ORACLE
University

Integrated Cloud Applications & Platform Services

Oracle Database Cloud for Oracle DBAs

Student Guide

D99336GC30 | D101205

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

ORACLE

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.

Contents

1

Getting Started

Objectives

After completing this lesson, you should be able to:

- Describe areas of focus in this course
- Describe the course objectives

Areas of Focus in this Course

- Creating and managing an Oracle Database Cloud Service database deployment
- Migrating an on-premises Oracle Database to Oracle Database Cloud Service
- Managing performance in an Oracle Database Cloud Service database deployment

Course Objectives

- Describe the main features of Oracle Database Cloud Service
- Subscribe to Oracle Cloud services
- Create your own database deployment
- Administer your database deployment
- Connect to your database deployment through different ways
- Ensure security of network access and data encryption for the service
- Administer the Oracle database in your database deployment
- Monitor your database deployment
- Migrate on-premises databases to your database deployment
- Monitor performance in your database deployment

What You Need for this Course

- Credentials to connect to your assigned lab environment
- Instructions on how to download your Student Guide and Activity Guide

Summary

In this lesson, you should have learned how to:

- Describe areas of focus in this course
- Describe the course objectives

Practice 1: Overview

There are no practices for this lesson.

2

Overview of Oracle Cloud Services

ORACLE

Oracle Cloud offers delivers best-in-class services designed specifically to increase IT agility and efficiency, reduce operational expenditure, and increase data security for organizations around the world. With services across SaaS (Software as a Service), DaaS (Data as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a service), users can leverage a familiar tool set and intuitive user experience on a standardized and secure enterprise-grade cloud platform, built with the same software and architecture they are used to.

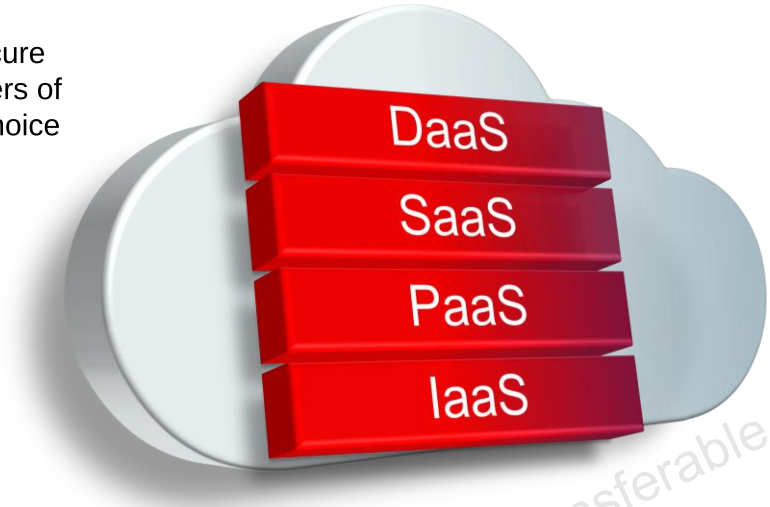
Objectives

After completing this lesson, you should be able to:

- Describe Oracle Cloud Services Offerings
- Understand Oracle Cloud Subscription Models
- Subscribe to Oracle Cloud Services

Oracle Cloud Services

- A complete, open, and secure platform that spans all layers of the cloud, and provides choice and access to innovation



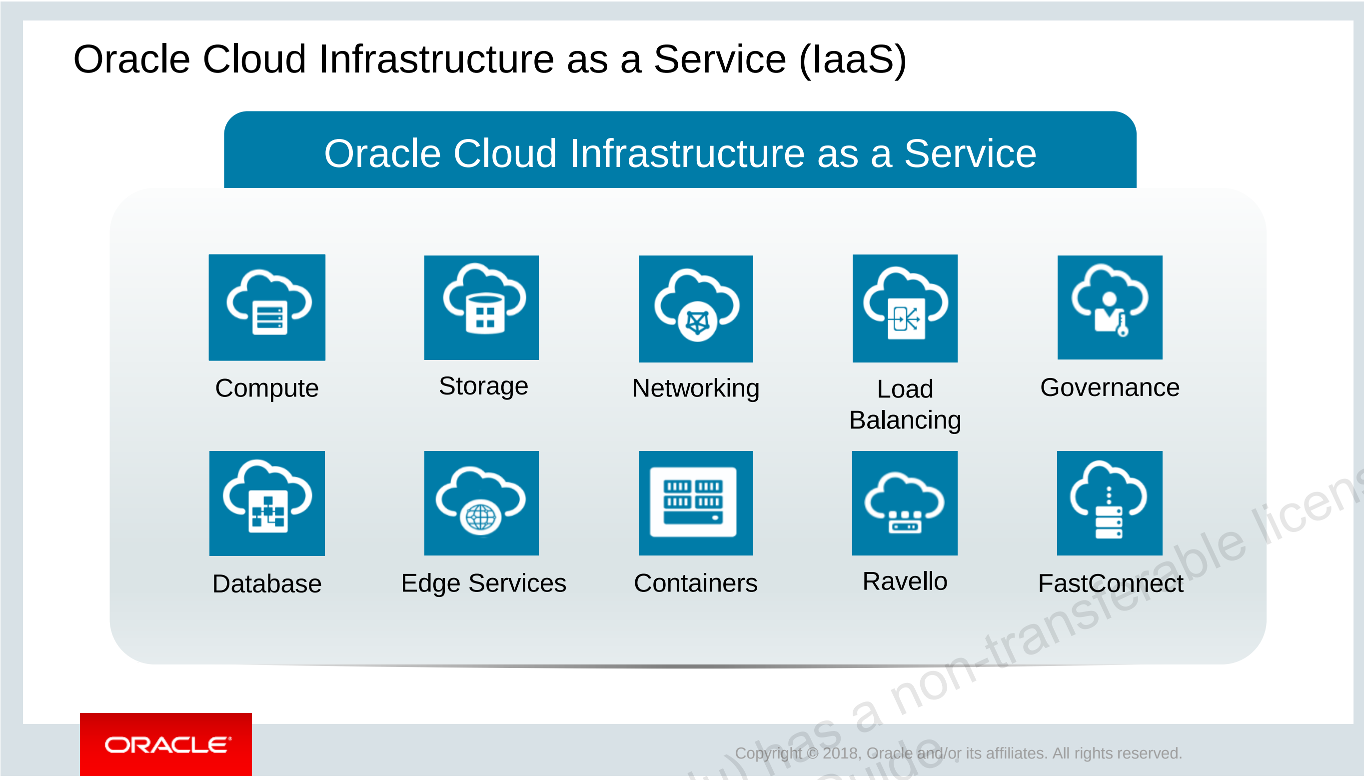
ORACLE

Data-as-a-Service (DaaS): Data Cloud provides data from a wide variety of Oracle and third-party sources that customers can leverage for deeper insights into their clients to enable modern marketing campaigns.

Software-as-a-Service (SaaS): Oracle offers the most integrated, complete Cloud suite of SaaS applications, enabling customers to modernize their business using the latest technologies such as artificial intelligence and machine learning.

Platform-as-a-Service (PaaS): Oracle offers the broadest range of PaaS services in the industry that enables developers, IT professionals and business leaders to develop, extend and secure applications that leverage advanced analytics.

Infrastructure-as-a-Service (IaaS): Oracle offers the highest performance, lowest cost IaaS in the industry, enabling customers to run their application workloads in the Oracle Cloud.



Oracle Infrastructure as a Service (IaaS) offers the broadest spectrum of cloud compute options available in the industry today.

Oracle cloud infrastructure provides a robust suite of compute options including virtual machines (VMs), bare metal instances, dedicated Compute and Oracle Cloud at Customer (on-premises public cloud behind the customer's firewall). Oracle has made cloud and Infrastructure adoption easier than ever before by offering migration solutions such as Oracle Ravello Cloud Service and Oracle Container Cloud Service.

- General Purpose VMs – Elastic compute server with a hypervisor that can run Linux, CentOS, Ubuntu, Windows, Oracle Solaris and scale elastically
- Bare Metal Servers – Get a dedicated high performance, physical compute server at a push of a button and scale elastically with additional servers
- Engineered Systems – Get Oracle Exadata, Big Data, and Sparc T7 Systems for fully isolated enterprise workloads
- Dedicated Compute – A dedicated rack of compute servers running Linux or Windows for full physical isolation
- Oracle Cloud Container Service – Run Docker-based Containers providing greater density and better performance than VMs on a physical host
- Oracle Ravello Cloud Service – Migrate and run virtual machines in the cloud extremely quickly and easily

Oracle Cloud Platform as a Service (PaaS)

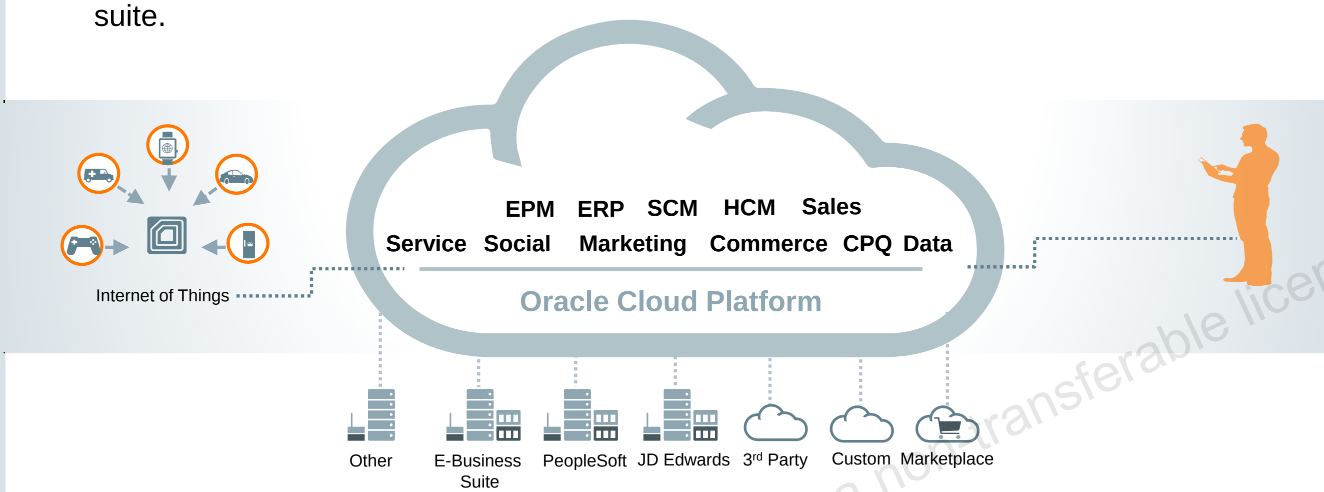
- Oracle Cloud Platform as a Service (PaaS) provides a comprehensive, open, integrated and Hybrid Cloud platform that is highly scalable, intelligent, secure, and globally available

ORACLE

Oracle Cloud Platform as a Service (PaaS) empowers businesses with faster innovation, higher productivity, and much lower operational costs. PaaS offers a comprehensive set of solutions to develop, extend, connect, secure, mobilize, and analyze business applications across a variety of services. Customers working with data management, application development, integration, business analytics, security, and more can all take advantage of the performance and agility, that Oracle PaaS offers.

Oracle Cloud Software as a Service (SaaS)

- Oracle Cloud Software as a Service (SaaS) applications provide you with the speed and innovation of best-of-breed cloud software in a complete, secure, and connected cloud suite.



Oracle Software as a Service (SaaS) offers a full range of cloud applications designed to increase operational efficiency and manage business processes from start to finish. Oracle's SaaS portfolio includes:

- Full breadth of financial applications for all of the functions of a CFO. Oracle SaaS covers the entire finance process from planning and budgeting, to narrative reporting. It also includes procure-to-pay, source-to-settle, as well as sourcing, contracts, and supplier management.
- Project management applications that focus on financial management, as well as project execution management. (Resource management/deliverable management.)
- Supply chain management applications that help businesses stay on top of logistics, order-to-cash, and product management.
- Human Capital Management that supports employees from hire to retire.
- Full breadth of Customer experience to Sales, Marketing, and Commerce applications.

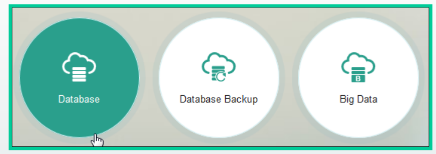
Oracle Cloud Data as a Service (DaaS)

Data Cloud

- Oracle's Data Cloud unifies data assets giving marketers the tools to understand the entire customer journey and reach consumers more effectively across all marketing channels.
- Provides a master data management solution that aggregates individual application data to maintain a complete, accurate 360º customer view.
- Designed to drive more agile front office and back office operations.
- The Oracle CX Cloud Suite provides a consistent and unified data protection regime for global businesses.

Data as a Service (Data as a Service) enables you to subscribe to, update your internal business data for your unique industry, selecting from new audience data, prospect and social data from Oracle's external data cloud service.

Oracle Database Cloud Offerings



- Oracle Database Cloud Service
  – Virtualized and Bare Metal
- Oracle Database Exadata Cloud Service
- Oracle Database Exadata Cloud Machine
- Oracle Database Exadata Express Cloud Service – Managed
- Oracle Database Schema Cloud Service - Managed

ORACLE

Oracle Database Cloud, in Oracle Platform as a Service (PaaS), offers the following services:

- **Oracle Database Cloud Service – Virtual:** Database Cloud Service offers elastic database services for application development, test, and production deployment. The service delivers an easy to use web console user interface and RESTful APIs to provision and administer Oracle Database on Oracle Compute Cloud offerings.
- **Oracle Database Cloud Service – Bare Metal:** Cloud Database Service on bare metal offers on-demand, pay-per-use database services with the performance of dedicated hardware and local NVMe storage, and the reliability of RAC, on a low latency, highly configurable, and secure Virtual Cloud Network.
- **Oracle Database Exadata Cloud Service:** Exadata Cloud Service brings the full power of Exadata to the Oracle Cloud. The service includes all the benefits of Exadata performance and 100% compatibility with existing business-critical applications and on-premises databases, ensuring a smooth transition to the cloud.
- **Oracle Database Exadata Cloud Machine:** Oracle Database Exadata Cloud Machine delivers the world's most advanced database cloud to customers who require their databases to be located on-premises.
- **Oracle Database Exadata Express Cloud Service – Managed:** Exadata Express provides your own Oracle Database Enterprise Edition running the latest database release on Exadata. It's a fully managed service packed with features for modern application development.
- **Oracle Database Schema Cloud Service – Managed:** Schema Service runs Application Express (APEX), giving customers a declarative development environment to rapidly create data-rich web apps.

This course focuses solely on Oracle Database Cloud Service.

Oracle Cloud Subscription Offerings

- Universal Credits
- Bring Your Own License (BYOL)
- Traditional Metered Service Offerings
- Traditional Nonmetered Service Offerings

ORACLE

Universal Credit Services: These are bundled Oracle Infrastructure as a Service (Oracle IaaS) and Oracle Platform as a Service (Oracle PaaS) cloud services and offer unlimited access to all the services in these categories.

Bring Your Own License: Customers can move on-premises licenses to Oracle PaaS.

Traditional Metered and Non-Metered Service Offerings: If you are signing up for an Oracle Applications (Software as a Service) offering, or if you are working directly with Oracle Sales to sign up for Oracle Cloud, then the procedures for signing up and accessing your services are different than those required when you sign up for Oracle Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) services via the Oracle Cloud website.

For example, when signing up for Oracle Cloud with help from Oracle Sales, you might need to use your Oracle Account (oracle.com) credentials and access the My Account application. In these cases, you might have to activate your service when prompted via email.

For more information, contact your Oracle Sales representative.

Universal Credits

- Universal access to all current and future IaaS and PaaS services
- Enables flexibility to upgrade, expand, or move services across data centers

Consumption Choices

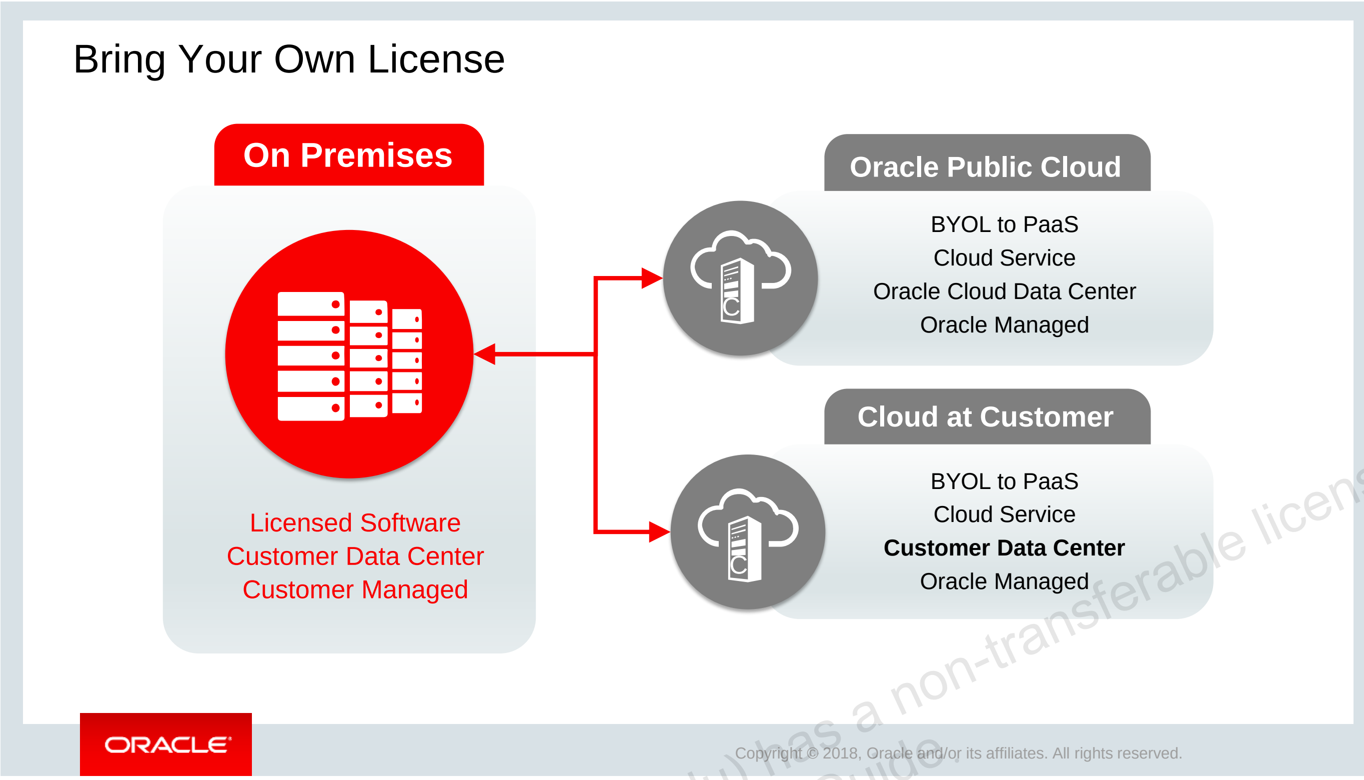
Pay As You Go	Universal Credits Monthly Flex
• No upfront commitment • Pay only for what you use • Pay in arrears based on usage	• 1 year minimum term • Based on the results of your monthly cost estimate
• List Price • Built for land and expand • Best when usage is uncertain • Elastic payments based on usage	• PaaS savings vs Pay As You Go • Additional discounts based on size of deal and term of deal • Predictable spending • Spend more, save more

ORACLE

Universal Credits: Universal Cloud Credits make it easy for customers to take advantage of Oracle Cloud services. Universal Cloud Credits can be applied to all Oracle IaaS and PaaS service in the public cloud, and allow customers to pay for services as they use them.

When you sign up for an Oracle Cloud Account, you have unlimited access to all eligible services, and have the flexibility to sign up for a pay as you go subscription or the Monthly Flex plan. The monthly flex plan allows customers to pay in advance for a year with estimates based on monthly usage, which can help reduce cost.

Both of these payment plans can be applied to any new eligible cloud service as soon as they become available.



Bring Your Own License (BYOL): The BYOL program gives customers the ability to modernize their infrastructure while using their existing investments and licenses. Customers with perpetual licenses for eligible Oracle Database services can convert them to the standard package with BYOL pricing. This program ensures a discounted price, with the added performance that Oracle Cloud offers.

Traditional Metered and Nonmetered Cloud Subscription

- Traditional Metered cloud subscription offerings are applicable for Oracle Infrastructure as a Service (Oracle IaaS) and Oracle Platform as a Service (Oracle PaaS) Cloud services.
- With the metered subscription, you're billed based on your actual usage, with Pay as You Go pricing.
- Metered Cloud subscriptions can be purchased from Oracle Store or by contacting Oracle Sales.
- With the nonmetered cloud subscription offerings, you pay for a set amount of users over the course of the service period. You're billed up front based on that committed quantity. The nonmetered offerings are applicable to Oracle IaaS, Oracle PaaS, and Oracle SaaS Cloud services (as well as a few other offerings).
- Nonmetered cloud subscriptions offer specific services at a negotiated price point, and are suitable for government accounts.

ORACLE

See Also:

- *Buying a Traditional Metered Subscription to an Oracle Cloud Service* in the *Getting Started with Oracle Cloud* guide:

  https://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/csgsg/buying-metered-subscription-oracle-cloud-service.html
- *Buying a Nonmetered Subscription to an Oracle Cloud Service* in the *Getting Started with Oracle Cloud* guide*:*

  https://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/csgsg/buying-nonmetered-subscription-oracle-cloud-service.html

Subscribe to an Oracle Cloud Service

To subscribe to an Oracle Cloud Service, perform the following steps:

1. Order an Oracle Cloud account in one of the following ways:
   - Sign up for a free trial Cloud account. See *Requesting a Trial Subscription* in Getting Started with Oracle Cloud.
   - Order a paid subscription to an Oracle Cloud Service. Estimate your monthly cost and choose the Pay As You Go and/or Monthly Flex subscription plans. See *Selecting a Payment Plan*
2. Activate the service:
   - For accounts running on Universal Credits, see *Activating Your Trial Subscription*
3. Verify that the service is running.
4. Upgrade to a paid Oracle Cloud account after your trial period.

ORACLE

Detailed information about subscribing to an Oracle Cloud Service trial and purchasing a subscription to an Oracle Cloud Service can be found in the *Getting Started with Oracle Cloud* guide.

- *Requesting a Trial Subscription*
  http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSGSG432
- *Buying an Oracle Cloud Subscription*
  https://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/csgsg/buying-oracle-cloud-subscription-new-cm.html
- *Signing Up for a Cloud Account*
  https://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/csgsg/signing-cloud-account.html
- *Activating Other Types of Subscriptions*
  http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSGSG486

Summary

In this lesson, you should have learned how to:

- Describe Oracle Cloud Services Offerings
- Understand Oracle Cloud Subscription Models
- Subscribe to Oracle Cloud Services

Practice 2: Overview

- 2-1: Exploring the Oracle Cloud Options
- 2-2: Exploring Your Oracle Cloud Account

3

Creating a Database Deployment

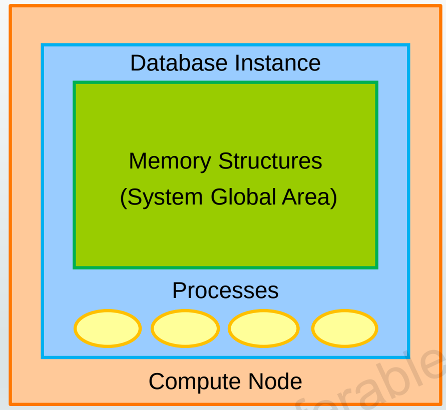
Objectives

After completing this lesson, you should be able to:

- Understand Oracle Database Cloud Service database deployment and Oracle Database instance
- Understand the ease of creation of Oracle Database Cloud Service database deployment compared to an on-premises Oracle Database installation
- Create a QuickStart and custom Oracle Database Cloud Service database deployment
- Describe the resources that are allocated when a database deployment is created

Overview of a Database Deployment and an Database Instance

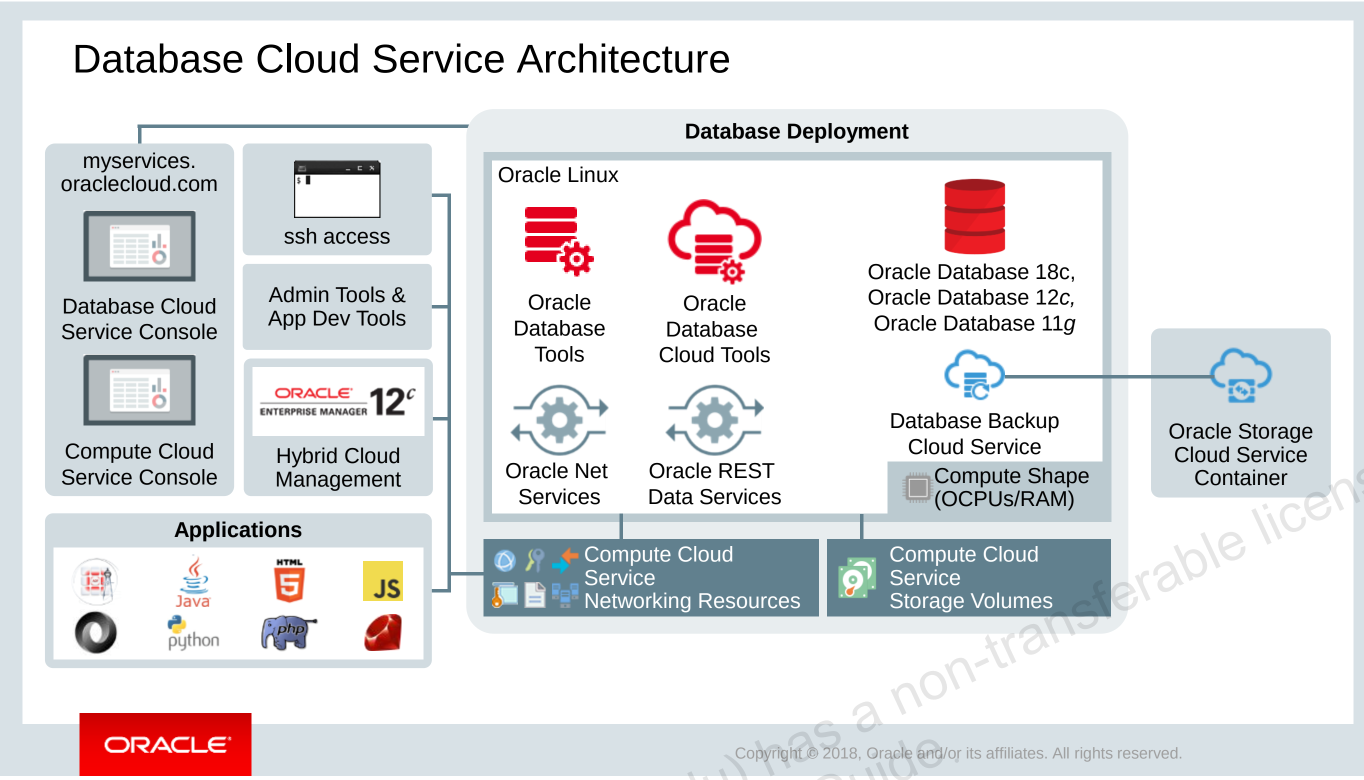
- An Oracle Database Cloud Service database deployment is a compute environment that includes:
  - A compute node (Linux virtual machine)
  - Oracle software
  - A pre-created Oracle Database
  - Additional cloud tooling
- An Oracle Database instance is a collection of:
  - Memory structures
  - Processes



ORACLE

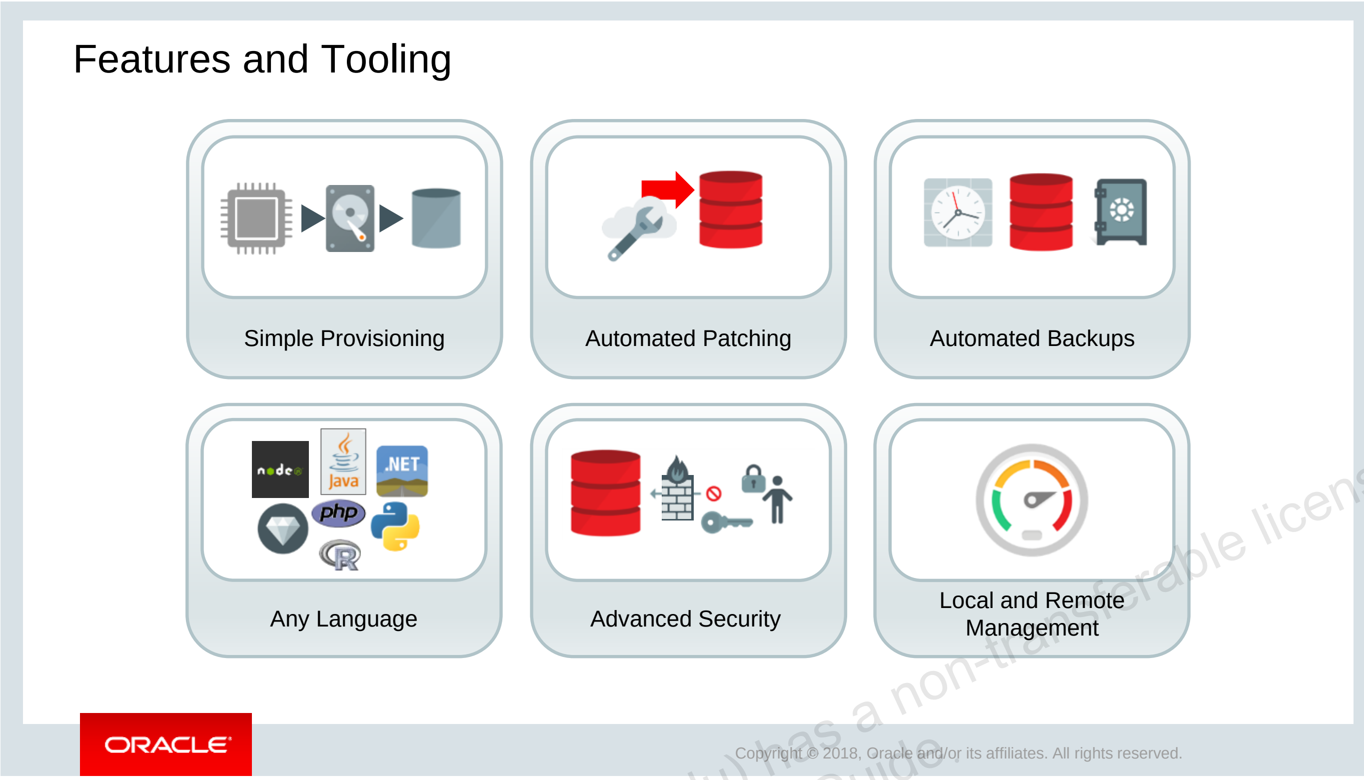
An Oracle Database Cloud Service database deployment contains a pre-created Oracle Database. You have full access to the features and operations available with Oracle Database, with Oracle providing the computing power, physical storage and (optionally) tooling to simplify routine database maintenance and management operations.

An Oracle Database instance comprises memory structure and processes. In your on-premises installation of Oracle Database, as well as in an Oracle Database Cloud Service database deployment, the memory structures and processes are allocated when you start the database instance.



When you create database deployments, Database Cloud Service creates compute nodes to host the database, using computing and storage resources provided by Oracle Compute Cloud Service. Additionally, it provides access to the compute nodes (and thus to the database) using networking resources provided by Oracle Compute Cloud Service.

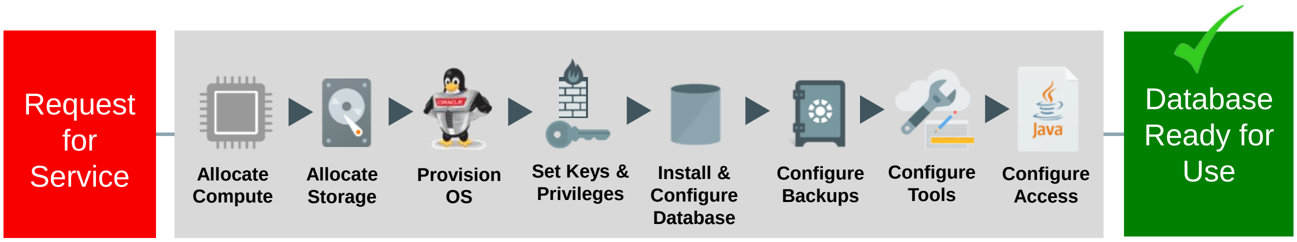
In this lesson, the components that comprise the Oracle Database Cloud Service architecture will be reviewed.



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 Cloud Service supports any language.
- **Advanced Security:** Built-in 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 18c and 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.

Automated Database Provisioning



Automated provisioning based on input to the Create Oracle Database Cloud Service Instance wizard or through REST APIs

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.

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

Comparison: DBCS Deployment and an On-Premises Database

Creating a DBCS database deployment through a menu-driven wizard:

1. Log into to your Cloud account.
2. Run the DBCS wizard and select the Database version, edition, and database type, and specify your administrative password and the backup option.
3. Confirm your selection, and your Database deployment is created for you.

