ORACLE
University

Integrated Cloud Applications & Platform Services

Oracle Autonomous Database Workshop

Student Guide

D105098GC10 | D105320

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

ORACLE

Authors

Soumyadeep Mitra

Aravind Gidijala

Technical Contributors and Reviewers

Susan Jang

Sailaja Pasupuleti

Sandeep Arora

Sharath Bhujani

Editors

Aju Kumar

Moushmi Mukherjee

Nikita Abraham

Arijit Ghosh

Graphic Editor

Prakash Dharmalingam

Publishers

Michael Sebastian Almeida

Asief Baig

Sumesh Koshy

1005022019

Contents

1

Course Overview

Introduction

Course Objectives

After completing this course, you should be able to:

- Identify Oracle Database Cloud Service offerings
- List the differences and benefits of User Managed and Autonomous Database Services on Oracle Cloud
- Describe the essentials of Oracle Cloud Infrastructure
- Identify available workloads in Oracle Autonomous Database Cloud Service
- Create and manage an Autonomous Database instance

Objectives

After completing this lesson, you should be able to:

- Provide an overview of the topics covered
- List the prerequisites for this course

Target Audience

This course is mainly intended for:

- Database Administrators
- Database Developers
- DevOps Developers
- Cloud Architects

ORACLE

Prerequisites

To successfully complete this course, you should have a:

- General understanding of cloud technology
- Basic understanding of Oracle Database and related tools

Suggested prerequisites:

- Working knowledge of administering Oracle Database

Course Roadmap

1. Course Overview
2. Oracle Cloud Platform for Database in the Cloud
3. Getting Started with Oracle Cloud Infrastructure Essentials
4. Oracle Autonomous Database Overview
5. Provisioning an Oracle Autonomous Database Instance
6. Connecting to Autonomous Database
7. Data Loading and Querying External Data with Autonomous Database
8. Managing Users in Autonomous Database
9. Using Oracle Machine Learning with Autonomous Database
10. Monitoring and Managing Autonomous Databases
11. Backing Up and Restoring Autonomous Databases
12. Starting, Stopping, Scaling, and Terminating an Autonomous Database Instance
13. Oracle Cloud Infrastructure Security
14. Managing Autonomous Database Using Command Utilities and APIs
15. Migrating Oracle Databases to Autonomous Database: Overview
16. Oracle Autonomous Database: Use Cases

This course is organized into 16 lessons. Each lesson has an associated activity guide that allows students to put their learning into practice.

Course Practices

- Lessons are reinforced with hands-on practices.
- Your practice environment consists of:
  - Your local system, i.e. a laptop or a desktop with a Windows 64-bit Operating System
  - An Oracle Cloud account assigned to you as part of the course environment using which you will create the required Oracle Cloud service instance

Note: You should be on an open internet connection, i.e. not connected to any VPN or working in a restricted network that blocks the connection from your local system to cloud service instances.

ORACLE

Summary

In this lesson, you should have learned how to:

- Provide an overview of the topics covered
- List the prerequisites for this course

ORACLE

Practice 1: Overview

There are no practices for this lesson.

2

Oracle Cloud Platform for Database in the Cloud

ORACLE

Objectives

After completing this lesson, you should be able to:

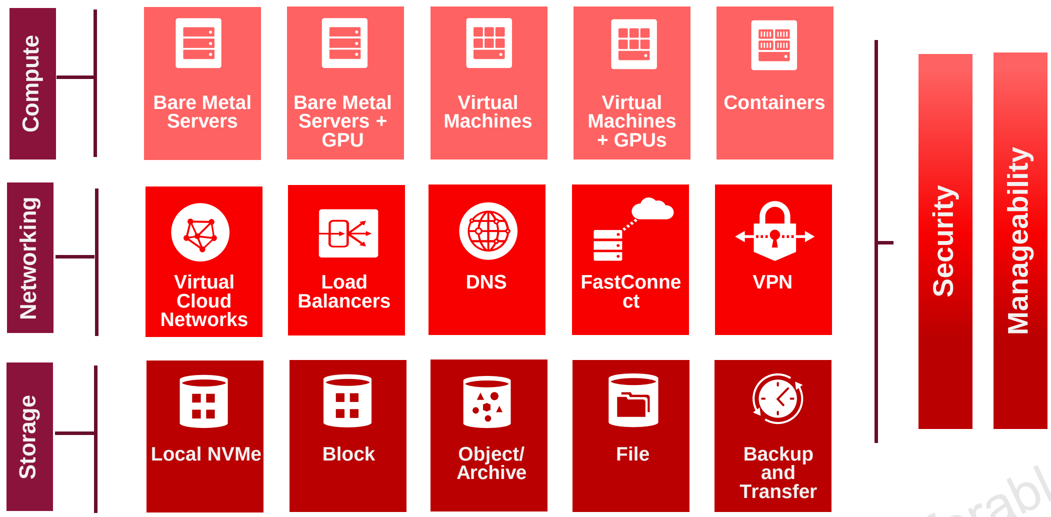
- List the offerings for Oracle Database Cloud Service
- Describe Oracle Cloud Platform for Database in the Cloud
- Describe Oracle Autonomous Cloud Platform
- Explain the difference between User Managed and Autonomous Database Service
- Discuss Oracle Cloud subscription models

ORACLE

Oracle Database Cloud Services

- Oracle offers database cloud services in a range of public cloud deployment choices:
  – Fully managed Autonomous Database 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 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, thereby allowing apps to have direct, low-latency access to high-performance DBs running on physical servers in the same infrastructure.

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 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
  – Virtual Machine (VM), Bare Metal (BM), Exadata
- Complete life cycle automation
  – Provisioning, patching, backup, restore, clone, replicate (complete flexibility)
- High availability and scalability
  – Robust infrastructure
  – Powerful 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 preconfigured for each of the shapes.
- Redundant InfiniBand fabric for cluster networking (Exadata, 2 Node, bare metal, RAC)

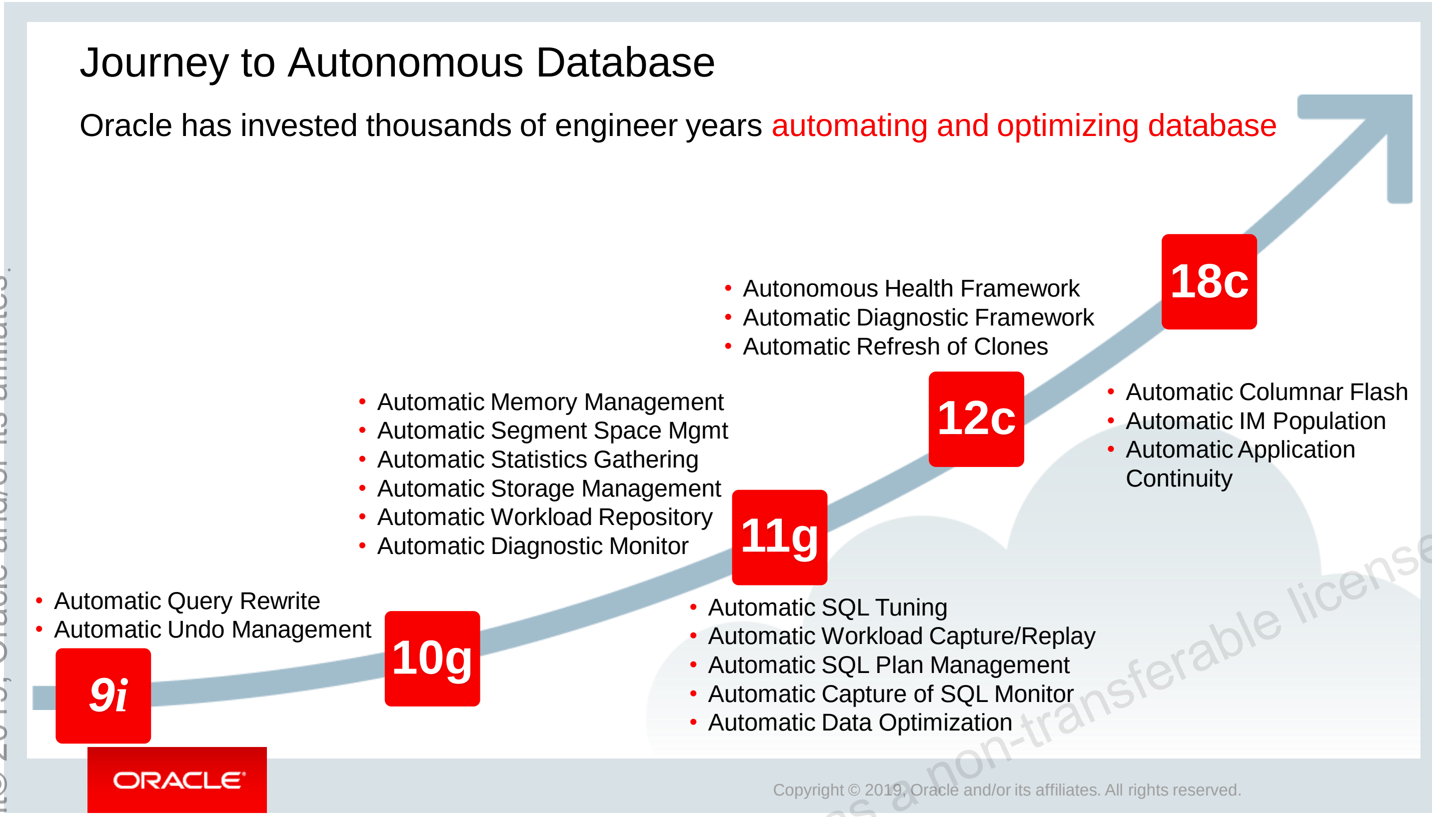
Powerful Database Options

- 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 systems (available within the AD and across AD).
- All the systems are created so that they follow Maximum Availability Architecture (MAA). This is considered a certified deployment.

OCI Security Features: Overview of Database Service

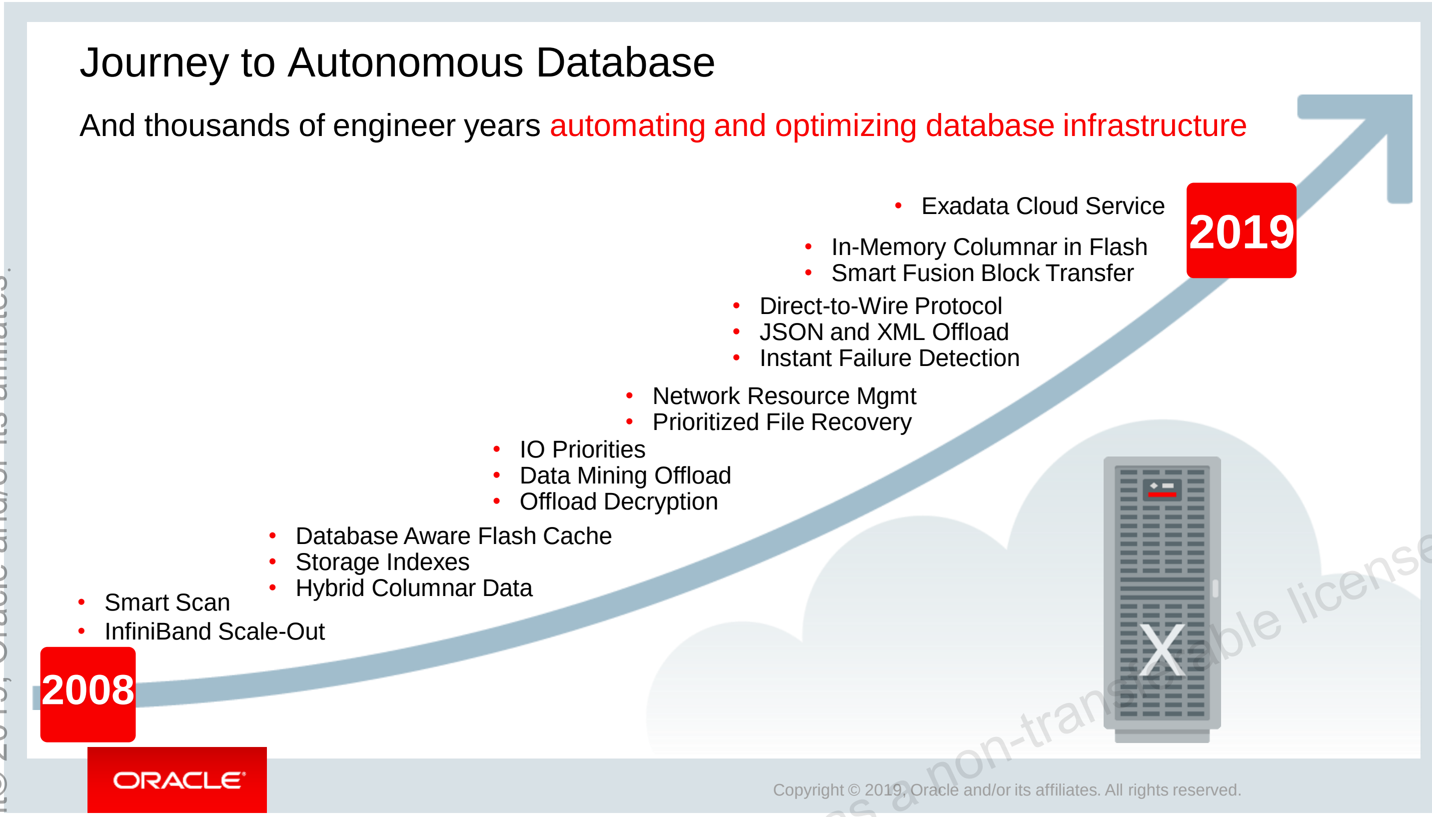
No.	Security capability	OCI DBCA security feature
1	Instance security isolation	OCI bare metal (BM) instance
2	Network security and access control	VCN, VCN security lists, VCN public and private subnets, VCN route table
3	Secure and highly-available connectivity	VPN DRGs
4	User authentication and authorization	IAM tenancy, compartments and security policies, console password, API signing key, SSH keys
5	Data encryption	DBaaS TDE, RMAN encrypted backups, storage and object encryption at rest
6	End-to-end TLS	LBaaS with TLS1.2, customer-provided certificates
7	Auditing	OCI API audit logs

Security features for the OCI database service include instance security isolation, network security and total isolation, IPSEC VPNs and FastConnect dedicated network circuits, granular controls for users, secure methods for API, and SSH access. We also have TDE on by default, with all backups and block and object storage encrypted by default. We will also look at the OCI load balancer, which supports encryption and auditing via OCI audit logs.



That's quite a tall order for a new product until you realize that we have been on this journey for over 20 years.

Starting with Oracle Database 9*i*, we began to introduce and mature many sophisticated automation capabilities from memory management to workload monitoring and tuning, all of which are used in the autonomous database.

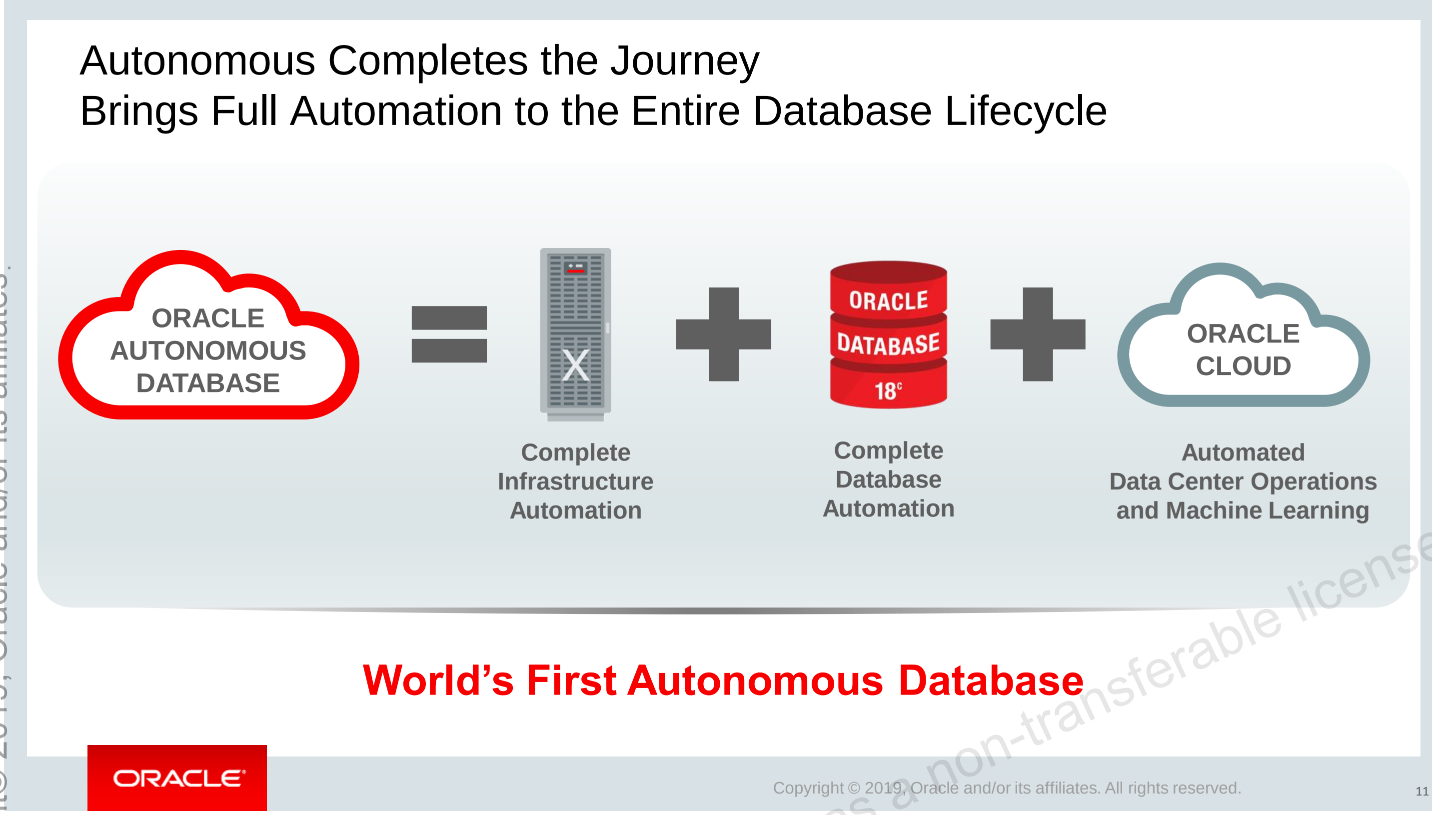


But it's not just the database management that Oracle has been automating. We have also spent the last decade working on the database infrastructure with our engineered systems, which provide the best platform for the Oracle database as they are the only preconfigured, pre-tested, and optimized platforms for the database.

Automatic or Autonomous?

	Automatic	Autonomous
 Autonomous Car	Examples: • Cruise control • Emergency stopping • Warnings for lane changes	• No need to use the steering wheel or brake. • Simply tell the car where you are going.
 Autonomous Database	Examples: • Automatic storage management • Automatic workload repository • SQL Plan Management	• All features are automatically implemented. • Simply tell the database your goals.

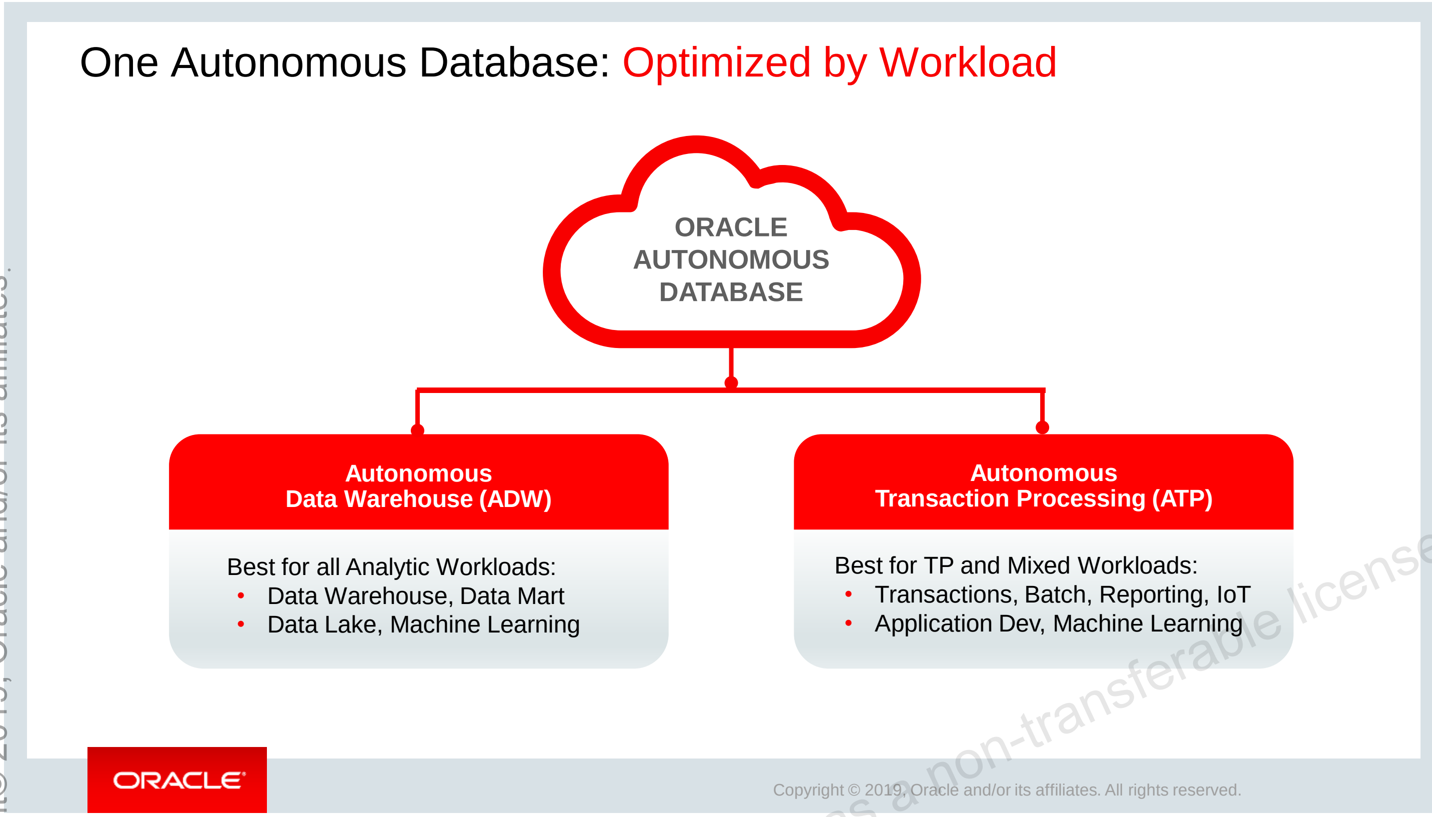
ORACLE



Oracle Cloud allows us to complete the journey to the autonomous database as it is the integration of our complete infrastructure automation with Oracle Database 18c, and our fully automated data center operations.

Automated data center operations include:

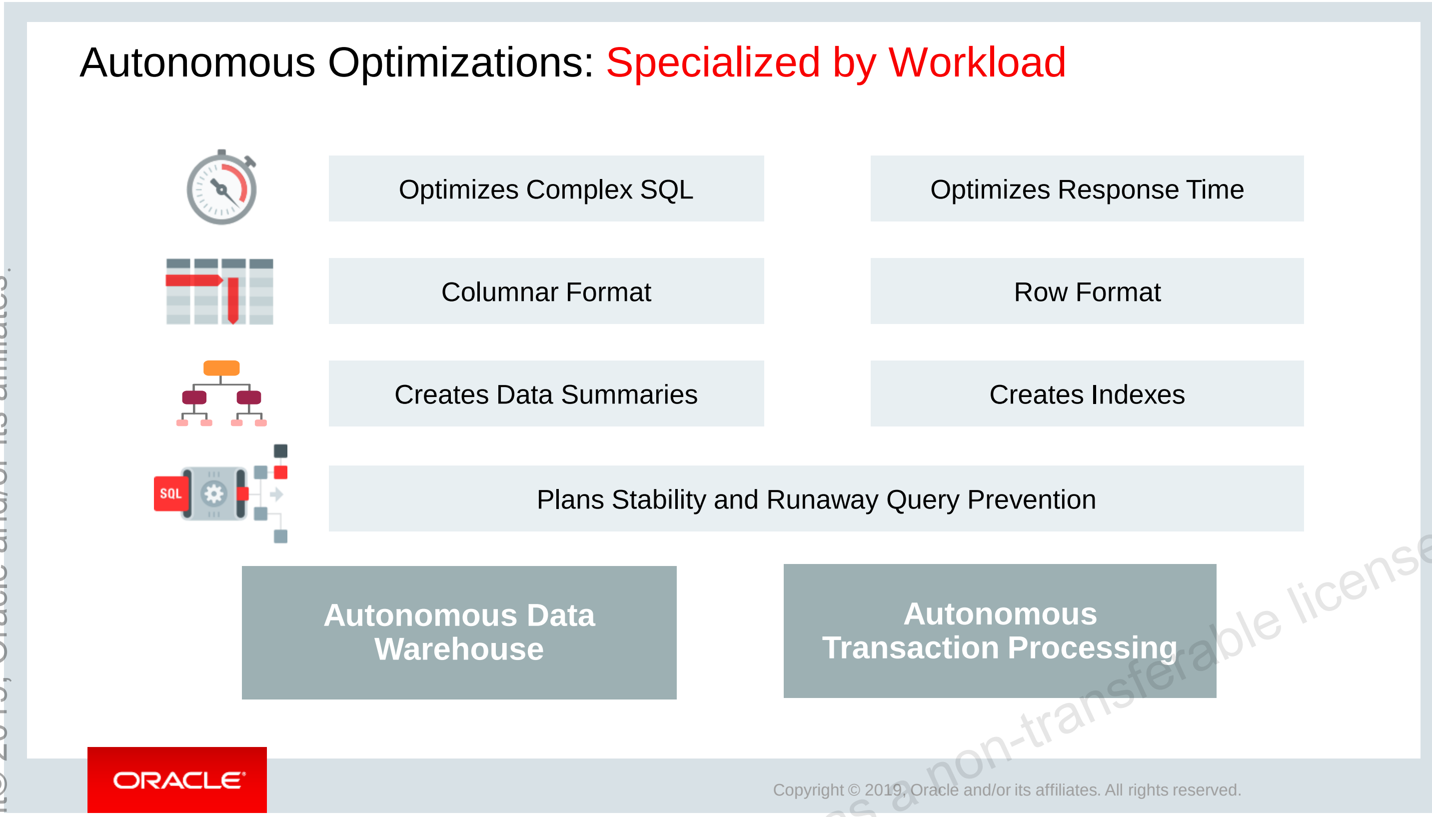
- Provisioning, patching, upgrading, and online backups
- Monitoring, scaling, diagnosing performance, tuning, optimizing
- Testing and change management of complex applications and workloads
- Automatically handling failures and errors



Oracle Autonomous Database is actually a family of cloud services, with each member of the family optimized by workload. The first member of the family to become available in March this year was the Autonomous Data Warehouse (ADW), which has been optimized for analytic workloads, such as data warehouse, data marts, or as part of a data lake.

The second member of the family that's just become available is Autonomous Transaction Processing (ATP).

ATP is optimized for transaction processing or mixed workload environments and makes an excellent platform for new application development.



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 store the data in 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 business 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 are created, there is potential for an execution plan to change. Any change could result in a performance regression and so we use Oracle SQL Plan Management to ensure that plans only change for the better.

User Managed or Autonomous?

User Managed

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

Autonomous

- Decision making, performing one or more tasks automatically

Example: Race Car Track

ORACLE

Subscribing 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.
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
- Activating Other Types of Subscriptions:
  http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSGSG486

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

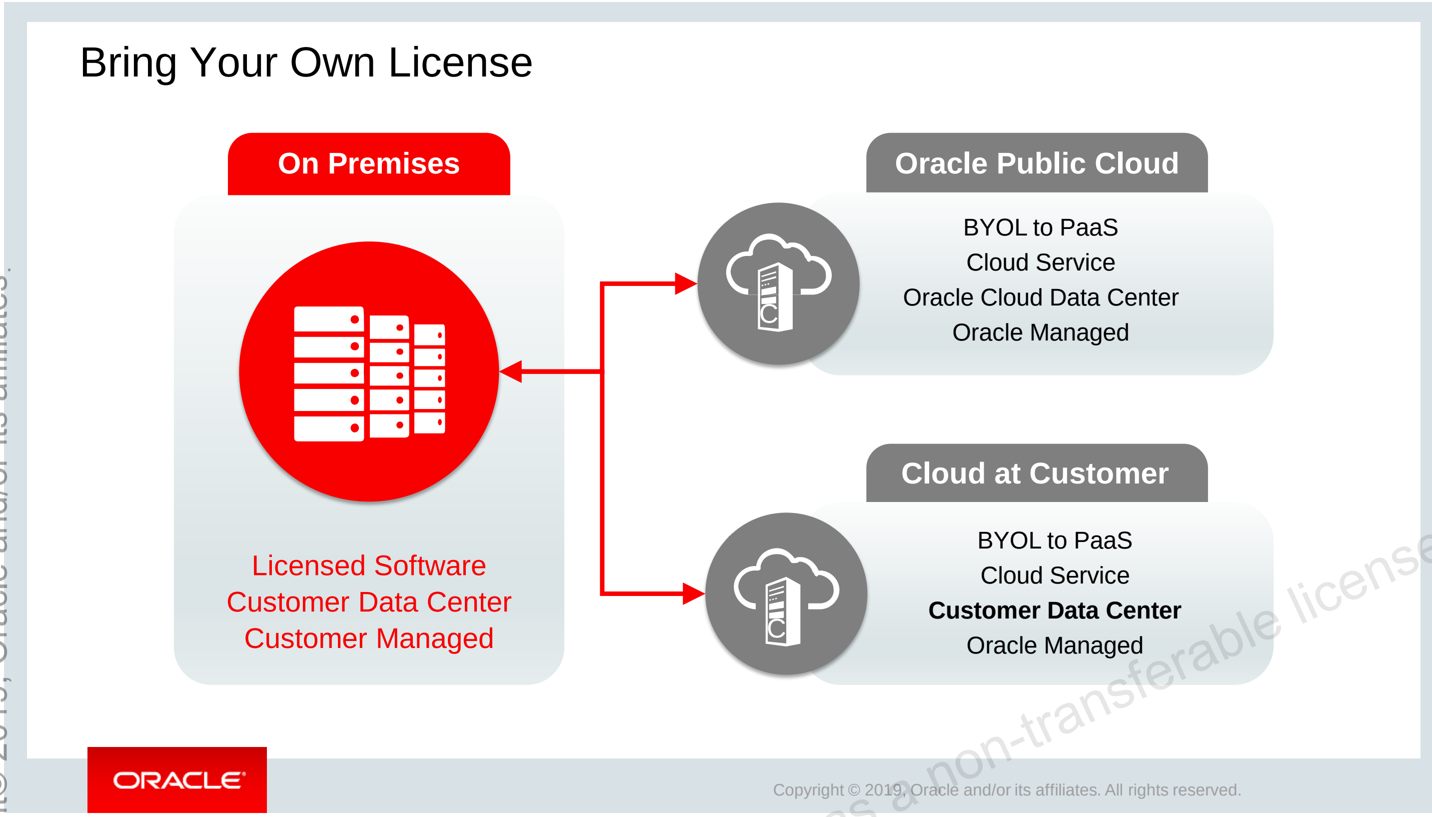
Pay As You Go	Universal Credits Monthly Flex
• No upfront commitment • Pay only for what you use • Pay in arrears based on usage	• 1 year minimum term • Based on the results of your monthly cost estimate
• List price • Built for land and expand • Best when usage is uncertain • Elastic payments based 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 a 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.

Summary

In this lesson, you should have learned how to:

- List the offerings for Oracle Database Cloud Service
- Describe Oracle Cloud Platform for Database in the Cloud
- Describe Oracle Autonomous Cloud Platform
- Explain the difference between User Managed and Autonomous Database Service
- Discuss Oracle Cloud subscription models

ORACLE

Practice 2: Overview

There are no practices for this lesson.

3

Getting Started with Oracle Cloud Infrastructure Essentials

ORACLE

Objectives

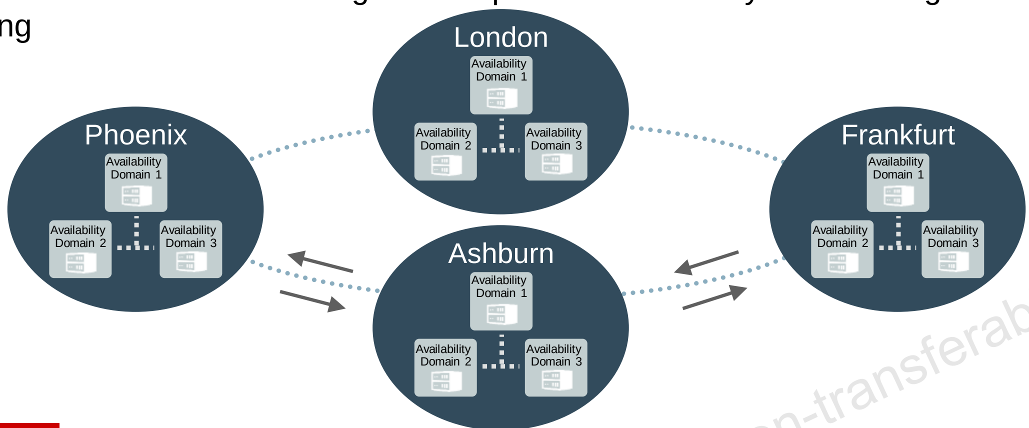
After completing this lesson, you should be able to:

- Describe the Oracle Cloud Infrastructure strategy
- Define key concepts and terminology
- Identify Oracle Cloud Infrastructure Services
- Get an overview of key Oracle Cloud Infrastructure Services:
  - Virtual Cloud Network (VCN) Service
  - Compute Service
  - Object Storage Service
  - Block Volume Service
  - Load Balancing Service
  - OCI DNS Services

In this lesson, we will describe the key concepts and terms used in the IAM service.

Regions, Availability Domains, and Backbone Network

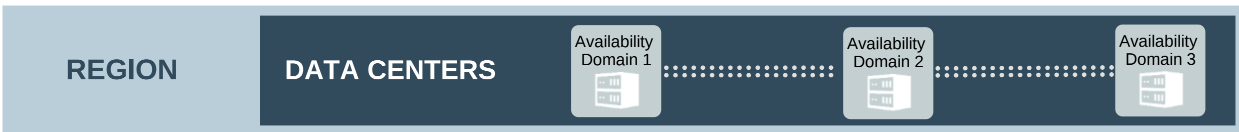
- Regions serve different geographies: Provide Disaster Recovery capabilities
- Availability Domains and Fault Domains: Provide an HA foundation within a region and an AD
- Backbone Network and Peering: Enable private connectivity between regions and direct peering



Let's start with a quick recap of our core concepts. We operate in regions. A region is a metropolitan area. Inside regions, we have multiple data centers that are called Availability Domains (ADs). Each region consists of three Availability Domains. These are isolated from each other and all your resources, like compute and database, go inside an AD. ADs are wired together over private dark fiber and there is very little latency between Ads, making it a perfect fit for a High Availability primitive replication of data. We have a dedicated backbone connecting these ADs. The backbone plugs into edge or peering points of presence where customers can get direct connections into our network.

