ORACLE UNIVERSITY

Oracle International Academy Use Only

Oracle Database Cloud for Oracle DBAs

Student Guide

D96069GC10

Edition 1.0 | September 2016| D97402

Learn more from Oracle University at oracle.com/education/

ORACLE

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

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle. The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government¿s rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

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

Authors

Dominique Jeunot, Donna K. Keesling, James L. Spiller

Technical Contributors and Reviewers

Jean-François Verrier

Editors

Arijit Ghosh, Vijayalakshmi Narasimhan

Graphic Designer

Maheshwari Krishnamurthy

Publisher

Sumesh Koshy

Oracle Internal & Oracle Academy Use Only

Table of Contents

Oracle Internal & Oracle Academy Use Only

Oracle Internal & Oracle Academy Use Only

Oracle Internal & Oracle Academy Use Only

Getting Started

- Course Objectives
- Lab Environment

Oracle Internal & Oracle Academy Use Only

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

Course Objectives

After completing this course, you should be able to:

- Understand the main features of Oracle Database Cloud Service (DBCS)
- Subscribe to Oracle Cloud services
- Create your own database deployment
- Administer your database deployment
- Connect to your database deployment through different ways
- Ensure security between database deployments
- Administer Oracle database in database deployment
- Monitor database deployment
- Migrate on-premises databases to your database deployment
- Monitor performance in database deployment

Oracle Internal & Oracle Academy Use Only

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

Lab Environment

All students have access to an individual identity domain that allows them to access the Oracle Public cloud and create and manage their own DBCS environment.

Attendees receive an email with directions to access their individual identity domains the week before the class.

Oracle Internal & Oracle Academy Use Only

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

Database Deployment

Oracle Internal & Oracle Academy Use Only

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

Lesson 1 Overview of Cloud Services

After completing this lesson, you should be able to:

- Explain what the Cloud Services are useful for
- Subscribe to Oracle Cloud services

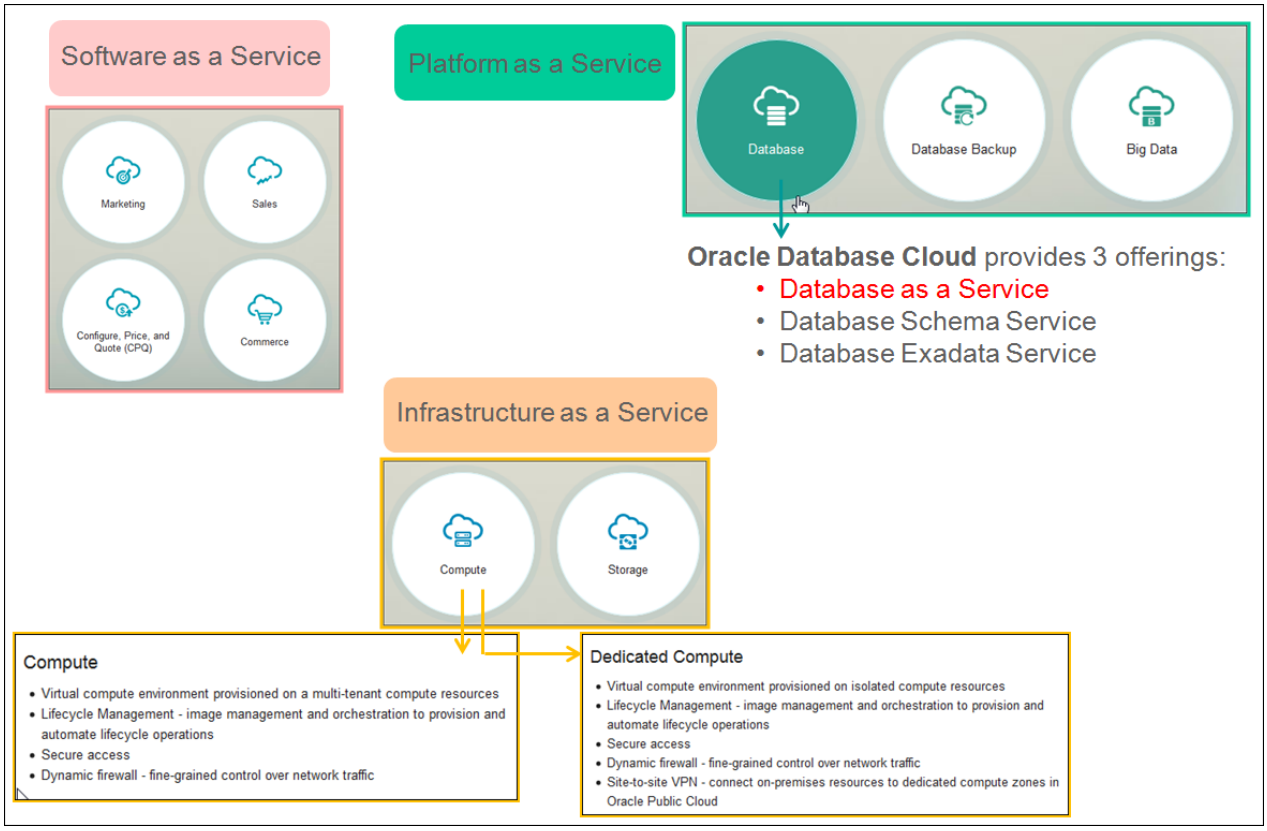
Topics:

- Cloud Services
- DBCS Offerings
- Subscribe to Oracle Cloud Services

Oracle Internal & Oracle Academy Use Only

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

Cloud Services



Oracle Cloud is an enterprise cloud for business. Oracle Cloud offers self-service business applications delivered on an integrated development and deployment platform with tools to extend and create new services rapidly.

With **Software as a Service** (SaaS), modern cloud applications from Oracle help you re-imagine your business. The best-of-breed SaaS applications in Oracle Cloud are integrated with social, mobile, and analytic capabilities to help you deliver the experiences customers expect, the talent to succeed, and the performance the market demands.

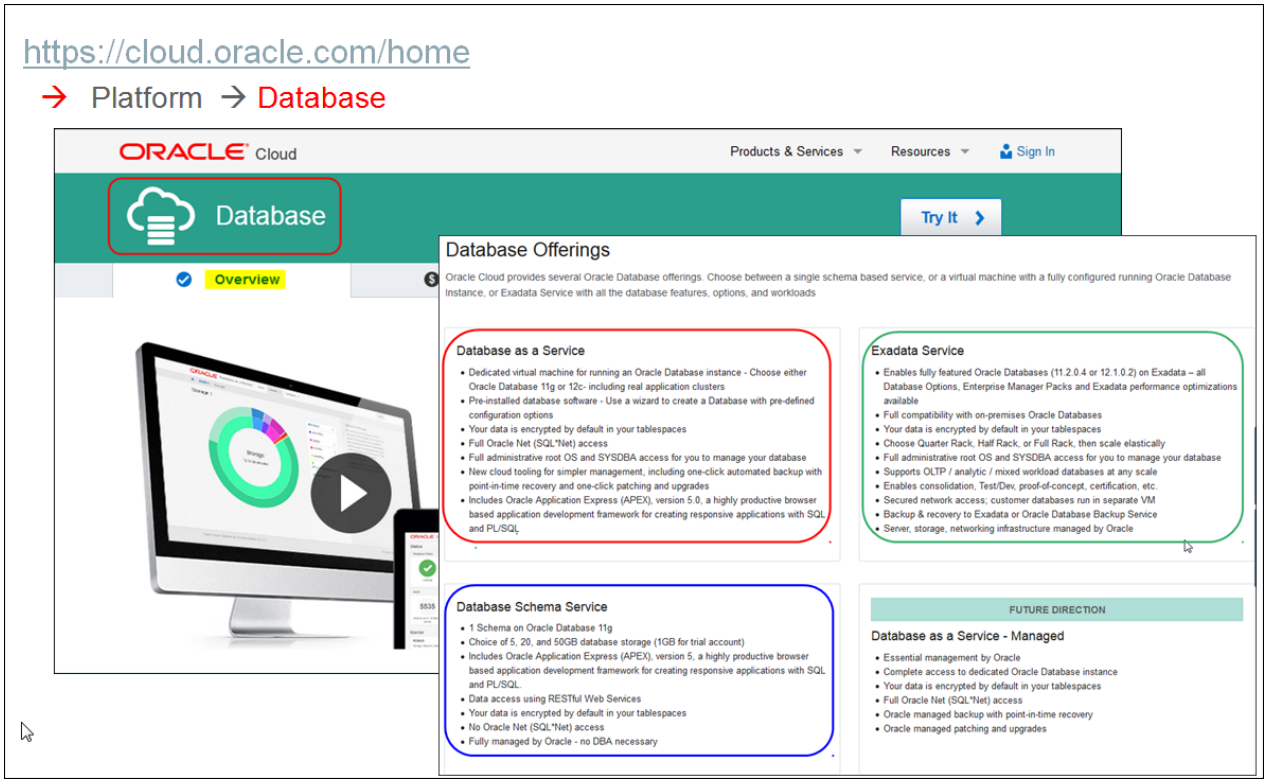
Platform services (PaaS) help enterprise IT and independent software vendor (ISV) developers rapidly build and deploy rich applications or extend Software as a Service (SaaS) applications using an enterprise-grade cloud platform based on the industry's #1 database and application server.

For other services, refer to the *Oracle Cloud web site*.

Oracle Internal & Oracle Academy Use Only

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

DBCS Offerings



Oracle Database Cloud Service (DBCS) allows you to have servers and databases allocated for your needs in the cloud.

You no longer need to purchase servers, disks, memory and have system administrators manage the servers.

Oracle Database Cloud Service provides different offerings:

- **Database Schema Service**: A dedicated schema with a complete development and deployment platform managed by Oracle
- **Database as a Service**: Dedicated database instances with full administrative control
- **Exadata Service**: Full Oracle Databases hosted on an Oracle Exadata Database Machine inside the Oracle Cloud

To get the most recent DBCS offerings, refer to the *Oracle Database Cloud - Database as a Service* Or *About Oracle Database Cloud Service Offerings*

Oracle Internal & Oracle Academy Use Only

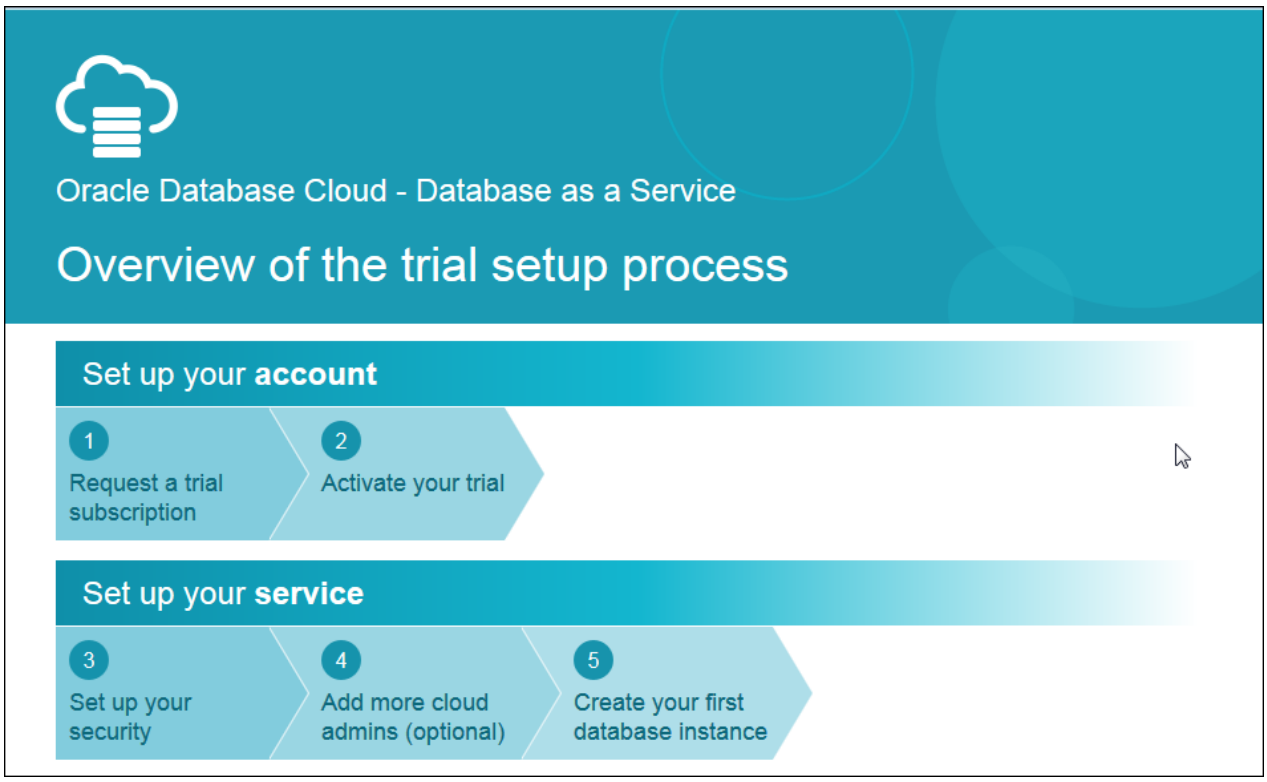
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Subscribe to Oracle Cloud Services

To get access to Oracle Cloud Database as a Service, perform the following steps:

1. **Order** one of the subscription types:
   - A free trial subscription to an Oracle Cloud Service: Follow the steps described in *Requesting a Trial Subscription* .
   - A paid subscription: Follow the steps described in *Buying a Nonmetered Oracle Cloud Service* or *Buying a Metered Oracle Cloud Service* .
2. **Activate** the service.
   - The free trial subscription: Follow the steps described in *Activating Your Trial Subscription* .
   - The paid subscription: Follow the steps described in *Activating the Oracle Cloud Service* .
3. **Verify** that the service is running.

Click the *trial subscription flow:*



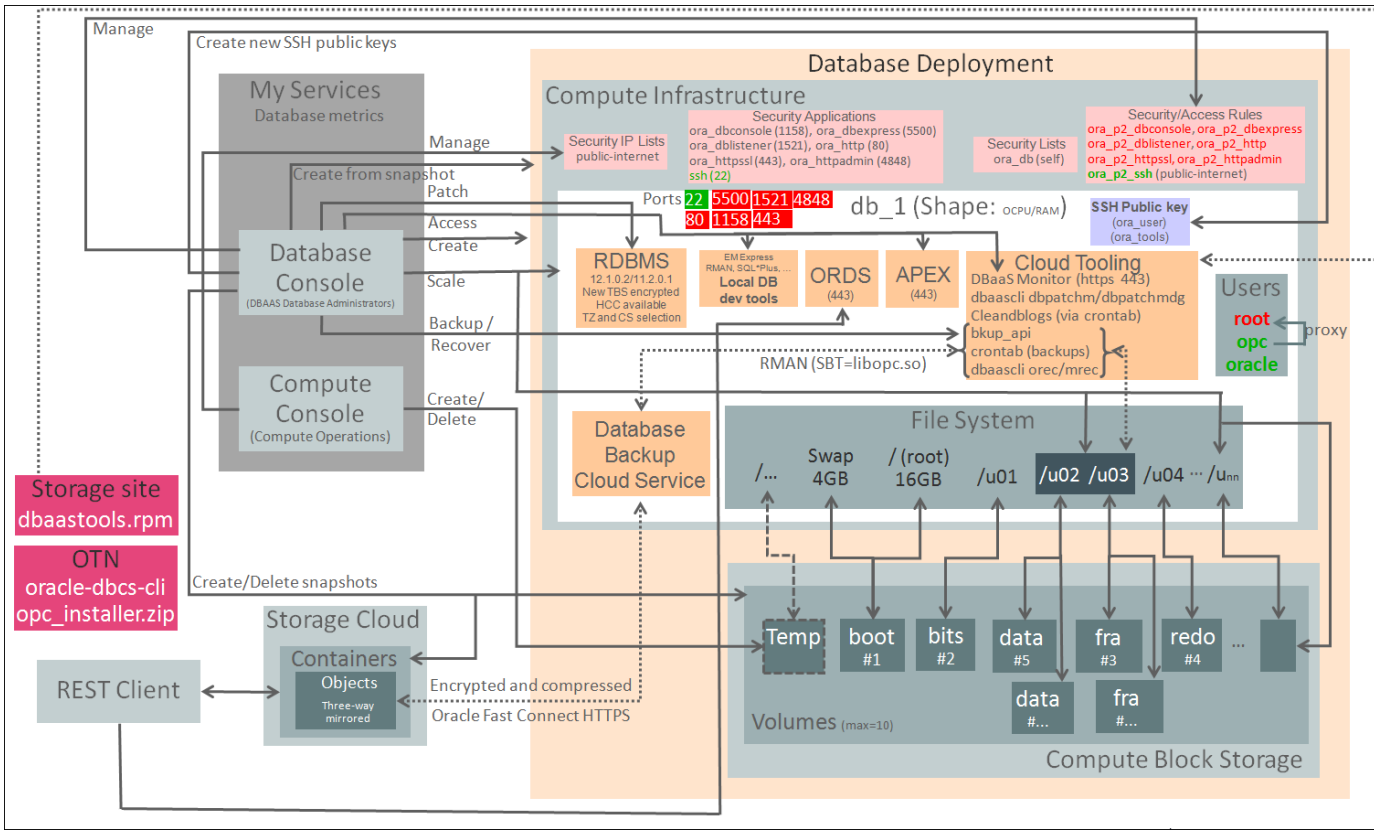
Oracle Internal & Oracle Academy Use Only

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

Architecture Diagram

This infographic shows you at a high level the various Oracle Database Cloud single-instance service components and their interactions. This service is currently referred to as Database Deployment. It was formerly known as "Database as a Service."

You can refer to the *DBCS_Arch.pdf* or the following video titled *Oracle Database Cloud: DBCS architecture for DBAs* .



From now on we will refer to this service as Database Cloud Service or DBCS.
We do not get into the details of how you can get an Oracle Cloud account and we suppose you are already done with this phase.
As an Identity Domain Administrator or a Service Administrator, your main entry point to manage your services is what we call the My Services web console or dashboard.
From that web page you can get an overall information about all services running in your identity domain.

From the My Services page, you can open what is called the Oracle Database Cloud Service Console: another web page allowing you to manage your DBCS environment.
Note that the My Services page computes database metrics periodically.

From the Database Console, you can create, delete, start, stop, restart, patch, backup, recover, add new ssh public keys, or scale up and down what are called DBCS instances or Database Deployments: a combination of resources used to give you access to an Oracle Database.
To be able to get to both the My Services and Database consoles, you need certain privileges granted to your Oracle Database Cloud Service account through roles.

Oracle Internal & Oracle Academy Use Only

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

For Database as a Service, one role exists and is automatically granted to the Identity Domain administrator: DBAAS DatabaseAdministrators.
With this role you can manage all instances in your domain.

One of the main components of a DBCS Instance is a Virtual Machine (VM) called db1, created with a certain compute power that we call a shape: a combination of a number of CPUs (OCPU) and RAM.

Within this VM, you find an Oracle Database already installed and created. This could be an 11gR2 or 12cR1 database depending on how you defined the creation of your service instance from the Database Console.
By default, this database is created using an AL32UTF8 character set but you can select a different one like you can select a time zone different from the default UTC.
Also by default, every non-system tablespace is encrypted using TDE technology, and Hybrid Columnar compression is also available.
You also find within the VM all local tools you are familiar with to administer your database.
They are the same as the ones you find on your on-premises installations like SQL*Plus, RMAN, EM Express, and so on.
In addition, you find Oracle REST Data Services (ORDS), which makes it easy to develop modern REST interfaces for relational data in the Oracle Database as well as on Oracle Application Express.
Something you will not find on your on-premises installations is the Cloud tooling for easier patching, backup, recover, and monitoring of your Oracle Database.
If you want, you can back up your database to Cloud using the Database Backup Cloud Service.

In terms of pre-created users, you will find `oracle` and `opc` for which you have access.
`opc` or the Oracle Public Cloud user is a user with sudo privileges to run root commands.
The `oracle` user is the one used to install the Oracle software.
Very important are the accessible network ports by which you can access your VM through the network.
By default, only SSH port 22 is open for communication, which allows you to create tunnels for the other ports while they are blocked.
As we will see, you have the possibility to unblock them if you want.
Last but not least, you will get a file system where software and data are staged.

You have 4 GB of swap space, 16 GB for the root, where /u01 stores software, /u02 your database data files, /u03 your database backup files, and /u04 your database redo log files.
All this is configured automatically for you.
When creating a database deployment, 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.
By adding more storage, you can create a database of up to 4.6 TB with local backups or up to 12 TB without local backups.

So, to create such a VM, all you need to do is to go to your Database Console, and answer a few questions like the shape of your VM; size of your database storage; where to store backups: locally, on storage cloud, both, or no backups at all.
You need to give your service instance a name, an SSH public key used in conjunction with a corresponding private key to securely connect to your VM once created through SSH communication.
You also specify the total size for your filesystem, a unique password for all key database users and encryption password, a database name, and optionally a Pluggable database name.
You can also select your time zone, character set, use RAC, Data Guard, or Golden Gate for replication.
There's also the option to include a PDB dedicated to demonstrate new features.

Oracle Internal & Oracle Academy Use Only

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

At the time you create your database deployment, you need to specify an SSH public key that will be used in conjunction with a corresponding private key to allow VM access.
By default, this public key is called `ora_user`.
Another one (`ora_tools`) is also created to allow your My Services console to securely communicate with your VM to gather various utilization statistics and metrics for your database and compute resources.
If there is a need, you can add new public keys for the `opc` and `oracle` users to access your VM.
This can be done directly from the Database Console.

One important operation you can do directly from the Database Console is scaling your VM shape if you want to add more CPUs and memory to your database service.
This operation will temporarily stop your service before the VM is re-created without loss of any data.

The second scaling operation you can do from the Database Console is scaling your storage if you need more space for your data, backups, or anything else.
All this done by a simple click!

Another important operation that is automated from the Database Console is Database patching.

You also have the possibility to access directly from your Database console the EM Express console as well as the Apex console and the Database Monitor.

As for Cloud Tooling, you can see a list of tools you can use to help you manage your Database, like `dbaascli` for patching, data guard, and recovering, `mrec` for last option recovery of your database, `cleandblogs` for purging log files, `bkup_api` for backups, and `crontab` to automate backup creation and log purging.
For backup and recovery, these tools are based on Recovery Manager with a simple syntax for basic operations.
You also find a new web interface for monitoring your Database and VM called DBaaS Monitor.

Regarding backup and recovery operations, you can create a backup or recover your database directly from the database console.
If you choose to back up to your local VM, backups are automatically generated for your database files, database configuration files, as well as important OS configuration files.
They are stored in /u03 using an incremental policy you can change.
The retention period is 7 days by default on your local storage.

In addition to block storage provided by the Compute Block Storage layer, Oracle also provides a longer term type of storage capability that can be managed independently of your database and is called Storage Cloud Service.

You interact with your Storage Cloud Service mainly from your browser using REST APIs by managing what are called containers.

If you configured your database backups to also use Storage Cloud Service, then you must create a new container before or as you are creating your DBaaS instance. The Creation wizard includes an option to create the storage container as you are creating the DBaaS Instance.
This container will then be used automatically by your backup jobs to create object storage inside your container to store your database backups up to 30 days.
Note that all generated backups are encrypted on both local and cloud storage.
The special SBT library `libopc.so` is automatically used and configured for RMAN to access Cloud Storage through

Oracle Internal & Oracle Academy Use Only

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

Cloud Tooling.

You can also use your REST client tool to access ORDS to manage your database deployment using REST APIs.

If you look at it in detail, your DBaaS instance comprises a VM allocated from the Compute Infrastructure and an underlying block storage from where its filesystem is created.
This additional resource is part of Compute Block Storage.

The goal of the Compute Block Storage layer is to allocate volumes you can attach to your VM.
A maximum of 10 volumes can be attached to a single VM, and by default, when you create your DBaaS instance, five volumes are created: **boot**, which is 20 GB; **bits** 30 GB; **data** depending on the size you specified for your data; **fra**, which is by default 1.7 times the data size if you are using backups; and **redo**, which is 10 GB.
These volumes are automatically attached to your VM, and then formatted and mounted appropriately.
When scaling storage from your database console, you can add new volumes that will be mounted in new directories, or you can extend your data and fra storage by adding new volumes to /u02 or /u03.

You can manage some Compute resources like volumes and network controls directly from another console called the Compute Console.
You can access this console as long as you have been granted appropriate privileges like the Compute Operations role.

From the Compute Console, you can add volumes that you will mount to your filesystem after creation.
This is considered a temporary need as the volume is not automatically attached to your file system.

For network controls, you can manage Security Rules that control which ports have access to your VM by other computers on the Internet.
As shown in the diagram, a number of Security Rules are predefined to control access to specific ports used by various Oracle software like EM Express, database listener, Database Control, or important protocols like HTTP, HTTPS, and SSH.
By default, all those predefined Security Rules are disabled except the SSH port (22) allowing any computer to communicate through SSH.
You can define Security IP Lists to group a set of computers outside of Oracle Public Cloud to be used in your Security Rules.

In fact, when you create a Security List, you can allow communication between the VMs in your identity domain part of the list.

With these three entities, you can construct efficient firewalls between your service VMs and the outside world.
When you create a security rule, you can specify a security list or security IP list as a source and a security list as the destination in that security rule.
A security list can be enabled or disabled. When enabled, all list members are allowed to communicate together. When disabled, none of the members can communicate with one another.

From the Database Console, you can manage Security Rules (also called Access Rules).
From your Compute Console, you can manage all network controls.

From time to time, you have the possibility to update your Cloud Tooling to take advantage of the latest functionality.
This operation is as simple as updating the `dbaastools.rpm` file you can download to your VM from the special storage site using the `wget` command.

Oracle Internal & Oracle Academy Use Only

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

You have the possibility to install the `oracle-dbcs-cli` utility directly on your Linux on-premises servers.
You can download this utility from OTN, and install it by unzipping it.
Using this utility you can connect to Oracle Cloud and perform a variety of lifecycle and administration operations on your DBaaS instances.
In addition, you have the option to back up your on-premises databases to Oracle Database Backup Cloud Service.
To do that, you first need to install the Oracle Database Cloud Backup Module.
You'll have to download the `opc_installer.zip` module from Oracle Technology Network (OTN) and install it on your database server.

Oracle Internal & Oracle Academy Use Only

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

Lesson 2 Creating a Database Deployment

After completing this lesson, you should be able to:

- Explain the purpose of the Oracle Database Cloud Service (DBCS or Database deployment)
- Differentiate a database deployment instance from a database instance
- Create a database deployment instance

Topics:

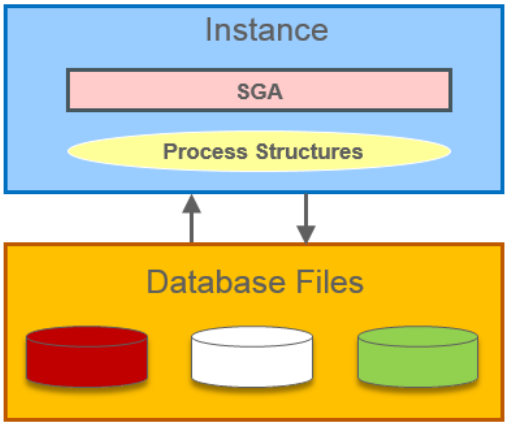
- Database Deployment vs. On-Premises Database
- Features and Tooling
- Database Deployment vs. On-Premises Database Workflow
- Additional Database Configuration Options
- Creating a Database Deployment
- Creating Linked Clones
- Storage Used for Database Files

Oracle Internal & Oracle Academy Use Only

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

Database Deployment vs. On-Premises Database

- A **Database deployment** is a compute environment that provides:
  - A Linux Virtual Machine
  - Oracle Software
  - A pre-created database
  - Additional Cloud Tools
- A **Database instance** is a set of memory structures that manage database files.



A database deployment (formerly called DBaaS instance) is different from an Oracle Database instance.

When you create a database deployment, you get the following resources allocated, according to your service level:

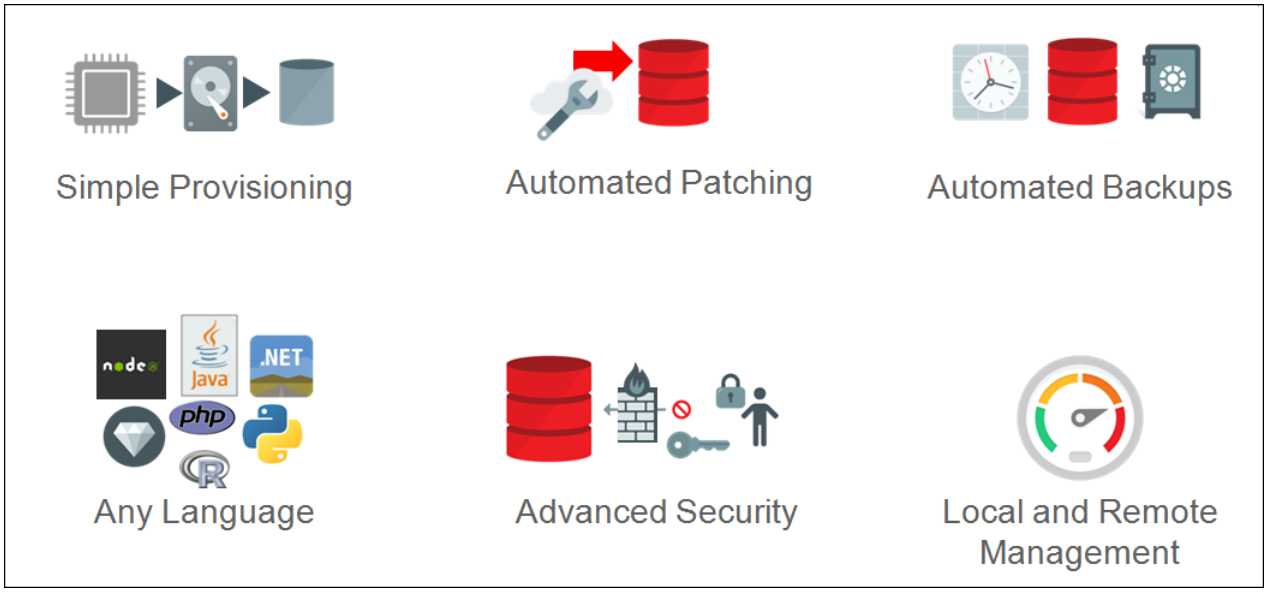
- A Linux Virtual Machine, a compute environment with pre-installed virtual machine images
- All software needed to create and run an Oracle Database
- A pre-created database
- Cloud tooling that provides:
  - Automated and on-demand backups and recovery operations
  - Automated patching and upgrading
  - Access to DBaaS Monitor. This tool is an Oracle Cloud Database Monitor web application to monitor the Oracle Database of the service instance.

To get the complete information about the service levels, refer to *Using Oracle Database Cloud - Database as a Service*

Oracle Internal & Oracle Academy Use Only

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

Features and Tooling



A database deployment includes the following features and tools:

- **Simple Provisioning**: The Create Database Cloud Service Instance wizard is used to create a new database deployment.
- **Automated Patching**: Available patches are displayed through the Database Cloud Service console and can be applied with a single click.
- **Automated Backups**: Automatic backups can be set up. Cloud tooling provides for simplified backup and recovery operations.
- **Any Languages**: Database as a Service supports any language.
- **Advanced Security**: Comprehensive security, including encryption of data at rest and data in transit.
- **Local and Remote Management**: Enterprise Manager Database Control for Oracle Database 11*g* databases and Enterprise Manager Database Express for Oracle Database 12*c* database are part of the installation and can be used to monitor and manage the database. On-premises Enterprise Manager Cloud Control 13*c* can also be used to manage the database with the Hybrid Cloud Agent.