Creating an on-premises Oracle database:

1. Perform preinstallation configuration to setup network, users, and operating system.
2. Install Oracle Database software, by running the installer in interactive or silent mode, or through response files.
3. Install database options and management packs.
4. Configure the database instance.
5. Configure a PDB (Oracle Database 18c or 12c).
6. Configure and schedule backups.
7. Install patches.

ORACLE

The creation of an Oracle Database Cloud Service database deployment is accomplished through a menu-driven wizard. The wizard provides many best practice default values and requires only a few choices by the user.

If you use a QuickStart template to create your Oracle Database deployment, then you only need to specify the deployment name, and the service provisions the deployment based on default values.

An installation of Oracle Database in an on-premises environment requires complex preinstallation configuration steps, technical knowledge, and decision-making by the user.

Using the Wizard to Create a Database Deployment

The wizard steps you through the database deployment creation, prompting for the following information:

- **Subscription:** Software release, software edition, and Database type
- **Service:** Service name, compute shape, time zone, and SSH key
- **Database Configuration:** Storage, database name (SID), PDB name, character set, and password for administrative accounts
- **Backup and Recovery Configuration:** Backup destination and cloud storage information, if applicable
- **Additional Deployment Choices:** Oracle Real Application Clusters (Oracle RAC), Active Data Guard, Hybrid Disaster Recovery (DR), In-Memory, and Oracle GoldenGate

ORACLE

To create a database deployment on Oracle Database Cloud Service, use the Provision New Oracle Database Cloud Service wizard.

Creating a Database Deployment:

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3299

Creating a QuickStart Database Deployment

- For accounts that use Universal Credits, Oracle Database Cloud Service provides QuickStart templates to create database deployments of commonly used configurations.
- QuickStart templates Database Standard Edition and Enterprise Edition available for creating single-instance Oracle Database.
- QuickStart Extreme Performance template available for creating Oracle Database Enterprise Edition clustered database using Oracle Real Application Clusters (Oracle RAC), housed on two compute nodes.
- Select a template and retain or change the default deployment name. Database Cloud Service then uses an Oracle Cloud Stack Manager template to provide all the other deployment information.
- Configure automatic backups for the database deployment as required.

ORACLE

Creating a QuickStart Database Deployment:

https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbi/create-quickstart-deployment.html#GUID-6BC76A65-4CAD-4230-9C99-E60B86F366E2

Software Editions: Included Capabilities

Edition	Included Capabilities
Standard	Oracle Database Transparent Data Encryption, Oracle Database Standard Edition 2
Enterprise	Oracle Database Transparent Data Encryption, Oracle Database Enterprise Edition, Data Masking and Subsetting Pack, Diagnostics and Tuning Packs, and Real Application Testing
Enterprise - High Performance	Oracle Database Transparent Data Encryption, Oracle Database Enterprise Edition, Data Masking and Subsetting Pack, Diagnostics and Tuning Packs, and Real Application Testing, Multitenant, Partitioning, Advanced Compression, Advanced Security, Label Security, Database Vault, OLAP, Advanced Analytics, Spatial & Graph, Database Lifecycle Management Pack and Cloud Management Pack for Oracle Database
Enterprise - Extreme Performance	Oracle Database Transparent Data Encryption, Oracle Database Enterprise Edition, Data Masking and Subsetting Pack, Diagnostics and Tuning Packs, and Real Application Testing, Multitenant, Partitioning, Advanced Compression, Advanced Security, Label Security, Database Vault, OLAP, Advanced Analytics, Spatial & Graph, Database Lifecycle Management Pack and Cloud Management Pack for Oracle Database, Oracle Real Application Clusters (Oracle RAC), In-Memory Database, Active Data Guard

When you create a database deployment, you can choose from four software editions. The slide lists the Database editions available and their included capabilities. For current information, see https://cloud.oracle.com/en_US/database/pricing.

General Purpose Shapes

Shape	OCPUs	vCPUs	Memory
OC3	1	2	7.5
OC4	2	4	15
OC5	4	8	30
OC6	8	16	60
OC7	16	32	120

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

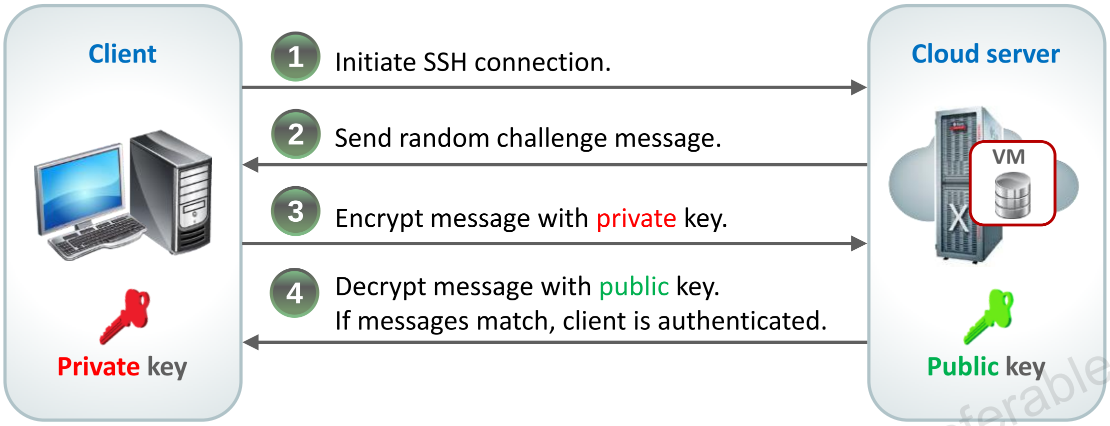
An OCPU is defined as the CPU capacity equivalent of one physical core of an Intel Xeon processor with hyper threading enabled. Each OCPU corresponds to two hardware execution threads, known as vCPUs.

High Memory Shapes

Shape	OCPUs	vCPUs	Memory
OC1M	1	2	15
OC2M	2	4	30
OC3M	4	8	60
OC4M	8	16	120
OC5M	16	32	240

The table describes the high memory shapes, including the shape name, number of OCPUs, number of vCPUs, and amount of memory.

How SSH Key Pairs are Used



By default, network access to the compute nodes associated with Oracle Database Cloud Service is provided by Secure Shell (SSH) connections on port 22. SSH is a cryptographic network protocol that uses two keys, one public and one private, to provide secure communication between two networked computers. Port 22 is the standard TCP/IP port that is assigned to SSH servers.

Creating an SSH Key Pair

Perform one of the following:

- Generate the SSH key pair as part of the database deployment creation.
- Create the SSH key pair before creating your database deployment.

1

Local PC

OPTIONAL: Generate and store private and public key pair on local PC using PuTTY keygen utility, or on Linux.

2

Cloud server

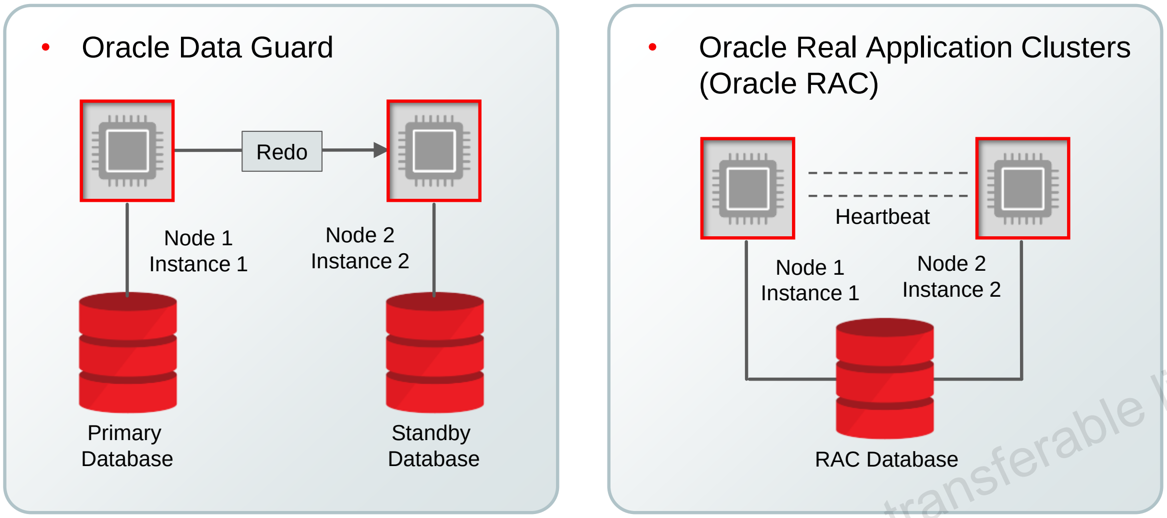
Create the Database Cloud Service Database deployment, creating the keys as part of the deployment creation or provide the pre-created public key.

ORACLE

When you create a database deployment, you can create an SSH key pair through the wizard or you can upload a public key file that you have created by using a utility.

See “Generating a Secure Shell (SSH) Public/Private Key Pair” in *Using Oracle Database Cloud Service* for additional information.

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.

Oracle Data Guard can also be used to offload reporting workload or backups from the primary (read-only physical standby) and can be used for testing (Snapshot standby).

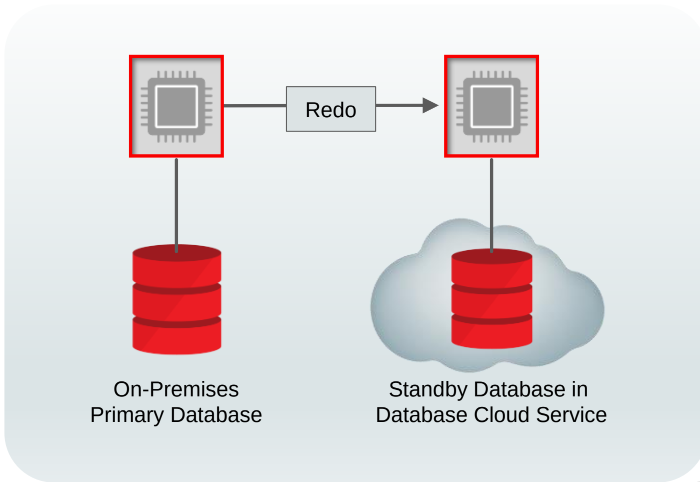
Oracle Real Application Clusters (Oracle RAC)

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

Oracle Real Application Clusters (Oracle RAC) is a clustered version of Oracle Database based on a comprehensive high-availability stack that can be used as the foundation of a database cloud system as well as a shared infrastructure, ensuring high availability, scalability, and agility for any application.

See *Using Oracle Database Cloud Service* for detailed information on these options.

Hybrid Disaster Recovery (HR) Deployment



ORACLE

You can create a Hybrid Disaster Recovery (DR) Oracle Database Cloud Service database deployment with a primary database on your own host system and a standby database in the cloud.

You can use the Oracle Database Cloud Service creation wizard to create the Hybrid DR deployment which uses Oracle Data Guard.

See *Using Oracle Database Cloud Service* for detailed information on creating a hybrid DR database deployment.

Storage Used for Database Files

Storage Volume	Description
bits	60 GB volume completely allocated to `/u01` on the virtual image.
boot	32 GB volume allocated to the following file system mounts on the virtual machine: `/` (root), `/boot`, and swap space.
data	GB size equal to the value provided in the Usable Data Storage field during the database deployment creation process, with a minimum of 15 GB. This volume is completely allocated to `/u02` on the compute node.
fra	If both cloud and local storage backups are configured, then GB size equal to 1.7 times the size of the `data` volume. If backups are not configured or only cloud backup is configured, then 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 compute node.
redo	25 GB volume completely allocated to `/u04` on the compute node.

ORACLE

When a Database Cloud Service deployment is created, the `bits`, `boot`, `data`, `fra`, and `redo` storage volumes are created.

File System Layout

File System Mount	Description
`swap`	Swap space; 4 GB allocated from the `boot` Compute Cloud storage volume
`/ (root)`	Operating system files; 25.5 GB allocated from the `boot` Compute Cloud storage volume
`/boot`	Operating system kernel; 500 MB allocated from the `boot` Compute Cloud storage volume
`/u01`	Oracle product software; the entire `bits` Compute Cloud storage volume
`/u02`	Oracle Database data storage; the entire `data` Compute Cloud storage volume
`/u03`	Database backup storage; the entire `fra` Compute Cloud storage volume
`/u04`	Database redo logs; the entire `redo` Compute Cloud storage volume

ORACLE

When a database deployment is created, Oracle Cloud Service storage volumes are created and allocated as described in the table.

Creating a Database Deployment by Using a Cloud Backup

- Create a database deployment where the database is from a backup created using Oracle Database Backup Cloud Service.
- The technique is also referred to as "instantiate from backup."
- Perform these steps to create the database deployment:
  1. Create a Database Cloud Service database deployment hosting a single-instance database.
  2. If necessary, scale up the deployment's storage to accommodate the source database.
  3. Replace the database on the deployment with a cloud backup by using the Oracle Database Cloud Service console or the `dbaasapi` utility.
- See Creating a Database Deployment Using a Cloud Backup in the *Using Oracle Database Cloud Service* guide for the most current instructions.

ORACLE

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

In brief, you create a Database Cloud Service database deployment hosting a single-instance database and then you replace the newly created database using another database's backup to Oracle Database Backup Cloud Service. The database from which the backup was made is called the "source database".

Creating a Database Deployment Using a Cloud Backup

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=GUID-2A704C0A-B27A-4604-A2A6-BFADC899FF9B

Creating a Database Deployment from a Snapshot

- Create a database deployment from a snapshot you have taken of another database deployment in the same identity domain.
- Database deployment is referred to as a "linked clone" because its storage is linked to the snapshot's storage.
- Perform two steps to use linked clones:
  1. Create a snapshot. See Creating a Snapshot in *Using Oracle Database Cloud Service*.
  2. Create a database deployment from a snapshot. See Creating a Database Deployment from a Snapshot.

ORACLE

On database deployments hosting a single-instance database, Oracle Database Cloud Service 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, then the file data in the new linked clone deployment will reflect the data from the snapshot and not from the other linked clone deployment.

When the snapshot is created, you can use it to create as many database deployments as needed.

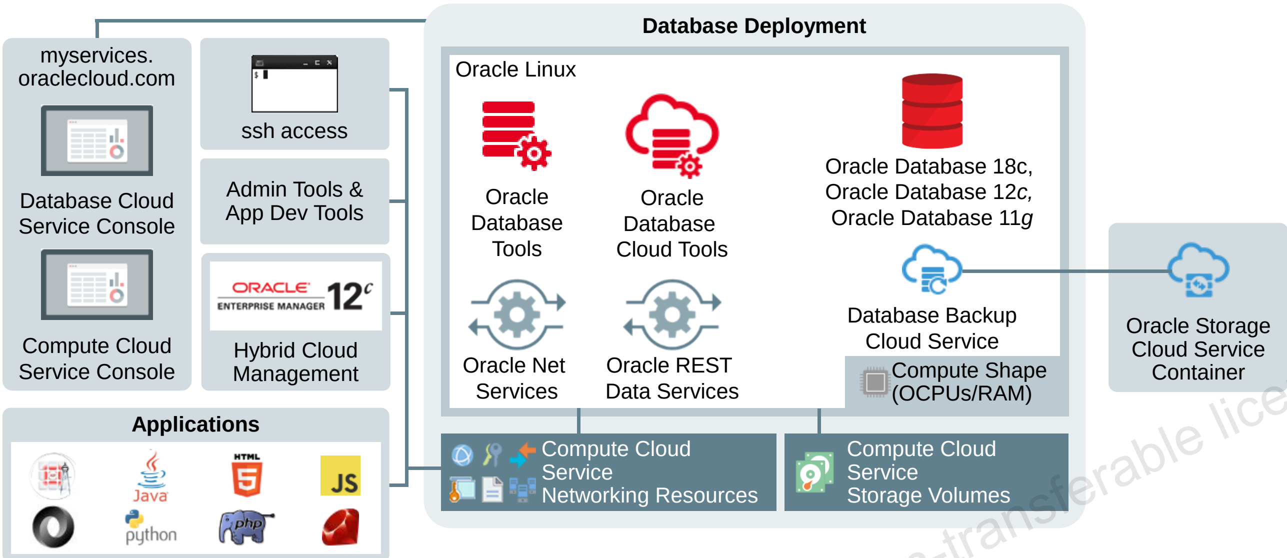
Creating a Snapshot

https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbi/create-and-manage-snapshots.html

Creating a Database Deployment from a Snapshot

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-66B29F07-4252-44EB-969B-70CBC8524C45

Review: Database Cloud Service Architecture



ORACLE

Summary

In this lesson, you should have learned how to:

- Understand Oracle Database Cloud Service database deployment and Oracle Database instance
- Understand the ease of creation of Oracle Database Cloud Service database deployment compared to an on-premises Oracle Database installation
- Create a QuickStart and custom Oracle Database Cloud Service database deployment
- Describe the resources that are allocated when a database deployment is created

ORACLE

Practice 3: Overview

- 3-1: Creating a Custom Database Deployment

Administering a Database Deployment

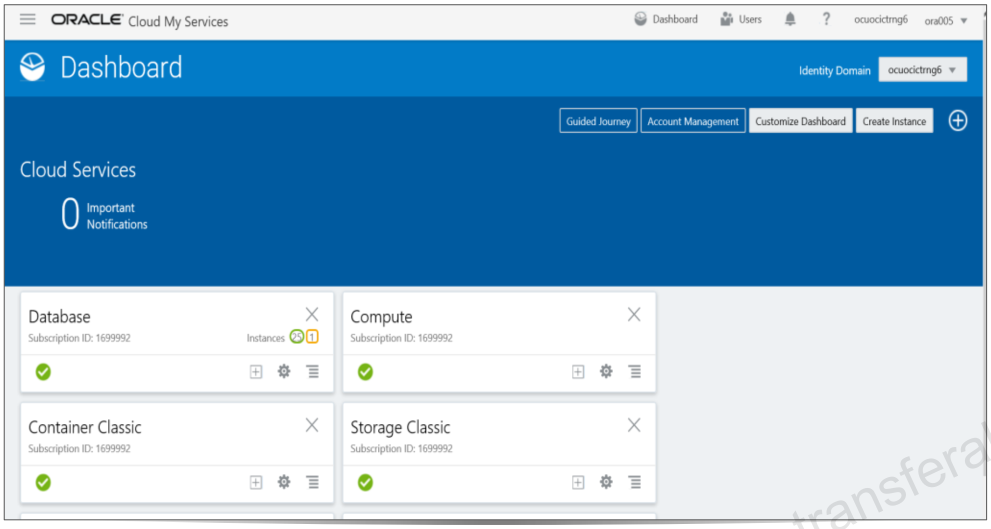
Objectives

After completing this lesson, you should be able to:

- Use the consoles
- Configure connections to the compute node
- Administer database deployment users and privileges
- Administer database deployment database users and privileges
- Administer storage
- Clean up log and diagnostic files
- Stop and restart the database deployment
- Patch your database deployment
- Upgrade a database deployment
- Delete a database deployment

ORACLE

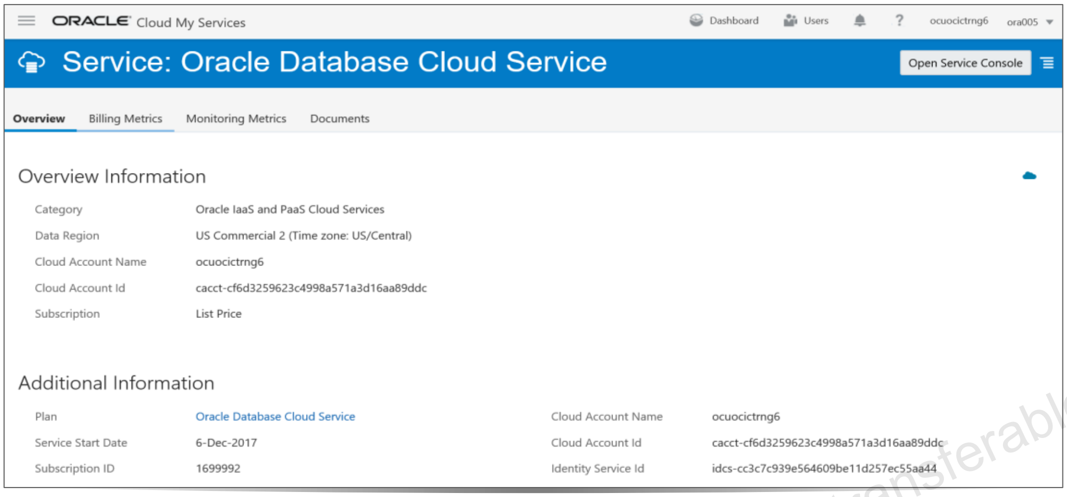
Using the Oracle Cloud My Services Dashboard



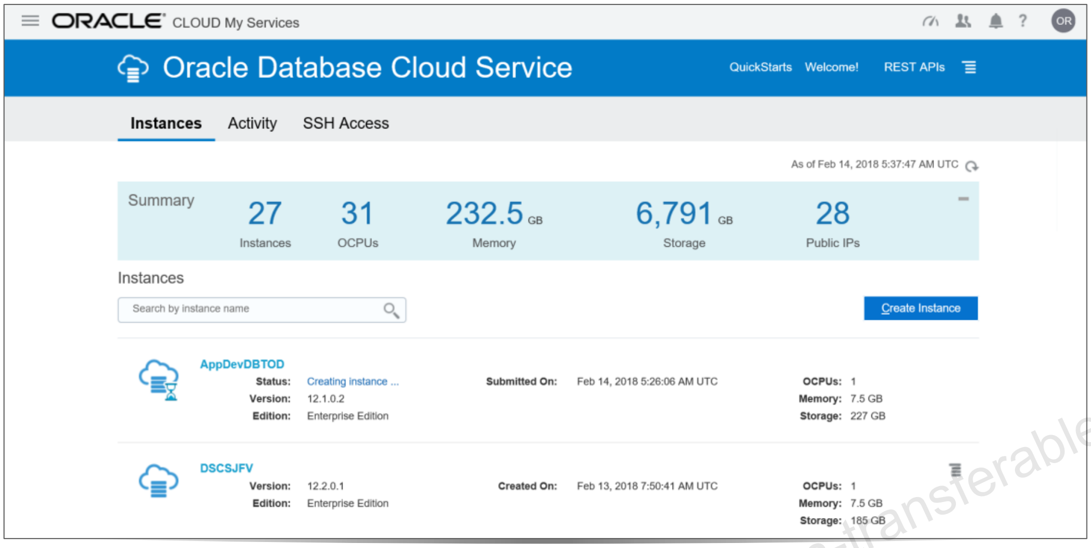
The My Services Dashboard console displays a summary of all subscription services.

See "Exploring the My Services Dashboard" in *Managing and Monitoring Oracle Cloud* for the most current information.

Viewing the Service Details



Using the Database Cloud Service Console



Configuring Connections to the Compute Node

To log in to the compute node of the database deployment:

1. Obtain the public IP address of the database deployment's compute node.
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 of the SSH key pair. If the private key was generated with PuTTYgen (PuTTY Key Generator), then 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.
   - **On UNIX/Linux:** Use the ssh-keygen utility to configure connections to the database deployment VM. Provide the private key file of the SSH key pair.

ORACLE

When you create a database deployment, Database Cloud Service creates compute nodes to host the database, using computing, storage and networking resources provided by various Oracle Cloud infrastructure services.

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.

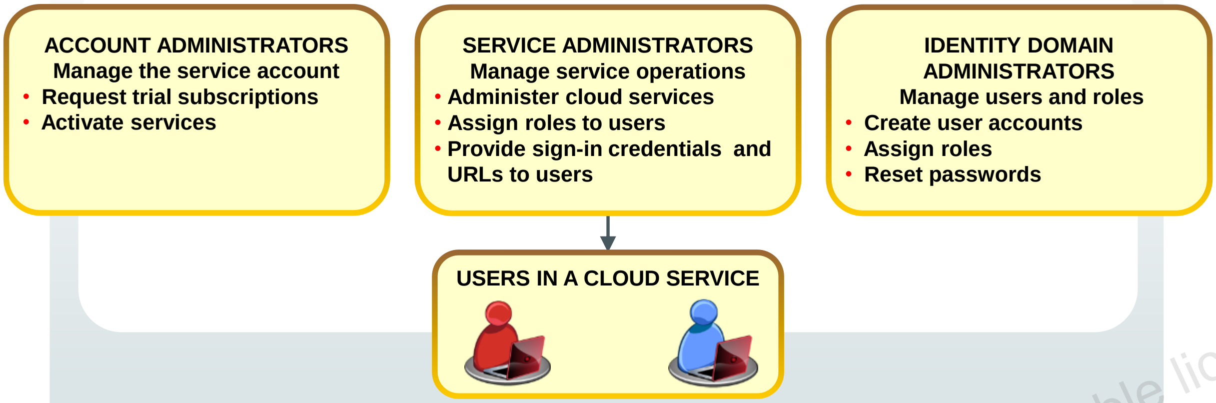
To access other ports, either enable network access to the port or create an SSH tunnel to the port. More information is provided later in this lesson.

You can create an SSH tunnel to the compute node port, using the ssh utility on Linux, or the PuTTY program on Windows. You can download PuTTY from http://www.putty.org/.

Viewing Detailed Information for a Database Deployment

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3302

Oracle Cloud User Roles and Privileges



See Oracle Cloud User Roles and Privileges in *Getting Started with Oracle Cloud* for additional information.

ORACLE

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.

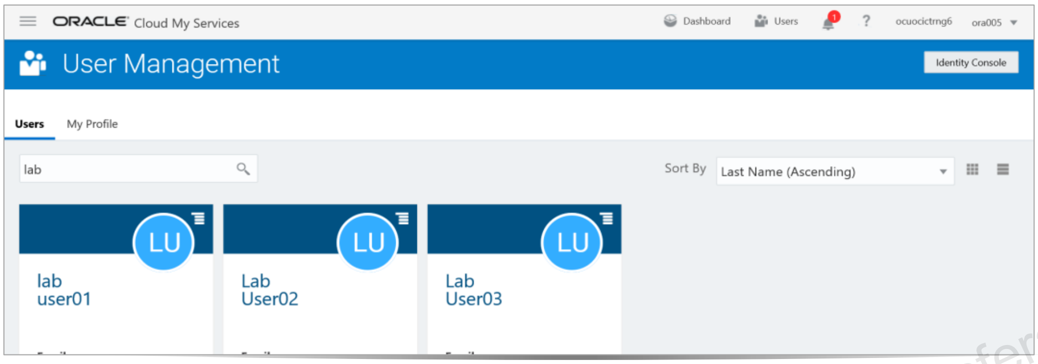
Service administrators (or the DBCS database administrator) can assign and remove roles only for the users of the services that they manage. Because service administrators cannot add users or roles, the users and roles must already be in the system before service administrators can assign a specific role to a user.

Oracle Cloud User Roles and Privileges

http://www.oracle.com/pls/topic/lookup?ctx=E38328-01&id=CSGSG117

Administering Users, Roles, and Privileges

- Administer Cloud Services users by accessing the Users page.
- Cloud Services users are different from Oracle Database users.



See Creating a User and Assigning a Role in *Getting Started with Oracle Cloud* for details.

ORACLE

The Users page may be selected from various consoles. On the Users page, you can add individual users and configure the roles (privileges) for each user.

User privileges are controlled through assigned roles. The roles assign privileges in a granular fashion and each role has specific privileges.

The Import button allows you import a list of users as a CSV file (comma delimited).

The Users page provides administration for:

- **SFTP Users for file transfers:** 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:** Used to configure Single Sign-On so your users can use their company credentials to log in to all applications, including Oracle Cloud applications.
- **OAuth Administration:** Used to manage client access to Oracle Cloud APIs using the OAuth 2.0 protocol.
- **My Profile:** Each user can modify basic user information, reset the password, change the roles, and remove the user.

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

Creating a User and Assigning a Role

http://www.oracle.com/pls/topic/lookup?ctx=E38328-01&id=CSGSG178

Managing Compute Node Users

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

OS User	Authorization
`opc`	Authorized to log in to the compute node Authorized to run `root` commands Can use `sudo -s`
`oracle`	Authorized to log in to the compute node Not authorized to run `root` commands
`root`	Not authorized to log in to the compute node

ORACLE

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

- **`opc`:** The `opc` user is authorized to run `root` commands. Certain operations such as database backup and recovery, or compute node reboot require the use of the `sudo` command. Connect as `opc` to perform these operations.
- **`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, connect as the `opc` user and use the `sudo` command.

Adding Compute Node Users

You can create additional Linux users to log in to the database deployment compute node:

1. Log in to your database deployment compute node as the `opc` user, start a root-user shell, 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` file.

```
AllowUsers oracle opc user1
```

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

ORACLE

You can create additional Linux users to log in to the compute node of your database deployment. The new users can perform standard OS operations, such as installing and running applications.

Keep in mind that the new users will need to be re-created if the compute node is automatically re-provisioned by Oracle.