Inside a Region: High Availability Building Blocks

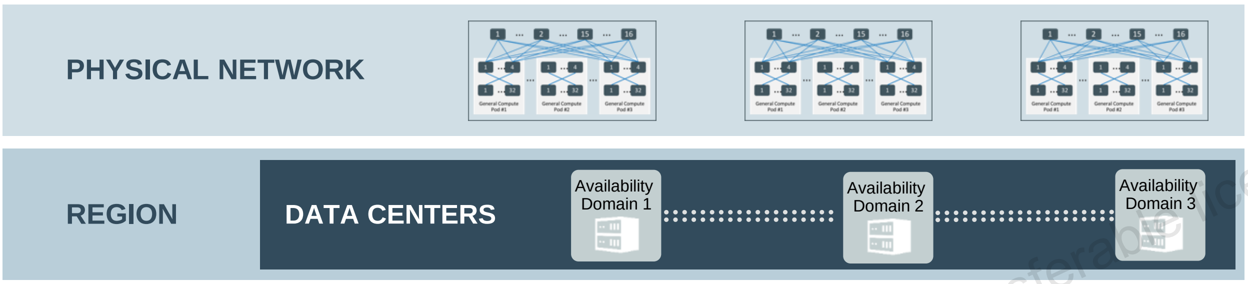
- Multiple fault-decorrelated, completely independent data centers: ADs
- Predictable low latency and high speed, encrypted interconnect between ADs
  - < 500µs expected one-way latency
- Enables zero-data-loss architectures (for example, Oracle MAA) and high availability scale-out architectures (for example, Cassandra)



Again, regions are constructed with isolated fault domains. You can see some latency numbers here: < 500 micro seconds one latency.

Inside an AD: High Scale, High Performance Network

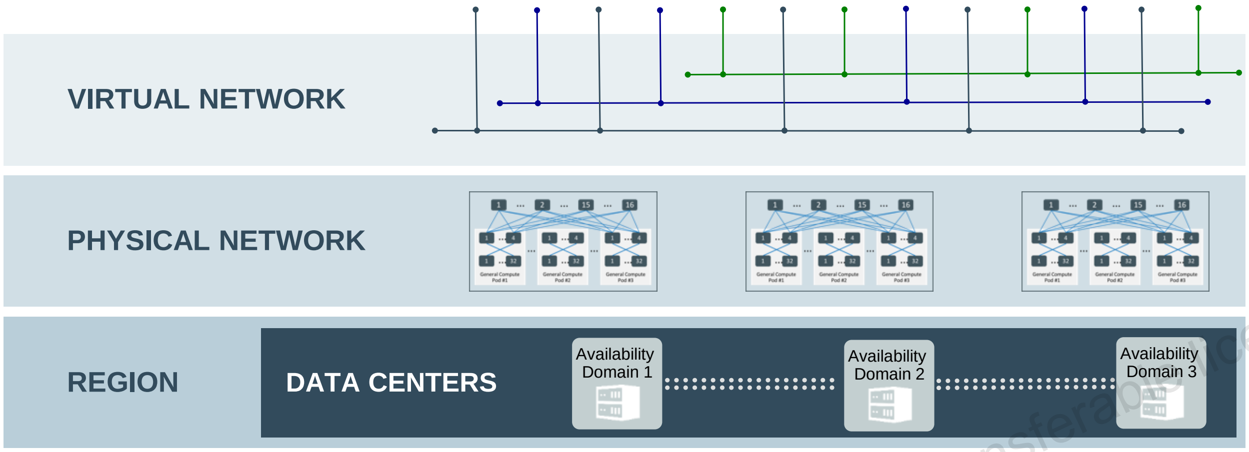
- Non-oversubscribed network: Flat, fast, predictable
- Very high scale: ~1 million network ports in an AD
- Predictable low latency and high speed interconnect between hosts in an AD
  - ~100µs expected one-way latency, 2 x 25Gb/s bandwidth



Inside these ADs, we have built one of the best public cloud networks. It is big, flat, and fast. It runs on a high scale to the tune of 1 million network works per AD. Flat means that it is not oversubscribed, so we get tremendously good latency. Fast means we support 25 Gbps network bandwidth between hosts.

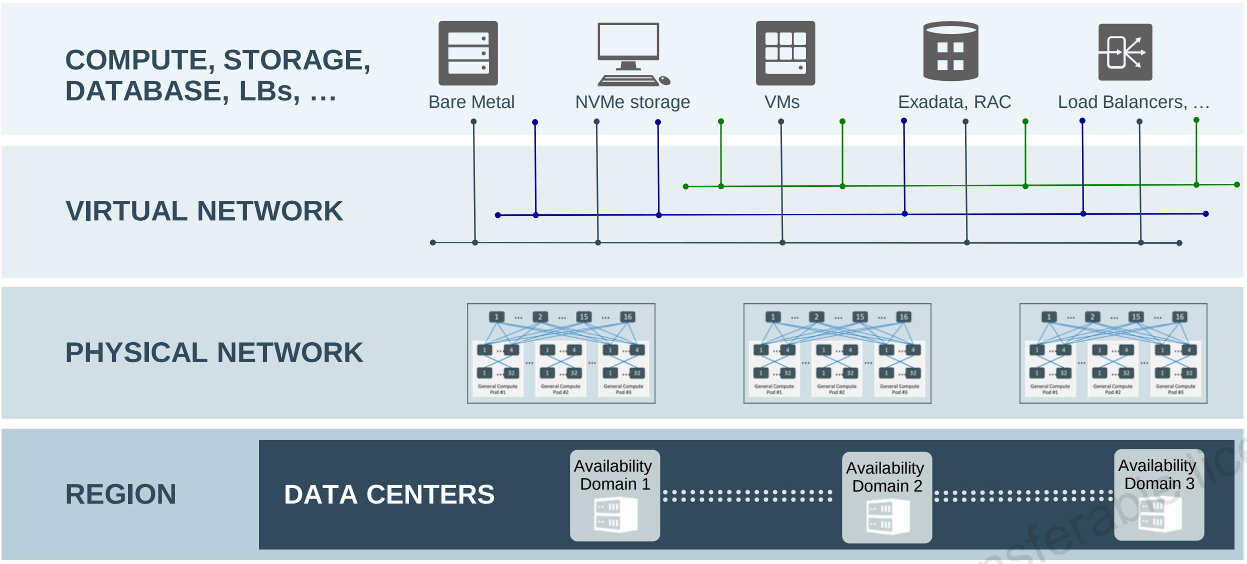
Comprehensive Virtual Network with Off-Box Virtualization

Highly configurable private overlay networks: Move management and I/O out of the hypervisor and enable lower overhead and bare metal instances.



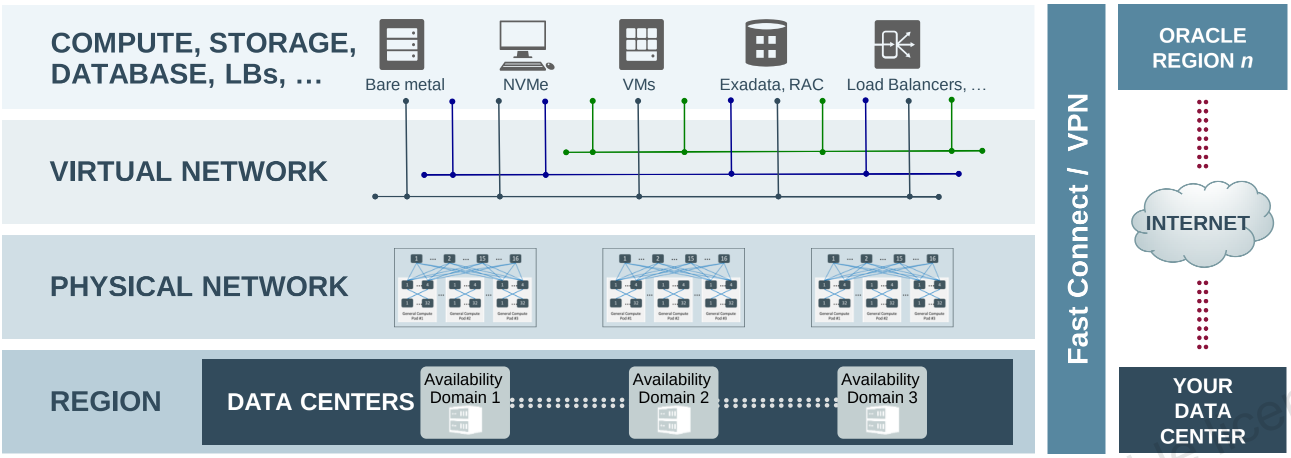
We have made some drastic changes to how virtual networking is done. We call it off-box virtualization. As the name implies, we pulled all the virtualization out into the network, including storage and network I/O virtualization. Generally, this enables the next layer up – so we can take any physical form-factor and plug that into our virtual network. This is the basis that lets us do bare metal and engineered systems like Exadata and plug it into this environment without making any changes. It is a massive enabler for us to deliver classes of services and meet our goals around performance and security.

Oracle Cloud Infrastructure: Innovation At Its Core

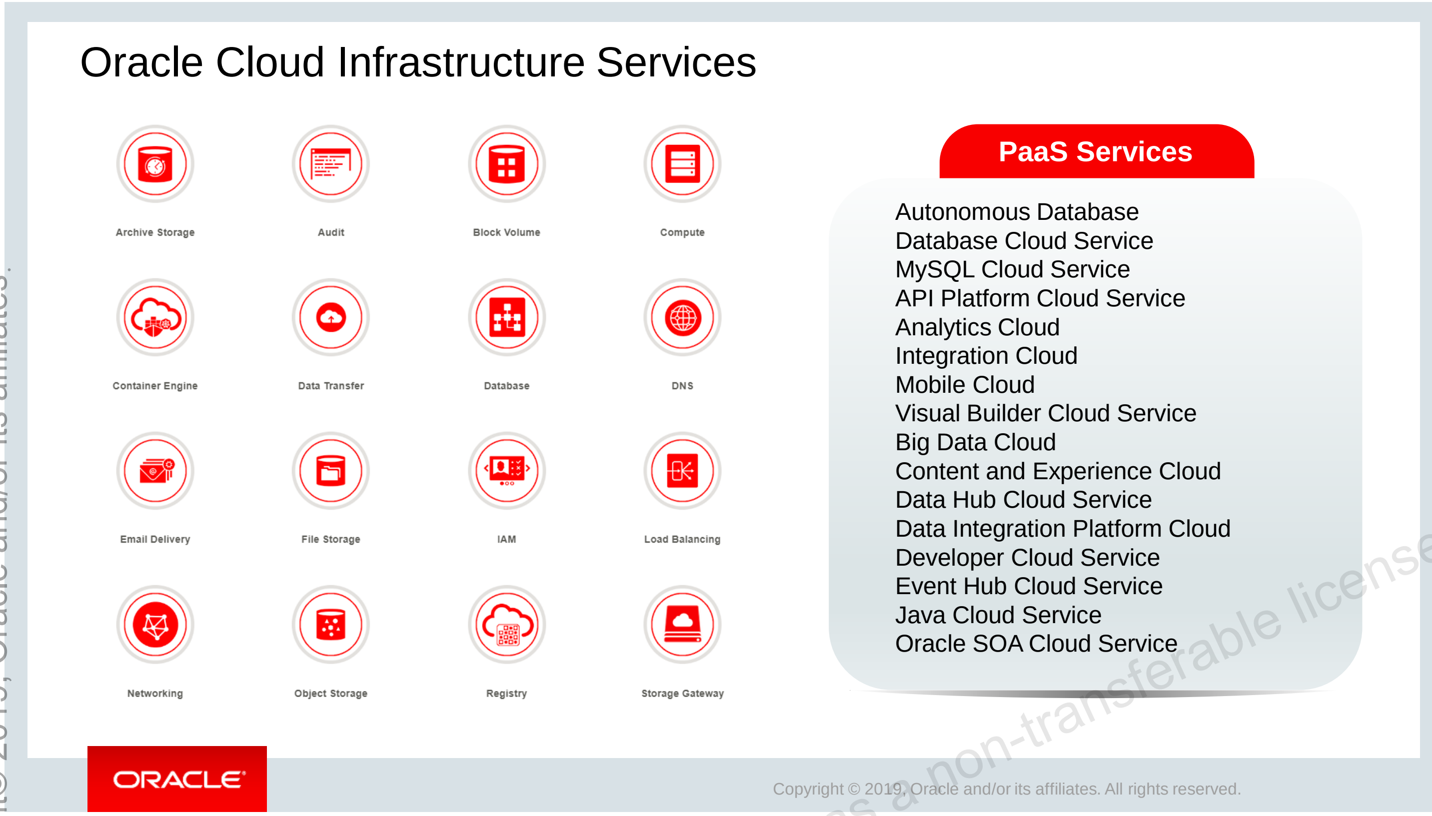


So this is what the fully integrated picture looks like. We have the largest class of compute and storage options plugged into a highly differentiated virtualized environment built using off-box network virtualization.

Oracle Cloud Infrastructure: Innovation At Its Core



And if we bundle that with the backbone network, this is what the picture looks like.



Now let's talk about the various services we have on our platform today. The IAM service helps you set up administrators, users, and groups, and specify their permissions. Audit service helps you track activity in your environment. Networking service helps you set up software-defined versions of traditional physical networks. Compute service helps you provision and manage compute instances.

Block Storage service helps you dynamically provision and manage block storage volumes. The Object Storage service helps you manage data as objects that are accessed over the Internet. Load Balancing service helps you create a load balancer within your virtual network. The Database service helps you provision and manage Oracle databases.

Oracle Cloud Infrastructure Core Themes

- **Lift and Shift:** Enable enterprise workload migration to the cloud without re-architecting
  - Unmatched Oracle on Oracle
  - Extend to non-Oracle workloads (VMware, SAP, custom apps)
- **Infrastructure Heavy Workloads:** Demand high scale/high performance
  - Best hardware, best performance, best price
  - Big Data, HPC, Machine Learning
- **Cloud Native Workloads:** Provide programmable infrastructure for cloud-first development
  - Self-service, cost, flexibility, agility
  - Modern apps and DevOps

There are three core themes of OCI: lift and shift, infrastructure heavy, and cloud native workloads. For lift and shift, OCI is the best place to run Oracle workloads and customers can bring in their Oracle apps like EBS and run them on OCI. We also want to make OCI the best place to run non-Oracle workloads, such as SAP, and so on. Our second core theme is infrastructure heavy workloads that require high scale and high performance. These are workloads like HPC and Big Data workloads. Our final core theme is around cloud native workloads, where customers can write modern apps using open source DevOps tools, such as Terraform, and run these apps on OCI.

Key Differentiators

Oracle apps and support for enterprise IaaS architecture:

- Best cloud to run Oracle Database and key enterprise Oracle apps
- Industry's first Bare Metal Cloud Service
- Flexibility and control (Bare Metal and VMs share the same set of APIs)
- Off-box network virtualization (with support for plugging Exadata appliances)
- Non-oversubscribed network, predictable performance with low latency and high throughput
- Robust security and governance capabilities

Industry-leading price performance:

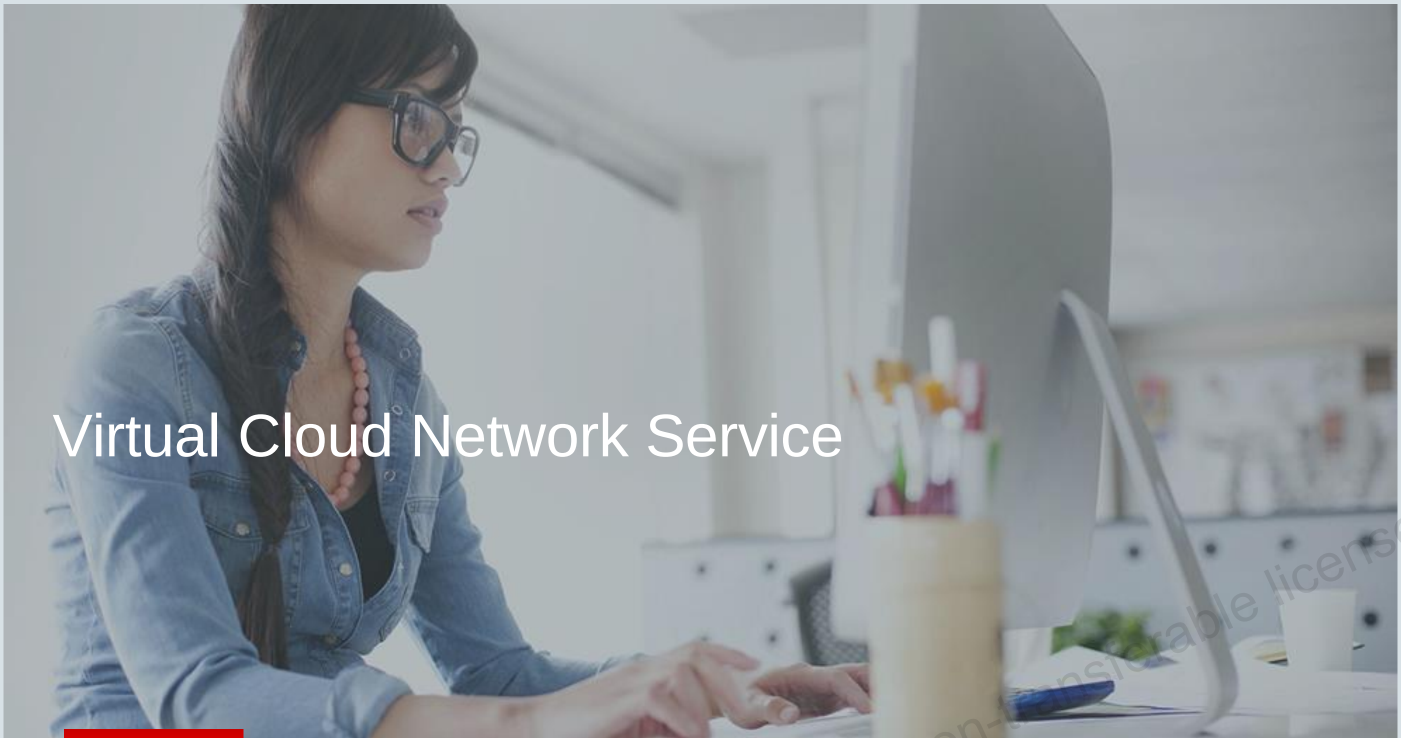
- Simple pricing; best performance
- Lower compute costs than AWS EC2 compute
- Fast, predictable block storage with no additional cost for IOPS; multiple X cheaper than AWS
- Bandwidth costs lower than AWS bandwidth by 85%

There are two main areas where we differentiate. First is the support for Oracle apps and enterprise IaaS architecture. OCI is the best platform for running Oracle Database and key enterprise Oracle apps. We are the only cloud that supports bare metal services and where VMs and bare metal servers have the same set of APIs. Also, our network is highly differentiated. We have a fundamentally different approach to networking through off-box network virtualization and our network is big, flat, and fast. As a result, you can get tremendous throughput and low latency. We also have a unique approach to security and governance through the use of compartments, which we'll talk about in later modules.

Second, we are the price performance leader. Our compute, storage, and network costs are all lower than AWS'.

Oracle Cloud Infrastructure Essentials

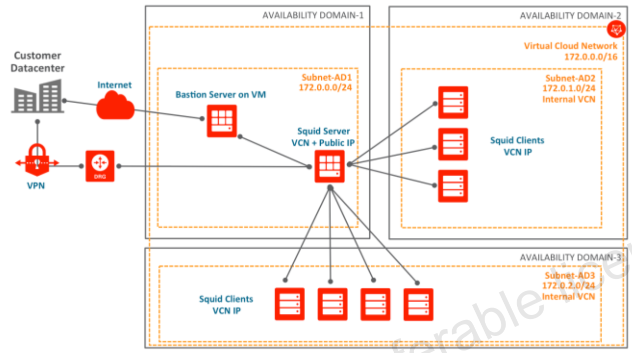
ORACLE



Virtual Cloud Network Service

Virtual Cloud Network (VCN)

- A Virtual Cloud Network is a software-defined version of a traditional physical network including subnets, route tables, and gateways on which your instances run.
- A VCN resides within a single region but can cross multiple Availability Domains.
- Internet Gateway provides a path for network traffic between your VCN and the Internet.
- It has a virtual router that provides a single point of entry for remote network paths coming into your VCN.
- You can use DRG to establish a connection with your on-premises network via IPSec VPN or FastConnect.



ORACLE

A VCN within Oracle Cloud Infrastructure is a software-defined version of a traditional physical network. VCN is a regional service and you can create a VCN by specifying a CIDR range. Each VCN network is subdivided into subnets, and each subnet is contained within a single Availability Domain.

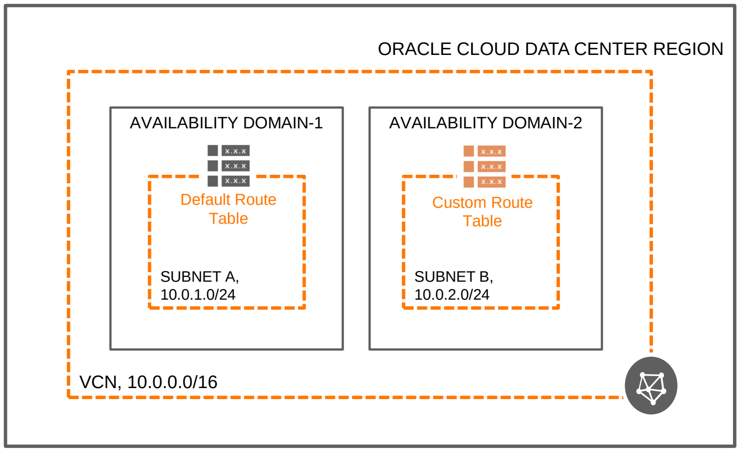
Internet Gateway provides a path for network traffic between your VCN and the Internet.

DNS enables lookup using host names. Remember this isn't a public DNS service, but DNS for VCN and subnets. The default choice is Internet and VCN Resolver, which lets the instances use host names to communicate. So the way it works is that you enable the Internet and VCN Resolver across your entire VCN. This means all instances in the VCN can communicate with each other without knowing their IP addresses.

With FastConnect, you can establish a connection in one of these ways:

- **Colocation:** By co-locating with Oracle in a FastConnect location
- **Provider:** By connecting to a FastConnect provider

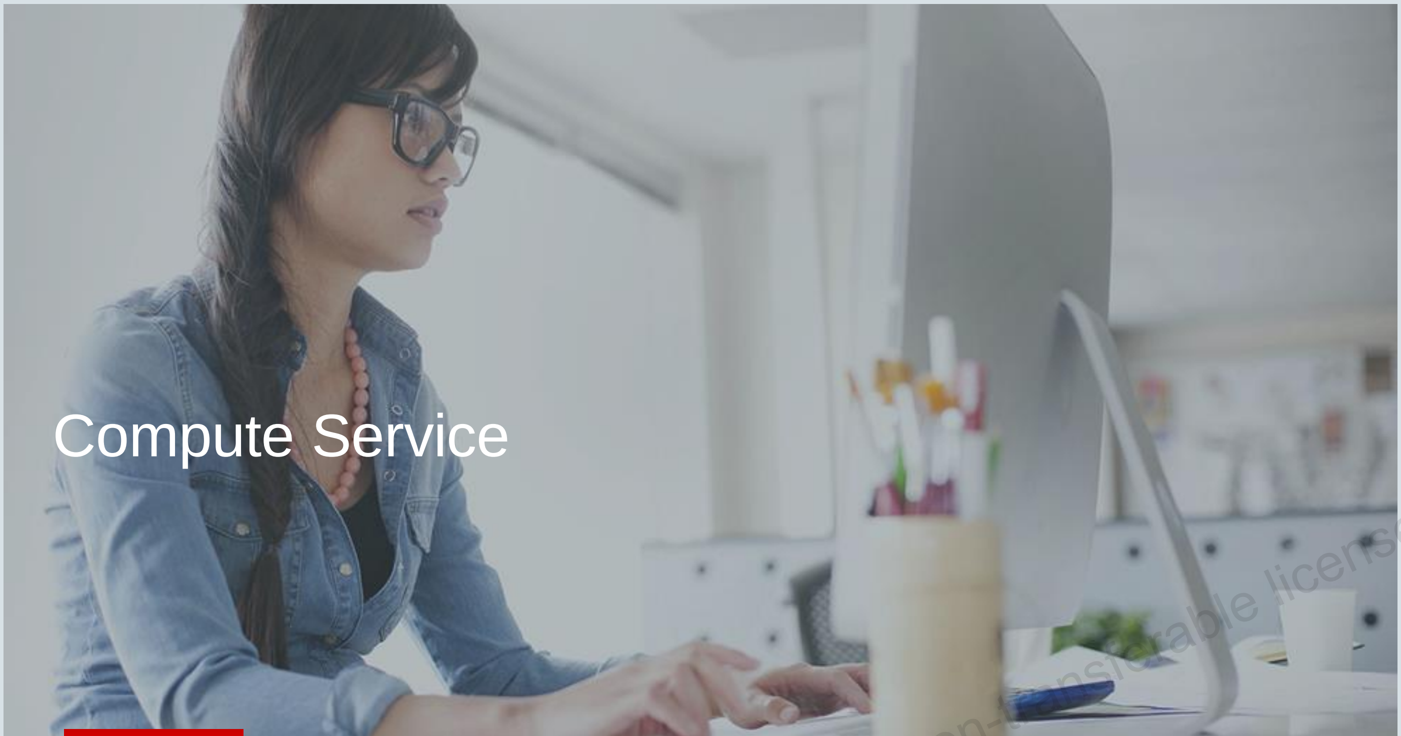
Default VCN Components



Your VCN automatically comes with some default components:

- Default Route Table
- Default Security List
- Default set of DHCP options

You can't delete these default components. However, you can change their contents (for example, individual route rules). Also, you can create more of each kind of component in your cloud network (for example, additional route tables).


Compute Service

Compute: Bare Metal and Virtual Machines

Bare Metal (BM)

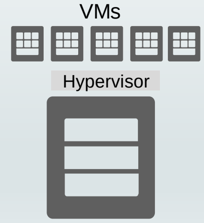
Direct Hardware Access: Customers get the full Bare Metal server

(Single-tenant model)

Virtual Machine (VM)

A hypervisor to virtualize the underlying Bare Metal server into smaller VMs

(Multi-tenant model)



VM compute instances run on the same hardware as Bare Metal instances, leveraging the same cloud-optimized hardware, firmware, software stack, and networking infrastructure.

OCI is the only public cloud that supports bare metal and VMs using the same set of APIs, hardware, firmware, software stack, and networking infrastructure. You can see the two models in the slide. Bare metal instances are instances where customers get the full server. This is also referred to as a single-tenant model. The advantage here is that there is no performance overhead, no shared agents, and no noisy neighbors. On the other spectrum are VMs, where the underlying host is virtualized to provide smaller VMs, also referred to as the multi-tenant model. The advantage here is flexibility with regard to choice of instance shapes.

Shape: Processor and Memory Resources

- Oracle Compute Cloud Service enables you to select from a range of predefined shapes that determine the number of CPUs, amount of RAM, and local storage available in an instance.
- Several predefined shapes are available for both bare metal and virtual machine instances.

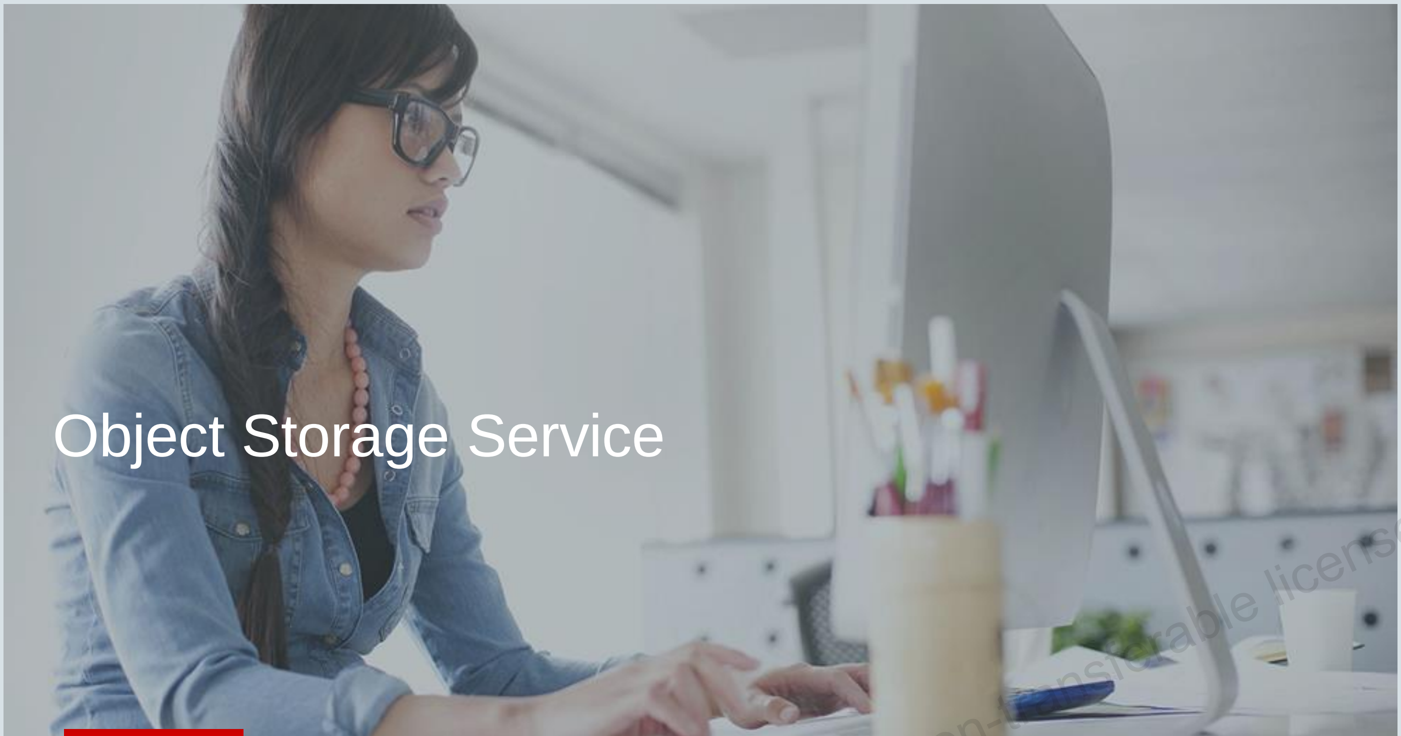
ORACLE

When creating Compute instances, you can assign CPU and memory resources by selecting from a wide range of resource profiles (called shapes), each of which is a carefully designed combination of processor and memory limits.

For information on Compute shapes, refer to:

https://docs.cloud.oracle.com/iaas/Content/Compute/References/computeshapes.htm

Object Storage Service



Object Storage Service

- An Internet-scale, high-performance storage platform
- Ideal for storing an unlimited amount of unstructured data (images, media files, logs, backups)
- Data is managed as objects using an API built on standard HTTP verbs
- Safely and securely store or retrieve data
- A regional service which is not tied to any specific compute instance

ORACLE

The Oracle Cloud Infrastructure Object Storage service is an internet-scale, high-performance durable storage platform that offers reliable and cost-efficient data durability.

The Object Storage service can store an unlimited amount of unstructured data of any content type, including analytic data and rich content, like images and videos.

Data is managed as objects and not as files or blocks. It is not managed via any storage protocol but rather uses standard HTTP methods, such as HTTP GET and HTTP PUT.

Moreover, with Object Storage, you can safely and securely store or retrieve data directly from the Internet or you can do it from within the cloud platform using multiple management interfaces that let you easily manage storage at scale.

One last thing to mention about Object Storage is that it is completely independent of the compute server and is a regional service. So as long as you have the Object Storage endpoints and relative authorization, you can access the data from anywhere.

Common Object Storage Scenarios

- Content Repository
  - Highly available and durable content repository for data, images, logs, video, etc.
- Archive/Backup
  - Use of object storage for preserving data for longer periods of time
- Log Data
  - Application log data for analysis and debugs/troubleshooting
- Large Data Sets
  - Large data, such as pharmaceutical trials data, genome data, and Internet of Things (IoT)
- Big Data/Hadoop Support
  - Use as a primary data repository for big data giving ~50% improvement
  - **HDFS Connector** provides connectivity to various big data analytic engines like Apache Spark and MapReduce

ORACLE

HDFS Connector:

- HDFS: A distributed file system used in Hadoop. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.
- HDFS Connector lets your Apache Hadoop application read and write data to and from the Oracle Cloud Infrastructure Object Storage service.

  https://docs.us-phoenix-1.oraclecloud.com/Content/API/SDKDocs/hdfsconnector.htm

Let's talk about common use cases for Object Storage.

The Object Storage service offers a scalable storage platform which enables you to not only store large data sets but also operate seamlessly, making it an ideal storage solution for Big Data applications. We also have a Hadoop Distributed File System (HDFS) Connector which lets your Apache Hadoop application read and write data to and from the Object Storage service.

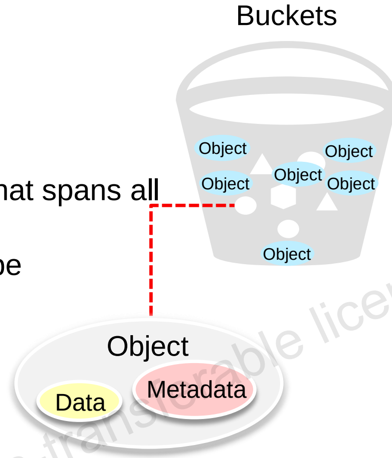
Another common use case is back up or archive, where data is typically written once and read many times. Or the data is archived for longer periods of time. The durability and low-cost characteristics of Object Storage tiers make it a perfect platform to store long living data.

Object Storage can also be used as a content repository for storing different kinds of data and scaling it seamlessly as the data grows.

Additional use cases include log data analysis and storing large data sets.

Object Storage Resources

- Object
  - All data, regardless of content type, is managed as objects (e.g. logs, videos).
  - Each object is composed of the object itself and the metadata of the object.
- Bucket
  - It is a logical container for storing objects.
  - Each object is stored in a bucket.
- Namespace
  - Each tenant is associated with one default namespace that spans all compartments.
  - Bucket names within a namespace are unique, but can be repeated across namespaces.
  - Buckets and objects exist in a flat hierarchy.
- Compartment
  - Buckets can only exist in one compartment.



ORACLE

Within the Object Storage service, there are some key resources.

As we discussed earlier, all individual data elements are stored as objects, regardless of the content type. Each object is stored in a bucket and is a combination of the object itself and its metadata. The metadata is a list of key value pairs of data, e.g. name, size, content type, last modified date, etc.

A bucket is a logical container created by the user, and can contain an unlimited number of objects. A bucket is associated with a single compartment which, in turn, defines policies that indicate what actions a user can perform on a bucket.

A namespace is a logical entity that gets created with a tenant, and spans all compartments in that tenant. Within a single namespace, the bucket names are unique. Buckets and objects in a namespace are in a flat hierarchy, but you can create directory structures for your ease.

Object Storage Service Features

- Strong consistency
  - The Object Storage service always serves the most recent copy of the data when retrieved.
- Durability
  - Data is stored redundantly across multiple storage servers across multiple ADs.
  - Data integrity is actively monitored, and corrupt data is detected and auto repaired.
- Performance
  - Compute and Object Storage services are co-located on the same fast network.
- Custom metadata
  - Define your own extensive metadata as key-value pairs.
- Encryption
  - A 256-bit Advanced Encryption Standard (AES-256) is employed to encrypt object data.

So what are the top features of Object Storage?

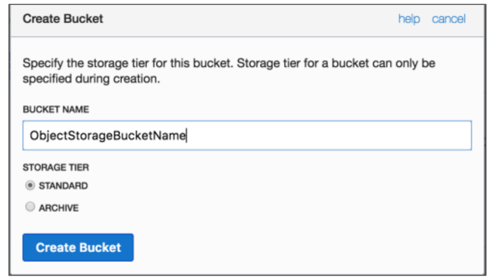
Object Storage has a strong consistency mode, which means when there is a read request made, you are guaranteed a consistent, most recent copy of the data that has been completely written in the system. We have a high performance network where lookups for data are fast and consistent.

To provide durability, the Object Storage service copies object data throughout a region across multiple availability domains. Data integrity is maintained with active scrubbing functions. Typically, there are 3-6 copies of a given object in the service.

The data on the object store is always encrypted. Each object has its own key and object keys are encrypted with a master key, which is frequently rotated so the encryption scheme is robust.