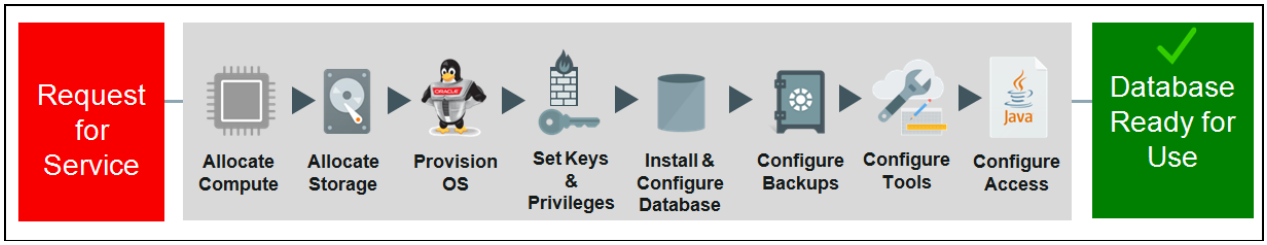
Oracle Internal & Oracle Academy Use Only

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

Database Deployment vs. On-Premises Database Workflow

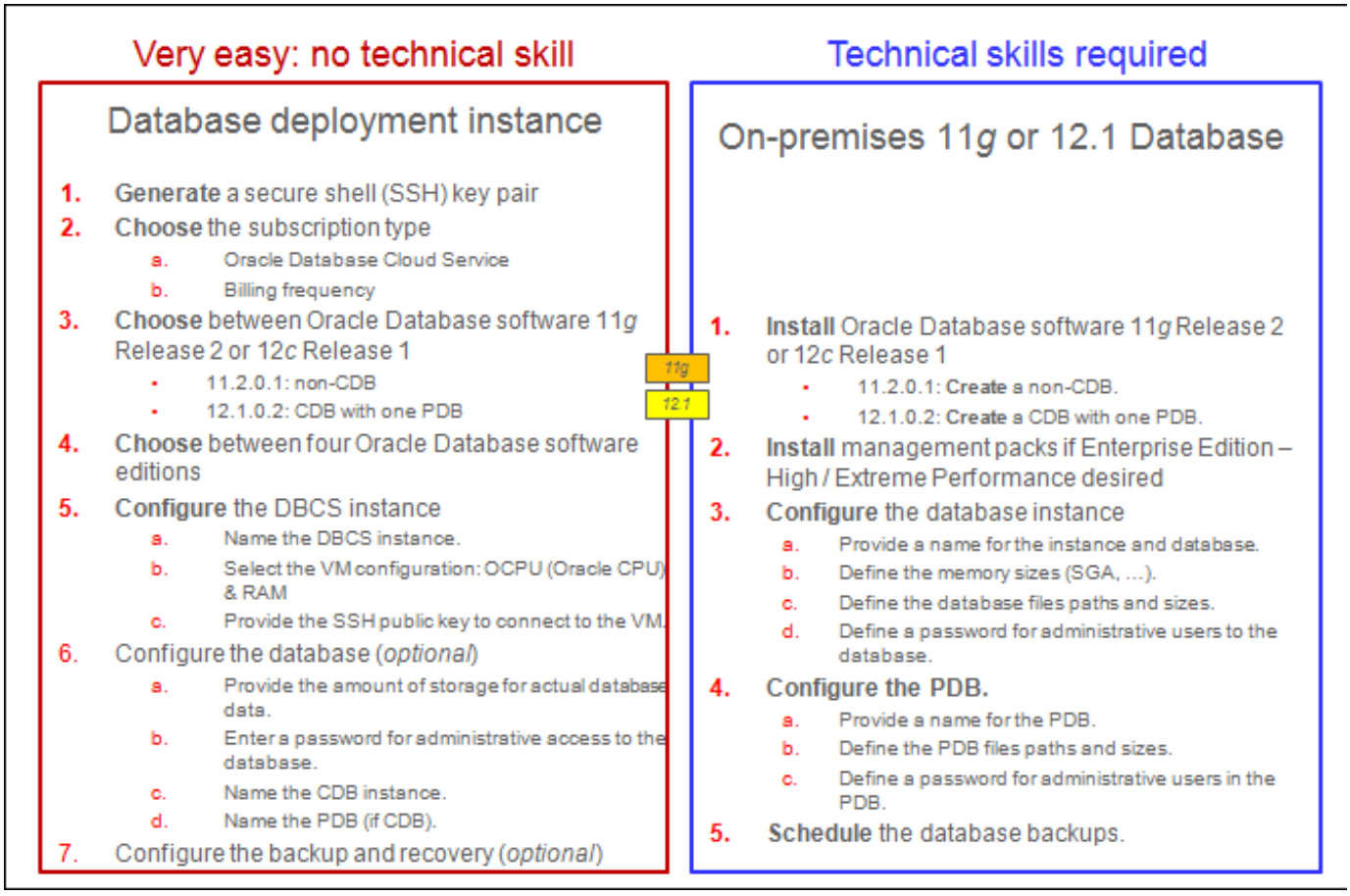
No manual server and database configuration

- The Create Oracle Database Cloud Service Instance wizard is used to create a new database deployment.
- After requesting the creation of the database deployment, the steps shown in the gray box happen automatically.



- Once the creation and configuration is complete, the database deployment is ready for use.

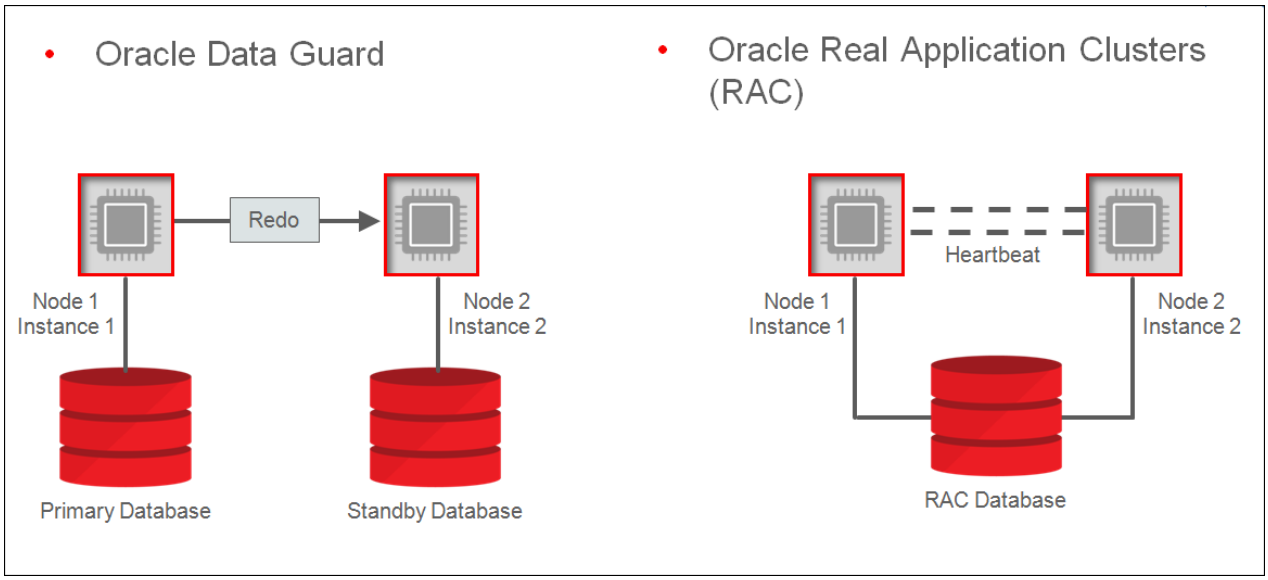
No technical expertise



Oracle Internal & Oracle Academy Use Only

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

Additional Database Configuration Options



Oracle Data Guard

You can optionally create an Oracle Data Guard configuration when you create a database deployment.

Oracle Data Guard enables production Oracle databases to survive disasters and data corruptions by providing a comprehensive set of services that create, maintain, manage, and monitor a standby database. Oracle Data Guard maintains the standby database as a copy of the production database. If the production database becomes unavailable because of a planned or unplanned outage, you can switch the standby database to the production role, minimizing the downtime associated with the outage.

Read Using Oracle Data Guard in Database as a Service.

Oracle Real Application Clusters (RAC)

You can optionally create an Oracle Real Application Clusters (RAC) database when you create a database deployment.

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.

Read Using Oracle Real Application Clusters (RAC) in Database as a Service.

Oracle GoldenGate

Oracle GoldenGate Cloud Service is a high performance, secure public cloud data integration and replication solution that delivers data in real time from on-premises databases to single-instance databases in Oracle Database Cloud - Database as a Service.

Read Using Oracle GoldenGate Cloud Service with Database as a Service.

Oracle Internal & Oracle Academy Use Only

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

Creating a Database Deployment

Proceed with the following sequence to create a database deployment.

1. From the Oracle Database Cloud Service console, click Create Service.

   You can also use the `oracle-dbcs-cli` utility on your Linux computer to connect to Oracle Cloud and create the database deployment. Read *The Data File for the `oracle-dbcs-cli` create Subcommand* .

2. Choose the subscription type.
   a. Oracle Database Cloud Service
   b. Billing frequency
   Read *Service Level* .

3. Choose between Oracle Database software, and then between Oracle Database software editions which may or may not include some options.
   Depending on the version, the pre-created database is a non-CDB or a PDB within a CDB. Read *Oracle Database Release Version.*

Edition	Included Options	Included Packs
Standard	None	None
Enterprise	None	None
Enterprise - High Performance	Advanced Analytics, Advanced Compression, Advanced Security, Database Vault, Label Security, Multitenant, OLAP, Partitioning, Real Application Testing, Spatial and Graph	Cloud Management for Oracle Database, Database Lifecycle Management, Data Masking and Subsetting, Diagnostics, Tuning
Enterprise - Extreme Performance	Active Data Guard, Advanced Analytics, Advanced Compression, Advanced Security, Database In-Memory, Database Vault, Label Security, Multitenant, OLAP, Partitioning, Real Application Clusters, Real Application Testing, Spatial and Graph	Cloud Management for Oracle Database, Database Lifecycle Management, Data Masking and Subsetting, Diagnostics, Tuning

4. Configure the database deployment instance.
   a. Name your database deployment instance.
   b. Select the compute node configuration: OCPU (Oracle CPU) & RAM
   c. Provide the SSH public key to connect to the compute node. Click Edit beside the *SSH Public Key* field.

   Have the wizard create a public/private key pair for you. In this case of system generated keys, click "Create a New Key" button => this creates a system generated key pair to connect to your compute node, then use the Download button to download the key pair on your local machine. The public key is automatically uploaded to the compute node allocated to the database deployment. If you plan to connect to your compute node with PuTTY, use PuTTY Key Generator to load the system generated private key file and save the file in PuTTY's own format.
   Or
   Upload a file containing the public key value, paste in the value of a public key: In this case of manual keys creation, proceed with the key pair creation before you start the creation of the service.

   The **private key file** is kept on your local compute node.

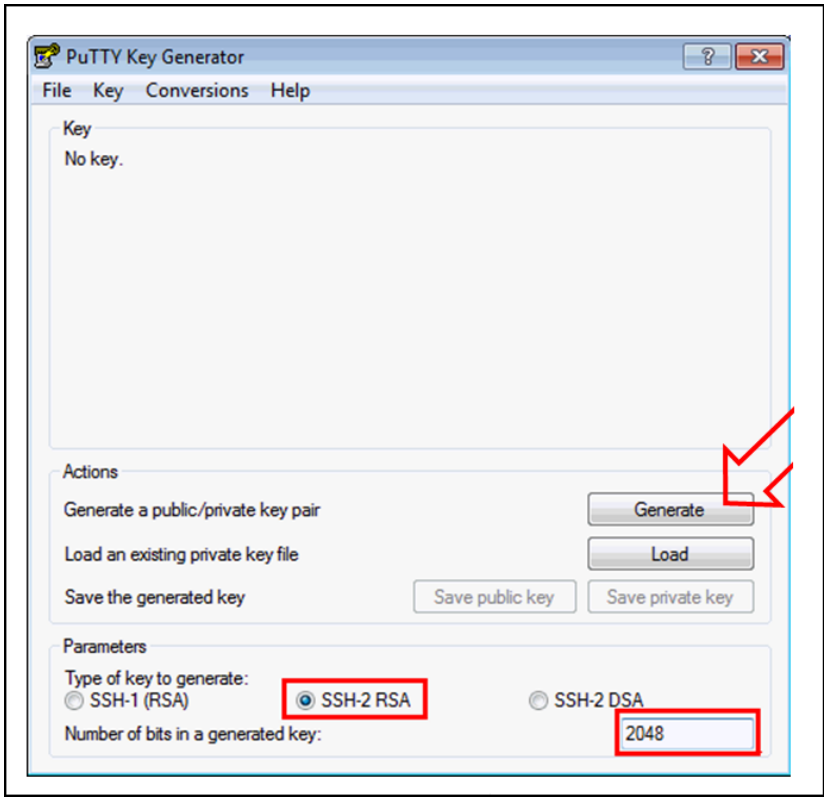
Oracle Internal & Oracle Academy Use Only

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

The **public key file** will be stored in the database deployment. Read *Generating a Secure Shell (SSH) Public/Private Key Pair*.

Find below the steps to proceed with the key pair creation.

- On Windows, use PuTTY Key Generator.



- On Linux, use the `ssh-keygen` command.

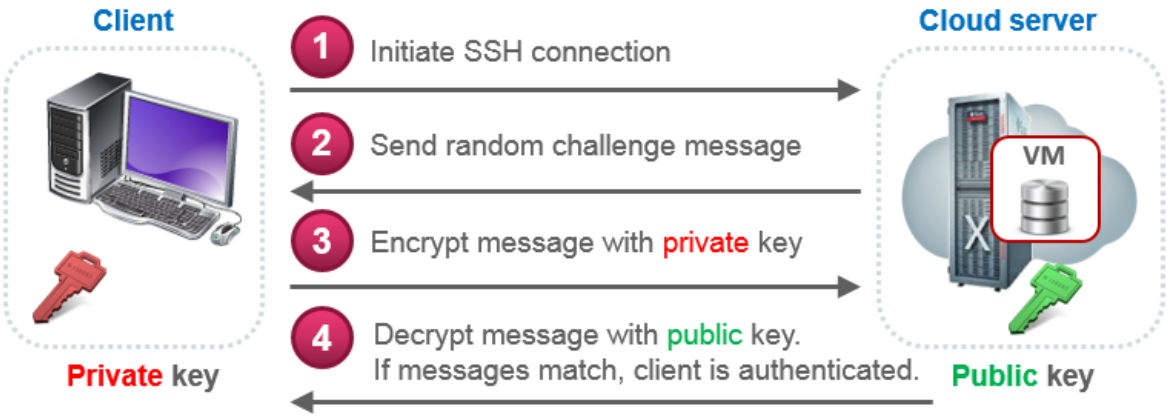
```
$ ssh-keygen -b 2048 -t rsa -f MyTestKey
Generating public/private rsa key pair.
```

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

Oracle Internal & Oracle Academy Use Only

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in MyTestKey.
Your public key has been saved in MyTestKey.pub.
```

What is the purpose of the SSH key pair in SSH authentication?



Read *Computing Power.*

5. Configure the database (optional).

    a. Provide the amount of storage for actual database data.
    Read *Database Storage.*
    b. Enter a password for administrative access to the database.
    c. Name the database instance.
    d. Name the PDB if a PDB is used.

6. Configure the backup and recovery (optional).
    Read *Automatic Backup Configuration.*

There are other operations that need to be completed to benefit from the database deployment possibilities. Read the *Typical Workflow for Using Database as a Service* .

The whole process is explained in *Creating a Database Deployment.*

Note: If you need to deploy an application to the Java Cloud Service instance, read the OBE Getting Started with Oracle Java Cloud Service.

Oracle Internal & Oracle Academy Use Only

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

Creating Linked Clones

DBCS supports the creation of storage snapshots, which you can then use to create new database deployments called “linked clones” useful for application testing or branched application development work.
The storage volumes of the new database deployment are cloned from the snapshot. Using the “copy on write” technology that Oracle Compute Cloud Service supports for storage volume snapshots, the file data on the linked-clone deployment can change without changing the snapshot itself.
If data has been changed on a linked-clone deployment and if a new linked clone deployment is created from the snapshot, the file data in the new linked clone deployment will reflect the data from the snapshot and not from the other linked clone deployment.

Two steps are required to use linked clones:

1. *Creating a Snapshot*
2. *Creating a Database Deployment from a Snapshot*

Once the snapshot is created, you can then perform step 2 only to create as many database deployments as needed.

Oracle Internal & Oracle Academy Use Only

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

Storage Used for Database Files

When a database deployment is created, Oracle Cloud Service storage volumes are created and allocated to the database deployment's compute node.

Read *Storage Volumes and File System Layout* below to get the details.

Go to main content

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 - Database as a Service.
Compute Cloud Service Storage Volumes

When a Database as a 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	21 GB volume allocated to the following file system mounts on the virtual machine: • `/` (root) • `/boot` • 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 11 GB. This volume is completely allocated to `/u02` on the virtual machine.
fra	If backups are being configured, GB size equal to 1.7 times the size of the data volume. If backups are not being configured, GB size equal to 0.1 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; 15.8 GB allocated from the boot Compute Cloud storage volume.
`/boot`	Operating system kernel; 200 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.
`/u03`	Database backup storage; the entire fra Compute Cloud storage volume.
`/u04`	Database redo logs; the entire redo Compute Cloud storage volume.

Scripting on this page enhances content navigation, but does not change the content in any way.

Oracle Internal & Oracle Academy Use Only

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

Lesson 3 Administering a Database Deployment

After completing this lesson, you should be able to:

- Navigate with the consoles
- Administer database deployment VM users/permissions
- Administer database deployment database users/privileges
- Administer storage/disks requirements
- Patch your database deployment
- Configure network groups for using tools like Oracle Enterprise Manager Cloud Control

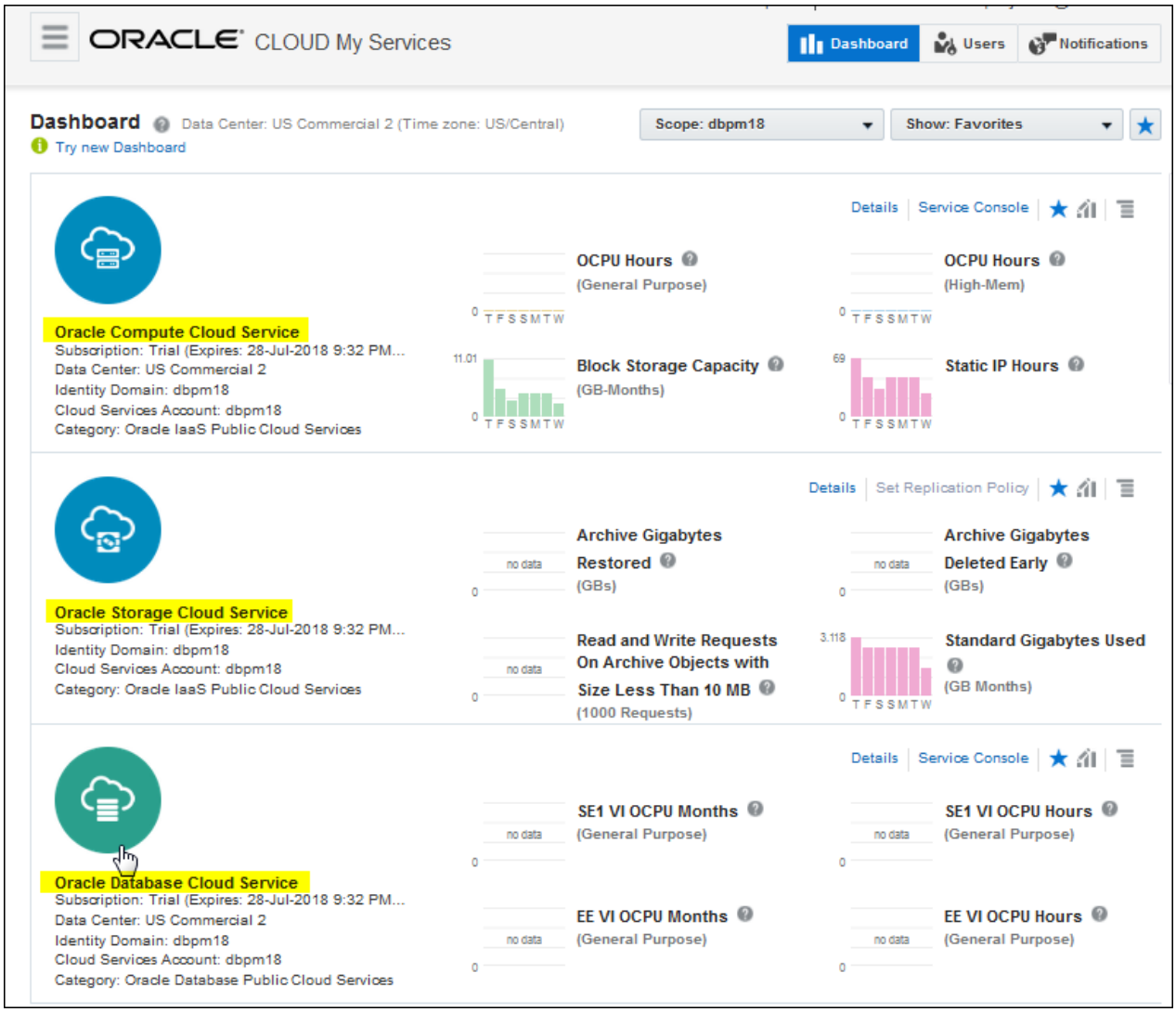
Topics:

- Navigating with Dashboard Console
- Navigating with Database Cloud Service Console
- Navigating with Compute Cloud Service Console
- Users/Roles/Privileges
- Administering Users/Roles/Privileges
- Configuring Connections
- Managing Compute Node Users
- Adding Compute Node Users
- Managing Database Users and Privileges
- Administering Storage / Disks Requirements
- Stopping and Restarting a Database Deployment
- Cleaning Up Log and Diagnostics Files
- Patching a Database Deployment
- Upgrading a Database Deployment
- Deleting a Database Deployment
- Reaching Known Issues Solutions

Oracle Internal & Oracle Academy Use Only

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

Navigating with Dashboard Console



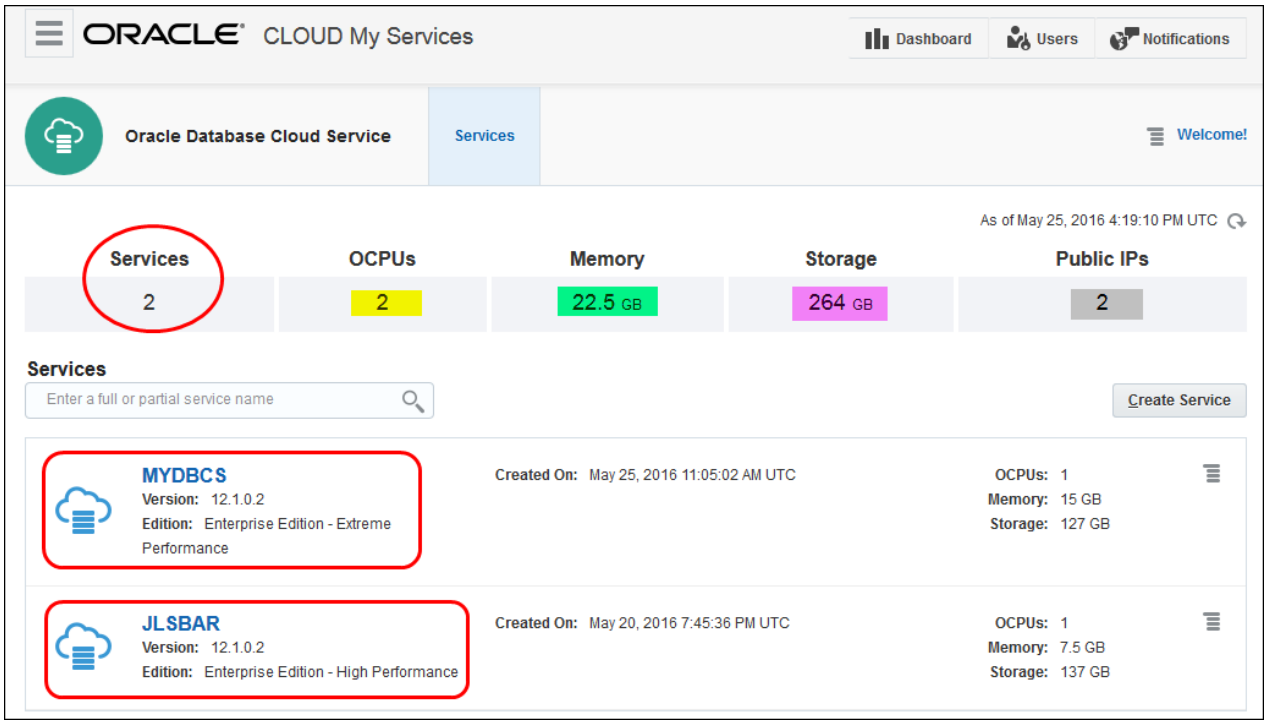
The My Services - Dashboard console displays a summary of all subscription services. Read *Exploring the My Services Dashboard* to understand all the details provided by the console.

As you can see, Oracle Cloud is evolving so fast that there are different ways to display the services consumption. There is the new Dashboard and the original dashboard. You can switch from one to the other as mentioned on each of them.

Oracle Internal & Oracle Academy Use Only

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

Navigating with Database Cloud Service Console

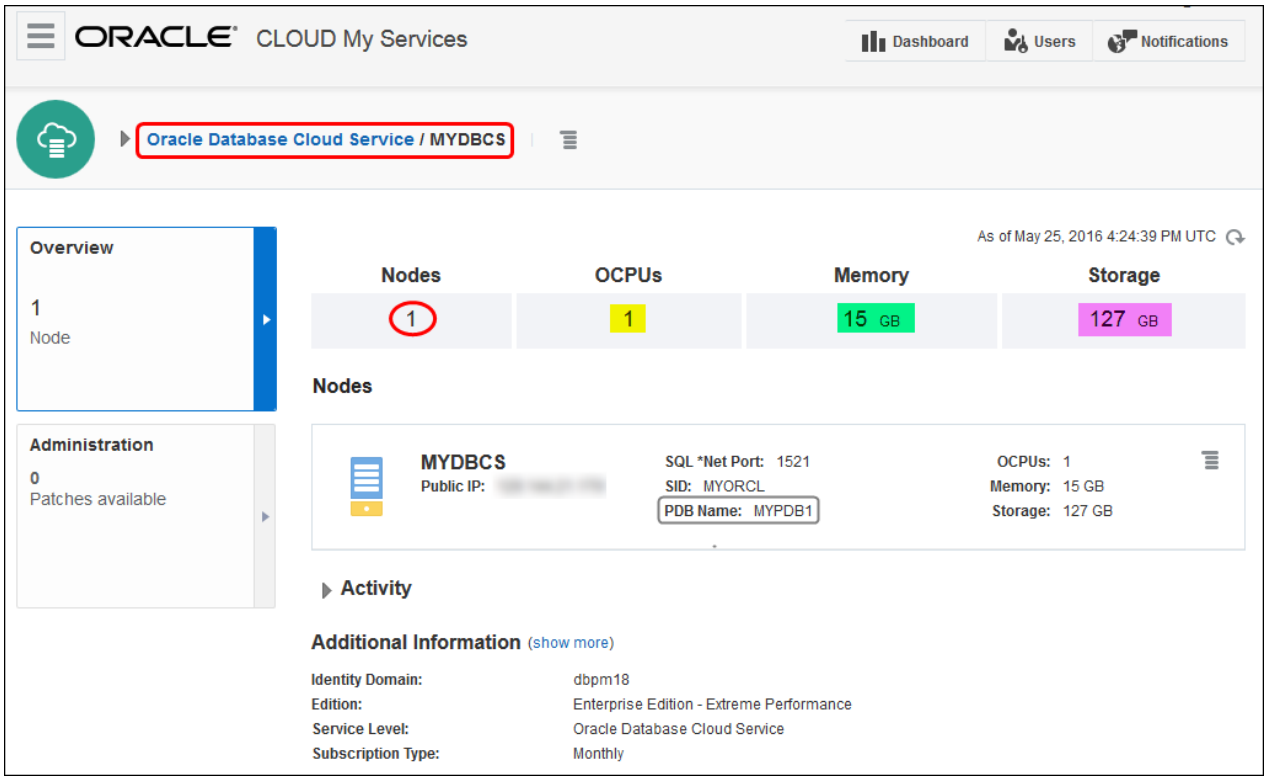


- What type of details do you retrieve about all database deployments?
  - What can you do from the DBCS Services page: View, create and delete database deployments.
  - What can you see on the DBCS Services page:
    a. Total number of configured deployments (2)
    b. Total number of Oracle CPUs allocated across all database deployments (2)
    c. Total amount of compute node memory allocated across all database deployments (22.5 GB)
    d. Total amount of storage allocated across all database deployments (264 GB)

  For all details provided, read *Oracle Database Cloud Service Services Page*
- How do you retrieve the details about your database deployment?

Oracle Internal & Oracle Academy Use Only

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



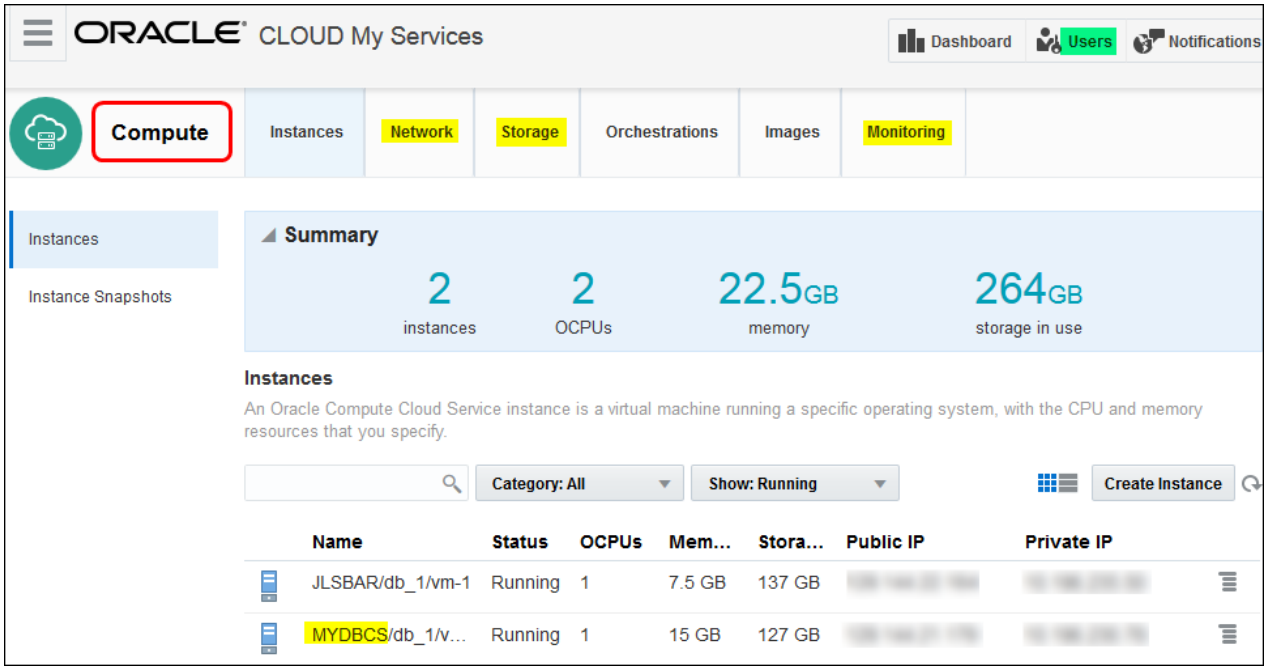
From the Oracle Database Cloud Service console, click your database deployment.
For details on accessing the console, see the documentation on *Accessing the Database Cloud Service Console*.

Oracle Internal & Oracle Academy Use Only

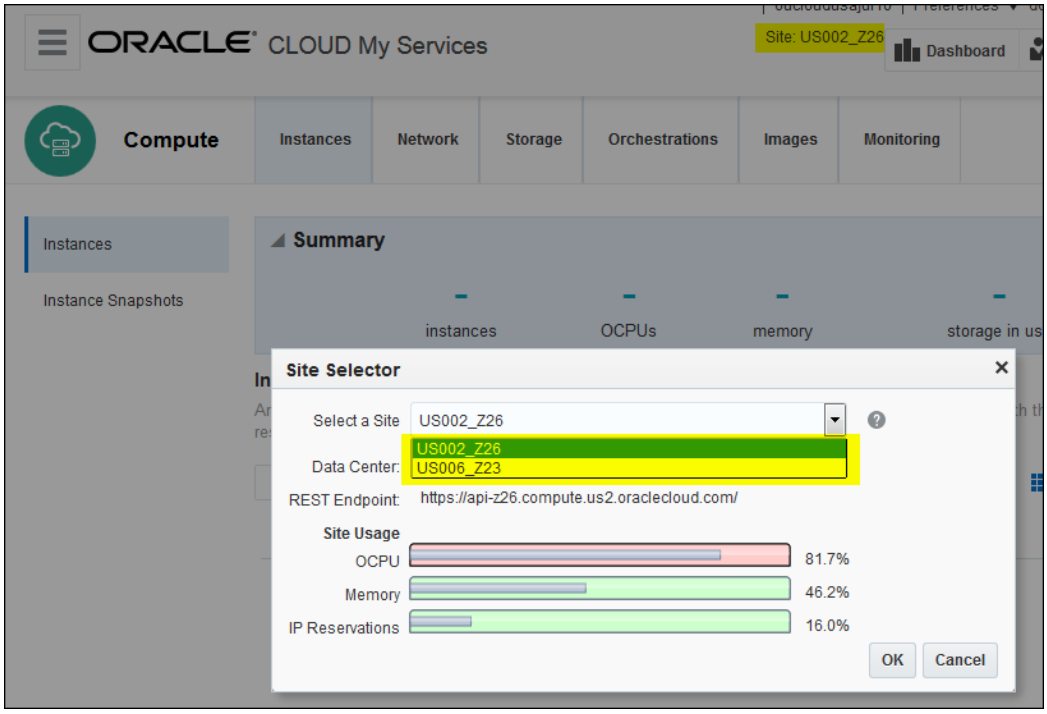
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Navigating with Compute Cloud Service Console

- What can you do with the Compute Cloud Service console? Read *Getting Started with Oracle Compute Cloud Service* and the next lessons to get details about the tabs displayed below.
  - Rapidly provision virtual machines on Oracle Cloud with all the necessary storage and networking resources
  - Manage and scale your virtual machine topology in the cloud easily
- How do you access the Compute console? Read *Accessing Oracle Compute Cloud Service Using the Web Console*.



Note: You may not see the information related to all instances, network, storage and other metadata. This depends on the site selector visible in the upper right of the compute console.



Oracle Internal & Oracle Academy Use Only

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

Your account may span two data centers. This allows you to deploy a database with the Data Guard option.

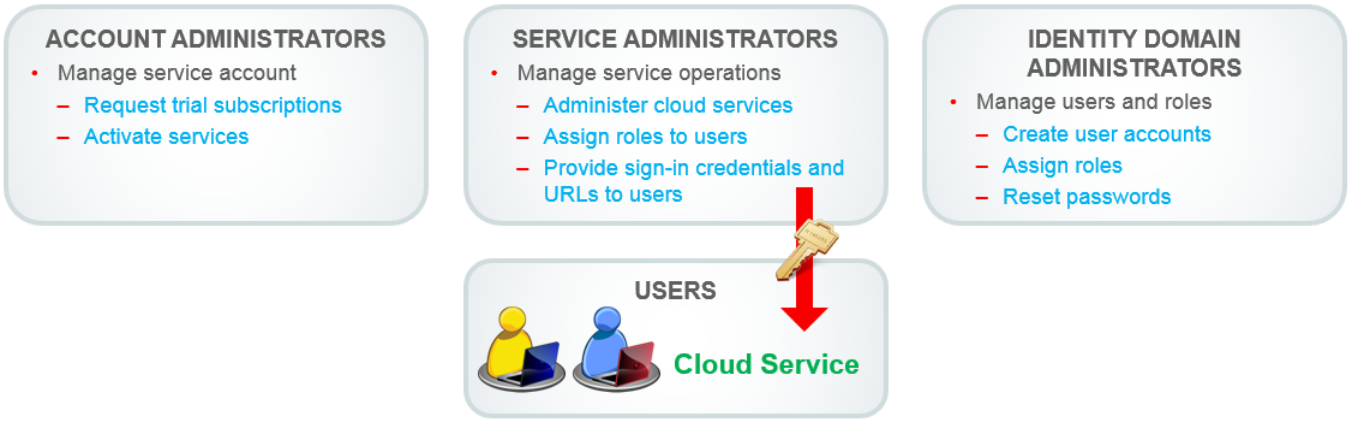
Oracle Internal & Oracle Academy Use Only

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

Users/Roles/Privileges

If you have subscribed to an entitlement to create instances of an Oracle Cloud Service, then you can create multiple service instances based on your business needs.

You have identity domain administrator privileges to create a user and assign the service entitlement administrator role to the user, who will create service instances. The actual name of this role depends on the Oracle Cloud service that you have subscribed to.



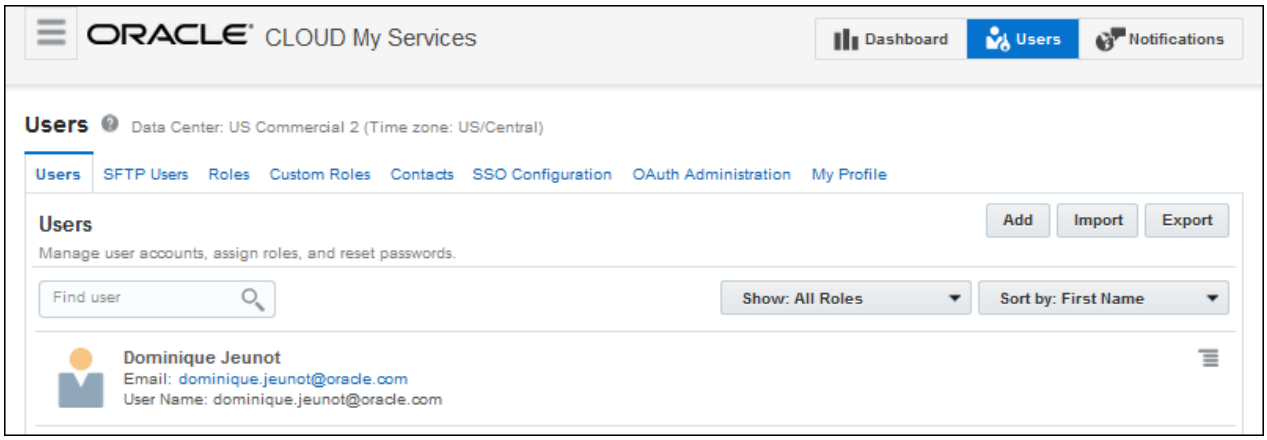
- **Identity domain administrators** can assign and remove roles only for the users in the identity domains that they manage.
- **Service administrators** (or DBaaS Database Administrator) can assign and remove roles only for the users of the services that they manage. Because service administrators cannot add users or roles, the users and roles must already be in the system before service administrators can assign a specific role to a user.
    - Create and delete database deployments
    - Scale, patch, and back up or restore database deployments
    - Monitor and manage service usage in Oracle Cloud
    - Grant the database deployment administrator role to existing users
- Which can be the roles and responsibilities of an Oracle Cloud service user? *Read Oracle Cloud User Roles and Privileges*.

Oracle Internal & Oracle Academy Use Only

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

Administering Users/Roles/Privileges

- Users in the Cloud are not database users.

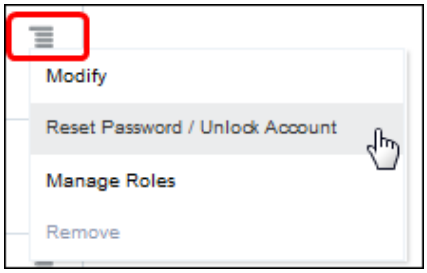


- The Users tab may be selected from various consoles. On the Users tabbed page, you can add individual users and configure the roles (privileges) each should have.
- User privileges are controlled through assigned roles. The roles assign privileges in a granular fashion and each role has specific privileges.
- The Import tab allows you import a list of users as a CSV file (comma delimited).

On the Users page are:

- SFTP Users for file transfers – These are predefined users with specific uses.
- Roles – Predefined roles for managing the Cloud services; roles are assigned to individual users or a single role can be assigned to a list of users in a CSV file.
- Custom Roles – Created and deleted by identity domain administrators and then used by application developers to secure applications.
- SSO Configuration – Configure Single Sign-On so your users can use their company credentials to log in to all applications, including Oracle Cloud applications.
- OAuth Administration – You can manage client access to Oracle Cloud APIs using the OAuth 2.0 protocol.
- My Profile – Each user can modify the basic user information, reset the password, change the roles, and remove the user.

For each user, there's a menu icon that can be used to modify basic user information, reset the password, change the roles, and remove the user.



- To add a new user, read *Creating a User and Assigning a Role.*
- How to assign one role to many users? Read *Assigning One Role to Many Users*.

Oracle Internal & Oracle Academy Use Only

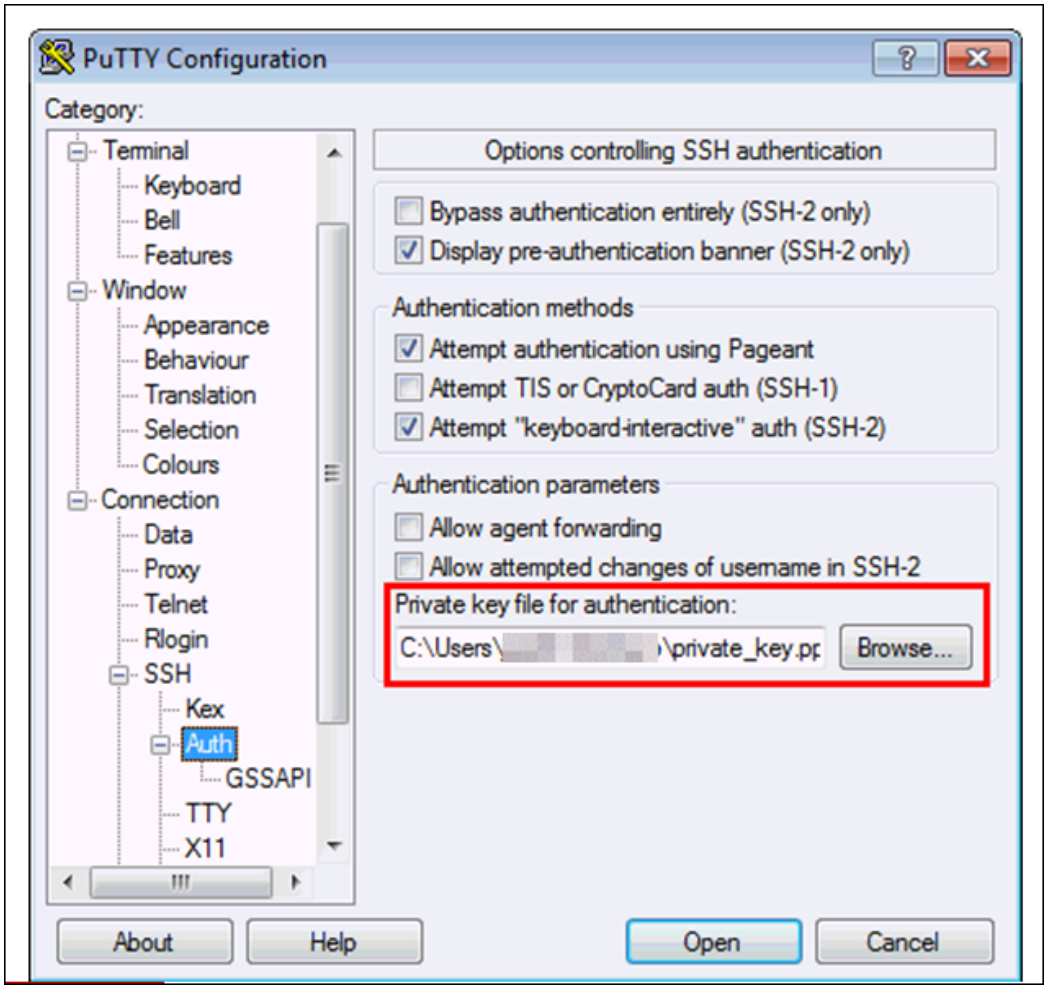
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Configuring Connections

- When a database deployment is created, network access to the database deployment's compute node is limited to Secure Shell (SSH) connections on port 22. This restriction of access ensures that the database deployment is secure by default. Access requires a public key that was supplied when the database deployment was created and stored in the database deployment.
- To access other ports, either enable network access to the port or create an SSH tunnel to the port (This topic is covered in another lesson).

To log in to the compute node of the database deployment, you need to:

1. First retrieve the public IP address of the compute node of the database deployment. Read *Viewing Detailed Information for a Database Deployment*.
2. Configure connections to the compute node through Secure Shell (SSH) using the private key file.
   - On Windows, use PuTTY Configuration to configure connections to the database deployment compute node. Provide the private key file generated for the database deployment creation.



   - On UNIX/Linux: Use `ssh` to configure connections to the database deployment VM. Provide the private key generated for the database deployment creation. The private key file must have very restrictive permissions.

Oracle Internal & Oracle Academy Use Only

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