Adding Compute Node Users

5. Restart the SSH daemon on your compute node.

```
# /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.

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

Managing Database Users and Privileges

- Manage users in a database deployment database as you manage users in an on-premises database:
  - Grant privileges to users so they can accomplish tasks required for their jobs.
  - 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.
- Do not grant `SYSDBA`, `SYSOPER`, `SYSBACKUP`, or `SYSKM` administrative privileges to users who do not perform system administrative tasks.

You manage users and privileges in a database deployment database just as you would manage users in an on-premises database.

Bear in mind that in an Oracle Database 11*g* database deployment, a non-CDB is created. In an Oracle Database 18c and 12*c* database deployment, a CDB with one PDB is created. User management should be appropriate for each architecture.

Administering Compute Node Storage

- Permanent storage: Storage volume that is a persistent virtual disk
  - Up to nine storage volumes for each database deployment
  - When 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.
- Temporary storage: Storage that is added to a database deployment temporarily for a short period of time, after which it is detached and deleted.

ORACLE

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

The storage volume is used to store:

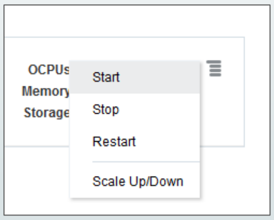
- Operating system files
- Oracle Database binaries
- The data files, control files, redo log files, and all files related to an Oracle Database instance (password file, trace files, alert log, backup files, and so forth)

Operations to manage the storage volume can be easily performed via the Oracle Database Cloud Service console.

You can attend the *Oracle Cloud IaaS: Compute and Storage Fundamentals* course to learn more about storage in Oracle Compute Cloud Service.

Scaling a Database Deployment

Scale the compute shape or storage from the Action menu in the Oracle Database Cloud Service console.



Action	
Scale Up	Select a new compute shape. Add raw storage to the database deployment.
Scale Down	Select a new compute shape.

ORACLE

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

Scaling the Compute Shape

When you scale the compute shape of a database deployment, the deployment is put into Maintenance status during the operation and it is restarted. As a result of the restarting, any resources that you have manually added using the Compute Cloud Service console become detached from the database deployment.

Scaling the compute shape is discussed in more detail in a later lesson.

Scaling Storage

When you scale up the storage for a database deployment, the deployment is put into Maintenance status during the operation and it is restarted. As a result of the restarting, any resources you've manually added to the database deployment using the Compute Cloud Service console become detached from the deployment.

When you scale up the storage for a database deployment, a Compute Cloud Service storage volume is created and attached to the deployment. This storage volume remains attached and available to the deployment even after it is restarted or is stopped and then started. Also, the storage volume exists until you delete the database deployment, at which time the storage volume is also deleted.

Cleaning Up Log and Diagnostics Files

- Log and diagnostics files consume space on the compute node of the database deployment.
- Automatic clean up:
  - The `/var/opt/oracle/cleandb/cleandblogs.pl` script runs weekly as a `crontab` job to archive key files and remove old log and diagnostic files.
  - The script uses the `cleandblogs.cfg` configuration file to determine how long to retain each kind of log or diagnostic file.
  - Edit parameters in the `cleandblogs.cfg` configuration file to change the default retention periods.
- Manual clean up: Execute the `/var/opt/oracle/cleandb/cleandblogs.pl` script as the oracle user.
- See Managing the Log and Diagnostic Files on Database Cloud Service in *Using Oracle Database Cloud Service* for additional information.

ORACLE

Because log and diagnostics files consume space on the compute node of the database deployment, an important administrative task is to ensure that the system does not run out of file storage space.

The `/var/opt/oracle/cleandb/cleandblogs.pl` script simplifies this task by archiving key files and removing old log and diagnostic files. It runs weekly as a crontab job. It uses a configuration file named `cleandblogs.cfg` to determine how long to retain each type of log or diagnostic file. You can edit this file to change the default retention periods.

Some of the parameters and their default retention values are:

- `AlertRetention`: 14 days retention
- `ListenerRetention`: 14 days retention
- `AuditRetentionDB`: Database audit (`*.aud`) 14 days retention
- `TraceRetention`: Trace file (`*.tr*` and `*.prf`) 7 days retention

Managing the Log and Diagnostic Files on Database Cloud Service

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-AED2A8A5-EE13-4D04-8121-EA0E379AA1BD

Stopping and Restarting a Database Deployment

Use the Oracle Database Cloud Service console to:

- Stop the compute node associated with a database deployment
- Restart the compute node associated with a database deployment

See Stopping, Starting, and Restarting Database Deployments in *Using Oracle Database Cloud Service*.

From the Oracle Database Cloud Service console, you can:

- Stop the compute node associated with a database deployment. After stopping the compute node:
  - There is no access to the compute node or database.
  - No management operations can be performed except starting or deleting the database deployment.
- Restart the compute node associated with a database deployment.

Stopping, Starting, and Restarting Database Deployments

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-20D20D45-D1D3-4789-AA70-22241168D78F

Patching a Database Deployment

- Use the Oracle Database Cloud Service console. See Applying a Patch in *Using Database Cloud Service*.
- Use the `dbaascli` commands:
  1. Check prerequisites. See Checking Patch Prerequisites.
     ```
     # dbaascli dbpatchm --run -prereq
     ```
  2. Back up the database deployment.
  3. Patch the database deployment. See Applying a Patch.
     ```
     # dbaascli dbpatchm --run -apply
     ```

ORACLE

Apply a patch to your database deployment when patches are required:

- Database PSU patch
- DBCS Tooling (rpm update)
- OS patch

You can view a list of patches to apply to your DBCS instance, in the Available Patches section of the Oracle Database Cloud Service console.

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

Applying a Patch

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3422

Checking Patch Prerequisites

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-C72A47FE-419F-4E1A-A8F6-5E1B8AB1F143

Upgrading a Database Deployment: Overview

To upgrade to a new database version:

1. Create a new database deployment, specifying the desired Oracle Database release version.
2. Select a method such as performing an export, creating a backup, or unplugging a PDB to transport the data from your original database deployment.
3. Transport the data (such as an export dump file, database files, and xml file) from the original database deployment compute node to the new database deployment compute node.
4. Use the method that corresponds to the method chosen in step 2 to load the data into your new database deployment.
5. Delete the original database deployment.

If you want to use the features of a new version of Oracle Database, you can upgrade your existing database deployment to a new database version.

To upgrade to a new version, you must create a new database deployment using the version of Oracle Database you want to upgrade to. Then you must move the data from your original database deployment to your new database deployment.

Detailed information on ways to migrate data is provided later in this course.

Upgrading a Database Deployment: Methods

Source Database Version	Target Database Version	Method
11.2	12.1 or 12.2 or 18c	1. Create a 12.1 or 12.2 or 18c database deployment. 2. Export data from source database and import into target database. 3. Delete the 11.2 database deployment.
12.1	12.2 or 18c	Same method as upgrading 11.2 to 12.2. OR 1. Create a 12.2 or 18c database deployment. 2. Unplug PDBs from the 12.1 database and plug PDBs into the 12.2 or 18c database. 3. Delete the 12.1 database deployment.
12.2	18c	1. Same method as upgrading from 12.1 to 12.2

ORACLE

The table describes the method that can be used to upgrade an 11.2 database deployment to 12.1, 12.2, or 18c, and to upgrade a 12.1 database deployment to 12.2 or 18c.

Export and import can be used in all cases. If you are upgrading from 12.1 to 12.2 or 18c, then you can also use the unplug/plug method for the PDBs in the database.

Detailed information on migration methods is provided in later lessons.

Deleting a Database Deployment

- Delete a database deployment by using the:
  - Oracle Database Cloud Service console. See Deleting a Database Deployment.
  - `oracle-dbcs-cli` utility. See The Delete Subcommand.
- When you delete the database deployment:
  - The database deployment terminates.
  - The Oracle database is dropped.
  - The storage is removed.
  - The compute node is deallocated.
  - All backups to cloud storage are deleted (optional).

You can delete a database deployment by using the Oracle Database Cloud Service console or the `oracle-dbcs-cli` utility.

Deleting a Database Deployment

http://www.oracle.com/pls/topic/lookup?ctx=E38328-01&id=CSDBI3305

The Delete Subcommand

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=GUID-D085E99F-259D-40DA-BBCD-E77C113D5111__DBAAS_ORACLE-DBCS-CLI_DELETE

Summary

In this lesson, you should have learned how to:

- Use the consoles
- Configure connections to the compute node
- Administer database deployment users and privileges
- Administer database deployment database users and privileges
- Administer storage
- Clean up log and diagnostic files
- Stop and restart the database deployment
- Patch your database deployment
- Upgrade a database deployment
- Delete a database deployment

ORACLE

Practice 4: Overview

- 4-1: Using the Consoles
- 4-2: Connecting to the Compute Node
- 4-3: Adding Compute Node Users
- 4-4: Managing Database Users and Privileges
- 4-5: Scaling Up Storage

5

Backing Up and Recovering

ORACLE

Objectives

After completing this lesson, you should be able to:

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

Backing Up and Recovering Database Deployments

- Database Cloud Service provides a backup feature that backs up:
  - The database
  - Database configuration files
  - Grid Infrastructure configuration files (on deployments hosting an Oracle RAC database)
  - System and cloud tooling files
- The backup feature relies on the following, which are installed in the database deployment:
  - System utilities
  - Oracle Database utilities
  - Oracle Database Backup Cloud Service

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

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

Using Utilities to Back Up and Recover the Database Deployment

- Utilities available with the database deployment to support backup and recovery operations based on the backup configuration selected:

Selected Configuration	Utility
No automatic backup	Recovery Manager (RMAN) for backup and recovery
Automated backup	`bkup_api` for backups `raccli` for RAC backups `dbaascli` for recovery

- Connect as the `opc` user to execute the `/var/opt/oracle/bkup_api` or `dbaascli` utility.

ORACLE

A database deployment includes the following utilities that can be used to perform backup and recovery operations:

- **Recovery Manager (RMAN):** If you have not configured automatic backups, then you can use RMAN to perform backup and recovery operations.
- **`bkup_api`:** Used to create an on-demand backup of a database deployment
- **`dbaascli`:** Used to perform a variety of life-cycle and administration operations, including recovery.

Choosing a Backup Destination

Backup Destination	Description	Retention
Both Cloud Storage and Local Storage	Backups are created automatically and stored both on local compute node storage and on an Oracle Storage Cloud Service container.	30 days 7 most recent days' backups available on local storage
Cloud Storage Only	Backups are created automatically and stored only on an Oracle Storage Cloud Service container.	30 days
None	No backups are created.	

- Specify the destination for automatically created backups when you create the database deployment
- See About Backing Up Database Deployments on Database Cloud Service in *Using Oracle Database Cloud Service*.

ORACLE

When you create a database deployment, you can specify the location for backups. Backup choices are:

- **None:** No automatic backups are configured if you make this choice.
- **Cloud Storage Only:** Backups are stored on an Oracle Storage Cloud Service container. You can use an existing Oracle Storage Cloud Service container and specify that the container must be created as part of the database deployment creation.
- **Both Cloud Storage and Local Storage:** You can also choose to have backups created on both the Oracle Storage Cloud Service Container and on the local compute node.

If you use the `oracle-dbcs-cli` utility or the REST API for Database Cloud Service to create the database deployment, then you can also select "Local Storage Only." In this case, the backups are only stored on the compute node of the database deployment.

About Backing Up Database Deployments on Database Cloud Service

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-21980FCF-FA0C-4FD8-94DC-7C373CFB4C52

Default Backup Configuration

- Files backed up by default:

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
System files	Full backup of selected system files

- Automatic backups daily at a time between 11 PM (23:00) and 3 AM (03:00), with the specific time set during database deployment creation
- Encryption:
  - Both Cloud Storage and Local Storage: All backups to cloud storage are encrypted; backups of Enterprise Edition databases to local storage are encrypted; backups of Standard Edition databases to local storage are not encrypted.
  - Cloud Storage Only: All backups to cloud storage are encrypted.

ORACLE

The backup configuration created when you choose a destination other than **None** follows a set of Oracle best-practice guidelines as listed in the slide.

Creating an On-Demand Backup

- Click “Backup Now” on the Oracle Database Cloud Service Instance Administration page.
- Command-line interfaces for backups:
  - For single-instance databases, use the `bkup_api` utility.
  - For RAC databases, use the `raccli` utility.
- See Creating an On-Demand Backup in *Using Oracle Database Cloud Service* for details.

ORACLE

You can create an on-demand backup of a Database Cloud Service database deployment from the Oracle Database Cloud Service Instance Administration page.

You can also create on-demand backups by using command-line utilities. Use the `bkup_api` utility to create an on-demand backup on database deployments hosting single-instance databases. Use the `raccli` utility to create an on-demand backup on database deployments hosting Oracle RAC databases.

Creating an On-Demand Backup

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3418

Changing the Backup Destination

Prerequisites:

- Changing from Cloud Storage Only or None: Increase the size of the local storage used for backups
- Changing to Both Cloud Storage and Local Storage or Cloud Storage Only: Requires an Oracle Storage Cloud Service container in your account that is reserved for backups

Basic steps:

1. Connect as the `opc` user to the compute node and start a root-user command shell.
2. Edit the `/var/opt/oracle/ocde/assistants/bkup/bkup.cfg` file to set the parameter values for the backup destination. See and Changing the Backup Configuration to a Different Backup Destination in *Using Oracle Database Cloud Service* for detail.
3. Run the backup assistant with the `bkup.cfg` file.

Important Note: The Database Cloud Service console does not recognize changes to the backup destination and may not display backups correctly.

ORACLE

After creating your database deployment, you can change the backup destination. There are certain prerequisites that must be met, depending on what backup destination you are changing from and what destination you are changing to:

- If you have not been taking backups to local storage and are changing the destination to include local storage, you must increase the size of local storage such that backup storage is 1.7 times the size of your database storage. You can use the **Extend Backup Storage Volume** option of the Oracle Database Cloud Service console's scaling feature to add storage.
- If you have not been taking backups to Cloud storage and are changing the destination to include Cloud storage, you must an Oracle Storage Cloud Service container to use for backups. If you do not have a container, you must create one.

To change the backup destination, edit the `/var/opt/oracle/ocde/assistants/bkup/bkup.cfg` file to set parameter values for the new destination. Then use the `bkup.cfg` file when you execute the backup assistant, `/var/opt/oracle/ocde/assistants/bkup/bkup`.

Important Note: The Database Cloud Service console does not currently recognize changes to the backup destination made by using the `bkup.cfg` file with the backup assistant. Therefore, the console will not reflect the new backup destination and may not display any backups taken, depending on what backup destination change you have made. If the backups are not displayed, then you will not be able to use the Database Cloud Service console to perform recovery.

Changing the Backup Configuration to a Different Backup Destination

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-E7EBDB67-007A-4910-8B5A-A84B5A4AA27E

Customizing the Backup Configuration

Customization	Description	Utility to Use or File to Edit
Backup Configuration Settings	Persistent configuration settings	Recovery Manager (RMAN)
System Files	System files and directories	`/home/oracle/bkup/oscfg.spec` file
Database Configuration Files	Wallet, initialization parameters, network configuration files	`/home/oracle/bkup/dbcfg.spec` file
Retention Period	Backup retention period (days)	`bkup_api` utility
Cycle Period	Backup cycle period (days)	`bkup_api` utility
Frequency	Time that `bkup_api` is run	`/etc/crontab` file

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

To change how the Oracle database is backed up, you use the RMAN utility. You should not use the RMAN utility to change the retention period. Instead use the `bkup_api` utility.

The backup feature provided by Database Cloud Service backs up files and directories specified in configuration files. You can edit the configuration files to change which files and directories are backed up:

- `/home/oracle/bkup/oscfg.spec`: Lists system files and directories to be backed up.
- `/home/oracle/bkup/dbcfg.spec`: Lists Oracle Database files to be backed up, including the file that contains the keystore (wallet), initialization parameters file, and network configuration files.

You can use the `bkup_api` utility to specify:

- The number of days backups should be retained
- The number of days in the backup cycle

The backup feature provided by Database Cloud Service uses the Linux `cron` job scheduler to perform automatic backups. There is an entry in the `/etc/crontab` file specifying when the `bkup_api` utility should run. Edit this entry if you want to change the scheduled time.

Performing Recovery by Using the Console

You can perform the following recovery operations through the Database Cloud Service console:

- Restore from the most recent backup and perform complete recovery. See Restoring from the Most Recent Backup in *Using Oracle Database Cloud Service*.
- Restore from a backup and recover to a specific data and time. See Restoring to a Specific Point in Time in *Using Oracle Database Cloud Service*.
- Restore from a backup and recover to a specific system change number (SCN). See Restoring from a Specific Backup in *Using Oracle Database Cloud Service*.

You can perform different types of recovery operations through the Database Cloud Service console.

Restoring from the Most Recent Backup

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3419

Restoring to a Specific Point in Time

https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbi/recover-specific-point-time.html

Restoring from a Specific Backup

https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbi/recover-specific-backup.html

Performing Recovery by Using the `dbaascli` Utility

Use the `orec` subcommand of the `dbaascli` utility to restore and recover the database:

- Restoring from the most recent backup and performing complete recovery

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

- Restoring from a specific backup and performing point-in-time recovery

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

- Restoring from a specific long-term backup and performing point-in-time recovery

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


To restore from a backup and perform recovery on a Database Cloud Service deployment hosting a single-instance database, you use the `orec` subcommand of the `dbaascli` utility, which is available on the compute node.

The `orec` subcommand must be run with `root` access. Therefore, you need to connect to the compute node as the `opc` user to perform recovery operations.

You can restore from the most recent backup, a specific backup, or a specific long-term backup.

Use the following command to obtain a list of backups, including the backup-tag for each:

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

Use this command to obtain a list of long-term backups, including the backup-tag for each:

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

Summary

In this lesson, you should have learned how to:

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

Practice 5: Overview

- 5-1: Backing Up a Database Deployment
- 5-2: Recovering a Database Deployment

6

Use Case: Create a Database Deployment by Using a Production Database Backup

Objectives

In this solution, you will learn to:

- List the benefits of using Oracle Database Cloud Service
- Describe the Database Service levels, releases, and editions available on DBCS
- Explain authentication on cloud instances
- Describe rapid provisioning of a 18c Database Cloud Service database deployment by using an existing cloud backup

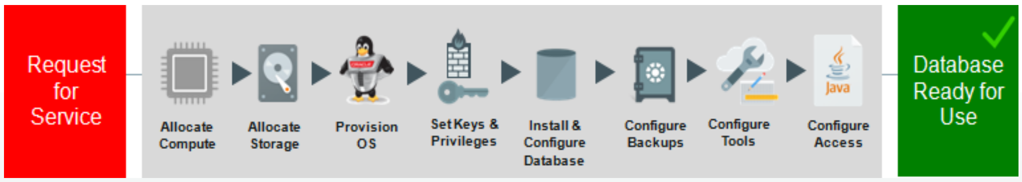
ORACLE

Creating a Database Deployment From a Production Database Backup

Scenario	Suggested Solution
• There is a business requirement to frequently restore production data for UAT purposes. • No additional hardware is available on-premises.	• Use Oracle Database Cloud Service (DBCS) to host nonproduction environments. • In DBCS, the hardware is auto-provisioned based on the service model purchased. • Use the existing cloud backup during provisioning. This will save time on migrating data.
Business Justification: There is no additional hardware available in the data center and the entire solution implementation time is three weeks. A backup of the database is available on Oracle Cloud.	
Oracle Cloud Benefits: • Reduced time and cost of provisioning and management • Self-service and ease of access • Standardization of services for effective database management • Easier adoption of full Oracle capabilities through automation • Elastic, capacity on demand • Subscription pricing	

ORACLE

Prerequisites for Restoring a Database Backup Using Backup Cloud Service



- The on-premises database hosted in the private data center that is to be backed up, should be Little endian format.
- The on-premises database character set should match that present on cloud.
- The on-premises database version should be the same as on DBCS.
- The backup should be taken stored on Oracle Storage Cloud using the Database Backup Cloud Service.
- The database backup on the cloud should be either password encrypted or use a database wallet for security purposes.

ORACLE

The Database Cloud Backup module can be downloaded from OTN using the following link:

http://www.oracle.com/technetwork/database/availability/oracle-cloud-backup-2162729.html

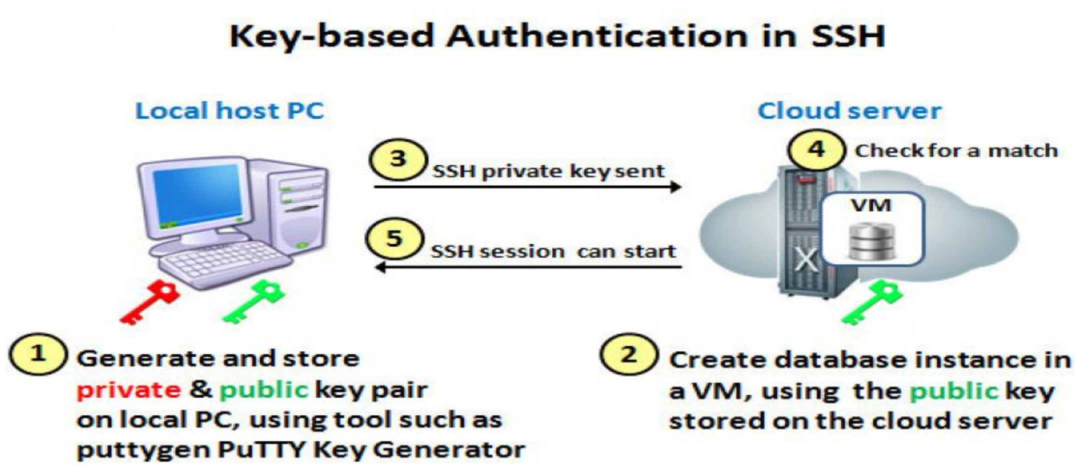
High-level Steps for Rapid Provisioning

- DBCS is an on-demand self-service platform where you can position your database requirements as per your needs. You can select the CPU, memory, and storage required to use your database on cloud.
- A pool of infrastructure resources is assigned from which you will be able to dynamically scale up or scale down your database configurations.
- You can select the software edition according to the application and database requirements, that is, Standard, Enterprise, High Performance, or Extreme Performance.
- Because DBCS is hosted on Oracle Public Cloud, you have the flexibility to access your database from anywhere using your Internet connection.
- The database can be restored from the existing backup on cloud during service creation.
- You have the added advantage of hosting your application on Oracle Cloud and integrating it with DBCS.

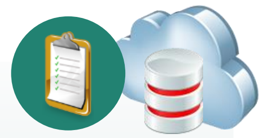
ORACLE

Oracle Database Cloud Service (DBCS) uses SSH to provide secure access to the virtual machine hosting a DBCS instance.

- The public key is provided when a service instance is created.
- The private key is used when connecting to the service instance.



Prerequisites

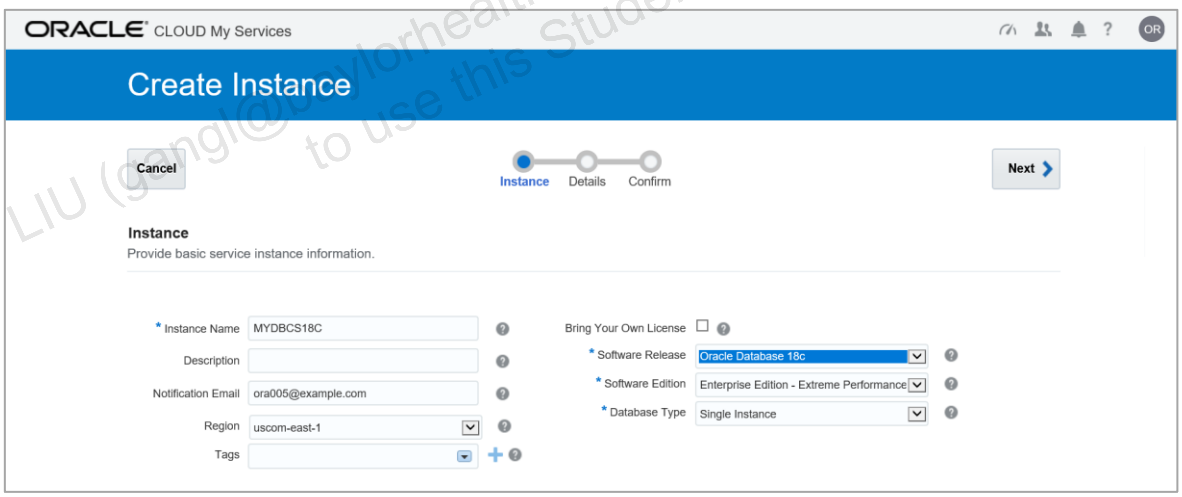


- Database Cloud Service subscription and account with provisioning privileges
- Out-of-the-box one super user account
- Provisioning capabilities to create additional users as per operational needs using predefined roles
- Set of SSH Key Pairs
- Database Backup Cloud Service (needed only for cloud backups)
  Cloud storage is needed for cloud backups.

Demo: Provisioning of a 18*c* Database Instance in DBCS

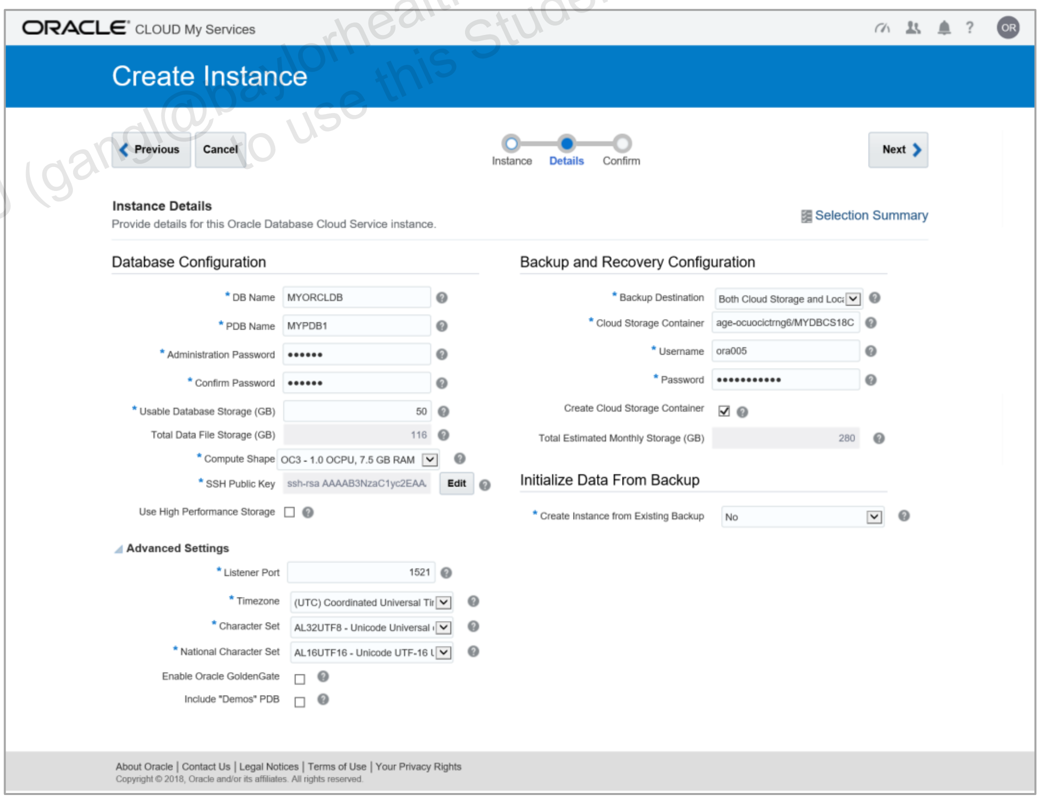
1. Log on to cloud.oracle.com using your Cloud credentials.
2. On the Cloud Dashboard, select the Database Cloud Service console.
3. Click Create New Instance to create a DBCS deployment.
4. DBCS provides you with two types of services: Virtual Image & DBCS. Virtual Image service does not have any auto tooling enabled.
5. For this demo, we will select DBCS - Automated.

ORACLE



Demo: Provisioning of a 18*c* Database Instance in DBCS

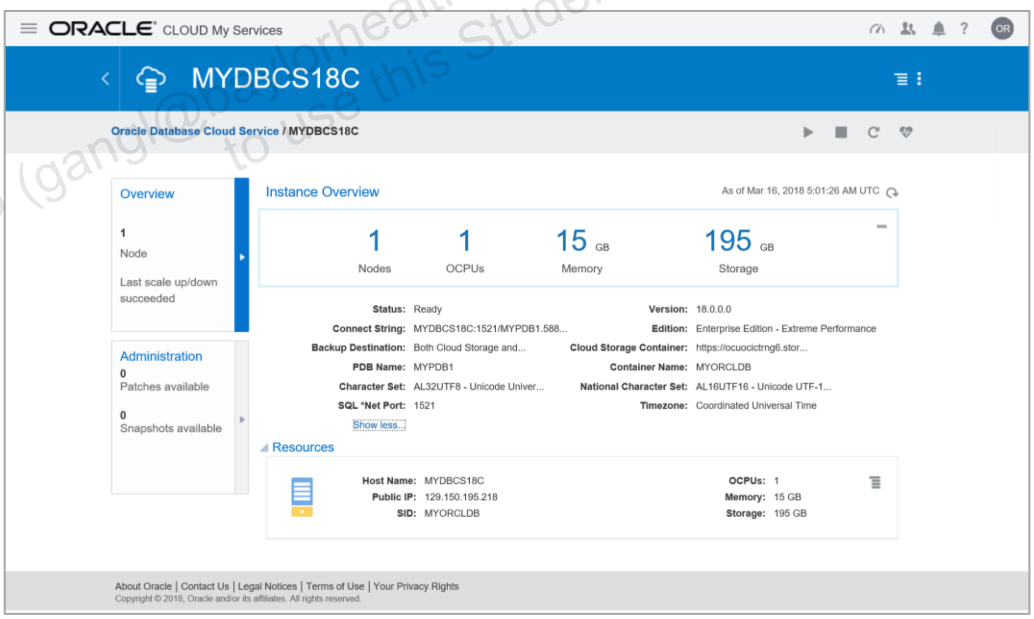
6. Consider the following, when providing information to create the DBCS deployment:
   - Oracle 11*g* Release 2 (11.2.0.4) is the minimum database version available on DBCS. For this demo, we will be provisioning a 18*c* container database (CDB).
   - Select the software edition required for your application according to the feature usage.
   - If the backup of the database is already available on Cloud Storage, then use it to create a backup of the deployment.
   - The Database Configuration section enables you to provision the 18*c* container database as per your requirement. You have the flexibility to select the database names, passwords, database size, and character sets.
   - The backup configuration wizard will guide you to configure automated backups for the provisioned database. We will skip the backup configuration for this exercise and select "None".



Demo: Provisioning of a 18*c* Database Instance in DBCS

7. The Service Configuration section configures the database server as per your requirement. You can select the required memory and CPU from a predefined template used for provisioning the DBCS deployment. Upload the SSH Key pairs generated as a prerequisite for this exercise.
8. After confirming the configuration choices, click Create to provision the DBCS deployment.
9. Once the provisioning completes, you will be able to see the database details on the Database Cloud Service console.
10. For your application connectivity to the database, note the following:
    - Host Name: The public IP address.
    - SQL Net Port: 1521, as per this example.
    - PDB Name: The format is of the type `DB1.<domain>.oraclecloud.internal`

ORACLE



Summary

In this solution, you should have learned to:

- List the benefits of using Oracle Database Cloud Service
- Describe the Database Service levels, releases, and editions available on DBCS
- Explain authentication on cloud instances
- Describe rapid provisioning of a 18*c* Database Cloud Service database deployment by using an existing cloud backup

Practice 6: Overview

There are no practices for this lesson.

7

Overview of Oracle Cloud Security

ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe Oracle Cloud Service security
- Ensure physical and operating system security
- Configure users' authentication and their roles
- Protect and secure data in the database deployment compute node and the database
- Ensure data security over the network

ORACLE

Cloud Security Guidelines

- 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 must perform operations on the compute node.
- Isolate sensitive data in distinct tenants or PDBs.

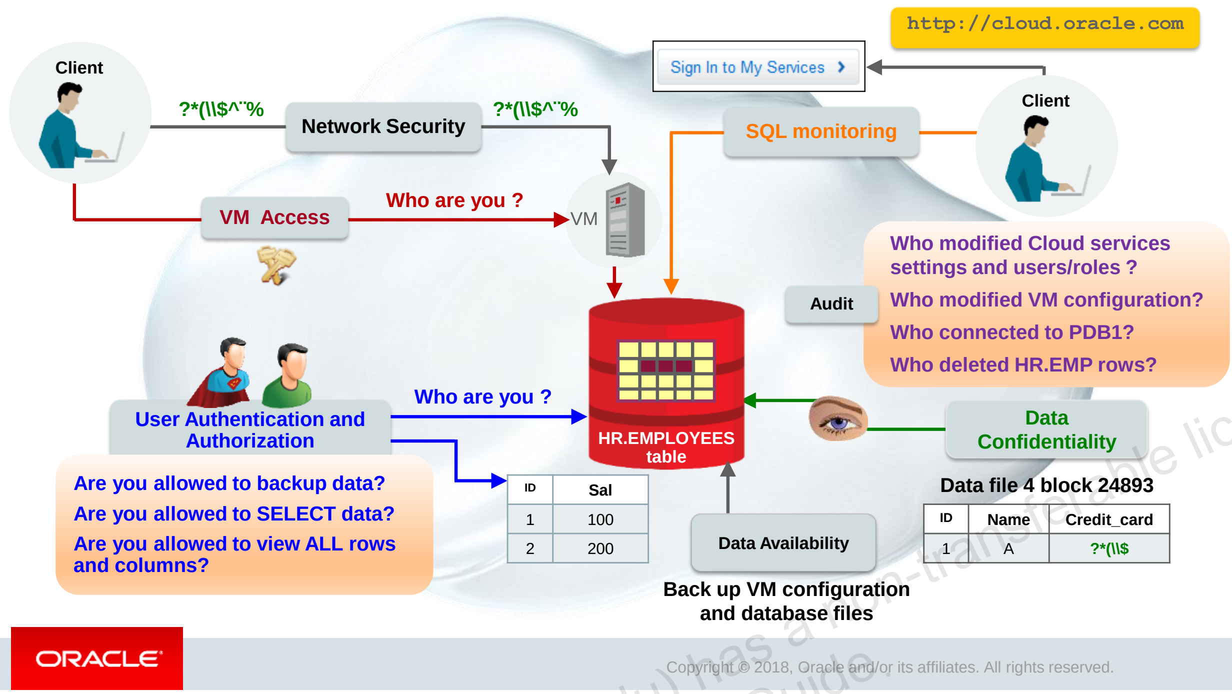
ORACLE

Security best practices for on-premises or Cloud environment include people, processes, and products.

As you prepare to move your on-premises environment to Oracle Database Cloud Service, consider the following guidelines:

- 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.

Enforcing Security in a Database Deployment



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 (Enterprise Manager Database Express, Oracle Application Express, Oracle SQL Developer Web, 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

Physical and Operating System Security of the Compute Node

- Physical security of the compute node:
  - Secured by Oracle
  - Customers have no access to the compute node
- Operating system users and credentials:
  - `oracle`: Minimal privileged Linux user (normal shell access)
  - `opc`: Privileged Linux user (root commands access)
- Operating system security:
  - SSH access
  - OS commands
  - 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`, and `dbca`
  - Access to utilities such as `dbaascli`, `bkup_cli` as `opc` OS user

ORACLE

Physical Security of the Compute Node

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

Operating System Credentials to Access the Compute Node

The following users are created when the database deployment is created

- **`oracle`:** Minimal privileged Linux user (normal shell access)
- **`opc`:** Privileged Linux 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.

Operating System Security

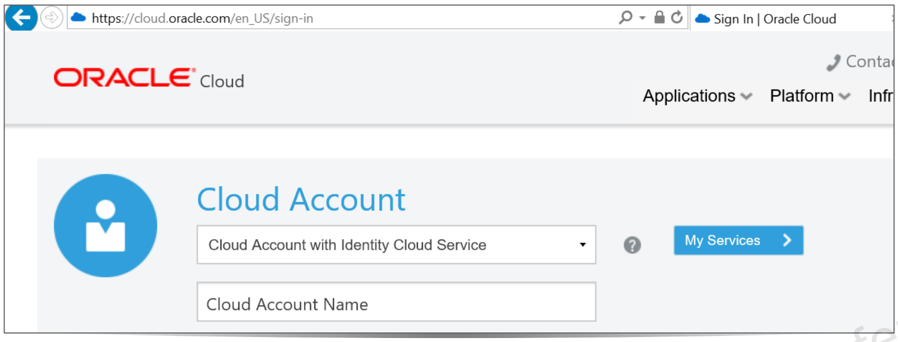
On the database deployment compute node, the `opc` OS user can have `root` privilege. The user can perform the following operations:

- 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.

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

User Authentication: Services and Compute Node Access

Sign in to My Services to use the services you have subscribed to.



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.

As an example, if a user wants to use Enterprise Manager Database Express to monitor his database deployment database, the customer logs in to the Cloud environment to access the services he has subscribed to. The user must sign in to the “My Account” application providing the Oracle.com account credentials.

As another example, the user wants to export data from his database deployment database. The user first logs in to 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.

User Authentication: Database Access

OS User

Database Deployment

Database Instance