Object Storage Tiers

- Standard Storage Tier (Hot)
  - Fast, immediate, and frequent access
  - The Object Storage service always serves the most recent copy of the data when retrieved.
  - Data retrieval is instantaneous.
  - Standard buckets can't be downgraded to Archive Storage.
- Archive Storage Tier (Cold)
  - Seldom or rarely accessed data but must be retained and preserved for long periods of time
  - The minimum retention requirement for Archive Storage is 90 days.
  - Objects need to be restored before download.
  - An Archive Bucket can't be upgraded to the Standard Storage tier.
  - Time To First Byte after an Archive Storage restore request is made is 4 hours.



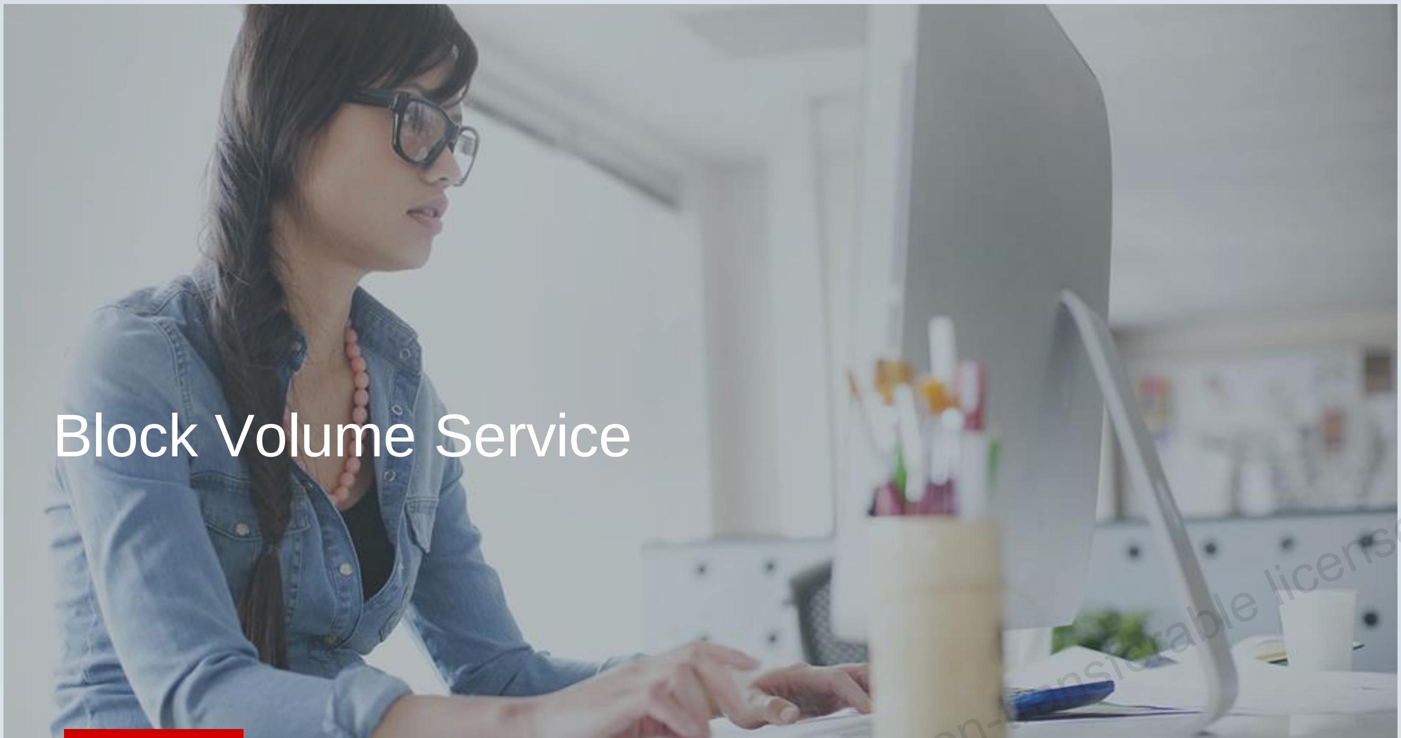
Currently, Object Storage provides two different storage tier options when creating a bucket.

A Standard Storage tier is for frequently accessed data. Any data stored in this tier is retrieved immediately and instantly. Data accessibility and performance justifies a higher price point to store data in the Standard tier. After a Standard Storage bucket is created, it can't be downgraded to an Archive Storage tier.

The other option is the Archive Storage tier. This is for data that is not frequently accessed, but must be preserved for longer periods of time.

While storing data in Archive Storage, a minimum 90-day retention period is required. Removal of the data before that results in a penalty fee.

After the data is uploaded in Archive Storage, it first needs to be restored before it can be accessed. The time to first byte is four hours, and the total time for full restoration depends on the size of the data stored.



Block Volume Service

Block Volume Service

- The Block Volume service lets you store data on block volumes independently and beyond the lifespan of compute instances.
- Block volumes operate at the raw storage device level and manage data as a set of numbered, fixed-size blocks using a protocol such as iSCSI.
- You can create, attach, connect, and move volumes, as needed, to meet your storage and application requirements.
- Typical Scenarios
  - Persistent and Durable Storage
  - Expand an Instance's Storage
  - Instance Scaling

ORACLE

To understand the service, let's begin by understanding what block volume is. Block volume is a type of data storage that is more expansive than file storage. It uses the iSCSI Ethernet protocol to deliver features and performance similar to on-premises storage area networks or SANS, and are designed for the security and durability of the data life cycle. Using this service, you can create block volumes and attach them to your compute instance.

The Oracle Cloud Infrastructure Block Volume service delivers a simple, scalable block volume service that fulfills all your workload performance needs. The service lets you dynamically provision and manage block storage volumes. You can create, attach, create backups, and move volumes, as needed, to meet your storage and application requirements.

After it is attached and connected to an instance, you can use the volume like a regular hard drive. The service also lets you store data on blocked volumes, manage block volumes, control your data, and achieve the storage configuration your application requires.

Block Volume service utilizes industry-leading highest performance NMVe drives and is offered over the network using standard iSCSI protocol.

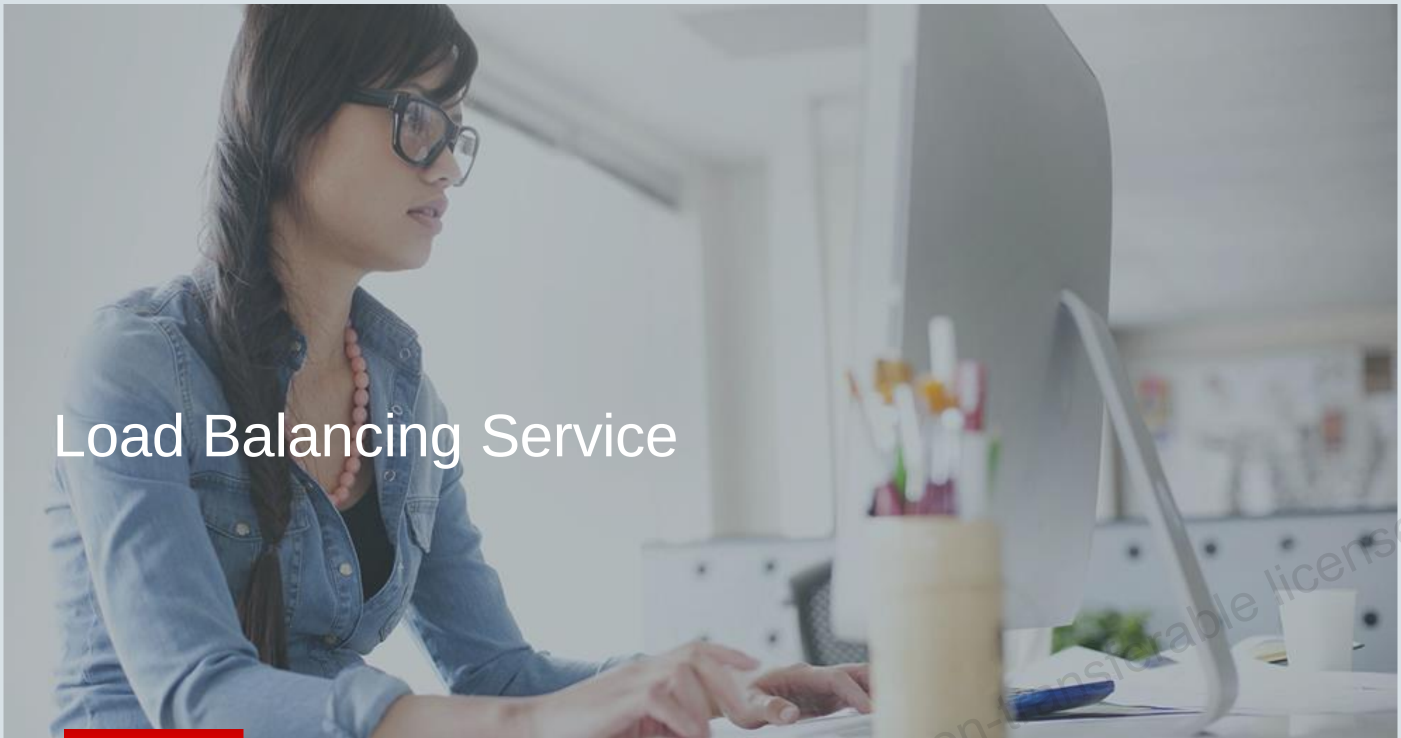
Some of the typical scenarios for Block Volume service include:

- **Expanding an instance storage:** Typically, block volumes are used to increase an instance storage. After any compute instance is launched, you can attach block volumes to the instance, increasing the instance's storage capacity.
- **Persistent and durable storage:** Block volumes can be detached from one instance and reattached to another without any loss of data. This data persistence allows the user to migrate data between the instances safely, and also safely store it even when it is not attached to an instance.

Additionally, Block Volume volumes offer a high level of data durability compared to locally attached drives. All volumes are automatically replicated for you, helping to protect against data loss.

- **Instance Scaling:** With new features like boot volume, the service allows you to scale instance CPUs, keeping the data secure in the process. We will talk about this later in the lesson.

One important thing to remember is that block volume is always associated with an instance. Therefore, it is always created within an availability domain. That is why it is an AD construct.



Load Balancing Service

OCI Load Balancing Service

- Provides automated traffic distribution from one entry point to multiple servers in a VCN
- Improves resource utilization, facilitates scaling, and helps ensure high availability
- Supports Public and Private Load Balancers
- The Public Load Balancer service is regional in scope and requires 2 ADs.
- Protocols supported: TCP, HTTP/1.0, HTTP/1.1, HTTP/2, WebSocket
- Supports SSL Termination, End-to-End SSL, and SSL Tunneling
- Supports Session Persistence and content-based routing
- Key differentiators
  - Private or Public Load Balancer (with Public IP address)
  - Provisioned Bandwidth: 100 Mbps, 400 Mbps, 8 Gbps
  - Single LB for TCP (layer 4) and HTTP (layer 7) traffic

ORACLE

So Oracle Cloud Infrastructure Load Balancing Service provides an automated traffic distribution from one entry point into multiple back-end servers in your Virtual Cloud Network.

This helps to load balance large amounts of traffic, which could overwhelm a single server. It gives a mechanism to scale out an application tier by adding more servers, and also provides the application higher availability so even if one availability domain has an issue, you can still be up and running in other availability domains.

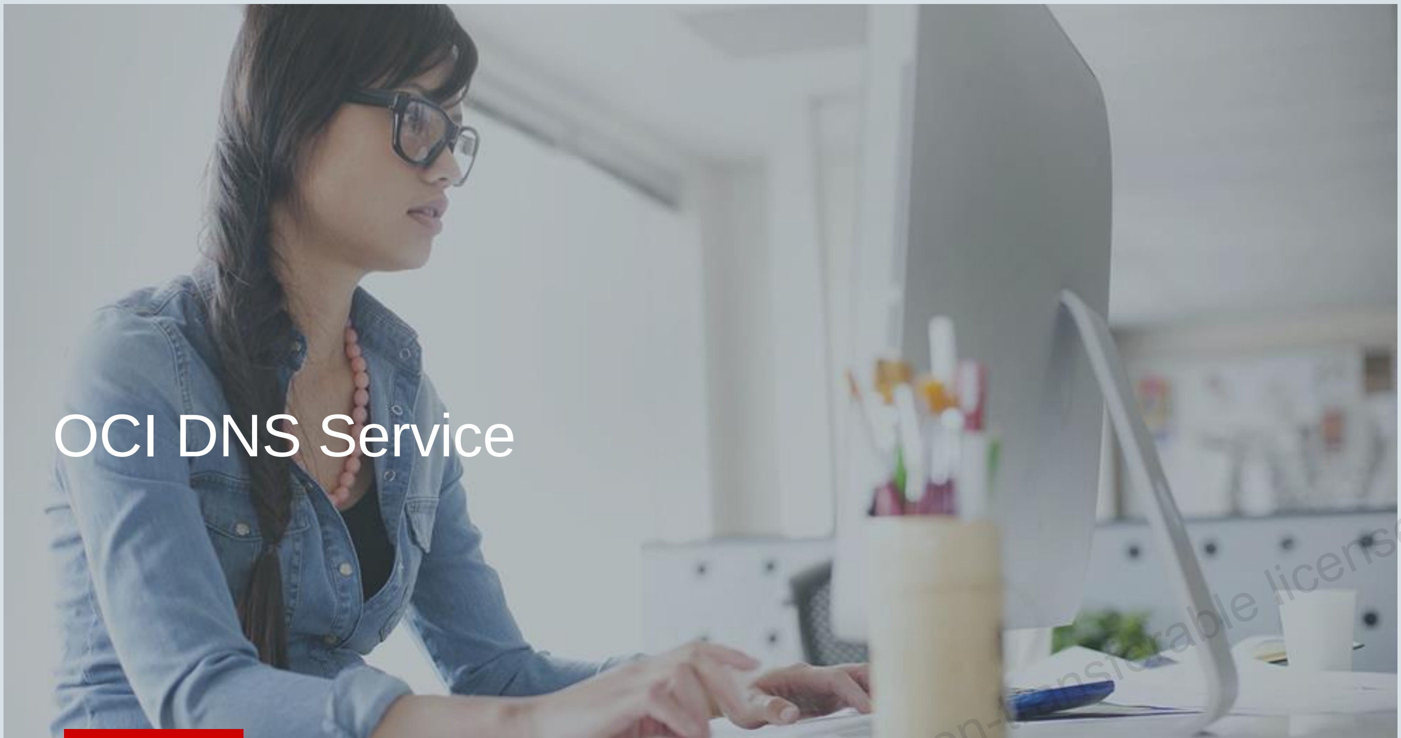
Load Balancer is a regional service. Load balancers come in pairs (active and passive), and public load balancers live in two separate availability domains providing HA, with no single point of failure.

The OCI Load Balancer supports TCP and the usual http protocols, as well as HTTP/2 and websocket, supporting things like Data Compression, Server Push, Multiplexing of Requests.

For security purposes, it supports SSL offloading, SSL termination, SSL end to end, and SSL tunneling.

Let's talk about the key differentiators for the LB service.

1. We can deploy the service either as public facing, where a listener is running on the public IP and the backend servers are on the inside, using the same service to load balance within OCI between tiers, keeping it entirely private.
2. The other nice feature of the OCI Load Balancer service is you get a public or a dedicated IP address. You don't have to worry about getting a CNAME and dealing with that to use this service. The listener listens on the service port on this IP address and it is mapped to the user's OCI tenancy.
3. The load balancers come in three sizes: 100Mbps, 400Mbits, and 8Gbits. These sizes are for aggregate throughput. The nice thing about having this much capacity provisioned is that it is always available to the user. There is no warm-up period when using these shapes. This aggregate throughput performance is always available.
4. There is a single load balancer for HTTP and TCP. This makes the service easier to use in general.



OCI DNS Service

ORACLE

Oracle Cloud Infrastructure: DNS

- Highly scalable, global anycast Domain Name System (DNS) network that assures high site availability and low latency
- Customers can manage DNS records. Domain names can be either cloud or non-cloud resources.
- Oracle OCI DNS service is used when:
  - Domains and Zones need to be exposed via the Internet for DNS resolution
  - Domains and Zones can reside in both Enterprise on-premises and OCI environments
  - DNS traffic needs to be intelligently handled across multiple resources

ORACLE

The Dyn network (Managed DNS, Email Delivery, Internet Guide) is deployed on top of a global IP network, consisting of 20 existing facilities and connectivity from a mix of Tier 1 Internet Service Providers (ISPs).

The network has been split into two diverse constellations to provide active/active failover between constellations in the event of catastrophic failure.

Within each constellation, we distribute traffic to multiple data centers, providing global active-active load balancing using the Anycast routing technique. Queries via the Anycast technique allow the fastest of the entire network to answer the query.

This network allows OCI DNS/Dyn DNS to offer its customers an industry-leading level of service and reliability.

The network continues to grow and add more points of presence.

This results in a superior end user experience connecting to OCI.

By configuring OCI DNS, enterprise and business customers can connect their DNS queries to various kinds of assets, such as OCI Compute as well as to third party and private assets.

Operators can manage their own DNS records for both cloud and non-cloud resources.

OCI DNS is used when zones need to be exposed to the Internet for resolution.

Domains and zones can be both on the OCI environment as well as in the enterprise.

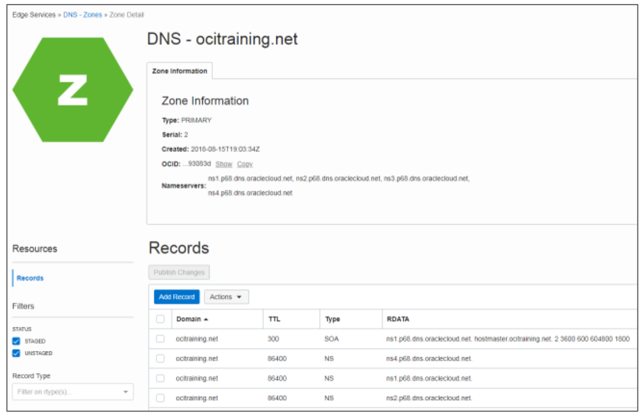
DNS needs to be handled across multiple resources.

DYN can also act either as a primary or secondary DNS and follow DNS specifications carefully and by the appropriate RFC whenever possible.

Capabilities of OCI DNS

The following functions are available:

- Create and manage zones.
- Create and manage records.
- Import/upload zone files.
- Transfer zones.
- Save and publish changes.
- View all zones.
- Query counts (total and per zone).



- The OCI DNS solution offers a complete set of functions for zone management within the user interface.
- It is possible to create zones within the tenancy. Zones are tenancy-wide and, along with IAM OCI DNS, crosses all regions.
- Users can manage records from the consoles.
- It is also possible to import complete zones via the OCI DNS console.
- It is possible to set up OCI DNS as a secondary DNS server and facilitate zone transfers from the primary DNS server.
- OCI DNS supports zone transfers via AXFR(full) or IXFR(incremental) zone transfer.
- OCI DNS keeps tracks of all queries against the service, both at the zone level and total.

Summary

In this lesson, you should have learned how to:

- Describe the Oracle Cloud Infrastructure strategy
- Define key concepts and terminology
- Identify Oracle Cloud Infrastructure Services
- Get an overview of key Oracle Cloud Infrastructure Services:
  - Virtual Cloud Network (VCN) Service
  - Compute Service
  - Object Storage Service
  - Block Volume Service
  - Load Balancing Service
  - OCI DNS Services

Practice 3: Overview

There are no practices for this lesson.

4

Oracle Autonomous Database Overview

ORACLE

Objectives

- Oracle Autonomous Database
- Key Features of Autonomous Database
- Typical Workflow of Autonomous Database
- Using Oracle Machine Learning with Autonomous Database
- The Oracle Cloud Infrastructure Console

ORACLE

Oracle Autonomous Database Overview

ORACLE AUTONOMOUS DATABASE

ORACLE

Oracle Autonomous Database

Next Generation Mission-Critical Data Management in the Cloud

Oracle Autonomous Database (Autonomous Database) combines the flexibility of cloud with the power of machine learning to deliver data management as a service.

Autonomous Database provides a level of performance and reliability that manually managed databases can't deliver.

- Self-Driving
  - The user defines workloads and policies, Autonomous Database makes them happen
- Self-Securing
  - Protection from both external attacks and internal users
- Self-Repairing
  - Automated protection from all down time

ORACLE

Like an autonomous car, Oracle Autonomous Database provides a level of performance and reliability manually managed databases can't deliver.

Compared to a manually managed database, Autonomous Database costs less to run, performs better, is more available, and eliminates human error.

Self-Driving

Autonomous Database eliminates human labor to provision, secure, monitor, backup, recover, troubleshoot, and tune databases. This greatly reduces database maintenance tasks, reducing costs and freeing scarce administrator resources to work on higher value tasks.

In addition, when it comes time to upgrade or patch, Autonomous Database can replay the real production workload on a test database to make sure the upgrade does not have any unexpected side effects on a mission-critical system.

Autonomous Database automatically tunes itself using Machine Learning algorithms, including automatically creating any indexes needed to accelerate applications. Users get the ultimate simplicity of a "load and go" architecture in which they can simply load their data and run SQL without worrying about creating and tuning their database access structures.

Self-Securing

Autonomous Database is more secure than a manually operated database because it protects itself rather than having to wait for an available administrator. This applies to defenses against both external and internal attacks.

Security patches are automatically applied every quarter. This is much sooner than most manually operated Oracle databases, narrowing an unnecessary window of vulnerability.

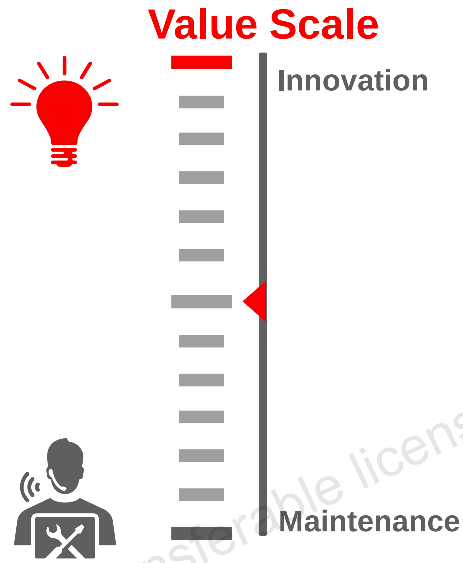
All data is encrypted using transparent data encryption. Network connections from clients to ADB are also encrypted using the client credentials wallet. Additional in-database features like Virtual Private Database and Data Redaction are also available. Customers are not given OS logons or SYSDBA privileges to prevent phishing attacks.

Self-Repairing

Autonomous Database is more reliable than a manually operated database. At startup, it automatically establishes a triple-mirrored scale-out configuration in one regional cloud data center, with an optional full standby copy in another region. Autonomous Database automatically recovers from any physical failures, whether at the server or data center level.

Traditional DBA Responsibilities

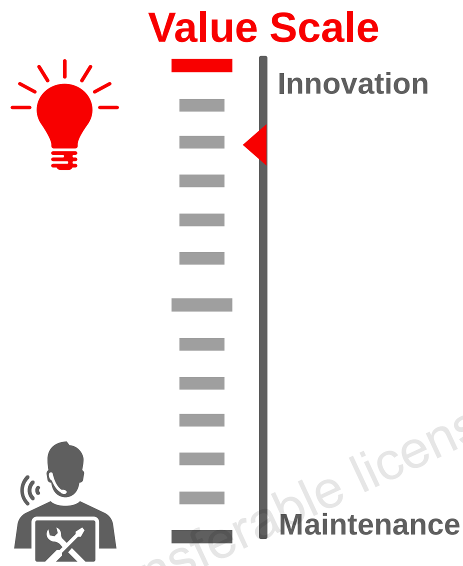
- **Tasks Specific to Business and Innovation**
  - Architecture, planning, data modeling
  - Data security and lifecycle management
  - Application-related tuning
  - End-to-end service-level management
- **Maintenance Tasks**
  - Configuration and tuning of systems, network, storage
  - Database provisioning,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



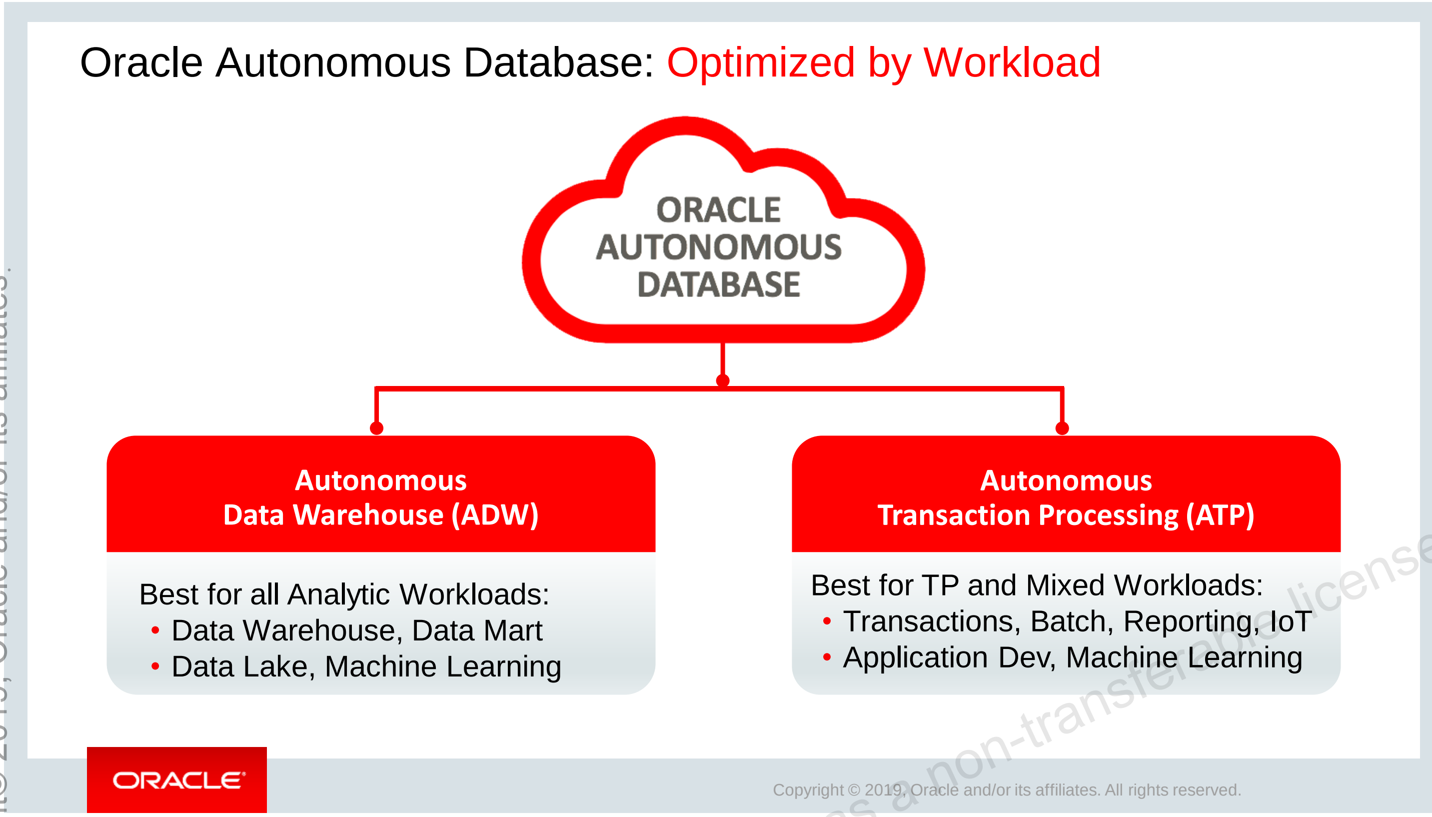
Autonomous Database Removes Generic Tasks

- **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



Optimized for Different Workloads

Oracle Autonomous Database is actually a family of products, with each member of the family optimized by workload. Because Autonomous Database is based on the extremely feature rich and proven Oracle Database, on the Exadata platform, it is able to run both OLTP and analytic workloads up to 100X faster.

- **Autonomous Data Warehouse (ADW):** The simplest and most efficient database optimized for analytic workloads, such as data warehouse, data marts, or as part of a data lake.
- **Autonomous Transaction Processing (ATP):** Optimized for transaction processing or mixed workload environments. Makes an excellent platform to run enterprise applications, including mixed workloads and real-time analytics, with no compromise on app performance.

Autonomous Optimizations: Specialized by Workload

		ADW	ATP
	Primary Goal	Fast Complex Analytics	Fast Transactions Processing
	Data Formats	Columnar	Row
	Data Access Acceleration	Creates Data Summaries	RDMA for Messaging and IO
	Memory Usage	Parallel Joins and Aggregations	Data Caching to Avoid IO
	Statistics	Automatically manages optimizer statistics as data changes	

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

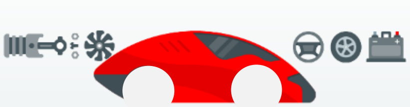
The difference is how the services have been optimized within the database. When you start loading data into the autonomous database, we store the data in the appropriate format for the workload.

- If it is ADW, then we store data in columnar format because that's the best format for analytics processing.
- If it is ATP, then we store the data in row format because 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 business 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 data volumes change, or new access structures are created, there is the potential for an execution plan to change, and any change could result in a performance regression. Therefore, we use Oracle SQL Plan Management to ensure that plans only change for the better.

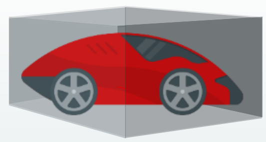
Full Database Lifecycle Automation: How It Works



Provision

Rapidly and easily creates mission critical databases

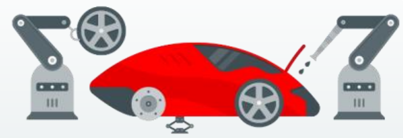
Creates **Exadata**+ Cloud Infrastructure and **Real Application Clusters**+, scales out the database



Secure

Protects data from all external and internal threats

Continuously detects threats, applies security **updates online**+, prevents admin snooping, **encrypts** all data



Manage

Automates all infrastructure and database maintenance

Patches all software **online**+, tunes settings, performs **all OS and SYSDBA** operations

+ Unique to Oracle

ORACLE

These are the six key elements in the life cycle of a database:

Provision

Oracle quickly deploys a mission critical database on our Exadata infrastructure cloud, a platform optimized for Oracle Database. The database is created as a RAC cluster, so it can scale out seamlessly, but also protect the database in case of a server failure in order to protect the system against complete failures.

Secure

After it is provisioned, the system needs to be secured. After all, being on the public cloud has the potential to expose you to more security risks. By default, we encrypt all the data in the database, but encrypting the data at rest is not enough. All connections to the database are fully encrypted too.

To ensure systems are not exposed to any security vulnerabilities, we also apply all security patches as soon as they are available.

We protect your system from both cloud operations and any malicious internal users by using Oracle Database Vault.

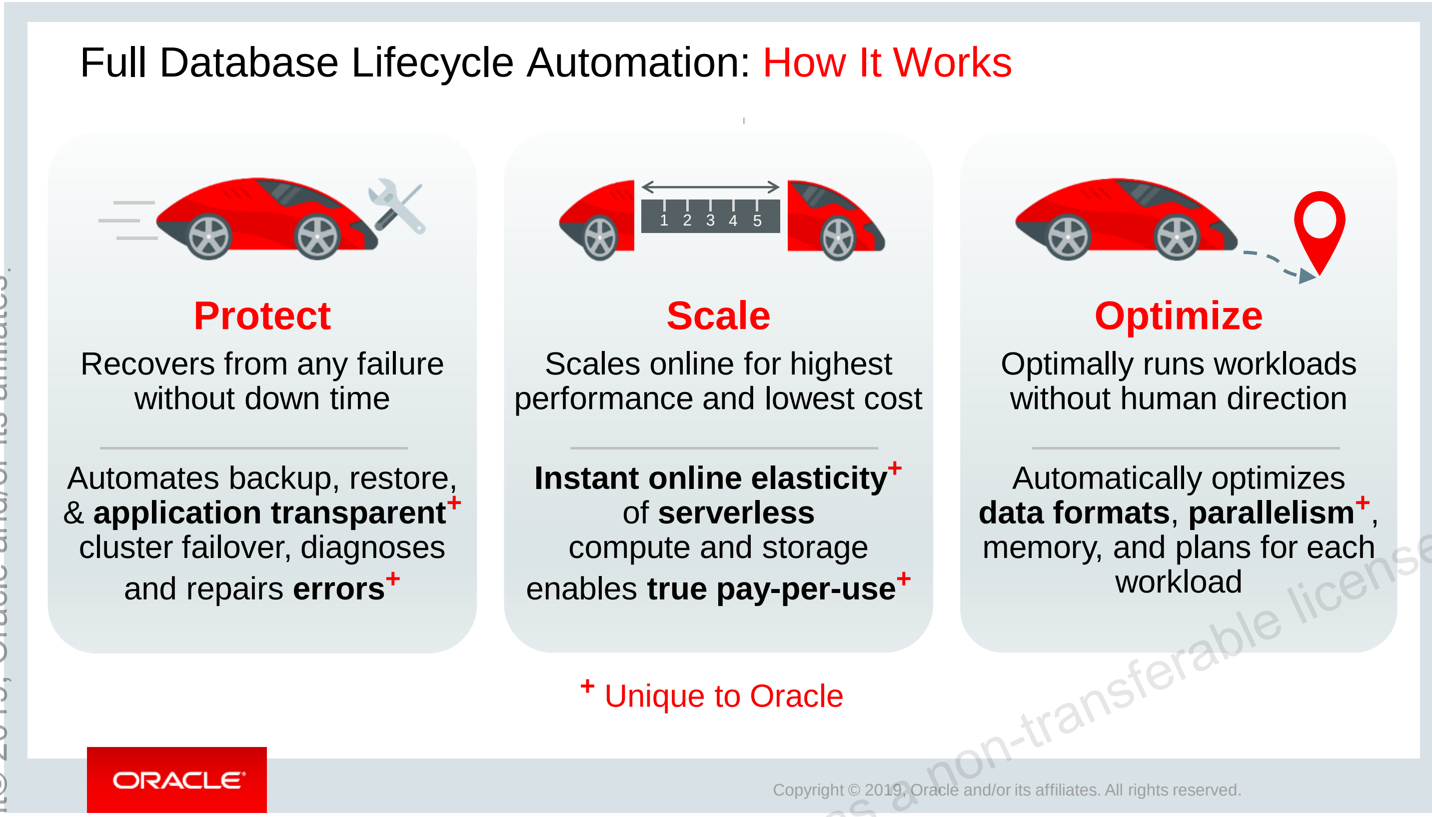
Embedded within the Oracle Database, we set up database vault, so none of the Oracle Cloud administrators can see your application data.

You can also use this feature to prevent your administrators from seeing that application data.

Manage

To increase security, we do not give customers access to the operating system or `SYSDBA` privileges. By revoking these privileges, your credentials cannot be stolen. However, this means that we have to do all the operations that need to be done by `SYSDBA` and at the operating system levels.

We will apply all the OS, storage, OS, VM, and database patches.



Protect

Protect your database against any down time.

The application continues to run without any additional code in the application.

We will back up your database nightly, and give you the ability to restore the database from any of the backups in your archive.

However, we will take it one step further by giving you a flashback database option.

For example, you have travelled to Japan and are jetlagged, and drop a table into production instead of test.

The flashback database will allow us to flash the database back to the original state without needing to go to a backup archive. This can be done online without any down time

If we need to fail over, we can do it transparently to the application.

No additional code is needed to get failover going in the application.

Also, we diagnose all problems.

We use machine learning to help detect and diagnose.

We continually monitor the cluster and look for any unusual changes in behavior.

If an anomaly does occur, we are able to compare that behavior to a known set of issues.

If a match is found, we automatically apply the known fix for the problem.

If no match is found, we automatically create an SR and escalate as appropriate.

