrity settings.

By default, database deployments on Database Cloud Service are configured to enable native Oracle Net integrity. You can use the `netsec config integrity` subcommand to change Oracle Net integrity settings. See "Configuring Network Data Encryption and Integrity" in *Oracle Database Security Guide* for Release 12.2, 12.1 or 11.2 for detailed information on Oracle Net encryption and integrity.

Execute this command as the oracle user.

```
dbaascli netsec config integrity
  --clevel accepted|rejected|requested|required
  --methods algorithm[,algorithm]...
  --target client|server
```

Options of this subcommand are as follows.

Option	Description
clevel accepted rejected requested required	<p>The clevel option is used to specify the checksum level.</p> <ul style="list-style-type: none">• rejected—Enter this value if you do not elect to enable data integrity, even if required by the client.• In this scenario, this side of the connection specifies that data integrity is not permitted. If the client side is set to required, the connection terminates with error message ORA-12650. If the client side is set to requested, accepted or rejected, the connection continues without error and without data integrity enabled.• accepted—Select this value to enable data integrity if required or requested by the client.• In this scenario, this side of the connection does not require data integrity, but it is enabled if the client side is set to required or requested. If the client side is set to required or requested, and an integrity algorithm match is found, the connection continues without error and with data integrity enabled. If the client side is set to required and no algorithm match is found, the connection terminates with error message ORA-12650.• If the client side is set to requested and no algorithm match is found, or if the client side is set to accepted or rejected, the connection continues without error and without data integrity enabled.• requested—Select this value to enable data integrity if the client permits it.• In this scenario, this side of the connection specifies that data integrity is desired but not required. Data integrity is enabled if the client side specifies accepted, requested, or required. There must be a matching algorithm available, otherwise data integrity is not enabled. If the client side specifies required and there is no matching algorithm, the connection fails.• required—Select this value to enable data integrity or preclude the connection.• In this scenario, this side of the connection specifies that data integrity must be enabled. The connection fails if the client side specifies rejected or if there is no compatible algorithm.
methods algorithm[,algorithm]...	<p>The methods option is used to specify the integrity algorithm. Valid values are: SHA1, SHA256, SHA384, and SHA512. SHA1 is the only algorithm supported by Oracle Database 11g.</p>

Option	Description
target client server	The target option is used to specify whether the integrity setting applies to the client or server. Use server.

dbaascli netsec status

The `netsec status` subcommand of the `dbaascli` utility is used to display information about network encryption and network integrity configuration.

Execute this command as the `oracle` user.

```
dbaascli netsec status [encryption|integrity]
```

If no options are specified, the utility will display information about network encryption and network integrity configuration.

Options of this subcommand are as follows.

Option	Description
encryption	The encryption option is used to display information about network encryption configuration.
integrity	The integrity option is used to display information about network integrity configuration.

dbaascli orec keep list

The `orec latest` subcommand of the `dbaascli` utility is used to list the available long-term backups.

You must execute this command as the `root` user.

```
dbaascli orec --args -keep -list
```

dbaascli orec keep tag

The `orec keep tag` subcommand of the `dbaascli` utility is used to restore a specific long-term backup and perform recovery.

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

```
dbaascli orec --args -keep -tag backup-tag
```

Options of this subcommand are as follows.

Option	Description
<code>-tag</code> <i>backup-tag</i>	The tag option is used to supply the backup tag of the long-term backup that should be restored for the recovery operation.

dbaascli orec latest

The `orec latest` subcommand of the `dbaascli` utility is used to restore the most recent backup and perform complete recovery.

You must execute this command as the `root` user.

```
dbaascli orec --args -latest
```

dbaascli orec list

The `orec latest` subcommand of the `dbaascli` utility is used to list the available normal backups.

You must execute this command as the `root` user.

```
dbaascli orec --args -list
```

dbaascli orec pitr

The `orec pitr` subcommand of the `dbaascli` utility is used to restore a specific normal backup and perform recovery.

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

```
dbaascli orec --args -pitrbackup-tag
```

Options of this subcommand are as follows.