```
$ cd /home/user-name/.ssh
$ chmod 600 private-key-file
$ ssh -i private-key-file user-name@vm-ip-address
```

Note: If the private key was generated with PuTTYgen originally, you must use an open-ssh format file with ssh. You can use the conversion in PuTTYgen to create an open-ssh format private key file.

Oracle Internal & Oracle Academy Use Only

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

Managing Compute Node Users

When a database deployment is created, three Linux users are created.

OS User	Authorization
Pre-created user `oracle`	*- Authorized to log in VM* *- Not authorized to run root commands*
Pre-created user `opc`	*Authorized to run root commands* `sudo -s`
Pre-created user `root`	*Not authorized to log in VM*

- `opc`: Certain operations like database backup and recovery or compute node reboot require the `sudo` command, and therefore to be connected as `opc`.
- `oracle`: The `oracle` user is the administrator account you use to access the system and perform operations on the compute node. A home directory, `/home/oracle`, is created for this user. This user cannot use the `sudo` command to perform operations that require root-user access.
- `root`: The root administrator for the system exists but you do not have direct connection to this account. To perform operations that require root-user access, use the `sudo` command as the `opc` user.
-

Variable	Description
HOME	The home directory of the user, either `/home/opc` or `/home/oracle`
ORACLE_HOSTNAME	The host name of the virtual machine
ORACLE_HOME	`/u01/app/oracle/product/12.1.0/dbhome_1`
ORACLE_SID	The SID provided when the instance was created
PATH	`/sbin;/usr/sbin;/bin;/usr/bin;$ORACLE_HOME/bin;$ORACLE_HOME/OPatch;$HOME`

It is recommended to configure two sessions, one as the `oracle` and the other as `opc`. Both have environment variables set in sessions.

Database deployment compute node Linux users access the compute node through SSH access using a key pair.

You can create other Linux users on the compute node who can be authorized to run root commands with `sudo -s`.

Oracle Internal & Oracle Academy Use Only

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

Adding Compute Node Users

You can create other Linux users to log in to the compute node of your database deployment.

1. Log in to your database deployment compute node as the `opc` user and create the new user.

   ```
   $ sudo -s
   # useradd user1
   # mkdir /home/user1/.ssh
   ```

2. Copy the SSH public key value to the authorization file.

   ```
   # echo "ssh-rsa AAAAB3..." > /home/user1/.ssh/authorized_keys
   ```

3. Add the new user to the list of users allowed on your instance by editing the `AllowUsers` option in the `/etc/ssh/sshd_config`.

   ```
   AllowUsers oracle opc user1
   ```

4. Set the ownership of the new user's home directory files.

5. Restart the SSH daemon on your instance.

   ```
   # /sbin/service sshd restart
   ```

6. From a machine that has the private key, the new user can now SSH to the compute node. The new user must specify the private key generated in step 1.

   ```
   $ ssh -i private_key user1@VM_IP_address
   ```

The new user can perform standard OS operations, such as installing and running applications.

Note: These changes made to your database deployment do not persist. If for any reason, the compute node is automatically reprovisioned by Oracle, any changes that you had made on that compute node will be lost. If this happens, you will need to repeat these steps to add new local users afresh.

Oracle Internal & Oracle Academy Use Only

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

Managing Database Users and Privileges

When a database deployment is created, if you choose to install Oracle Database 11*g*R2, a non-CDB is created.

If you choose to install Oracle Database 12*c*R1, a CDB with one PDB is created.

You manage your users in database deployment databases as you manage your users in on-premises databases.

- You grant privileges to users so they can accomplish tasks required for their jobs.
- You should grant a privilege or role to a user who requires that privilege or role to accomplish the necessary work.
- Excessive granting of unnecessary privileges can compromise security.
  For example, you never should grant `SYSDBA`, `SYSOPER`, `SYSBACKUP` or `SYSKM` administrative privilege to users who do not perform system administrative tasks.
- Connect to the database deployment non-CDB or CDB database instance with system administrative privileges.
- Connect to the database deployment non-CDB or CDB or PDB as a DBA.
- Connect to the database deployment PDB as a common user.
- Connect to the database deployment PDB as a local user.

Oracle Internal & Oracle Academy Use Only

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

Administering Storage / Disks Requirements

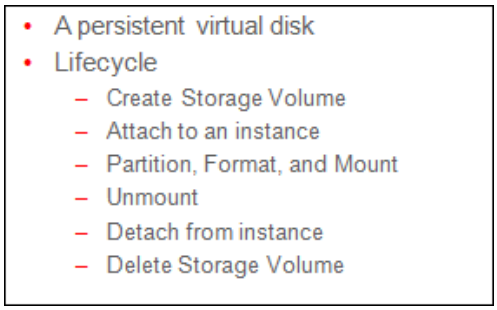
The Compute Cloud service provides the resource profiles called "shapes," which are chosen in the database deployment creation dialog.

The storage volume is used to store:

- Operating system files
- Oracle Database binaries
- The data files, control files, redo log files, and all files related to an Oracle database instance (password file, trace files, alert log, backup files ...)

Permanent Storage

A Storage Volume is a persistent virtual disk. The volume can be attached to different database deployments, but only to one at a time.



- You may create up to nine storage volumes to each database deployment.
- Once the volume is created and attached, the volume must be formatted and mounted as the root user before it can be used.
- The volume must be unmounted before it may be detached, and detached before deleted.
- Storage volumes incur a cost, even if they are not attached.
- Storage volumes remain attached and available to the database deployment even after it is restarted or is stopped and then started.
- Storage volumes exist until you delete the database deployment, at which time the storage volumes are also deleted.

These operations can be easily performed via the Oracle Database Cloud Service console. The storage volume request will automatically create, attach, and mount it to the database deployment.
To learn how to scale up the storage for a database deployment, read *Scaling Up the Storage for a Database Deployment*.

Temporary Storage

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.
To learn how to scale up the storage for a database deployment, read *Adding Temporary Storage to a Database Deployment*.

Oracle Internal & Oracle Academy Use Only

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

Stopping and Restarting a Database Deployment

From the Oracle Database Cloud Service console, you can:

- Stop the compute nodes associated with a database deployment
  - No access to the compute nodes and database is possible
  - No management operations except to start the database deployment or to delete the database deployment.
  - Similar to turning off your personal computer
- Restart the compute nodes associated with a database deployment
  - Similar to turning your personal computer back on

Read *Stopping, Starting and Restarting Database Deployments*.

Oracle Internal & Oracle Academy Use Only

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

Cleaning Up Log and Diagnostics Files

- Log and diagnostics files consume space on the compute node of the database deployment.
- To avoid running out of file storage space is an important administrative task.

Automatic Clean Up

- The `/var/opt/oracle/cleandb/cleandblogs.pl` script simplifies 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
  - `AlertRetention`: 14 days retention
  - `ListenerRetention`: 14 days retention
  - `AuditRetentionDB:` Database audit (`*.aud`) 14 days retention
  - `TraceRetention`: Trace file (`*.tr*` and `*.prf`) 7 days retention
    Read *Managing the Log and Diagnostic Files on Database as a Service* to get all possible configurable parameters.

Manual Clean Up

Log on the compute node as the `oracle` user and execute the `/var/opt/oracle/cleandb/cleandblogs.pl` script
.

Oracle Internal & Oracle Academy Use Only

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

Patching a Database Deployment

- Apply a patch when patches are required to your database deployment.
  - Database PSU patch
  - DBCS Tooling (rpm update)
  - OS patch (new UEK3)

    Note: Patches applied are based on a patch `map.xml`, which is stored in a known Oracle Storage service container.

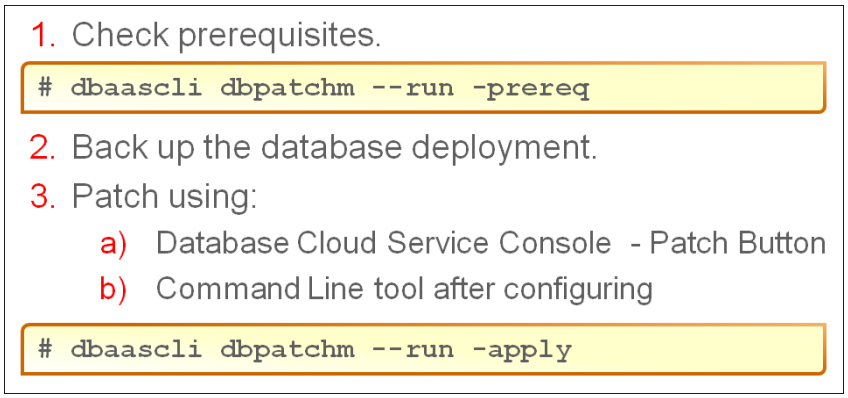
- Roll back a patch as necessary. Read *Rolling Back a Patch or Failed Patch*.

How to proceed?

Either use the UI Cloud interface from the Oracle Database Cloud Service console:

Read *Applying a Patch by Using the Oracle Database Cloud Service Console*.

Or use the command-line utility:



The first command is similar to `$ORACLE_HOME/OPatch/opatch prereq`.
The second command is similar to `$ORACLE_HOME/OPatch/opatch apply` followed by SQL data patching scripts.

1. Read *Checking Prerequisites Before Applying a Patch.*
2. Read *Creating an On-Demand Backup*.
3. Read *Applying a Patch*.

You can also use the `oracle-dbcs-cli` utility subcommands to perform various life-cycle and administration operations on your database deployments.
Read *The patch apply Subcommand* to apply patches to the compute node of the database deployment or *The patch check Subcommand* to precheck patches to apply to the compute node of the database deployment or *The patch rollback Subcommand* to roll back patches to the compute node of the database deployment.

Oracle Internal & Oracle Academy Use Only

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

Upgrading a Database Deployment

You can upgrade your current database deployment to a new database deployment version because you want to use the new features of the Oracle Database Release Version.

The main steps to proceed:

1. Create a new database deployment and specify the desired Oracle Database Release Version.
2. Choose any method to transport data: Export, backup, unplug data from the source database deployment database.
3. Transport the data (export dump file, database files, xml file) from the source database deployment compute node to the target new database deployment compute node.
4. Import, restore, or plug data (using the corresponding method selected in step 2) into the target new database deployment database. Read the Migrating to Oracle Database Cloud Service module from the same course.
5. Delete the old database deployment.

Source Database Deployment	Target Database Deployment	Methods
11g	12.1.0.2	1. Create 12.1.0.2 database deployment 2. Export data from source database + import into target database 3. Delete 11g database deployment
11g	12.2	Same process
12.1.0.2	12.2	Same process Or 1. Create 12.2 database deployment 2. Unplug PDBs from 12.1.0.2 database + Plug PDBs into 12.2 database 3. Delete 12.1.0.2 database deployment

Oracle Internal & Oracle Academy Use Only

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

Deleting a Database Deployment

What happens when you delete a database deployment?

- The database deployment terminates.
- The Oracle database is dropped.
- The storage is removed.
- The compute node is deallocated.

How to delete a database deployment?

- Use the UI Cloud interface, read *Deleting a Database Deployment*.
- You can also use the `oracle-dbcs-cli` utility on your Linux computer to connect to Oracle Cloud. Read *The delete Subcommand*. Many of the subcommands that the `oracle-dbcs-cli` utility supports require a configuration file that specifies information to connect to the correct Oracle Database Cloud identity domain. Read *The Configuration File for oracle-dbcs-cli Subcommands*.

Oracle Internal & Oracle Academy Use Only

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

Reaching Known Issues Solutions

You may encounter miscellaneous issues related to any topic:

- DBaaS Monitor
- Backup
- APEX and EM Express access
- Database startup
- Users tablespace
- Java Cloud service
- Password change ...

To get a possible solution to known issues, read *Known Issues and Deprecated Features*.

Oracle Internal & Oracle Academy Use Only

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

Lesson 4 Backing Up and Recovering

After completing this lesson, you should be able to:

- Explain the default backup configuration
- Customize backup configuration
- Perform an on-demand backup
- Perform a database recovery

Topics:

- Backup Destinations
- Utilities to Back Up and Recover
- Backup Configuration
- On-Demand Backup and Recovery

Oracle Internal & Oracle Academy Use Only

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

Backup Destinations

- During the database deployment, you select the destination for the automatically created backups.
  - **None**: No backup configuration is created.
  - **Local Storage Only**: Backups are configured to be created automatically and stored only on local storage on the compute nodes associated with the deployment. (Not available with the UI Cloud interface, but only with the *`oracle-dbcs-cli`* utility)
  - **Cloud Storage Only**: Backups are configured to be created automatically and stored on an Oracle Storage Cloud Service container. The container must have been created before creating the Database as a Service database deployment. Read *Creating Containers* .
  - **Both Cloud Storage and Local Storage**: Backups are configured to be created automatically and stored both on local compute node storage and on an Oracle Storage Cloud Service container. The Storage Cloud container can either be created during the database deployment creation or be created in advance.

  Read *About Backing Up Database Deployments on Database as a Service* .

- After the database deployment creation, you can change the backup destination.

  Read *Changing the Backup Configuration to a Different Backup Destination* .

Oracle Internal & Oracle Academy Use Only

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

Utilities to Back Up and Recover

- When a database deployment is created, backup and recovery can be performed with the following utilities that are installed automatically:

Service Level	Utility
No automatic backup	RMAN
Automated backup service level	`bkup_api`
Automated recovery service level	`dbaascli`

- Database backup and recovery operations require the `sudo` command and therefore to be connected as `opc` to execute the `/var/opt/oracle/bkup_api` or `dbaascli` utility.

Oracle Internal & Oracle Academy Use Only

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

Backup Configuration

The default backup configuration

Files Backed Up	Backup Type
Database	Full (level 0) backup of the database followed by rolling incremental (level 1) backups on a 7–day cycle
Database configuration files	Full backup of selected database configuration files
Important system files	Full backup of selected system files

To perform backup configuration changes, log in to the database deployment compute node.

Configuration Change	Utility
Database files	RMAN
Database configuration files	Edit `/home/oracle/bkup/<DBNAME>/dbcfg.spec`
System files configuration	Edit `/home/oracle/bkup/<DBNAME>/oscfg.spec`
Retention and cycle period	`bkup_api` (as `opc`)
Frequency of automatic backups	`crontab` (as `opc`)

Read *Customizing the Current Backup Configuration on Database Deployments Hosting Single-Instance Databases* to know how to:

- Customize **how** the database is backed up
- Customize **which system files** are backed up
- Customize **which database configuration files** are backed up
- Customize the **retention period** for backups
- Customize the **cycle period** for backups
- Customize the **frequency** of automatic backups

Oracle Internal & Oracle Academy Use Only

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

On-Demand Backup and Recovery

To perform an on-demand backup, you can use:

- The `bkup_api` utility (as described below)
- The Database Cloud Service console: Read *Creating an On-Demand Backup by Using the Oracle Database Cloud Service Console*

1. Connect to the database deployment VM.
2. Disable the scheduled backup configuration (`crontab`).
3. Create the backup.

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

4. Check the progess of the backup.