Scale

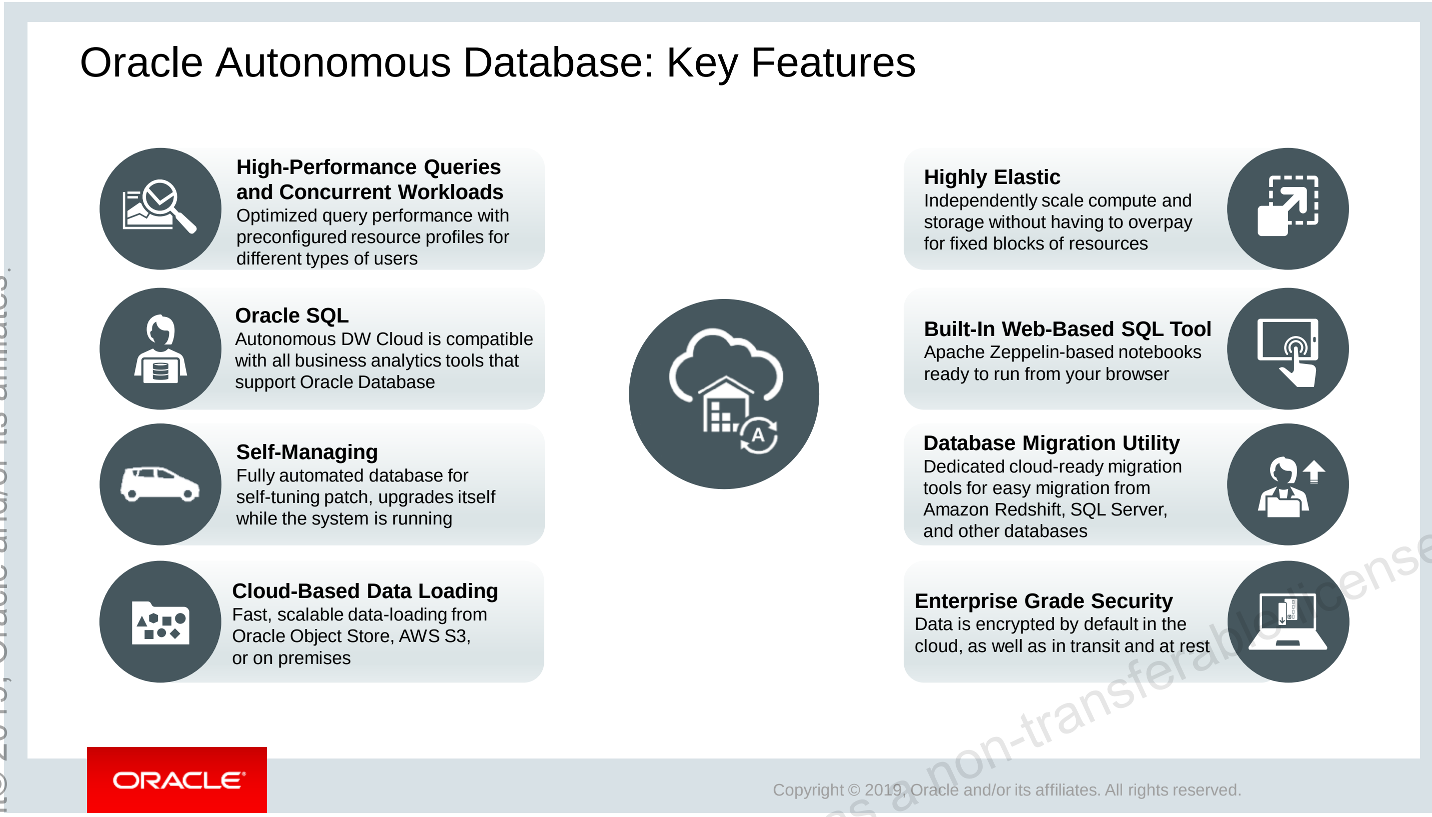
Many customers want to move to the cloud because of the elasticity a cloud environment can offer. Being able to scale both in terms of compute and storage only when needed allows people to pay per use.

Oracle allows you to not only scale compute and storage but also do it independently. There is no need to do it together like other cloud vendors require you do it. Oracle allows these scaling activities fully online (no down time required), while the application continues to run, and customers are really only paying for what they want to use.

Optimize

Oracle Autonomous Database is actually a family of products, with each member of the family optimized by workload. The first member of the family, Autonomous Data Warehouse (ADW), is optimized for analytic workloads, such as data warehouse, data marts, or as part of a data lake.

The second member of the family, Autonomous Transaction Processing (ATP), is optimized for transaction processing or mixed workload environments and makes an excellent platform for new application development.



Key Features

- **Managed:** Oracle simplifies end-to-end management of the data warehouse
  - Provisioning new databases
  - Growing or shrinking storage and compute resources
  - Patching and upgrades
  - Backup and recovery
- **Fully Tuned:** “Load and go”
  - Defines tables, load data, run queries
  - Provides good performance out-of-the-box
  - Runs your queries using any business analytics tool or cloud service
  - Includes a built-in SQL worksheet and notebook
- **Highly Elastic Scaling:** Scale compute and storage independently to fit your data warehouse workload with no down time
  - Size Autonomous Database to the exact compute and storage required
  - Scale Autonomous Database on demand: Independently scale compute or storage
  - Shut off idle compute to save money

Autonomous Database Supports:

- Existing applications, running in the cloud or on premises
- Connectivity via SQL*Net, JDBC, ODBC
- Third-party data-integration tools
- **Oracle cloud services:** Analytics Cloud Service, Golden Gate Cloud Service, Integration Cloud Service, and others
- **High-performance queries and concurrent workloads:** Optimized query performance with preconfigured resource profiles for different types of users
- **Oracle SQL:** Autonomous Database is compatible with existing applications that support Oracle Database.
- **Built-in web-based data analysis tool:** Web-based notebook tool for designing and sharing SQL based data-driven, interactive documents
- **Database migration utility:** Easily migrate from Amazon AWS Redshift, SQL Server, and other databases.
- **Simple cloud-based Data Loading:** Autonomous Database provides fast, scalable data-loading from Oracle Object Store, Azure Blob Storage, AWS S3, or on-premises data sources.

SQL Developer Autonomous Database

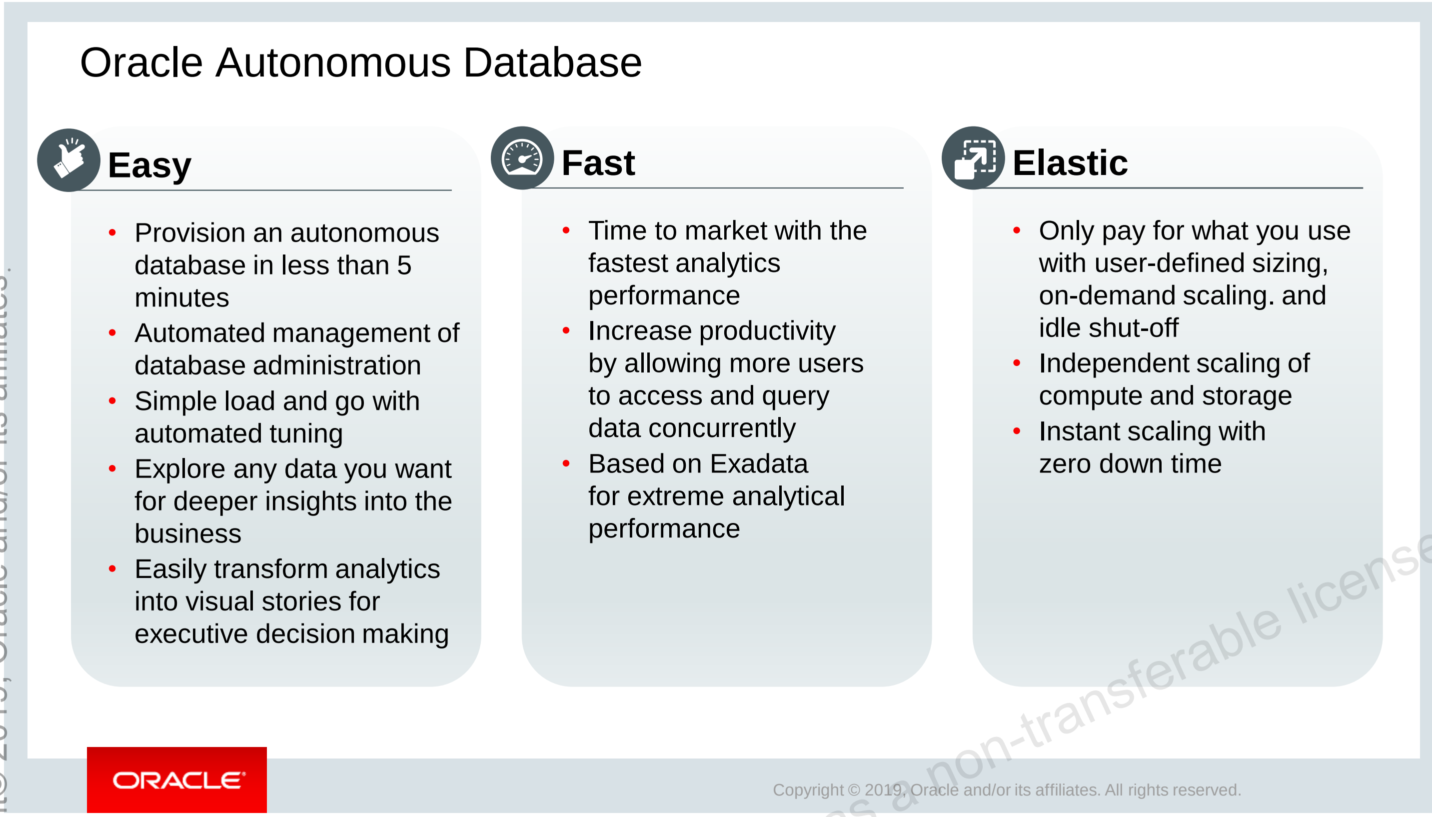
Using Autonomous Database with SQL Developer, you can:

- Connect to Autonomous Database
- Create tables in Autonomous Database
- Load data into Autonomous Database
- Copy tables to Autonomous Database
- Transfer a schema to Autonomous Database

Business Intelligence Tools

Autonomous Database is compatible with a number of business intelligence and data visualization tools from Oracle and trusted third parties:

- Oracle Analytics Cloud
- Oracle Data Visualization Desktop
- Third-party Business Intelligence tools



Provisioning requires only five simple things:

- What is the database name?
- Which is the data center?
- How many CPUs?
- How many TBs?
- What is the admin password?

A new service can be created in less than 5 to 10 minutes (regardless of size).

- Ready to connect via sqlnet

Automated management of Oracle automates end-to-end management of data warehouse.

- Provisioning new database instances
- Growing/shrinking storage and/or compute
- Patching and upgrades
- Backup and recovery

Full life cycle managed using the ADWC service console

Automated Tuning

"Load and go"

- Define tables, load data, run queries
  - No tuning
  - No special database expertise required
- Good performance out-of-the-box
- Query using any business analytics tool or cloud service
  - Built-in SQL notebook also included

Oracle Autonomous Database: Easy

Connect & Integrate

Plug & play connectors for easy data loading

Explore & combine any data you want

Store & Process

Simple load & go with automated tuning

Easy access to quality data you need

Analyze & Visualize

Increased self-service to blend data sources

360 degree view of the business

ORACLE

Oracle Autonomous Database: Fast

Connect & Integrate

Offload heavy data workloads using ELT

Tuned for high speed ingest and transformation

Store & Process

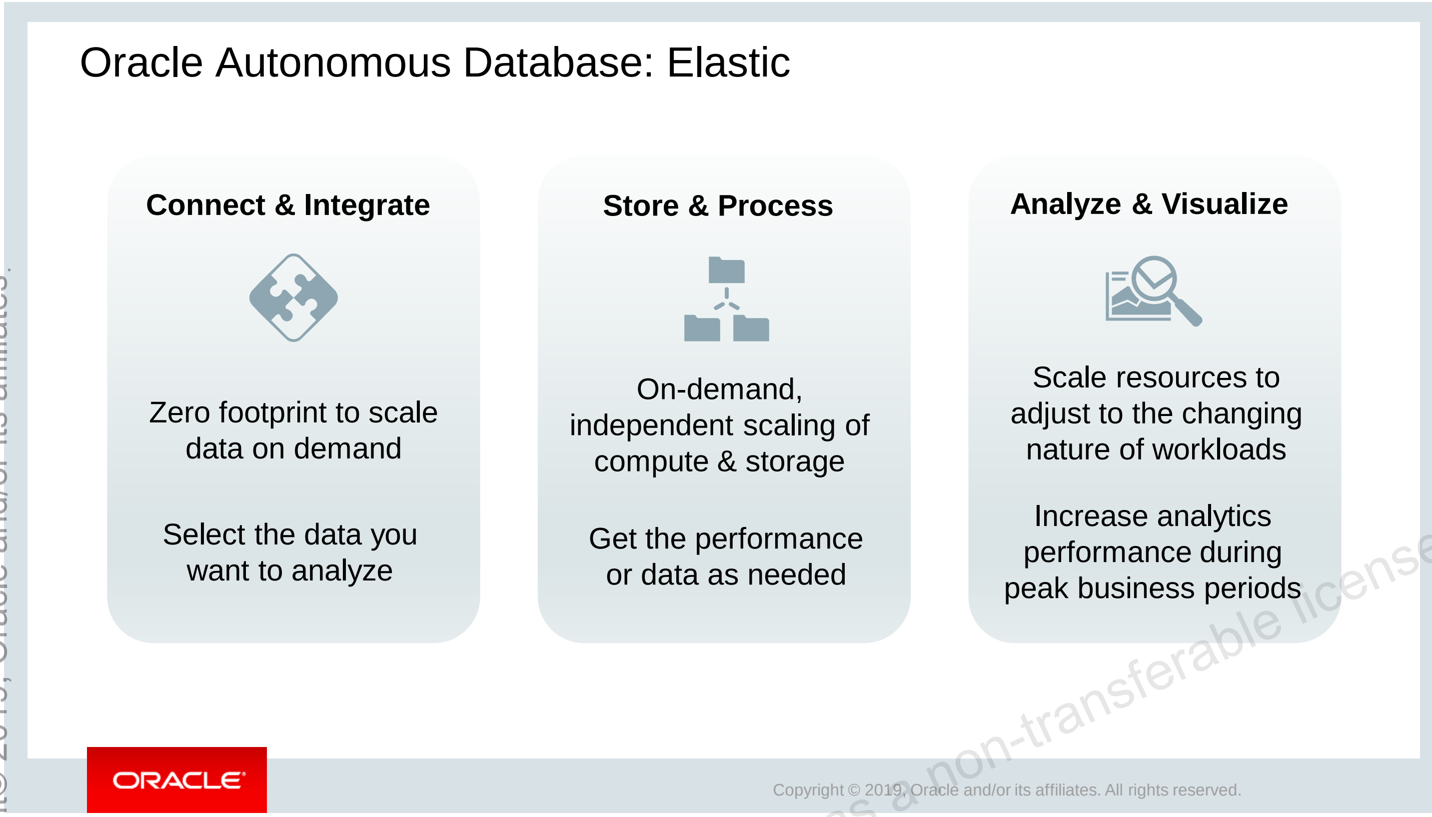
Runs on Exadata for extreme performance

Designed for fast analytical queries

Analyze & Visualize

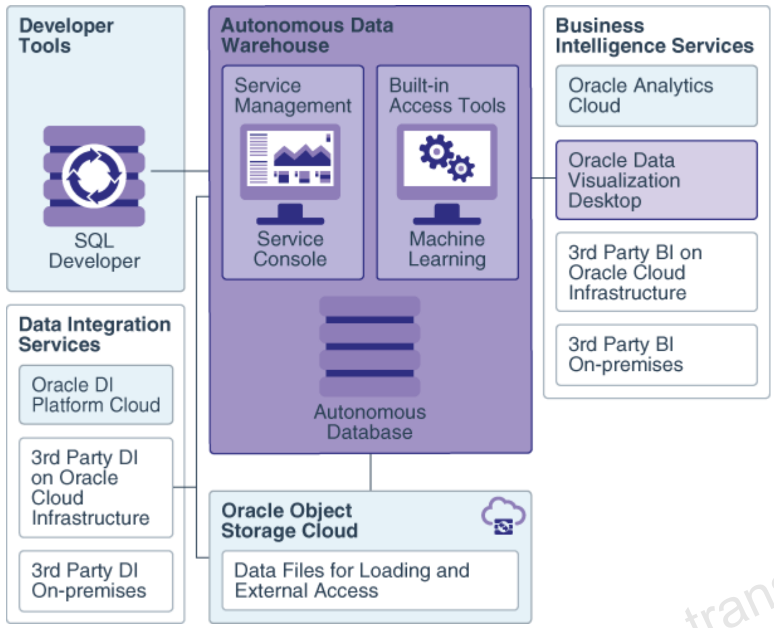
New analytical data mart ready in <1min

Boost productivity with data-driven self-service


Oracle Autonomous Database: Elastic
Connect & Integrate
Zero footprint to scale data on demand
Select the data you want to analyze
Store & Process
On-demand, independent scaling of compute & storage
Get the performance or data as needed
Analyze & Visualize
Scale resources to adjust to the changing nature of workloads
Increase analytics performance during peak business periods
ORACLE
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Oracle Autonomous Data Warehouse

Autonomous Data Warehouse Architecture



Autonomous Data Warehouse Architecture

Autonomous Data Warehouse provides an easy-to-use, fully autonomous data warehouse that scales elastically, delivers fast query performance, and requires no database administration. It is designed to support all standard SQL and business intelligence (BI) tools, and provides all the performance of the market-leading Oracle Database in an environment that is tuned and optimized for data warehouse workloads.

Autonomous Data Warehouse components, including a center box with Service Management (service console), Built-in Access Tools (Oracle Machine Learning), and the Autonomous Database. This connects to Developer Tools (SQL Developer), Data Integration Services: Oracle DI Platform Cloud, 3rd Party DI on Oracle Cloud Infrastructure, and 3rd Party DI On Premises. Autonomous Data Warehouse connects to Business Intelligence Services: Oracle Analytics Cloud, Oracle Data Visualization Desktop, 3rd Party BI on Oracle Cloud Infrastructure, and 3rd Party BI On Premises. There is also a connection from Autonomous Data Warehouse to Oracle Object Storage Cloud. Object Storage contains the data files for loading external access.

Oracle Autonomous Data Warehouse Customer Benefits

Lower Cost

- Reduce administration costs with complete automation of operations and tuning.
- Reduce runtime costs by paying only for the resources needed at any given time.
- Reduce data management costs.

Accelerate Innovation

- Provision a data warehouse in seconds and accelerate time to innovation.
- Instantly scaling to meet fluctuating business demands without interrupting the business increases innovation momentum.
- Built-in AI and ML accelerate the speed to uncover trends & anomalies for new revenue opportunities or operational efficiencies.

Predictive Insights

- Proactively discover new insights with ML-based continuous data analysis.
- Analyze data using AI with predictive data visualization, narration, & intelligent data discovery.
- Get proactive insights with comprehensive, in-database analytics for the SQL, ML, Graph, R, and Times series.

ORACLE

Lower Cost

- Reduce admin costs up to 80% with complete automation of operations and tuning.
- Reduce runtime costs up to 70% by dynamically adjusting resources, eliminating underutilization.
- Deploy new apps in minutes vs months, which saves tens of thousands of dollars; faster TTI/TTD.
- Reduce labor cost and lower risk of costly human error by automated provisioning, patching, tuning, securing, monitoring.
- Consume fewer resources with higher performance database and platform.

Reduce Risk

- Be protected from attacks by automatically applying security updates.
- Mitigate breach impact by avoiding reputational damage, associated breach costs, and revenue losses.
- By eliminating human intervention, ADWC removes the most common cause of system/data errors.

Increase Productivity

- Gain insights from data in just a few minutes, not days and weeks (provision, load data, and start running queries in minutes).
- Built-in AI and ML helps find patterns that could identify undiscovered anomalies or new trends that open up new opportunities.
- Drive innovation and gain a competitive advantage from your data instantly by provisioning a data warehouse in seconds with fully automated management.
- Business analysts and developers can quickly load their data and be fully productive.

Increase Agility

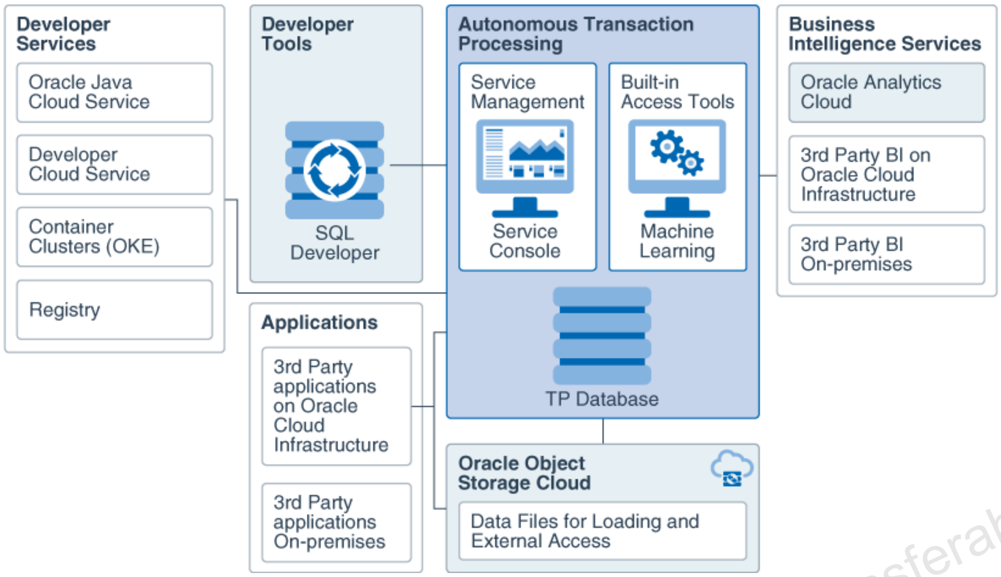
- With elastic scalability, you can scale compute and storage up or down, while the system is still running to meet the varying, seasonal demands of the business.
- Businesses can instantly adapt to fluctuating cycles that could be high or low.
- This means you only pay for the compute and storage in use. Use idle shut off if you are not using compute resources (for example, over the weekend).

Oracle Autonomous Transaction Processing

ORACLE AUTONOMOUS DATABASE

ORACLE

Autonomous Transaction Processing Architecture



Autonomous Transaction Processing is designed to support all standard business applications and deliver scalable query performance.

Autonomous Transaction Processing provides all the performance of the market-leading Oracle Database in an environment that is tuned and optimized for transaction processing workloads.

As a service, Autonomous Transaction Processing does not require database administration. With Autonomous Transaction Processing, you do not need to configure or manage any hardware or install any software. It handles creating the database, backing up the database, patching and upgrading the database, and growing or shrinking the database.

The Autonomous Transaction Processing components, including a center box with Service Management (service console), Built-in Access Tools (Oracle Machine Learning), and the Transaction Processing (TP) database. This connects to Developer Tools (SQL Developer), Applications: Third-party applications on the Oracle Cloud Infrastructure and third-party applications On Premises. Autonomous Transaction Processing also connects to Business Intelligence Services: Oracle Analytics Cloud, 3rd Party BI on Oracle Cloud Infrastructure, and 3rd Party BI On Premises. There is also a connection from Autonomous Transaction Processing to Oracle Object Storage Cloud. Object Storage contains the data files for loading external access.

Autonomous Transaction Processing: Customer Benefits

Lower Cost

- Complete automation of database and infrastructure operations cuts administrative costs.
- The efficiency of a self-optimizing database, together with elastic pay-per-use, cuts run time costs.

Accelerate Innovation

- It is easy to develop and deploy new applications with no complex management or tuning required.
- Oracle's extreme performance cloud infrastructure supports any size workload, allowing for unlimited cloud flexibility.

Reduce Risks

- Automatic application of the latest security updates with no down time eliminates cyber attack vulnerabilities.
- There is protection from all types of failures, including system failures, regional outages, and user errors.
- It has high availability, or just minutes of down time a month, including maintenance.
- Database Vault prevents administrators from snooping on user data.

ORACLE

Lower Cost

- Reduce admin costs up to 80% with complete automation of operations and tuning.
- Reduce runtime costs up to 70% by dynamically adjusting resources, eliminating underutilization,
- Deploy new apps in minutes vs. months, saving tens of thousands of dollars; faster TTI/TTD.
- Reduce labor cost and lower risk of costly human error by automated provisioning, patching, tuning, securing, and monitoring.
- Consume fewer resources with higher performance database and platform.

Reduce Risk

- Be protected from attacks by automatically applying security updates.
- Mitigate breach impact by avoiding reputational damage, associated breach costs, and revenue losses.
- By eliminating human intervention, ATP removes the most common cause of system/data errors.

Increase Agility

- Elastic scalability to scale compute and storage up or down, while the systems are still running to meet the varying, seasonal demands of the business
- Helps the business instantly adapt to fluctuating cycles that could be high or low
- This means you only pay for the compute and storage in use. Use idle shut off if you are not using compute resources (for example, over the weekend).

Typical Workflow of Autonomous Database

Typical Workflow of Autonomous Database

- Log in to your Oracle Cloud account.
- Provision Autonomous Database with the required workload.
- Start the database.
- Create database users and obtain security credentials.
- Connect to your database.
- Scale the database.
- Load data into the database.
- Monitor the database.
- Manage the database.

ORACLE

- **Create and log in to your cloud account:**
  - Provide your information and sign up for an Oracle Cloud Service.
- **Provision Autonomous Database:**
  - Create an Autonomous Database.
- **Start the database:**
  - Start and verify that an Autonomous Database instance is up and running.
- **Create database users and obtain security credentials:**
  - Create accounts for your users and assign them appropriate privileges.
- **Connect to your database using SQL Developer:**
  - Download SQL Developer or another database client tool. Obtain security credentials and connect to your Autonomous Database.
- **Connect to your database using a database client:**
  - Download a database client. Obtain security credentials and connect to your Autonomous Database instance.
- **Scale the database:**
  - Use the Service Console to scale an instance to add or remove CPU or storage resources.
- **Load data into the database:**
  - Use the available tools to load data into the database.

- **Monitor the database:**
  - Check on the health and performance of the database.
- **Manage the database:**
  - Check on the day-to-day operation of your service, monitor performance, and review important notifications.
  - All aspects of the database lifecycle are simplified by the service, including patching and upgrading the database. You can obtain service information, including the service and database version, from the service console.

Using Oracle Machine Learning with Autonomous Database

- Sift through large amounts of data.
- Create models that find patterns.
- Create valuable new insights and predictions.
- Let ML do the exploring for you.
- Identify unique new trends.
- Make decisions that transform your business.

ML automatically sifts through large volumes of data, which means you don't have to spend hours and hours manually pouring over the data yourself.

As it is doing that, it is looking to find patterns that otherwise would have been hidden (for example, spotting an increase in social activity and buzz around restaurants in a certain area offering perhaps certain food types might mean an opportunity for an entrepreneur). Those patterns/trends can then be picked up by analysts/business users and used to make certain predictions based on historical data, such as blood hound could be looking at vibration levels over the desert track and predict that the stress caused on the wheels might make it fail at a certain point, or that if it remains below a certain level it can push the speed up to the next level.

Using Oracle Machine Learning with Autonomous Database

Oracle Machine Learning can be accessed via the Autonomous Database Service console.

Key features of Oracle Machine Learning (OML):

- Provides steps to access and create OML users
- Allows collaboration among data scientists, developers, and business users
- Leverages the scalability and performance of Oracle Platform and its cloud services

ORACLE

ADB provides steps to access and create Oracle Machine Learning users. Oracle Machine Learning provides a notebook style application designed for advanced SQL users and provides interactive data analysis that lets you develop, document, share, and automate reports based on sophisticated analytics and data models.

Key Features of Oracle Machine Learning

- Allows collaboration among data scientists, developers, business users
- Leverages the scalability and performance of Oracle Platform and its Cloud Services

To use Oracle Machine Learning with Autonomous Transaction Processing, you need to create users and access the system:

- **OML User Management** lets the **Admin** (user with administrative privileges) create and modify Oracle Machine Learning user accounts.
- **OML Application:** Application **users** access Oracle Machine Learning to create, view, and share notebooks for data analytics, data visualization and other Oracle Machine Learning tasks.

To use Oracle Machine Learning, from the Autonomous Transaction Processing Service console:

- Click the **Administration** tab.
- Select **Manage Oracle ML Users**.

This provides access to OML User Management. Access the User Management page as Admin and create and manage Oracle Machine Learning user accounts. After you create user accounts, you or users that you provide credentials to, can use the user credentials and password to access and work with Oracle Machine Learning.

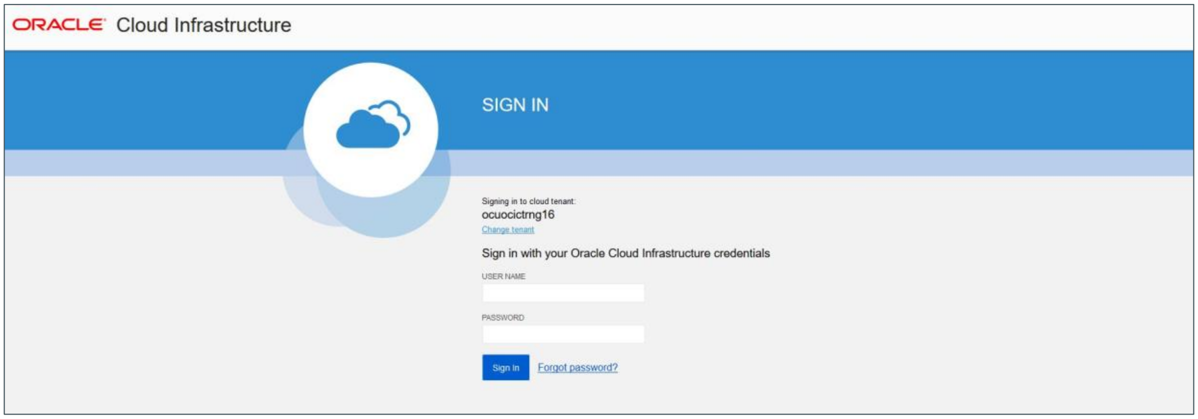
Working with Oracle Analytics

By building reports and dashboards with analytics in Autonomous Database, you can:

- Use Oracle Analytics with Autonomous Database
- Use Oracle Analytics Cloud to select interactive visualizations and automatically create advanced calculations to reveal the insights in your data

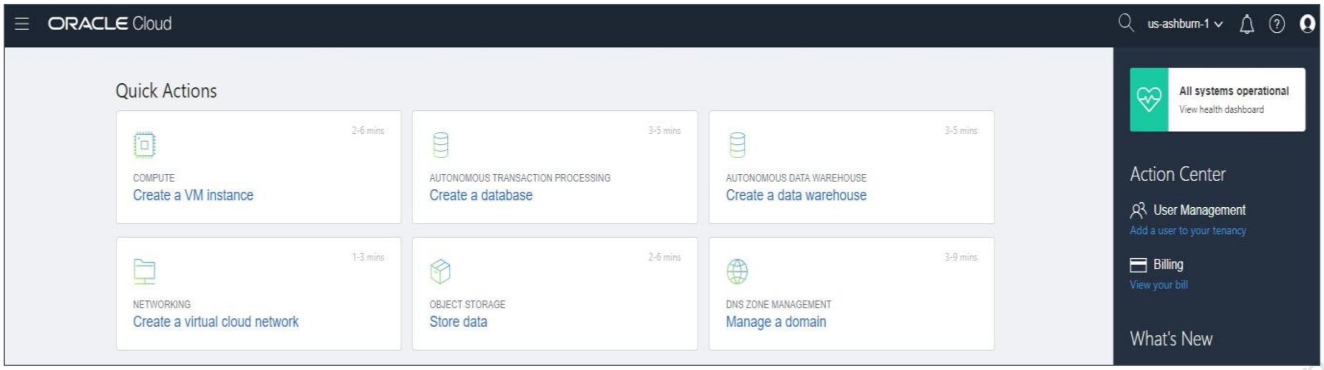
The Oracle Cloud Infrastructure Console

Logging In to Your Cloud Account



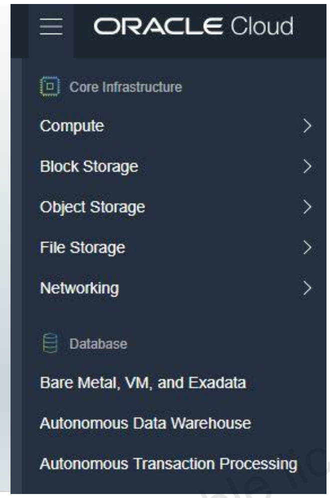
ORACLE

The Dashboard of Your Cloud Account



Services Assigned to Your Cloud Account

- In the Navigation menu, you will see a list of cloud services you have with your cloud account. They will be grouped by the category of the cloud service model.
- Select Autonomous Data Warehouse or Autonomous Transaction Processing to go to the home page.



ORACLE

Summary

In this lesson, you should have learned how to:

- Give an overview of Oracle Autonomous Database
- List the key features of Autonomous Database
- Explain the typical workflow of Autonomous Database
- Use Oracle Machine Learning with Autonomous Database
- Describe the Oracle Cloud Infrastructure Console

ORACLE

Practice 4: Overview

- Practice 4-1: Understanding the Lab Environment
- Practice 4-2: Exploring the Oracle Cloud Infrastructure Console

5

Provisioning an Oracle Autonomous Database Instance

ORACLE

Objectives

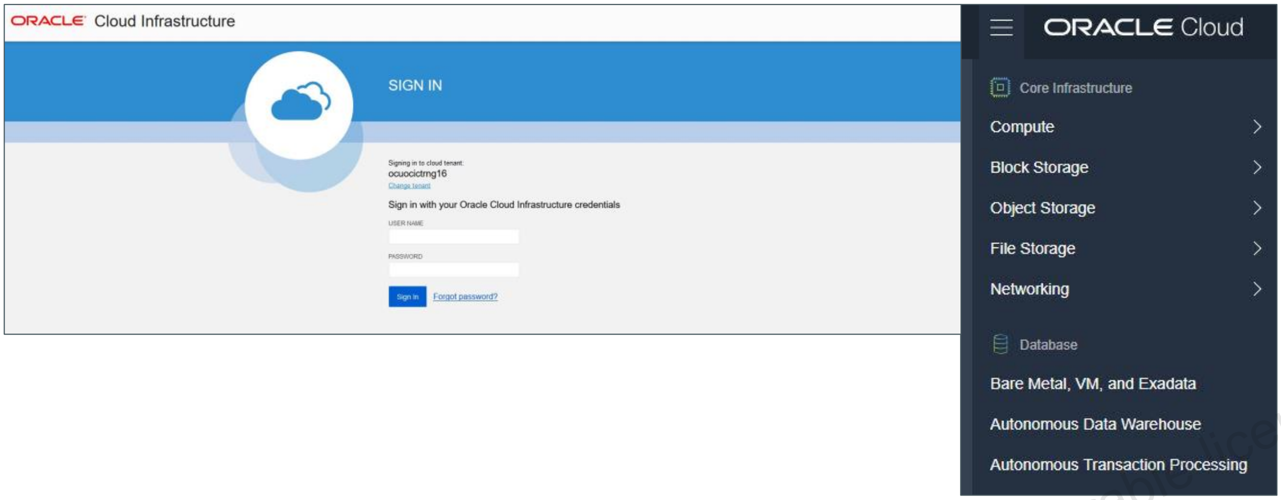
- Provision an Autonomous Database Instance with ADW Workload
- Provision an Autonomous Database Instance with ATP Workload
- Create a Clone from an Existing Autonomous Database

Creating an Oracle Autonomous Database

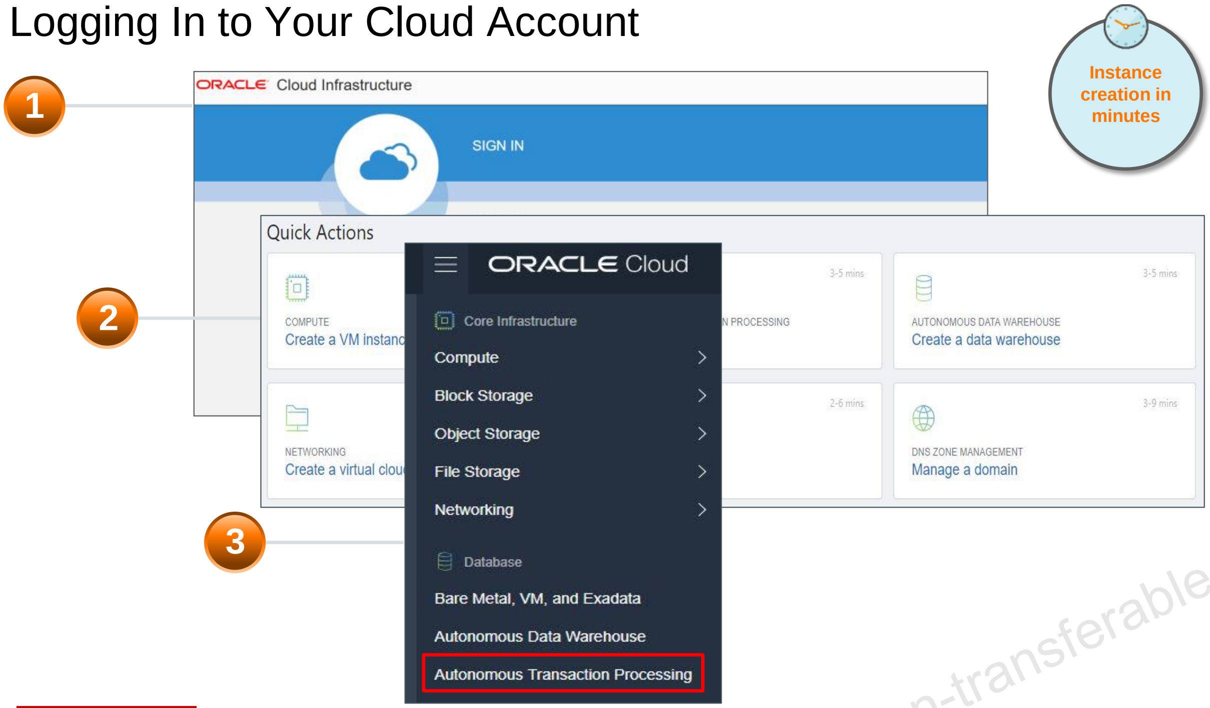
What Do You Need?