Option	Description
<code>-pitr backup-tag</code>	The <code>pitr</code> option is used to supply the backup tag of the backup that should be restored for the recovery operation.

dbaascli orec scn

The `orec scn` subcommand of the `dbaascli` utility is used to restore the most recent backup and perform recovery through the specified system change number (SCN).

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

```
dbaascli orec --args -scn SCN
```

Options of this subcommand are as follows.

Option	Description
<code>-scn SCN</code>	The <code>scn</code> option is used to supply the system change number (SCN) for the end point of the recovery operation.

dbaascli tde rotate masterkey

The `tde rotate masterkey` subcommand of the `dbaascli` utility is used to change (rotate) the master encryption key.

Execute this command as the `oracle` user.

```
dbaascli tde rotate masterkey
```

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

dbaascli tde status

The `tde status` subcommand of the `dbaascli` utility is used to view information about the software keystore used in tablespace encryption.

Execute this command as the `oracle` user.

```
dbaascli tde status
```

Output from the command includes the type of keystore and the status of the keystore.

dbaascli tde status

The raccli Utility

The `raccli` utility is provided on Oracle Database Cloud Service deployments that use Oracle Real Application Clusters (RAC) to perform a variety of life-cycle and administration operations.

Using the `raccli` utility, you can perform operations like:

- Backing up the database
- Recovering the database from a backup
- Changing configuration of automatic backups
- Patching the Oracle Database, Grid Infrastructure and cloud tooling software
- Changing the configuration of security features
- Tracking the progress and completion of long-running operations performed as asynchronous jobs

To use the `raccli` utility:

1. Connect as the `opc` user to a compute node associated with the Database Cloud Service deployment.

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

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

```
$ raccli subcommand subcommand-options
```

3. When you are finished using the `raccli` utility, disconnect from the compute node:

```
$ exit
```

The `raccli` utility supports these subcommands:

`apply patch`: applies a patch to the Grid Infrastructure or Oracle Database home.

`create backup`: creates a backup of the configuration files, inventory configuration files, Grid Infrastructure and Oracle Database homes, database, and tooling metadata.

`create recovery`: initiates database recovery.

`describe job`: provides information about the progress and status of a long-running operation.

`describe system`: provides information about the installation of Grid Infrastructure, Oracle Database, and the RDK cloud tooling.

`list backup`: provides a list and status of all the backup jobs that have been submitted through the `raccli` utility.

- `list backupconfig`: provides a list of the backup configuration settings.
- `list jobs`: provides a list of all the jobs that have been submitted through the `raccli` utility.
- `list recovery`: provides information about recovery jobs.
- `update backupconfig`: updates the backup configuration.
- `update netsec`: updates the Oracle Net security configuration.
- `update rdk`: updates the cloud tooling on the Database Cloud Service deployment.
- `update tde`: enables transparent data encryption (TDE) and rotates the TDE key.

raccli apply patch

The `apply patch` subcommand of the `raccli` utility is used to apply a patch to the Grid Infrastructure or Oracle Database home on an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli apply patch {-gi|-db} [-tag tag-name] [-local]
```

This subcommand runs asynchronously. That is, it creates a job to apply the patch, reports the job ID of the created job, and then exits. To track the progress of the job to its completion, use the `raccli describe job` command.

If the subcommand fails, it reports a FAILURE status and provides a message describing the reason for the failure.

Options of this subcommand are as follows.

Option	Description
<code>-gi</code>	Causes <code>raccli</code> to update the Grid Infrastructure home.
<code>-db</code>	Causes <code>raccli</code> to update the Oracle Database home.
<code>-tag tag-name</code>	The name of the patch to apply. If this option is omitted, the latest patch available is applied.
<code>-local</code>	Causes <code>raccli</code> to update the Grid Infrastructure or Oracle Database home only on the current (local) compute node. The local node is taken offline, patched, and then brought back online. This enables you to test the patch before applying it to a second node. To apply it to a second node, you must execute the command on that node. If this option is omitted, the Grid Infrastructure or Oracle Database home is updated on all compute nodes associated with the Database Cloud Service deployment. The node from which you run the command is taken offline, patched, and then brought back online. Then the second node is taken offline, patched, and brought back online.

raccli create backup

The `create backup` subcommand of the `raccli` creates a backup of the configuration files, inventory configuration files, Grid Infrastructure and Oracle

Database home, database, and tooling metadata of an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli create backup [-tag tag-name]
```

This subcommand runs asynchronously. That is, it creates a job to perform the backup, reports the job ID of the created job, and then exits. To track the progress of the job to its completion, use the [raccli describe job](#) command.