```
# /var/opt/oracle/bkup_api/bkup_api bkup_status
```

5. Reenable the scheduled backup configuration (`crontab`).

To perform a database recovery, you can use:

- The `dbaascli` utility (as described below)
- The Database Cloud Service console: Read *Restoring from a Specific Backup* or *Restoring from the Most Recent Backup* or *Restoring to a Specific Point in Time* .

1. Connect to the database deployment VM.
2. Disable the scheduled backup configuration (`crontab`).
3. Restore a backup and recover the database.

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

4. Reenable the scheduled backup configuration (`crontab`).

Oracle Internal & Oracle Academy Use Only

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

Lesson 5 Configuring Network Access to Database Deployment

After completing this lesson, you should be able to:

- Configure network access to a database deployment
- Configure network settings through Oracle Compute Cloud Service

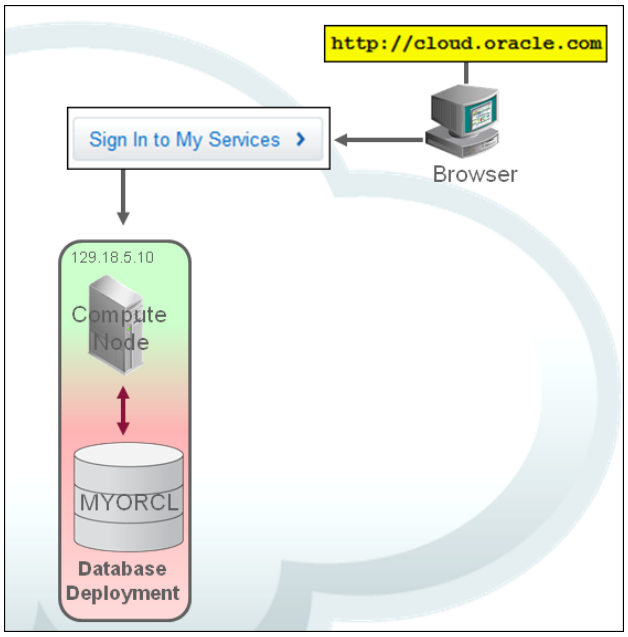
Topics:

Oracle Internal & Oracle Academy Use Only

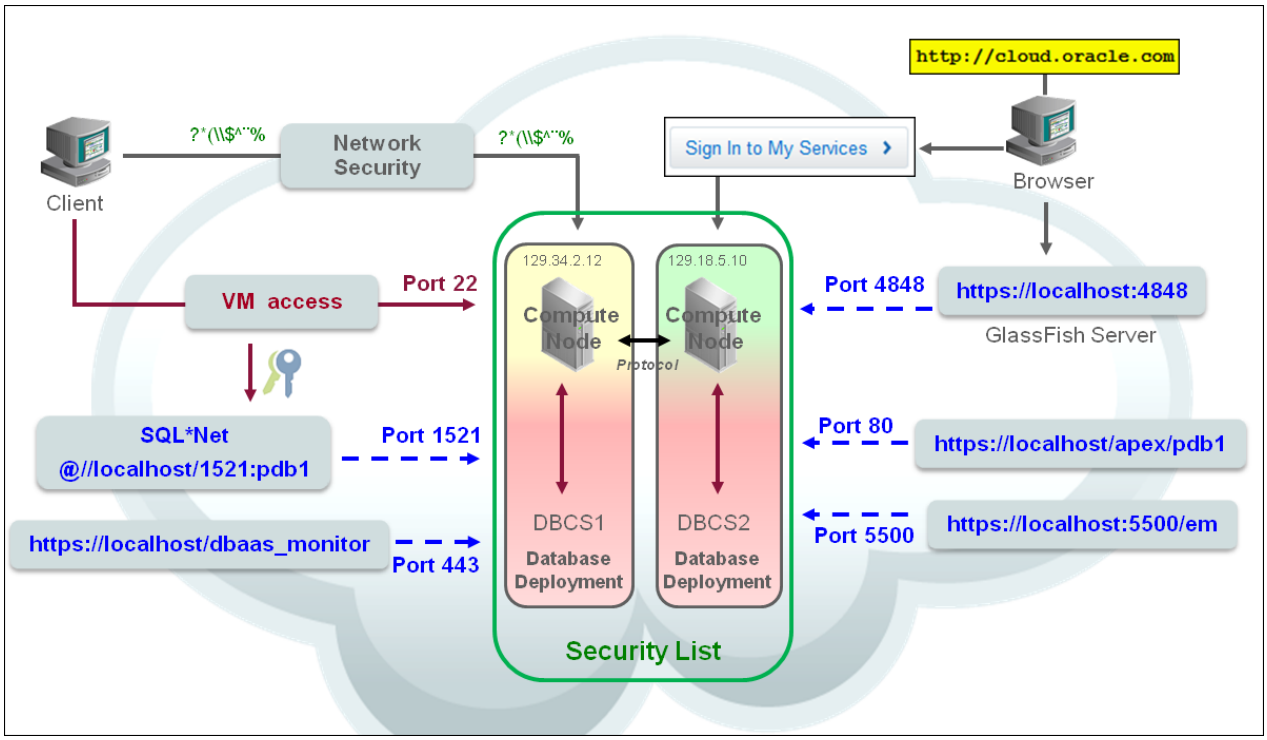
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Security Lists

- When you create a database deployment, the associated compute node by default does not allow access from any other database deployment compute node or external host.



- You can **create security lists** to control network traffic among database deployment compute nodes.

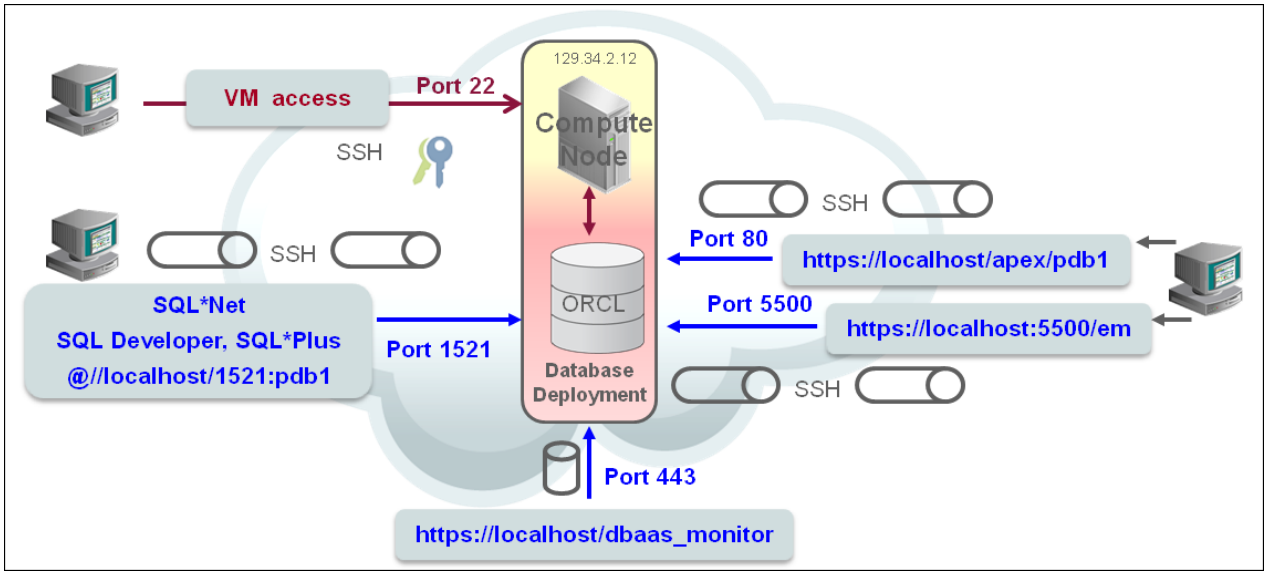


Oracle Internal & Oracle Academy Use Only

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

Opening Ports to Compute Node

When a database deployment is created, network access to the compute node of the database deployment is limited to Secure Shell (SSH) connections on port 22.



This restriction ensures that the database deployment is secure by default. Access requires the private key that was supplied when the database deployment was created; the public key was added to the database deployment.

Two ways to enable a port to a database deployment:

- The first requires access to the Compute Cloud Service console to open ports to a set of IP addresses.
  - The access is persistent.
  - The port remains open to communication all the time.
    Read *Enabling Access to a Compute Node Port*.
- The second is to create a tunnel for port forwarding using `ssh`.
  - The configured `ssh` process must be running on the consumer side of the tunnel.
    Read either *Creating an SSH Tunnel Using the ssh Utility on Linux* or *Creating an SSH Tunnel Using the PuTTY Program on Windows* according to your consumer side type.

Oracle Internal & Oracle Academy Use Only

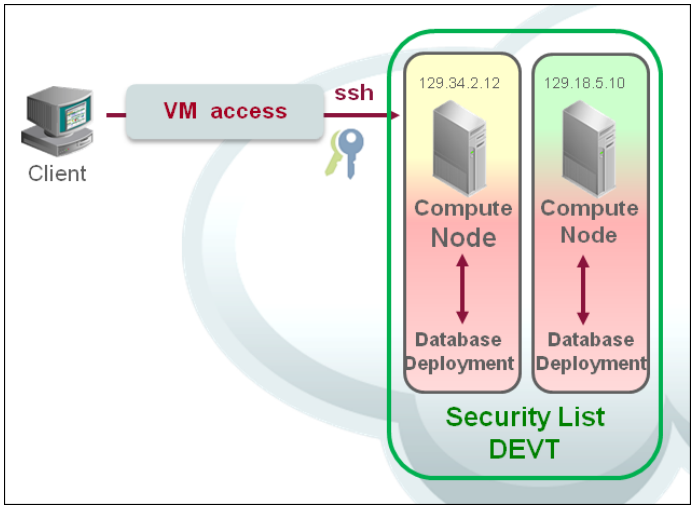
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Fine-Grained Control of Network Traffic

Control network traffic among database deployment compute nodes in the same security list

To enable unrestricted communication between your database deployment compute nodes dedicated to development, create a security list and then associate development database deployment compute nodes to the security list.

- A security list associates a set of database deployment compute nodes and policies.
- When you associate a compute node to a security list, the compute node can communicate with all the other compute nodes in the same security list, but not with the compute nodes of other security lists.
  Read *About Network Settings*.



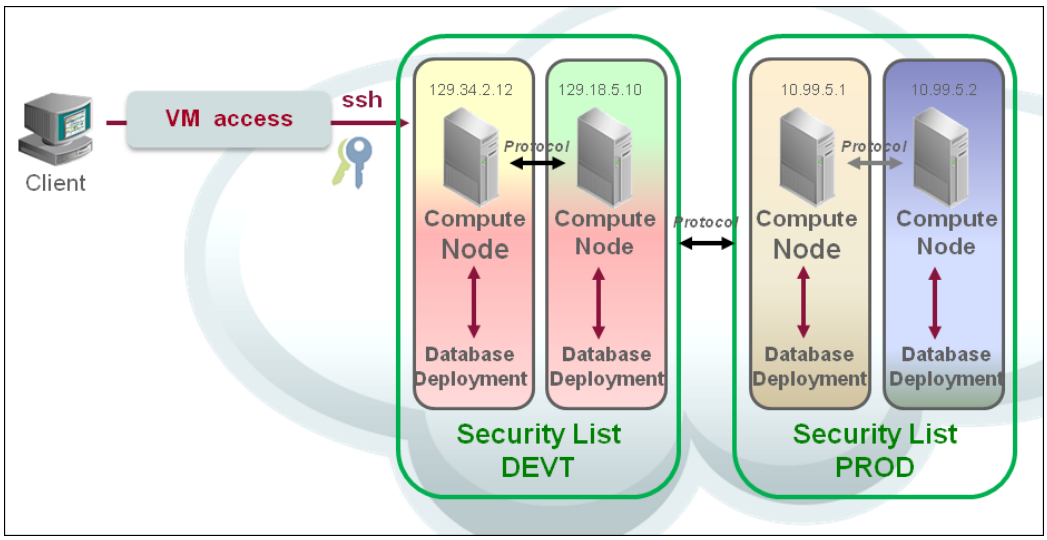
Control network traffic between database deployment compute nodes in different security lists

To enable communication between your development security list and associated database deployment compute nodes and some database deployment compute nodes of the production associated with the production security list, create security rules.

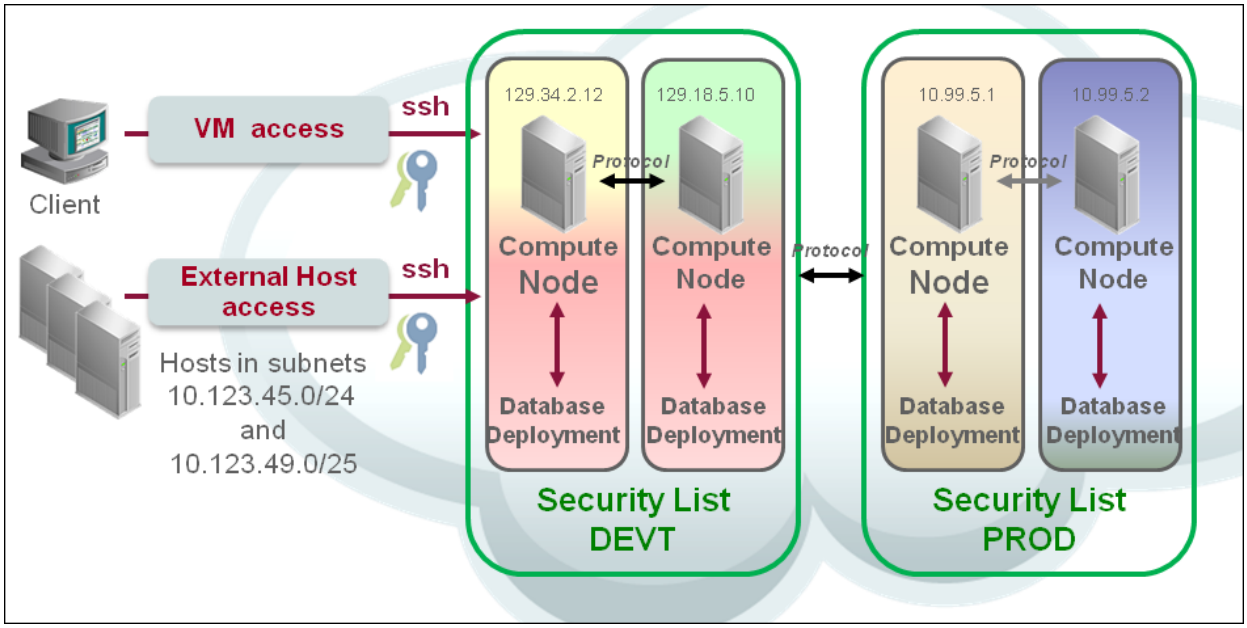
- Creating security rules overrides these default access settings of security lists.
- Each security rule defines a specific communication path, consisting of:
  - A source (can be a security list or security IP list)
  - A destination (can be a security list)
  - A protocol-port combination over which communication is allowed (security application)

Oracle Internal & Oracle Academy Use Only

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



Control network traffic between database deployment compute nodes and external hosts

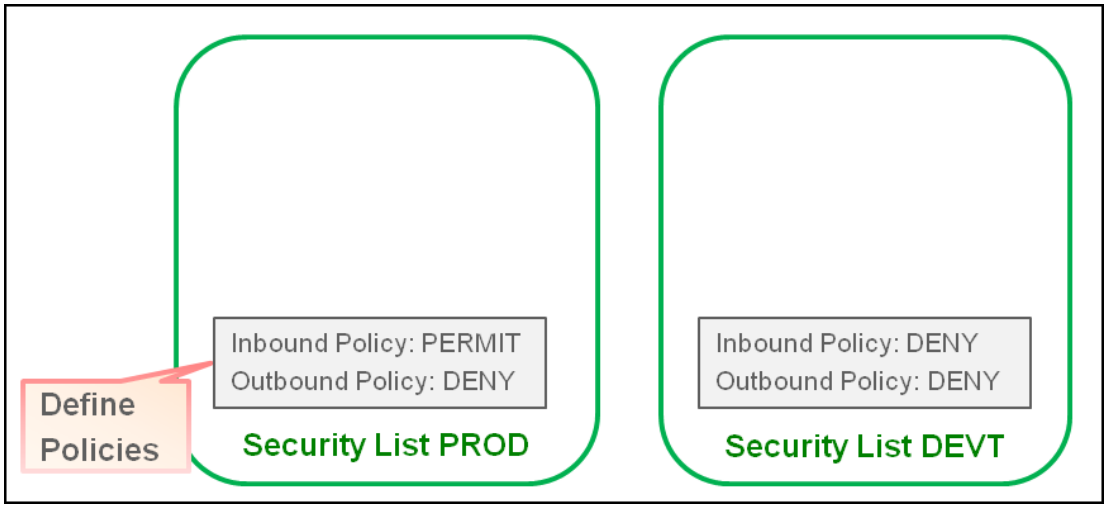


Oracle Internal & Oracle Academy Use Only

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

Configuring Network Security

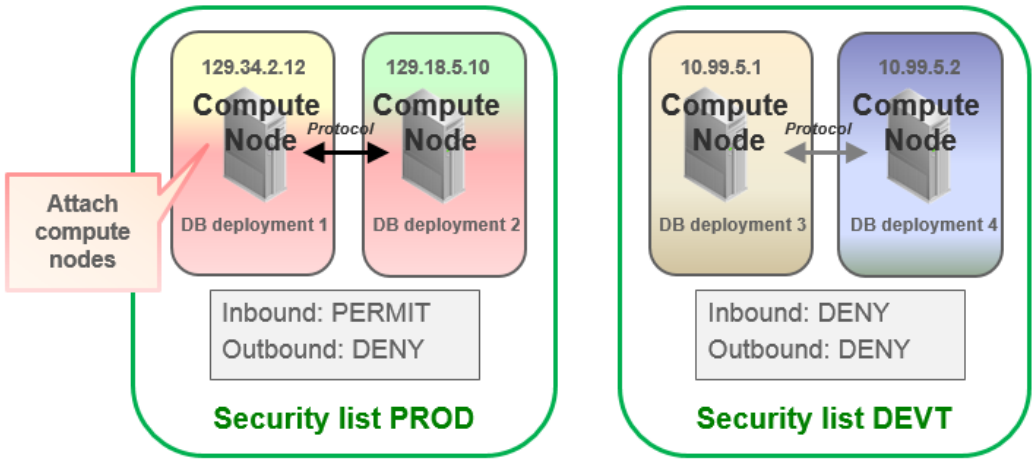
1. **Create security lists.**



Read *Creating a Security List*.

- The inbound policy controls the flow of traffic into the security list.
  For example, if the inbound policy is set to `permit`, packets from all sources using any port or protocol are permitted to the compute nodes in the security list. To control the flow of traffic to the compute nodes in a secu rity list, ensure that the inbound policy is set to `deny`, and then define security rules to allow only traffic from specified sources to access your compute nodes using specified ports and protocols.
- The outbound policy controls the flow of traffic out of the security list.
  For example, if the outbound policy is set to `deny`, packets can't flow out of the security list. To allow instances in a security list to communicate with hosts outside the security list, set the outbound policy to `permit`.
- By default, a security list has its inbound policy set to `deny` and outbound policy set to `permit`.

2. **Associate compute nodes to security lists.**



Read *Adding an Instance to a Security List*.

Associating compute nodes to security lists defines default security rules within the security list.

Oracle Internal & Oracle Academy Use Only

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

The PROD security list allows compute nodes 1 and 2 in the production environment to communicate with each other over any protocol. Hosts outside this security list may communicate with any production compute node. No production compute node can communicate with any host outside this security list.

Compute nodes 3 and 4 are attached to the DEVT security list. This allows all compute nodes in the development environment to communicate with each other over any protocol. By default, no host outside this security list can communicate with any development compute nodes, and no development compute node can communicate with any host outside this security list.

Oracle Internal & Oracle Academy Use Only

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

Defining Security Rules

What are security rules?

Security rules are essentially firewall rules defining a specific communication path to permit traffic between:

- Compute nodes in different security lists
- Compute nodes and external hosts

Read *About Security Rules*.

Security Rule = Source (*Security List* or *IP list*) + Protocol + Destination (*Security List or IP list*)

where the protocol-port combination allows communication (Read *About Security Applications*).

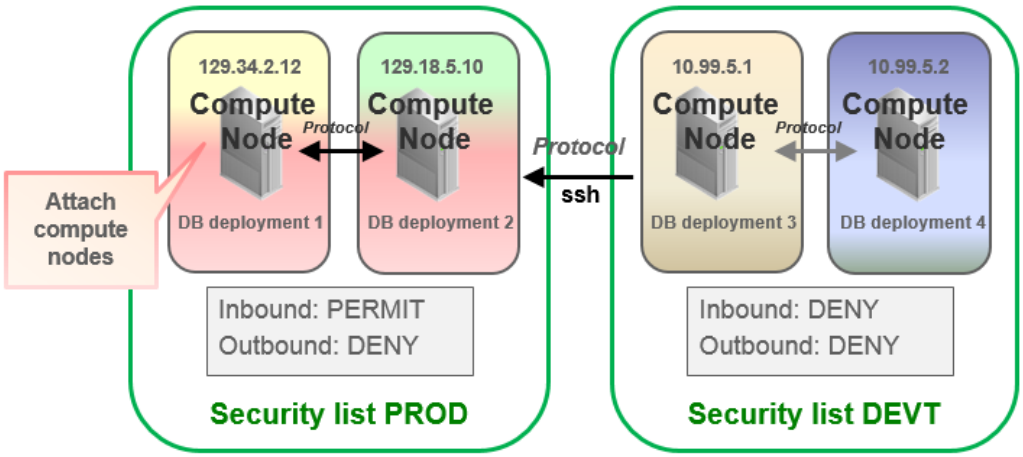
How to create security rules?

To create rules, read *Creating a Security Rule*. By default, new security rules are enabled.

Examples of security rule

DEVT to PROD via protocol `ssh`

Any development compute node of the DEVT security list can communicate over SSH with any production compute node of the PROD security list. No production compute node can communicate over SSH with any development compute node.

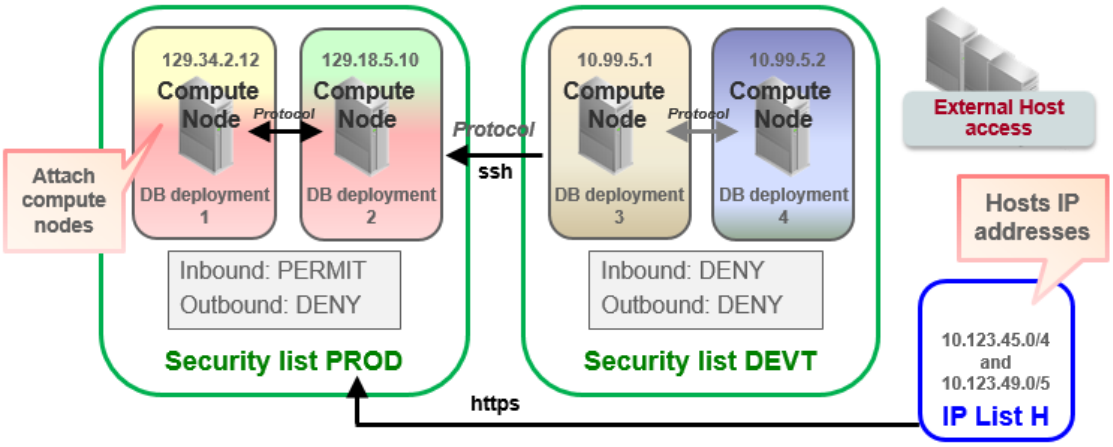


External hosts from the H security IP list to PROD via protocol `https`

Any host on the Internet can communicate over `https` with production compute nodes of database deployments 1 and 2, but not to development compute nodes of database deployments 3 and 4.

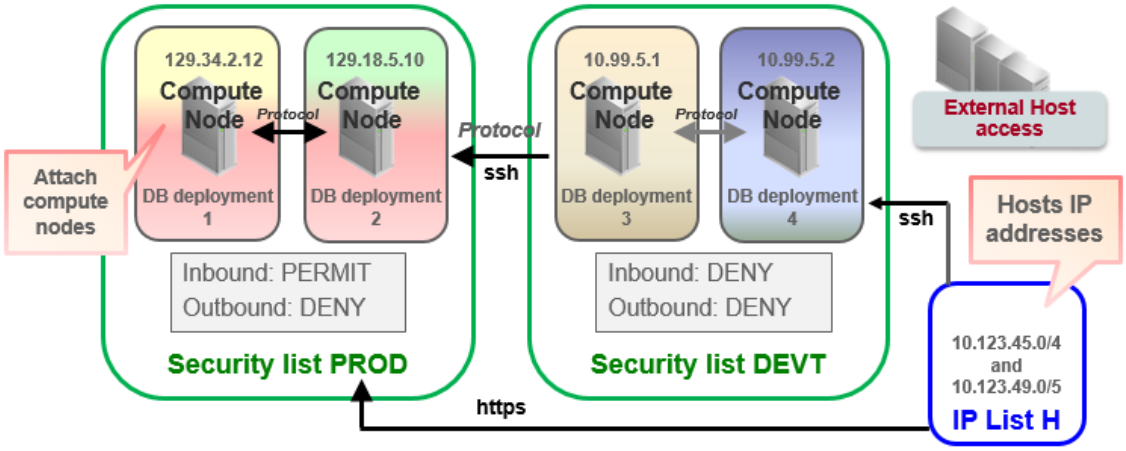
Oracle Internal & Oracle Academy Use Only

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



H security IP list to DEVT via protocol ssh

Any host in the 10.123.45.0/4 and 10.123.49.0/5 subnets can communicate over `ssh` with compute nodes of database deployments 3 and 4.



Oracle Internal & Oracle Academy Use Only

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

Lesson 6 Overview of Oracle Cloud Service Security

After completing this module, you should be able to:

- Understand the Oracle Cloud Service security
- Ensure physical and operating system security
- Configure users' authentication and their roles
- Understand the PDB isolation
- Protect and secure data of the database deployment VM and the database
- Ensure data security over network
- Ensure application level security

Topics:

- Oracle Cloud Service Security Overview
- Enforcing Security at Different Steps
- Physical and Operating System Security
- User Authentication - Services and Compute Node Access
- User Authentication - Database Access
- Database Access with EM Express
- Database Access with EM Cloud Control
- Database Access with DBaaS Monitor and APEX Consoles

Oracle Internal & Oracle Academy Use Only

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

Oracle Cloud Service Security Overview

As you prepare to move your on-premises environment to the database deployment environment, consider the following guidelines:

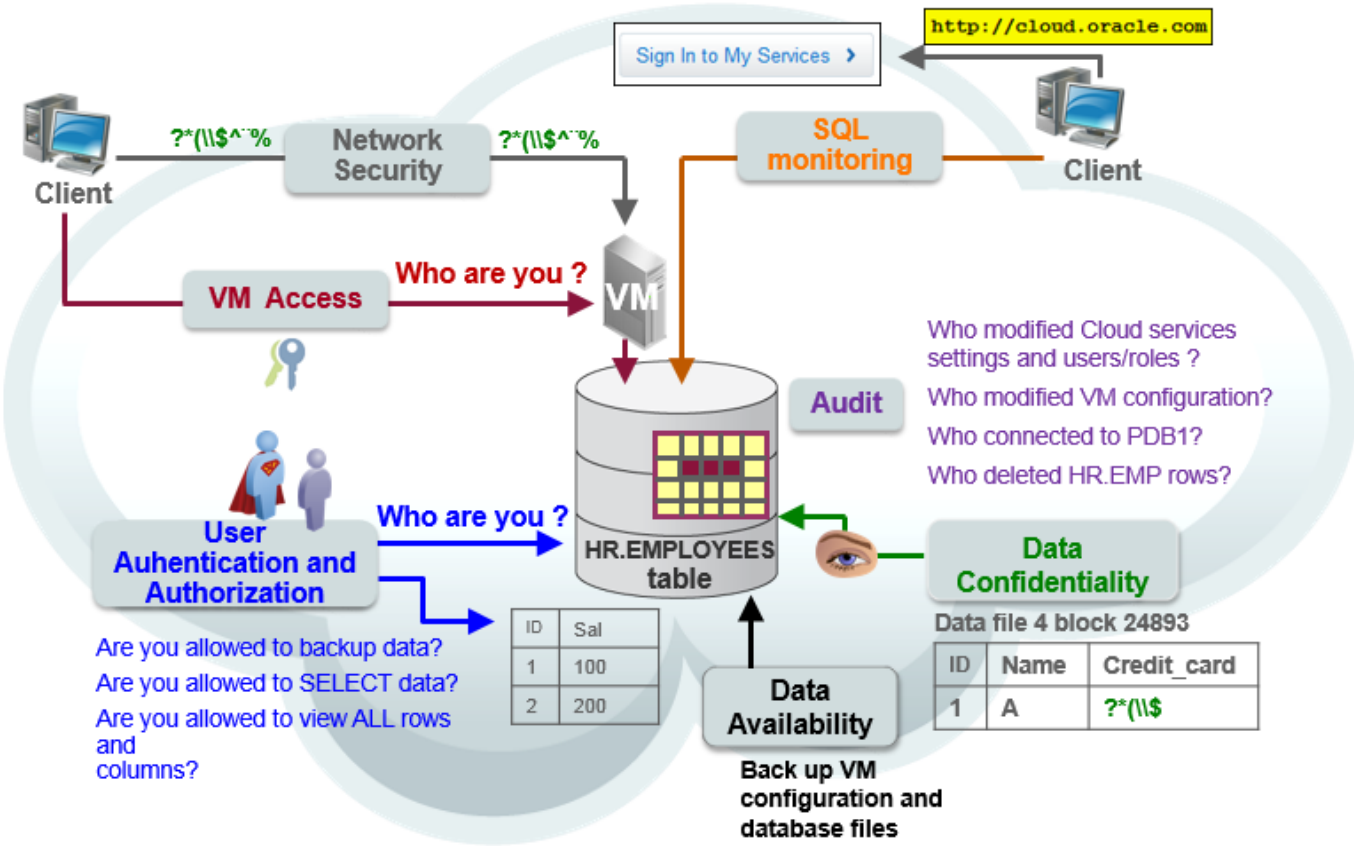
- Security best practices for on-premises or Cloud environment still include people, processes, and products.
- These guidelines are more specific to the Cloud environment:
    - Educate and familiarize your users with Oracle Cloud identity management services:
        - Use credentials and an identity domain to sign in.
        - Set up challenge questions for the password reset procedure.
    - Grant access to an Oracle Cloud service only to employees who require the service.
    - Grant access to the database deployment compute node only to employees who need to perform operations on the compute node.
    - Isolate sensitive data in distinct tenants or PDBs.

Oracle Internal & Oracle Academy Use Only

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

Enforcing Security at Different Steps

Security is the foundation of Oracle Cloud services including physical security, operating system and virtualization layer security, and tenant isolation. In a cloud-based application environment, security is a shared responsibility model. You need to take responsibility for implementing best practices as applicable to your business or organization. End-to-end security requires network and application-level security in your own environment.



Enforcing security in a database deployment environment means that the data stored in the database deployment database is protected, kept confidential, and made available at different steps of accessibility:

- When users sign in to Cloud services
- When you access the compute node of the database deployment over an SSH tunneling connection using an SSH key pair and mainly your own private key for authentication
- When data is transmitted over the network using encryption such as when authenticating to the compute node or copying on-premises database files to the database deployment database
- When you log in to the database through different applications (EM Express, APEX, DBaaS Monitor, SQL*Plus, SQL Developer, RMAN) using a database user account authentication method such as a password
- When you access the data on the compute node such as configuration files by using different UNIX user permissions, or the data in tables of the database by using different database user privileges

Oracle Internal & Oracle Academy Use Only

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

Physical and Operating System Security

Physical security

The physical security of the server that hosts the database deployment compute node and database allocated to the customer is secured by Oracle. The customer has no access to the physical server.

OS credentials to access the database deployment compute node

- Minimal privileged Linux `oracle` user (normal shell access)
- Privileged Linux `opc` user (root commands access)

  Login to the database deployment compute node requires a secure access from remote hosts by using a secure Linux shell. When a database deployment is created, network access to the database deployment's compute node is limited by default to SSH connections on port 22. This restricted access ensures that the instance is secure by default. To be able to log in to the compute node, the OS user authenticates by an SSH key pair.

OS security

- SSH access
- OS commands

  On the database deployment compute node, the `opc` OS user can have `root` privilege. He can perform the following operations:
  - Load and run software in the compute node environment
  - View log files from the instance creation stored in subdirectories of `/var/opt/oracle/log`. The directory is accessible only to the root user.
- Directory access for log files as `root` OS user
- Directory access for database files
- Directory access for external files
- Access to binaries such as `sqlplus` , `sqlldr`, `rman`, `dbca`
- Access to utilities such as `dbaascli`, `bkup_cli` as `opc` OS user

  On the database deployment database instance, the `oracle` OS user has full administrative privileges.

  - Database files are stored in directories:
    - `/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.
  - The Oracle database software is installed in the following locations:
    - Oracle Database 12*c*: `/u01/app/oracle/product/12.1.0/dbhome_1`
    - Oracle Database 11*g*: `/u01/app/oracle/product/11.2.0/dbhome_1`
  - The Oracle Database Cloud software is installed in the following location:

    `/var/opt/oracle`
    - `dbaascli`

Oracle Internal & Oracle Academy Use Only

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

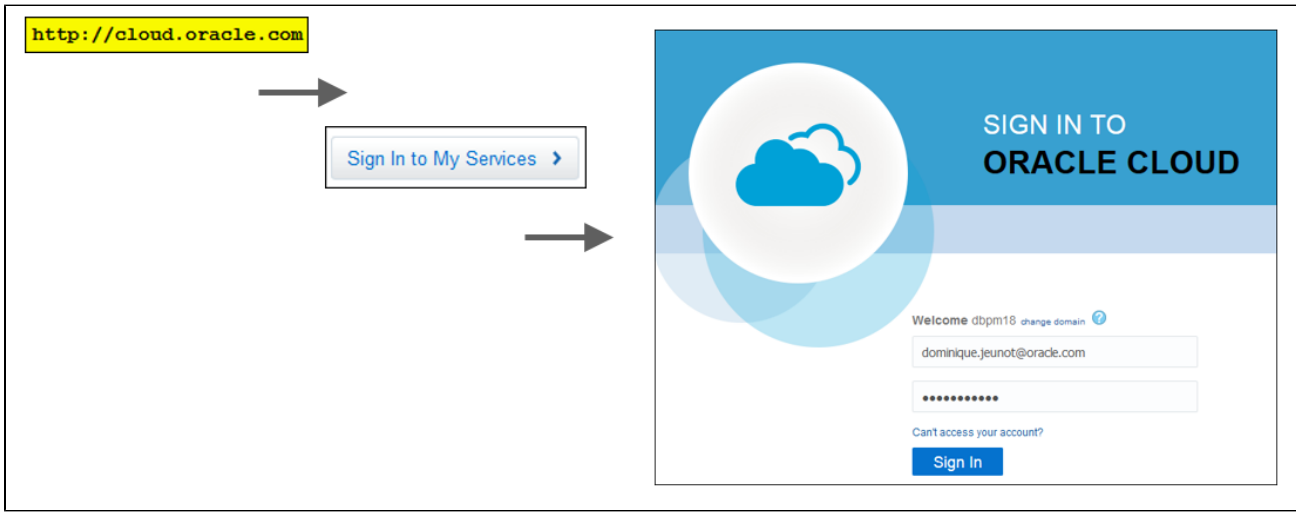
- `bkup_api`
- `exapatch`

Oracle Internal & Oracle Academy Use Only

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

User Authentication - Services and Compute Node Access

A basic security requirement is that you must know your users. You must identify them before you can determine their privileges and access rights, so that you can audit their actions on the data. Users are authenticated at different steps before they are allowed to create a database deployment database session.

- **Cloud services: Cloud account credentials**
- 

  If for example, the user wants to use EM Express to monitor his database deployment database, one of the methods is that the customer logs in the Cloud environment to access the services he has paid the subscription for. Therefore, he needs to sign in to the “My Account” application providing the Oracle.com account credentials.

- **Database deployment compute node Linux user: SSH access using a key pair:**

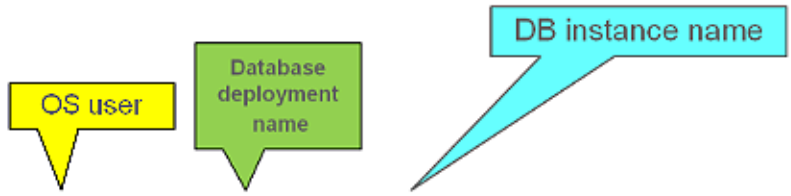
  

  If for example, the user wants to export data from his database deployment database, the user first logs in the database deployment compute node under a Linux OS user using a Secure Shell (SSH) access to the compute node and authenticating himself with an SSH key pair.

Oracle Internal & Oracle Academy Use Only

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

User Authentication - Database Access




```
[oracle@MYDBCS ~]$ . oraenv
ORACLE_SID = [MYORCL] ?
The Oracle base has been set to /u01/app/oracle
[oracle@MYDBCS ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Mon Jun 6 09:42:29 2016

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Oracle Label Security option

SQL>
```

Logged in to the compute node, launch any Oracle binary or Cloud tooling:

- SQL*Plus/SQL Loader/Oracle Data Pump/RMAN/DBCA
- `bkup_api` / `dbaascli`

When the OS user is connected, he or she can execute any shell command but cannot execute any `root` command. To run the `root` commands, the OS user must be logged as the `opc` user and perform a `sudo` command. `sudo -s` starts a shell as the `root` user where you can enter multiple commands or `sudo -c 'command'` performs a single command as the `root` user.
The environment variables such as `ORACLE_SID` and `ORACLE_HOME` are set to those defined by default during the database deployment creation for the `oracle` and `opc` default users. Read *Linux User Accounts*.

```
[oracle@MYDBCS ~]$ dbaascli orec --args -help
DBAAS CLI version 1.0.0
Executing command orec --args -help
Unable to run command. This command should be run as user: root. Currently: oracle
[oracle@MYDBCS ~]$
```

Oracle Internal & Oracle Academy Use Only

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