- A subscription to Oracle Autonomous Database (ADB)
- Tenancy in Oracle Cloud Infrastructure, including the creation of a compartment and the setting of access to resources
- Oracle SQL Developer

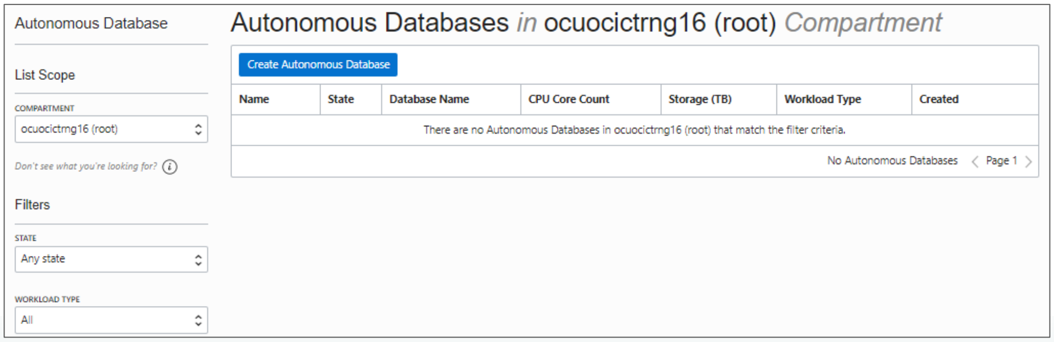
Logging In to Your Cloud Account



Logging In to Your Cloud Account



Compartments in Autonomous Databases



- Compartments help you organize resources to make it easier to control access to them.
- The root compartment is created by Oracle when a tenancy is provisioned.
- An administrator can create more compartments in the root compartment.
- By default, the page points to the root compartment.
- Select the assigned compartment to create an Autonomous Database.

ORACLE

Creating an Autonomous Database Instance

Before you create an Autonomous Database instance:

- Sign up for a free Oracle Cloud trial subscription or purchase a subscription
- Leverage an object store for data loading (Optional)
  - Oracle Cloud Infrastructure Object Storage
  - Oracle Cloud Infrastructure Object Storage Classic
- Manually back up for Autonomous Database (Optional)
  - Back up Autonomous Database to cloud storage
  - Associate an instance with a cloud storage backup location
  - Create a bucket for cloud storage manual backups

ORACLE

Creating an Autonomous Database Instance

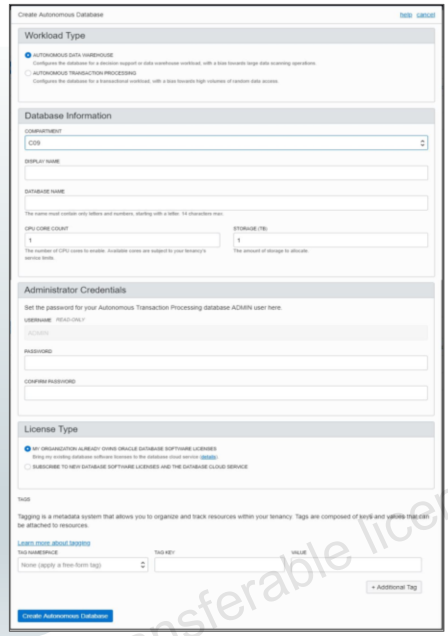
Creating an Autonomous Database

Provisioning requires only a few simple bits of information:

- Workload Type
- Database Name
- Number of CPUs
- Number of TBs
- Admin Password

The new service is created in minutes (regardless of size).

- The database is open and ready for connections.



ORACLE

The Create Autonomous Database Dialog Box

- Workload Type:
  - Autonomous Data Warehouse
  - Autonomous Transaction Processing
- Display Name
  - Name of the Instance
- Database Name
  - Name of the database
- CPU Core Count
  - Number of CPUs for your service
- Storage (TB)
  - Actual space available in terabytes to your service instance, including system-related space allocations
- Administrator Password
  - Password for the ADMIN user of the service instance

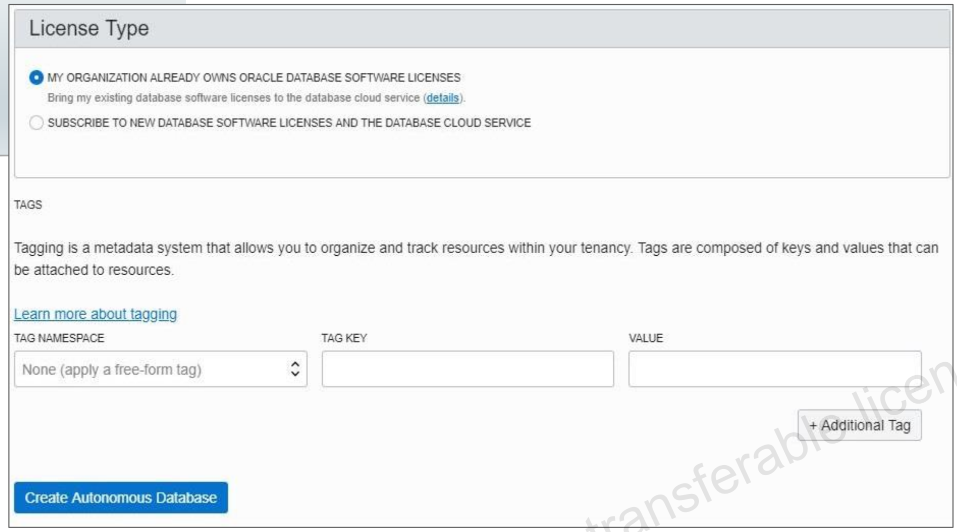
ORACLE

The Create Autonomous Database Dialog Box

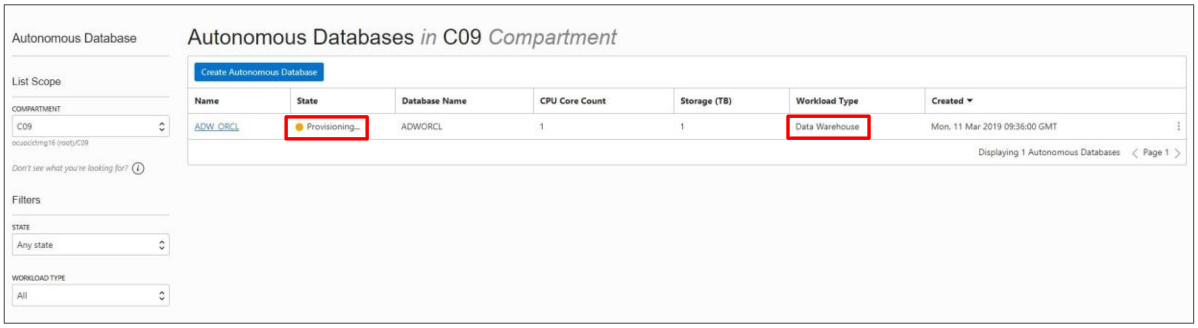
License Type

Use an included license

Bring your own license

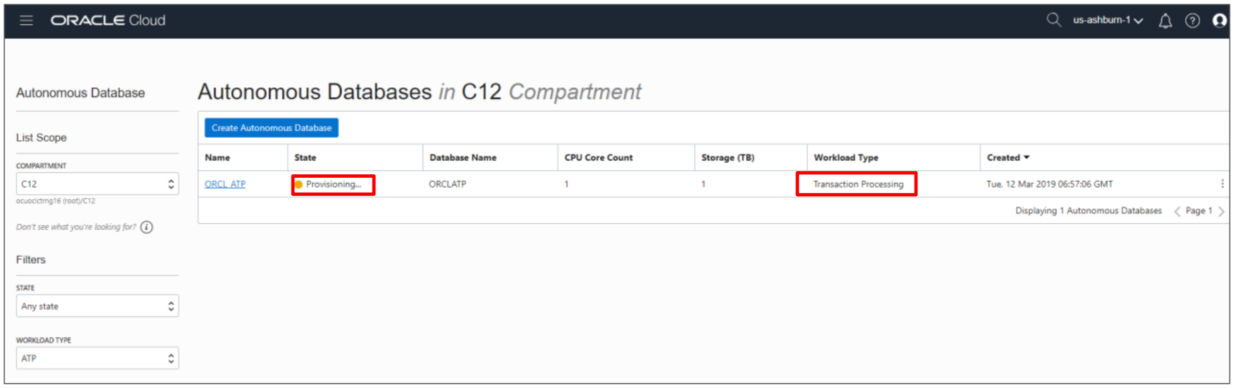


Autonomous Database with Data Warehouse Workload



- On the console, the State field indicates that the ADB with Data Warehouse (ADW) workload is **Provisioning**.
- When creation is completed, the State field changes from Provisioning to **Available**.

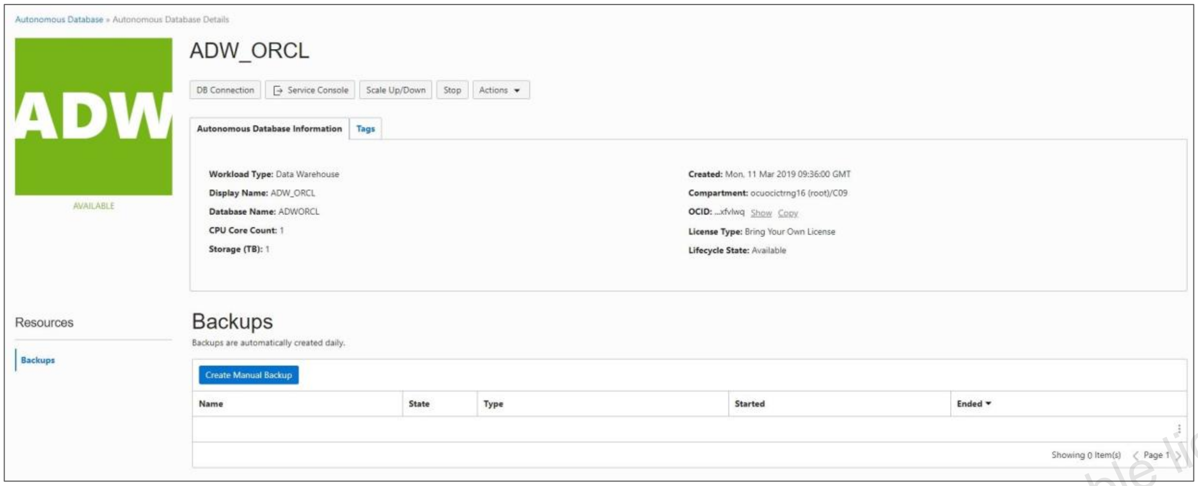
Autonomous Database with Transaction Processing Workload



- On the console, the State field indicates that the ADB with Transaction Processing (ATP) workload is **Provisioning**.
- When creation is completed, the State field changes from Provisioning to **Available**.

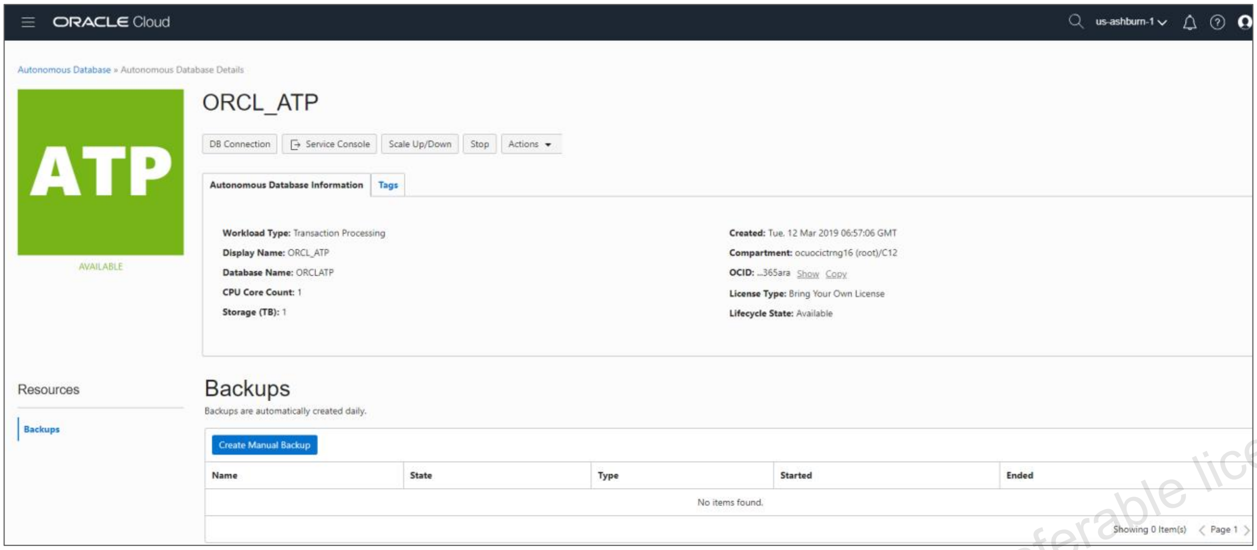
ORACLE

Autonomous Database: Details of the Instance with ADW Workload



ORACLE

Autonomous Database: Details of the Instance with ATP Workload



ORACLE

What Didn't You Just See?

No decisions for:

- Details of the database software
- Configuration of the hardware
- Characteristics of the database
- Architecture of backups and availability

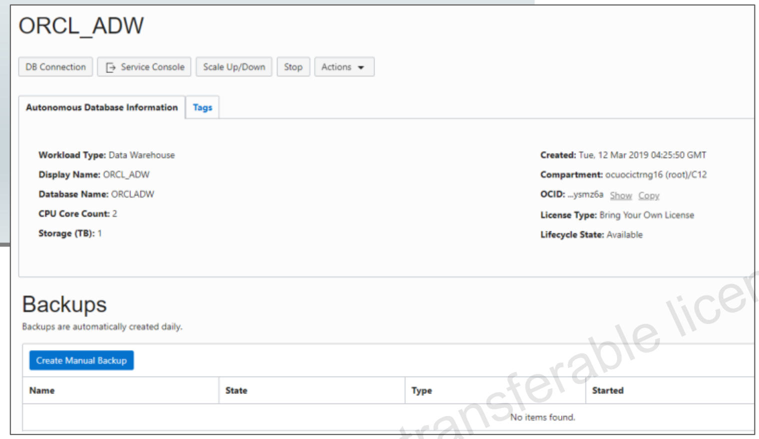
All of this (and more) is automatically configured.

Automated Management

Oracle automates end-to-end management of Autonomous Databases.

- Provisioning new database instances
- Growing/shrinking storage and CPU
- Patching and upgrades
- Backup and recovery

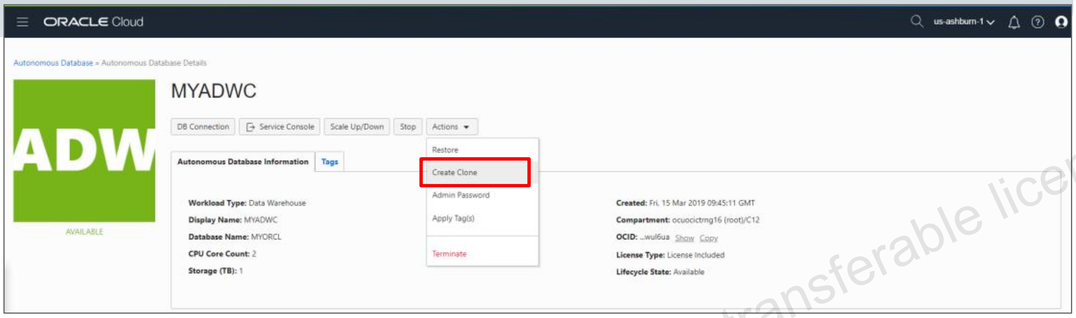
The full life cycle is managed using ADB Service Console.



ORACLE

Cloning an Autonomous Database

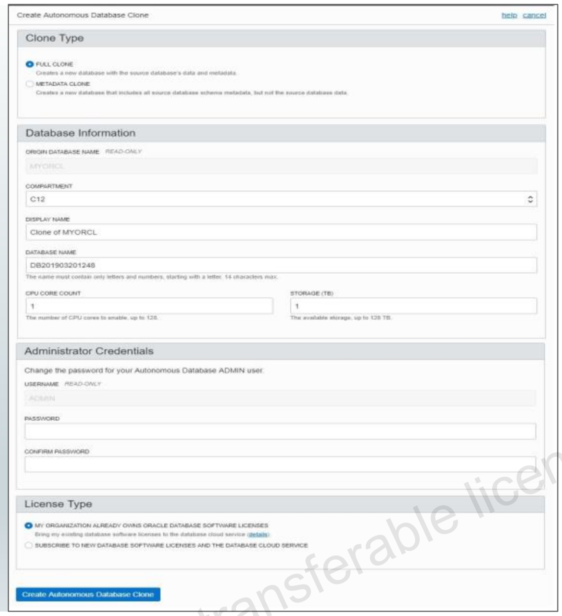
- Clone an Autonomous Database using the Oracle Cloud Infrastructure console or API.
- The point-in-time copy of the Autonomous Database can be created using the Create Clone option.
- A clone can be used for testing, development, or analytics purposes.
- The **Full Clone** option can be used to create a new database with the source database's data and metadata.
- The **Metadata Clone** option can be used to create a new database with source database schema metadata.



ORACLE

Creating a Clone of an Autonomous Database

1. Open the navigation menu.
   Under **Database**, click **Autonomous Transaction Processing** or **Autonomous Data Warehouse**.
2. Choose your **Compartment**.
3. In the list of Autonomous Databases, click the display name of the database you want to clone.
4. Hover over the **Actions** button, and then click **Create Clone**.
5. In the **Create Autonomous Database Clone** dialog box, enter the required details.



ORACLE

Post Autonomous Database Creation Tasks

After you create an Autonomous Database instance:

- Reset the administrator password
- Add Oracle Machine Learning users
- Download Oracle Data Visualization Desktop and add a connection to Autonomous Database

You will perform these tasks during the practice session.

Summary

In this lesson, you should have learned how to:

- Provision an Autonomous Database with Data Warehouse Workload
- Provision an Autonomous Database with Transaction Processing Workload
- Create a clone from an existing Autonomous Database

ORACLE

Practice 5: Overview

- Practice 5-1: Creating an ADB with Data Warehouse Workload on Oracle Cloud Infrastructure (OCI)
- Practice 5-2: Creating an ADB with Transaction Processing Workload on Oracle Cloud Infrastructure (OCI)
- Practice 5.3: Creating a Clone from an Existing Autonomous Database

6

Connecting to Autonomous Database

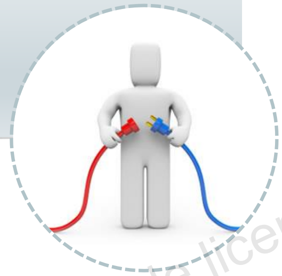
ORACLE

Objectives

- Connect to Autonomous Database Using a Client Application
- Connect to Autonomous Database Using Oracle Database Tools

Connecting to the Database

- Secure a connection using credential wallets via SQL*Net, JDBC, ODBC.
- Download Wallet from the service console.



Connecting to an Autonomous Database Instance

Applications and tools connect to an Autonomous Database instance using Oracle Net Services (also known as SQL*Net).

Oracle Net Services enables a network session from a client application to an Oracle Database server.

You can connect using the following connection types:

- Oracle Call Interface
- ODBC Drivers
- JDBC OCI
- JDBC Thin Driver

ORACLE

Connecting to an Autonomous Database Instance

Autonomous Database is preconfigured to support Oracle Net Services (a TNS listener is installed and configured to use secure TCPS and client credentials). The client computer must be prepared to use Oracle Net Services to connect to Autonomous Database.

When a network session is established, Oracle Net Services acts as the data courier for both the client application and the database. It is responsible for establishing and maintaining the connection between the client application and the database, as well as exchanging messages between them.

Oracle Net Services supports a variety of connection types to Autonomous Database, including:

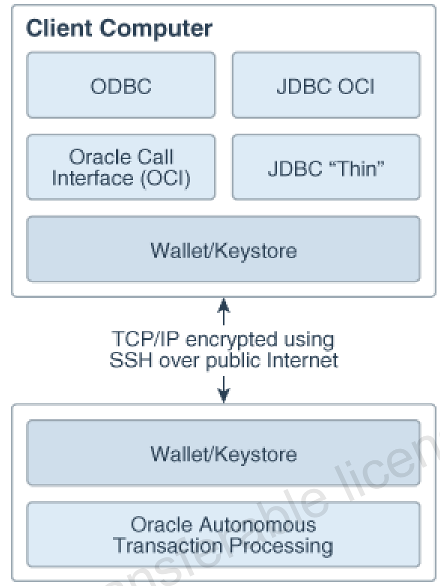
- Oracle Call Interface (OCI), which is used by many applications written in C language. Examples include Oracle utilities such as Oracle SQL*Plus, SQL*Loader, and Oracle Data Pump.
- ODBC drivers, which can be used by applications running on Microsoft Windows. ODBC drivers are layered over Oracle Call Interface (OCI).
- JDBC OCI, which is used by Java language applications. JDBC OCI adds a layer over Oracle Call Interface for Java applications. The Oracle SQLcl command-line interface uses JDBC OCI.
- JDBC Thin Driver, also for Java applications, which is a pure Java driver. Oracle SQL Developer supports JDBC Thin Driver connections.
- Java Connectivity with ADB using 18.3 JDBC thin driver needs JDK version 8 and above.
- If the client is behind a firewall and configurations requires an HTTP proxy to connect to the internet then JDBC thin client 18.1 to enable connections through HTTP proxies.
- Database links to other databases are not allowed for enhanced security. Database links from other databases to Autonomous Database are allowed.

Third-party products and custom applications may use any of these connection types. Refer to this link for more details:

https://docs.oracle.com/en/cloud/paas/atp-cloud/atpug/connect-preparing.html

Securing Connections to an Autonomous Database (ADB) Instance

- Connections to an ADB instance are made over the public Internet.
- All applications use a secure connection to the ADB instance.
- All connection types use certificate authentication and Secure Sockets Layer (SSL).
- The client credentials zip file, available after instance creation, contains security credentials necessary to establish a secure connection to the ADB database instance.



The graphic in the slide shows a client secure connections to an Autonomous Transaction Processing instance.

Many applications provide support for more than one connection type, but each type of connection to Autonomous Transaction Processing uses certificate authentication and Secure Sockets Layer (SSL). This ensures that there is no unauthorized access to Autonomous Transaction Processing and that communications between the client and server are fully encrypted and cannot be intercepted or altered.

Certification authentication uses an encrypted key stored in a wallet on both the client (where the application is running) and the server (where your database service on the Autonomous Transaction Processing is running). The key on the client must match the key on the server to make a connection. A wallet contains a collection of files, including the key and other information needed to connect to your database service in Autonomous Transaction Processing. All communications between the client and the server are encrypted.

Connecting to an Autonomous Database Using Oracle Database Tools

ORACLE AUTONOMOUS DATABASE

ORACLE

Downloading Client Credentials (Wallet)

To download the Client Credentials Wallet:

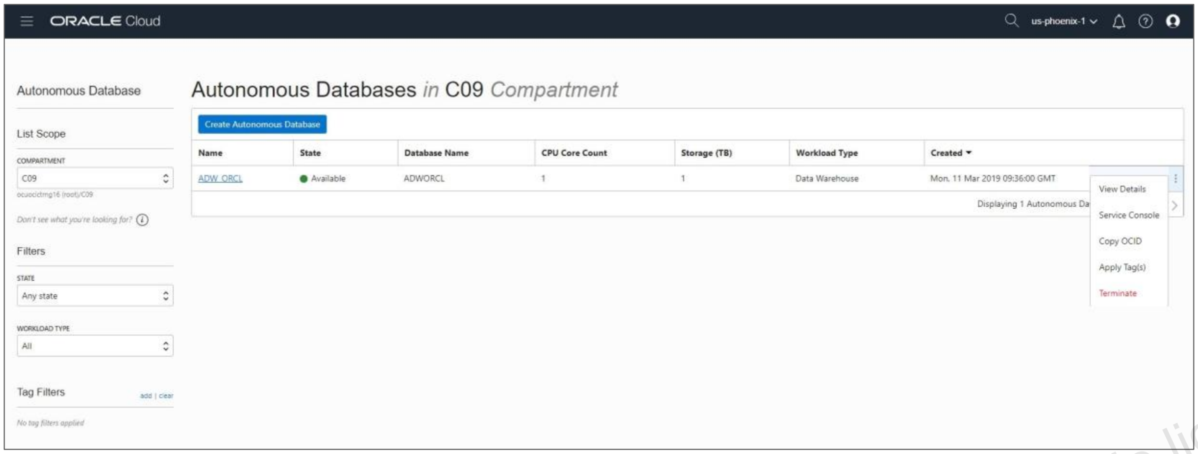
- Navigate to the Service Console for Autonomous Database.
- Select **Administration**.
- On the **Administration** page, click **Download Client Credentials (Wallet)**.
- On the **Client Credentials** dialog box, enter a wallet password and confirm the password.
- Click **Download** to save the client credentials zip file.

Protect this file to prevent unauthorized database access.

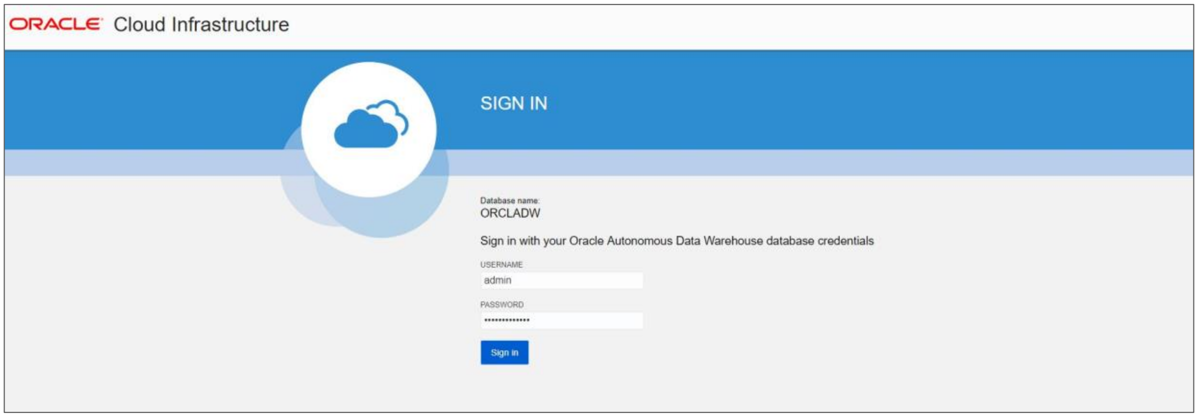
Wallet files, along with the Database user ID and password, provide access to data in your Autonomous Data Warehouse Cloud.

Oracle client credentials (wallet files) are downloaded from the Autonomous Database administrative service console by a service administrator. If you are not an Autonomous Database administrator, your administrator should provide you with the client credentials.

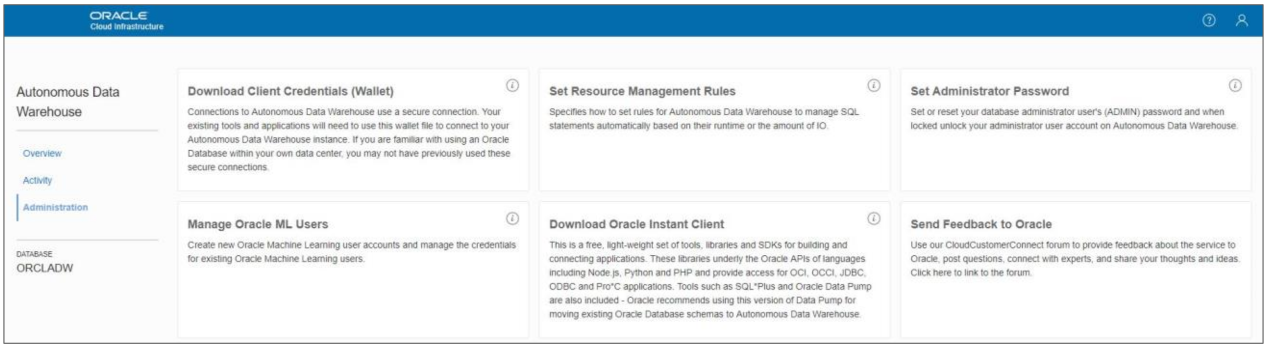
Launching Autonomous Database Service Console



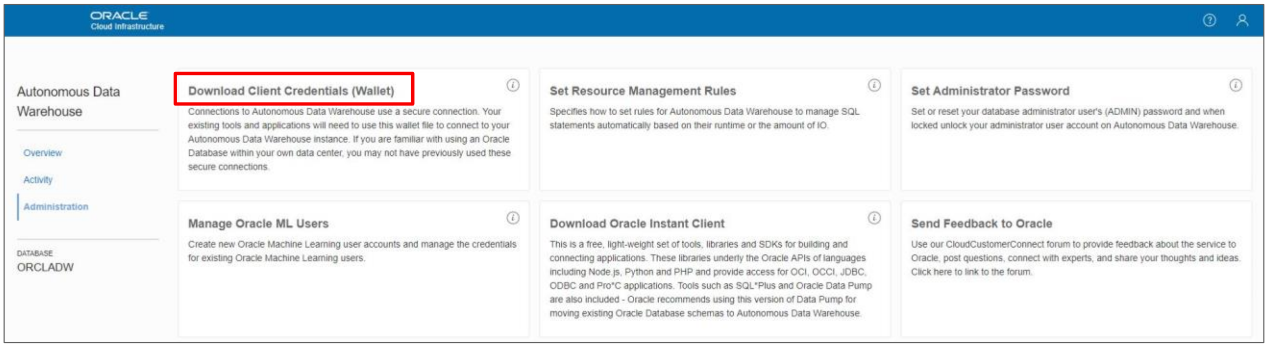
Logging In to Autonomous Database Service Console



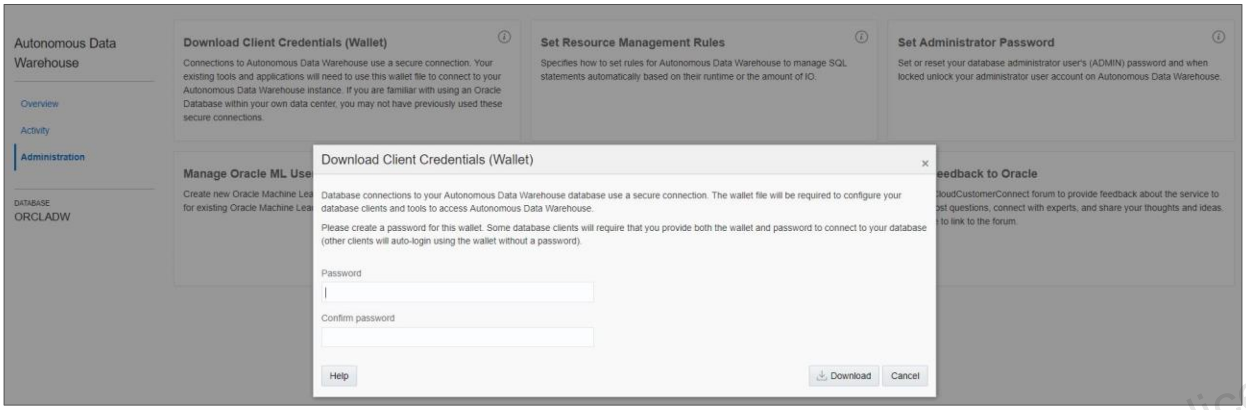
Administration: Autonomous Database Service Console



Downloading Client Credentials



Downloading Client Credentials



Connecting to Autonomous Database Using Oracle Database Tools

- Oracle Database Tools
  – SQL Developer, SQL*Plus, and SQLcl
- Connect with Oracle SQL Developer 18.2 (or later)
  – Connect to Autonomous Database using an Oracle Wallet
  – Enhancements for key Autonomous Database features
  – Support for wallet files using the Cloud PDB Connection Type
- To create a new connection to Autonomous Database:
  – Start Oracle SQL Developer
  – In the connections panel, right-click **Connections** and select **New Connection**.

ORACLE

Connecting to Autonomous Database Processing Through a Firewall

- Most organizations protect their network using firewall.
- The ADB instance is accessed using the public Internet.
- The default port number for ADB is 1522.
- Firewall must permit the use of this port to connect to the ADB instance.
- You can find the port number in the connection string from the `tnsnames.ora` file in your `credentials ZIP` file.

Most organizations protect networks and devices on a network using a firewall. A firewall controls incoming and outgoing network traffic using rules that allow the use of certain ports and access to certain computers (or, more specifically IP addresses or host names). An important function of a firewall is to provide separation between internal networks and the public Internet.

Autonomous Transaction Processing is accessed using the public Internet. To access Autonomous Transaction Processing from behind a firewall, the firewall must permit the use of the port specified in the database connection when connecting to the servers in the connection. The default port number for Autonomous Transaction Processing is 1522. (Find the port number in the connection string from the `tnsnames.ora` file in your `credentials ZIP` file.) For example, see the `port` value in the following `tnsnames.ora` file:

```
my_atpc_high = (description = (
                address=(protocol=tcps)
                (port=1522)
                (host=atpc.example.oraclecloud.com))
                (connect_data=(service_name=high.atp.oraclecloud.com))

  (security=(ssl_server_cert_dn="CN=atpc.example.oraclecloud.com,
                OU=Oracle BMCS US,O=Oracle Corporation,L=Redwood
City,ST=California,C=US")))
```

Your firewall must allow access to servers within the `.oraclecloud.com` domain using port 1522. To connect to Autonomous Transaction Processing, depending on your organization's network configuration, you may need to use a proxy server to access this port or you may need to request that your network administrator open this port.