If the subcommand fails, it reports a FAILURE status and provides a message describing the reason for the failure.

Options of this subcommand are as follows.

Option	Description
<code>-tag tag-name</code>	The name of the backup job. This is not an RMAN tag.

Example

Here is an example of the `create backup` subcommand. The name of the backup in this example is `backuptest`. You can view details of the backup job by using the [raccli describe job](#) subcommand. You can view details of the backup by using the [raccli list backup](#) subcommand.

```
[opc@example1 ~]$ raccli create backup -tag backuptest
{
  "joburi" : "http://localhost:7070/dcs/7/responses",
  "requestStatus" : "SUCCESS",
  "jobid" : "7"
}
[opc@example1 ~]$
```

raccli create recovery

The `create recovery` subcommand of the `raccli` initiates database recovery of an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli create recovery [-tag tag-name][-latest][-pitr][-scn SCN][-timestamp time]
```

This subcommand runs asynchronously. That is, it creates a job to perform the recovery, reports the job ID of the created job, and then exits. To track the progress of the job to its completion, use the [raccli describe job](#) command.

If the subcommand fails, it reports a FAILURE status and provides a message describing the reason for the failure.

Options of this subcommand are as follows.

Option	Description
<code>-tag tag-name</code>	The name of the recovery job. This is not an RMAN tag. If this option is omitted, a default name of "auto" is given.
<code>-latest</code>	Indicates that a complete database recovery should be performed.

Option	Description
<code>-pitr</code>	Indicates that a database point-in-time (incomplete) recovery should be performed. The recovery end point is specified by the <code>-scn</code> or <code>-timestamp</code> option.
<code>-scn SCN</code>	The system change number (SCN) for the end point of the recovery.
<code>-timestamp time</code>	The time for the end point of the recovery. The format is <code>MM/DD/YYYY HH24:MI:SS</code> .

Example

Here is an example of the `create recovery` subcommand. The name of the recovery job in this example is `recovertest`. This is an example of a request for a complete database recovery operation. You can use the job ID displayed in the output as input to the `raccli describe job` command to track the job's progress.

```
[opc@example1 ~]$ raccli create recovery -tag recovertest -latest
{
  "joburi" : "http://localhost:7070/dcs/8/responses",
  "requestStatus" : "SUCCESS",
  "jobid" : "8"
}
[opc@example1 ~]$
```

raccli describe job

The `describe job` subcommand of the `raccli` utility provides information about the progress and status of a long-running operation.

```
raccli describe job job-id
```

The output of this subcommand depends on the kind of operation that the given job is.

Options of this subcommand are as follows.

Option	Description
<code>job-id</code>	The ID of the job you want to view detailed information about.

Example

Here is an example of the `describe job` subcommand. This example shows details about the backup job in the [raccli create backup](#) example.

```
opc@example1 ~]$ raccli describe job 7
```

```
{
  "requestStatus" : "SUCCESS",
  "jobStatus" : "SUCCESS",
  "message" : null,
  "response" : [ {
```