```
[opc@MYDBCS ~]$ /var/opt/oracle/bkup_api/bkup_api bkup_status
API::ERROR Api requires root rights or sudoer
[opc@MYDBCS ~]$ sudo -s
[root@MYDBCS opc]#
[root@MYDBCS opc]#  /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Single Instance
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
* Current backup settings:
  catalog: no
  type: diskoss
* Last registered Bkup: 06-05 02:18 API::15851:: Starting dbaas backup process
* Bkup state: running
**************************************************
* API History: API steps
  API:: NEW PROCESS 15851
  API:: Starting dbaas backup process
**************************************************
* Backup steps
 -> API:: Oracle database state is up and running
 -> API:: Performing backup to local storage (primary backup)
 -> API:: Executing rman instructions
*
* RETURN CODE:0
##################################################
[root@MYDBCS opc]#
```

Oracle Internal & Oracle Academy Use Only

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

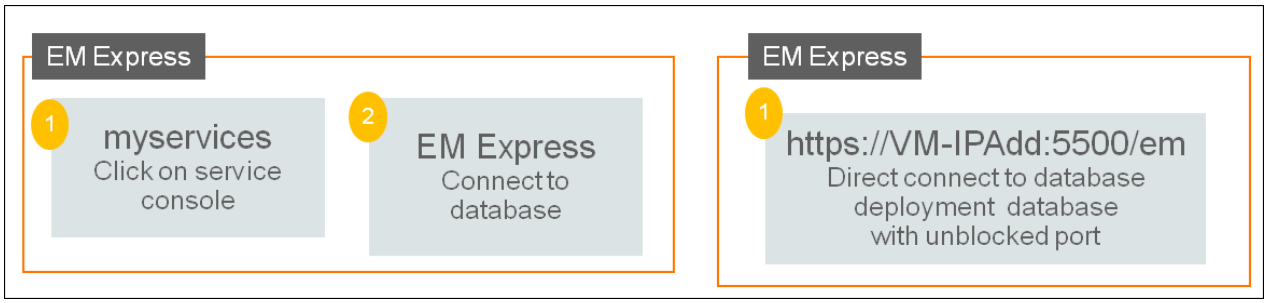
Database Access with EM Express

When a database deployment is created, the EM Express port (5500) is blocked to ensure network security. Launch EM Express consists of using the little icon from the right-hand side from the list of available consoles of the Oracle Database Cloud Service page.

First way to access EM Express:

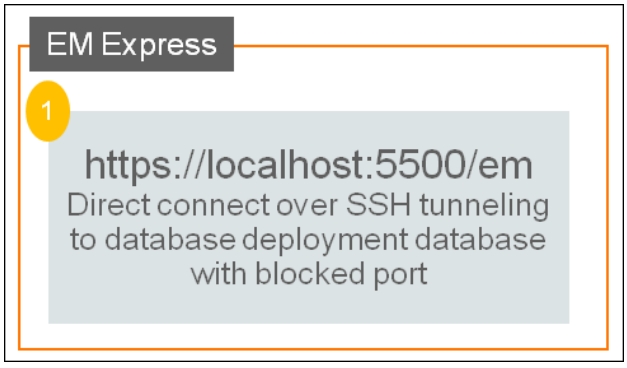
An attempt is made to access the database deployment database instance. The port being blocked, the access fails.

You can unblock the EM Express port 5500. Read *Enabling Access to a Compute Node Port*.



Second way:

The second way consists of creating an SSH tunnel to the EM Express port (5500). After you created an SSH tunnel to 5500 port on the database deployment's VM, you can use the direct access as follows: https://localhost:5500/em. Read *Creating an SSH Tunnel to a Compute Node Port*.



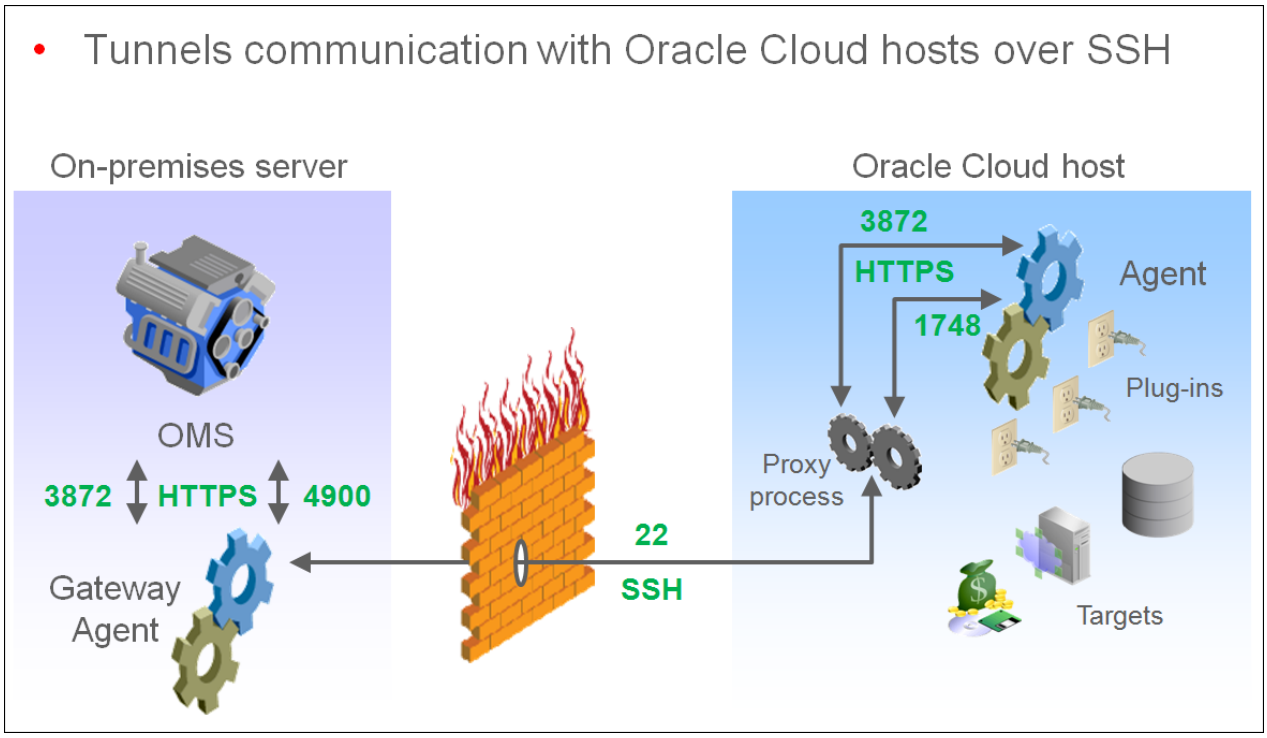
Oracle Internal & Oracle Academy Use Only

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

Database Access with EM Cloud Control

In order for an OMS to manage and monitor a host and its targets on the other side of a firewall, directional holes corresponding to the communications between OMS and targets need to be opened in the firewall to allow full functionality. Depending upon the targets being monitored, this can result in a large number of holes in the firewall, and such a topology is anathema to basic security principles and the purpose of firewalls.

The Hybrid Cloud Agent elegantly resolves this topological dilemma for hosts in the Oracle Cloud in several ways using one tunnel for all traffic.



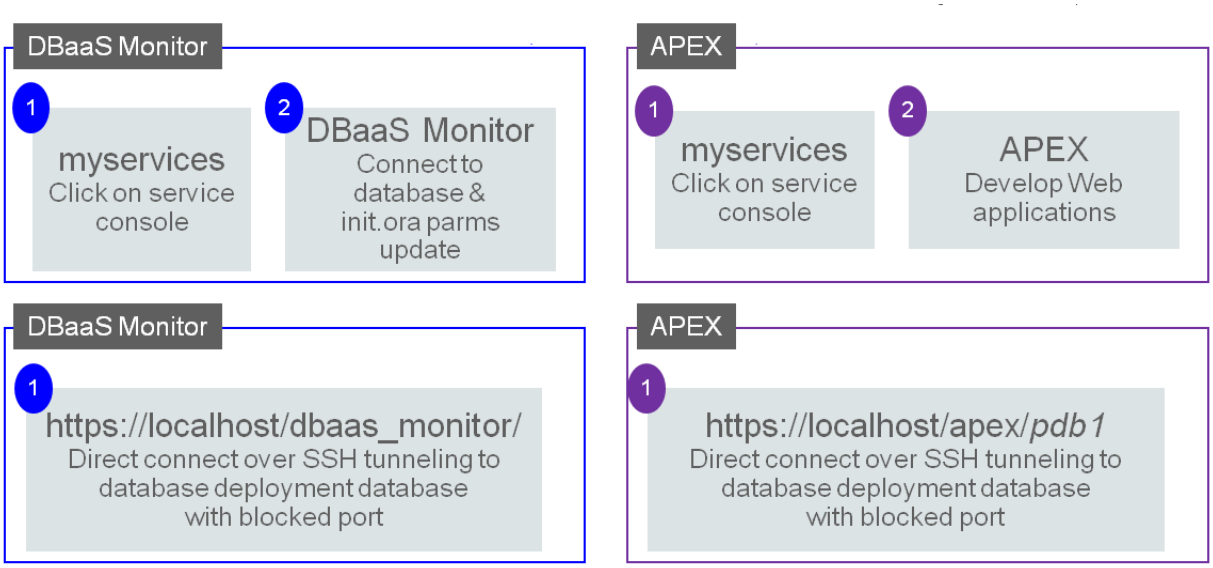
- A Secure Shell (SSH) tunnel is configured in the OMS by providing named SSH credentials for the Oracle Cloud.
- When the OMS initiates communication with the agent in the Oracle Cloud, it does so via the SSH tunnel using an EMCTL dispatcher on the Oracle Cloud host.
- When the Agent initiates communication with the OMS, it does so via a proxy process that, in turn, communicates via the SSH tunnel with an on-premises agent configured as a gateway agent.
- When the OMS initiates direct communication with any of the targets on the Oracle Cloud managed host, such as a database, it does so via the SSH tunnel.

Read *Enabling Hybrid Cloud Management*.
Read also *Using Oracle Enterprise Manager Cloud Control with Database as a Service*.

Oracle Internal & Oracle Academy Use Only

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

Database Access with DBaaS Monitor and APEX Consoles



Oracle DBaaS Monitor console

DBaaS Monitor provides monitoring and management of the Oracle database and listener on database deployment. DBaaS Monitor provides quick and easy access to a variety of information about the database instance running on a database deployment.

First way to access DBaaS Monitor console

The first way consists of releasing the HTTPS port (443) on the database deployment's compute node. This will unsecure the access to the database deployment database instance through on-instance database monitor. Read *Enabling Access to a Compute Node Port.*
In this case, direct your browser to the URL `https://VM-ip-address/dbaas_monitor/`

Second way

The second way consists of using the HTTPS port (443) on the database deployment's compute node being kept blocked by default. The access fails. In this case, create an SSH tunnel to port 443 on the database deployment's compute node. Read *Creating an SSH Tunnel to a Compute Node Port*.

Read *Accessing Oracle DBaaS Monitor*.

Oracle APEX console

Application Express enables you to design, develop, and deploy beautiful, responsive, database-driven applications using only your web browser.
Accessing the Oracle Application Express Administration Console follows the same rules as accessing the EM Express and DBaaS Monitor consoles. Read *Accessing the Oracle Application Express Administration Console*.

The access works with secured port 443. The ways to use APEX are exactly the same as for DBaaS Monitor. The URLs are respectively secured `https://localhost/apex/pdb1` and unsecured `https://VM-ip-address/apex/pdb1`. Then the user now has to provide a database account authentication.

Oracle Internal & Oracle Academy Use Only

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

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.

Oracle Internal & Oracle Academy Use Only

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

Lesson 7 Implementing Database Deployment Security

After completing this lesson, you should be able to:

- Secure the access and configuration files on the database deployment VM
- Maintain the availability of the data of the database deployment database
- Control the access in the database deployment database
- Protect the data in the database deployment database
- Audit operations performed in the database deployment

Topics:

- Securing Access and Configuration Files on Database Deployment Compute Node
- Backing Up OS and Database Configuration and Data Files
- Restricting Access to Database Deployment Database
- Protecting Data in Database Deployment
- Auditing

Oracle Internal & Oracle Academy Use Only

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

Securing Access and Configuration Files on Database Deployment Compute Node

Certain operations like database backup and recovery or compute node reboot require the `sudo` command, and therefore require a connection as `opc`.

- To perform operations that require root-user access, use the `sudo` command when connected as `opc`.
- The `root` administrator for the system exists but you do not have access to this account. Connect as `opc` and run the `sudo` command:
  - Option to add other sudoers than `opc`
  - Option to edit OS configuration files like `/etc/passwd`, `/etc/group`

Secured Files	OS Permissions
/etc/sudoers	-r--r-----. 1 root root
/etc/ssh/sshd_config	-rw------- 1 root root
/etc/passwd & **/etc/group**	-rw-r--r-- 1 root root

Oracle Internal & Oracle Academy Use Only

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

Backing Up OS and Database Configuration and Data Files

Configuration Change	File
Database configuration file	Edit /home/oracle/bkup/<*dbname*>/dbcfg.spec
System configuration file	Edit /home/oracle/bkup/<*dbname*>/oscfg.spec

The two configuration files providing the list of system and database files to be backed up automatically are automatically backed up during a backup.
Refer to *Practice 4-1 Backing up Database Deployment.*

Oracle Internal & Oracle Academy Use Only

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

Restricting Access to Database Deployment Database

Restrict access to your database deployment database as you do in on-premises databases.

1. Connect to the database deployment non-CDB (Oracle Database 11*g*) or PDB (Oracle Database 12*c*) as the DBA.
2. Create a new user.
3. Grant the new user the privileges required to complete his job tasks and only those privileges.
    a. If the user is a schema, create the schema objects while you are still connected under the DBA, and do not grant the schema user ANY privileges, not even the `CREATE SESSION` privilege.
    b. If the user is an end-user, grant the user object privileges.

Oracle Internal & Oracle Academy Use Only

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

Protecting Data in Database Deployment

Restrict access to the database deployment database

- SSH key to access to the compute node
- Password to log in to the database

Control data access in the database deployment database

Use the same security methods as in on-premises databases.

- Privileges
- Privilege Analysis
- Virtual Private Database (VPD)
- Oracle Label Security (OLS)
- Database Vault

Protect the confidentiality of data in the database deployment database

- Data Redaction
- Data Masking
- Data encryption
  - User-created tablespaces are encrypted by default with the AES128 encryption algorithm because the database instance parameter `encrypt_new_tablespaces = cloud_only`.
    Read *Creating Encrypted Tablespaces* .

    In `/u01/app/oracle/admin/<db-name>/tde_wallet`, the `ewallet.p12` wallet saves the TDE encryption password and the keys. The password-based wallet is the default keystore for TDE master keys. Its location is defined in `sqlnet.ora` through the parameter
    ```
    ENCRYPTION_WALLET_LOCATION =
    (SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/
    MYORCL/tde_wallet)))
    ```

    The TDE wallet is used by the database to encrypt the tablespaces other than non-user defined tablespaces created initially by DBCA.
    The TDE wallet is also used by RMAN to encrypt backups as cloud backups, and should always be encrypted.
    The TDE wallet is also configured as an auto-login wallet (`cwallet.sso`) so that there is no user interaction required across a database and compute node restart by default. But one can change to manual if desired.

    - You can change (rotate) the master encryption key by using the `tde rotate masterkey` sub command of the `dbaascli` utility. The wallet stores all the previous keys internally so it can decrypt data that has been encrypted using the earlier key. The new key will be used for all new encryption. The rotate requires the key-store password which is set during initial provisioning. The TDE key rotation is similar to on-premise. Read *Managing Tablespace Encryption*.
    - Newly created PDBs (from scratch or by plugging) require the creation and activation of a

Oracle Internal & Oracle Academy Use Only

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

master encryption key for the PDB. The SYSTEM, SYSAUX, and TEMP tablespaces are not encrypted, but any user-created tablespace will require the key. Read *Creating and Activating a Master Encryption Key for a PDB*.

- Backups are automatically encrypted.

  In `/u01/app/oracle/admin/<db-name>/opc_wallet`, the `cwallet.sso` wallet (auto-login wallet) is used by RMAN to connect to the cloud storage for backups with no human interaction to enter a wallet password.

- Tooling can connect as SYS without specifying the wallet password.

  In `/u01/app/oracle/admin/<db-name>/db_wallet`, the `cwallet.sso` wallet saves the database SYS user password. Its location is defined in `sqlnet.ora` through the parameter `WALLET_LOCATION = (SOURCE=(METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/MYORCL/db_wallet))(METHOD=FILE))`.

Protect data in transit

- Security of data across the network is provided by native Oracle Net encryption and integrity capabilities. Read *Security of Data in Transit*.
  - Encryption of network data provides data privacy so that unauthorized parties are not able to view data as it passes over the network.
  - Integrity algorithms protect against data modification and illegitimate replay.
    Read *Using Network Encryption and Integrity*.

Oracle Internal & Oracle Academy Use Only

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

Auditing

Compute Node Connections and Actions

- Control account connections: `/var/log/secure`

```
Jun  4 23:49:06 localhost sshd[26163]: input_userauth_request: invalid user root
Jun  4 23:49:06 localhost sshd[26163]: Received disconnect from [redacted]: 11:
Jun  4 23:50:52 localhost sshd[26956]: User root from [redacted] not allowed because not lis
ted in AllowUsers
Jun  4 23:50:52 localhost sshd[26957]: input_userauth_request: invalid user root
Jun  4 23:50:52 localhost sshd[26957]: Received disconnect from 121.18.238.29: 11:
Jun  4 23:55:42 localhost sshd[29180]: Did not receive identification string from [redacted]
```

- Updated configuration files: `/etc/passwd`, `/etc/crontab`
- Find files modified within the last 2 months:
  `$ find /etc -mtime 3 -printf "%u %p"`
- Find all of the history commands:
  `$ cat /home/oracle/.bash_history`

Auditing in the Database

The auditing mode configured in the Oracle Database 12*c* database deployment database is the mixed mode. Predefined audit policies are enabled in the Oracle Database 12*c* database deployment database:

- `ORA_LOGON_FAILURES`: audits logon failures
- `ORA_SECURECONFIG`: audits 46 system privileges like `GRANT ANY PRIVILEGE`, `GRANT ANY ROLE`, `EXEMPT ACCESS POLICY`, `DROP PLUGGABLE DATABASE`
- `ORA_ACCOUNT_MGMT`: audits users creation/alter/drop
- `ORA_DATABASE_PARAMETER`: audits parameters changes, recreation of the spfile

Remark: The `/var/opt/oracle/cleandb/cleandblogs.pl` script automatically, every week, cleans up database audit files (`*.aud`) that are older than 14 days by default. Read *Managing the Log and Diagnostic Files on Database as a Service* to know how to change the default retention period.

Oracle Internal & Oracle Academy Use Only

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

Lesson 8 Monitoring the Database Deployment

After completing this module, you should be able to:

- Monitor the database deployment database
- Monitor the database deployment listener
- Monitor the database deployment OS

Topics:

- Monitoring the Database Deployment Database
- Monitoring the Database Deployment Listener
- Monitoring the Database Deployment OS
- Administering Oracle REST Data Services

Oracle Internal & Oracle Academy Use Only

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

Monitoring the Database Deployment Database

What does DBaaS Monitor display against the database?

- *View and Modify Initialization Parameters*
- *View Alert Log Entries and Check for Errors*
- *View Tablespace and Segment Space Usage*
- The backup operations
- The sessions currently logged on
- The wait events
- Real Time SQL Monitor

Which actions can you perform against the database?

- *Start and Stop the Database Instance*
- *Administer Pluggable Databases*
    - Create PDBs
    - Clone PDBs
    - Unplug/Plug PDBs
    - Close/Open PDBs
    - Drop PDBs

Oracle Internal & Oracle Academy Use Only

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

Monitoring the Database Deployment Listener

What does DBaaS Monitor display against the listener?

- *View Listener Status Information*

Which actions can you perform against the listener?

- *Start and stop the listener*

Oracle Internal & Oracle Academy Use Only

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

Monitoring the Database Deployment OS

What does DBaaS Monitor display against the OS?

- The global used RAM and the memory usage by process
- The CPU usage
- The overall OS storage and the percentage used on each file system
- The OS processes

Note: You can filter out columns displayed in results.

Which actions can you perform against the OS?

None.

Oracle Internal & Oracle Academy Use Only

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

Administering Oracle REST Data Services

The REST APIs allow you to create and manage database deployments without necessarily using the Oracle Cloud UI.

- By default, when a database deployment is created, Oracle REST Data Services (ORDS) is started.

```
[opc@MYDBCS ~]$ sudo -s
[root@MYDBCS opc]# /u01/app/oracle/product/ords/ords status
INFO: Obtaining Oracle REST Data Services status...
INFO: Oracle REST Data Services is already running with PID 5569
[root@MYDBCS opc]#
```

- What actions can you perform on a database deployment? Read *About the REST APIs* .
  - Create a database deployment
  - Delete a database deployment

```
curl -v -X DELETE -u <your_username>:<your_password> -H
"X-ID-TENANT-NAME:<your_domain>" \
https://dbaas.oraclecloud.com/paas/service/dbcs/api/v1.1/instances/<your_domain>/
<your_database_deployment>
```

```
* About to connect() to dbaas.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.107... connected
* Connected to dbaas.oraclecloud.com (160.34.0.107) port 443 (#0)
...
* Server auth using Basic with user 'dominique.jeunot@oracle.com'
> DELETE /paas/service/dbcs/api/v1.1/instances/oucloudusajul10/DJDB
CS HTTP/1.1
...
> X-ID-TENANT-NAME:oucloudusajul10
>
< HTTP/1.1 202 Accepted
...
< Service-URI: https://dbaas.oraclecloud.com:443/paas/service/dbcs/
api/v1.1/instances/oucloudusajul10/DJDBCS
...
* Closing connection #0
{"service_name":"DJDBCS","version":"12.1.0.2","status":"Terminating
","description":"DJDBCS","identity_domain":"oucloudusajul10","creat
ion_time":"Tue Jul 5 15:1:31 UTC 2016","last_modified_time":"Tue
Jul 5 15:1:31 UTC 2016","created_by": "dominique.jeunot@oracle.com"
,"sm_plugin_version":"16.3.1-132","service_uri":"https:\/\/dbaas.or
aclecloud.com:443\/paas\/service\/dbcs\/api\/v1.1\/instances\/oucl
oudusajul10\/DJDBCS"}
```

Oracle Internal & Oracle Academy Use Only

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

- Scale a database deployment
- Stop, Start, or Restart a database deployment or compute node
- View a database deployment