Predefined Service Names for Data Warehouse Workload

Three predefined database services
- Choice of performance and concurrency

HIGH
- Highest resources, lowest concurrency
- Queries run in parallel

MEDIUM
- Less resources, higher concurrency
- Queries run in parallel

LOW
- Least resources, highest concurrency
- Queries run serially

Example of a database with 16 OCPUs

	No. of concurrent queries	Max idle time	CPU shares
HIGH	3	5 mins	4
MEDIUM	20	5 mins	2
LOW	32	1 hour	1

ORACLE

The `tnsnames.ora` file provided with the credentials zip file contains three database service names: High, Medium, and Low. The predefined service names provide different levels of performance and concurrency for Autonomous Data Warehouse.

- **High:** The High database service provides the highest level of resources to each SQL statement resulting in the highest performance, but supports the fewest number of concurrent SQL statements. Any SQL statement in this service can use all the CPU and I/O resources in your database. The number of concurrent SQL statements that can be run in this service is 3. This number is independent of the number of CPUs in your database.
- **Medium:** The Medium database service provides a lower level of resources to each SQL statement potentially resulting in a lower level of performance, but supports more concurrent SQL statements. Any SQL statement in this service can use multiple CPU and I/O resources in your database. The number of concurrent SQL statements that can be run in this service depends on the number of CPUs in your database and scales linearly with the number of CPUs.
- **Low:** The Low database service provides the least level of resources to each SQL statement, but supports the most number of concurrent SQL statements. Any SQL statement in this service can use a single CPU and multiple I/O resources in your database. The number of concurrent SQL statements that can be run in this service is twice the number of CPUs in your database.

The maximum idle time for the High, Medium, and Low services is set to 60 minutes. If your session stays idle for more than 60 minutes, it will be disconnected. Your session may also get disconnected if it stays idle for more than five minutes and other users' sessions require the resources consumed by the idle session. This allows resources to be freed for other active users in your database.

Predefined Service Names for Transaction Processing Workload

Five predefined database services for ATP:

TPURGENT

Highest priority application connection for time-critical operations. Manual parallelism.

TP

Typical application connection for transaction processing operations. No parallelism.

HIGH

High priority application connection for reporting and batch operations. All operations run in parallel.

MEDIUM

Typical application connection for reporting and batch operations. All operations run in parallel with the degree of parallelism limited to four.

LOW

A lowest priority application connection for reporting and batch operations. No parallelism.

ORACLE

The `tnsnames.ora` file provided with the credentials zip file contains five database service names: tpurgent, tp, high, medium, and low. The predefined service names provide different levels of performance and concurrency for Autonomous Transaction Processing.

tpurgent: The highest priority application connection service for time-critical transaction processing operations. This connection service supports manual parallelism.

tp: A typical application connection service for transaction processing operations. This connection service does not run with parallelism.

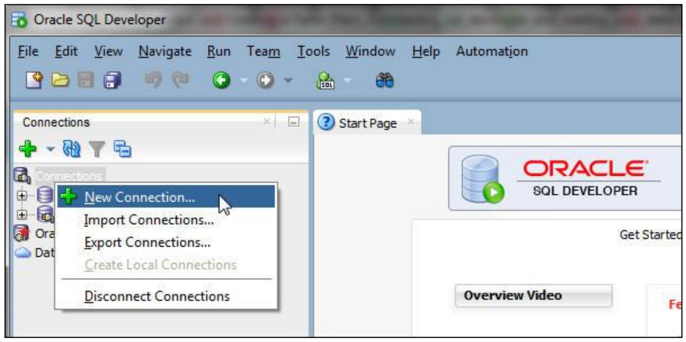
high: A high priority application connection service for reporting and batch operations. All operations run in parallel and are subject to queuing.

medium: A typical application connection service for reporting and batch operations. All operations run in parallel and are subject to queuing. Using this service, the degree of parallelism is limited to four (4).

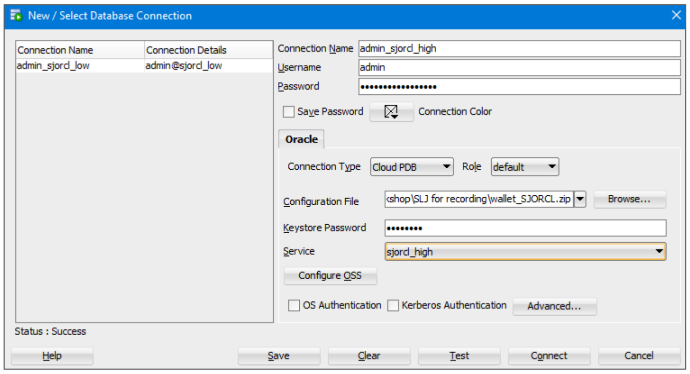
low: A lowest priority application connection service for reporting or batch processing operations. This connection service does not run with parallelism.

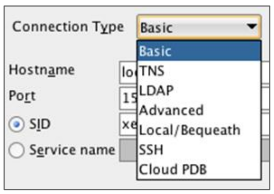
Sessions in these services may get disconnected if they stay idle for more than five (5) minutes and other users' sessions require the resources consumed by the idle session. This allows resources to be freed for other active users in your database.

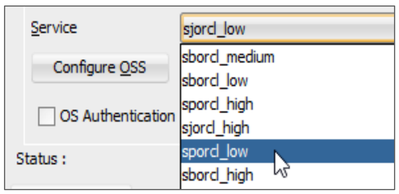
Creating a New Connection in SQL Developer to Autonomous Database



Credentials for a Database Connection from SQL Developer





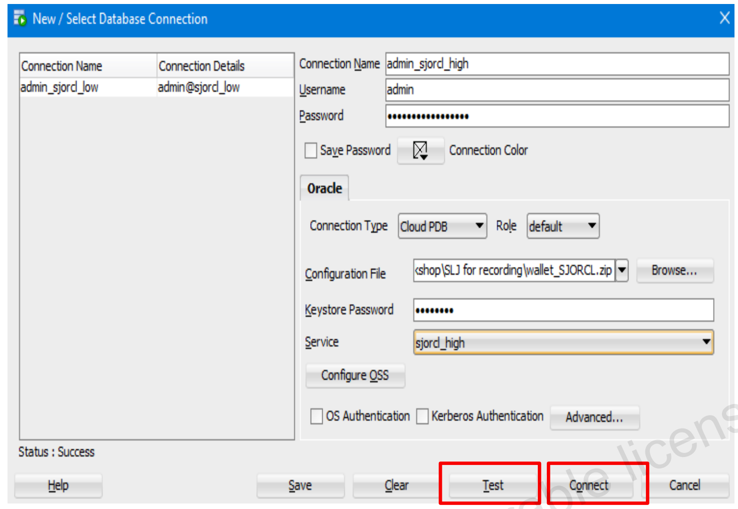


- Connection Name
  - Name of the connection
- Username
  - The database username
- Default administrator (ADMIN) provided as part of the service
- Password
  - Password for the database user
- Connection Type
  - Cloud PDB
- Configuration File
  - Browse and select the client credentials zip file

ORACLE

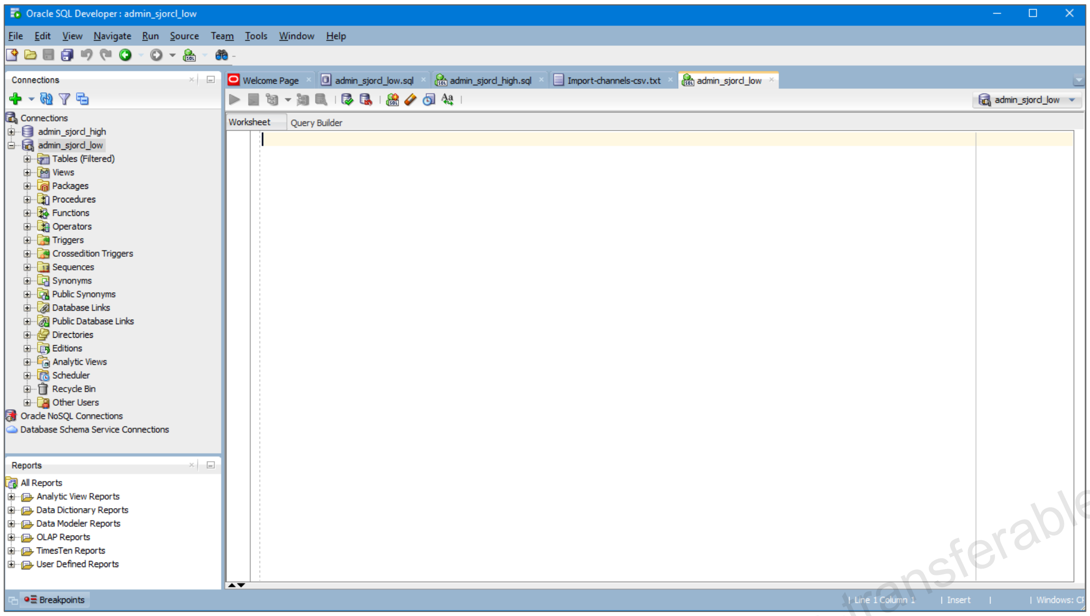
Connect to Autonomous Database Using SQL Developer

- Click test to check the connection to the database.
- Click Connect to connect to the database



ORACLE

SQL Developer Worksheet for Queries After Connection



Connecting to Oracle Analytics Cloud

1. Sign in to Oracle Analytics Cloud.
2. On the Home page, click **Connect to Oracle Autonomous Data Warehouse.**
3. In Create Connection, enter a **Connection Name** (for example, MyADW_connection).
4. In **Description**, enter a brief description.
5. Click **Select** next to **Client Credentials.** In File Upload, select the wallet zip file from your download location. The Client Credentials field is populated with cwallet.sso, and the Service Name field contains a value.
6. Enter your Oracle Autonomous Data Warehouse **Username** and **Password**.
7. From the **Service** list, select the service for your data, and then click **Save.**

ORACLE

Summary

In this lesson, you should have learned how to:

- Discuss connection options to Autonomous Database using a client application
- Describe a connection to Autonomous Database using Oracle Database tools

ORACLE

Practice 6: Overview

- Practice 6-1: Connecting to the Autonomous Database with Data Warehouse Workload Using SQL Developer
- Practice 6-2: Connecting to the Autonomous Database with Transaction Processing Workload Using SQL Developer
- Practice 6-3: Downloading, Extracting, and Staging Lab Files

ORACLE

7

Data Loading and Querying External Data with Autonomous Database

Objectives

- Describe Data Loading
- Load Data from Files in the Cloud
- Data Loading from Cloud Object Storage Using SQL Developer
- Data Loading from Cloud Object Storage Using PL/SQL
- Import Data Using Oracle Data Pump
- Load Data from Local Files Using SQL*Loader
- Querying and Validating External Data
- Load and Manage Using Data Sync

ORACLE

Loading Data with Autonomous Database

ORACLE AUTONOMOUS DATABASE

ORACLE

About Data Loading

- ADW uses Oracle Database tools and Oracle or other third-party data integration tools for data loading.
- Traditionally, transaction processing systems ingest data through routine transactions or DML operations.
- Load data from files:
  - Local to your client computer
  - Files stored in a cloud-based object store
    - For data loading from files in the cloud - PL/SQL package, DBMS_CLOUD
- For fastest data loading:
  - Upload the source files to a cloud-based object store - Oracle Cloud Infrastructure Object Storage
  - Then load the data into your Autonomous Data Warehouse Cloud

You can load data into the autonomous database using Oracle Database tools and Oracle and third-party data integration tools. You can load data from either of the following:

- Files local to your client computer
- Files stored in a cloud-based object store

For the fastest data loading experience, Oracle recommends uploading the source files to a cloud-based object store, such as Oracle Cloud Infrastructure Object Storage, before loading the data into your autonomous database.

To load data from files in the cloud into your autonomous database, use the new PL/SQL `DBMS_CLOUD` package. The `DBMS_CLOUD` package supports loading data files from the following cloud sources: Oracle Cloud Infrastructure Object Storage, Oracle Cloud Infrastructure Object Storage Classic, and Amazon AWS S3.

Data Loading Options

Data loading via SQL*Net

- Suitable for small volumes of data
  - SQL*Loader from local file system
  - ETL scripts that use DML to insert/update data

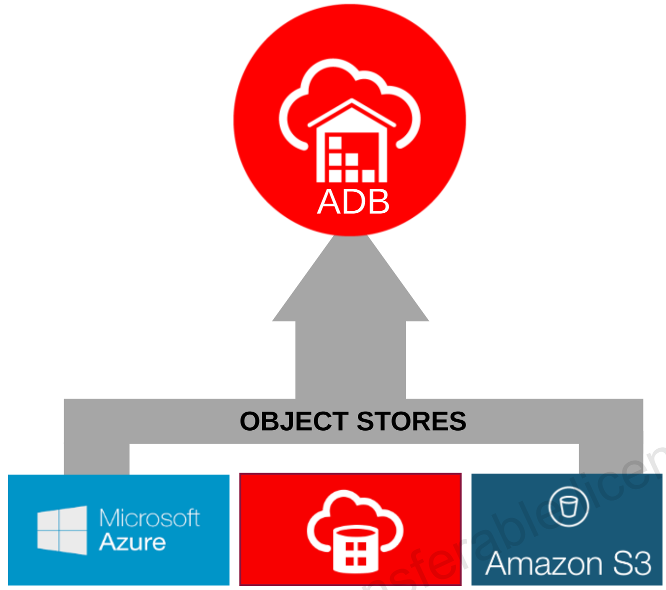
Data loading from Oracle Object Storage

- Preferred technique for large volumes of data
  - Additionally enables data sharing with other cloud services
- Stage data in Oracle Object Storage and then load into the database using new PL/SQL APIs

Loading Data from Object Stores

Source data from files on object stores for data loading:

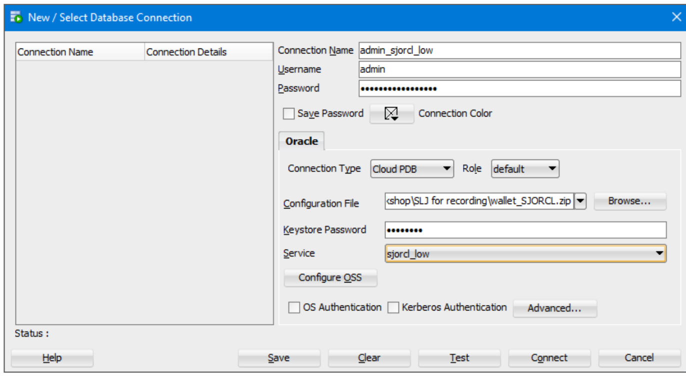
- OCI Object Storage, OCI Object Storage Classic, AWS S3, or Microsoft Azure
- Any supported ORACLE_LOADER file format
- Roadmap: any Hadoop file format



ORACLE

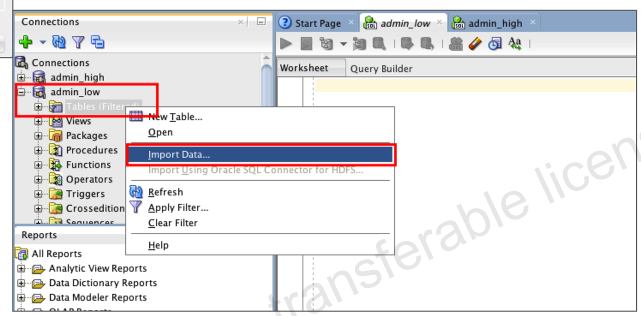
Loading Data from Local File Using SQL Developer

Connect to ADB to Load a Local File



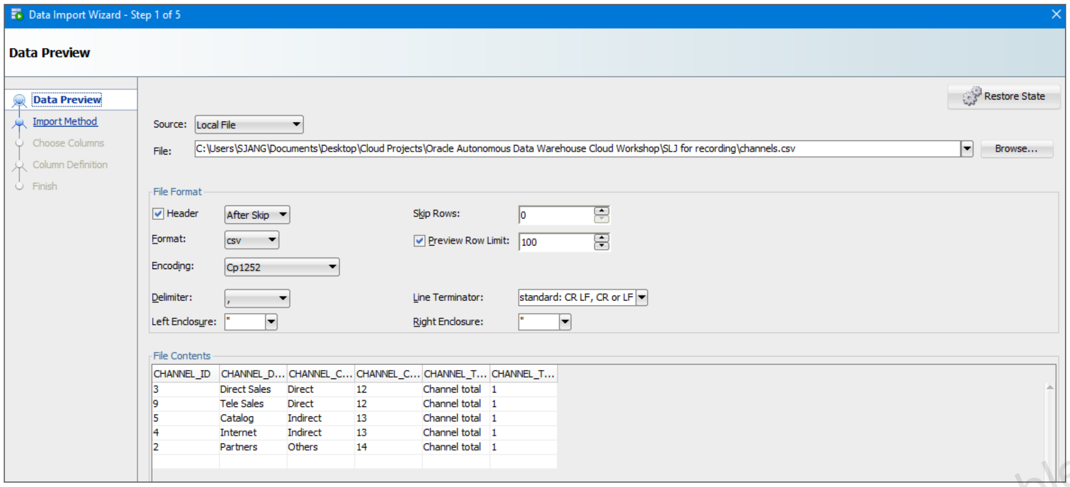
In SQL Developer

- Connect to the database
- Expand the connection
- Right-click “**Tables**”
- Click “**Import Data**”



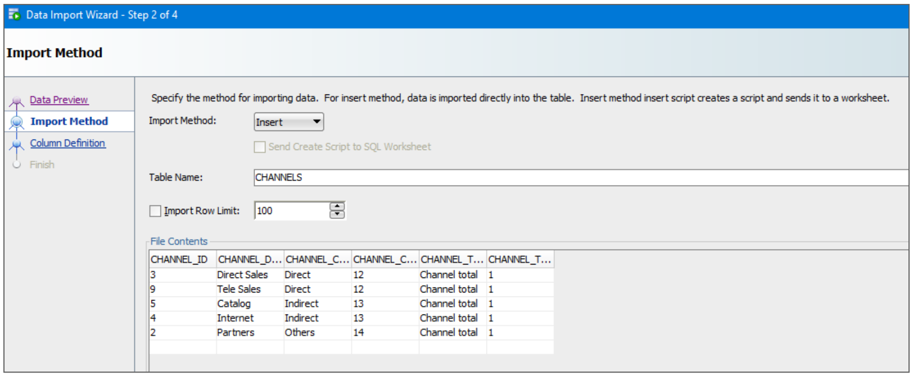
ORACLE

SQL Developer: Data Import Wizard



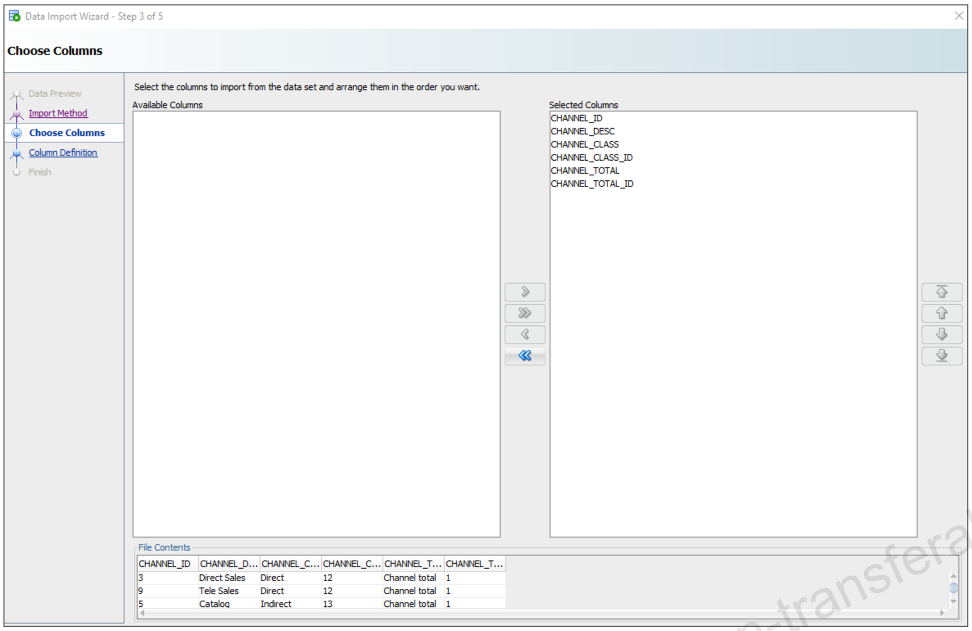
ORACLE

SQL Developer: Data Import Wizard



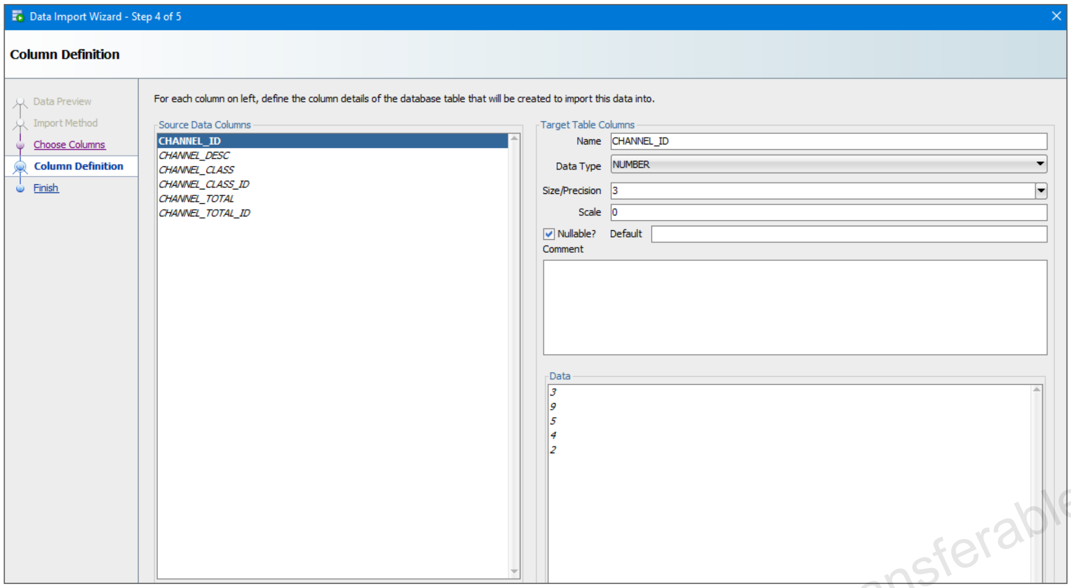
ORACLE

SQL Developer: Data Import Wizard

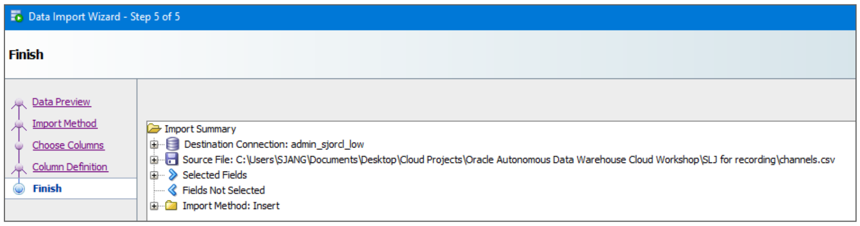


ORACLE

SQL Developer: Data Import Wizard



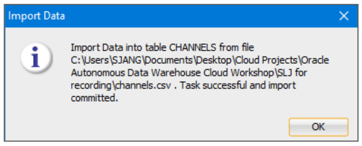
SQL Developer: Data Import Wizard



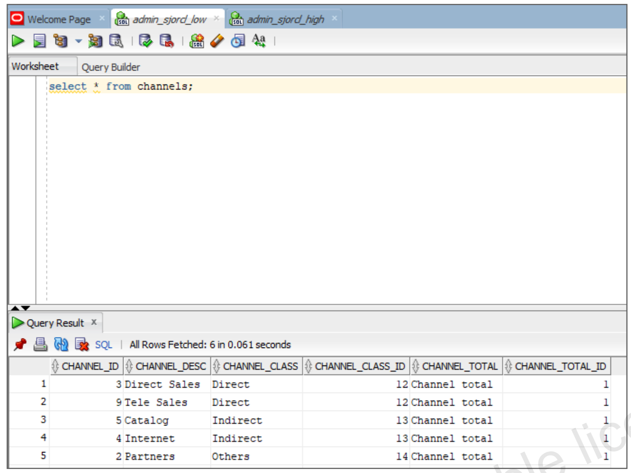
Review what you have selected and input in the Data import Wizard before starting load.

ORACLE

Query Data After Data Load



- When the data load finishes you will see a message saying the import was completed.
- Source file is now loaded into a table in the ADB.
- Run a query on the table to see data.



ORACLE

Loading Data from Cloud Object Storage Using SQL Developer

Loading Data from Files in the Cloud

Getting Started with Loading Data from Files in the Cloud

- To load data from files in the cloud:
  - Store Cloud Object Storage credentials in the ADB using DBMS_CLOUD.CREATE_CREDENTIAL.

```
DBMS_CLOUD.CREATE_CREDENTIAL (
credential_name IN VARCHAR2
username IN VARCHAR2
password IN VARCHAR2);
```

  - Load data into an existing table using the DBMS_CLOUD.COPY_DATA procedure.

```
DBMS_CLOUD.COPY_DATA (
table_name IN VARCHAR2,
credential_name IN VARCHAR2
file_uri_list IN CLOB
schema_name IN VARCHAR2
DEFAULT field_list IN CLOB DEFAULT
format IN CLOB DEFAULT);
```


Getting Started Loading Data from Files in the Cloud

For loading data from files in the Cloud, you need to first store your Object Storage credentials in your autonomous database and then use the `DBMS_CLOUD.COPY_DATA` procedure to load data.

The source file in this example, `channels.txt`, has the following data:

S,Direct Sales,Direct

T,Tele Sales,Direct

C,Catalog,Indirect

I,Internet,Indirect

P,Partners,Others

You can store your object store credentials using the `DBMS_CLOUD.CREATE_CREDENTIAL` procedure.

For example:

```
SET DEFINE OFF
BEGIN
    DBMS_CLOUD.CREATE_CREDENTIAL(
        credential_name => 'DEF_CRED_NAME',
        username => 'atpc_user@oracle.com',
        password => 'password'
    );
END;
/
```

This operation stores the credentials in the database in an encrypted format. You can use any name for the credential name. Note that this step is required only once unless your object store credentials change. After you store the credentials, you can use the same credential name for all data loads.

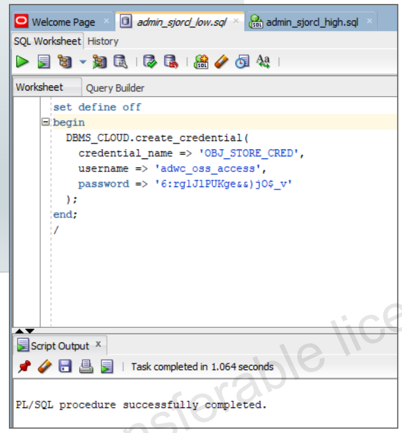
Load data into an existing table by using the `DBMS_CLOUD.COPY_DATA` procedure. For example:

```
CREATE TABLE CHANNELS (channel_id char(1), channel_desc varchar2(20),
channel_class varchar2(20) );


BEGIN
     DBMS_CLOUD.COPY_DATA(
             table_name =>'CHANNELS',
            credential_name =>'DEF_CRED_NAME',
            file_uri_list =>'https://swiftobjectstorage.us-phoenix-
1.oraclecloud.com/v1/atpc/atpc_user/channels.txt',
            format => json_object('delimiter' value ',')
     );
END;
/
```

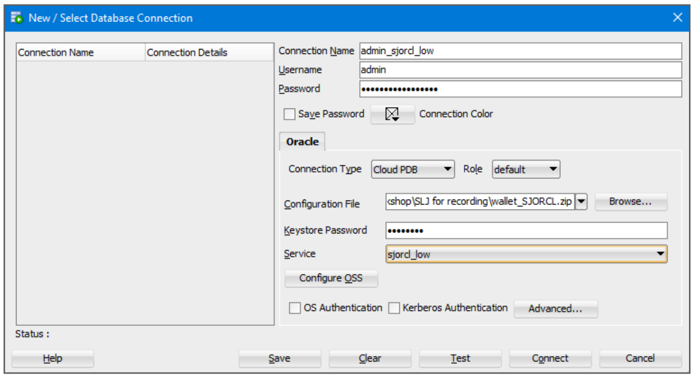
Authenticate Database User with the Object Store

- To access data in the Object Store
  - Enable database user to authenticate itself with the Object Store
  - Provide Object Store Swift password
- Create a private **CREDENTIAL** object for user
  - Stores information encrypted in ADB
  - Information is only usable for your user schema



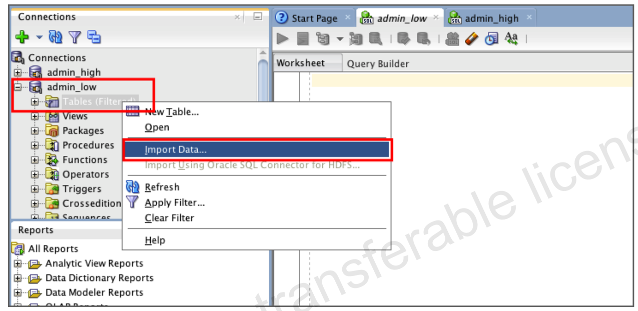
ORACLE

Connect to ADB to Load Data from the Cloud Object Store

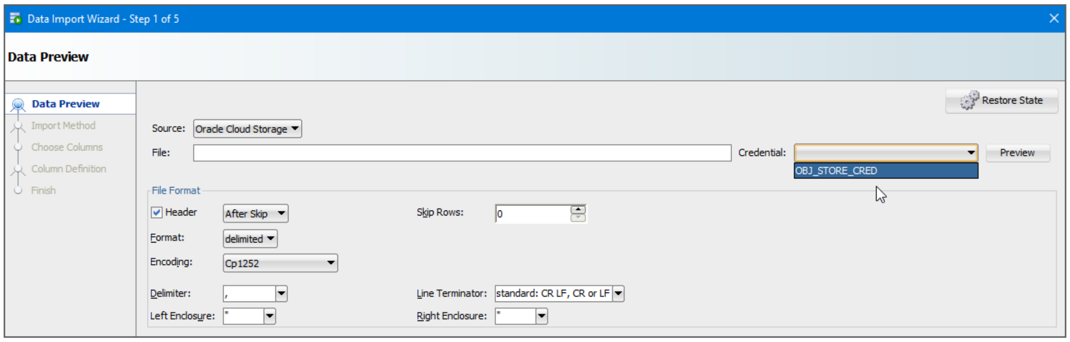


In SQL Developer:

- Connect to the database
- Expand the connection
- Right-click "**Tables**"
- Click "**Import Data**"



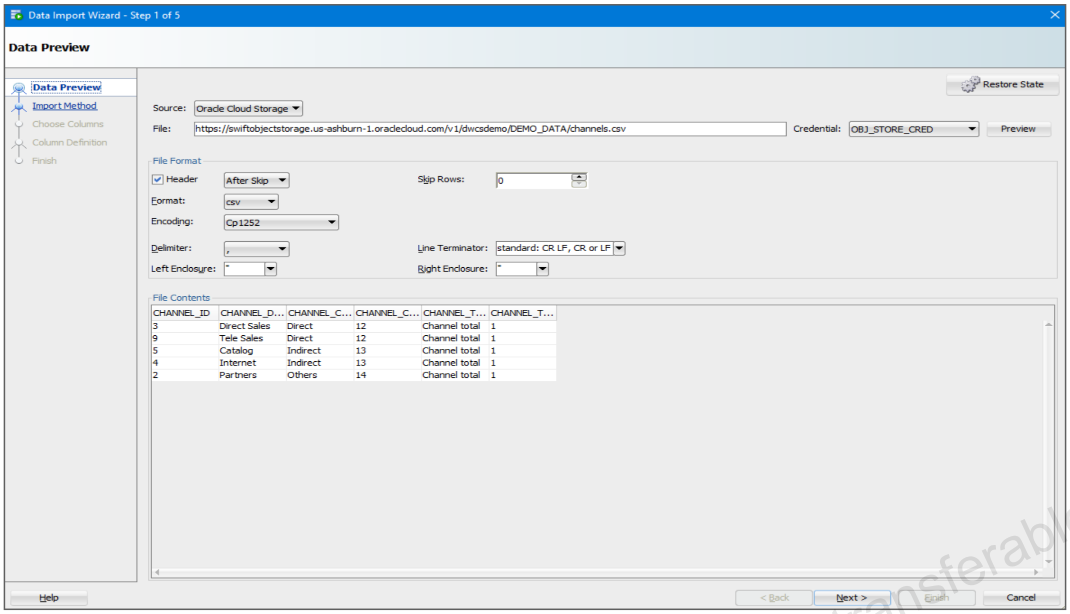
Load Data from the Object Store



- Select Oracle Cloud Storage as source for the data load.
- Enter the URL as the file to load.
- Select the credential previously created for authentication with the Object Store.

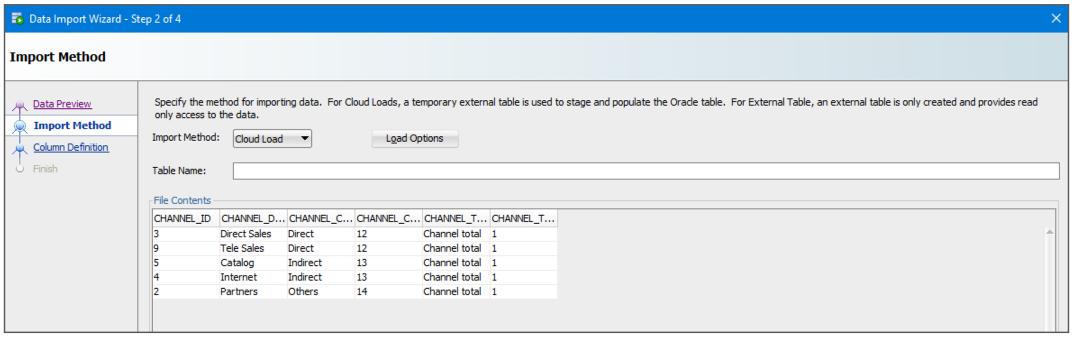
ORACLE

Preview the Data and Select the Appropriate File Formats



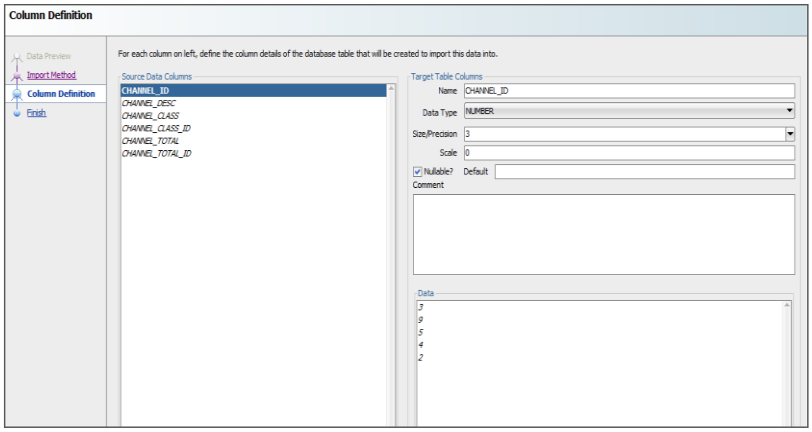
ORACLE

Create Table to Load Data



ORACLE

Table's Columns and Data Types

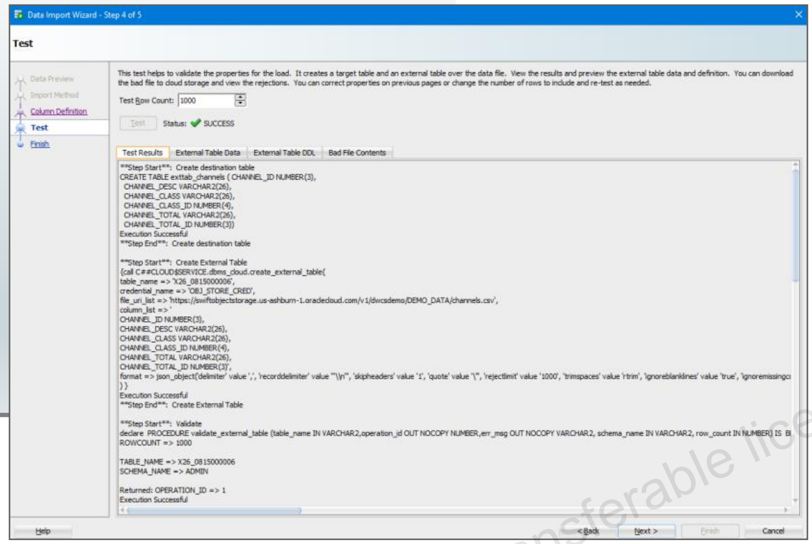


View sample data for each column.