```

    "startTime" : "Thu Oct 15 18:37:38 UTC 2015",
    "endTime" : "Thu Oct 15 18:37:39 UTC 2015",
    "status" : "SUCCESS",
    "taskId" : "TaskZJsonRpcExt_6939",
    "taskResult" : "Resource { id: 1444934258987, name: null, description: null }",
    "taskName" : "DB Config files backup",
    "taskDescription" : null
  },
  {
    "startTime" : "Thu Oct 15 18:37:39 UTC 2015",
    "endTime" : "Thu Oct 15 18:38:10 UTC 2015",
    "status" : "SUCCESS",
    "taskId" : "TaskZJsonRpcExt_6941",
    "taskResult" : "Resource { id: 1444934260034, name: null, description: null }",
    "taskName" : "Database Backup",
    "taskDescription" : null
  },
  {
    "startTime" : "Thu Oct 15 18:38:11 UTC 2015",
    "endTime" : "Thu Oct 15 18:38:11 UTC 2015",
    "status" : "SUCCESS",
    "taskId" : "TaskZJsonRpcExt_6943",
    "taskResult" : "Resource { id: 1444934291029, name: null, description: null }",
    "taskName" : "Persisting Backup metadata",
    "taskDescription" : null
  }
]
[opc@example1 ~]$
```

raccli describe system

The `describe system` subcommand of the `raccli` utility provides information about the installation of Grid Infrastructure, Oracle Database, and the RDK cloud tooling on an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli describe system
```

Example

Here is an example of the `describe system` subcommand.

```
[opc@example1 ~]$ raccli describe system

"serviceName": "example"
{
  "NodeName": "example1"
  {
    "componentType": "GridHome"
    "componentName": "OraGrid12102"
    "componentVersion": "12.1.0.2.0(21297657, 20299018)"
  }
  {
    "componentType": "DatabaseHome"
    "componentName": "OraDB12102_home1"
    "componentVersion": "12.1.0.2.10 (21125181)"
  }
  {
    "componentType": "RDK"
    "componentName": "RDK"
    "componentVersion": "15.4.1.0.0"
  }
}
```

```
raccli list backup
```

```
{  
    "NodeName": "example2"  
    {  
        "componentType": "GridHome"  
        "componentName": "OraGrid12102"  
        "componentVersion": "12.1.0.2.0(21297657, 20299018)"  
    }  
    {  
        "componentType": "DatabaseHome"  
        "componentName": "OraDB12102_home1"  
        "componentVersion": "12.1.0.2.10 (21125181)"  
    }  
    {  
        "componentType": "RDK"  
        "componentName": "RDK"  
        "componentVersion": "15.4.1.0.0"  
    }  
}  
[opc@example1 ~]$
```

raccli list backup

The `list backup` subcommand of the `raccli` utility provides a list and status of all the backup jobs on an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli list backup
```

Example

Here is an example of the `list backup` subcommand.

```
[opc@example1 ~]# raccli list backup  
[ {  
    "createTimeStamp" : "09/13/2016 03:50:02 UTC",  
    "id" : 1473738602916,  
    "updatedTimestamp" : "09/13/2016 03:52:08 UTC",  
    "jobId" : "2",  
    "tagName" : "auto",  
    "name" : "rdbaas_backup",  
    "backupComponentTypes" : "DBConfig,Database",  
    "backupType" : "REGULAR",  
    "type" : "Backup",  
    "waitLock" : false  
, {  
    "createTimeStamp" : "09/14/2016 03:50:01 UTC",  
    "id" : 1473825001959,  
    "updatedTimestamp" : "09/14/2016 03:51:24 UTC",  
    "jobId" : "3",  
    "tagName" : "auto",  
    "name" : "rdbaas_backup",  
    "backupComponentTypes" : "DBConfig,Database",  
    "backupType" : "REGULAR",  
    "type" : "Backup",  
    "waitLock" : false  
, {  
    "createTimeStamp" : "09/18/2016 03:50:02 UTC",  
    "id" : 1474170602497,  
    "updatedTimestamp" : "09/18/2016 03:52:01 UTC",  
}
```

```

"jobId" : "8",
"tagName" : "auto",
"name" : "rdbaas_backup",
"backupComponentTypes" : "GIConfig,OSConfig,DBConfig,Database,DerbyDB",
"backupType" : "REGULAR",
"type" : "Backup",
"waitLock" : false
} ]

```

[opc@example1 ~]#

raccli list backupconfig

The `list backupconfig` subcommand of the `raccli` utility provides a list of the backup configuration settings for an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

`raccli list backupconfig`

Example

Here is an example of the `list backupconfig` subcommand.