```
curl -v -X GET -u <your_username>:<your_password> -H
"X-ID-TENANT-NAME:<your_domain>"
https://dbaas.oraclecloud.com/paas/service/dbcs/api/v1.1/instances/<your_domain>/
<your_database_deployment>
```

- View a database deployment's compute nodes
- View all database deployments
- View the job status of an operation

Oracle Internal & Oracle Academy Use Only

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

Migrating to Oracle Database Cloud Service

Oracle Internal & Oracle Academy Use Only

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

Lesson 1 Overview of Migrating to Database Cloud Service

After completing this lesson, you should be able to:

- Choose a migration method based on characteristics of your on-premises database and other configuration considerations
- Describe the migration methods

Topics:

Oracle Internal & Oracle Academy Use Only

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

Choosing a Migration Method

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:

- Database version of your on-premises database
- For on-premises Oracle Database 12*c* Release 1 databases, the architecture of the database (multitenant or non-CDB)
- Endian format (byte ordering) of your on-premises database's host platform
- Database character set of your on-premises database and your Database Cloud Service database
- Database version of your Database Cloud Service database

Review *Choosing a Migration Method* for additional information.

Oracle Internal & Oracle Academy Use Only

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

Choosing a Migration Method: Considerations

Choosing a Method: Oracle Data Pump Considerations

Method	Pros	Cons
Conventional export/import	Can export from 11*g* and import into 12*c*	Limitations such as database character sets Slow
Transportable tablespace export/import	Fast	Limitations such as database character sets, encryption, partitioned tables, referential constraints, LOBs
Full transportable export/import	Fast	Limitations such as encryption, LONG or LONG RAW columns, database object defined in both an administrative and a user-defined tablespace Can only import into an Oracle Database 12*c* database

Additional information is provided in the *Using Oracle Data Pump* lesson.

Choosing a Method: Unplug/Plug and Remote Cloning Considerations

Method	Pros	Cons
Unplug/plug	Very fast Very easy	Unplug is only for Oracle Database 12*c* databases On-premises and Cloud databases must have: • The same endianness • The same set of installed database options • Compatible database character set
Remote cloning	Very fast Can keep the on-premises database	On-premises non-CDB must be 12.1.0.2 or later On-premises and Cloud databases must have: • The same endianness • The same set of installed database options • Compatible database character set

Additional information is provided in the *Using the Unplug/Plug Method* and *Using the Remote Cloning Method* lessons.

Choosing a Method: RMAN, SQL*Loader, and GoldenGate Considerations

Method	Pros	Cons

Oracle Internal & Oracle Academy Use Only

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

RMAN	Fast Can be used for different endian format platforms Between 11*g* and 12*c*: Faster than conventional export/import	RAW, LONG RAW, BLOB, ANYTYPE/ANYDATA/ANYDATASET and user-defined types or Oracle abstract types cannot be converted when conversion is necessary Transporting an entire PDB to a different platform requires the same endian format: transport each tablespace one by one
SQL*Loader	Supports loading of LOB data types, BLOB, CLOB, NCLOB, BFILE and RAW and LONG RAW	Loading data requires unloading the data from the on-premises database. Transferring the external file might be a long operation.
GoldenGate	Real-time data integration	Lengthy configuration

Additional information is provided in the *Using Recovery Manager (RMAN)* lesson.

Oracle Internal & Oracle Academy Use Only

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

Migrating On-Premises Database to Cloud Database

The following methods are applicable when migrating an Oracle Database 11*g* or 12*c* on-premises databases to an Oracle Database 11*g* or 12*c* database in the Cloud:

Method	On-Premises 11*g* database to Cloud 11*g* database	On-Premises 11*g* to Cloud 12*c* PDB	On-Premises 12*c* non-CDB to Cloud 12*c* PDB	On-Premises 12*c* PDBs to Cloud 12*c* PDB
Data Pump Conventional Export/Import	Y	Y	Y	Y
Data Pump Transportable Tablespace	Y	Y	Y	Y
Data Pump Full Transportable	N	Y	Y	Y
RMAN Transportable Tablespace with Data Pump	Y	Y	Y	Y
RMAN CONVERT Transportable Tablespace with Data Pump	Y	Y	Y	Y
RMAN Cross-Platform Transportable Tablespace Backup Sets	N	N	Y	Y
RMAN Cross-Platform Transportable PDB	N	N	N	Y
Unplugging/Plugging	N	N	Y	Y
Remote Cloning	N	N	Y	Y
SQL Developer and SQL*Loader to Migrate Selected Objects	N	N	Y	Y
SQL Developer and INSERT Statements to Migrate Selected Objects	N	N	Y	Y

See *Migrating from Oracle Database 11g to Oracle Database 11g in the Cloud* for additional information.
See *Migrating from Oracle Database 11g to Oracle Database 12c in the Cloud* for additional information.
See *Migrating from Oracle Database 12c Non-CDB to Oracle Database 12c in the Cloud* for additional information.
See *Migrating from Oracle Database 12c CDB to Oracle Database 12c in the Cloud* for additional information.

Oracle Internal & Oracle Academy Use Only

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

Lesson 2 Using Oracle Data Pump

After completing this lesson, you should be able to:

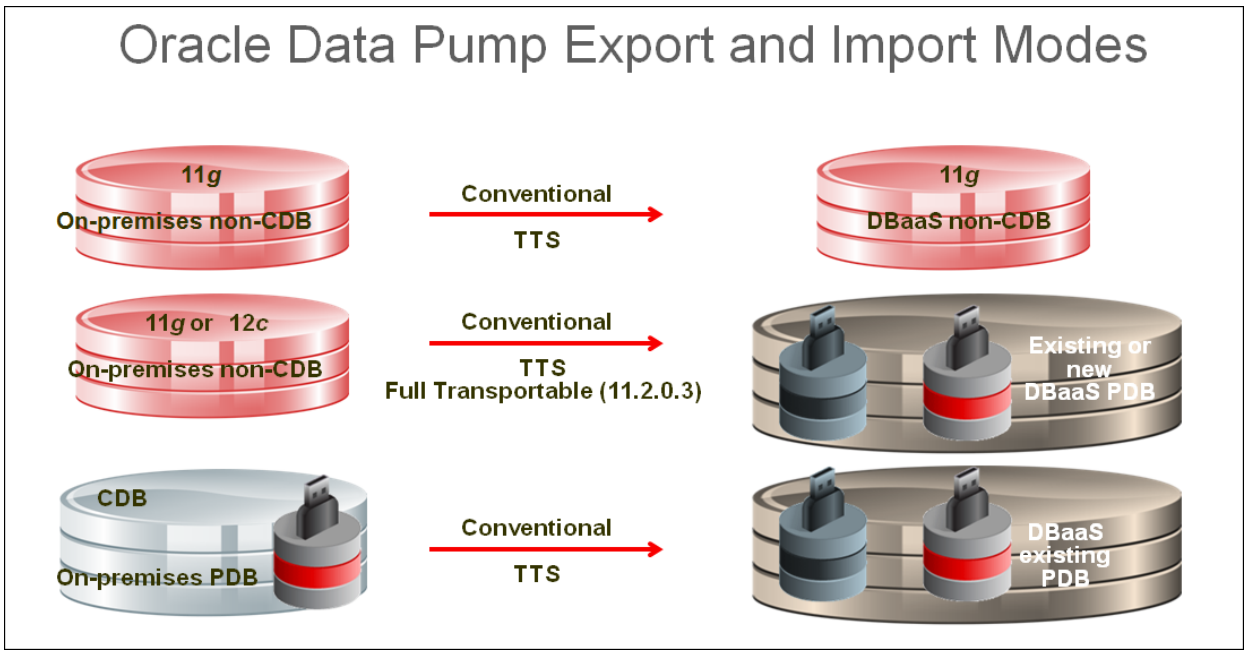
- Choose a Data Pump migration method based on your configuration
- Use the Data Pump conventional export/import method
- Use the Data Pump full transportable method
- Use the Data Pump transportable tablespace method

Topics:

Oracle Internal & Oracle Academy Use Only

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

Overview of Using Oracle Data Pump to Migrate



You can use Oracle Data Pump to move an on-premises 11*g* non-CDB database or an on-premises 12*c* non-CDB database or a PDB to an 11*g* database or a 12*c* database in Database Cloud Service.

There are three Oracle Data Pump modes that can be used to migrate an on-premises database:

- Conventional export/import
- Full transportable
- Transportable tablespace

Oracle Internal & Oracle Academy Use Only

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

Migration Compatability: Oracle Data Pump

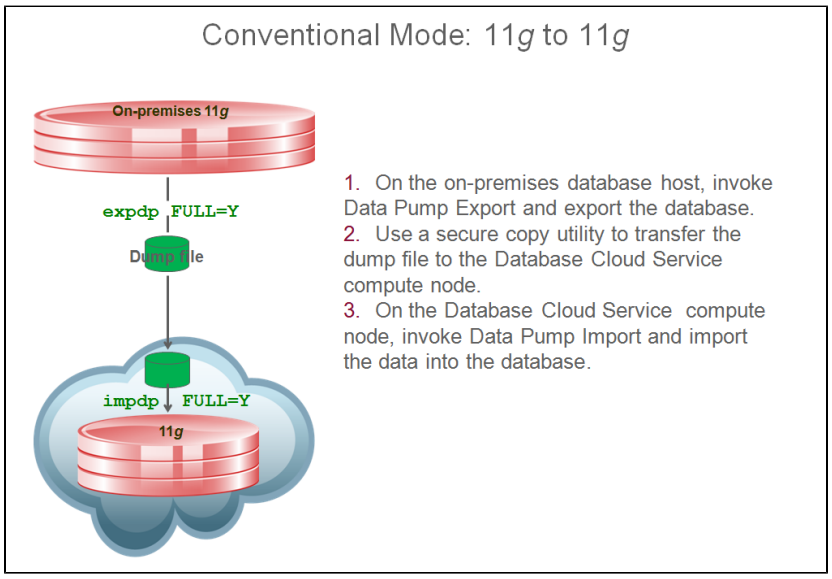
Method	On-premises Database Version	Database Cloud Service Database Version
Conventional export/import	11*g*	11*g*, 12*c* (new PDB or existing PDB)
	12*c* non-CDB or PDB	12*c* (new PDB or existing PDB)
Transportable tablespace	11*g*	11*g*, 12*c* (new PDB or existing PDB)
	12*c* non-CDB or PDB	12*c* (new PDB or existing PDB)
Full transportable	11.2.0.3 or later	12*c* (new PDB or existing PDB)
	12*c* non-CDB or PDB	12*c* (new PDB or existing PDB)

Oracle Internal & Oracle Academy Use Only

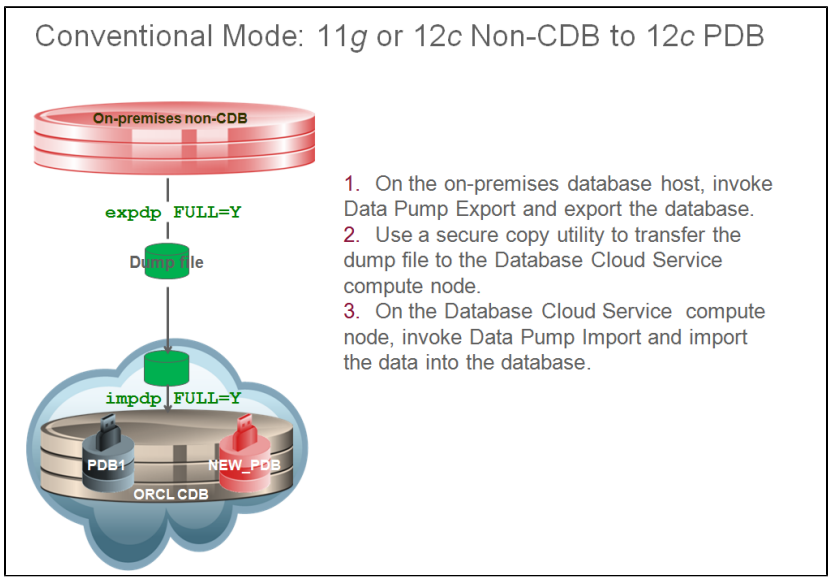
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Data Pump Conventional Export/Import

You can use conventional export/import to migrate an Oracle Database 11*g* database to an Oracle Database 11*g* database in the Cloud.



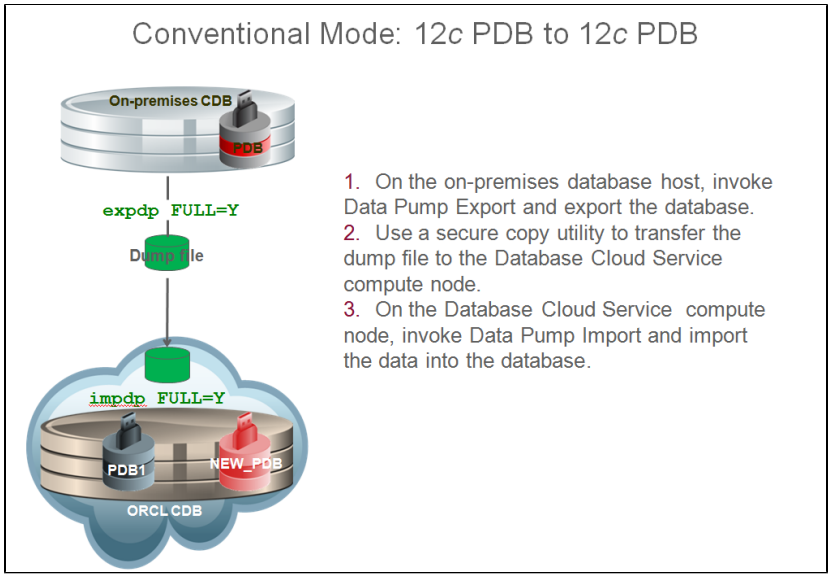
You can also use conventional export/import to migrate an Oracle Database 11*g* or Oracle Database 12*c* non-CDB database to an Oracle Database 12*c* PDB in the Cloud.



And you can use conventional export/import to migrate an Oracle Database 12*c* PDB to an Oracle Database 12*c* PDB in the Cloud.

Oracle Internal & Oracle Academy Use Only

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



The basic steps for all of these combinations are the same.

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

As a reminder, this method is applicable for the following migration release combinations:

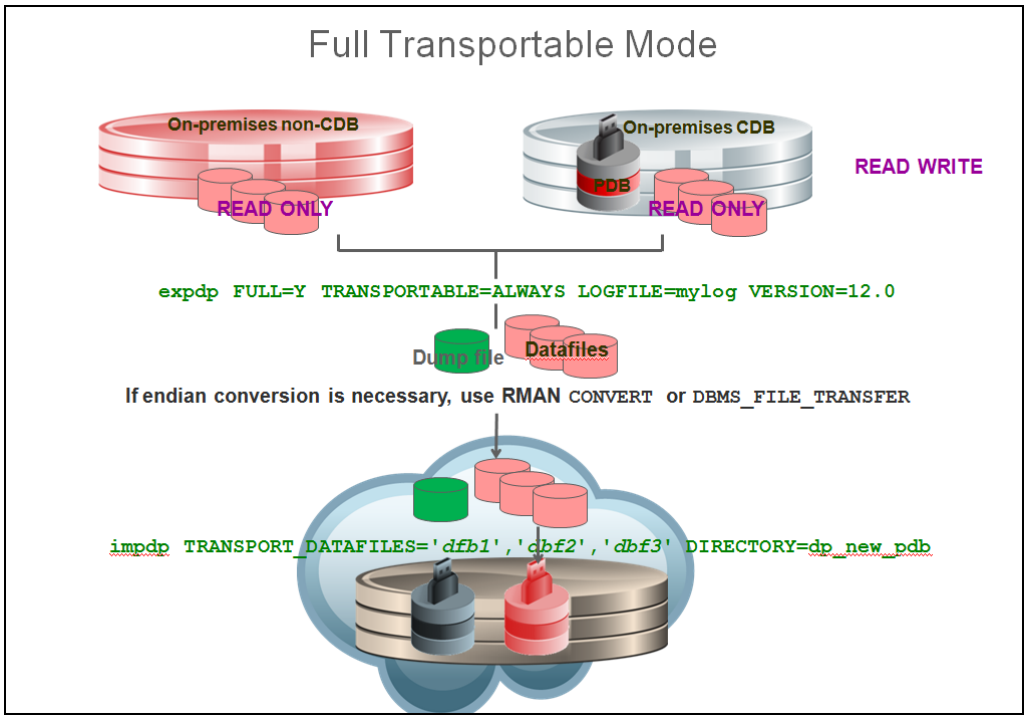
- From Oracle Database 11*g* to Oracle Database 11*g* in the Cloud
- From Oracle Database 11*g* to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

See Data Pump Conventional Export/Import for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

Data Pump Full Transportable



Perform these tasks to migrate an Oracle Database 11*g* on-premises database to the Oracle Database 12*c* database deployment using the Data Pump full transportable method:

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 Enterprise 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.

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 the Database Cloud Service database are compatible.

As a reminder, this method is applicable for the following migration release combinations:

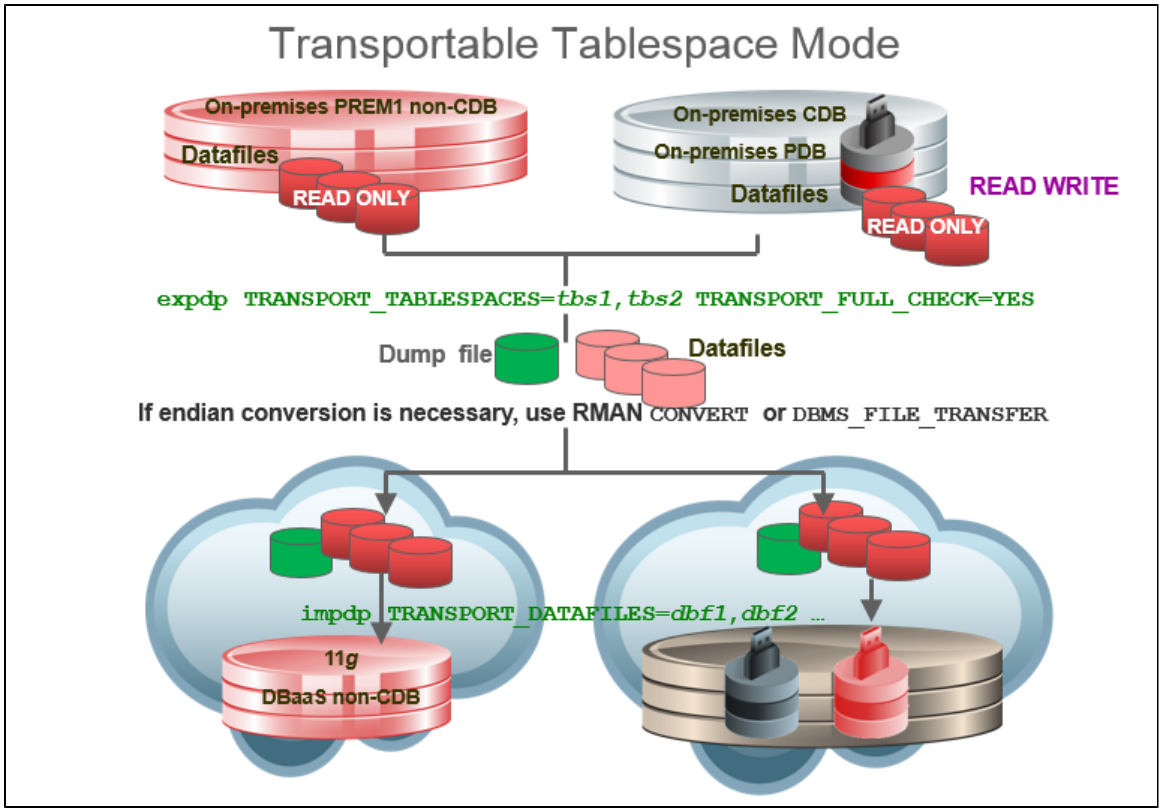
- From Oracle Database 11*g* to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

See Data Pump Full Transportable for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

Data Pump Transportable Tablespace



Perform these tasks to migrate an on-premises source database to the database deployment on Database Cloud Service using the Data Pump Transportable Tablespace method:

1. Export the tablespaces from the on-premises database:
    a. Set all tablespaces in the transportable set to read-only.
    b. If the on-premises database is a PDB, create a data pump directory object in the PDB pointing to the OS directory where the dump file will be generated.
    c. Use expdp with a userid granted the DATAPUMP_EXP_FULL_DATABASE privilege and use the PDB or non-CDB net service name to connect to the on-premises PDB or non-CDB.
2. Transfer the dump file and all data files of the transportable set from the on-premises server to the Oracle Database Cloud Service Virtual Machine. If endian conversion is necessary, use RMAN `CONVERT` or `DBMS_FILE_TRANSFER`.
3. Import the dump file into the DBaaS non-CDB or into the existing PDB of the CDB or into a new PDB:
    a. If the recipient is a PDB, create a data pump directory object in the existing PDB or in the new PDB pointing to the OS directory where the dump file and data files have been transferred.
    b. Use impdp with a userid granted the DATAPUMP_IMP_FULL_DATABASE privilege and use the net service name to connect to the DBaaS non-CDB or PDB.
4. If you plan to work with the on-premises database, connect to the on-premises database to set all tablespaces of the transportable set back to to read-write.

As a reminder, this method is applicable for the following migration release combinations:

- From Oracle Database 11*g* to Oracle Database 11*g* in the Cloud
- From Oracle Database 11*g* to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

See Data Pump Transportable Tablespace for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

Additional Considerations for Data Pump

Encryption

- Data Pump transportable tablespace cannot be used to transport encrypted tablespaces or tablespaces containing tables with encrypted columns.
- Data Pump full transportable cannot be used to transport an encrypted tablespace to a platform with different endianness.
- To transport an encrypted tablespace to a platform with the same endianness, set the `ENCRYPTION_PWD_PROMPT` export utility parameter to YES or use the `ENCRYPTION_PASSWORD` export utility parameter. During import, use the equivalent import utility parameter and set the value to the same password that was used for the export. Transport the wallet along with the data pump file.

BFILES and External Tables

- You must transport BFILEs to the 11*g* or 12*c* Database Cloud Service database.
- You must transport external table files to the 11*g* or *12c* Database Cloud Service database.

Oracle Internal & Oracle Academy Use Only

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

Lesson 3 Using the Unplug/Plug Method

After completing this lesson, you should be able to:

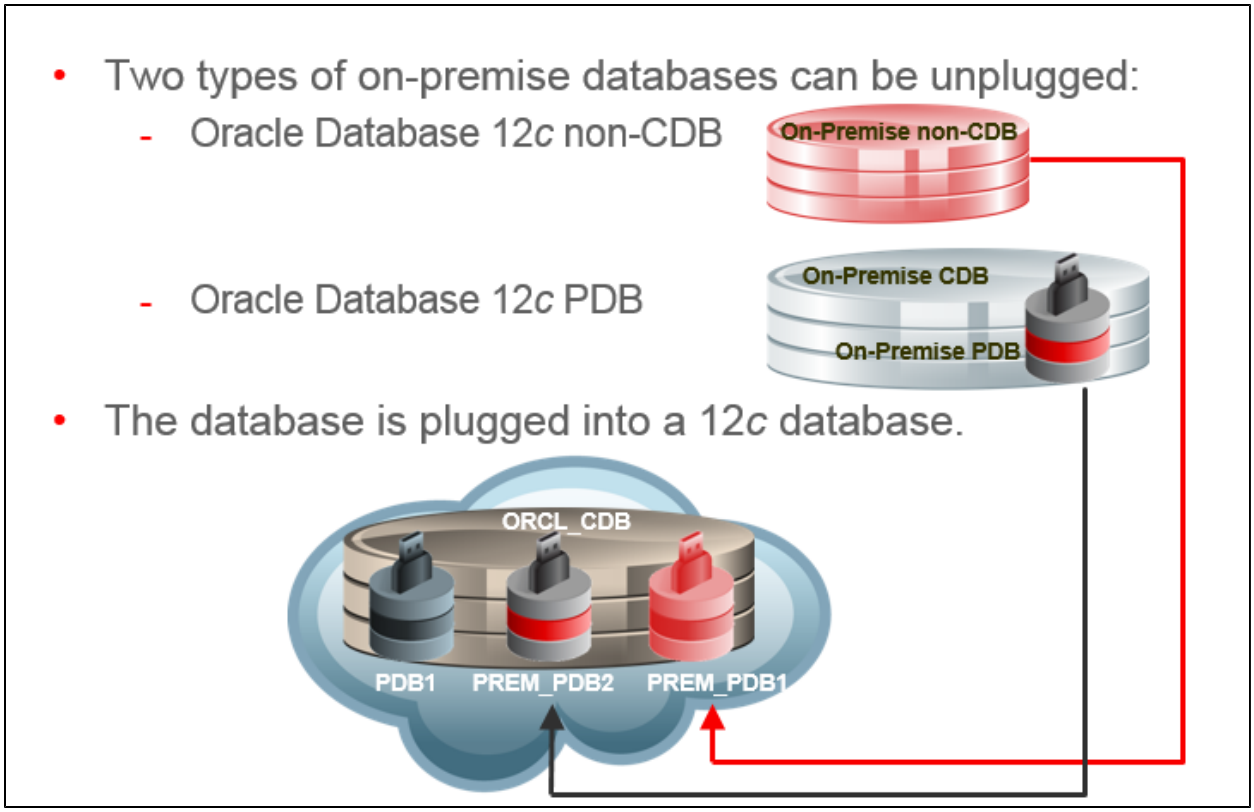
- Determine whether the unplug/plug method can be used in your configuration
- Use the unplug/plug method to migrate an on-premises database to Database Cloud Service

Topics:

Oracle Internal & Oracle Academy Use Only

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

Overview of Using the Unplug/Plug Method



You can use the unplug/plug method to migrate an on-premises 12*c* non-CDB database or a PDB to a 12*c* database in Database Cloud Service.

Oracle Internal & Oracle Academy Use Only

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

Migration Compatability: Unplug/Plug Method

You can unplug an on-premises 12*c* non-CDB database or PDB and plug it into a 12*c* CDB as a new PDB.

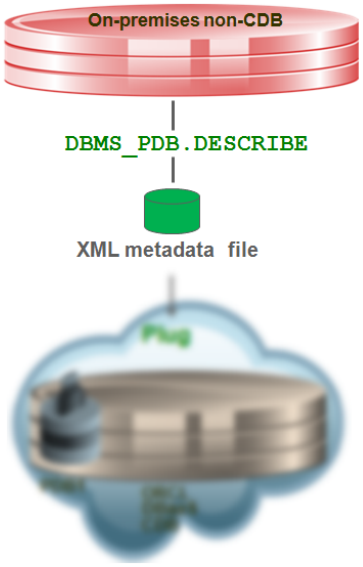
Method	On-premises Database Version	Database Cloud Service Database Version
Unplug/Plug	12*c* non-CDB or PDB	12*c* (new PDB)

Oracle Internal & Oracle Academy Use Only

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

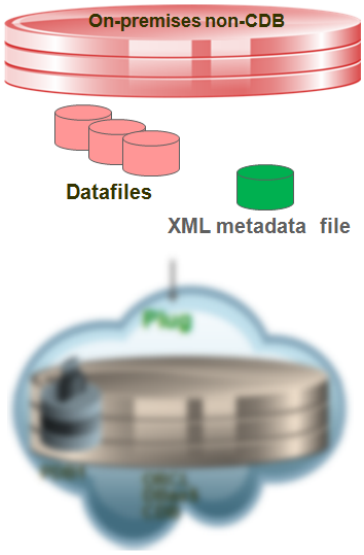
Unplugging/Plugging a Non-CDB

Unplugging an On-Premises Non-CDB



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.

Transferring Files to the Cloud

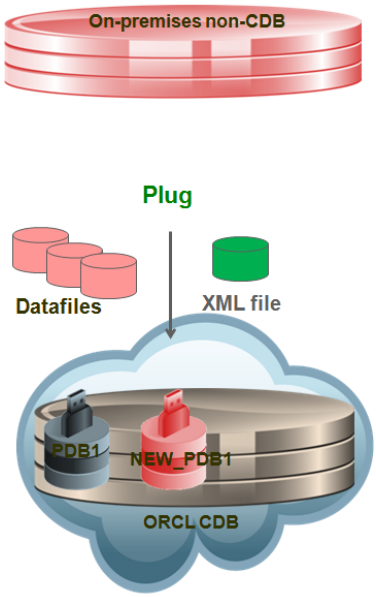


3. Use a secure copy utility to transfer the XML file and the datafiles to the Database Cloud Service compute node.

Oracle Internal & Oracle Academy Use Only

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

Plugging In an On-Premises Non-CDB



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 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.

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.

As a reminder, this method is applicable for the following migration release combinations:

- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

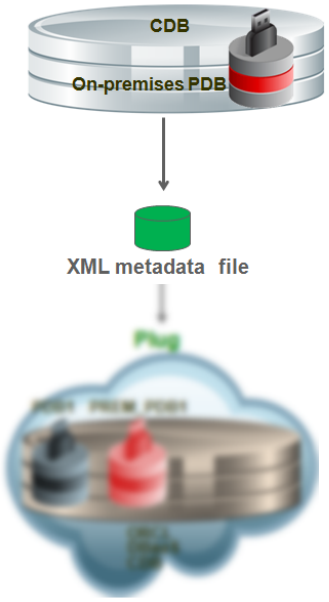
See Unplugging/Plugging Non-CDB for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

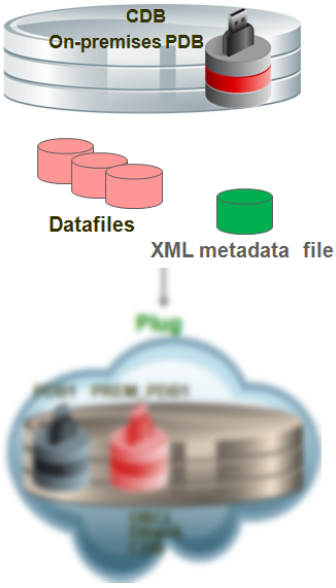
Unplugging/Plugging a PDB

Unplugging an On-Premises PDB



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 Cloud Service database.

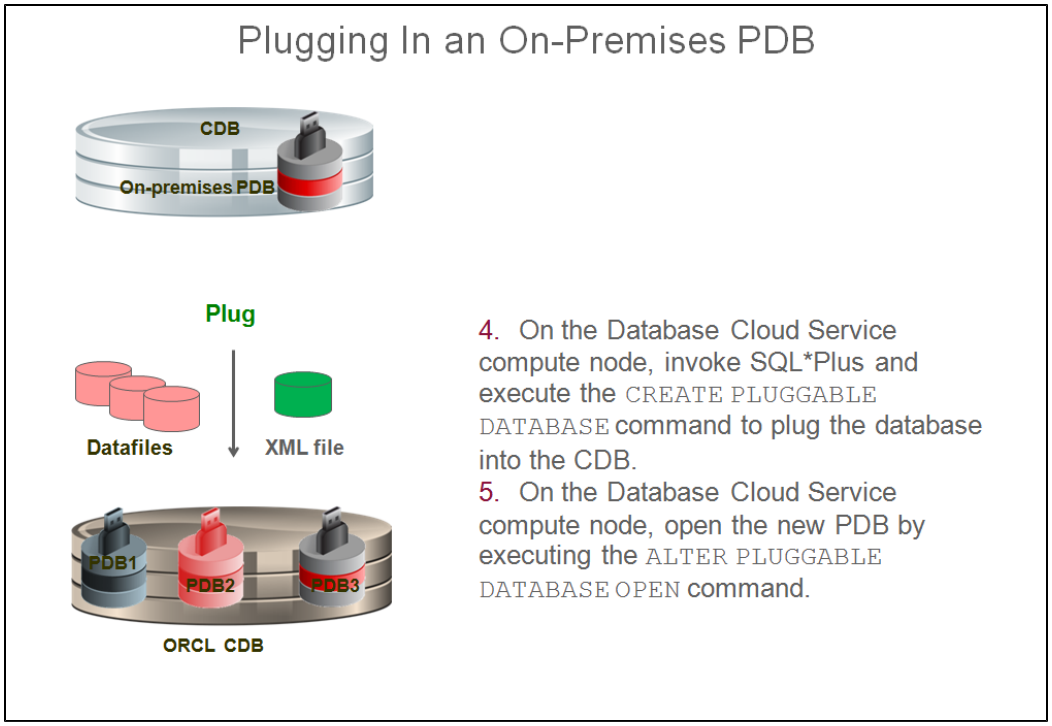
Transferring Files to the Cloud



3. Use a secure copy utility to transfer the XML file and the datafiles to the Database Cloud Service compute node.

Oracle Internal & Oracle Academy Use Only

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



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.

As a reminder, this method is applicable for the following migration release combinations:

- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud

See Unplugging/Plugging a PDB for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

Additional Considerations for the Unplug/Plug Method

COMPATIBLE Initialization Parameter Values

On-premises Database Release	COMPATIBLE Value	Cloud Database Release	COMPATIBLE Value	Upgrade
Oracle 12.1.0.2	12.1.0.2	Oracle 12.1.0.2	12.1.0.2	No upgrade required Reverse unplug works
Oracle 12.1.0.2	12.1.0	Oracle 12.1.0.2	12.1.0.2	No upgrade required `COMPATIBLE` is changed implicitly Reversing unplug is not possible
Oracle 12.1.0.2	12.1.0.2	Oracle 12.1.0.2	12.1.0	Plug does not work as `COMPATIBLE` cannot be lowered

To plug an unplugged on-premises PDB into a Database Cloud Service CDB, you need to be aware of the `COMPATIBLE` initialization parameter as this is a constraining factor for the unplug/plug method.

Use the `DBMS_PDB.CHECK_PLUG_COMPATIBILITY` function to determine whether the on-premises unplugged PDB is compatible with the Database Cloud Service CDB. An example of the CHECK_PLUG_COMPATIBILITY function:

```
-- ensure that the to-be-plugged-in PDB is compatible with the new host CDB. Execution of the PL/SQL block raises an error if it is not compatible.

set serveroutput on

DECLARE

compatible BOOLEAN := FALSE;

BEGIN

compatible := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(

pdb_descr_file => '/home/oracle/.../TDMSSTG.xml');

if compatible then

DBMS_OUTPUT.PUT_LINE('Is pluggable PDB1 compatible? YES');

else DBMS_OUTPUT.PUT_LINE('Is pluggable PDB1 compatible? NO');

end if;

END;