```
[oracle@MYDBCS ~]$ . oraenv
ORACLE_SID = [MYORCL] ?
The Oracle base has been set to /u01/app/oracle
[oracle@MYDBCS ~]$
[oracle@MYDBCS ~]$ sqlplus system
SQL*Plus: Release 12.2.0.1.0 Production on Mon Feb 13 21:35:36 2017
Copyright (c) 1982, 2018, Oracle.  All rights reserved.
Enter password:
Last Successful login time: Mon Jan 22 2018 20:41:41 +00:00
Connected to:
Oracle Database 12c EE Extreme Perf Release 12.2.0.1.0 - 64bit Production
SQL>
```

ORACLE

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.

Summary

In this lesson, you should have learned how to:

- Describe Oracle Cloud Service security
- Ensure physical and operating system security
- Configure users' authentication and their roles
- Protect and secure data in the database deployment compute node and the database
- Ensure data security over the network

Practice 7: Overview

- 7-1: Connecting to the Compute Node and Database

8

Configuring Network Access to a Database Deployment

ORACLE

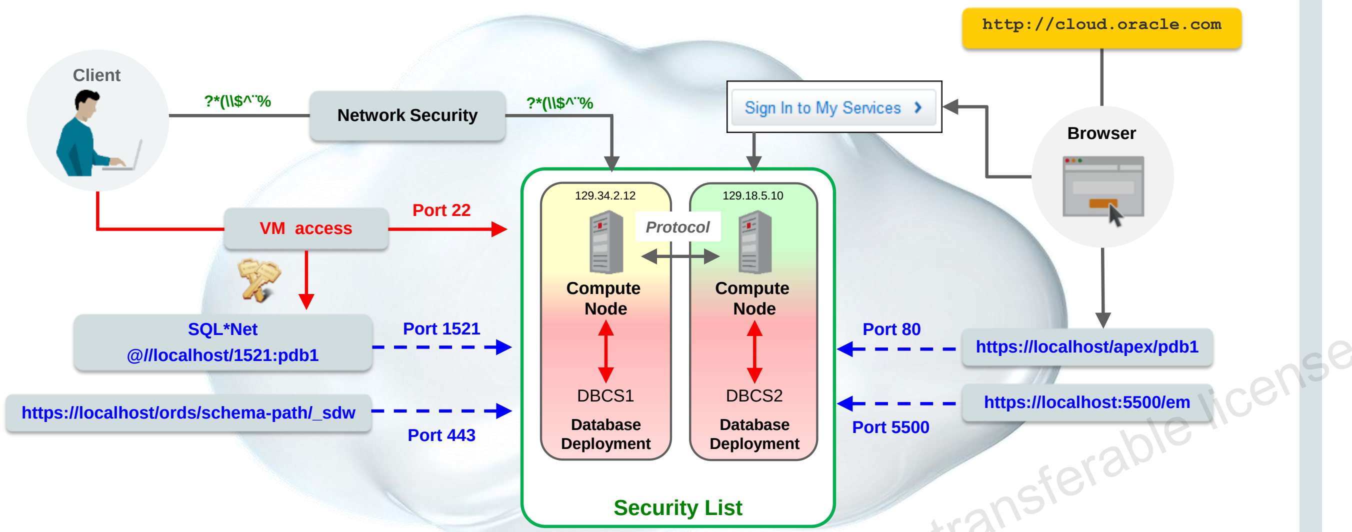
Objectives

After completing this lesson, you should be able to:

- Configure network access to a database deployment
- Configure network settings through the Oracle Compute Cloud Service
- Manage access by tools and applications to the database deployment

ORACLE

Describing Network Access to the Compute Node and Database



ORACLE

When you create a Database Cloud Service (DBCS) database deployment, by default it does not allow access from any other DBCS database deployment or external host. You can use network groups and access rules to control network traffic among DBCS database deployment and also between specific DBCS database deployments and external hosts.

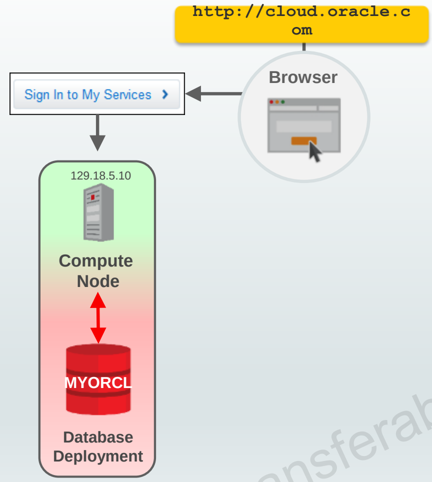
To enable unrestricted communication among some of your DBCS database deployments (for example, to enable all the instances hosting your development environment to communicate with each other), you can create a network group and add all the DBCS database deployments to that network group. When you add a DBCS database deployment to a network group, the DBCS database deployment can communicate with all the other DBCS database deployments in the same group.

By default, the DBCS database deployments in the network group are isolated from hosts outside the group. You can override these default access settings of network groups by creating access rules. Each access rule defines a specific communication path, which consists of a source, a destination, and a protocol-port combination over which communication is allowed.

You can enable secure access to DBCS compute node and database instance from remote hosts by using SSH.

Creating Security Lists

- When the database deployment is created, the compute node does not allow access from any other database deployment compute node or external host.
- Create security lists to control network traffic among database deployment compute nodes.

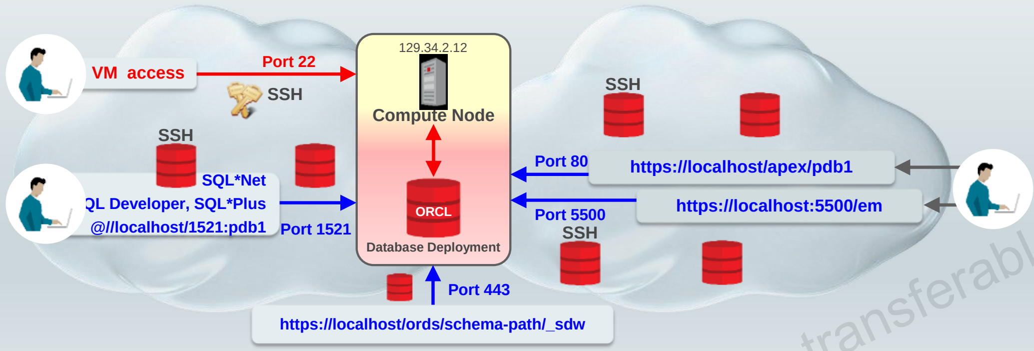


When you create a database deployment, the associated compute node by default does not allow access from any other database deployment compute node or external host.

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

Configuring Connections to the Compute Node

- When a DBCS database deployment is created, network access to the compute node is limited to Secure Shell (SSH) connections on port 22.
- To use tools on other ports:
  - Enable the access rule for the port through the DBCS console
  - Create a tunnel for port forwarding using `ssh`



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.

You can enable the access rule for a port through the Database Cloud Service console. When you enable the access rule, the port remains open until you disable the access rule.

Another option is to create a tunnel for port forwarding using `ssh`. The configured `ssh` process must be running on the consumer side of the tunnel.

Implementing Fine-Grained Control of Network Traffic

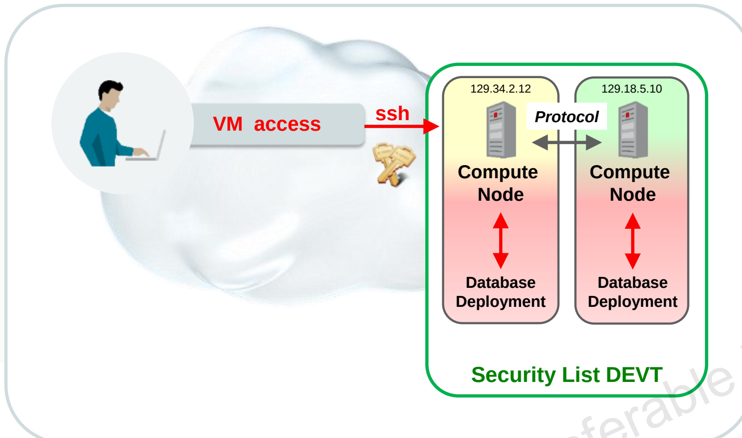
- Control network traffic among database deployment compute nodes in the same security list.
- Control network traffic between database deployment compute nodes in different security lists.
- Control network traffic between database deployment compute nodes and external hosts.

The network settings for the DBCS database deployment are set by default to be secure. Network settings enable you to implement fine-grained control over network access to your DBCS database deployments, both from other DBCS database deployments in the Oracle Cloud as well as from external hosts. By default, the DBCS database deployments in the network group are isolated from hosts outside the group.

To enable unrestricted communication among some of your DBCS database deployments (for example, to enable all the instances hosting your development environment to communicate with each other), you can create a network group. A network group associates a set of instances and policies. Then you can add all the DBCS database deployments to that network group. When you add a DBCS database deployment to a network group, the DBCS database deployment can communicate with all the other DBCS database deployments in the same group but not with the DBCS database deployments of another network group.

Controlling Network Traffic: Compute Nodes in the Same Security List

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



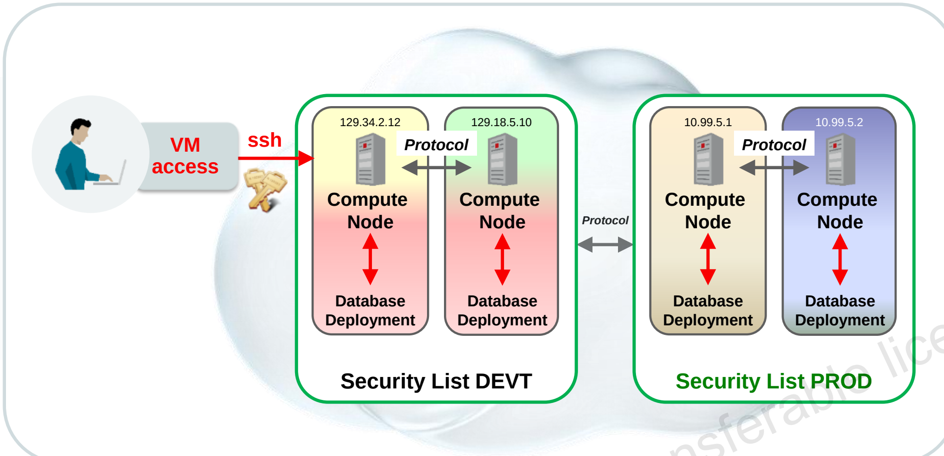
To enable unrestricted communication between your database deployment compute nodes dedicated to development, create a security list and then associate development database deployment compute nodes to the security list.

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.

Controlling Network Traffic: Compute Nodes in Different Security Lists

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.

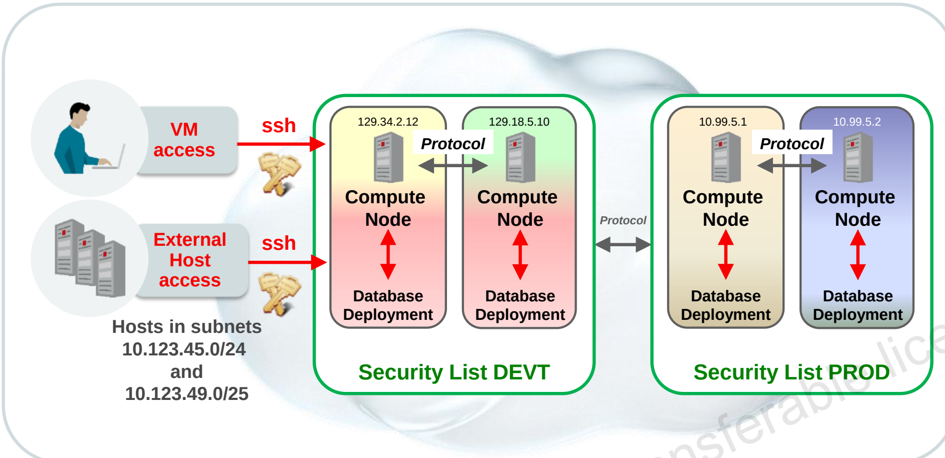
Creating security rules overrides these default access settings of security lists.

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)

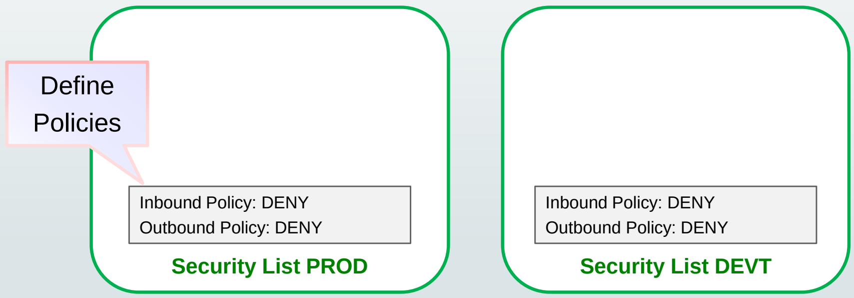
Controlling Network Traffic: Compute Nodes and External Hosts

Control network traffic between database deployment compute nodes and external hosts.



Configuring Network Security: Create Security Lists

1. Create security lists. See Creating a Security List in *Using Oracle Compute Cloud Service*.



ORACLE

The inbound policy controls the flow of traffic into the security list. The default setting for inbound policy is DENY, for Oracle Cloud.

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 security 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 cannot 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.

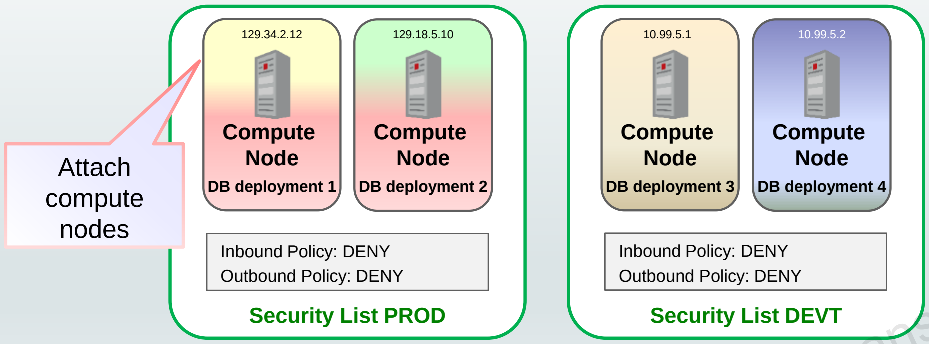
By default, a security list has its inbound policy set to DENY and outbound policy set to PERMIT.

Creating a Security List

http://www.oracle.com/pls/topic/lookup?ctx=stcomputecs&id=GUID-E6901FA2-5286-4F81-A424-82B544F29F6D

Configuring Network Security: Associate Compute Nodes to Security Lists

2. Associate compute nodes to security lists. See Adding an Instance to a Security List in *Using Oracle Compute Cloud Service*.



ORACLE

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

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.

Adding an Instance to a Security List

http://www.oracle.com/pls/topic/lookup?ctx=stcomputecs&id=OCSUG179

Defining 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
- Security Rule = Source (Security List or IP list) + Protocol + Destination (Security List or IP list) where the protocol-port combination allows communication
- See About Security Rules and Creating a Security Rule in *Using Oracle Compute Cloud Service*.

ORACLE

Each security rule consists of a source, a destination, and a protocol-port combination over which communication is allowed.

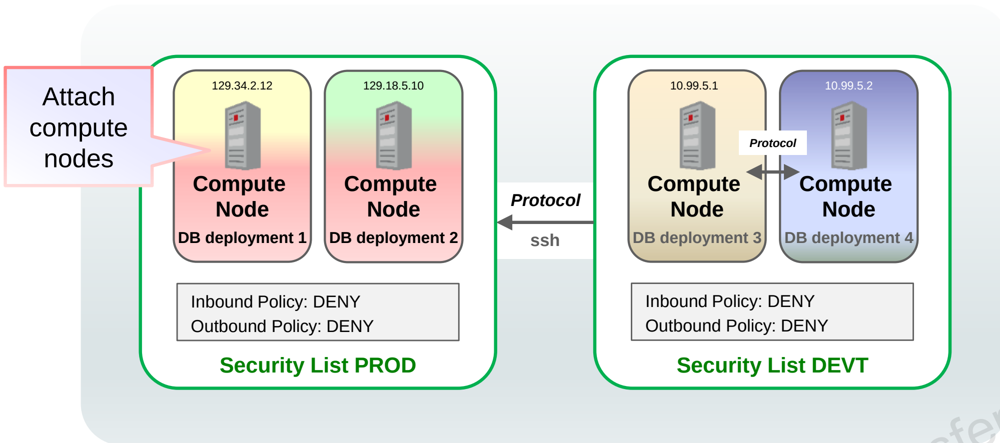
About Security Rules

http://www.oracle.com/pls/topic/lookup?ctx=stcomputecs&id=OCSUG172

Creating a Security Rule

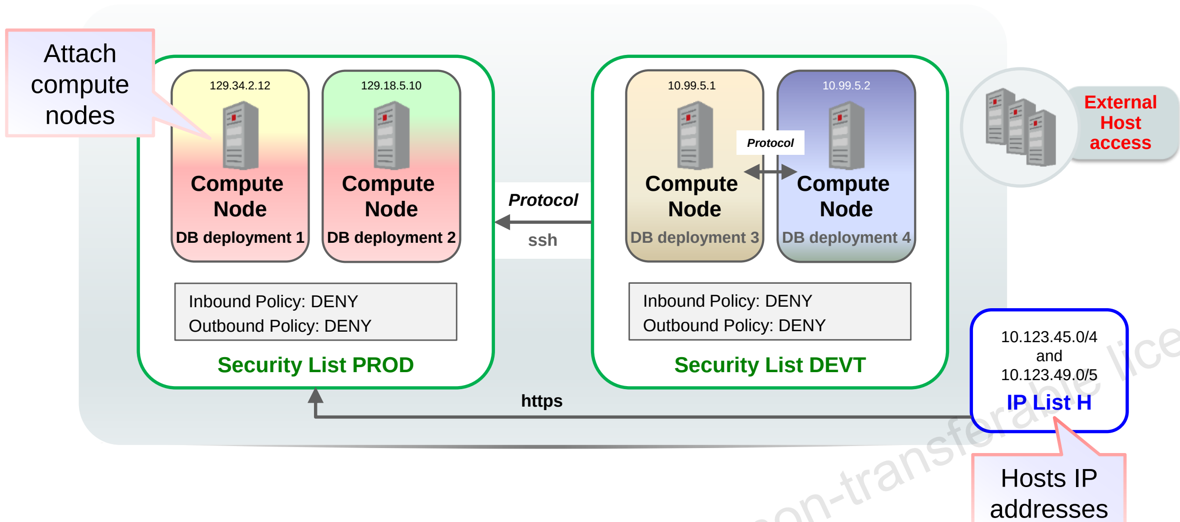
http://www.oracle.com/pls/topic/lookup?ctx=stcomputecs&id=GUID-A3BEB363-C262-4F1E-909D-AA73AF2D65C9

Security Rule Example: DEVT to PROD via SSH Protocol



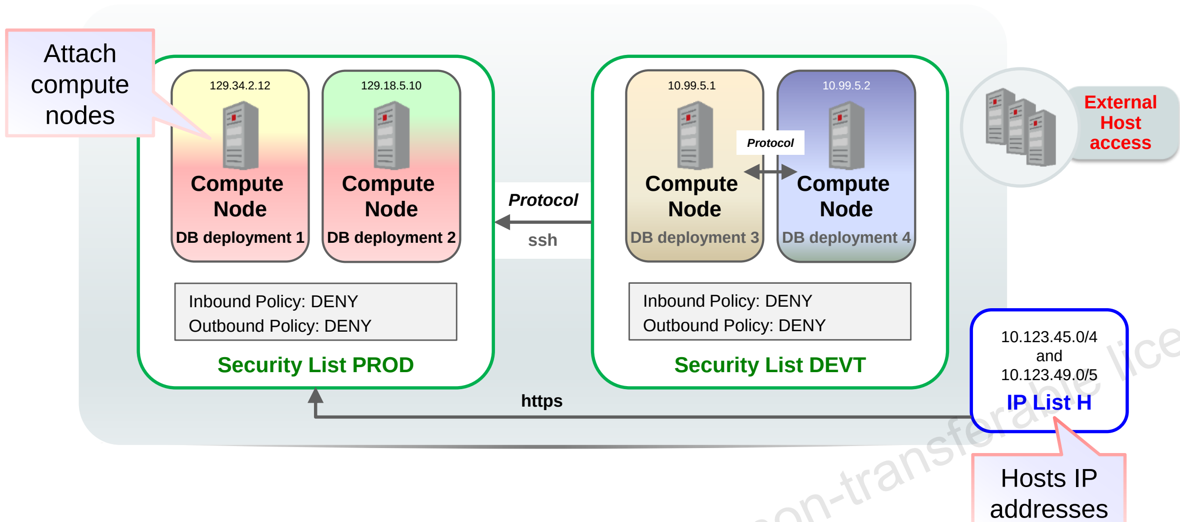
Any development compute node of the DEVT security list can communicate over SSH with any production compute node of the PROD security list. No production compute node can communicate over SSH with any development compute node.

Security Rule Example: External Hosts from Security IP List H to PROD via HTTPS Protocol



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.

Security Rule Example: Security IP List H to DEVT via SSH Protocol



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.

Accessing the Database by Using Enterprise Manager Database Control

- The port for Enterprise Manager Database Control (port 1158) is blocked when the database deployment is first created.
- You can access Enterprise Manager Database Control in either of the following ways:
  - Unblock the port by enabling the `ora_p2_dbconsole` security rule. See Enabling Access to a Compute Node Port in *Using Oracle Database Cloud Service*.
  - Creating an SSH tunnel to port 1158. See Creating an SSH Tunnel to a Compute Node Port in *Using Oracle Database Cloud Service*.

ORACLE

Enterprise Manager Database Control is available for use with an Oracle Database 11*g* database deployment. However, the port is blocked when the database deployment is first created. You can unblock the port by enabling the `ora_p2_dbconsole` security rule. This is a one-time operation. Or you can create an SSH tunnel to port 1158 whenever you want to access Enterprise Manager Database Control.

Enabling Access to a Compute Node Port

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3494

Creating an SSH Tunnel to a Compute Node Port

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3437

Accessing the Database by Using Enterprise Manager Database Express

- The port for Enterprise Manager Database Express (port 5500) is blocked when the database deployment is first created.
- You can access Enterprise Manager Database Express in either of the following ways:
  - Unblock the port by enabling the `ora_p2_dbexpress` security rule. See Enabling Access to a Compute Node Port in *Using Oracle Database Cloud Service*.
  - Creating an SSH tunnel to port 5500. See Creating an SSH Tunnel to a Compute Node Port in *Using Oracle Database Cloud Service*.

ORACLE

Enterprise Manager Database Express is available for use with an Oracle Database 18*c* or 12*c* database deployment. However, the port is blocked when the database deployment is first created. You can unblock the port by enabling the `ora_p2_dbexpress` security rule. This is a one-time operation. Or you can create an SSH tunnel to port 5500 whenever you want to access Enterprise Manager Database Control.

Enabling Access to a Compute Node Port

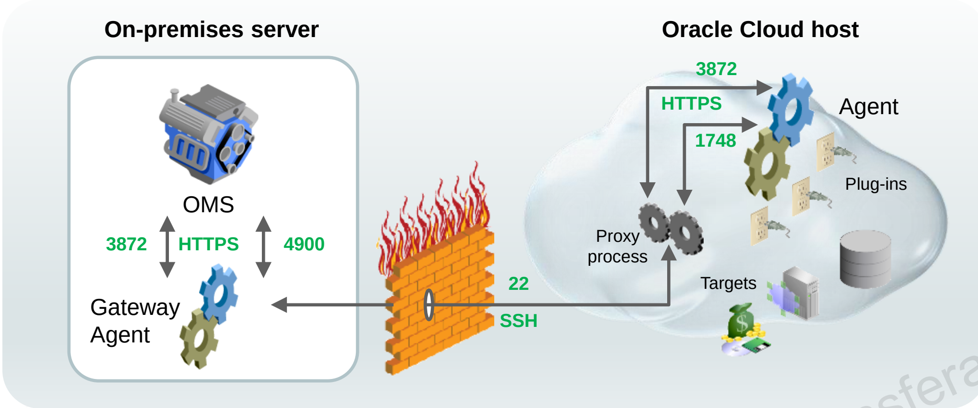
http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3494

Creating an SSH Tunnel to a Compute Node Port

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3437

Accessing the Database by Using Enterprise Manager Cloud Control

You can monitor and manage DBCS with on-premises Enterprise Manager Cloud Control.



See Using Oracle Enterprise Manager Cloud Control with Database Cloud Service in *Using Oracle Database Cloud Service*.

You can use Enterprise Manager Cloud Control 13c to manage and monitor your cloud services and deployments, just as you do the hosts and applications in your own data centers. Cloud Control agents can be deployed to the compute nodes that underpin the Oracle Cloud services. The same plug-ins that manage and monitor your local targets can be used with targets running on the compute nodes. You can install the Enterprise Manager Cloud Control management agent, referred to as the Hybrid Cloud Agent, on the compute node that is deployed for you as part of DBCS.

In order for an Oracle Management Service (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 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.

Using Oracle Enterprise Manager Cloud Control with Database Cloud Service

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-F90BABF1-F0D9-49E4-A649-A94A6A587310

Accessing the Oracle Application Express Console

- The port used by the Oracle Application Express console (port 443) is blocked when the database deployment is first created.
- You can access the Oracle Application Express console in either of the following ways:
  - Unblock the port by enabling the `ora_p2_httpssl` security rule. See Enabling Access to a Compute Node Port in *Using Oracle Database Cloud Service*.
  - Creating an SSH tunnel to port 443. See Creating an SSH Tunnel to a Compute Node Port in *Using Oracle Database Cloud Service*.

ORACLE

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

The port used by the Oracle Application Express console is blocked when the database deployment is first created. You can unblock the port by enabling the `ora_p2_httpssl` security rule. This is a one-time operation. Or you can create an SSH tunnel to port 443 whenever you want to access the Oracle Application Express console.

Enabling Access to a Compute Node Port

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3494

Creating an SSH Tunnel to a Compute Node Port

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3437

Summary

In this lesson, you should have learned how to:

- Configure network access to a database deployment
- Configure network settings through the Oracle Compute Cloud Service
- Manage access by tools and applications to the database deployment

ORACLE

Practice 8: Overview

- 8-1: Opening Ports to a Compute Node
- 8-2: Creating an SSH Tunnel for Port Forwarding

9

Using Oracle SQL Developer Web

Objectives

After completing this lesson, you should be able to do the following:

- Enable a schema for SQL Developer Web
- Access SQL Developer Web
- Use Oracle SQL Developer Web to:
  - Manage and monitor the database
  - Monitor the listener
  - Monitor the operating system

ORACLE

Enabling a Schema for SQL Developer Web

You must enable the database user's schema for SQL Developer Web before accessing the database:

1. Connect as the `opc` user to the deployment's compute node.
2. Start a root-user command shell:

```
[opc@MYDBCS ~]$ sudo -s
[root@MYDBCS opc]#
```

3. Create a text file containing the password of the user whose schema you want to enable.

Enter the password in this file.

```
# touch /home/oracle/password.txt
# chmod 600 /home/oracle/password.txt
```

4. Use the ords assistant to enable the schema.

Change directory for ords assistant.

Run ords assistant.

```
# cd /var/opt/oracle/ocde/assistants/ords
# ./ords -ords_action="enable_schema_for_sdw" \
-ords_sdw_schema="schema-name" \
-ords_sdw_schema_password="/home/oracle/password.txt" \
-ords_sdw_schema_enable_dba="dba-boolean"
```

ORACLE

When using Oracle SQL Developer Web in an Oracle Database Cloud Service database deployment, you sign in as a database user. Before you can do so, however, you must enable the database user's schema for SQL Developer Web.

Before attempting to enable a database user's schema for SQL Developer Web, you need to make sure the database deployment is using cloud tooling version 18.2.3 or later. For instructions on checking and updating the cloud tooling, see Updating the Cloud Tooling by Using the dbpatchm Subcommand at:

https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbi/update-cloud-tooling-using-dbpatchm.html

For a schema in Oracle Database 11*g* or in the CDB (container database) in Oracle Database 12.1 or later, run a command as shown in the slide.

- **schema-name** is the name of the schema you want to enable. If it doesn't exist, it will be created.
- **dba-boolean** is TRUE or FALSE. If you enter TRUE, the schema will be enabled to support the DBA (database administrator) features of SQL Developer Web.

For a schema in a PDB (pluggable database) in Oracle Database 12.1 or later, enter a command of the following form:

```
# ./ords -ords_action="enable_schema_for_sdw" \
-ords_sdw_schema="schema-name" \
-ords_sdw_schema_password="/home/oracle/password.txt" \
-ords_sdw_schema_container="pdb-name" \
-ords_sdw_schema_enable_dba="dba-boolean"
```

Accessing SQL Developer Web

You can access Oracle SQL Developer Web in an Oracle Database Cloud Service database deployment in the following ways:

- Using the Database Deployment's Landing Page
- Using a Direct URL:

```
https://node-ip-address/ords/schema-path/_sdw
```

- Using an SSH Tunnel:

```
https://localhost/ords/schema-path/_sdw
```

ORACLE

The ways to access SQL Developer Web require you to provide the schema path reported by the ords assistant when the database user's schema was enabled for SQL Developer Web access. If you've forgotten this value, use these guidelines to determine it:

- For a schema in Oracle Database 11*g* or in a CDB of Oracle Database 12*c* or later:

  The schema path is the schema name with all letters lowercase and special characters changed to underscores. Multiple special characters in a row are changed to a single underscore. For example, the schema path for the C##CORPDBA1 schema is c_corpdba1.

- For a schema in a PDB of Oracle Database 12*c* or later:

  The schema path is the pdb name, a slash (/), and the schema name with all letters lowercase and special characters changed to underscores. Multiple special characters in a row are changed to a single underscore. For example, the schema path for the ORGDBA1 schema in the HRORG PDB is hrorg/orgdba1.

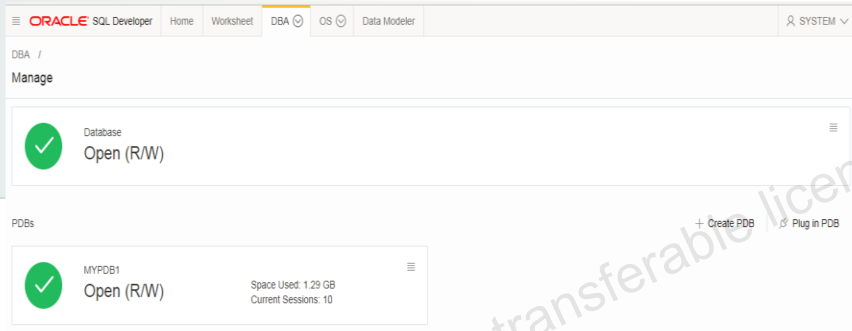
For more details, see:

https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbi/use-sql-dev-web-this-service.html#GUID-ADC70C7F-1703-4FAC-8D4B-ECB799D6E9C9

Managing the Database

You can use Oracle SQL Developer Web to perform the following database management tasks:

- Starting and stopping the database instance
- Administering pluggable databases (PDBs):
  - Create PDBs
  - Clone PDBs
  - Unplug/Plug PDBs
  - Close/Open PDBs
  - Drop PDBs



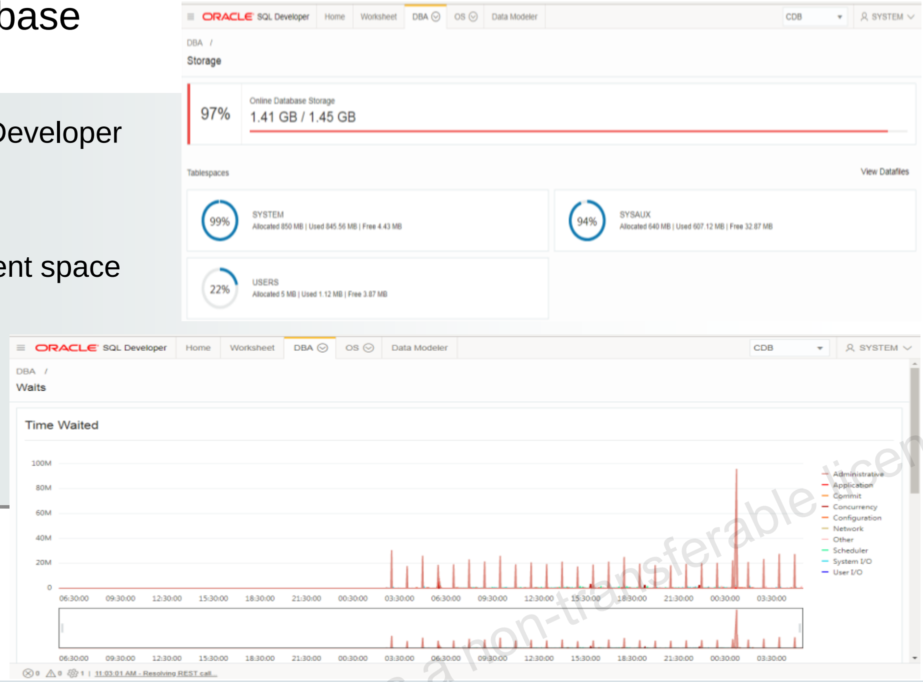
ORACLE

The Manage DBA page enables you to see the status of the database instance, and to start and stop the database instance. In Oracle Database 18*c* databases, the status of the individual pluggable databases (PDBs) is also displayed. On this page, you can create, clone, plug, unplug, or modify PDBs.

Monitoring the Database

You can use Oracle SQL Developer Web to monitor:

- Alert log entries
- Tablespace and segment space usage
- Backup operations
- Current sessions
- Wait events

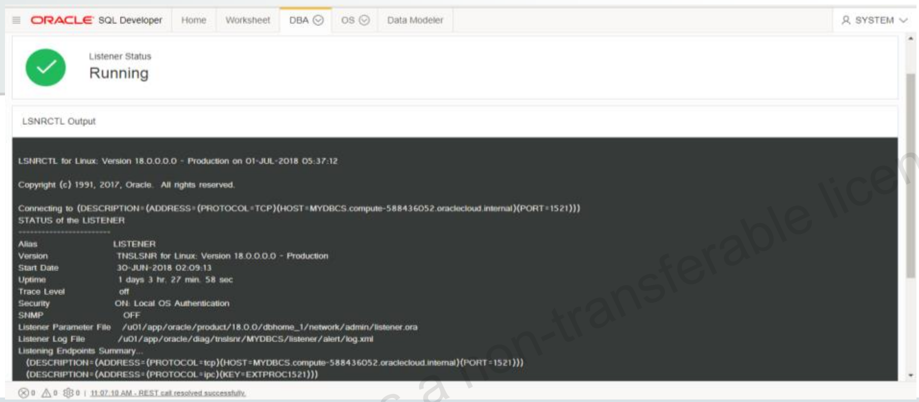


The DBA Storage page shows a breakdown of the storage currently allocated by the database. You can click "show tablespaces" to view the storage information for each tablespace. In Oracle Database 18*c* databases, the information is broken down by container and tablespace.

Monitoring the Listener

You can use Oracle SQL Developer Web to view:

- The status of the listener, including start time
- Protocol addresses on which the listener is configured to listen
- Registered database services



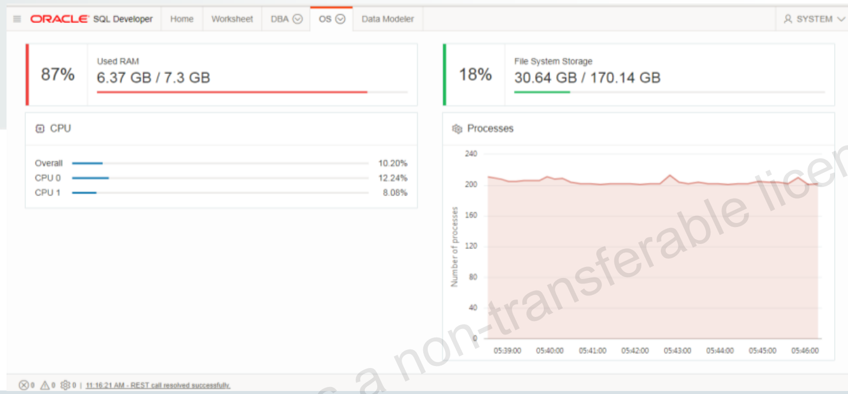
ORACLE

The Database Listener page shows the status of the listener and output from the `lsnrctl status` command.

Monitoring the Operating System

You can use Oracle SQL Developer Web to monitor:

- Global used RAM and the memory usage by process
- CPU usage
- Overall OS storage and the percentage used on each file system
- OS processes



The OS Storage page displays the used and total space available on local mounted file systems. The displayed total space for the file systems does not consider the unusable space in each file system.

Summary

In this lesson, you should have learned how to:

- Enable a schema for SQL Developer Web
- Access SQL Developer Web
- Use Oracle SQL Developer Web to:
  - Manage and monitor the database
  - Monitor the listener
  - Monitor the operating system

ORACLE

Practice 9: Overview

- 9-1: Enabling a Schema for Oracle SQL Developer Web
- 9-2: Accessing Oracle SQL Developer Web
- 9-3: Monitoring the Database Deployment

10

Implementing Database Deployment Security

Objectives

After completing this lesson, you should be able to:

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

ORACLE

Secure Access to Configuration Files on the Compute Node

Connect as the `opc` user and use the `sudo` command to perform operations that require root-user access such as:

- Adding additional `sudoers`
- Editing configuration files

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

ORACLE

Certain operations such as database backup and recovery, or compute node reboot require root-user access. To perform these operations, connect as the `opc` user and use the `sudo` command. You can use this access to:

- Add additional `sudoers`
- Edit OS configuration files such as `/etc/passwd` and `/etc/group`

Backing Up Operating System and Database Configuration Files

- Configuration files contain the list of files backed up during automatic backups:
  - `/home/oracle/bkup/`*`dbname`*`/dbcfg.spec`
  - `/home/oracle/bkup/`*`dbname`*`/oscfg.spec`
- Configuration files are also backed up during automatic backups

The two configuration files providing the list of system and database files to be backed up automatically are also automatically backed up during a backup.

Restricting Access to the Database

1. Connect to the database deployment database (non-CDB in Oracle Database 11*g* or PDB in Oracle Database 18*c* and 12*c*) as a user with privileges to administer users.
2. Create a new user.
3. Grant the new user only the privileges required to complete his job tasks.
   - If the user is a schema, create the schema objects. Do not grant the schema user ANY privileges, including the `CREATE SESSION` privilege.
   - If the user is an end user, grant the necessary object privileges.

ORACLE

Restrict access to your database deployment database as you would an on-premises database.

Protecting Data in the Database Deployment

- Restrict access to the database by requiring:
  - SSH key to access to the compute node
  - Password to log in to the database
- Control access to data in the database by implementing:
  - Privileges
  - Privilege analysis
  - Virtual Private Database (VPD)
  - Oracle Label Security (OLS)
  - Oracle Database Vault
- Protect confidentiality of data in the database by implementing:
  - Data Redaction
  - Data Masking

Tablespace Encryption by Default

- In Oracle Database Cloud Service, user-created tablespaces are encrypted by default.
- Tablespaces created when the database is first created (in the root container, PDB seed, and PDB1) are NOT encrypted.
- The default encryption algorithm is AES128.
- The underlying architecture supporting this feature is transparent data encryption (TDE).
- See Using Tablespace Encryption in Database Cloud Service in *Using Oracle Database Cloud Service*.

ORACLE

All new tablespaces you create in a Database Cloud Service database are encrypted by default. In Oracle Database 18c and 12*c* database, the tablespaces in the root (`CDB$ROOT`), the seed (`PDB$SEED`), and the PDB that is created when the database is created are not encrypted.

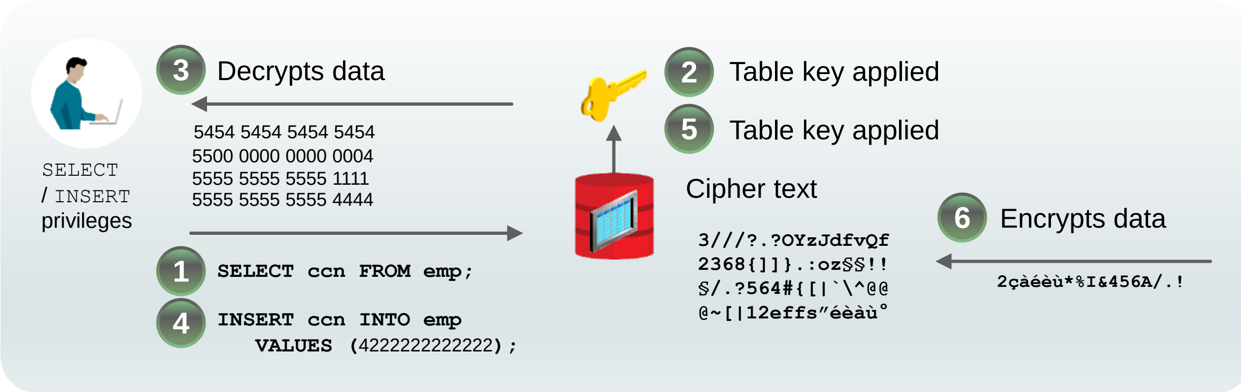
By default, any new tablespaces you create by using the SQL `CREATE TABLESPACE` command or any tool executing the `CREATE TABLESPACE` command are encrypted with the `AES128` encryption algorithm. You do not need to include the `USING 'encrypt_algorithm'` clause to use the default encryption.

Using Tablespace Encryption in Database Cloud Service

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-26B5DB17-E631-4C2D-A172-C02C0D8C1D14

TDE eliminates the ability of anyone who has direct access to the data files to gain access to the data by circumventing the database access control mechanisms. Even users with access to the data file at the operating system level cannot see the data unencrypted. TDE stores the master key outside the database in an external security module, thereby minimizing the possibility of both personally identifiable information (PII) and encryption keys being compromised. TDE decrypts the data only after database access mechanisms have been satisfied.

Transparent Data Encryption (TDE): Overview



- Encrypts data in data files, redo log files, archived redo log files, and backup files
- Encrypts data in memory (only for column encryption)
- Manages keys automatically
- Does not require application changes

ORACLE

Oracle Transparent Data Encryption (TDE) is available with all Database Cloud Service packages, and provides easy-to-use protection for your data without requiring changes to your applications.

TDE allows you to encrypt sensitive data in individual columns or entire tablespaces without having to manage encryption keys. TDE does not affect access controls, which are configured by using database roles, secure application roles, system and object privileges, views, Virtual Private Database (VPD), Oracle Database Vault, and Oracle Label Security. Any application or user that previously had access to a table will still have access to an identical encrypted table.

TDE is designed to protect data in storage, but does not replace proper access control.

TDE is transparent to existing applications. Encryption and decryption occur at different levels depending on whether it is tablespace or column level, but in either case, encrypted values are not displayed and are not handled by the application. For example, with TDE, applications designed to display a 16-digit credit card number do not have to be recoded to handle an encrypted string that may have many more characters.

Auditing: Compute Node Connections and Actions

- Control account connections: `/var/log/secure`
- Updated configuration files: `/etc/passwd` and `/etc/crontab`
- Find files modified within the last two months:

```
$ find /etc -mtime 3 -printf "%u %p"
```

- View command history:

```
$ cat /home/oracle/.bash_history
```

ORACLE

Auditing: Database

- Oracle Database 18*c* and 12*c* database deployments use *mixed mode auditing*, which enables both traditional audit facilities and unified auditing.
- Oracle Database 11*g* database deployments use only traditional audit facilities.
- In Oracle Database 18*c* and 12*c* database deployments, the following predefined unified audit policies are enabled:
  - `ORA_LOGON_FAILURES`: Logon failures
  - `ORA_SECURECONFIG`: System privileges
  - `ORA_ACCOUNT_MGMT`: User creation/alter/drop
  - `ORA_DATABASE_PARAMETER`: Parameter changes and re-creation of the server parameter file (SPFILE)
- `/var/opt/oracle/cleandb/cleandblogs.pl` script executes automatically every week and cleans up database audit files (*.aud) that are older than 14 days

ORACLE

Oracle Database 18*c* and 12*c* database deployment databases use *mixed mode auditing*, which enables both traditional (the audit facility from releases earlier than Release 12*c*) and the Oracle Database 18*c* and 12*c* audit facilities (unified auditing). In unified auditing, the unified audit trail captures audit information from a variety of sources. The unified audit trail, which resides in a read-only table in the `AUDSYS` schema in the `SYSAUX` tablespace, makes this information available in a uniform format in the `UNIFIED_AUDIT_TRAIL` data dictionary view.

Predefined unified audit policies are enabled in the Oracle Database 18*c* and 12*c* database deployments.

Refer to *Oracle Database Security Guide 18c* for details about mixed mode auditing, unified audit policies, and the unified audit trail.

The `/var/opt/oracle/cleandb/cleandblogs.pl` script executes automatically every week and cleans up database audit files (`*.aud`) that are older than 14 days.

Summary

In this lesson, you should have learned how to:

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

Practice 10: Overview

- 10-1: Protecting Data at Rest by Using Encryption
- 10-2: Checking Data Protection in Transit
- 10-3: Cleaning Up Audit Files

11

Use Case: Configuring Network Isolation

ORACLE

Configuring Network Isolation

Scenario	Suggested Solution
• Both production and nonproduction databases have been hosted on Oracle Database Cloud Service. Fine-grained control needs to be implemented over network access to Oracle Cloud Service instances, both from other instances as well as from external hosts. • Access to production database servers should be restricted from internal as well as external hosts. • Grouping of the environment will help in easy management of security rules. Instead of applying security rules to individual servers, they can be applied to security groups.	• Implement fine-grained control over network access to your Oracle Cloud Service instances. • Access control is easy to implement by using the Network console. • Use the Network console to define security lists and security rules in order to define inbound and outbound traffic from instances. • Rule enablement and disablement can be easily controlled by the cloud administrator.

Business Justification: The infrastructure architecture roadmap suggests grouping of the production and nonproduction environments on Oracle Database Cloud Service. Production data sets should not be accessible from nonproduction environments in order to maintain confidentiality. Presently, both production and nonproduction infrastructures have been provisioned in the same domain.

Oracle Cloud Benefits: Fine-grained network access control can be easily implemented by defining a set of security lists and security rules in the Network console. Creation, deletion, enablement and disablement of rules can be directly controlled by the Cloud administrator. In-depth expertise of a network engineer is not essential.

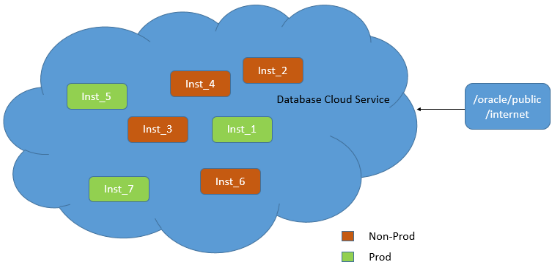
ORACLE

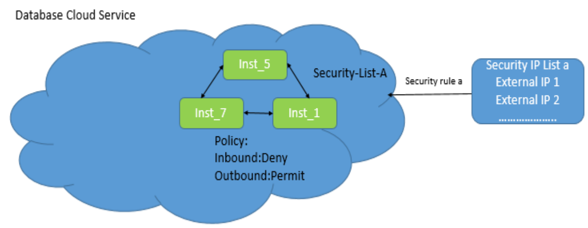
Overview

This use case exhibits how to set up network isolation and fine-grained access of production and nonproduction environments by using Oracle Network console.

Scenario	Suggested Solution
• Both production and nonproduction databases have been hosted on Oracle Database Cloud Service. Fine-grained control needs to be implemented over network access to Oracle Cloud Service instances, both from other instances as well as from external hosts. • Access to production database servers should be restricted from internal as well as external hosts. • Grouping of the environment will help in easy management of security rules. Instead of applying security rules to individual servers, they can be applied to security groups.	• Implement fine-grained control over network access to your Oracle Cloud Service instances. • Access control is easy to implement by using the Network console. • Use the Network console to define security lists and security rules in order to define inbound and outbound traffic from instances. • Rule enablement and disablement can be easily controlled by the cloud administrator.
Business Justification: The infrastructure architecture roadmap suggests grouping of the production and nonproduction environments on Oracle Database Cloud Service. Production data sets should not be accessible from nonproduction environments in order to maintain confidentiality. Presently, both production and nonproduction infrastructures have been provisioned in the same domain.	
Oracle Cloud Benefits: Fine-grained network access control can be easily implemented by defining a set of security lists and security rules in the Network console. Creation, deletion, enablement and disablement of rules can be directly controlled by the Cloud administrator. In-depth expertise of a network engineer is not essential.	

Achieving Network Isolation: Use Case





Architecture: Current State
Database instances are present in a single domain without network isolation. All instances are accessible over public Internet.

Architecture: Desired State
Network isolation between production and nonproduction environments. Instances are accessible from external IP only, based on IP whitelisting.

This diagram shows the following communication paths:

- Instances in Security-list-A comprise production servers and can communicate with each other. They are also accessible from the whitelisted external IPs.
- Instances in Security-list-B comprise nonproduction servers and can communicate with each other. They are also accessible from the whitelisted external IPs.

Demo: High-Level Steps for Implementing Network Isolation of Production and Nonproduction Environments

1. Create Security Lists.
2. Add Instances to Security Lists.
3. Create Security IP Lists.
4. Provide access to Oracle Cloud Service Instances.

ORACLE

Demo: High-Level Steps for Implementing Network Isolation of Production and Nonproduction Environments

1. Creating Security Lists

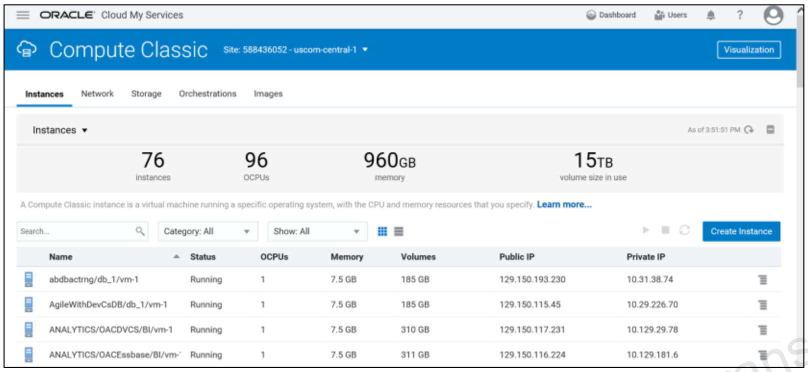
A security list is a group of Oracle Compute Cloud Service instances that you can specify as the source or destination in one or more security rules. The instances in a security list can communicate fully, on all ports, with other instances in the same security list.

When you add an instance to a security list, the inbound and outbound policies of the security list are applicable to that instance.

- The inbound policy controls the flow of traffic into the security list. The inbound policy is always set to “Deny”, so by default traffic from any source outside the security list can't access the instances that are part of the security list.
- 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”.

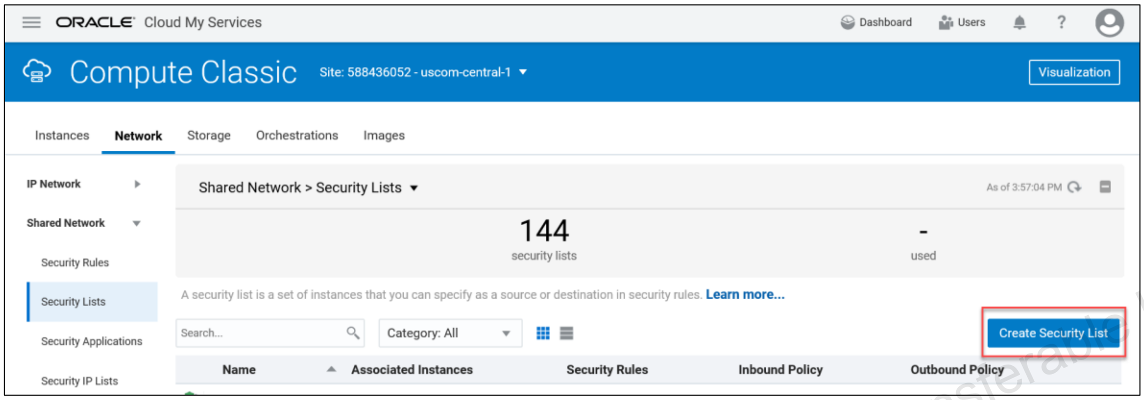
Creating Security Lists

1. Sign in to the Oracle Compute Classic Cloud Service console and navigate to the Network tab.



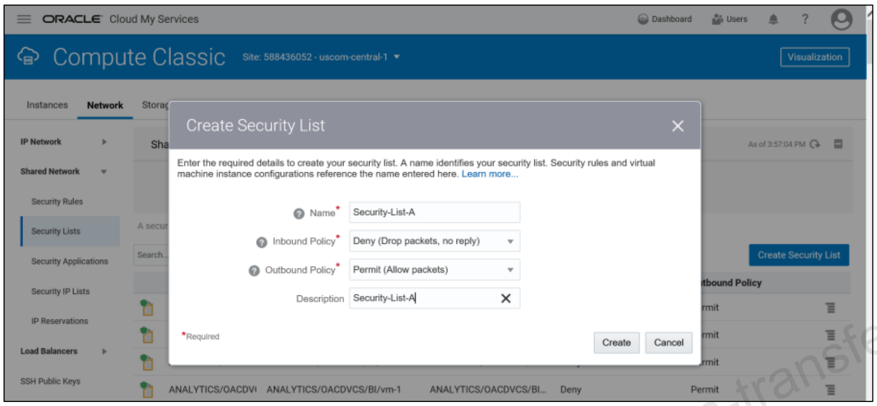
Creating Security Lists

2. Navigate to the Network tab and click on Create Security List.



Creating Security Lists

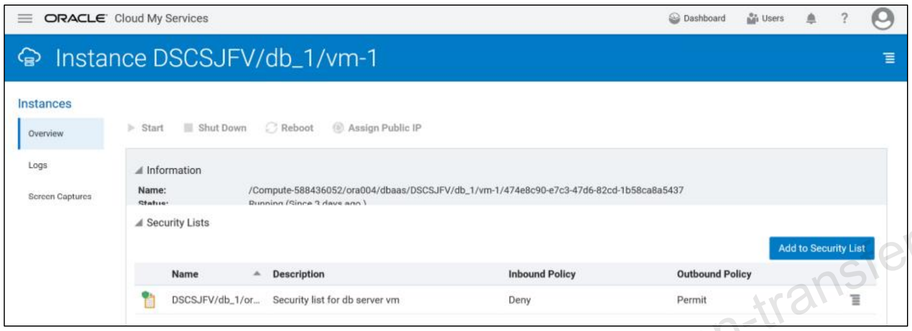
3. Create Security-List-A. Specify a name and description, the inbound and outbound policies, and click Create. Similarly, create Security-List-B.



ORACLE

Adding Instances to Security Lists

1. In the Oracle Compute Classic Cloud Service console, navigate to the Network tab.
2. On the Instances page, identify the instance that you want to update. From the menu, select View. On the instance details page, click Add to Security List.
3. Select the security list to which you want to add your instance, and click Add.

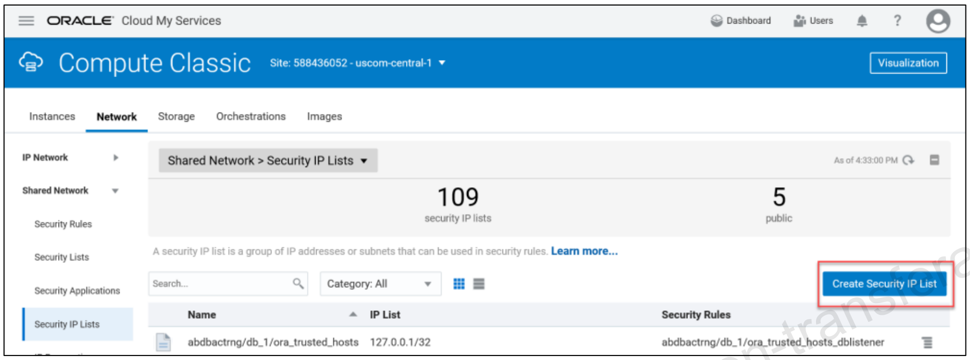


You can add an instance to a security list by updating the instance. When you add an instance to a security list, the instance can communicate freely with all the other instances in the same security list. Any security rules that are defined for the security list are applicable to all the instances in that security list.

For this demo, add all the production instances to Security-List-A and all nonproduction instances to Security-List-B.

Creating Security IP Lists

1. On the Oracle Compute Classic Cloud Service console, navigate to the Network tab.
2. Click on Security IP Lists under Shared Network.
3. Click **Create Security IP List.**



To permit traffic from external hosts to Oracle Compute Cloud Service instances, you must define those hosts in a Security IP List.

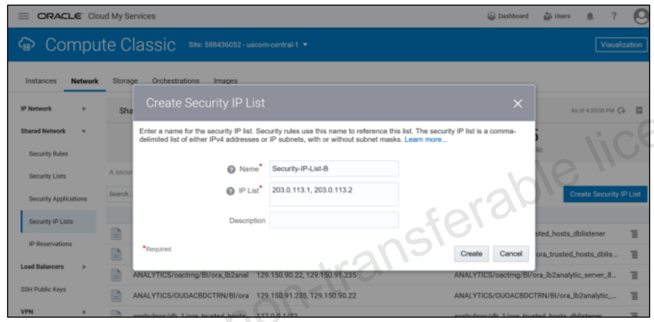
For this demo, add all the production instances to Security-IP-List-A and all nonproduction ones to Security-IP-List-B.

Creating Security IP Lists

4. In the Create Security IP List dialog box, specify a name and provide a comma-separated list of the subnets (in CIDR format) or IPv4 addresses for which you want to create the security IP list.

   For example, to create a security IP list containing the IP addresses 203.0.113.1 and 203.0.113.2, specify entries as 203.0.113.0/30 or 203.0.113.1, 203.0.113.2. You can specify up to 100 entries in a security IP list.

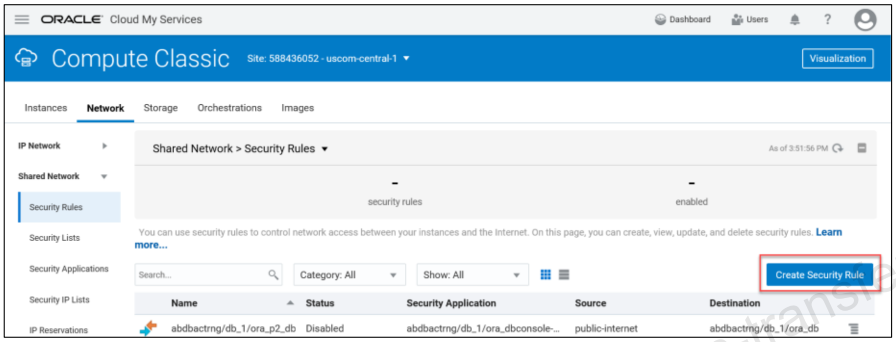
5. Click **Create**.



ORACLE

Permitting Access to Oracle Cloud Service Instances

1. In the Oracle Compute Classic Cloud Service console, navigate to the Network tab.
2. Click on Security Rules under Shared Network.
3. Click **Create Security Rule.**

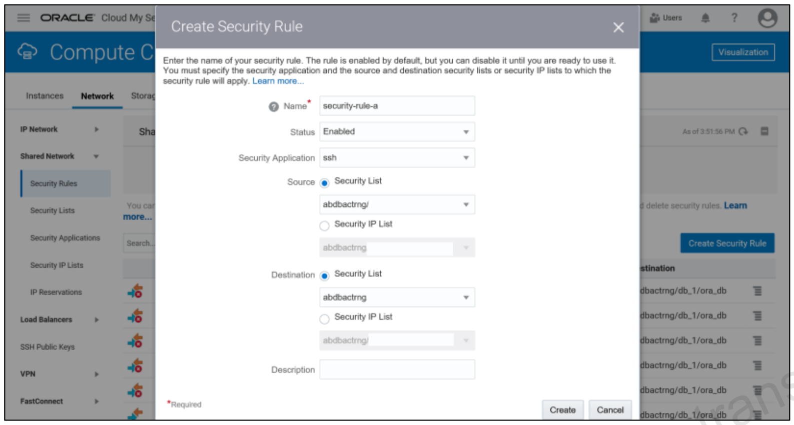


Security rules are essentially firewall rules that you can use to permit traffic between Oracle Compute Cloud Service instances in different security lists, as well as between instances and external hosts.

The source and destination specified in a security rule can be either a security IP list (that is, a list of external hosts) or a security list.

Permitting Access to Oracle Cloud Service Instances

4. Specify the values for Security Rules and click Create.



Summary

In this solution, you should have learned how to:

- Use the Network console to create and manage security lists, rules, IP lists, and network groups
- Use Oracle Database Cloud Network console for network isolation of production and nonproduction servers

ORACLE

Practice 11: Overview

There are no practices for this lesson.

12

Overview of Migrating to Oracle Database Cloud Service

ORACLE

Objectives

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

Choosing a Migration Method: Considerations

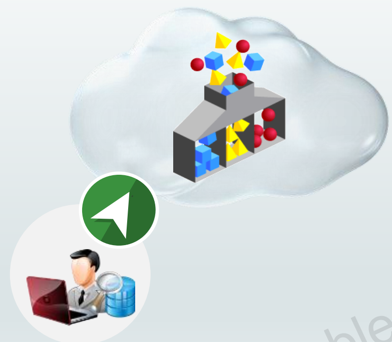
Some of the characteristics and factors to consider when choosing a migration method are:

- 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

Choosing a Migration Method: Understanding Your Requirements

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* 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



See Choosing a Migration Method in *Using Oracle Database Cloud Service* for additional information.

Choosing a Migration Method

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-D1398B68-D9DD-4BEE-8A52-5BE108374E80

Choosing a Method: Oracle Data Pump Considerations

Method	Pros	Cons
Conventional export/import	Can export from 11*g* and import into 18*c* or 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

Choosing a Method: Unplug/Plug and Remote Cloning Considerations

Method	Pros	Cons
Unplug/plug	Very fast Very easy	Unplug is only for Oracle Database 18*c* and 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

ORACLE

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

Method	Pros	Cons
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) to Migrate* lesson.

Applicable Migration Methods

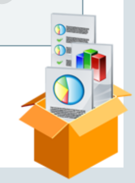
Method	On-premises 11*g* database to Cloud 11*g* database	On-premises 11*g* database to Cloud 12*c* PDB	On-premises 12*c* non-CDB to Cloud 12*c* PDB	On-premises 12*c* or 18c PDB to Cloud 12c or 18c PDB
Data Pump Conventional Export/Import	Y	Y	Y	Y
Data Pump Transportable Tablespace	Y	Y	Y	Y
Data Pump Full 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

ORACLE

The methods shown in the table are applicable when migrating an Oracle Database 11*g* or 12*c* on-premises databases to an Oracle Database *18c*, 12*c,* or 11*g* database in the Cloud.

Applicable Migration Methods

Method	On-premises 11*g* database to Cloud 11*g* database	On-premises 11*g* database to Cloud 12*c* PDB	On-premises 12*c* non-CDB to Cloud 12*c* PDB	On-premises 12*c* PDB to Cloud 12*c* PDB	On-premises 12*c* or 18*c* PDB to Cloud 12*c* or 18*c* PDB
Unplugging/Plugging	N	N	Y	Y	Y
Remote Cloning	N	N	Y	Y	Y
SQL Developer and SQL*Loader to Migrate Selected Objects	N	N	Y	Y	Y
SQL Developer and `INSERT` Statements to Migrate Selected Objects	N	N	Y	Y	Y



The methods shown in the table are applicable when migrating an Oracle Database 11*g* or 12*c* on-premises databases to an Oracle Database 18*c*, 12*c*, or 11*g* database in the Cloud.

Summary

In this lesson, you should have learned how to:

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

ORACLE

Practice 12: Overview

There are no practices for this lesson.

13

Using SQL Developer to Migrate

Objectives

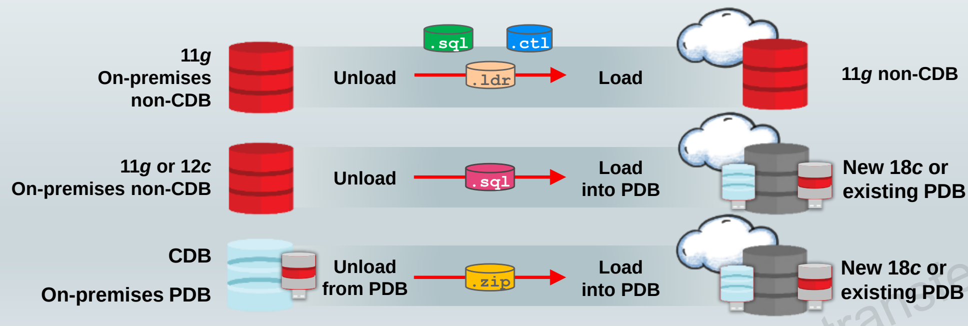
After completing this lesson, you should be able to:

- 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

Overview of Using SQL Developer

Generate files in an on-premises database and use files on the Database Cloud Service database deployment:

- `CREATE` DDL statements + SQL*Loader control and data files
- `CREATE` DDL statements + `INSERT` statements

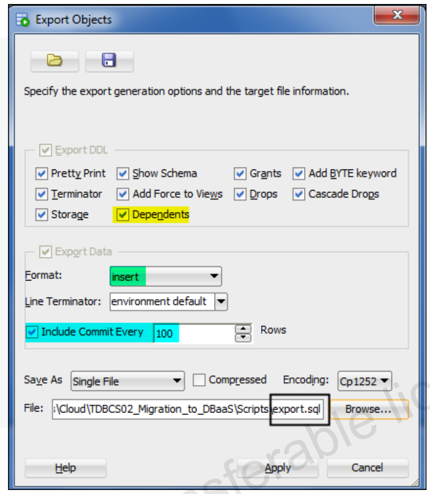


ORACLE

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

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.



See SQL Developer and `INSERT` Statements to Migrate Selected Objects in *Using Oracle Database Cloud Service* for examples.

ORACLE

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

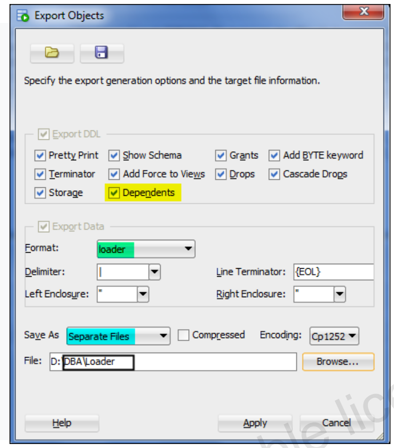
- From Oracle Database 18*c* or 12*c* CDB to Oracle Database 18*c* or 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 18*c* or 12*c* in the Cloud

SQL Developer and `INSERT` Statements to Migrate Selected Objects

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-4B26453A-73D8-44C4-AD44-A2F86626E64F

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.



See SQL Developer and SQL*Loader to Migrate Selected Objects in *Using Oracle Database Cloud Service* for examples.

ORACLE

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

- From Oracle Database 18*c* or 12*c* CDB to Oracle Database 18*c* or 12*c* in the Cloud
- From Oracle Database 12*c* Non-CDB to Oracle Database 18*c* or 12*c* in the Cloud

*SQL Developer and SQL*Loader to Migrate Selected Objects*

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-5B30EC77-26FB-4252-8D95-CE923E0553C8

Summary

In this lesson, you should have learned how 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

ORACLE

Practice 13: Overview

- 13-1: Using SQL Developer and `INSERT` Statements to Migrate Selected Objects

14

Use Case: Automated Patching of Database Cloud Service

Applying Patches to Database Cloud Service Deployments

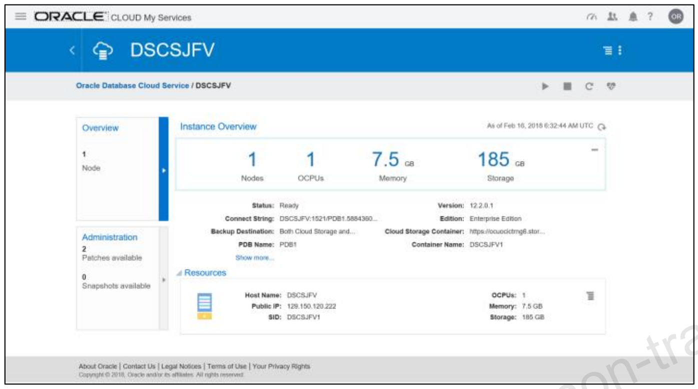
Scenario	Suggested Solution
• The databases hosted on DBCS are not at the current patch set and are vulnerable to security threats.	• Use DBCS cloud tooling to check for a list of required patches and apply them.
Business Justification: Databases that are not patched to the current patch set release are vulnerable to security threats, and are at potential risk from product bugs that could cause service interruption.	
Oracle Cloud Benefits: Patching operations in Oracle cloud is automated through the Database Cloud Service console. The DBA can select the appropriate patch from a list of pending patches and apply it to all the databases hosted in DBCS by using the console. In case of a patch failure, it can be rolled back from the console.	

ORACLE

This use case describes how Oracle Database Cloud Service (DBCS) effortlessly handles mass database patching and simplifies patch maintenance by using the Database Cloud Service console.

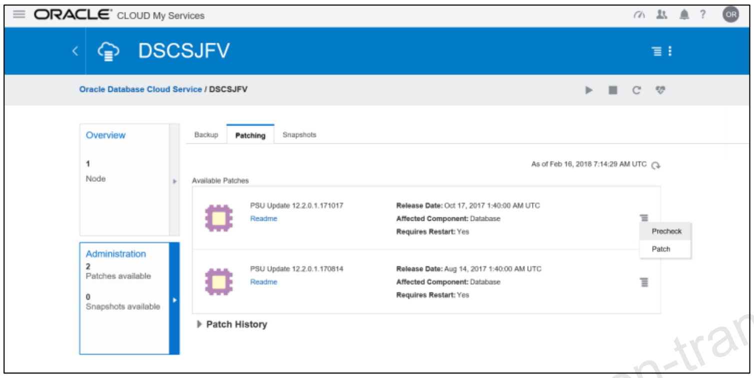
Procedure

1. Navigate to the Oracle Database Cloud Service console.
2. Click on the name of the database deployment that you want to patch.
3. The Oracle Database Cloud Service Instance Overview page is displayed.



Procedure

4. Click on Administration, and navigate to the Patching tab. A list of available patches is displayed.
5. Select **Precheck** for the patch you want to apply.



ORACLE

Procedure

6. Once the prechecks are successful, click **Patch**.
7. When the patching operation completes, the Administration section shows the completion time of the patching operation.

You can roll back a patch or failed patch attempt on a database deployment by using the Oracle Database Cloud Service console.

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

1. Open the Oracle Database Cloud Service console.
2. Click the database deployment for which you want to roll back a patch. The Oracle Database Cloud Service Overview page is displayed.
3. Click the Administration tile and then click the Patching tab. The Oracle Database Cloud Service Patching page is displayed.
4. Click **Rollback**.

Summary

In this solution, you should have learned:

- How the automated patching feature of DBCS helps in updating the patch levels to the latest PSU patches, thereby ensuring adherence to audit requirements
- That the patching operation is automated and does not require manual intervention
- That the latest patches are readily available and do not have to be downloaded separately

Practice 14: Overview

There are no practices for this lesson.

15

Overview of DBCS Performance Management

Objectives

After completing this lesson, you should be able to:

- Describe the performance management methodology
- Describe what may be tuned
- Describe the steps of the tuning methodology

Performance Management in the Database Cloud Environment

- Two components of performance management
  - Performance monitoring
  - Performance tuning
- Database instance is configured per best practices
- Resource usage is dependent on the application
- Subscription cost is related to the resource usage

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

Performance Monitoring and Tuning

Types of tuning:

- Proactive monitoring/tuning: Examining performance statistics at a regular interval to identify whether system behavior and resource usage has changed
- Reactive tuning: Identifying an overused resource in the system and determining potential fixes

Proactive monitoring/tuning: 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: 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.

Tuning Methodology

- Identify the scope of the problem (operating system, database, and so on).
- Tune the following from the top down:
  - The design before tuning the application code.
  - The code before tuning the instance.
- Tune the area with the greatest potential benefit. Identify bottlenecks:
  - Identify the performance problem by using tools such as ADDM, AWR, Oracle SQL Developer Web.
  - Analyze the problem, looking for skewed and tunable components.
  - Use appropriate tools to tune the components implicated.
- Stop tuning when the goal is met.

Oracle Corporation has developed a tuning methodology based on years of experience. The methodology presented in this course is also presented in *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:

- Identify the scope of the problem (operating system, database, and so on). Check the operating system 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 18*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, Oracle SQL Developer Web).
  - 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.

Effective Tuning Goals

Effective tuning goals are:

- Specific
- Measurable
- Achievable
- Cost effective

Tuning example:

- Service level agreement (SLA) states that user response time to a particular request must be no more than 30 seconds.
- Current situation: Average response time is 25 seconds and increasing.
- Tuning goal: User response time to a particular request is 20 seconds.

ORACLE

A tuning goal is the elimination of a defined problem. The goal may be based on a service level agreement (SLA) or a problem.

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 people 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.

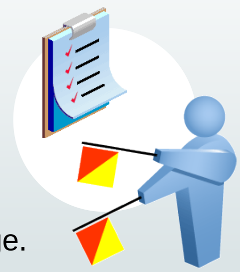
Even these situations are not impossible to change in an absolute sense, but the DBA always has business constraints that limit the amount of money and resources that can be applied to the solution. So every goal should consider the cost to benefit. A goal that costs a great deal but solves a problem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

General Tuning Session

Recommended tuning methodology:

1. Define the problem and state the goal.
2. Collect current performance statistics.
3. Consider common performance errors.
4. Build a trial solution.
5. Implement the solution and measure the change.
6. Assess whether the solution met the goal.



ORACLE

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!

What Can be Tuned in a DBCS Environment?

- Application design
- SQL statements
- Database initialization parameters
- Memory
- Redo and archive logging

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 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:** 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, Enterprise Manager Database Express cannot perform actions that will require a restart of the database instance. Oracle SQL Developer Web 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 DBCS 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.

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.

When 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.

Summary

In this lesson, you should have learned how to:

- Describe the performance management methodology
- Describe what may be tuned
- Describe the steps of the tuning methodology

ORACLE

Practice 15: Overview

There are no practices for this lesson.

16

Tuning Performance Issues

Objectives

After completing this lesson, you should be able to:

- Use various tools for performance management
- Identify performance issues
- Diagnose performance issues

Oracle Database Performance Tuning Guide and *Oracle Database SQL Tuning Guide* provide a comprehensive methodology for tuning database instance and SQL performance issues. The scope of this lesson includes identifying differences in using familiar tools with Oracle Database Cloud Service (DBCS) database deployments and tools that are unique to DBCS.

Tools for Performance Management

- Database Cloud Service tools:
  - Oracle SQL Developer Web
- Other tools:
  - Oracle Enterprise Manager Database Control (Oracle Database 11*g*)
  - Oracle Enterprise Manager Database Express (Oracle Database 18*c* and 12*c*)
  - SQL Developer
  - SQL*Plus

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

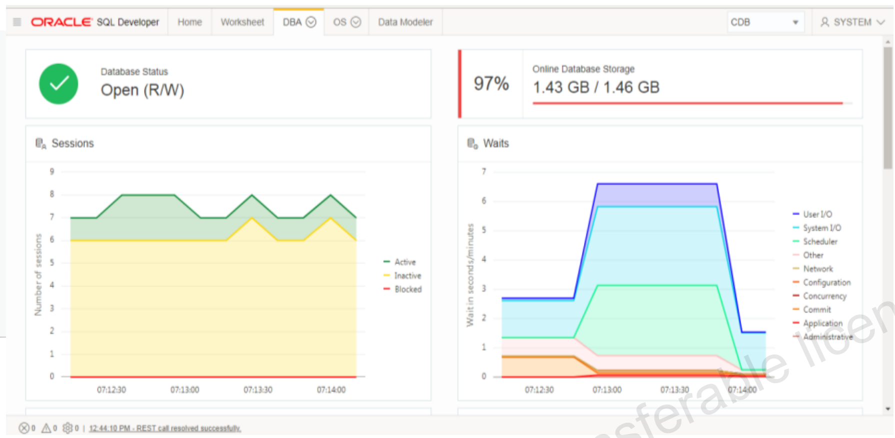
- Oracle Enterprise Manager Database Control is used to manage database deployments that include an Oracle Database 11*g* database.
- Oracle Enterprise Manager Database Express is used to manage database deployments that include an Oracle Database 18*c* or 12*c* database. In Oracle Database 18c or 12*c*, you must configure a separate port to manage the root container (CDB) and each pluggable database (PDB.) 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.
- 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.
- 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.

Oracle SQL Developer Web is a browser-based tool that provides database monitoring. Your options to access the deployment with Oracle SQL Developer Web are:

- Using the Database Deployment's Landing Page
- Using a Direct URL by enabling the security rule in the Compute Service to open the Oracle SQL Developer Web
- Using an SSH Tunnel for Oracle SQL Developer Web port (443)

Oracle SQL Developer Web: Database Information

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



The overview page of the Oracle SQL Developer Web gives a visual representation of the following areas:

- Database
  - Database status
  - Listener status
  - Last backup
  - Online database storage
  - Alerts, sessions, and waits
- Operating system
  - Memory
  - CPU
  - Processes
  - File system storage

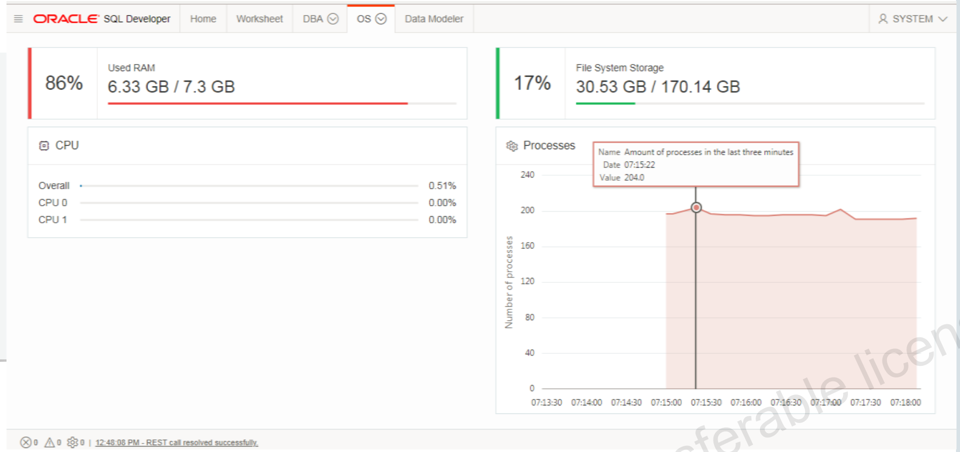
Of particular interest in regard to performance are the Waits, Sessions, and Alerts section. These provide a general view of the health of the instance.

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 DB Time.

Oracle SQL Developer Web: Operating System Information

- Memory
- CPU
- Processes
- File system storage

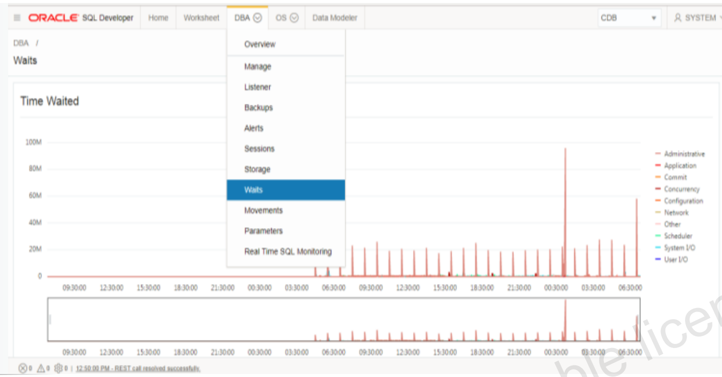


ORACLE

Identifying Performance Issues by Using the Oracle SQL Developer Web

High-level overview of database waits:

- Shows overall waits in the database deployment
- Select waits by PDB to see individual contributions to overall waits
- View wait details showing specific SQL statements



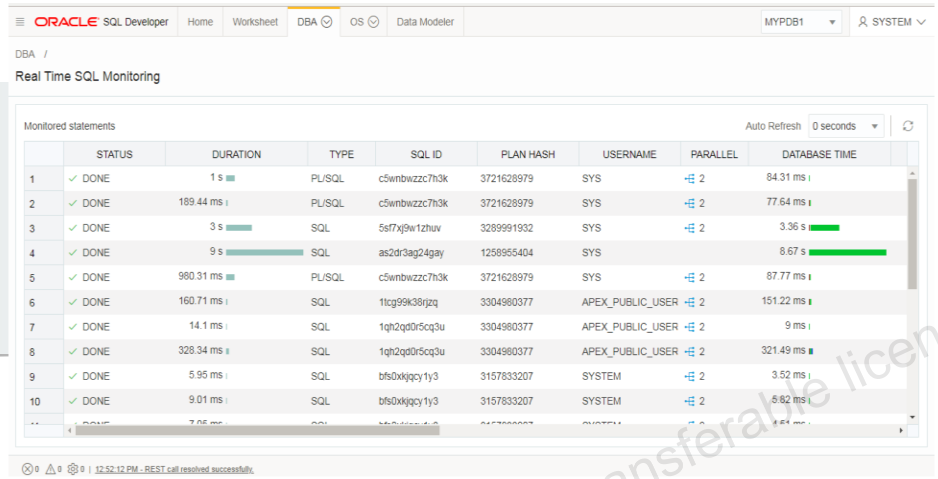
ORACLE

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. Follow the tuning model and determine where the greatest waits are occurring.

Oracle SQL Developer Web 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. Oracle SQL Developer Web shows the overall waits in the entire deployment. 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.

Identifying Performance Issues by Using the Oracle SQL Developer Web

Real Time SQL Monitor: Use to identify high-cost SQL statements that are running or have run in the recent past.

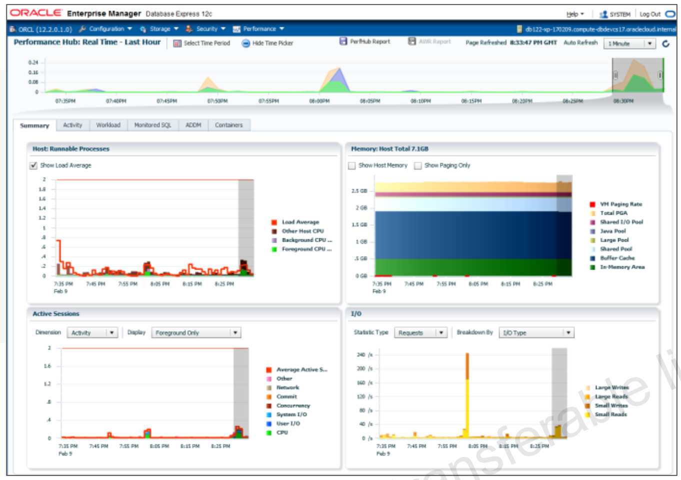


ORACLE

The Oracle SQL Developer Web also has a Real Time SQL Monitor available. Use this feature to 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.

Using the Enterprise Manager Database Express Performance Hub

Performance Hub: Provides a visual overview of the health of the database instance and activity over the last hour by default



Oracle Enterprise Manager Database Express is the same tool that is available in an on-premises database.

In Enterprise Manager 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.

The Performance Hub Summary tab panes show:

- Host Runnable Processes: Shows processes from the OS perspective.
- Active Sessions: Shows the activity in the instance by service or container.
- Memory: Shows the allocation of memory to the database instance.
- I/O: Shows database instance I/O activity.

With Oracle Database 12*c* Release 1, Enterprise Manager Database Express uses a different port for each PDB. With a PDB connection, the Performance Hub does not have all of the options that are available for the CDB.

In Oracle Database 12*c* Release 2, you can set a global port which enables Enterprise Manager Database Express access to the CDB and all PDBs on that single port.

Identifying Performance Issues by Using Enterprise Manager Database Express

Identify waits and high-cost SQL statements:

- Wait class pull-down on the Activity tab:
  - Enables you to filter results
  - Classify waits in different ways
- Stored Automatic Database Diagnostic Monitor reports on the ADDM tab:
  - Identify performance issues with the highest impact in terms of database time
  - Identify the root cause
  - Provide recommendations

Whatever tool you use, the key to identifying performance issues are wait events and high cost SQL. With Enterprise Manager Database Express the waits are displayed graphically in several ways.

In the Activity tab, the wait class pull-down enables 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 enables you to view stored Automatic Database Diagnostic Monitor reports. The ADDM feature requires the High Performance or Extreme Edition of Oracle Database Cloud Service.

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.

Identifying Performance Issues by Using SQL Developer

- Instance Viewer: Provides a real-time overview of the database instance (CDB level for Oracle Database 18c)
- ADDM reports, findings, and recommendations

SQL Developer provides much of the same information that is found in Enterprise Manager Database Express, presented in a different format.

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.

Using ADDM to Diagnose Performance Issues

ADDM report sample:

```
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
```


You can access ADDM reports in a variety of ways. Two ways were shown earlier in this lesson. 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 “Overview of Automatic Database Diagnostic Monitor” in *Oracle Database Development Guide*.

The snippet of an ADDM report shown on the slide indicates that the SQL Tuning Advisor is recommended and the particular SQL statement that might be tuned.

Using the SQL Tuning Advisor

- SQL Tuning Advisor: Analyzes the highest cost SQL statements to determine if there is a better SQL execution plan.
- SQL Tuning Advisor modes:
  - Automatic SQL Tuning Advisor: Runs every night in the maintenance window.
  - On-demand SQL Tuning Advisor: Runs on demand to tune one or more SQL statements.
- Invoke the SQL Tuning Advisor from:
  - Enterprise Manager Database Express
  - SQL Developer

ORACLE

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 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 may be 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 Enterprise Manager Database 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.

Consider the following when determining which tool to use:

- In a primarily development environment, the users may already be 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.

- In a production environment, you many not be able to change the code. The code might be provided by a third party and if changed it would not be supported. Accessing the SQL Tuning Advisor through Enterprise Manager Database Express enables you to use SQL Profiles, which does not change any code, but gives directions to the Optimizer to produce more efficient execution plans.

- In a production environment, you many not be able to change the code. The code might be provided by a third party and if changed it would not be supported. Accessing the SQL Tuning Advisor through Enterprise Manager Database Express enables you to use SQL Profiles, which does not change any code, but gives directions to the Optimizer to produce more efficient execution plans.

Summary

In this lesson, you should have learned how to:

- Use various tools for performance management
- Identify performance issues
- Diagnose performance issues

Practice 16: Overview

There are no practices for this lesson.

17

Performance Management

ORACLE

Objectives

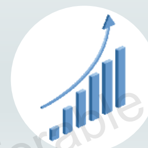
After completing this lesson, you should be able to:

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

Oracle Database Performance Tuning Guide and *Oracle Database SQL Tuning Guide* provide a comprehensive methodology for tuning database instance and SQL performance issues. The scope of this lesson includes identifying differences in using familiar tools with Oracle Database Cloud Service (DBCS) database deployments and tools that are unique to DBCS.

Database Deployment Scaling: Overview

- A database deployment can be scaled in two ways:
  – Increase CPU and memory allocated to a database deployment.
  – Scale up the storage of a database deployment.
- Before scaling up the database deployment, examine the resources allocated and required by different consumers within the database.
- Resource Manager can be used to allocate resources (CPU) between PDBs and among users inside a PDB.
- Database initialization parameters can be used to control SGA and PGA allocations.



ORACLE

If an Oracle Database Cloud Service database deployment 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 database deployment can be scaled in two ways:

- Increase CPU and memory allocated to a database deployment. Scale up the compute shape of a database deployment. The Compute Shape of a database deployment is the combination of the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) for the compute node hosting the new database deployment. Database Cloud Service (DBaaS) offers several OCPU/RAM combinations.
- Scale up the storage of a database deployment. This results in adding a storage volume to the database deployment's compute node. 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 database deployment, 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:

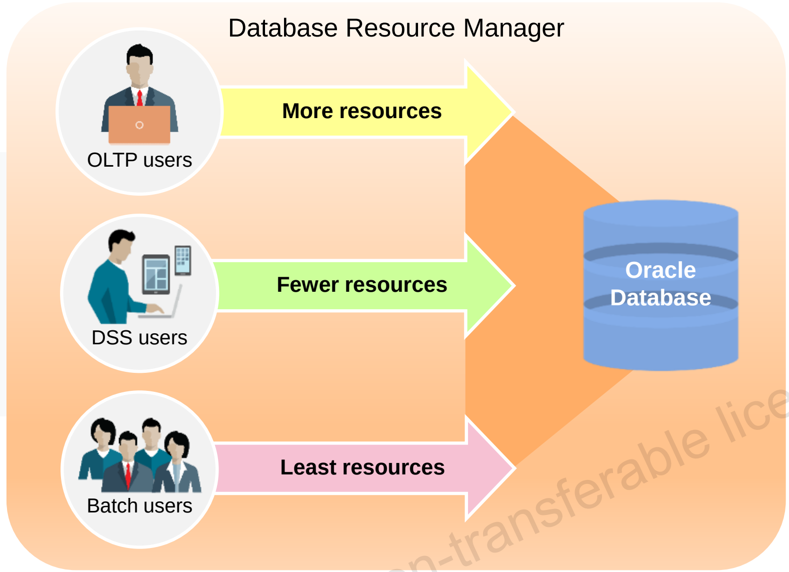
- Shares of the system resources allocated to PDBs within the CDB so that resources are allocated fairly to all PDBs or more resources to the more important PDBs
- Resources to consumer groups within the non-CDB or PDB

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

Database Resource Manager: Overview

Use the Resource Manager to:

- Manage mixed workload
- Control system performance



ORACLE

By using the Database Resource Manager (also called the Resource Manager), you have more control over the allocation of machine resources than is normally possible through operating system resource management alone. If resource management decisions are made by the operating system, it can lead to problems such as:

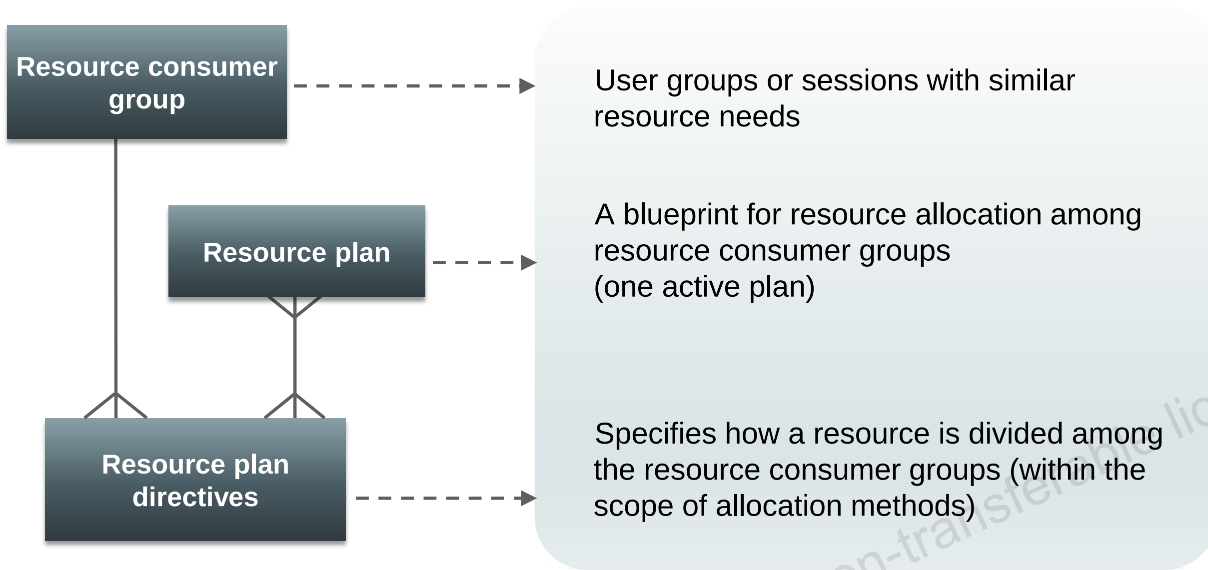
- Excessive overhead resulting from operating system context switching of Oracle Database server processes when the number of server processes is high
- Suspension of a database server process that is holding a latch
- Unequal distribution of resources among all Oracle Database processes, and an inability to prioritize one task over another
- Inability to manage database-specific resources, such as parallel execution servers and active sessions

The Database Resource Manager controls the distribution of resources among various sessions by controlling the execution schedule inside the database. By controlling which sessions run and for how long, the Database Resource Manager can ensure that resource distribution matches the plan directive and, therefore, the business objectives. With the Database Resource Manager, you can guarantee groups of users a minimum amount of processing resources regardless of the load on the system and the number of users.

The `DBMS_RESOURCE_MANAGER_PRIVS` package contains the procedures to grant and revoke the `ADMINISTER_RESOURCE_MANAGER` system privilege, which is a prerequisite for invoking the Resource Manager.

For a full description of Resource Manager, see “Managing Resources with Oracle Database Resource Manager” in *Oracle Database Administrator’s Guide*.

Database Resource Manager: Concepts



Administering systems by using the Database Resource Manager involves the use of resource plans, resource consumer groups, and resource plan directives.

A *resource consumer group* defines a set of users or sessions that have similar requirements for using system and database resources.

A *resource plan* specifies how the resources are distributed among various resource consumer groups. The Database Resource Manager also allows for creation of plans within plans, called *subplans*.

Resource plan directives specify how a particular resource is shared among consumer groups or subplans. You associate resource consumer groups and subplans with a particular resource plan through plan directives.

Resource allocation methods determine what policy to use when allocating for any particular resource. Resource allocation methods are used by resource plans and resource consumer groups.

Using Resource Manager to Control PDB Resource Usage

- Resource Manager ensures fairness between multiple PDBs in the same CDB by:
  - Allocating shares of CPU
  - Limiting CPU utilization percent
  - Limiting parallel server percentage
- DEFAULT_CDB_PLAN: Allows one share per PDB, guaranteeing each PDB an equal share when the CPU is fully loaded.
- DEFAULT_CDB_PLAN example: CDB with 5 PDBs
  - Each PDB is guaranteed 1/5 of the CPU capacity when the CPU is fully loaded
  - Any other time each PDB is allowed to use as much CPU as it requests
- Create a custom plan to set priorities for the PDBs by giving more important PDBs additional shares.

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 initialization 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 places 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 Enterprise Manager Database Express. Using Enterprise Manager Database 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 learn how to create a new plan, see “Creating a CDB Resource Plan” in *Oracle Database Database Administrator’s Guide*.

To change the active plan or edit the parameters, see “Using EM Express to Modify a CDB Resource Plan” in *Oracle Database 2 Day DBA*.

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.

Controlling Resource Usage by Consumer Groups

- Resource Manager plan at the PDB level allocates resources by consumer groups
- PDB-level plan restrictions:
  – Cannot have subplans
  – Limited to 8 consumer groups
  – Cannot have a multiple level scheduling policy
- 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.

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.

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 “Creating a PDB Resource Plan” in *Oracle Database Administrator’s Guide*.

Determining When to Scale Up the Database Deployment

- Determine if the DBCS instance is exceeding the CPU capacity of the DBCS deployment:
  - Oracle SQL Developer Web: Review the "On CPU" wait event on the Waits page
  - Enterprise Manager Database Express Performance Hub: Compare CPU line in the Activity tab with CPU waits
  - View ADDM reports
- Determine if the DBCS deployment is running out of memory:
  - Oracle SQL Developer Web: DBA Menu > View Memory page
  - Enterprise Manager Database Express: Access the Performance Hub
- Determine if the database deployment is running out of space:
  - Oracle SQL Developer Web: View tablespace space usage
  - Enterprise Manager Database Express: View tablespace and archived log space usage
  - SQL Developer: View tablespace space usage

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

How can you determine if the DBCS instance is exceeding the CPU capacity of the DBCS deployment?

- Oracle SQL Developer Web can help you determine if the DBCS instance is exceeding the CPU capacity. Consider the "On CPU" wait event on the Oracle SQL Developer Web 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 Oracle SQL Developer Web 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?

- Oracle SQL Developer Web: DBA Menu > Memory page
- Enterprise Manager 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 database deployment is running out of space?

- Oracle SQL Developer Web can help you determine if the DBCS database is running out of space. Oracle SQL Developer Web shows the fullness for every tablespace.
- Enterprise Manager Database Express can help you determine if the DBCS database is running out of space. Enterprise Manager Database Express shows the fullness for every tablespace in each container, PDB, and root. Enterprise Manager Database Express also shows the space used by archived log files.
- SQL Developer can also display the same type of information.

Scaling 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

See Scaling a Database Deployment in *Using Oracle Database Cloud Service*.

ORACLE

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.

Scaling a Database Deployment

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3339

Summary

In this lesson, you should have learned how to:

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

ORACLE

Practice 17: Overview

- 17-1: Scaling Up the Database Deployment

18

Using REST APIs to Manage Oracle Database Cloud Service

ORACLE

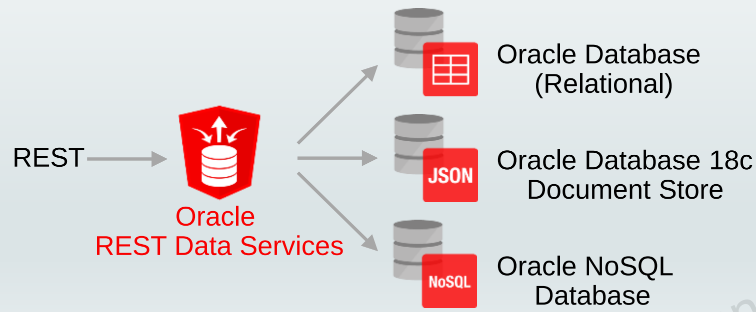
Objectives

After completing this lesson, you should be able to:

- Describe Oracle REST Data Services (ORDS)
- Use REST APIs to manage Oracle Database Cloud Service

What is Oracle REST Data Services (ORDS)?

- Turns your Oracle databases into RESTful API services
- Runs on application servers (WebLogic, GlassFish, and Tomcat) for production
- Provides a simple, stand-alone mode for development and test



Oracle REST Data Services (ORDS) is a Java EE-based alternative for Oracle HTTP Server and `mod_plsql`. The Java EE implementation offers increased functionality including a command line based configuration, enhanced security, file caching, and RESTful web services. Oracle REST Data Services also provides increased flexibility by supporting deployments using Oracle WebLogic Server, GlassFish Server, and Apache Tomcat. A stand-alone mode that doesn't require an application server can be used for development and testing.

Oracle REST Data Services (ORDS) makes it easy to develop modern REST interfaces for relational data in an Oracle database, both in the Cloud and on-premises. ORDS can also be used with the Oracle Database 18c JSON Document Store and Oracle NoSQL Database.

Refer to *Oracle REST Data Services Installation, Configuration, and Development Guide* for detailed information on ORDS.

Note: ORDS was formerly known as Oracle APEX Listener.

Key Principles of Representational State Transfer (REST)

Ubiquitous client API

- Uses same protocol as World Wide Web -> HTTP/HTTPS
- Supported by all languages

Resource-focused API at server

- URIs uniquely identify resources, such as orders, employees, and products
- Fixed set of HTTP/HTTPS operations: GET (read), POST (insert), PUT (update), and DELETE (delete)

Stateless

- Single call; no sessions
- JSON documents are used to return results of queries and input for inserts/updates

ORACLE

Representation State Transfer (REST) uses the same protocol as the World Wide Web, namely HTTP and HTTPS and is supported by all programming languages.

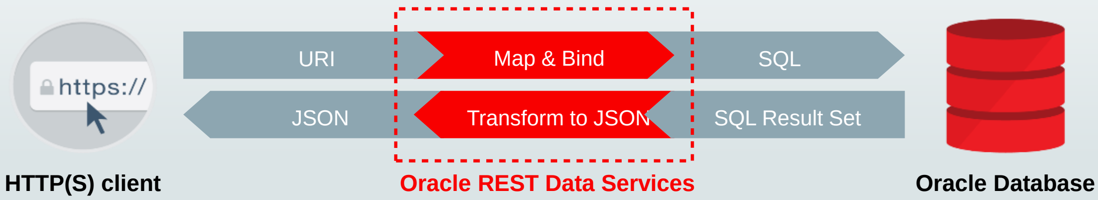
A fixed set of HTTP/HTTPS operations are used with URIs, which uniquely identify specific resources such as orders, employees, and products. There are also calls to retrieve metadata.

REST is stateless, meaning there are no sessions. Everything is done with a single call.

JSON documents are used for input and output.

What Does ORDS Do?

- For input, maps/binds URI and parameters to SQL and PL/SQL handlers
- For output, transforms results to JSON and other formats



ORDS maps the resource specified in the URI to a module and template. It then invokes a handler, SQL or PL/SQL for Oracle Database. The request is processed and results are transformed, typically into JSON, and then returned to the client application.

ORDS and Database Cloud Service

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]#
```

ORACLE

When a database deployment is created on Oracle Database Cloud Service, ORDS is started.