```

[opc@example1 ~]$ raccli list backupconfig

{
  "updatedTimestamp" : "2016-09-12, 09:24:33",
  "id" : 1473672273813,
  "name" : "rdbaas_backup_config",
  "description" : null,
  "createTimeStamp" : "2016-09-12, 09:24:33",
  "type" : "BackupConfig",
  "dbHomeBackup" : false,
  "giHomeBackup" : false,
  "backupOsConfigFiles" : true,
  "backupGiConfigFiles" : true,
  "backupDerby" : true,
  "diskEnabled" : false,
  "ossEnabled" : false,
  "nfsEnabled" : true,
  "diskRecoveryWindow" : 7,
  "ossRecoveryWindow" : 30,
  "nfsRecoveryWindow" : 7,
  "cronDate" : "03:50",
  "osConfigFilesRef" : "/opt/oracle/dcs/rdbaas/config/oscfg.spec",
  "giConfigFilesRef" : "/opt/oracle/dcs/rdbaas/config/gicfg.spec",
  "dbConfigFilesRef" : "/opt/oracle/dcs/rdbaas/config/dbcfg.spec",
  "nfsBackupLocation" : "192.168.64.1:/export/EPC/scat03/psmbackup",
  "cloudStorageServiceName" : null,
  "cloudStorageIdentityDomain" : null,
  "cloudStorageUser" : null,
  "cloudStorageHost" : null,
  "cloudStoreContainer" : null,
  "cloudStorageContainerUrl" : "nullnull-null/null"
}

```

[opc@example1 ~]#

raccli list jobs

The `list jobs` subcommand of the `raccli` utility provides a list of all the jobs on an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli list jobs
```

Example

Here is an example of the `list jobs` subcommand.

```
[opc@example1 ~]$ raccli list jobs
```

Job Id	Job Name	Status	Message
1	service creation	Success	null
2	backup creation	Success	null
3	backup creation	Success	null
4	backup creation	Success	null
5	backup creation	Success	null
6	backup creation	Success	null
7	backup creation	Success	null
8	Create recovery	Running	null

```
[opc@example1 ~]$
```

raccli list recovery

The `list recovery` subcommand of the `raccli` utility provides information about recovery jobs.

```
raccli list recovery [-id recovery-id][-tag recovery-tagname]
```

Options of this subcommand are as follows.

Option	Description
<code>-id recovery-id</code>	The system-generated ID of the recovery job. Use this option to display information about a specific recovery job.
<code>-tag recovery-tagname</code>	The name supplied when the recovery job was submitted. Use this option to display information about a specific recovery job.

Example

Here is an example of the `list recovery` subcommand. The output in this example shows the detail of the recovery performed in the [raccli create recovery](#) example.

```
[opc@example1 ~]$ raccli list recovery

[ {
  "updatedTimestamp" : "2015-10-15, 18:47:17",
  "id" : 1444934529920,
  "name" : "rdbaas_recovery",
  "description" : null,
  "createTimeStamp" : "2015-10-15, 18:42:09",
  "type" : "Recovery",
  "recoveryComponents" : [ {
    "updatedTimestamp" : "2015-10-15, 18:47:17",
```

```

"id" : 1444934530186,
"name" : "rdbaas_recovery_database",
"description" : null,
"createTimeStamp" : "2015-10-15, 18:42:10",
"type" : null,
"component_type" : "Database",
"recoveryParams" : [ {
    "updatedTimestamp" : "2015-10-15, 18:42:10",
    "id" : 1444934530485,
    "name" : "rdbaas_recovery_database_parameters",
    "description" : null,
    "createTimeStamp" : "2015-10-15, 18:42:10",
    "type" : null,
    "parameter" : "latest",
    "value" : "true"
} ],
"recoveryTag" : "backuptest",
"endTime" : null,
"jobId" : "8"
}
[opc@example1 ~]$