ORACLE

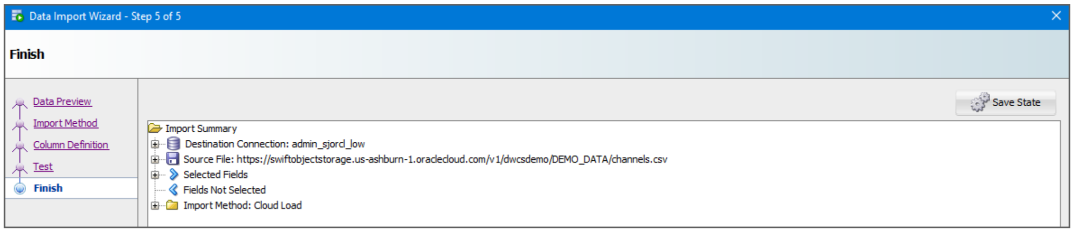
Read of the Source File

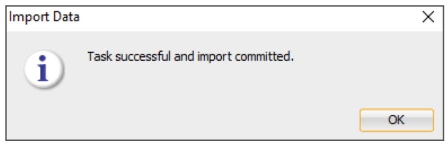
- Shows read of the source file:
  - Successful
  - Not Successful
    - see possible errors
- Must resolve error before starting actual data load



ORACLE®

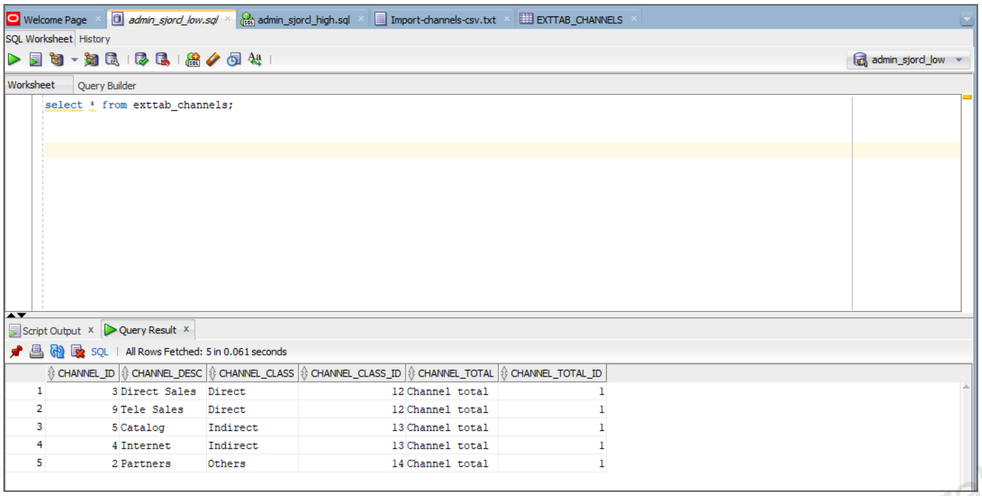
Summary of Selections in the Wizard





ORACLE

Source File Is Now Loaded into ADB. Query the Table to Check the Data



ORACLE

Loading Data from Files in the Cloud

Loading Data: Monitoring and Troubleshooting Loads

- All data load operations done using `DBMS_CLOUD` are logged in:
  - `dba_load_operations`: Shows all load operations
  - `user_load_operations`: Shows the load operations in your schema.
- Use the `SELECT` statement with the `WHERE` clause on the `TYPE` column with `COPY` on the previous table to check the details of load operations.
- The `LOGFILE_TABLE` column displays a table to check the log of a load operation.
- The `BADFILE_TABLE` column displays a table that can be queried to look at the rows that received errors during loading.

All data load operations done using the `DBMS_CLOUD` PL/SQL package are logged in the `dba_load_operations` and `user_load_operations` tables:

- `dba_load_operations`: Shows all load operations
- `user_load_operations`: Shows the load operations in your schema

Query these tables to see information about ongoing and completed data loads. For example:

```
SELECT table_name, owner_name, type, status, start_time, update_time,
logfile_table, badfile_table FROM user_load_operations WHERE type = 'COPY';
```

Using the `SELECT` statement with a `WHERE` clause predicate on the `TYPE` column shows load operations with the type `COPY`.

The `LOGFILE_TABLE` column shows the name of the table you can query to look at the log of a load operation.

The `BADFILE_TABLE` column shows the name of the table you can query to look at the rows that got errors during loading.

Depending on the errors shown in the log and the rows shown in the specified `BADFILE_TABLE` table, you can correct the error by specifying the correct format options in `DBMS_CLOUD.COPY_DATA`.

Loading Data from Cloud Object Storage Using PL/SQL

ORACLE AUTONOMOUS DATABASE

ORACLE

Loading Data from the Oracle Object Store

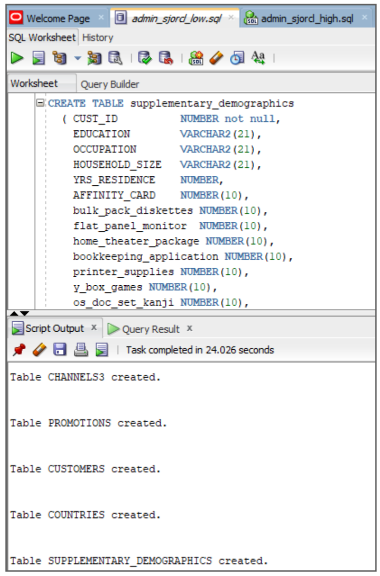
Load data directly into the target table without any intermediate steps.

Data format in the source file is easily specified as JSON.

```
begin
 dbms_cloud.copy_data(
    table_name =>'CHANNELS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>'https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/dwcsdemo/DEMO_DATA/chan_v3.dat',
    format => json_object('ignoremissingcolumns' value
'true',
                           'removequotes' value 'true')
 );
end;
/
```

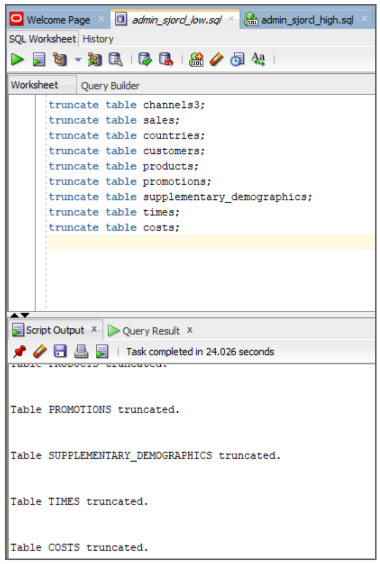
ORACLE

Create Tables for Data Load



ORACLE

Ensure No Data in Table(s)



ORACLE

Loading Data from Files in the Cloud

- PL/SQL package DBMS_CLOUD:
  - Load data from files in the cloud to your tables in the autonomous database.
  - Load from files in the following cloud services:
    - Oracle Cloud Infrastructure Object Storage, Oracle Cloud Infrastructure Object Storage Classic, Azure Blob Storage, and Amazon S3.
- For data loading from files in the cloud:
  - You first need to store Object Storage credentials in your autonomous database
  - Use the procedure DBMS_CLOUD.COPY_DATA to load data

ORACLE

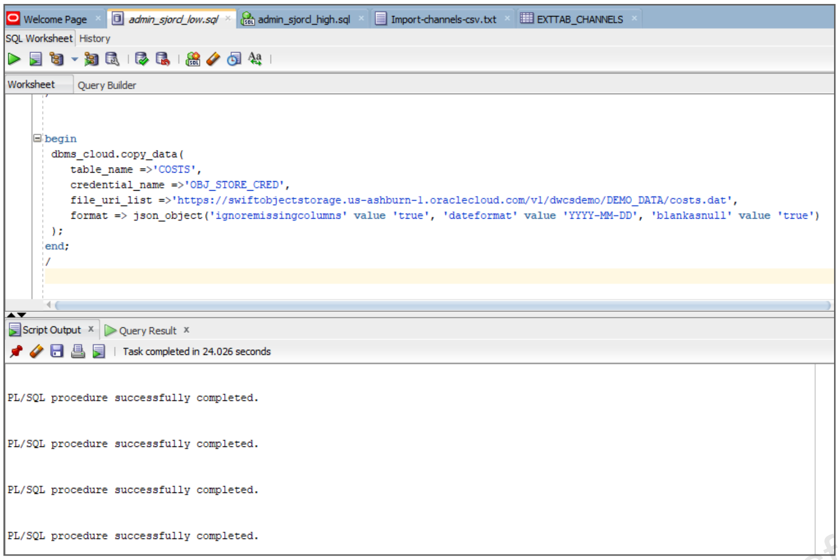
Procedure DBMS_CLOUD.COPY_DATA to Load Data

- table_name
  - Name of the target table on the autonomous database
  - The target table needs to be created before you run COPY_DATA.
- credential_name
  - Name of the credential to access the Cloud Object Storage
- file_uri_list
  - Source file URIs
- Format
  - Describe the format of the source files

```
begin
 dbms_cloud.copy_data(
    table_name =>'COSTS',
    credential_name
=>'OBJ_STORE_CRED',
    file_uri_list
=>'https://objectstorage.us-ashburn
  oraclecloud.com/v1/dwcsdemo/DEMO_DA
TA/costs.dat',
format =>
json_object('ignoremissingcolumns'
value 'true', 'dateformat' value
'YYYY-MM-DD', 'blankasnull' value
'true')
 );
end;
/
```

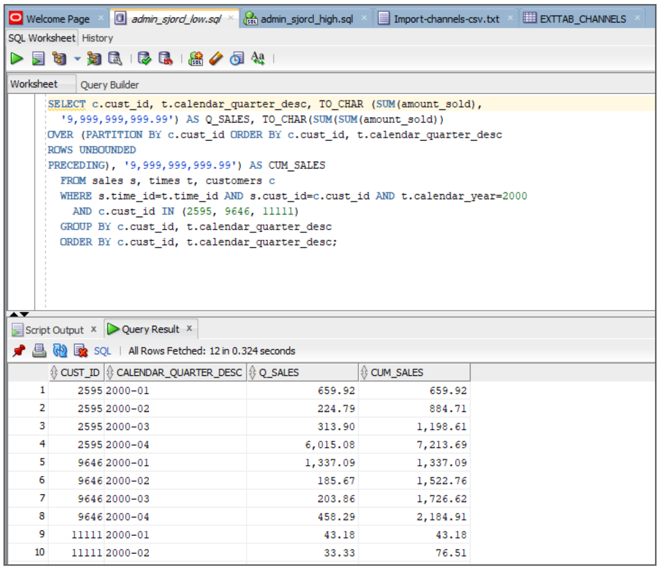
ORACLE

Run PL/SQL to Load Data from the Object Store



ORACLE

Run Query to See the Data from the Load




```sql
SELECT c.cust_id, t.calendar_quarter_desc, TO_CHAR (SUM(amount_sold),
  '9,999,999,999.99') AS Q_SALES, TO_CHAR(SUM(SUM(amount_sold))
OVER (PARTITION BY c.cust_id ORDER BY c.cust_id, t.calendar_quarter_desc
ROWS UNBOUNDED
PRECEDING), '9,999,999,999.99') AS CUM_SALES
  FROM sales s, times t, customers c
  WHERE s.time_id=t.time_id AND s.cust_id=c.cust_id AND t.calendar_year=2000
    AND c.cust_id IN (2595, 9646, 11111)
  GROUP BY c.cust_id, t.calendar_quarter_desc
  ORDER BY c.cust_id, t.calendar_quarter_desc;
```

	CUST_ID	CALENDAR_QUARTER_DESC	Q_SALES	CUM_SALES
1	2595	2000-01	659.92	659.92
2	2595	2000-02	224.79	884.71
3	2595	2000-03	313.90	1,198.61
4	2595	2000-04	6,015.08	7,213.69
5	9646	2000-01	1,337.09	1,337.09
6	9646	2000-02	185.67	1,522.76
7	9646	2000-03	203.86	1,726.62
8	9646	2000-04	458.29	2,184.91
9	11111	2000-01	43.18	43.18
10	11111	2000-02	33.33	76.51

ORACLE

Troubleshooting Loads

Load operations logged for troubleshooting and historical load tracking

- New table user/dba_load_operations

Log and bad files accessible as tables

```
select table_name,status,rows_loaded,logfile_table,badfile_table
from user_load_operations;

TABLE_NAME        STATUS    ROWS_LOADED LOGFILE_TABLE   BADFILE_TABLE
-------------------- --------- ----------- -------------------- --------------------
CHANNELS        FAILED       COPY$1_LOG     COPY$1_BAD
CHANNELS        COMPLETED     5 COPY$2_LOG     COPY$2_BAD
```

ORACLE

Loading Data – Monitoring and Troubleshooting Loads

- All data load operations done using the PL/SQL package DBMS_CLOUD are logged in the tables:
  - dba_load_operations: shows all load operations
  - user_load_operations: shows the load operations in your schema
- Query tables to see information about ongoing and completed data loads.
  - SELECT table_name, owner_name, type, status, start_time, update_time, logfile_table, badfile_table FROM user_load_operations WHERE type = 'COPY';
- TYPE column - load operations with the type COPY
- LOGFILE_TABLE column - name of the table you can query to look at the log of a load operation
  - Select * from COPY$5_LOG;
- BADFILE_TABLE - name of the table to see rows that got errors and were rejected for the load operation
  - Select * from COPY$5_BAD;

Importing Data Using Oracle Data Pump

ORACLE
AUTONOMOUS
DATABASE

ORACLE

Importing Data Using Oracle Data Pump

- Oracle Data Pump offers very fast bulk data and metadata movement between Oracle databases and autonomous databases.
- Data Pump Import lets you import data from Data Pump files residing on:
  - Oracle Cloud Infrastructure Object Storage
  - Oracle Cloud Infrastructure Object Storage Classic
  - AWS S3

ORACLE

Oracle Data Pump offers very fast bulk data and metadata movement between Oracle databases and autonomous databases.

Data Pump Import lets you import data from Data Pump files residing on the Oracle Cloud Infrastructure Object Storage, Oracle Cloud Infrastructure Object Storage Classic, and AWS S3. You can save your data to your Cloud Object Store and use Oracle Data Pump to load data to the autonomous database.

Importing Data Using Oracle Data Pump

Exporting an Existing Oracle Database to Import into the autonomous database:

- Oracle Data Pump Export is used to export existing Oracle Database schemas.
- Oracle Data Pump Import is used to migrate schemas to ADB.
- Oracle recommends using the schema mode for migrating to ADB
- For faster migration, use parallelism and set the `parallel` parameter to at least a number of CPUs in ADB.
- Use the `exclude` and `data_options` parameters to ensure that object types not available in ADB are not exported.

Exporting Your Existing Oracle Database to Import into Autonomous Database: Oracle Data Pump Export provides several export modes. Oracle recommends using the schema mode for migrating to autonomous database. You can list the schemas you want to export by using the *schemas* parameter.

For faster migration, export your schemas into multiple Data Pump files and use parallelism. You can specify the dump file name format you want to use with the *dumpfile* parameter. Set the *parallel* parameter to at least the number of CPUs you have in your autonomous database.

The *exclude* and *data_options* parameters ensure that the object types not available in the autonomous database are not exported and table partitions are grouped together so that they can be imported faster during the import to the autonomous database.

The following example exports the SH schema from a source Oracle Database for migration to an autonomous database with 16 CPUs:

```
expdp sh/sh@orcl \ exclude=cluster, db_link \ parallel=16 \ schemas=sh \
dumpfile=export%u.dmp
```

Importing Data Using Oracle Data Pump

Importing Data Using Oracle Data Pump (Latest Version for Linux x86-64):

- Oracle recommends using the latest Oracle Data Pump version for importing data from Data Pump files into your ADB.
- In the latest version of Oracle Data Pump, the `credential` argument authenticates Data Pump to the Cloud Object Storage service you are using for your source files.
- The `dumpfile` argument is a comma delimited list of URLs for your Data Pump files.
- For the best import performance, use the HIGH database service for your import connection and set the `PARALLEL` parameter to the number of CPUs in your ADB.

Importing Data Using Oracle Data Pump (Latest Version for Linux x86-64): Oracle recommends using the latest Oracle Data Pump version for importing data from Data Pump files into your autonomous database as it contains enhancements and fixes for a better experience.

You can download the latest version from the autonomous database service console by going to the Administration tab and clicking "Download Oracle Instant Client". This downloads the latest Oracle Instant Client for the Linux x86-64 platform. For instructions on how to install Oracle Instant Client after downloading it from the service console, use the following URL:

https://www.oracle.com/pls/topic/lookup?ctx=en/cloud/paas/autonomous-data-warehouse-cloud&id=instant-client-download-linux.

In the latest version of Oracle Data Pump, the *credential* argument authenticates Data Pump to the Cloud Object Storage service you are using for your source files. The *dumpfile* argument is a comma-delimited list of URLs for your Data Pump files.

Importing with Oracle Data Pump and Setting the Credential **Parameter**

1. Store your Cloud Object Storage credential using `DBMS_CLOUD.CREATE_CREDENTIAL`. For example:

```
BEGIN
      DBMS_CLOUD.CREATE_CREDENTIAL(
            credential_name => 'DEF_CRED_NAME',
            username => 'atpc_user@oracle.com',
            password => 'password'
      );
END;
/
```

2. Run Data Pump Import with the *dumpfile* parameter set to the list of file URLs on your Cloud Object Storage and the *credential* parameter set to the name of the credential you created in the previous step. For example:

```
impdp admin/password@ATPC1_high directory=data_pump_dir credential=def_cred_name
dumpfile= https://swiftobjectstorage.us-
phoenix1.oraclecloud.com/v1/atpc/atpc_user/sh.dmp parallel=16
transform=segment_attributes:n transform=dwcs_cvt_iots:y
transform=constraint_use_default_index:y exclude=cluster, db_link
```

For the best import performance, use the `HIGH` database service for your import connection and set the `PARALLEL` parameter to the number of CPUs in your Autonomous Transaction Processing as shown in the example.

This example shows the recommended parameters for importing into your Autonomous Transaction Processing. These parameters ensure you do not see errors related to the object types not allowed in Autonomous Transaction Processing.

Importing Data Using Oracle Data Pump

Importing Data Using Oracle Data Pump (Versions 12.2.0.1 and Earlier):

- Data Pump Import versions not downloaded from the ADB service console do not have the credential parameter.
- To use an older version of import, you first need to define a default credential property for ADB.
- The `default_credential` keyword needs to be used in the `dumpfile` parameter.
- For the best import performance, use the HIGH database service for your import connection and set the `PARALLEL` parameter to the number of CPUs in your ADB.

ORACLE

Importing Data Using Oracle Data Pump (Versions 12.2.0.1 and earlier): You can import data from Data Pump files into your autonomous database using existing Data Pump clients, older versions, which you did not download from the service console by setting the `default_credential` parameter.

Data Pump Import versions not downloaded from the autonomous database service console do not have the *credential* parameter. If you are using an older version of Data Pump Import, you need to define a default credential property for Autonomous Transaction Processing and use the `default_credential` keyword in the `dumpfile` parameter.

Importing with Older Oracle Data Pump Versions and Setting `default_credential`

1. Store your Cloud Object Storage credential using `DBMS_CLOUD.CREATE_CREDENTIAL`. For example:

```
BEGIN
      DBMS_CLOUD.CREATE_CREDENTIAL(
      credential_name => 'DEF_CRED_NAME',
      username => 'atpc_user@oracle.com',
      password => 'password'
      );
END;
/
```

2. Set the credential as the default credential for your Autonomous Transaction Processing, as the `ADMIN` user. For example:

```
alter database property set default_credential = 'ADMIN.DEF_CRED_NAME'
```

3. Run Data Pump Import with the `dumpfile` parameter set to the list of file URLs on your Cloud Object Storage and set the `default_credential` keyword. For example:

```
     impdp admin/password@ATPC1_high directory=data_pump_dir
dumpfile=default_credential:https://swiftobjectstorage.us-phoenix-
1.oraclecloud.com/v1/atpc/atpc_user/sh.dmp parallel=16
transform=segment_attributes:n  exclude=cluster, db_link
```

For the best import performance, use the `HIGH` database service for your import connection and set the **PARALLEL** parameter to the number of CPUs in your Autonomous Transaction Processing as shown in the example.

This example shows the recommended parameters for importing into your Autonomous Transaction Processing. These parameters ensure that you do not see errors related to the object types not allowed in Autonomous Transaction Processing.

Importing Data Using Oracle Data Pump

Accessing the Log Files for Data Pump Import:

- The `DATA_PUMP_DIR` directory stores the log files for the Data Pump Import operations.
- The `DATA_PUMP_DIR` directory can only be specified for the Data Pump `directory` parameter.
- The log file can be accessed after moving it to Oracle Cloud Storage using `DBMS_CLOUD.PUT_OBJECT`.

ORACLE

Accessing the Log Files for Data Pump Import: The log files for Data Pump Import operations are stored in the `DATA_PUMP_DIR` directory; this is the only directory you can specify for the Data Pump `directory` parameter.

To access the log file, you need to move the log file to your Cloud Object Storage by using the `DBMS_CLOUD.PUT_OBJECT` procedure. For example, the following PL/SQL block moves the `import.log` file to your Cloud Object Storage:

```
BEGIN
     DBMS_CLOUD.PUT_OBJECT(
            credential_name => 'DEF_CRED_NAME',
            object_uri => 'https://swiftobjectstorage.us-phoenix-
1.oraclecloud.com/v1/atpc/atpc_user/import.log',
            directory_name => 'DATA_PUMP_DIR',
            file_name => 'import.log');
END;
/
```

Loading Data from Local Files Using SQL*Loader

ORACLE AUTONOMOUS DATABASE

ORACLE

Loading Data from Local Files Using SQL*Loader

Accessing the Log Files for Data Pump Import:

- It is used to load data from local files in your client machine into ADB.
- It is suitable for loading small amounts of data.
- To load multiple files at the same time, invoke a separate session for each file.
- Oracle recommends using the following parameters for the best load performance:
  - `readsize=100M`
  - `bindsize=100M`
  - `direct=N`

ORACLE

Loading Data from Local Files Using SQL*Loader: You can use Oracle SQL*Loader to load data from local files in your client machine into the autonomous database.

Using SQL*Loader may be suitable for loading small amounts of data, because the load performance depends on the network bandwidth between your client and the autonomous database. For large amounts of data, Oracle recommends loading data from the Cloud Object Storage. (For information about loading from Cloud Object Store, see https://docs.oracle.com/en/cloud/paas/atp-cloud/atpug/load-data.html#GUID-07900054-CB65-490A-AF3C-39EF45505802.)

Oracle recommends using the following SQL*Loader parameters for best load performance:

- `readsize=100M`
- `bindsize=100M`
- `direct=N`

The autonomous database gathers optimizer statistics for your tables during bulk-load operations if you use the recommended parameters. If you do not use the recommended parameters, then you need to gather optimizer statistics manually.

Querying External Data with Autonomous Database

ORACLE AUTONOMOUS DATABASE

ORACLE

Getting Started with Querying External Data

- To query data in files, first store Object Storage credentials in ADB and then create an external table.

```
DBMS_CLOUD.CREATE_CREDENTIAL (
      credential_name IN VARCHAR2
      username IN VARCHAR2
      password IN VARCHAR2);
```

- Use `DBMS_CLOUD.CREATE_EXTERNAL_TABLE` to create an external table.

```
DBMS_CLOUD.CREATE_EXTERNAL_TABLE (
      table_name IN VARCHAR2,
      credential_name IN VARCHAR2
      file_uri_list IN CLOB
      column_list IN CLOB
      format IN CLOB DEFAULT);
```

- Run the select query on the external table to access the data.

ORACLE

Validating External Data

- The `DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE` procedure is used to validate an external table.
- The validate operation, by default, scans all the rows in the source files and stops when a row is rejected.
- To validate only a subset of the rows, use the `rowcount` parameter.
- Set the `stop_on_error` parameter to `FALSE` so that the validate operation does not stop when a row is rejected.
- Query the `dba_load_operations` and `user_load_operations` tables to see the results of the validate operation.
- Use the `VALIDATE` value for the `TYPE` column in the `WHERE` clause for the query.

Load and Manage Data Using Data Sync

ORACLE AUTONOMOUS DATABASE

ORACLE

Data Sync Overview

- Data Sync can be used to upload and manage data.
- Data can be loaded from files like .CSV and .XLSX, tables, views, SQL statements OTBI, JDBC data sources, and Oracle Service Cloud.
- Data can be loaded to relational tables or data sets.

Data Sync: Data Sources

- Data Sync can be used to:
  - Load data sources other than Oracle
  - Load a combination of data sources, such as CSV, XLSX, and Oracle relational files
  - Perform incremental data loads or rolling deletes
  - Perform insert-only or append strategies
  - Merge data from multiple sources
  - Schedule data loads. You can replace, append, and update data in tables by scheduling data loads and using the Load Strategy option of this utility.

Use Data Sync when you want to:

- Load data sources other than Oracle
- Load a combination of data sources, such as CSV, XLSX, and Oracle relational files
- Perform incremental data loads or rolling deletes
- Perform insert-only or append strategies
- Merge data from multiple sources
- Transform your data (if you're using Database As A Service or an on-premises database that is configured using the "Oracle (Thin)")
- Schedule data loads. You can replace, append, and update data in tables by scheduling data loads and using the Load Strategy option of this utility.

Use Data Sync to Load Data from Different Databases and JDBC Data Sources

Databases
- Oracle
- DB2
- Microsoft SQL Server
- MySQL
- Teradata
- TimesTen

JDBC Data Sources
- Greenplum
- Hive
- Impala
- Informix
- MongoDB
- NetSuite
- PostgreSQL
- Redshift
- Salesforce
- Sybase

Summary

You should now be able to:

- Describe Data Loading
- Load Data from Files in the Cloud
- Load Data from Cloud Object Storage Using SQL Developer
- Load Data from Cloud Object Storage Using PL/SQL
- Import Data Using Oracle Data Pump
- Load Data from Local Files Using SQL*Loader
- Query and Validate External Data
- Load and Manage Data Using Data Sync

ORACLE

Practice 7: Overview

- Practice 7-1: Loading a Local File into ADB
- Practice 7-2: Uploading Data Files to Your Object Store
- Practice 7-3: Creating an Object Store Auth Token
- Practice 7-4: Creating Object Store Credentials in an ADB Schema
- Practice 7-5: Copying Data from an Object Store to ADB Tables

ORACLE

8

Managing Users in Autonomous Database

ORACLE

Objectives

After completing this lesson, you should be able to:

- Create and remove users in Autonomous Database
- Manage user privileges in Autonomous Database
- Manage the administrator account in Autonomous Database
- Create user accounts for Oracle Machine Learning

ORACLE

Creating and Removing Users in an Autonomous Database

Creating Users in Autonomous Database (ADB)

- To create users in a database, connect to the database as the ADMIN user and execute:

```
CREATE USER new_user IDENTIFIED BY password;
GRANT CREATE SESSION TO new_user;
```

- ADB requires strong passwords. The database checks the following requirements for passwords:
  - Must be between 12 and 30 characters and include at least one uppercase letter, one lowercase letter, and one numeric character
  - Cannot contain the username
  - Cannot be one of the last four passwords used
  - Cannot contain the double quotation (") mark
- The database account gets locked for 24 hours after 10 failed login attempts. To unlock:

```
ALTER USER username IDENTIFIED BY password ACCOUNT UNLOCK;
```

ORACLE

Removing Users in Autonomous Database

- To remove users from a database, connect to the database as the `ADMIN` user by using any SQL client tool and execute:

```
DROP USER user_name CASCADE;
```

- This removes *user_name* and the objects owned by that user.

ORACLE

Managing User Privileges in Autonomous Database

- Autonomous Databases with data warehouse workload come with a predefined database role, `DWROLE`.
- The `DWROLE` role includes all necessary privileges for data warehouse developer.
- To grant `DWROLE`, connect to the database as the `ADMIN` user by using any SQL client tool and execute:

```
GRANT DWROLE to user;
```

ORACLE

Autonomous Databases come with a predefined database role named `DWROLE`. This role provides the common privileges for a database developer or data scientist to performance real-time analytics.

To grant the `DWROLE` role to your developers, connect to the database as the `ADMIN` user by using any SQL client tool. For example, connect using Oracle SQL Developer.

As the `ADMIN` user, run the following SQL statement:

```
GRANT DWROLE TO user;
```

The privileges in `DWROLE` are:

```
CREATE ANALYTIC VIEW
CREATE ATTRIBUTE DIMENSION
ALTER SESSION
CREATE HIERARCHY
CREATE JOB
CREATE MINING MODEL
CREATE PROCEDURE
CREATE SEQUENCE
CREATE SESSION
CREATE SYNONYM
CREATE TABLE
CREATE TRIGGER
```

Managing the Administrator Account in Autonomous Database (ADB)

ORACLE AUTONOMOUS DATABASE

ORACLE

Changing the Administrator Password in ADB

To change the password for the **`ADMIN`** user:

1. Access the Autonomous Database **Service Console**.
2. Click the **Administration** tab.
3. Click **Set Administrator Password**.
4. In the dialog box, enter and confirm the new password.
5. Click **OK**.

ORACLE

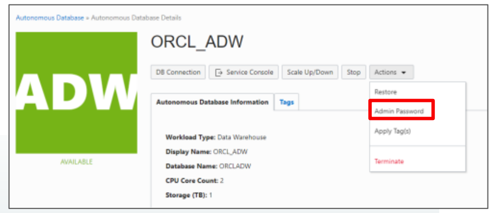
Changing the Administrator Password in Autonomous Database

From the service console, change the password for the `ADMIN` user by performing the following steps:

1. Access the Autonomous Database service console.
2. Click the **Administration** tab.
3. Click **Set Administrator Password**.
4. On the Set ADMIN Password screen, enter and confirm the new password.
5. Click **OK**.

The password for the default administrator account, `ADMIN`, has the same password complexity rules as any other user.

Unlocking the Administrator Account in ADB



To unlock the **ADMIN** account after 10 failed login attempts:

1. Sign in to the Oracle Cloud Account.
2. On the OCI page, click the navigation icon in the top left corner of the page, click ATP or ADW, and select a Compartment.
3. Select an ADB instance from the links in the **Name** column.
4. On the **Actions** tab, click **Admin Password**.
5. In the Admin Password dialog box, enter the password and confirm.
6. Click update.

ORACLE

Creating User Accounts for Oracle Machine Learning

ORACLE AUTONOMOUS DATABASE

ORACLE

Using Oracle Machine Learning with Autonomous Database

To use Oracle Machine Learning with Autonomous Database, you need to create users and access the system.

OML User Management

- **Admin** (a user with administrative privileges)
- Create and modify Oracle Machine Learning user accounts.

OML Application

- Application **users** access Oracle Machine Learning.
- Create, view, and share notebooks for data analytics, data visualization.
- Create, view, and share other Oracle Machine Learning task reports based on sophisticated analytics and data models.

ORACLE

OML Components

- OML User Administration Application
  - It is a web-based administrative UI for managing (list, create, update, delete) Oracle Machine Learning users.
  - Oracle Machine Learning users map to Autonomous Data Warehouse database users.
  - Access to Oracle Machine Learning User Management is limited to the administrator.
- OML Application
  - Web-based application for data scientists
  - Enables creation of workspaces, projects, and notebooks
  - Notebook execution in Zeppelin servers
- Zeppelin Server
  - Edit and run notebooks.
  - Secure (TCPS) JDBC connections to the Autonomous Database for data access.

Creating User Accounts in Oracle Machine Learning

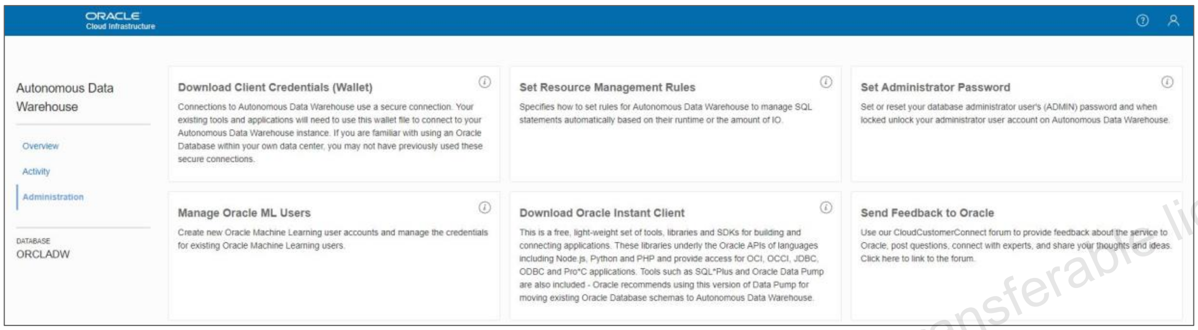
- The `ADMIN` user creates the user account and user credentials for Oracle Machine Learning (OML) in the User Management interface.
- You must have the administrator role to access the Oracle Machine Learning User Management interface.

To create a user account:

1. Log in to the Oracle Machine Learning User Management interface.
2. In the **Username** field, enter a username for the account. Using the username, the user will log in to an Oracle Machine Learning instance.
3. Enter the **First Name** and **Last Name** of the user in the respective fields.
4. In the **Password** field, enter a password for the user.
5. In the **Confirm Password** field, enter the same password that you entered in step 4. By doing so, you create the password for the user. The user can change the password when he or she logs in the first time.
6. In the **Email Address** field, enter the email ID of the user.
7. Select **Email Account details to user** to send the user account credentials to the user through email.
8. Click **Create**.

Creating OML Users

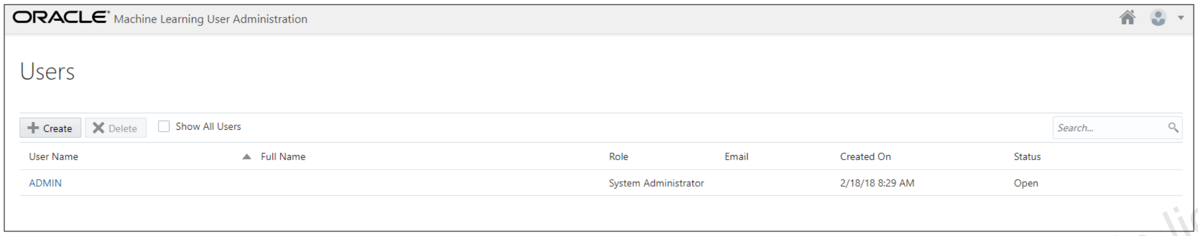
- Go to the Cloud Console and open the Instances screen.
- Find your database, click the **action menu**, and select **Service Console**.
- Log in to the service with your admin.
- Go to the **Administration tab** and click **Manage Oracle ML Users**.