/
```

ENCRYPTION

Oracle Internal & Oracle Academy Use Only

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

If you are using encryption, follow this procedure:

1. Before unplugging the PDB, export the TDE master encryption key of the PDB by using the `ADMINISTER KEY MANAGEMENT EXPORT` command.
2. Unplug the PDB and plug the PDB.
3. If there is no keystore in the CDB, create it.
4. Reimport the TDE master encryption key by using the `ADMINISTER KEY MANAGEMENT IMPORT` command.

Oracle Internal & Oracle Academy Use Only

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

Lesson 4 Using the Remote Cloning Method

After completing this lesson, you should be able to:

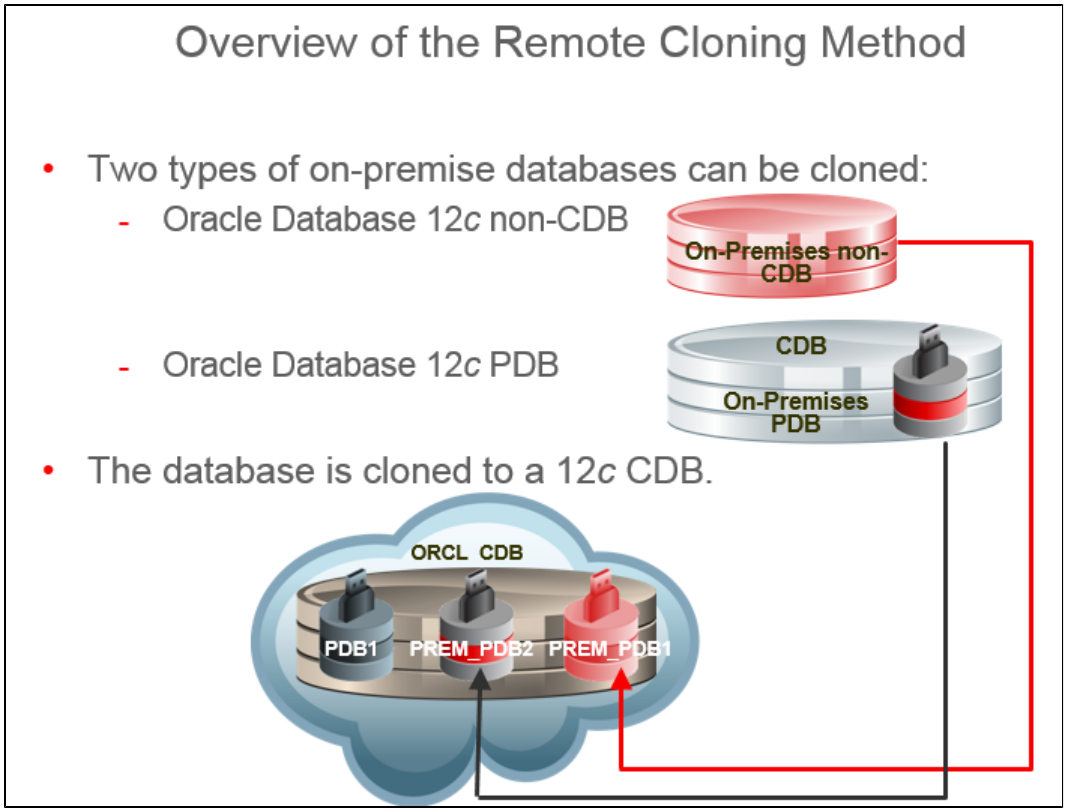
- Determine whether the remote cloning method can be used in your configuration
- Clone an on-premises non-CDB into a Database Cloud Service CDB
- Clone an on-premises PDB into a Database Cloud Service CDB

Topics:

Oracle Internal & Oracle Academy Use Only

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

Overview of Using the Remote Cloning Method



You can use the remote cloning method to migrate an on-premises 12*c* non-CDB database or a PDB to a 12*c* database in Database Cloud Service.

Oracle Internal & Oracle Academy Use Only

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

Migration Compatability: Remote Cloning Method

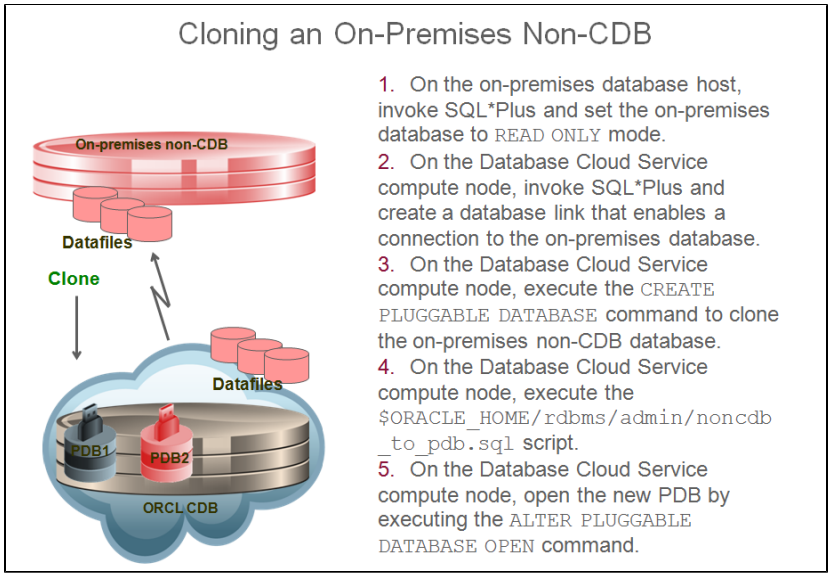
You can clone an on-premises 12*c* non-CDB database or PDB to a 12*c* CDB as a new PDB. This method uses a database link to connect from the target 12*c* CDB to the on-premises database.

Method	On-premises Database Version	Database Cloud Service Database Version
Remote Cloning	12*c* non-CDB or PDB	12*c* (new PDB)

Oracle Internal & Oracle Academy Use Only

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

Remote Cloning a 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.

As a reminder, this method is applicable for the following migration release combinations:

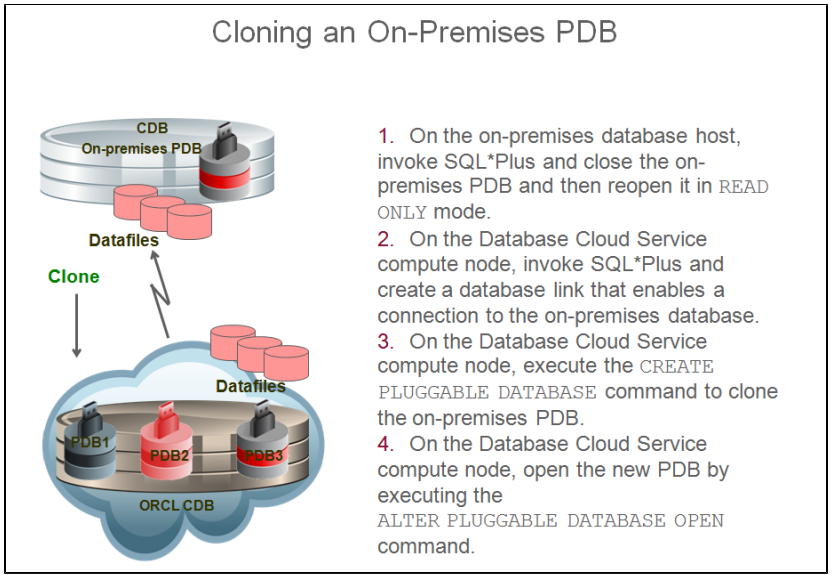
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

See Remote Cloning Non-CDB for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

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.

As a reminder, this method is applicable for the following migration release combinations:

- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud

See Remote Cloning a PDB for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

Lesson 5 Using Recovery Manager (RMAN)

After completing this lesson, you should be able to:

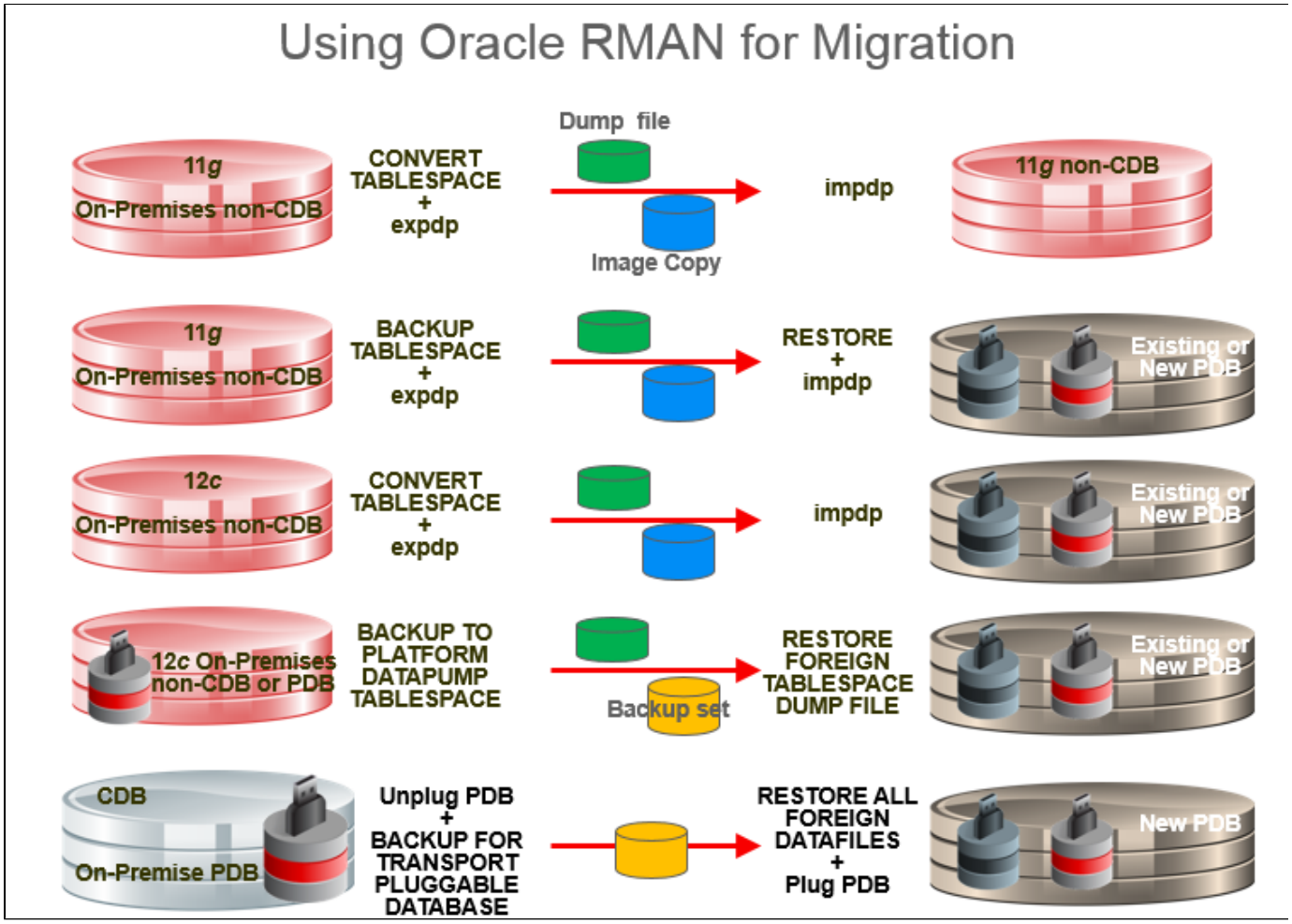
- Choose an RMAN migration method based on your configuration
- Use any of the following methods to migrate your on-premises database to Oracle Database Cloud Service:
  - RMAN CONVERT Transportable Tablespace with Data Pump
  - RMAN Cross-Platform Transportable PDB
  - RMAN Cross-Platform Transportable Tablespace Backup Sets
  - RMAN Transportable Tablespace with Data Pump

Topics:

Oracle Internal & Oracle Academy Use Only

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

Overview of Using Recovery Manager



You can use Recovery Manager (RMAN) to migrate an on-premises database to a database in Database Cloud Service.

There are four primary methods to migrate an on-premises database to a database in the Cloud:

- RMAN CONVERT Transportable Tablespace with Data Pump
- RMAN Cross-Platform Transportable PDB
- RMAN Cross-Platform Transportable Tablespace Backup Sets
- RMAN Transportable Tablespace with Data Pump

Oracle Internal & Oracle Academy Use Only

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

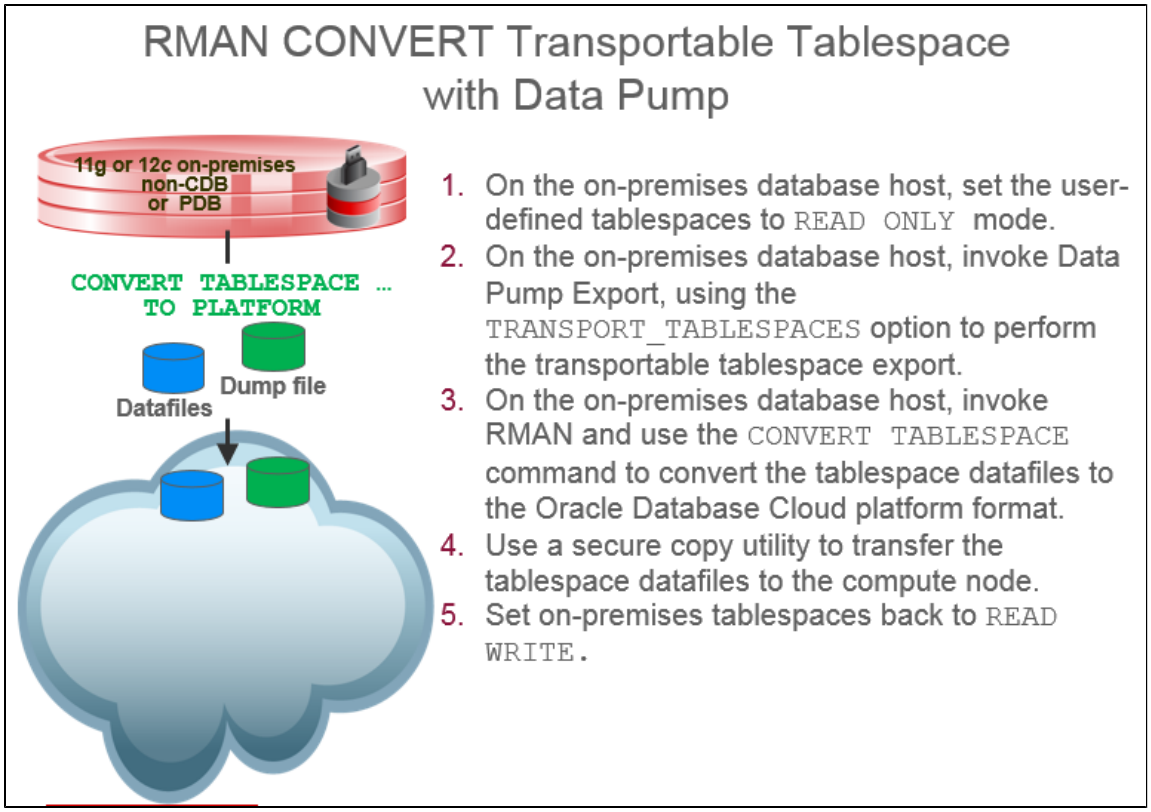
Migration Compatibility: RMAN

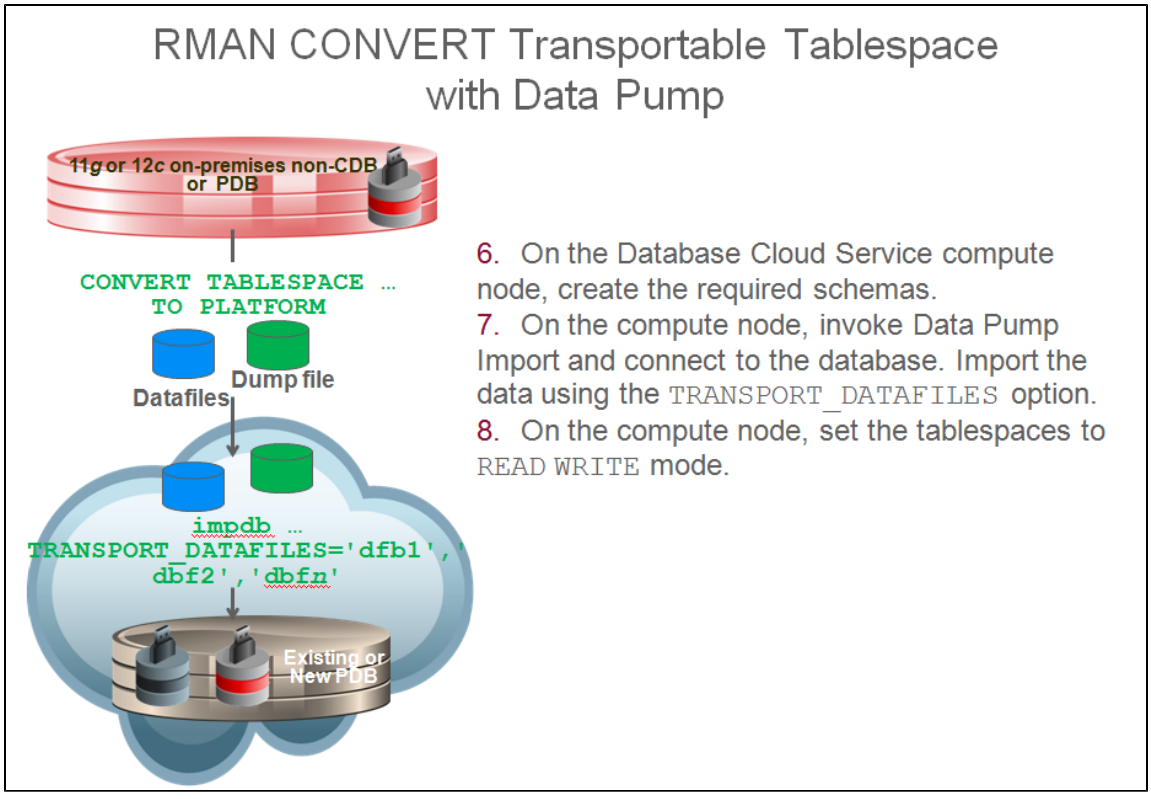
Method	On-Premises Database Version and Commands	Database Cloud Service Database Version and Commands
Transportable tablespace (CONVERT)	11*g* tablespace backup with conversion: RMAN CONVERT + export	11*g* tablespace import
Transportable tablespace	11*g* backup set: RMAN BACKUP + export	12*c* RESTORE + import
Cross-platform transportable non-CDB tablespace: Image copies	12*c* non-CDB or PDB tablespace: RMAN CONVERT + export	12*c* import
Cross-platform transportable non-CDB tablespace: Backup sets	12*c* non-CDB tablespace: RMAN BACKUP TO PLATFORM DATAPUMP	12*c* RESTORE FOREIGN TABLESPACE DUMP FILE
Cross-platform transportable PDB	12*c* PDB: BACKUP FOR TRANSPORT PLUGGABLE DATABASE	12*c* PDB: RESTORE PLUGGABLE DATABASE

Oracle Internal & Oracle Academy Use Only

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

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

Oracle Internal & Oracle Academy Use Only

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

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 in little-endian format.

As a reminder, this method is applicable for the following migration release combinations:

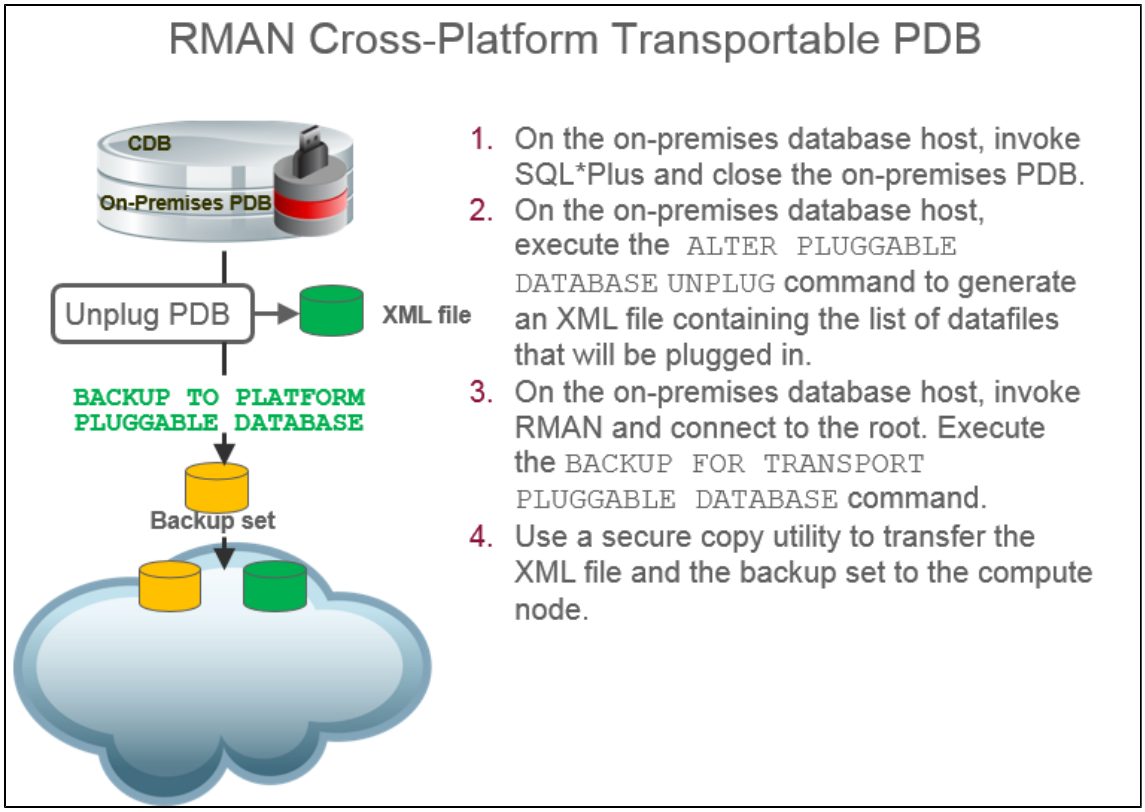
- From Oracle Database 11*g* to Oracle Database 11*g* in the Cloud
- From Oracle Database 11*g* to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

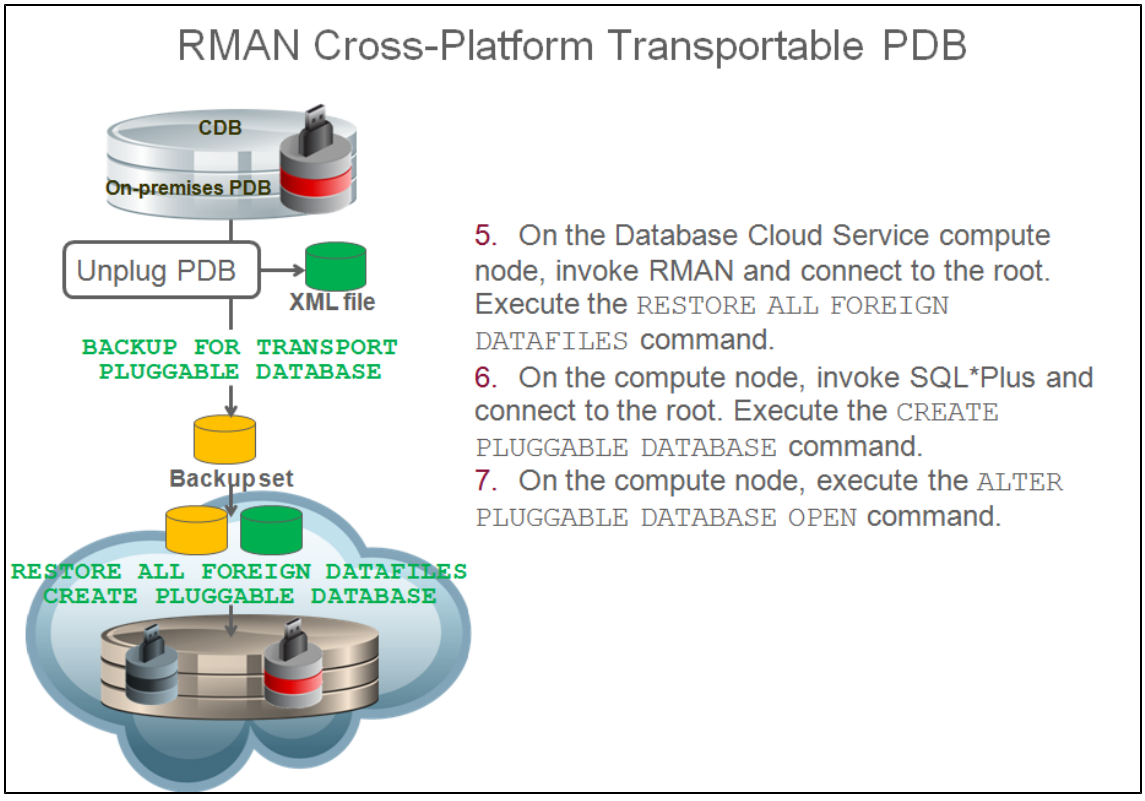
See RMAN CONVERT Transportable Tablespace with Data Pump for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

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 Database Cloud Service database are compatible.

Oracle Internal & Oracle Academy Use Only

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

As a reminder, this method is applicable for the following migration release combinations:

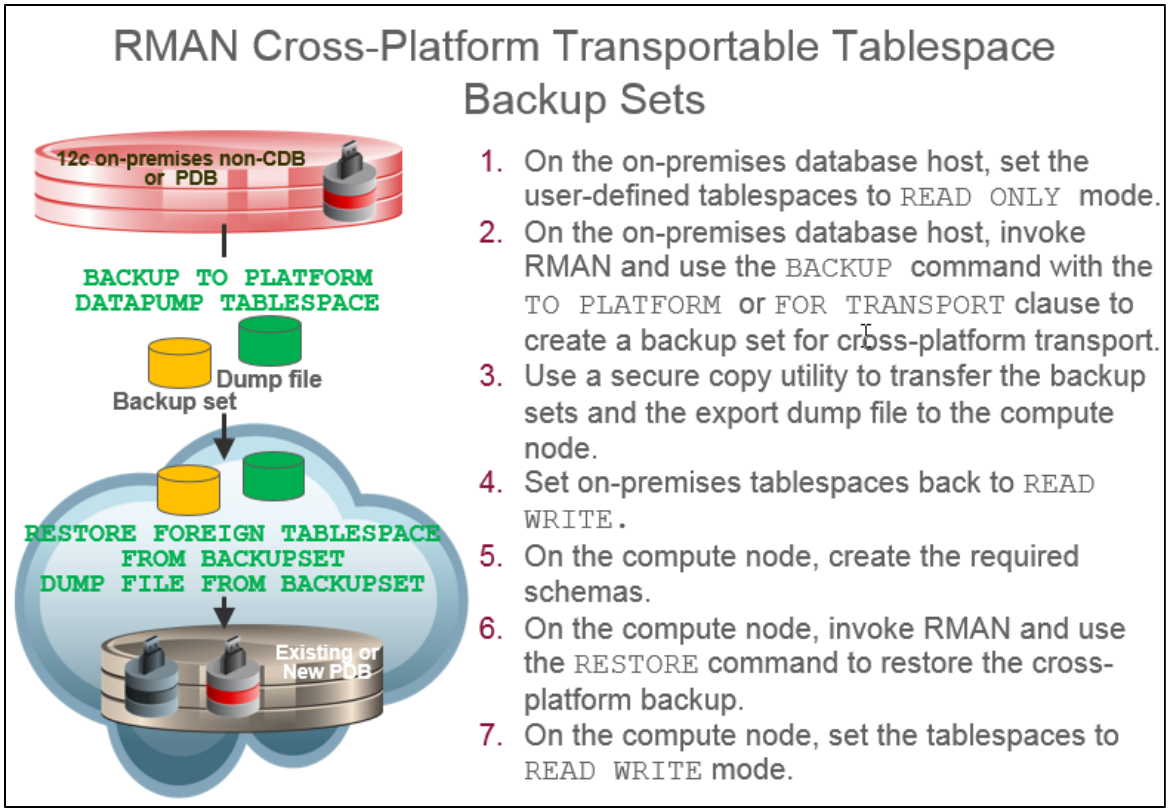
- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud

See RMAN Cross-Platform Transportable PDB for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

RMAN Cross-Platform Transportable Tablespace Backup Sets



You can use this method only if the database character sets of your on-premises database and Database Cloud Service database are compatible.

As a reminder, this method is applicable for the following migration release combinations:

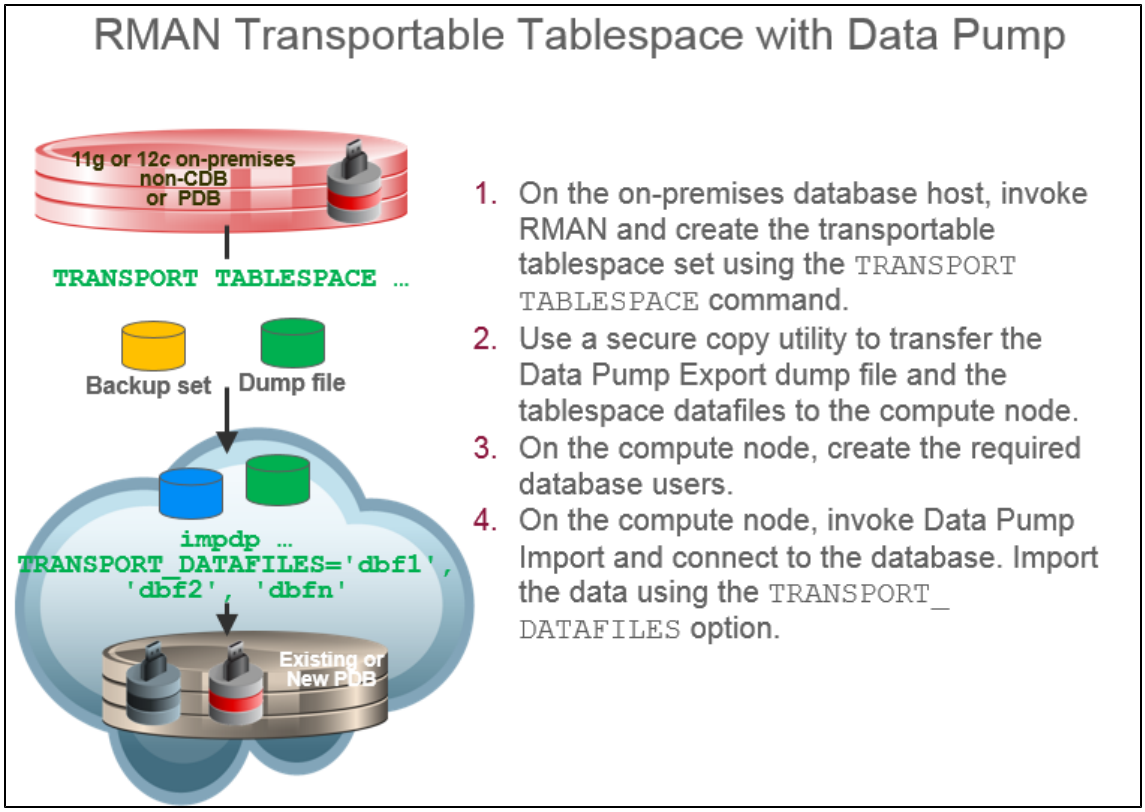
- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

See RMAN Cross-Platform Transportable Tablespace Backup Sets for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

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 Database Cloud Service database are compatible.

As a reminder, this method is applicable for the following migration release combinations:

- From Oracle Database 11*g* to Oracle Database 11*g* in the Cloud
- From Oracle Database 11*g* to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

See RMAN Transportable Tablespace with Data Pump for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

Lesson 6 Using SQL Developer

After completing this lesson, you should be able to:

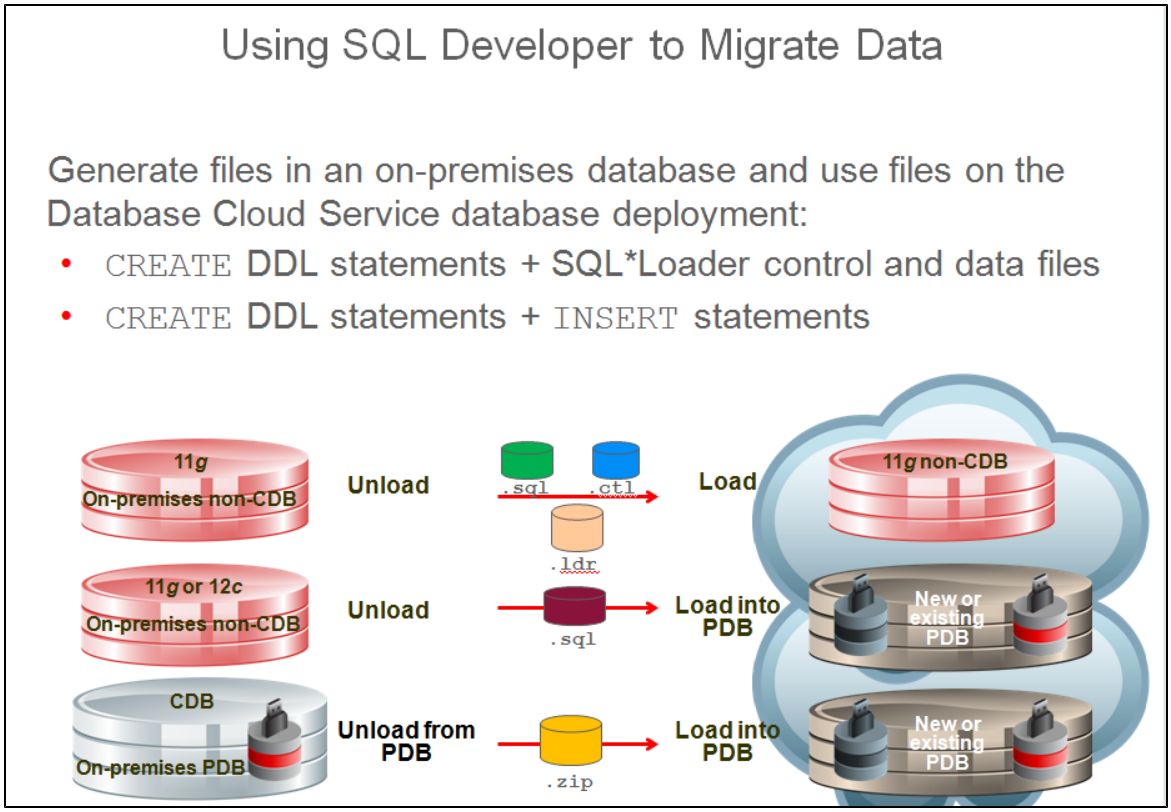
- Determine whether the SQL Developer methods can be used in your configuration
- Use SQL Developer to create a cart of selected objects from your on-premises database and then use generated SQL INSERT statements to load the data into your cloud database
- Use SQL Developer to create a cart of selected objects from your on-premises database and then use SQL*Loader to load the data into your cloud database

Topics:

Oracle Internal & Oracle Academy Use Only

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

Overview of Using SQL Developer



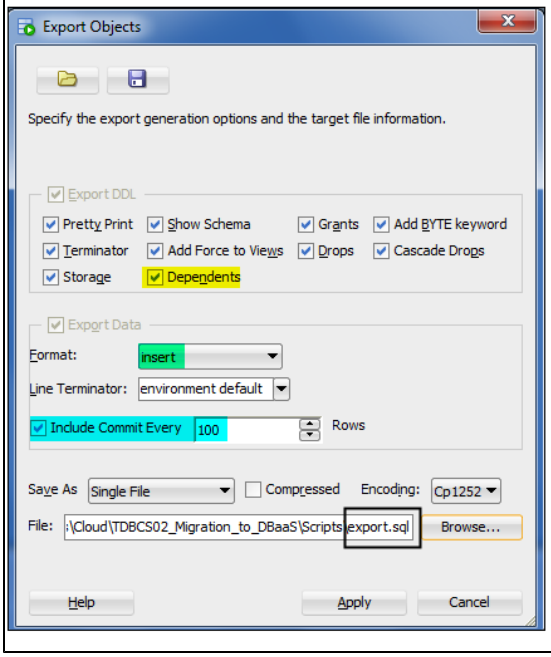
You can use SQL Developer to migrate an on-premises database to a database in Database Cloud Service.

Oracle Internal & Oracle Academy Use Only

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

SQL Developer and INSERT Statements to Migrate Selected Objects

Using SQL Developer and INSERT Statements to Migrate Selected Objects



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 database on Database Cloud Service and execute the generated script to create the database objects.
4. In SQL Developer, open a connection to the database on Database Cloud Service and run the generated script to create the objects and load the data.

As a reminder, this method is applicable for the following migration release combinations:

- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

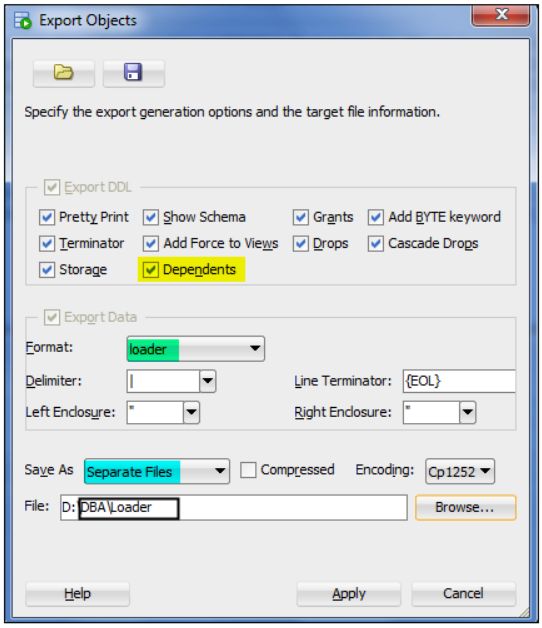
See SQL Developer and INSERT Statements to Migrate Selected Objects for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

SQL Developer and SQL*Loader to Migrate Selected Objects

Using SQL Developer and SQL*Loader to Migrate Selected Objects



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 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.

As a reminder, this method is applicable for the following migration release combinations:

- From Oracle Database 12*c* CDB to Oracle Database 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 12*c* in the Cloud

See SQL Developer and SQL*Loader to Migrate Selected Objects for detail and examples.

Oracle Internal & Oracle Academy Use Only

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

Performance

Oracle Internal & Oracle Academy Use Only

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

Lesson 1 Overview of DBCS Performance Management

After completing this lesson, you should be able to:

- Describe Performance Management Methodology
- Understanding what may be tuned
- Describe the steps of the tuning methodology

Topics:

- Performance Management in the Database Cloud Environment
- Performance Monitoring and Tuning
- Tuning Methodology
- Effective Tuning Goals
- General Tuning Session
- What Can be Tuned in a DBCS Environment?

Oracle Internal & Oracle Academy Use Only

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

Performance Management in the Database Cloud Environment

- Subscription cost is related to the resource usage
- Database instance is configured per best practices
- Resource usage is dependent on the application

Performance Management has two main components: performance monitoring, and performance tuning. In the Database Cloud Service (DBCS) several areas of tuning are managed in the cloud environment removing the need for tuning. Other areas are tuned per best practice and would seldom need any intervention by an administrator. In this course we will focus on performance management areas, that are more likely occur, and available to administrators to modify.

In the database deployment, the cost in general is determined by resource usage. So it is in your best interest as a subscriber to reduce resource while maintaining the required performance levels.

Oracle Internal & Oracle Academy Use Only

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

Performance Monitoring and Tuning

Can be either:

- Proactive monitoring/tuning: Examining performance statistics at a regular interval to identify whether the system behavior and resource usage has changed

  Proactive monitoring (or proactive tuning) usually occurs at a regularly scheduled interval and encompasses examining several performance statistics to identify whether the system behavior and resource usage has changed. Monitoring does not usually result in system configuration changes, unless the monitoring exposes a serious or developing problem. Consider monitoring as part of capacity planning, where resource consumption is examined to see changes in the way the application is being used, and the way the application is using the database and host resources.

- Reactive tuning: Identifying an overused resource in the system and determining potential fixes

  When tuning is left until the database is in production, it often becomes a reactive process to identify and fix the most serious bottleneck. The goal of tuning is usually to improve the use of a particular resource. In general, performance problems are a result of the overuse of a particular resource which becomes a bottleneck in the system.

Oracle Internal & Oracle Academy Use Only

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

Tuning Methodology

Oracle Corporation has developed a tuning methodology based on years of experience. The methodology presented in this course is also presented in the Oracle Database Performance Tuning Guide. This methodology is applied independently of the tools that you use. The Automatic Database Diagnostic Monitor (ADDM) tool follows this methodology automatically. The basic steps are the following:

- Identify the scope of the problem (OS, database, and so on).
  Check the OS statistics and the general machine health before tuning the instance to be sure that the problem is in the database instance.
- Tune the following from the top down:
  - Tune the design before tuning the application code.

    Start with the design, then the application, and then the instance. As an example, try to eliminate the full table scans causing the I/O contention before tuning the tablespace layout on disk. The design should use appropriate data structures for the application and load characteristics. For example: Applications should avoid processes that require serialization through a single resource. A simple example is a single check number generator used by multiple processes.
  - Tune the code before tuning the instance

    Tuning at the instance level is often limited by design and application choices. With existing applications, this step is often not available, since the design and code are not modifiable.
- Tune the area with the greatest potential benefit.

The tuning methodology presented in this course is simple. Identify the biggest bottleneck and tune it. Repeat. The Oracle tuning tools use DB Time to identify problem areas. All tools have some way to identify the SQL statements, resource contention, or services that are taking the most time. Oracle Database 12*c* provides a time model and metrics to automate the process of identifying bottlenecks. The steps are:

- Identify the performance problem (Use available tools ADDM, AWR, DBaaS monitor, SQL Developer).
- Analyze the problem, looking for skewed and tunable components.
- Use appropriate tools to tune the components implicated.
- Stop tuning when the goal is met. This implies that you have set tuning goals.

This is a general approach to tuning any database instance and may require multiple passes.

Ideally someone with database tuning experience will be involved in the design and development from the beginning. This individual, for example, could suggest indexes to limit full table scans on frequently accessed tables.

From a practical perspective, tuning during the design and development phases of a project tends to be more top down. The tuning efforts during testing and production phases are often reactive and bottom up. In all phases, tuning depends on actual test cases because theoretical tuning does not know all the variables that can be present. After a problem area is suspected, or discovered, a test case is created and the area tuned. Tune the area that has the greatest potential benefit. Reduce the longest waits and the largest service times.

As you may notice, the techniques are very much the same, no matter what life cycle phase. A test case or actual application is run, the available diagnostic tools are applied, a solution is proposed and tested.

Oracle Internal & Oracle Academy Use Only

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

Effective Tuning Goals

Effective tuning goals are:

- Specific
- Measurable
- Achievable
- Cost effective

The tuning goal is the elimination of a defined problem. The goal may be based on an SLA or a problem. For example: The SLA says that user response time to a particular request must be no more than 30 seconds. The problem is that the average response time is 25 seconds and increasing.
A tuning goal could be: user response time to a particular request is 20 seconds.

Both tuning goals and SLAs must have three characteristics to be effective. They must be:

- Specific
- Measurable
- Achievable

“Make the instance run as fast as possible” is not specific. A specific goal would be “The month end report suite must complete in less than 4 hours.”

In addition, a goal must be cost effective. Tuning simply for the sake of tuning, or for elegance is not cost effective. There must be a needed benefit.

A measurable goal has objective quantities that can be measured. There is no doubt whether the goal is being met when it is measurable. A goal that is specific is easily made measurable as well. The goal of “user response time to a request is 10 seconds” is easily stated, but is this for all user requests? Is it the average response time? How do you measure average response time? Having specific definitions for the words of your goal is essential. By restating the goal as “User response time to a particular request is 20 seconds or less,” you can objectively determine when the goal has been met.

Achievable goals are possible and within the control of the persons responsible for tuning.

The following are examples of unachievable goals for a typical DBA:

- When the goal is to tune the instance to create a high-performance application, but you are not allowed to change the SQL or the data structures, there is a limited amount of tuning that is possible.
- When the goal is to have a response time of 1 second, but the network latency between the server and the client is 2 seconds. Without a change to the network, a response time of 1 second is impossible.

Even these situations are not impossible to change in an absolute sense, but the DBA always has business constraints that limit the amount of money and resources that can be applied to the solution. So every goal should consider the cost to benefit. A goal that costs a great deal but solves a problem that has a marginal benefit, is best left undone.

You should always establish measurable tuning goals. Without a tuning goal, it is difficult to determine when you have performed enough tuning.

Oracle Internal & Oracle Academy Use Only

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

General Tuning Session

When tuning, you focus on specific areas that offer the greatest return for your tuning effort. The steps are generic and apply to any performance monitoring tool. The recommended tuning methodology is as follows:

1. **Define the problem and state the goal**: This is the analysis step. The Oracle performance and diagnostic tools use a time model that can be used to quickly identify the problem areas. The information source could be users, database statistics, metrics, or database diagnostic reports. Be sure to collect accurate and factual data that corresponds to the problem. State the problem in terms that are measurable and directly related to the database operations. As an example, if the run time on the "XYZ" report is two times the baseline, the goal becomes: Make the run time on the "XYZ" report equal to or less than the baseline.
2. **Collect current performance statistics:** Examine the host system and the database statistics. Collect a full set of operating system and database statistics, and compare these with your baseline statistics. The baseline statistics are a set of statistics that are taken when the instance is running acceptably. Examine the differences to determine what has changed on the system. Did the "XYZ" report change? Did the data change? Is the session producing the report waiting on something?
3. **Consider common performance errors:** From your list of differences in the collected statistics, make a comparison with common performance errors. Determine whether one of these errors has occurred on your system.
4. **Build a trial solution:** Include a conceptual model in your solution. The purpose of this model is to assist you with the overall picture of the database. You are looking for answers to the following questions:

   Why is the performance degraded?
   How can you resolve the problem to meet your goal?

5. **Implement the solution and measure the change:** After you have developed the trial solution, make the appropriate change. Make only one change at a time. If you make multiple changes at the same time, you will not know which change is effective. If the changes do not solve the problem, you would not know whether some changes helped and others hindered. Collect statistics to measure the change.
6. **"Did the solution meet the goal?"** Compare the current and the baseline sets.

- **'No' :** If you determine that more tuning is required, return to step 3 and repeat the process.
- **'Yes':** If your solution meets the goal, make the current set of statistics the new baseline set. You met your goal! Stop tuning!

Oracle Internal & Oracle Academy Use Only

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

What Can be Tuned in a DBCS Environment?

One of the benefits of the DBCS environment is that the amount of tuning required is minimal, but there will be uses that do not fall in the median group. For instance: Applications that require more memory, either for Process memory, PGA, or Shared Global Memory, SGA; Compute intensive applications that require more CPU; or applications that require low latency disk access. The DBCS instances can be scaled up, for CPU or Memory, and storage can be added to meet these kind of needs. These resources have an associated cost.
So before you pay for more of whatever resource you believe is required, tune the following areas:

- Application Design

  A badly designed application is very difficult to tune without changing the design and code. It is good practice to have a DBA experienced in Performance Tuning as part of your design team to avoid common design flaws and assist in developing test cases for prototypes.

- SQL Statements

  As a customer you do not want to pay for unnecessary resource usage due to poorly performing SQL. So tuning the application SQL and the SQL optimization to use the available resources efficiently can reduce the overall cost of using a DBCS instance.

- Database Instance Initialization Parameters

  The initialization parameters of the database instance affect: memory sizing (SGA and PGA), SQL optimization, SQL Plan use, resource plans, background process behavior and more. These parameters have been pre-tuned for most database deployments, but could be helpful in some deployments.

- Memory Sizing

  In the initial database deployment an initial size is preset and the allocation between SGA and PGA is automatically adjusted as workload changes. A few uses of the database deployment may benefit by controlling the memory allocation more closely.

- Redo and Archive Logging

  The amount of redo generated can affect performance. Configuring larger or more redo log files may help when there is a large volume of redo being generated

Each tuning tool has its own limitations.
For Example: EM Database Express cannot perform actions that will require a restart of the database instance. DBaaS Monitor can only change initialization parameters at the CDB level, and they will affect all the PDBs in that database instance. SQL Developer has a full range of capabilities but some changes require a detailed knowledge of SQL, or PL/SQL to implement, and in SQL*Plus any change is possible if you master the command line interface.

Common instance tuning areas:

- Memory sizing – PGA and SGA
- Redo and archive tuning
- Initialization Parameters
- Database structure – placement of database files on file systems and physical disks is NOT tunable in DBaaS in the default storage, as you do not know how the virtual disks are mapped to physical hardware. Placing specific database files on storage volumes with low latency is possible.
- Resource Management – specific applications and PDBs may be given priority to certain resources such as CPU with Resource Manager plans.

Oracle Internal & Oracle Academy Use Only

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

SQL Tuning is often the most cost effective tuning method, so identification of candidate statements is important.

- SQL Tuning Advisor will identify and make recommendations for high cost SQL.
- ADDM will identify SQL statements that use the most resources.
- SQL Monitor will identify long running SQL statements, or operations.

Once identified these statements can be tuned

- SQL Tuning Advisor can recommend and implement SQL Profiles, and make other recommendations.
- SQL*Developer may be used revise and test SQL execution plans for individual statements.

Oracle Internal & Oracle Academy Use Only

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

Lesson 2 Tuning Performance Issues

After completing this lesson, you should be able to:

- Use available tools
- Identify Performance Issues
- Diagnose Performance Issues

Note: The Performance Tuning Guide, and the SQL Tuning Guide provide a comprehensive methodology for tuning instance and SQL performance issues. The scope of this lesson includes the differences in using the tools with DBCS instances, and tools that are unique to DBCS.

Topics:

Oracle Internal & Oracle Academy Use Only

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

Available Tools

For the DBCS deployment The same tools are available as in the on-premises database environment:

- SQL*Plus
- SQL Developer
- EM Express

And an additional tool: DBAAS Monitor

In previous lessons you have seen how to access the database deployment using these tools. To review:

SQL*Plus is a command-line tool you can use in an SSH terminal session connected to the Compute node for the database deployment. You can also use it from any SQL client machine if the SQL*Net port (1521) has been opened.

SQL Developer is a GUI tool that provides all the functionality of SQL*Plus and adds a high degree of ease of use. SQL Developer in recent releases (4.1.x) has added several cloud-specific features. There are three ways to access your deployment.

- Open SQL*Net port(1521) on the compute node.
- Create an SSH tunnel for port 1521.
- Use the SQL Developer built in ability to create an SSH tunnel.

EM Express in its current version of EM express (12.1.0), the CDB, and each PDB that you wish to access, use different ports. Again you have multiple options to access the database deployment:

- Create an SSH tunnel for the EM express port of interest (default is 5500 for the CDB, and the port for PDBs is configured by the DBA)
- Create a security application and security rule in the Compute Service to open the ports you desire to access.

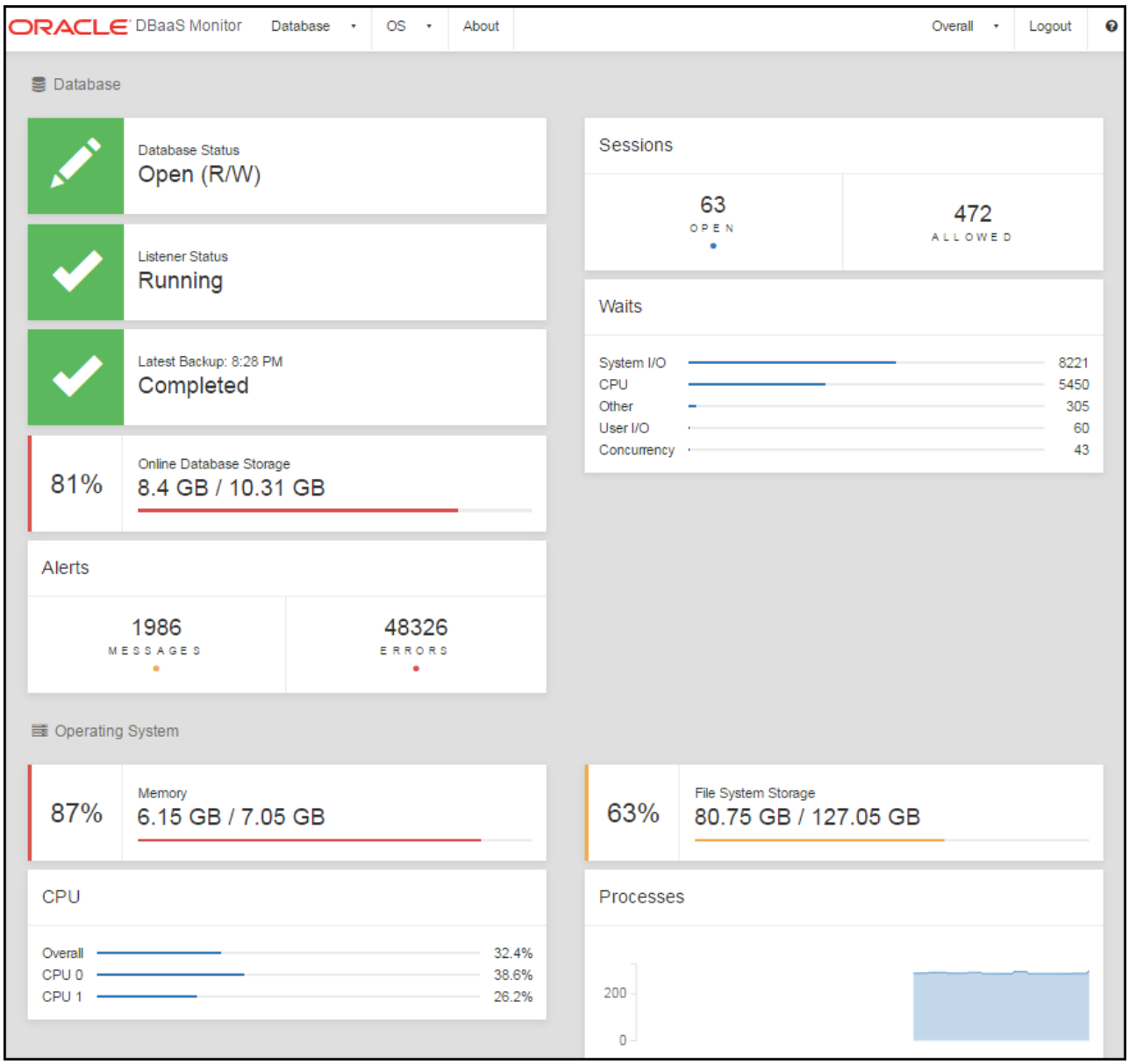
DBAAS Monitor is a browser-based tool that provides database monitoring. Your options to access the deployment are:

- Create an SSH tunnel for the DBAAS monitor port (the https port: 443)
- Enable the security rule in the Compute Service to open the dbaas monitor port

Oracle Internal & Oracle Academy Use Only

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

DBaaS Monitor



The overview page of the DBaaS Monitor gives a quick visualization of the following areas:

- Database
  - Database status
  - Listener status
  - Last backup
  - Online database storage
  - Alerts, sessions, waits

Of particular interest in regard to performance are the Waits, Sessions, and Alerts section.
These provide a general view of the health of the instance.

- Alerts will provide information about warnings and errors posted to the alert log.
- In the Waits section, you would expect in normal instance that User I/O waits will consume the majority of the

Oracle Internal & Oracle Academy Use Only

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

DB Time.

- Operating system
  - Memory
  - CPU
  - Processes
  - File system storage

Oracle Internal & Oracle Academy Use Only

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

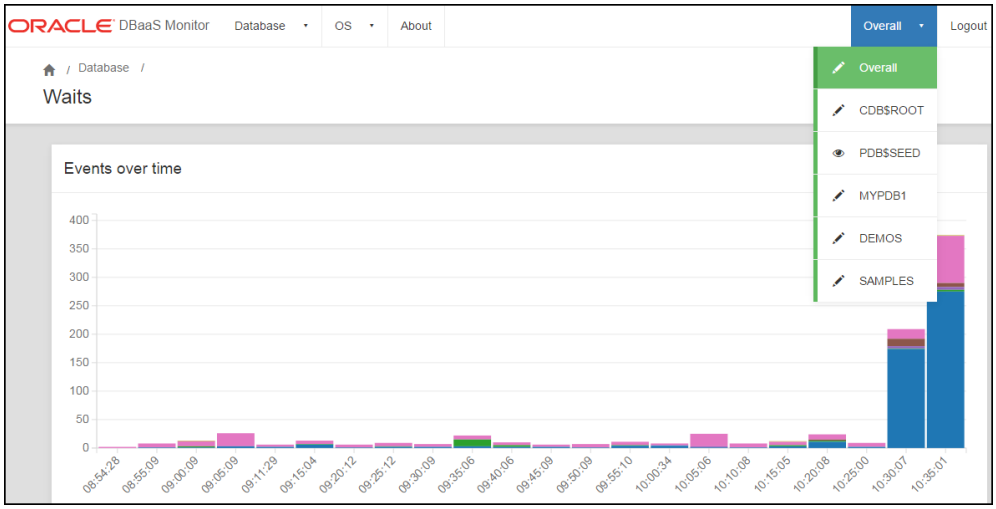
Identifying Performance Issues with DBaaS Monitor

In on-premises databases, you use tools that give you insight into database waits, and high cost SQL to find the areas where tuning could increase performance. The same applies to your database deployment in DBCS.

Following the tuning model:

Determine where the greatest waits are occurring. Use the available tools:

- DBaaS monitor provides a high-level overview of database waits. This tool shows if waits are a problem compared to the capacity of the DBCS service, and a general category of the wait. DBaaS monitor shows the overall waits in the entire deployment and you can select the waits by PDB to see the individual contributions to the overall waits. In a lower panel you can see the wait details showing specific SQL statements that are experiencing the waits.



- The DBaaS monitor also has a Real Time SQL Monitor available. To access, use the menu Database > Real Time SQL Monitor. Using this feature you can identify SQL statements that are running or have run in the recent past and are high-cost statements. The SQL statements can be filtered by PDB in the same way the wait events can be filtered.

Oracle Internal & Oracle Academy Use Only

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

Viewing EM Database Express Performance Hub

EM Database Express or EM Express is the same tool that is available in an on-premises database.

In EM Database Express, the Performance Hub provides a visual overview of the health of the database instance and activity over the last hour by default. You can use the time picker slider at the top for the page to select a specific time period with the time frame shown, or use the Select Time Period button at the top of the page to choose other time frames such as the last day or week. Using the tabs below the time picker, you can view other important performance metrics and diagnostics. In the summary tab view you can see four panes:

- Host Runnable Processes, these are from the OS perspective.
- Active Sessions showing the activity in the instance, the tabs under Active Sessions will show activity by Service, or Container.
- The Memory tab shows the amount and allocation of memory to the Database Instance by default.
- The I/O pane shows the breakdown of the Database instance I/O activity.

For details of the Performance Hub panes, see Enterprise Manager Database Active Report

Note: The screenshots for the Performance Hub show the CDB, or root container level. In the current version of EM express (12.1.0.2) EM express uses a different port for each PDB. With the PDB connection the Performance Hub does not have all of the options that are available for the CDB.

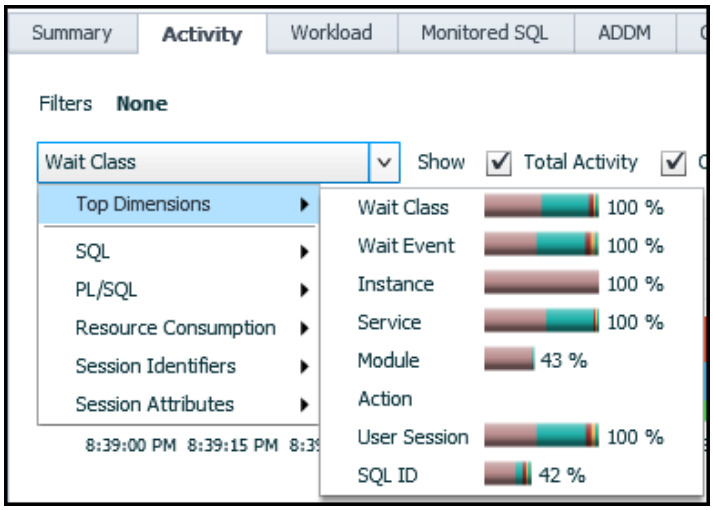
Oracle Internal & Oracle Academy Use Only

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

Identifying Performance Issues with EM Express

Whatever tool you use, the key to identifying performance issues are Wait events, and high cost SQL. With EM express the waits are displayed graphically in several ways:

In the Activity tab, the wait class pull down allows you to filter the results, as well as classify waits in different ways.Waits and the associated SQL are key indicators for determining the root cause of the issue.



The ADDM tab allows you view stored Automatic Database Diagnostic Monitor reports. The ADDM feature requires the High Performance or the Extreme Edition, and is not available in Standard Edition.

The Current ADDM Findings display the latest findings from the ADDM report.

The ADDM reports and findings identify the performance issues with the highest impact in terms of database time, and provide recommendations based on the experience of oracle performance consultants and analysis of the root cause of the issue. Even if a recommendation is not provided, the root cause is identified.

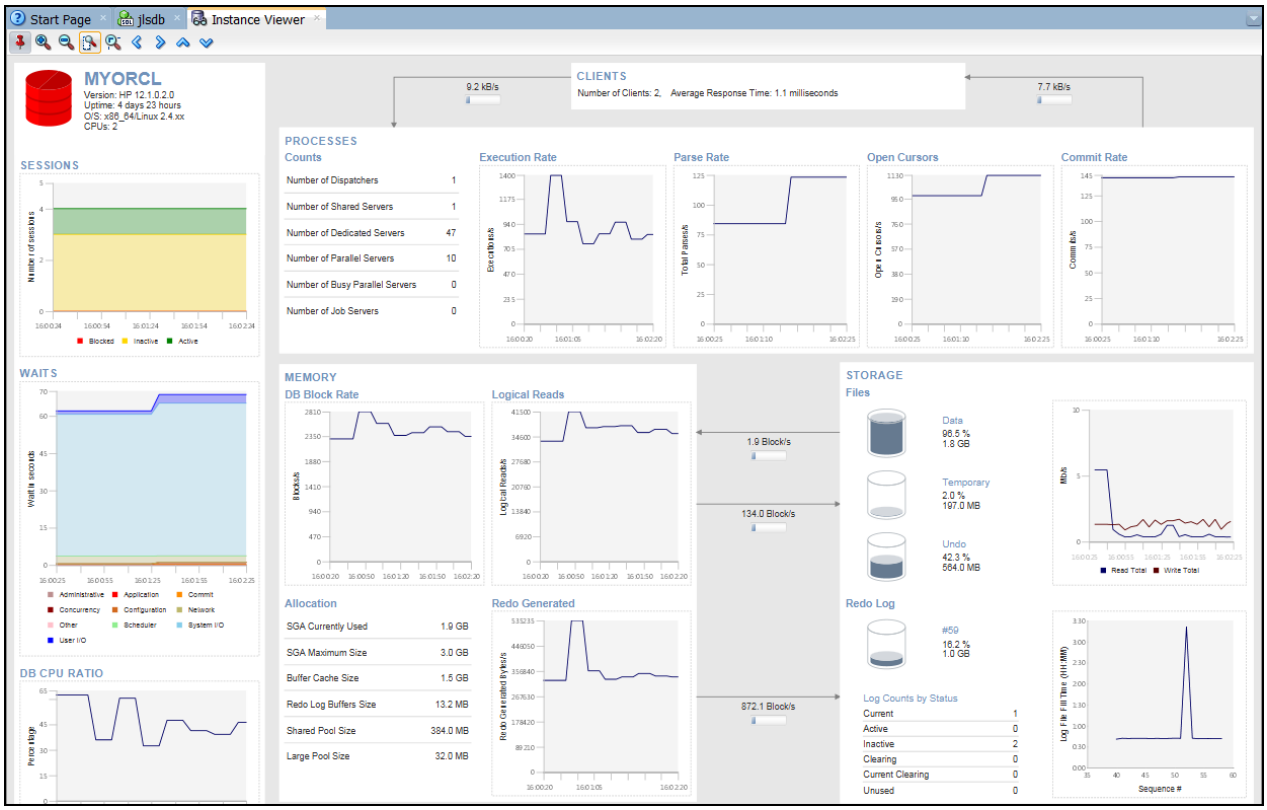
Oracle Internal & Oracle Academy Use Only

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

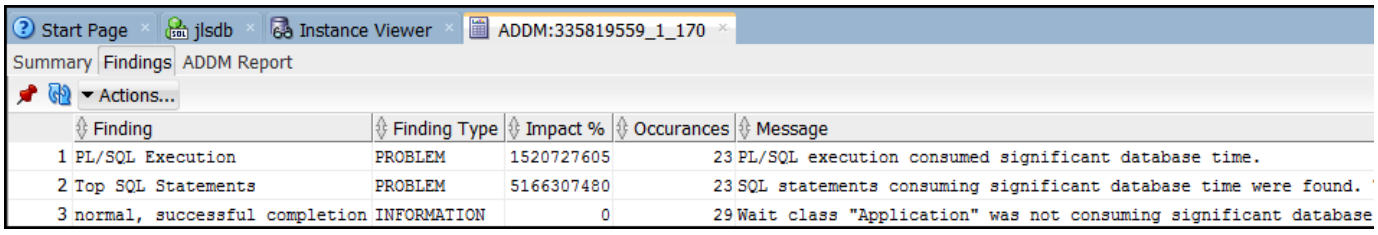
Identifying Performance Issues with SQL Developer

SQL Developer provides much of the same information that is found in EM Express, but formatted in a different manner.

The Instance Viewer provides a real-time overview of the database instance (CDB level). You can navigate to this view by creating a connection in the DBA pane (use the menu View > DBA) then expanding Database Status > Instance Viewer.



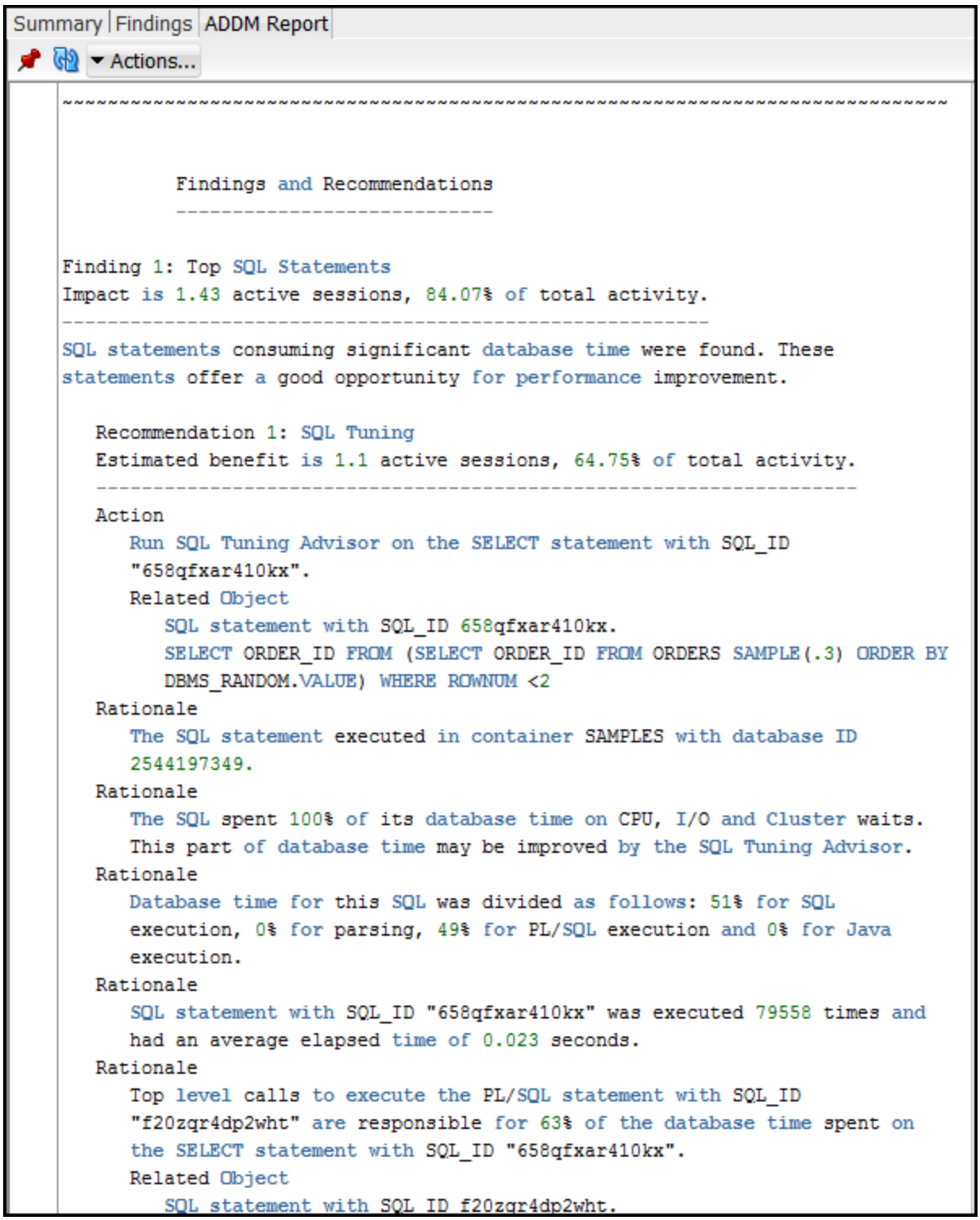
The ADDM reports and findings are also available in the DBA pane. Expand Performance > Automatic Database Diagnostic Monitor, and select the latest ADDM report from a list. The report and findings are displayed in separate tabs. The findings are shown by selecting the Findings tab.



The recommendations are part of the ADDM report. Select the ADDM Report tab and scroll down.

Oracle Internal & Oracle Academy Use Only

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



Oracle Internal & Oracle Academy Use Only

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

Diagnosing Performance Issues with ADDM

You can access ADDM reports in a variety of ways, two ways are shown in the preceding pages. The ADDM report may give findings and possibly recommendations. The findings and recommendations are ranked by the impact they could have on the database performance. The recommendations can be given in several forms: SQL statements, scripts or refer you to another tool such as SQL Tuning Advisor. The possible recommendations are listed in the Database Development Guide: *Overview of Automatic Database Diagnostic Monitor*. On a previous page the ADDM report shows the SQL Tuning Advisor is recommended and the particular SQL statement that might be tuned. A snippet of that report is shown here.