```

raccli update backupconfig

The `update backupconfig` subcommand of the `raccli` utility updates the backup configuration on an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli update backupconfig -params parameter-string
```

This subcommand runs synchronously.

Options of this subcommand are as follows.

Option	Description
<code>-params <i>parameter-string</i></code>	The parameters to update.

Parameters

The parameter string is a single JSON object that specifies the backup configuration settings to be updated.

The following table describes the parameters that can be specified in this JSON string.

Parameter	Description
<code>backupDerby</code>	Controls whether a backup of the metadata store for each RAC compute node is taken. Enter <code>true</code> (default) or <code>false</code> .
<code>backupGiConfigFiles</code>	Controls whether a backup of the Grid Infrastructure configuration files is taken. Enter <code>true</code> (default) or <code>false</code> .
<code>backupOsConfigFiles</code>	Controls whether a backup of operating system configuration files is taken. Enter <code>true</code> (default) or <code>false</code> .

Parameter	Description
dbHomeBackup	Controls whether a backup of the Oracle Database home is taken. Enter <code>true</code> or <code>false</code> (default).
cronDate	Specifies the time (using 24-hour, HH:MM format) when daily backups are to occur. For example, 02:45 is 2:45 AM, and 14:45 is 2:45 PM. The default value is 01:00.
nfsBackupLocation	The path of the NFS remote backup location where backups are to be stored. This path must be of the format <code>host-name: absolute-path</code> or <code>ip-address: absolute-path</code> .

Example

Here is an example of the `update backupconfig` subcommand. You can use the [raccli list backupconfig](#) subcommand to view your changes.

```
[opc@example1 ~]$ raccli update backupconfig -params '{"cronDate" : "02:45"}'

{
  "requestStatus" : "SUCCESS",
  "jobStatus" : "SUCCESS",
  "message" : null,
  "response" : [ ]
}
[opc@example1 ~]$
```

raccli update netsec

The `update netsec` subcommand of the `raccli` utility updates the Oracle Net security configuration on an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli update netsec {-encryption|-integrity} {-server|-client} -type type - algorithm algorithm
```

This subcommand runs synchronously.

If the subcommand fails, it reports a FAILURE status and provides a message describing the reason for the failure.

Options of this subcommand are as follows.

Option	Description
<code>-encryption</code>	Controls whether Oracle Net Services data encryption is configured.
<code>-integrity</code>	Controls whether Oracle Net Services data integrity is configured.
<code>-server</code>	Specifies that the Oracle Net Services data encryption or data integrity configuration is for the server. Use this value.
<code>-client</code>	Specifies that the Oracle Net Services data encryption or data integrity configuration is for the client.

Option	Description
<code>-type type</code>	<p>Specifies how Oracle Net Services data encryption or data integrity is negotiated with clients.</p> <p>For data encryption enter one of these values:</p> <ul style="list-style-type: none"> • <code>rejected</code>—Enter this value if you do not elect to enable data encryption, even if required by the client.

In this scenario, this side of the connection specifies that data encryption is not permitted. If the client side is set to `required`, the connection terminates with error message ORA-12650. If the client side is set to `requested`, `accepted` or `rejected`, the connection continues without error and without data encryption enabled.

- `accepted`—Select this value to enable data encryption if required or requested by the client.

In this scenario, this side of the connection does not require data encryption, but it is enabled if the client side is set to `required` or `requested`. If the client side is set to `required` or `requested`, and an encryption algorithm match is found, the connection continues without error and with data encryption enabled. If the client side is set to `required` and no algorithm match is found, the connection terminates with error message ORA-12650.

If the client side is set to `requested` and no algorithm match is found, or if the client side is set to `accepted` or `rejected`, the connection continues without error and without data encryption enabled.

- `requested`—Select this value to enable data encryption if the client permits it.

In this scenario, this side of the connection specifies that data encryption is desired but not required. Data encryption is enabled if the client side specifies `accepted`, `requested`, or `required`. There must be a matching algorithm available, otherwise data encryption is not enabled. If the client side specifies `required` and there is no matching algorithm, the connection fails.