Using REST APIs in Oracle Cloud

- REST APIs can be used to return details of a service instance built in Oracle's Public Cloud.
- Base format for a REST API:

```
{RESTendpoint}/paas/service/dbcs/api/v1.1/instances/{identityDomainID}/{serviceid}
```

- Complete REST command using cURL:

```
curl -v -X get -u "{firstname.lastname@domain.com:mypassword}" -H "X-ID-TENANT-
NAME:{identitydomainID}" https://{rest endpoint for database service
deployment}/paas/service/dbcs/api/v1.1/instances/{identityDomainID}/{serviceID}
```

ORACLE

A *REST endpoint* is the location of one or more objects, (resource or resources), exposed at the end of an API endpoint. An endpoint by itself is just a reference to a URI that accepts web requests.

cURL is a command-line tool used to get and send files using URL syntax.

Using REST APIs with Database Cloud Service

- You can use REST APIs to:
  - Create a database deployment
  - Delete a database deployment
  - Scale a database deployment
  - Stop, start, or restart a database deployment or compute node
  - View a database deployment
  - View a database deployment's compute nodes
  - View all database deployments
  - View the job status of an operation
- See About the REST APIs in *REST API for Oracle Database Cloud Service* for detailed information.

ORACLE

REST APIs can be used to create and manage a database deployment in Oracle Database Cloud Service.

About the REST APIs

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBR

Using cURL and REST APIs

Parameter	Required?	Description
`-v`	No	Makes the fetching more verbose. Usually used for debugging. Lines starting with '>' means "header data" sent by cURL, '<' means "header data" received by cURL that is hidden in normal cases, and lines starting with '*' means additional info provided by cURL.
`-u`	Yes	Authorization string in the format of username and password of user making the request.
`-H`	Yes	Header parameters follow.
`X-ID-TENANT-NAME`	Yes	Name of the identity domain for the Database Cloud Service account.
`REST Endpoint`	Yes	Path of the name REST endpoint for the database deployment.
`identityDomainID`	Yes	Name of the identity domain for the Database Cloud Service account.
`serviceid`	Yes	The database deployment name.

ORACLE

The table on the slide provides a list of parameters used in the cURL command with REST APIs.

Example: Viewing 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_do
 main>/
 <your_database_deployment>
```

ORACLE

The example in the slide shows how you can use cURL and the REST API to view information about your database deployment.

Example: Deleting 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/DJDBCS HTTP/1.1
...
> X-ID-TENANT-NAME:oucloudusajul10
>
< HTTP/1.1 202 Accepted
…
```

ORACLE

The example in the slide shows how you can use cURL and the REST API to delete your database deployment.

Example: Deleting a Database Deployment

```
< 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","i
dentity_domain":"oucloudusajul10","creation_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.oraclecloud.com:443\/paas\/service\/dbcs\/api\/v1.1\/insta
nces\/oucloudusajul10\/DJDBCS"}
```

ORACLE

Reference: REST API HTTP Status Codes

HTTP Status Code	Description
200 OK	The request was successfully completed a 200 status is returned for a successful GET or POST method.
202 Accepted	The request has been accepted for processing, but the processing has not been completed. The request may or may not eventually be acted upon, as it may be disallowed at the time processing actually takes place. The response contains a LOCATION header of a job that the client should pool to determine when the job has finished.
400 Bad Request	The request could not be processed because it contains missing or invalid information (such as, a validation error on an input field, a missing required value, and so on).
401 Unauthorized	The request is not authorized. The authentication credentials included with this request are missing or invalid.
403 Forbidden	The user cannot be authenticated. The user does not have authorization to perform this request.
404 Not Found	The request specifies an endpoint that does not exist.

The table on this page and the next provides a list of the REST API status codes that may be returned.

Reference: REST API HTTP Status Codes

HTTP Status Code	Description
405 Method Not Allowed	The HTTP verb specified in the request (DELETE, GET, POST, PUT) is not supported for the specified endpoint.
406 Not Acceptable	The object identified by the request is not capable of generating a representation corresponding to one of the media types in the Accept header of the request. For example, the client's Accept header request XML be returned, but the endpoint can only accept JSON.
415 Not Acceptable	The client's ContentType header is not correct (for example, the client attempts to send the request in XML, but the endpoint can only accept JSON).
500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
503 Service Unavailable	The server is unable to handle the request due to a temporary overloading or maintenance of the server. The database Cloud Service REST web application is not running.

ORACLE

Summary

In this lesson, you should have learned how to:

- Describe Oracle REST Data Services (ORDS)
- Use REST APIs to manage Oracle Database Cloud Service

Practice 18: Overview

- 18-1: Using REST APIs

19

Deleting a Database Deployment

Objectives

After completing this lesson, you should be able to:

- Describe what happens when you delete a database deployment
- Describe the tools you can use to delete a database deployment
- Delete a database deployment by using the Oracle Database Cloud Service console

Review: 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.
- All backups to cloud storage are deleted (optional).

ORACLE

When you delete a database deployment, resources associated with the deployment are released. Optionally, backups associated with the database deployment.

You cannot delete a database deployment that has any linked-clone deployments created from any of its snapshots. You must first delete all linked-clone deployments derived from the database deployment before deleting it.

Review: How Can You Delete a Database Deployment?

Delete a database deployment by using the:

- Oracle Database Cloud Service console. See Deleting a Database Deployment.
- `oracle-dbcs-cli` utility. See The delete Subcommand.
- REST API. See Delete a Service Instance.

You can delete a database deployment by using the Oracle Database Cloud Service console or the `oracle-dbcs-cli` utility.

Deleting a Database Deployment

http://www.oracle.com/pls/topic/lookup?ctx=E38328-01&id=CSDBI3305

The delete Subcommand

http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=GUID-D085E99F-259D-40DA-BBCD-E77C113D5111__DBAAS_ORACLE-DBCS-CLI_DELETE

Delete a Service Instance

http://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/csdbr/op-paas-service-dbcs-api-v1.1-instances-%7bidentityDomainId%7d-%7bserviceId%7d-delete.html

Summary

In this lesson, you should have learned how to:

- Describe what happens when you delete a database deployment
- Describe the tools you can use to delete a database deployment
- Delete a database deployment by using the Oracle Database Cloud Service console

Practice 19: Overview

- 19-1: Deleting a Database Deployment

20

Use Case: Creating a Cloned Database Deployment from a Snapshot

Overview of the Use Case

- You can create an Oracle Database Cloud Service database deployment from a snapshot you have taken of another database deployment in the same identity domain. The resulting deployment is known as a linked clone because its storage is linked to the snapshot's storage.
- When you create a linked clone deployment, Database Cloud Service creates a new database deployment whose storage volumes 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. Thus, you can create several linked clones from the same snapshot to use for application testing or branched application development work.

ORACLE

Creating a Cloned Database Deployment from a Snapshot

Scenario	Suggested Solution
• The testing team needs to frequently test data as part of the process of testing newly built modules. • The need for frequent testing necessitates cloning of the production database. • The solution should be scalable and simple.	• Use Oracle Database Cloud Service to host the new cloned environments. • Create a database deployment using snapshots for clones.

Business Justification: The production database is hosted on Database Cloud Service. The application team has a requirement to test data at das part of testing each of the newly built modules. A quick solution is recommended to create cloned environments from the production database.

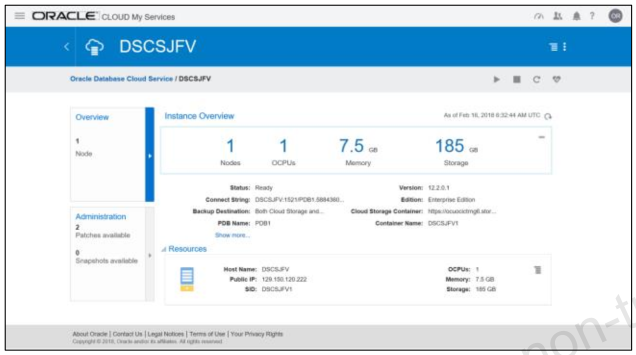
Oracle Cloud Benefits:

- Reduced time and cost of provisioning and management
- Self-service and ease of access
- Standardization of services for effective database management
- Easier adoption of full Oracle capabilities through automation
- Elastic, capacity on demand
- Universal Credit Model, with Pay as You Go and Bring Your Own License features.

ORACLE

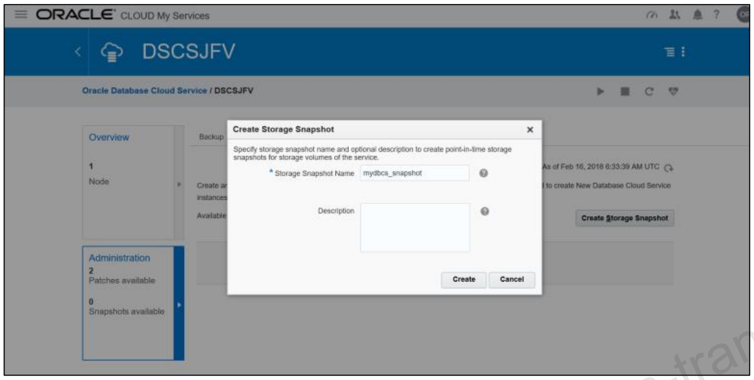
Procedure

1. Navigate to the Oracle Database Cloud Service console.
2. Click on the name of the database deployment whose snapshot you want to use as the basis for a linked-clone deployment.
3. The Oracle Database Cloud Service Instance Overview page is displayed.



Procedure

4. Click on the Administration tile and then navigate to the Snapshots tab. The Oracle Database Cloud Service Snapshots page is displayed. Click Create Storage Snapshot.



Procedure

5. In the Available Storage Snapshots list, locate the snapshot from which you want to create a linked-clone deployment, and choose Create Database Clone from that snapshot's menu.
6. The Create Instance wizard is displayed.
7. Navigate through the pages of the wizard to provide information about the linked-clone deployment.
8. Provide a new service name, specify an SSH public key, and provide an administrator password. You can change the default values for the deployment, for example, Shape and Backup Destination.
9. Confirm and submit your selection. The DBCS deployment from the storage snapshot is created.

ORACLE

Summary

In this solution, you should have learned to:

- List the benefits of using Oracle Database Cloud Service
- Create a Clone Database Deployment from a Snapshot

ORACLE

Practice 20: Overview

There are no practices for this lesson.

A

Oracle Database Cloud Overview: User Managed Vs Autonomous

ORACLE

Hi, I am Ted.
I am an Oracle Database Administrator.

My company has decided to move our databases and applications to the Oracle Cloud.
I have been hearing about Autonomous Database Cloud Services and about Oracle Cloud Infrastructure Database.

Which cloud service should I use to deploy my databases to the Oracle Cloud?

Objectives

After completing this lesson, you should be able to:

- Describe the offerings for Oracle Database Cloud Service
- Describe the Oracle Cloud Platform for Database in the Cloud
- Describe Oracle Autonomous Cloud Platform
- Describe the features of Autonomous Database
- Explain the difference between User Managed and Autonomous Database Service

Learning Agenda

- Database Cloud Service: Overview
- Oracle Cloud Platform for Database in the Cloud
- Oracle Autonomous Cloud Platform
- Autonomous Database: Overview
- Difference between User Managed and Autonomous

Database Cloud Services

- Oracle offers database cloud services in a range of public cloud deployment choices:
  - Fully managed pluggable databases with Exadata Express Cloud Service
  - Virtualized and bare metal databases with Database Cloud Service
  - Databases running on world class engineered infrastructure with Exadata Cloud Service
- Oracle also offers Cloud At Customer for customers who want Exadata database cloud services behind their firewalls, on their own premises.

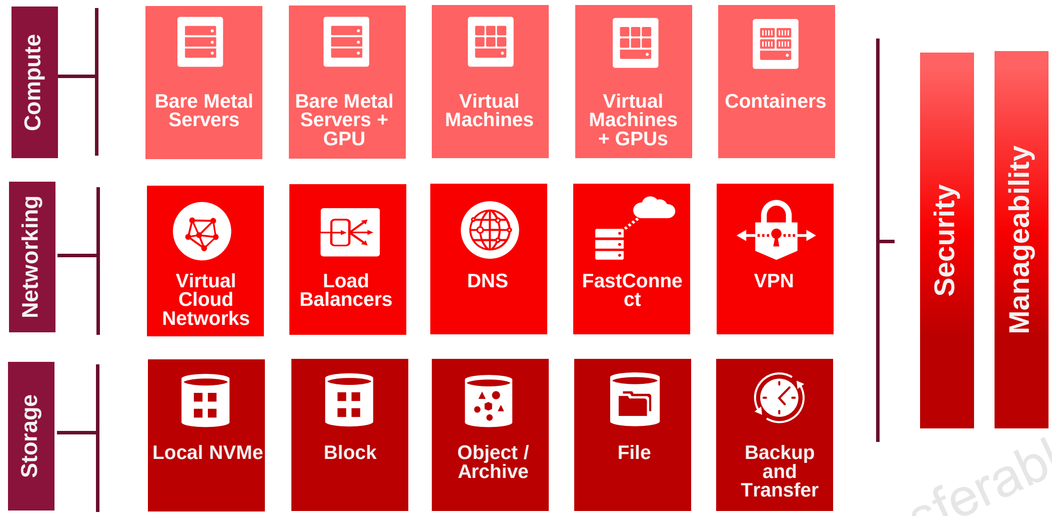
ORACLE

Learning Agenda

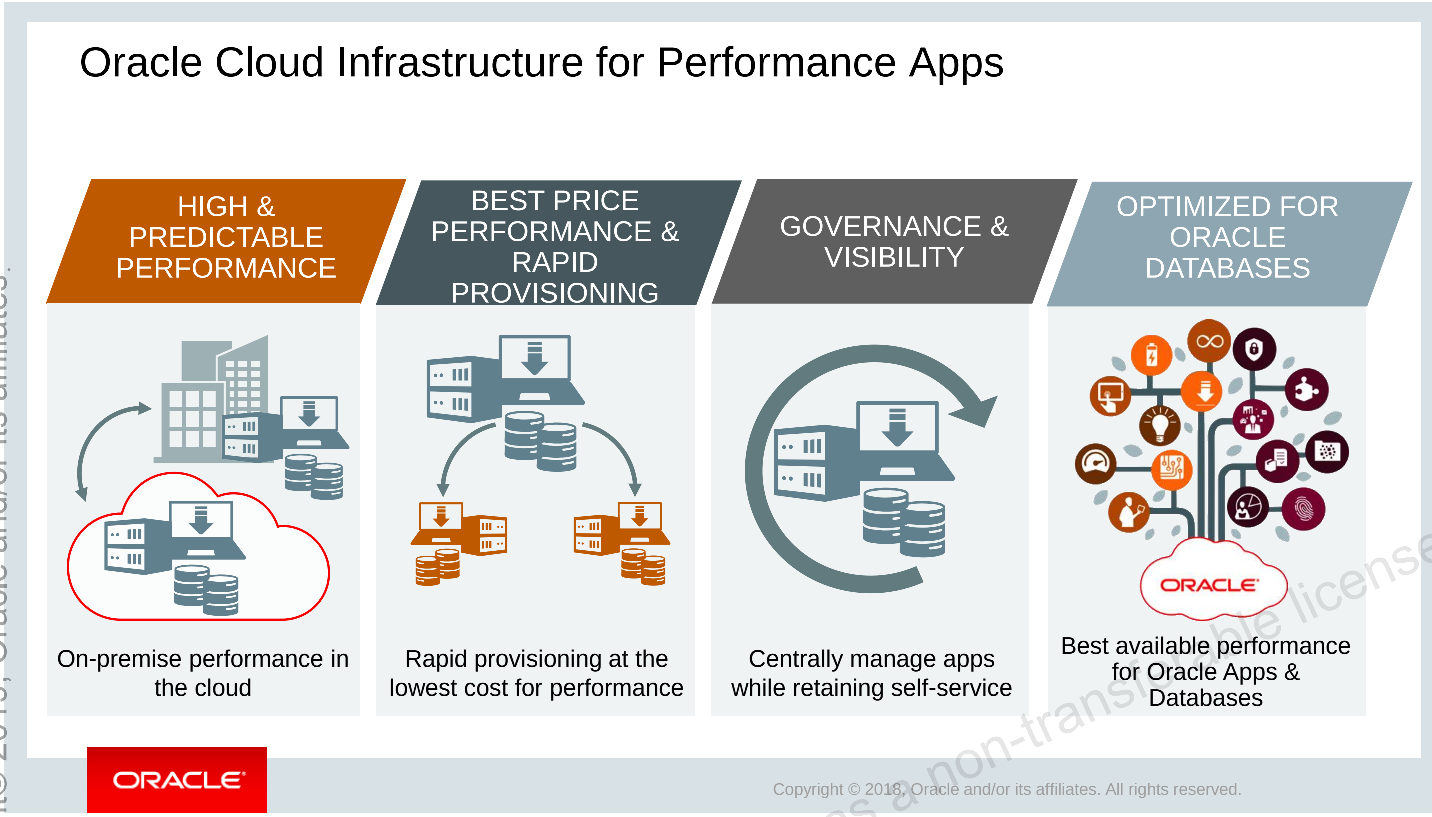
- Database Cloud Service: Overview
- Oracle Cloud Platform for Database in the Cloud
- Oracle Autonomous Cloud Platform
- Autonomous Database: Overview
- Difference between User Managed and Autonomous

ORACLE

Oracle Cloud Infrastructure (OCI)



- Oracle Cloud Infrastructure (OCI) takes advantage of high-scale, high-bandwidth networks that connect cloud servers to high performance local, file, block, and object storage to deliver a cloud platform that yields the highest performance for traditional and distributed applications, as well as highly available databases.
- OCI offers the ability to run everything from small VMs to large bare metal clusters and highly available databases on the same isolated networks, accessible through the same APIs and console – allowing apps to have direct, low-latency access to high-performance DBs running on physical servers in the same infrastructure.



What we have built for Infrastructure as a service is a technology platform consisting of the compute, storage and network resources that provide not only the benefits of cloud – primarily the benefits of pay as you go and fast provisioning of servers and storage – but also retains many of the benefits of the corporate data center.

This is a key differentiator for Oracle – the cloud offers from many of today's incumbents does not provide the level of performance, governance, control and integration that can be had from Oracle's infrastructure as a service offer.

All this is available on a platform that is optimized for Oracle Databases, at the best available price performance.

Specifically Architected for Oracle Databases

- Oracle is the only vendor that allows you to run clustered databases on bare metal servers, providing performance that typically exceeds on-premises deployments.
- Benchmarking results show Oracle Databases run way faster on Oracle Cloud Infrastructure when compared to others.
- Deep integration with Oracle Exadata provides high performance, availability and scalability for performance-intensive workloads.

OPTIMIZED FOR ORACLE DATABASES

Best available performance for Oracle Databases

ORACLE

OCI has a number of architectural design optimizations that are geared to run Oracle's Databases with unmatched scalability and reliability. For example, Oracle Cloud Infrastructure is optimized for Oracle 18c, Oracle Real Application Clusters (RAC), as well as deep integration with Oracle Engineered Systems such as Oracle Exadata.

Optimized to run Oracle Databases, benchmarking results show Oracle Databases run faster than others, so customers get data driven results faster and at a lower cost with Oracle.

As a leader in data management and technology, innovations such as Oracle Real Application Clusters (RAC), Exadata, and Active Data Guard (ADG) for Oracle DB deliver scale, availability, and reliability that are requirements for high performance applications – these features aren't available at our competition, where scaling up database CPUs requires down time, and can interrupt business operations and ultimately, the end-user experience.

Oracle Cloud Infrastructure: Database Service

- Mission-critical enterprise-grade cloud database service with comprehensive offerings to cover all enterprise database needs
  - VM, Bare Metal, Exadata
- Complete life cycle automation
  - Provisioning, patching, backup, restore, clone, replicate (complete flexibility)
- High availability and scalability
  - Robust infrastructure
  - Robust database options
  - Dynamic CPU and storage scaling
- Security
  - Infrastructure (IAM, security lists, audit logs)
  - Database (Transparent Data Encryption)

ORACLE

OCI and OCI Database Services

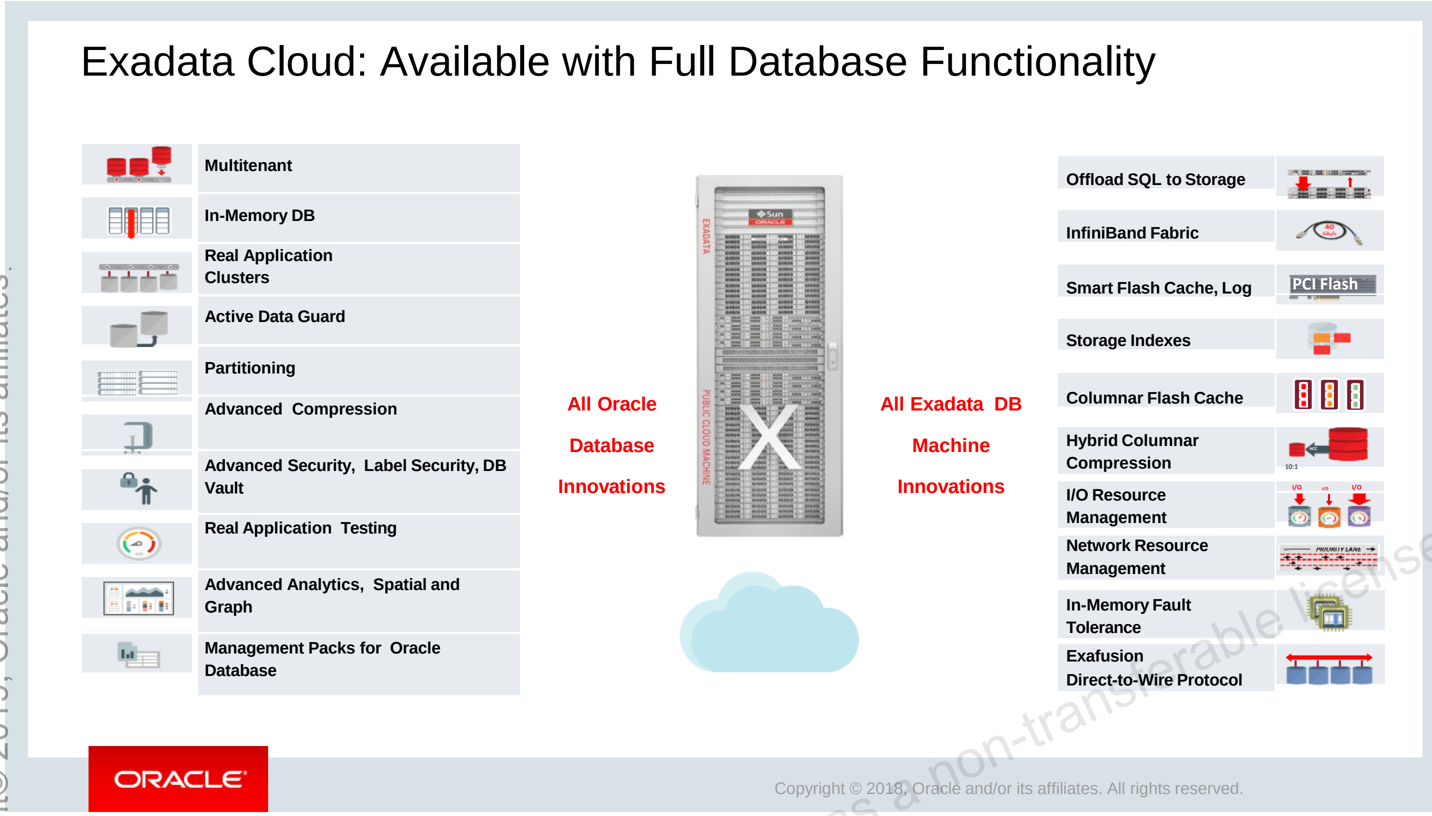
OCI database service runs on top of OCI, which specializes in bare metal servers, off-box networking (which accommodates any workload, engineered system, VM, or bare metal host all on the same network) and high speed storage.

OCI has a robust infrastructure

- Three Availability Domains: Multi-region architecture, currently three with a fourth region being brought up
- Fully redundant and nonblocking networking fabric accommodating up to 2 * 25 Gbe networking to the hosts
- Three-way mirrored storage (optional Two-way mirroring) for database systems. Disk management set up by OCI Database Service is according to best practices, so ASM comes pre-configured for each of the shapes.
- Redundant InfiniBand fabric for cluster networking (Exadata, 2 Node, bare metal, RAC)

Robust Database Options

- Database RAC Option for both VM and bare metal shapes
- Automatic backups to object storage are set up for users when the database is started. Automated DataGuard configuration for both primary and standby system (available within the AD and across AD).
- All the systems are created so that they follow the Maximum availability architecture (MAA). This is considered a certified deployment.



Decades of Database Innovation Proven at Millions of Mission-Critical Deployments

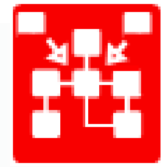
OCI Exadata systems have all of the features of Exadata enabled. There is no disadvantage to running the database on OCI versus on-premises. In addition to all of the advanced features, setup and following MAA and fully supported backups to object storage are trivial, fast, and relatively inexpensive. You can avail of the best of both implementations: the flexibility of cloud with all of the features of Exadata.

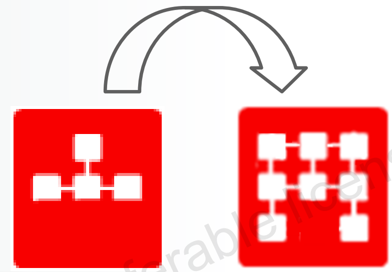
- You get:
  - Full Oracle Database with all database options: enterprise extreme performance and all Exadata features
  - Physical Exadata: Scalable, highly available, robust (in between shape scaling available)
  - Cloud-based pricing, with Oracle cloud-based provisioning and fully supported
- Operators have full control of their database.
- Available at Oracle's OCI data centers:
  - Powered by Oracle Cloud Infrastructure: Networking is non-blocking fabric and has off-box network virtualization, so complete isolation and line rate is the rule.
  - Oracle manages all Exadata and data center infrastructure; no firmware or OS patching by customers needed.
- Ideal customer profile
  - Need for a consumption-based cloud strategy; Exadata can be consumed monthly.
  - Use cases and workload identified to migrate to a public cloud
  - Flexible with data residency: As long as a copy or primary can be off site, everything can be moved to OCI. All storage and data at rest is encrypted.

Scaling Exadata DB Systems

Two ways of scaling Exadata DB Systems:

- **Scaling within an Exadata DB System:** Lets you modify compute node processing power within the system
  - Can be done without disruption
  - Can be accomplished by the customer
- **Scaling across Exadata DB System configurations:** Lets you move to a different configuration (for example, from a quarter rack to a half rack)
  - Requires movement of database deployment
  - Planned and executed in coordination with Oracle





Options for scaling the Exadata DB systems on Oracle Cloud Infrastructure:

- **Scaling within:** You can scale up the number of enabled CPU cores in the system if an Exadata DB system requires more compute node processing power. Just modify the number of enabled CPU cores.
- **Scaling across:** Exadata DB System configurations enable you to move to a different system configuration. This is useful when a database deployment requires:
  - Processing power that is beyond the capacity of the current system configuration
  - Storage capacity that is beyond the capacity of the current system configuration
  - A performance boost that can be delivered by increasing the number of available compute nodes
  - A performance boost that can be delivered by increasing the number of available Exadata storage servers

  Scaling from a quarter rack to a half rack, or from a half rack to a full rack, requires that the data associated with your database deployment is backed up and restored on a different Exadata DB system, which requires planning and a maintenance window, but can happen very quickly due to the speed of the underlying infrastructure and networking.

Database Service Patching

Database service patching simplifies the steps that are required to patch a fleet of OCI databases along with the underlying infrastructure including the database system, operating system, and drivers.

- **Automated Applicable Patch Discovery:** It enables automatic patch discovery, pre-flight checks/tests, rollback support.
- **Patch Scheduling:** Set maintenance windows, automate patching, and allow critical patch automation.
- **Patch Notifications:** In addition to Console patch notifications, email can be used to alert when patches become available.
- **Identity and Access Controls:** Granular Permissions: It's possible to control who can list patches, apply them, and so on.
- **Availability during patching:** For Exadata and RAC shapes, patches are rolling. For single-node systems, if Active Data Guard is configured, this can be leveraged by the patch service.

Oracle regularly makes patches available for various reasons such as feature upgrade, security fixes, or to fix a problem.

There are different types of patches available.

- Some of these patches have a regular quarterly release cadence (for example, PSUs or Bundle Patches) while others are shared with an operator for a specific problem (known as one-off or interim).
- Patches can be for the underlying operating system, hardware drivers, DB System components (for example, Grid Infrastructure) and DB Home components. Operators use utilities such as OPatch to view and apply patches.
- Database service patching simplifies the steps that are required to patch a fleet of OCI databases along with the underlying infrastructure including the database system, operating system and hardware drivers.
- Operators no longer have to go through the process of searching and manually installing relevant patches on their OCI database host.
- Database service patching will list all patches relevant to the operator's database fleet and the underlying infrastructure.
- Operators just have to select the patches they want to apply.
- Optionally, operators can run prechecks to test patch compatibility with the target component and, if required, roll back patches.
- Operators can also schedule patching jobs to run during a maintenance window they define.
- The database service can also be set up to "auto apply" quarterly patches or critical security patches as and when they are released.

Database Service High Availability: DataGuard

OCI Database Cloud Service offers a high availability configuration that is built on top of Oracle DataGuard, but offers simple steps to configure and manage DataGuard using the OCI Console and APIs.

- **Simplified HA deployment:** Easily deploy HA configurations of DataGuard within a region or across regions
- **Oracle Certified:** Follows HA configuration adhering to Oracle's standard best practices for high availability as recommended by the Maximum Availability Architecture (MAA)
- **Flexible Data Protection modes:** Three protection modes offered by Oracle DataGuard
- **Reduce DBA overhead:** Offers a completely managed HA experience: deploying the HA configuration across ADs, provide simple actions to easily switchover or failover databases without the need to log in to the hosts for management
- **Managed Failover:** Automatic managed failover capable
- **IAM Integrated:** Granular access controls for monitoring, configuring, and manipulating DataGuard

ORACLE

Oracle DataGuard ensures high availability, data protection, and disaster recovery for enterprise data.

Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions.

Data Guard maintains these standby databases as transactionally consistent copies of the production database. Then, if the production database becomes unavailable because of a planned or an unplanned outage, DataGuard can switch any standby database to the production role, minimizing the down time associated with the outage. DataGuard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability.

With DataGuard, administrators can optionally improve production database performance by offloading resource-intensive backup and reporting operations to standby systems.

OCI Database systems have a managed configuration capability for DataGuard as well as having all the DataGuard tools available for operators to construct custom configurations.

The following are some fundamental advantages of Oracle Data Guard technology:

- **Complete Data Protection:** Data Guard can ensure no data loss, even in the face of unforeseen disasters. A standby database provides a safeguard against data corruption and user errors.
- **Automatic Gap Detection and Resolution:** If connectivity is lost between the primary and one or more standby databases (for example, due to network problems), redo data being generated on the primary database cannot be sent to those standby databases. After a connection is re-established, the missing archived redo log files (referred to as a gap) are automatically detected by DataGuard and applied.