```
Finding 1: Top SQL Statements
Impact is 1.43 active sessions, 84.07% of total activity.
---------------------------------------------------------
SQL statements consuming significant database time were found. These
statements offer a good opportunity for performance improvement.

   Recommendation 1: SQL Tuning
   Estimated benefit is 1.1 active sessions, 64.75% of total activity.
   ------------------------------------------------------------------
   Action
      Run SQL Tuning Advisor on the SELECT statement with SQL_ID
      "658qfxar410kx".
      Related Object
         SQL statement with SQL_ID 658qfxar410kx.
         SELECT ORDER_ID FROM (SELECT ORDER_ID FROM ORDERS SAMPLE(.3) ORDER BY
         DBMS_RANDOM.VALUE) WHERE ROWNUM <2
```

For more details about the ADDM reports, and configuring ADDM, see: *Interpretation of Automatic Database Diagnostic Monitor Findings* and *Configuring Automatic Database Diagnostic Monitor*. Note these references are using the Enterprise Cloud Control 12*c* to access the ADDM reports, and have additional features that include one button implementation of the recommendations not available in EM Express.

Oracle Internal & Oracle Academy Use Only

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

Using SQL Tuning Advisor

SQL Tuning Advisor tool analyzes the highest cost SQL statements for possible better SQL execution plans. The SQL Tuning Advisor has two modes Automatic SQL Tuning Advisor which is run every night in the maintenance window, and the SQL Tuning Advisor which is run on demand to tune one or more SQL statements.

For more details see:

- *About the SQL Tuning Advisor*
- *Configuring the Automatic SQL Tuning Advisor*
- *Running the SQL Tuning Advisor*

For tuning a high-cost SQL statement, the SQL Tuning Advisor is an efficient first pass way to tune. It attempts multiple execution plans to find the best, and runs them multiple times to get reliable statistics. If a better plan is found, it is reported and maybe implemented as a SQL profile. If a better plan is not found, other changes may be recommended such as collecting statistics (if they are stale), adding indexes or other objects, or rewriting the SQL.

SQL Tuning Advisor may be accessed in a variety of ways. In EM Express, any Performance Hub tabs that show the SQL ID also allow you to select that statement for SQL Tuning. In SQL Developer, It is more complicated. Copy and paste the identified SQL statement into a worksheet of the user that is running the statement, and then use the SQL Tuning Advisor button to submit the statement to SQL Advisor. SQL Developer does have the advantage of being able to run and test various rewrites of the SQL statement in the same tool.

There are two cases to consider for your primary access choice:

1. This is primarily a development environment. The users are already using SQL Developer and have full access to the code. Accessing the SQL Tuning Advisor through SQL Developer is convenient and efficient, and allows the developers to quickly access and change code or schema.
2. This is primarily a production environment. The code is not to be changed here. The code might be provided by a third party and if changed it would not be supported. In this case, accessing the SQL Tuning Advisor through EM Express allows you to use SQL Profiles, which does not change any code, but gives directions to the Optimizer to produce more efficient execution plans.

Note: The SQL Tuning Advisor behavior is the same as on-premises, and requires the Oracle Database Tuning Pack.

Oracle Internal & Oracle Academy Use Only

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

Lesson 3: Performance Management

After completing this lesson, you should be able to:

- Determine when the database deployment requires scaling
- Use Resource Manager to avoid scaling
- Increase CPU and memory allocated to a database deployment

Topics:

- Avoiding Database Deployment Scaling
- Control PDB Resource Usage with Resource Manager
- Control Resource Usage by Consumer Groups
- When Does the Deployment Need to Scaleup
- Scale Up CPU and Memory

Oracle Internal & Oracle Academy Use Only

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

Avoiding Database Deployment Scaling

If an Oracle Database Cloud Service (Database as a Service) deployment database is performing poorly or is running out of storage, you can scale up the instance. If your performance tuning activities indicate that you need more computing power or more storage, you can scale your database deployment to satisfy the need.

A deployment database can be scaled in two ways:

- Increase CPU and memory allocated to a deployment database. Scale up the compute shape of a deployment database. The Compute Shape of a deployment database is the combination of the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) for the VM hosting the new deployment database. Database Cloud Service (DBaaS) offers several OCPU/RAM combinations.
- Scale up the storage of a deployment database: This results in adding a storage volume to the deployment database's VM. This operation was shown in *Practice 3-5 Scaling Up Storage*.

Analyzing database performance could point to short-term intermittent load peaks or occasional competition between PDBs or within a PDB for resources.
Before scaling up the deployment database, you can examine the resources allocated and required by the different consumers within the database. Resource Manager can allocate resources (CPU) between PDBs, and among users inside a PDB by allocating:

- Shares of the system resources allocated to PDBs within the CDB so that resources are allocated fairly to all PDBs or more resources to the more important PDBs
- Resources to consumer groups within the non-CDB or PDB

Memory allocations are limited by the deployment database shape, but database initialization parameters can be used to control SGA and PGA allocations as in an on-premises database.

Oracle Internal & Oracle Academy Use Only

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

Control PDB Resource Usage with Resource Manager

Resource Manager is an Oracle Database feature and works in the same way in a DBCS database deployment as on-premises.

Resource Manager will ensure fairness between multiple PDBs in the same CDB by allocating shares of CPU, limiting CPU utilization percent, or parallel server percentage. Resource Manager does not limit CPU usage until the CPU is fully loaded. This policy can be modified with utilization percent that limits the PDB to a specified percentage of CPU even when there is no other demand. The `PARALLEL_SERVER_LIMIT` is a percentage of the `PARALLEL_SERVER_TARGET` ini tialization parameter limiting the number of parallel server processes the PDB can allocate.

The `DEFAULT_CDB_PLAN` allows one share per PDB, so each PDB is guaranteed an equal share when the CPU is fully loaded. When the CPU is not fully loaded, each PDB is allowed as much CPU as it requests. The `DEFAULT_CDB PLAN` p laces no limit on CPU utilization or parallel servers. For example, in a CDB with 5 PDBs using the `DEFAULT_CDB_PLAN`, each PDB is guaranteed 1/5 of the CPU capacity when the CPU is fully loaded, whereas at any other time each PDB is allowed to use as much CPU as it requests.

A custom plan may be created using SQL*Plus or SQL Developer and then edited in EM Express. Using EM Express you can change which plan is active and edit the plan, but not create a new plan. A custom plan allows you to set priorities for the PDBs by giving more important PDBs additional shares. For example, in a CDB with four PDBs (A,B,C,and D) the plan could give each PDB two shares, with PBD B and C getting three shares and PDB D getting four shares for a total of 12 shares. When the total load on the CPU reaches 100%, PDB D is guaranteed four of 12 shares or 1/3 of the total CPU, PDB B and C are guaranteed three of 12, or 1/4 of the total CPU. If any PDB is not using its guaranteed share, any other CPU can request the unused CPU. There is an internal algorithm that allocates the CPU in a fair manner.

To create a new plan, see the Oracle Database 12c: Database Administrators Guide: *Creating a CDB Resource Plan*.

To change the active plan or edit the parameters, see *Using EM Express to Modify a CDB Resource Plan*.

The Resource Manager can force fairness between PDBs when there are intermittent load peaks, so that one PDB does not consume all the resources hurting the performance of the other PDBs. If your applications can tolerate slowdowns as long as all the applications are being serviced, Resource Manager is a reasonable way to provide all the PDBs with a predetermined share of the CPU and not have to scale up or use burst capacity to satisfy load peaks.

Oracle Internal & Oracle Academy Use Only

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

Control Resource Usage by Consumer Groups

The Resource Manager plan at the CDB level allocates resources by PDB, the Resource manager plan at the PDB level allocates resources by consumer groups. The Resource Manager plan at the PDB level works in the same manner as Resource Manager plan in a non-CDB with the exception the PDB level plan has a few restrictions. A PDB level plan cannot have subplans, it is limited to 8 consumer groups and cannot have multiple level scheduling policy.

For a full description of Resource Manager see Oracle 12*c* Database Administrator's guide: *Managing Resources with Oracle Database Resource Manager*

The PDB level plan allows fine control over the resource allocated to consumer groups. In the PDB level plan consumer groups within a PDB may be allocated a percentage of CPU, CPU utilization limits, parallel server limits, active session pool limits, undo space usage limits, and session group switching conditions. User sessions can be assigned to consumer groups by a variety of characteristics: user name, service name, client OS name, and others, This assignment is made by consumer group mapping.

For details of the PDB level Resource Manager plans see Oracle 12*c* Database Administrator's guide: *Creating a PDB Resource Plan*.

Oracle Internal & Oracle Academy Use Only

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

When Does the Deployment Need to Scaleup

How will you determine if scaling the DBaaS instance is required?

- How can you determine if the DBaaS instance is exceeding the CPU capacity of the DBCS deployment?
  - DBaaS Monitor can help you determine if the DBaaS instance is exceeding the CPU capacity. Consider the "On CPU" wait event on the DBaaS Monitor Waits page. The On CPU wait event will be increasing, or have reached a maximum and is holding. Further investigation is needed. *The OS > CPU page in DBaaS monitor can also be of assistance. In the % Idle column the values over many refreshes will be close to zero or less than 10%.*
  - EM Express Performance Hub can show a CPU line in the Activity tab that you can compare with CPU waits; when the CPU waits are coming close the CPU line the CPU is reaching saturation.
  - ADDM reports will also point to overloaded CPU.

Note: Overloaded CPU may be caused by inefficient SQL statements, so tuning the SQL is a cost effective first step. Inefficient use of memory, often caused by poorly designed SQL or poor schema design (missing indexes) has a side effect of loading the CPU and causing excessive I/O. In both of these cases ADDM will point to the root cause, SQL Tuning Advisor will make recommendations, and help you avoid CPU or Memory scaling.

- How can you determine if the DBCS deployment is running out of memory?
  - DBaaS monitor Database > Memory page
  - EM Database Express can help you determine if the DBCS deployment is exceeding the memory capacity through the Performance Hub page.
- How can you determine if the DBaaS instance is running out of space?
  - DBaaS Monitor can help you determine if the DBaaS database is running out of space. DBaaS Monitor shows the fullness for every tablespace.
  - EM Database Express can help you determine if the DBaaS database is running out of space. EM Database Express shows the fullness for every tablespace in each container, PDB, and root. EM Database Express also shows the space used by archived log files.
  - SQL Developer can also display the same type of information.

Note: How to perform storage scaling is covered in Database Deployment Activity Guide : Practice 3-5 Scaling Up Storage

Oracle Internal & Oracle Academy Use Only

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

Scale Up CPU and Memory

Scaling up a database deployment is the operation of increasing two types of resources allocated to the database deployment.

- The number of Oracle Compute Units (OCPUs)
- The amount of memory (RAM) for the compute node hosting the database deployment

Database Cloud Service offers several OCPU/RAM combinations. Read *Scaling Database as a Service*.

Be aware that the database deployment is unavailable while the scaling operation is in progress.

Oracle Internal & Oracle Academy Use Only

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

Summary: Major Differences

The major differences between on-premises databases and database deployment database:

Type of Operation	On-premises Database	Database Deployment Database
Installation	Manual • Oracle Database 11*g* or 12*c* • Database creation	Automatic • Oracle Database 11*g* or 12*c* • Pre-created database
Oracle Database 12*c*	Non-CDBs and CDBs	Only CDBs
Location for database files and backups	Manual	Automatic
User and group	`oracle` user and `oinstall` group	`oracle` and `opc` users, and `oinstall` group
Types of server connection	All types (password, SSH ...)	SSH
Storage allocation	Manual Unix commands	GUI tool: Oracle Database Cloud Service console
Tablespace encryption	None by default	Default encryption (TDE) for user-defined tablespaces: Initialization parameter `encrypt_new_tablespaces = cloud_only`
Backups	Manual or manual scheduling: RMAN> `backup`	Automatic: `bkup_api`
Backed up files	Database files + controlfiles + SPFILE	• All database files + SPFILE + password file and others from `/home/oracle/bkup/dbcfg.spec` file • OS files from `/home/oracle/bkup/oscfg.spec` file
Recovery	RMAN> `recover`	`dbaascli orec`
Backup destination	Single or Cloud	Dual • Local compute node storage • Oracle Storage Cloud Service container
Patch discovery	None Oracle Support EM Cloud Control	GUI tool: Oracle Database Cloud Service console
Log and diagnostics files cleanup	None	Automatic, using a configuration file
Upgrade	GUI tool: `dbua`	None

Oracle Internal & Oracle Academy Use Only

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

Port access	Automatic configuration via dbca • EM Express • EM Cloud Control • Listener registration	Automatic configuration via pre-defined security rules to enable when required
Monitoring tools	EM Express, EM Cloud Control, SQL Developer	DBaaS Monitor, EM Express, EM Cloud Control, SQL Developer

Oracle Internal & Oracle Academy Use Only

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