- `required`—Select this value to enable data encryption or preclude the connection.

In this scenario, this side of the connection specifies that data encryption must be enabled. The connection fails if the client side specifies `rejected` or if there is no compatible algorithm.

For data integrity enter one of these values:

- `rejected`—Enter this value if you do not elect to enable data integrity, even if required by the client.

In this scenario, this side of the connection specifies that data integrity is not permitted. If the client side is set to `required`, the connection terminates with error message ORA-12650. If the client side is set to `requested`, `accepted` or `rejected`, the connection continues without error and without data integrity enabled.

Option	Description
	<ul style="list-style-type: none"> accepted—Select this value to enable data integrity if required or requested by the client. In this scenario, this side of the connection does not require data integrity, but it is enabled if the client side is set to required or requested. If the client side is set to required or requested, and an integrity algorithm match is found, the connection continues without error and with data integrity enabled. If the client side is set to required and no algorithm match is found, the connection terminates with error message ORA-12650. If the client side is set to requested and no algorithm match is found, or if the client side is set to accepted or rejected, the connection continues without error and without data integrity enabled. requested—Select this value to enable data integrity if the client permits it. In this scenario, this side of the connection specifies that data integrity is desired but not required. Data integrity is enabled if the client side specifies accepted, requested, or required. There must be a matching algorithm available, otherwise data integrity is not enabled. If the client side specifies required and there is no matching algorithm, the connection fails. required—Select this value to enable data integrity or preclude the connection. In this scenario, this side of the connection specifies that data integrity must be enabled. The connection fails if the client side specifies rejected or if there is no compatible algorithm.
<code>-algorithm</code> <i>algorithm</i>	The algorithm to be used for data encryption or data integrity. For encryption, the choices are AES128, AES192, and AE256. For integrity with Oracle Database 12c, the choices are SHA1, SHA512, SHA384, and SHA25. For integrity with Oracle Database 11g, the only accepted value is SHA1.

raccli update rdk

The `update rdk` subcommand of the `raccli` utility updates the cloud tooling on an Oracle Database Cloud Service database deployment that uses Oracle Real Application Clusters (RAC).

```
raccli update rdk -tag tag-number
```

This subcommand runs asynchronously. That is, it creates a job to update the cloud tooling, reports the job ID of the created job, and then exits. To track the progress of the job to its completion, use the `raccli describe job` command.

If the subcommand fails, it reports a FAILURE status and provides a message describing the reason for the failure.

Options of this subcommand are as follows.

Option	Description
<code>-tag tag-number</code>	The tag of the cloud tooling to update to. For <code>tag-number</code> enter the version of tooling you want to update to without the dots in the version number. For example, to update to 17.1.2 tooling you would enter 1712 .

Example

Here is an example of the `update rdk` subcommand. You can use the [raccli describe job](#) subcommand to view details about the job that is started when you execute this subcommand.

```
[opc@example1 ~]$ raccli update rdk -tag 1712

{
  "jobId" : "10",
  "requestStatus" : "SUCCESS"
}
[opc@example1 ~]$
```

raccli update tde

The `update tde` subcommand of the `raccli` utility provides a way to enable transparent data encryption (TDE) and rotate the TDE key.

```
raccli update tde {-enable|-rotatekey} -passwd password
```

This subcommand runs synchronously.

If the subcommand fails, it reports a FAILURE status and provides a message describing the reason for the failure.

Options of this subcommand are as follows.

Option	Description
<code>-enable</code>	Used to enable the TDE configuration.
<code>-rotatekey</code>	Used to rotate the key.
<code>-passwd password</code>	The password specified when the database deployment was created, unless you have changed the wallet password since the deployment was created.

Example

Here is an example of the `update tde` subcommand. This example rotates (changes) the value of the master key.

```
[opc@example1 ~]$ raccli update tde -rotatekey -passwd Welcome_1

{
  "requestStatus" : "SUCCESS",
  "jobStatus" : "SUCCESS",
  "message" : null,
  "response" : [ ]
```

raccli update tde

```
}
```

```
[opc@example1 ~]$
```