Migrating Databases to Oracle Cloud Infrastructure

Migrate your existing Oracle Database to an Oracle Cloud Infrastructure DB system using a number of different methods that use several different tools.

- **Migration Methods:** Export/import, Data Pump, remote cloning, RMAN, unplug/plug PDB, SQL Developer
- **Factors Considered During Migration:** Database versions, database upgrade (for example, 12*c* > 18*c*), Endian Changes, Character Set Matching, Database Multitenancy (CDB or non-CDB)
- **Connectivity:** Speed of connection between existing database and Oracle Cloud Infrastructure. Options include FastConnect, IPSec tunnels, and using an Internet Gateway.
- **Bulk Data Transfer**: If database sizes are very large and bandwidth is limited **Bulk Data Transfer Services** for OCI is an option. The bulk data transferred can be resynced reducing overall data transfer bandwidth used.

ORACLE

The method that applies to a given migration scenario depends on several factors, including the version, character set, and platform endian format of the source and target databases.

Not all migration methods apply to all migration scenarios. Many of the migration methods apply only if specific characteristics of the source and destination databases match or are compatible. Moreover, additional factors can affect which method you choose for your migration from among the methods that are technically applicable to your migration scenario.

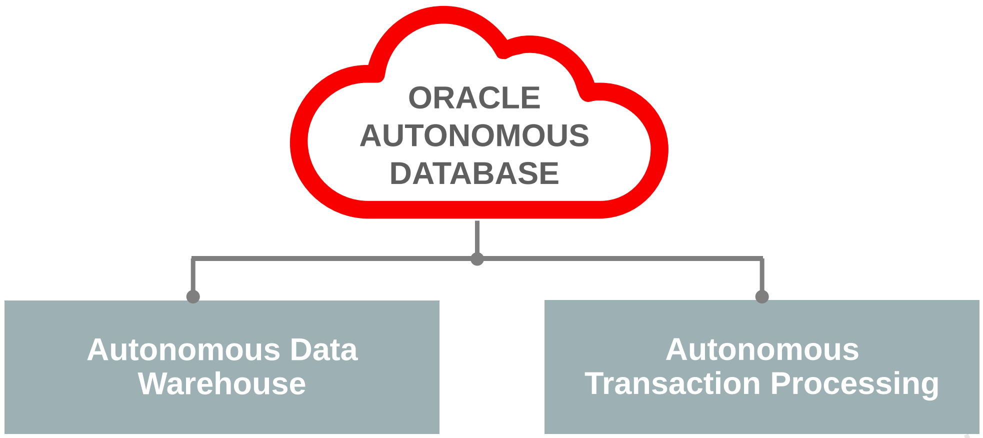
Some of the characteristics and factors to consider when choosing a migration method are:

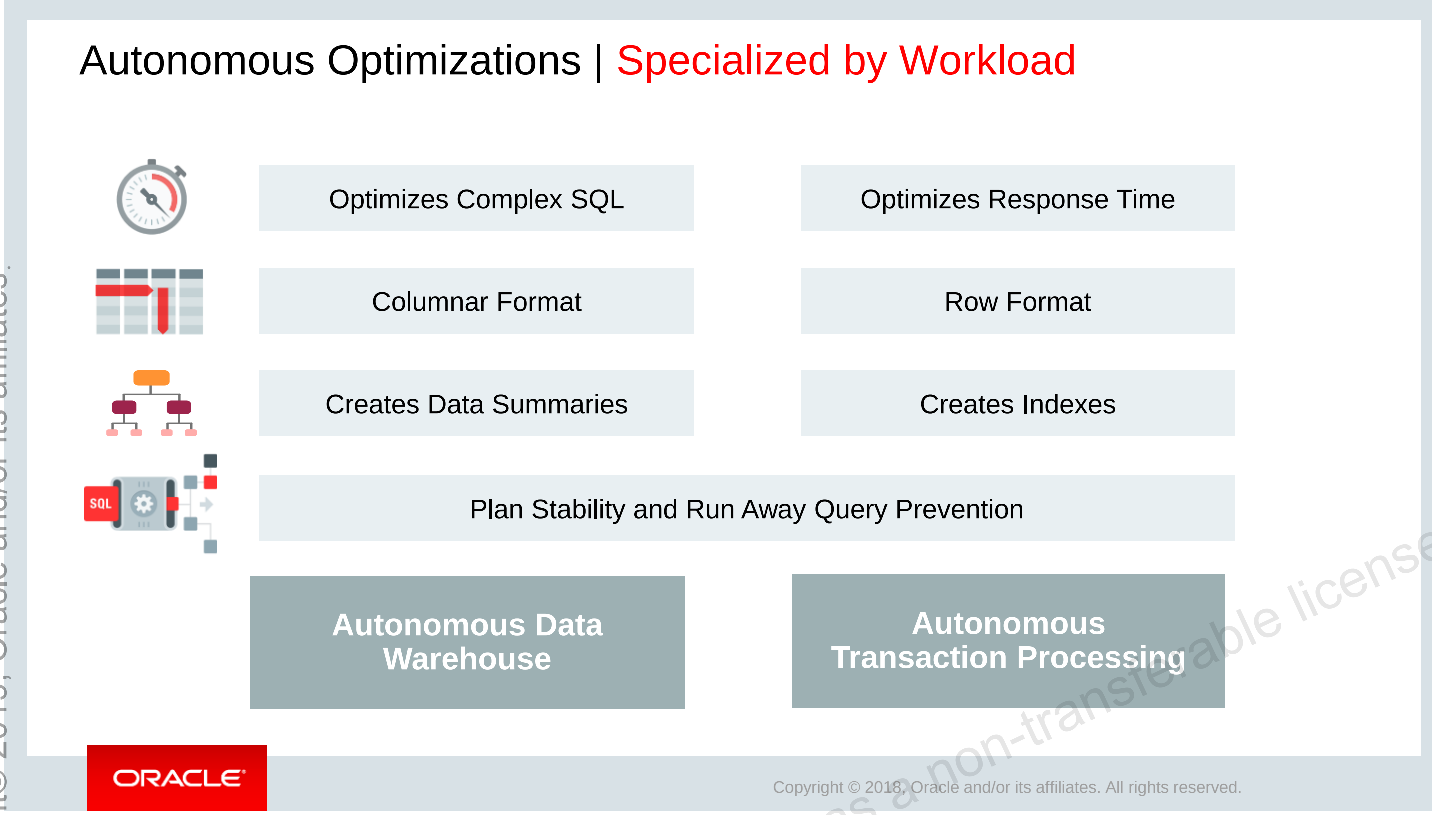
- Preference for certain tools: Import/export, Data Pump, RMAN, and so on.
- On-premises database version
- Database service database version
- On-premises host operating system and version
- On-premises database character set
- Quantity of data, including indexes
- Data types used in the on-premises database
- Storage for data staging
- Acceptable length of system outage
- Network bandwidth
- If transfer sizes are very large, bulk data transfer services can also help with cost and time.

Learning Agenda

- Database Cloud Service: Overview
- Oracle Cloud Platform for Database in the Cloud
- Oracle Autonomous Cloud Platform
- Autonomous Database: Overview
- Difference between User Managed and Autonomous

Autonomous Database | Optimized by Workload





Both ADW and ATP share the Autonomous Database platform of Oracle Database 18c on our Exadata Cloud infrastructure.

The difference is how the services have been optimized within the database. When you start loading data into the autonomous database, we store the data in the appropriate format for the workload.

- If it is ADW, then we store data in columnar format as that's the best format for analytics processing.
- If it is ATP, then we will store the data in a row format as that's the best format for fast single row lookups.

Query optimization

For analytics workload, we automatically parallelize the query execution to access large volumes of data in a short amount of time to answer biz questions. If it is a transaction processing system, then we will automatically detect missing indexes and create them for you.

Regardless of the workload, we need to keep optimizer statistics current to ensure we get optimal execution plans. With ADW we are able to achieve this by gathering statistics as part of all bulk load activities. With ATP, where data is added using more traditional insert statements, statistics are automatically gathered periodically.

As the data volumes change, or new access structures is created, there is the potential for an execution plan to change and any change could result in a performance regression so we use Oracle SQL Plan Management to ensure that plans only change for the better.

Autonomous Data Warehouse | Key Use Cases

Data Marts / Warehouses

Business Analytics

Sandboxes for Data Scientists

Machine Learning

Data Lakes

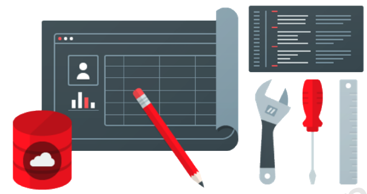
Query All Data

Autonomous Transaction Processing | Key Use Cases

Departmental / Mission Critical Applications

Mixed Workloads

Application Development



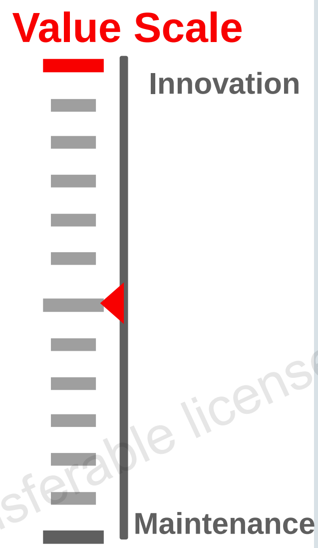
Learning Agenda

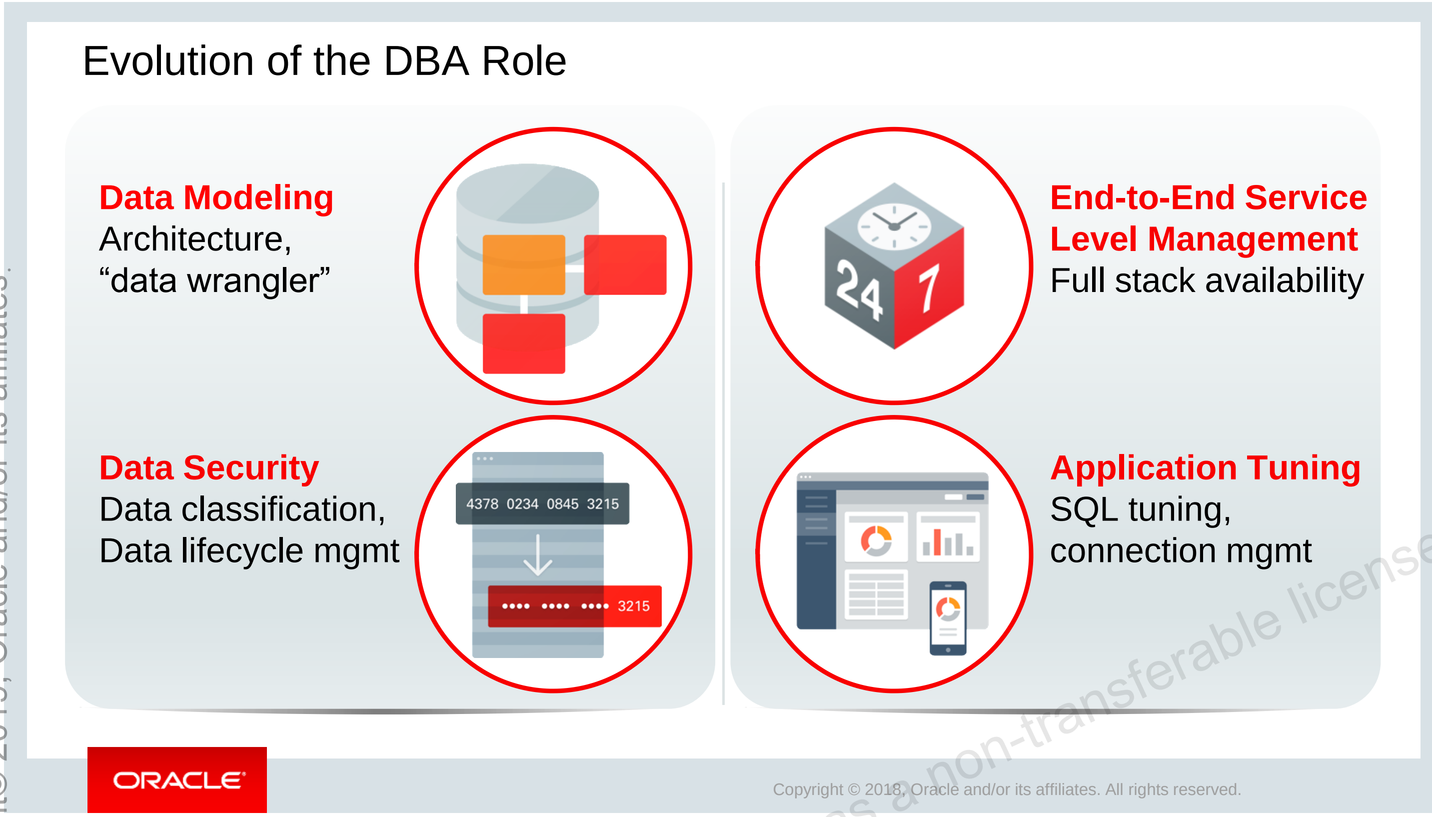
- Database Cloud Service: Overview
- Oracle Cloud Platform for Database in the Cloud
- Oracle Autonomous Cloud Platform
- Autonomous Database: Overview
- Difference between User Managed and Autonomous

Role of DBAs

Traditionally, DBAs are responsible for:

- **Tasks specific to business and innovation**
  - Architecture, planning, data modeling
  - Data security and lifecycle management
  - Application-related tuning
  - End-to-end service level management
- **Maintenance tasks**
  - Configuration and tuning of systems, network, storage
  - Database provisioning, patching
  - Database backups, H/A, disaster recovery
  - Database optimization





Your job is not going away but it will evolve, so you must change with it.

You will spend less time on generic maintenance, more time innovating.

More time with the business:

- Executing more projects, reducing backlog, getting more value from data
- Cloud’s fast provisioning and pay-as-you-go enables rapid experimentation.

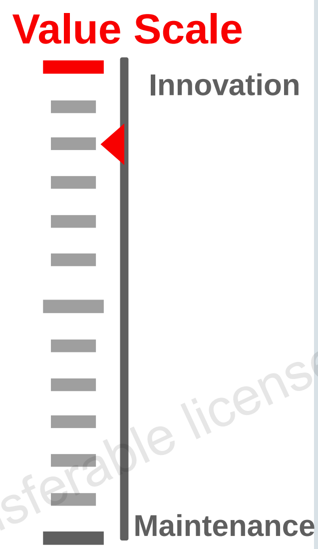
More time with developers

- Optimizing data access, improving end-user experience

Autonomous Database Removes Generic Tasks

Freedom from mundane tasks for DBA: More Time to Innovate and Improve the Business

- **Tasks specific to business and innovation**
  - Architecture, planning, data modeling
  - Data security and lifecycle management
  - Application related tuning
  - End-to-End service level management
- **Maintenance tasks**
  - Configuration and tuning of systems, network, storage
  - Database provisioning, patching
  - Database backups, H/A, disaster recovery
  - Database optimization



ORACLE

Oracle Autonomous Database: Introduction

New “Autonomous” Database ***Category*** and suite of associated ***Services***

- New Category
  - Oracle redefines the data management category
  - It sets new industry requirements for “Autonomous”: Self-Driving, Self-Securing, Self-Repairing
- New Suite of Cloud Services
  - Autonomous Data Warehouse Cloud
  - Autonomous Database Cloud for OLTP/Mixed Workloads

Oracle Autonomous Database: Features

Self-Driving – Lower Cost & Increased Productivity

- Eliminates human labor to provision, secure, monitor, back up, recover, troubleshoot; no manual tuning
- Automatically upgrades & patches itself while running. Testing automation ensures changes are safe.
- Elastically grows and shrinks compute or storage without down time. Pay only for what you use.

Self-Securing – Lower Risk

- Protection from external attacks & from malicious internal users
- Automatically applies security updates with no down time
- Automatic encryption of all data

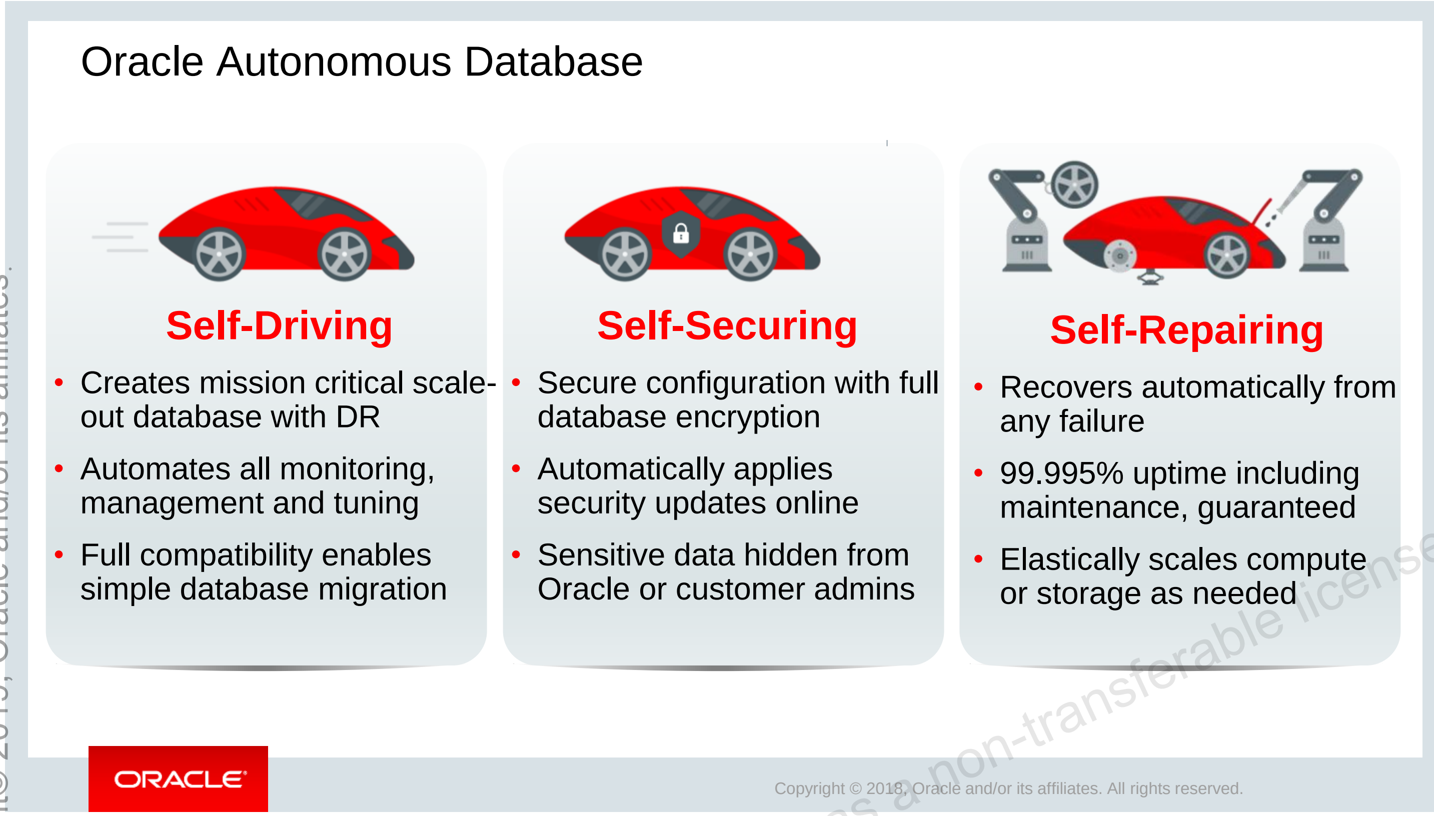
Self-Repairing – Higher Availability

- **Eliminates human error** through Automation

User defines policies, database makes them happen.

Always encrypted, always current on security updates

Automatically protects from all failures



Self-Driving – Lower Cost & Increased Productivity

- Eliminates human labor to provision, secure, monitor, backup, recover, troubleshoot, and tune the database
- Automatically upgrades and patches itself while running. Testing automation ensures changes are safe.
- Elastically grows and shrinks compute or storage without down time. Pay only for what you use.

Self-Securing – Lower Risk

- Protection from external attacks and from malicious internal users
- Automatically applies security updates with no down time
- Automatic encryption of all data

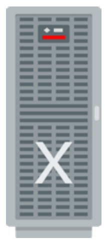
Self-Repairing – Higher Availability

- Automation eliminates administrator errors

Autonomous Database Machine Learning

Diagnostics, recovery and optimizations for each layer of the deployment stack

Database Infrastructure



Detection and recovery of failed/sick server, storage or switch/link

Database Operations

Hang Management
Anomaly Detection
Maintenance Slot Identification
Bug Identification and Prioritization

Workload Optimizations

Query Optimizer
Real-time statistics
Automatic Indexing

ORACLE

Oracle Autonomous Database

Total Automation Based on Machine Learning

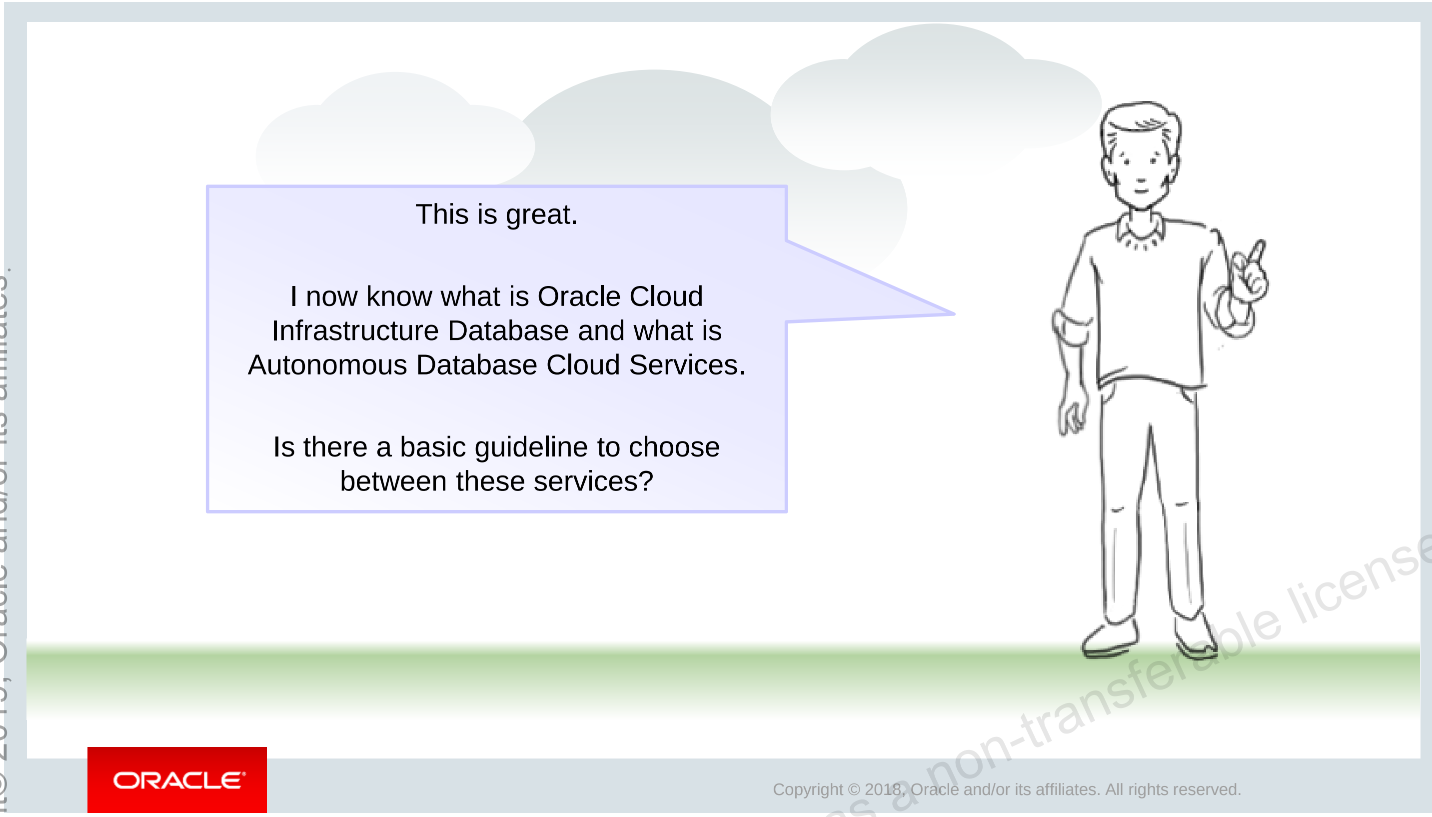
- No Human Labor: Eliminates human labor to manage the database
  - Database automatically provisions, upgrades, patches, tunes itself while running.
    - Automated real-time security patching with no downtime window required.
- No Human Error: SLA guarantees 99.995% reliability and availability
  - Minimizes costly planned plus unplanned down time to less than 2.5 mins a month
- No Human Performance Tuning: Consumes less compute and storage than at Amazon
  - Lower labor costs is an even bigger savings.

ORACLE

Learning Agenda

- Database Cloud Service: Overview
- Oracle Cloud Platform for Database in the Cloud
- Oracle Autonomous Cloud Platform
- Autonomous Database: Overview
- Difference between User Managed and Autonomous

ORACLE


This is great.
I now know what is Oracle Cloud Infrastructure Database and what is Autonomous Database Cloud Services.
Is there a basic guideline to choose between these services?
ORACLE
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Autonomous or User Managed?

User Managed

- Automated with human intervention to take control in a customized environment for tuning to meet very specific business requirements
- Need to have complete operational control, including OS access, full DBA privileges

Autonomous

- Decision making, performing one or more tasks automatically

Example: Race Car Track

ORACLE

What Autonomous Database Means for Developers

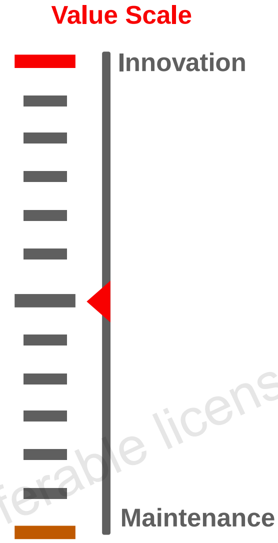
- Rapid provisioning, online elastic scaling
  - Lowest time to usage (TTU)
- No IT support required for provisioning, tuning, patching and upgrading
- One database service for all your data needs:
  - Relational, JSON, Spatial, Graph, Text, and so on
  - Industry standard compliant; ISO SQL, JDBC, PEP249, and so on
  - RESTful data access
  - APEX for low-code application development

ORACLE

What Autonomous Database Means for DBAs

- **Tasks specific to the business**
  – Architecture, planning, data modeling
  – Data security and data lifecycle management
  – Application related tuning
  – End-to-end service level management

- **Tactical operations**
  – Configuration and tuning of systems, network, storage
  – Database provisioning, patching
  – Database backups, H/A, disaster recovery
  – Database optimization

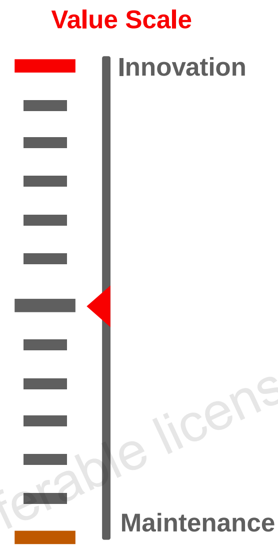


ORACLE

What Autonomous Database Means for DBAs

Removes tactical drudgery, more time to innovate

- **Tasks specific to the business**
  - Architecture, planning, data modeling
  - Data security and data lifecycle management
  - Application-related tuning
  - End-to-end service level management
- **Tactical operations**
  - Configuration and tuning of systems, network, storage
  - Database provisioning, patching
  - Database backups, H/A, disaster recovery
  - Database optimization



New Projects or Applications

- **Simplicity and faster time-to-market** for **NEW** applications
  - All new applications should go with Autonomous Database.
- Autonomous Database:
  - **Eliminates** dependence and delays on others for servers, storage, and databases
  - **Eliminates** database tuning, auto adapts to changing workload
  - Provides advanced SQL and PL/SQL capabilities to accelerate developer productivity

ORACLE

Summary: Oracle Autonomous Database

Self-driving, self-securing and self-repairing cloud services

CIOs lower the cost and risk of their IT services

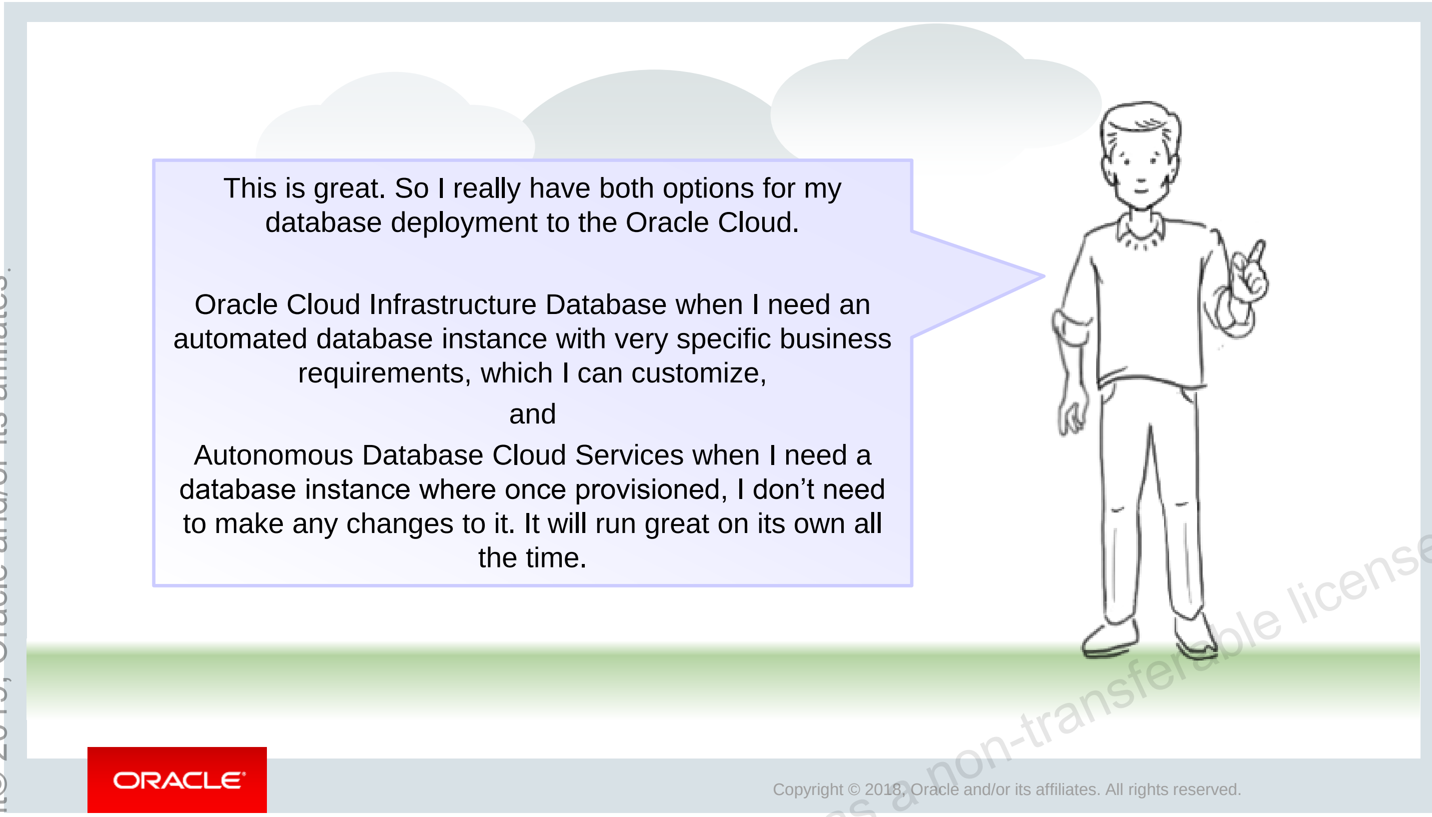
- Full-stack admin costs eliminated
- No more cyber attacks on unpatched or unencrypted databases
- Less than 2.5 minutes of down time a month including patching

Developers build and enhance applications faster

- With easy-to-use, self-tuning service, and no IT required
- Via 3GL coding and APEX rapid application development

DBAs become more valuable to the business

- With less time on mundane tasks and more time on tasks of high value to business


This is great. So I really have both options for my database deployment to the Oracle Cloud.
Oracle Cloud Infrastructure Database when I need an automated database instance with very specific business requirements, which I can customize,
and
Autonomous Database Cloud Services when I need a database instance where once provisioned, I don't need to make any changes to it. It will run great on its own all the time.
ORACLE
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Describe the offerings for Oracle Database Cloud Service
- Describe the Oracle Cloud Platform for Database in the Cloud
- Describe Oracle Autonomous Cloud Platform
- Describe the features of Autonomous Database
- Explain the difference between User Managed and Autonomous Database Service