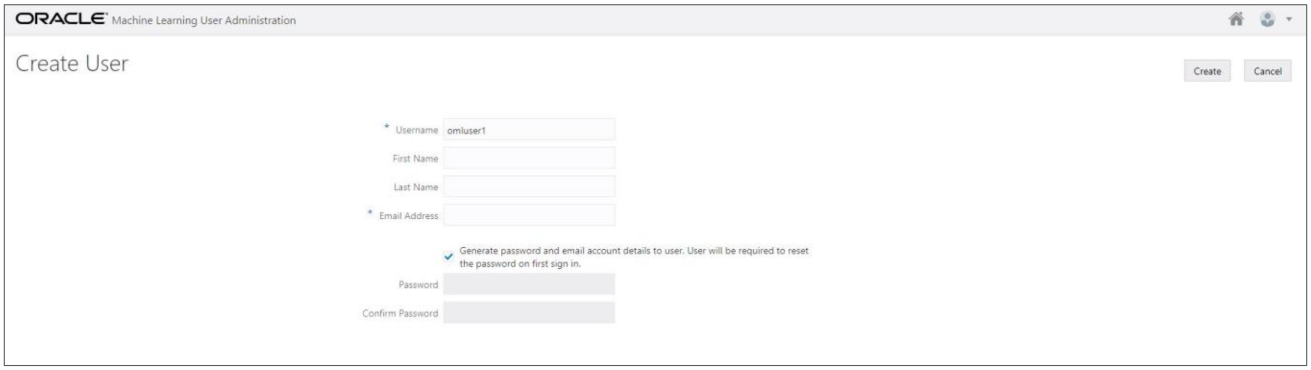
OML User Management

Access the User Management page as Admin.

- Create and manage Oracle Machine Learning user accounts.

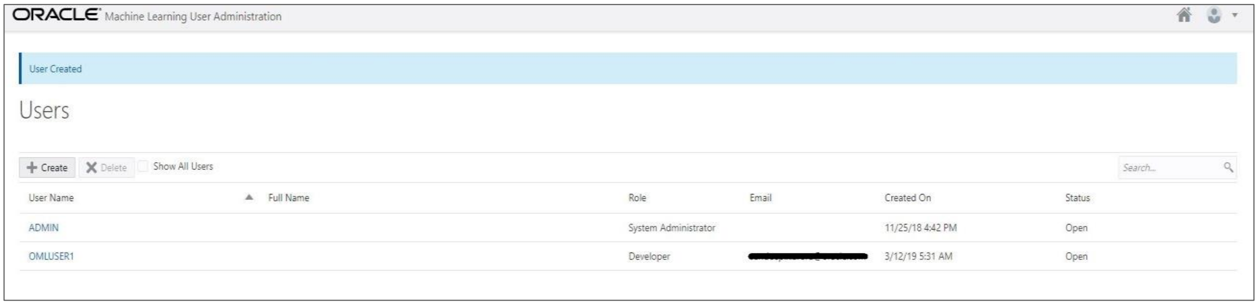


Creating Users for Oracle Machine Learning



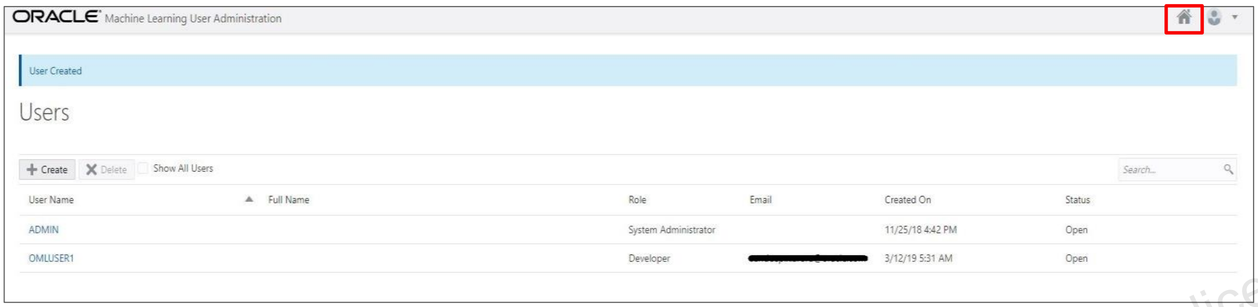
This will create a new user.

Creating Users for Oracle Machine Learning



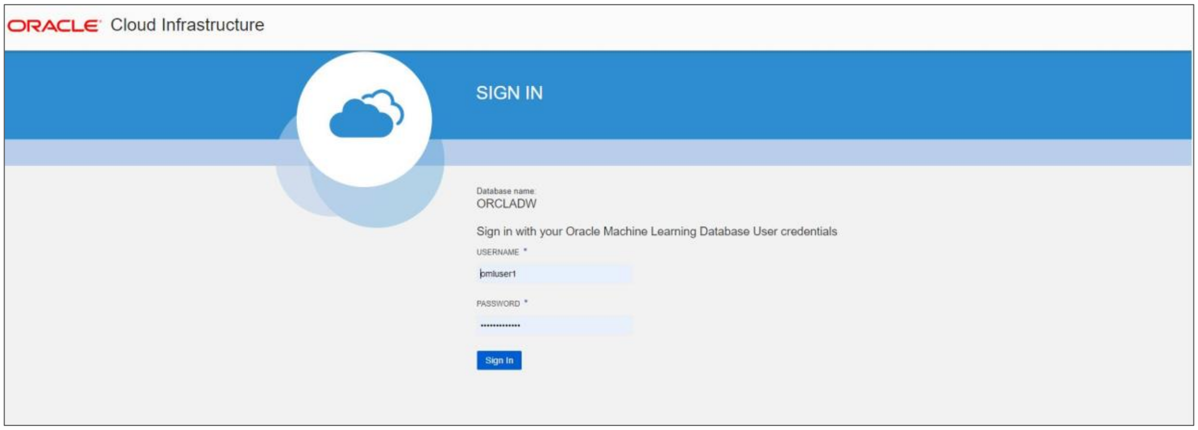
Logging In as a User to Oracle Machine Learning (OML)

Click the home icon to log in as a user to OML.



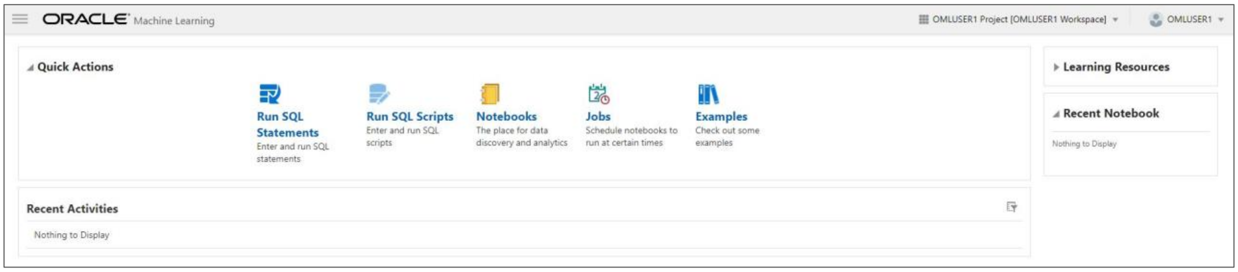
ORACLE

Logging In as a User to Oracle Machine Learning



ORACLE

The Oracle Machine Learning Login Screen



Summary

In this lesson, you should have learned how to:

- Create and remove users in Autonomous Database
- Manage user privileges in Autonomous Database
- Manage the administrator account in Autonomous Database
- Create user accounts for Oracle Machine Learning

ORACLE

Practice 8: Overview

- Practice 8-1: Managing User Accounts in Oracle Machine Learning

ORACLE

9

Using Oracle Machine Learning with Autonomous Database

ORACLE

Objectives

- Overview of Machine Learning
- Creating Dashboards, Reports, and Notebooks
- Accessing the Oracle Machine Learning User Management Page
- Working with Oracle Machine Learning for Data Access, Analysis, and Discovery
- Working with Analytics and Visualization

ORACLE

Using Oracle Machine Learning with Autonomous Database

Key features of Oracle Machine Learning:

- Allows collaboration among data scientists, developers, and business users
- Leverages the scalability and performance of Oracle Platform and its Cloud Services

Oracle Machine Learning:

- Provides a notebook-style application designed for advanced SQL users
- Provides interactive data analysis that lets you develop, document, share, and automate reports based on sophisticated analytics and data models

ORACLE

An Autonomous Database includes built-in support for data analytics using Oracle Machine Learning (OML), which is a browser-based interactive data analysis environment for data scientists.

Creating Users in Autonomous Database

ORACLE

Simplified User Creation

Simplified user creation via the new database role:

- No need to specify anything except the password.
- DWROLE includes all necessary privileges for a DW developer/user.

```
create user sh identified by sh;
grant dwrole to sh;
```

ORACLE

Creating Users for Oracle Machine Learning (OML)

Using Oracle Machine Learning with Autonomous Database

To use Oracle Machine Learning with Autonomous Database, you need to create users and access the system:

OML User Management:

- **Admin** (user with administrative privileges)
- Create and modify Oracle Machine Learning user accounts.

OML Application:

- Application **users** access Oracle Machine Learning.
- Create, view, and share notebooks for data analytics and data visualization
- Create, view, and share other Oracle Machine Learning tasks reports based on sophisticated analytics and data models.

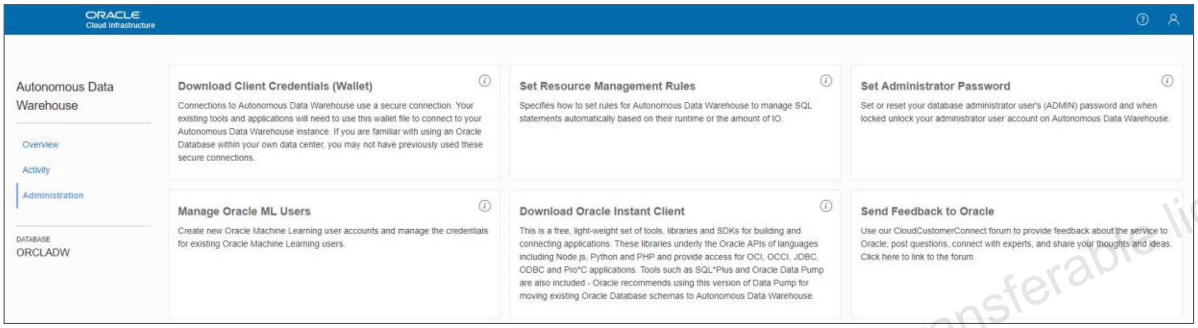
ORACLE

OML Components

- OML User Administration Application:
  - It is a web-based administrative UI for managing (list, create, update, delete) Oracle Machine Learning users.
  - Oracle Machine Learning users map to Autonomous Database users.
  - Access to Oracle Machine Learning User Management is limited to the administrator.
- OML Application:
  - Web-based application for data scientists
  - Enables creation of workspaces, projects, and notebooks
  - Notebook execution in Zeppelin servers
- Zeppelin Server:
  - Edit and run notebooks.
  - Secure (TCPS) JDBC connections to Autonomous Database for data access.

Creating OML Users

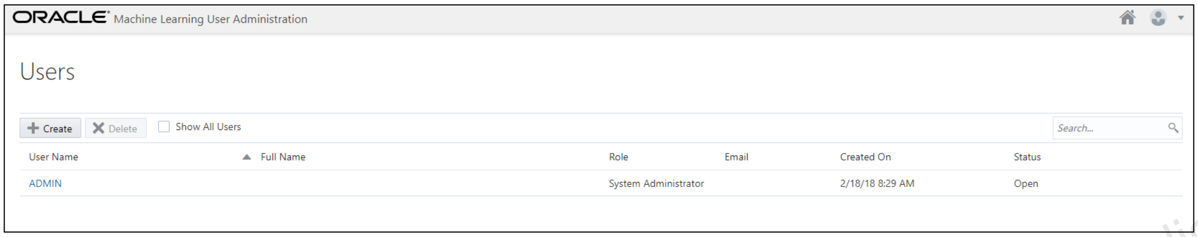
- Go to the Cloud Console and open the Instances screen.
- Find your database, click the **action menu**, and select **Service Console**.
- Log in to the service with your admin.
- Go to the **Administration tab** and click **Manage Oracle ML Users**.



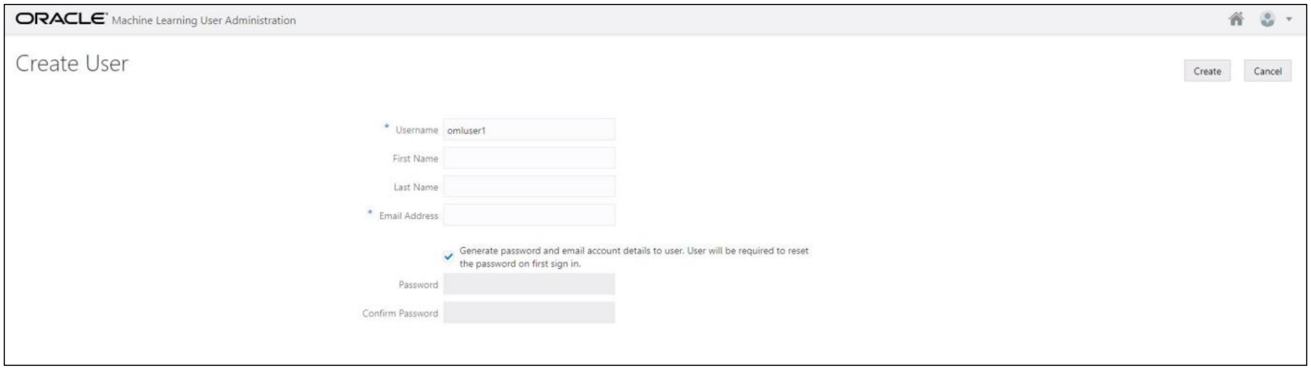
OML User Management

Access the User Management page as Admin:

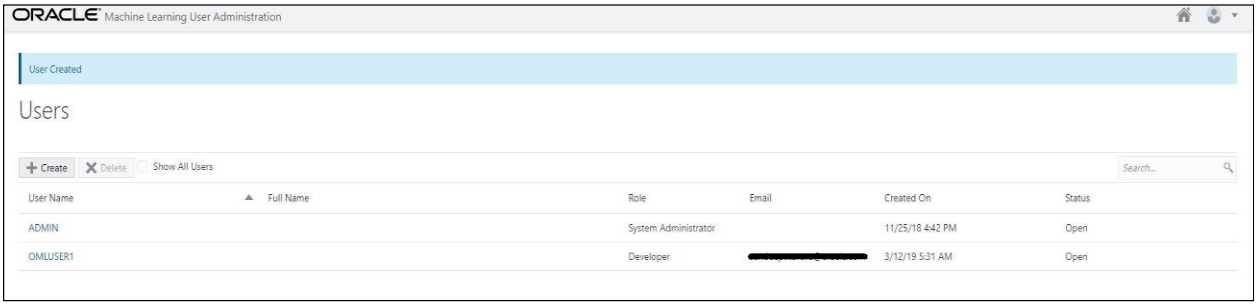
- Create and manage Oracle Machine Learning user accounts.



Create Users for Oracle Machine Learning



Create Users for Oracle Machine Learning



ORACLE

Accessing Oracle Machine Learning User Management

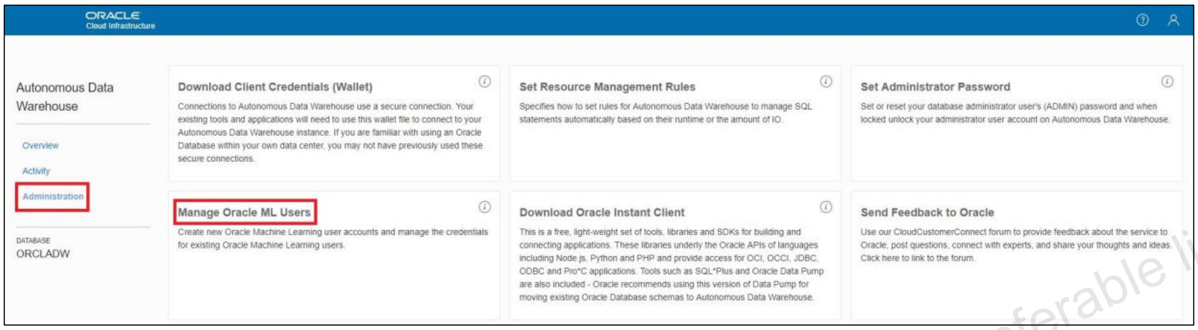
Accessing the Oracle Machine Learning User Management Page

To access the Oracle Machine Learning **Manage Oracle ML Users** page:

1. Sign in to your Oracle Cloud account at cloud.oracle.com
2. From the Oracle Cloud Infrastructure (OCI) page, click the top corner of the page, click ATP, and select a compartment.
3. Select an ATP instance and click Service Console on the details page.

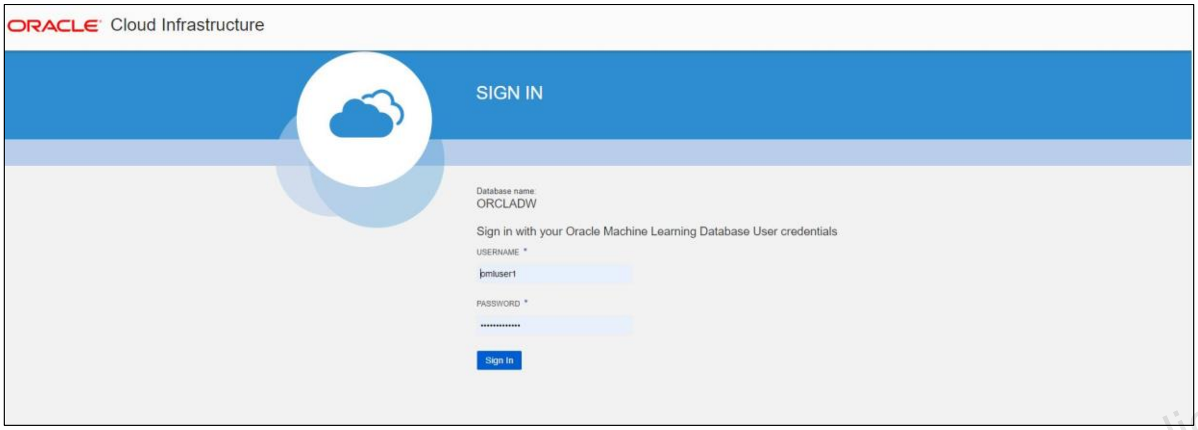
Accessing the Oracle Machine Learning User Management Page

4. At the prompt, enter ADMIN for the username and enter the password for the ADMIN user.
5. Click Administration.
6. On the Administration page, click Manage Oracle ML Users.



ORACLE

Log In as User to Oracle Machine Learning

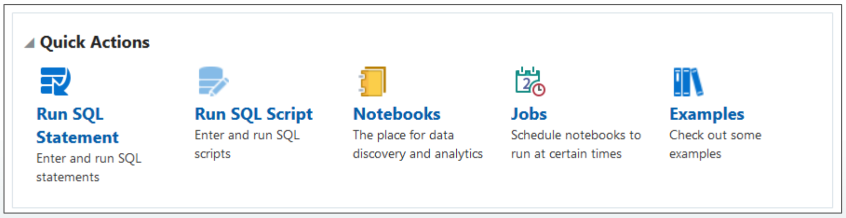


Working with Oracle Machine Learning for Data Access, Analysis, and Discovery

ORACLE AUTONOMOUS DATABASE

ORACLE

Working with OML for Data Access, Analysis, and Discovery



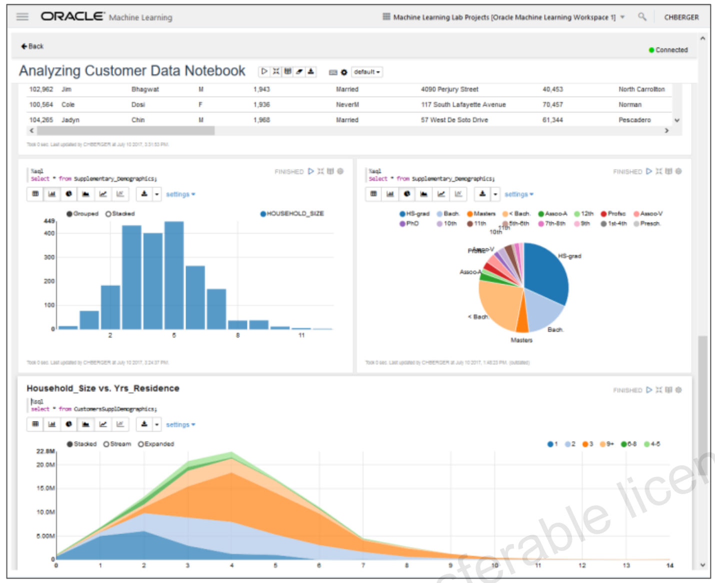
- Oracle Machine Learning allows you to access data in ADB and build notebooks with the following:
  - Data Ingestion and Selection
  - Data Viewing and Discovery
  - Data Graphing, Visualization, and Collaboration
  - Data Analysis
- Oracle Machine Learning allows you to create SQL statements and scripts that access data in ADB.

Built-in SQL Worksheet and Notebook

Quickly start running queries with built-in web-based notebooks.

- No need to install a client query tool
- Initially supports SQL and PL/SQL
- More languages in the roadmap

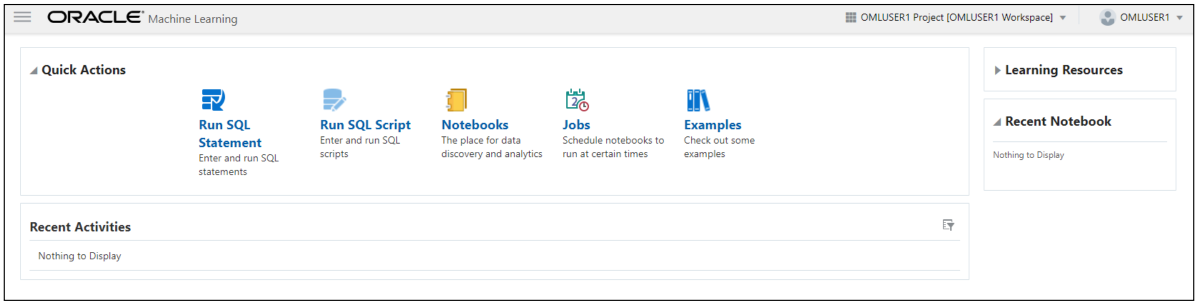
Based on Apache Zeppelin



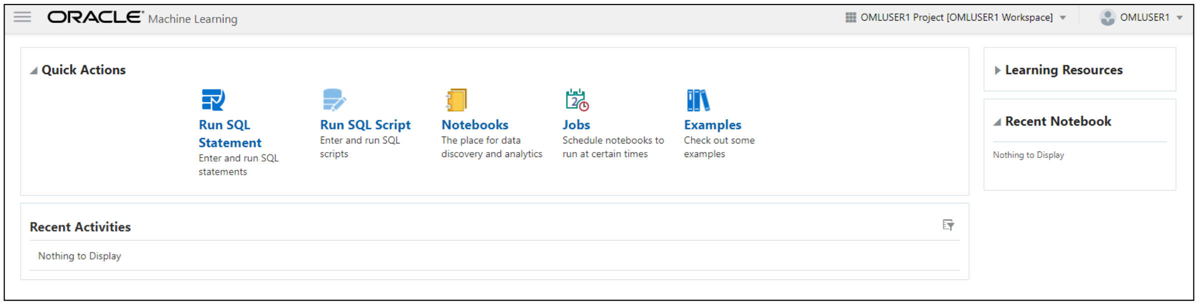
ORACLE

About Creating Dashboards, Reports, and Notebooks

OML Home Page – “**Quick Actions**” Panel



Running a SQL Statement

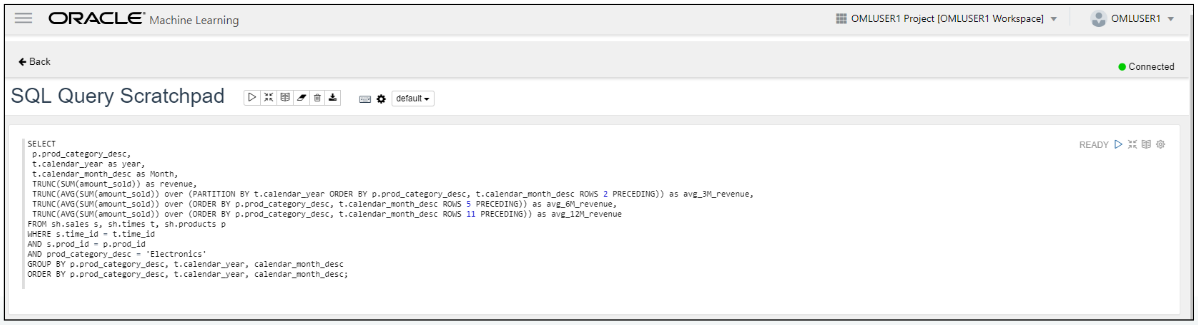


All your work is automatically saved.

There is no “**Save**” button when you are writing scripts and/or queries.

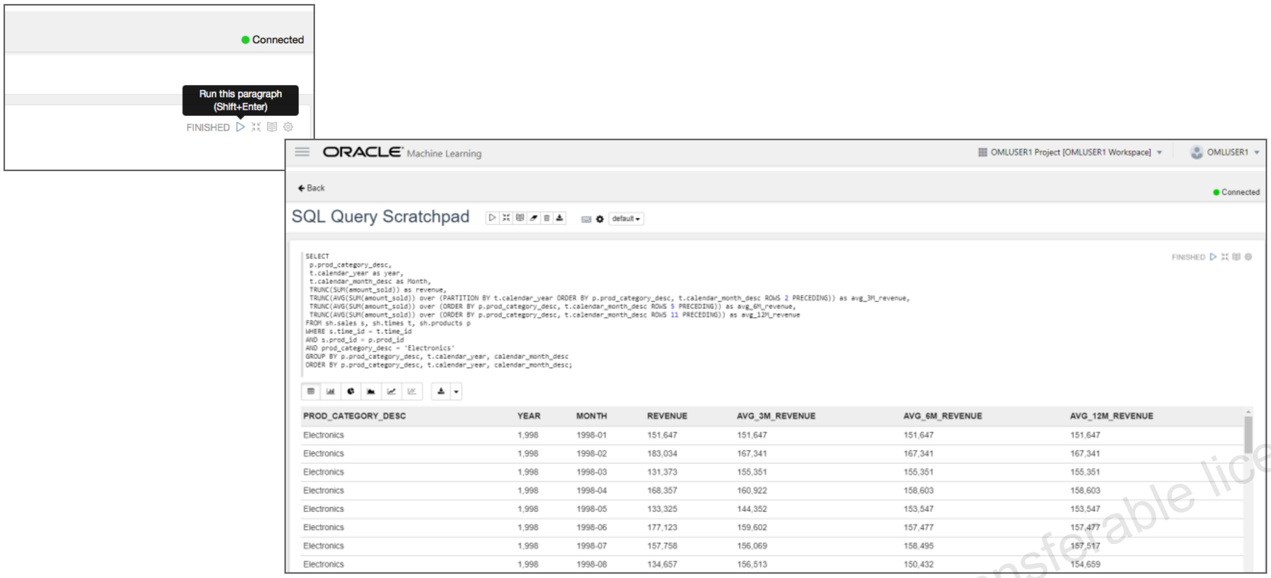
Running a SQL Statement

- From the home page, click the "**Run SQL Statement**" link in the Quick Actions panel to open a new SQL query scratchpad.



- The white panel is an area known as "paragraph."
- Within a scratchpad, you can have multiple paragraphs.
- Each paragraph can contain one SQL statement or a SQL script.

“Run the paragraph”

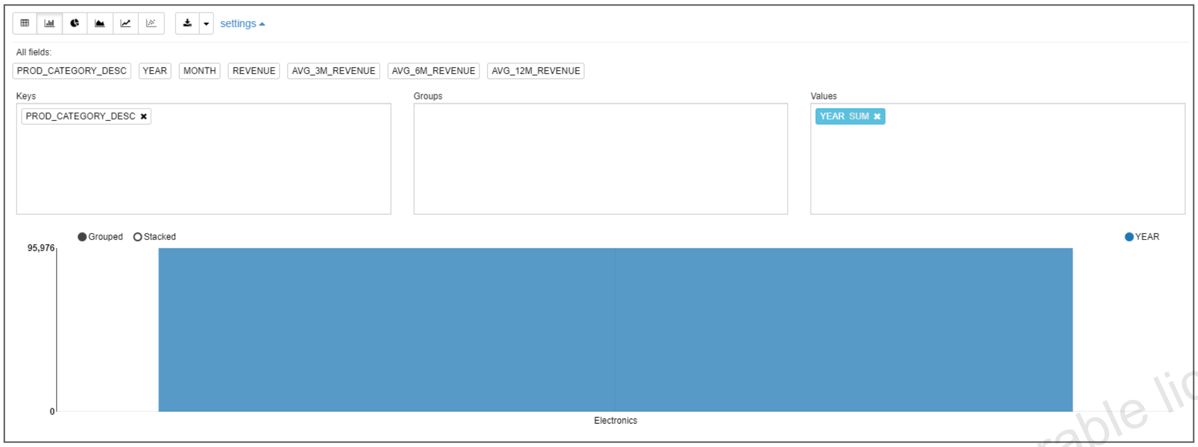


Changing the Report Type

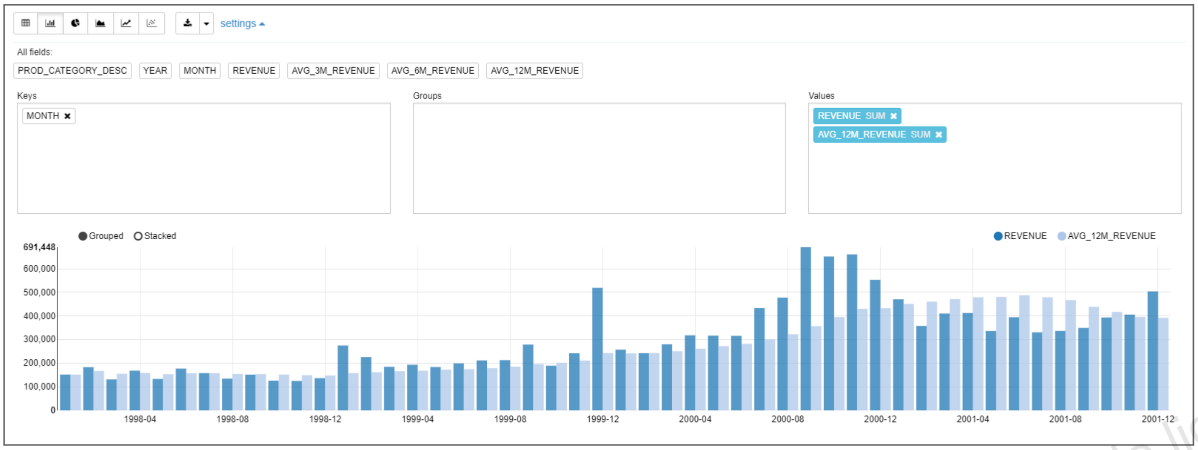
```
SELECT
 p.prod_category_desc,
 t.calendar_year as year,
 t.calendar_month_desc as Month,
 TRUNC(SUM(amount_sold)) as revenue,
 TRUNC(AVG(SUM(amount_sold)) over (PARTITION BY t.calendar_year ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 2 PRECEDING)) as avg_3M_revenue,
 TRUNC(AVG(SUM(amount_sold)) over (ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 5 PRECEDING)) as avg_6M_revenue,
 TRUNC(AVG(SUM(amount_sold)) over (ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 11 PRECEDING)) as avg_12M_revenue
FROM sh.sales s, sh.times t, sh.products p
WHERE s.time_id = t.time_id
AND s.prod_id = p.prod_id
AND prod_category_desc = 'Electronics'
GROUP BY p.prod_category_desc, t.calendar_year, calendar_month_desc
ORDER BY p.prod_category_desc, t.calendar_year, calendar_month_desc;
```

ORACLE

Changing the Report Type

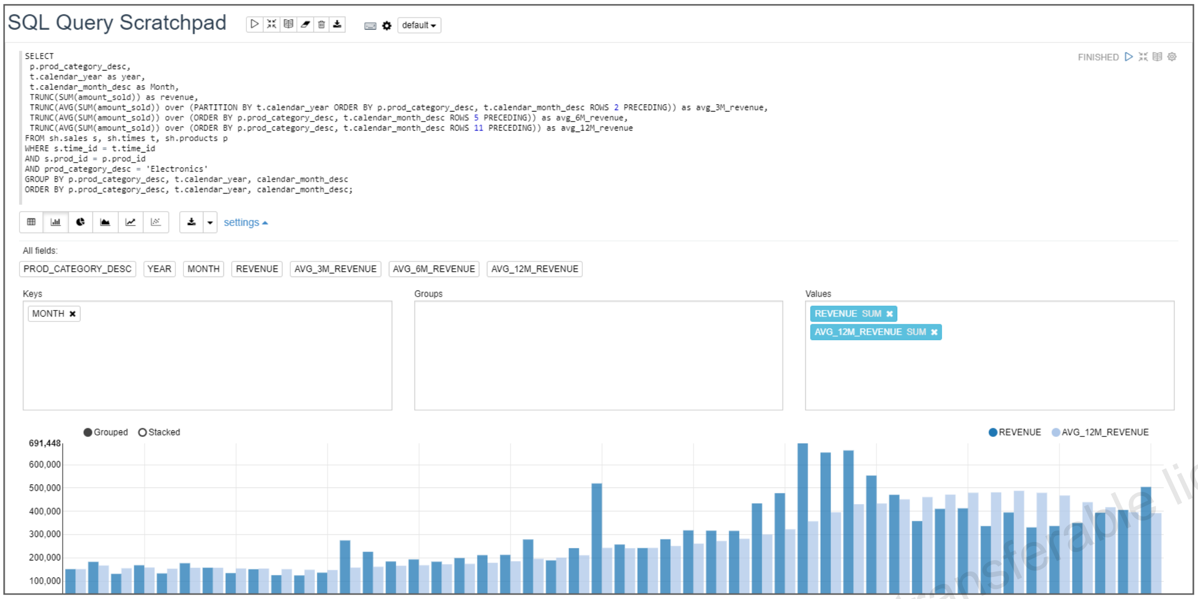


Changing the Report Type



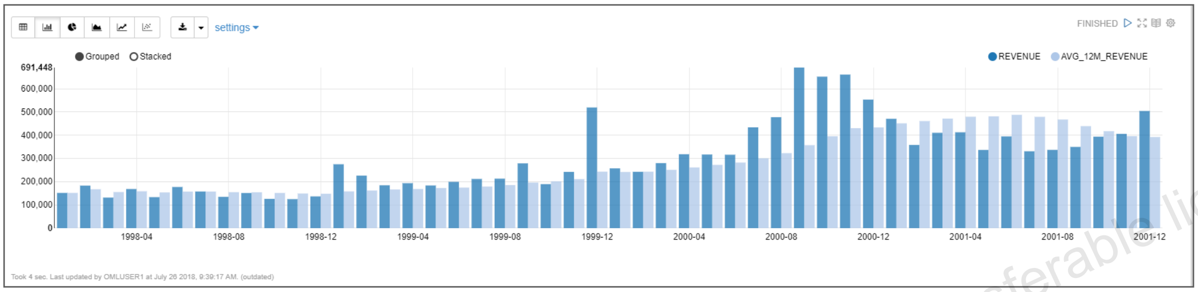
ORACLE

Settings Panel for the Graph

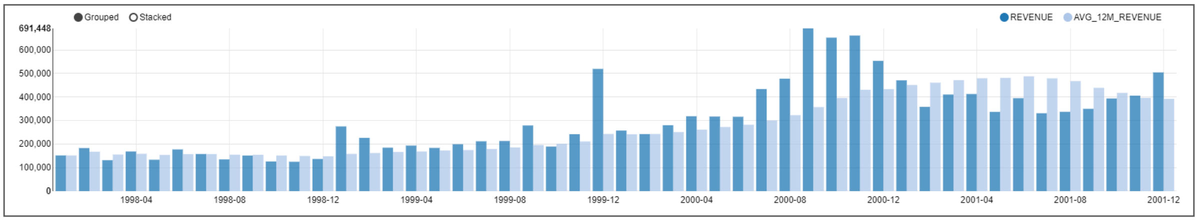


ORACLE

No Editor Button to Display Just Report Results



Just Data Results in a Graph



Summary

You should now be able to:

- Describe how to create users for Autonomous Database
- Describe how to create User Accounts for Oracle Machine Learning
- Describe how to access Oracle Machine Learning User Management
- Discuss working with Oracle Machine Learning for data access, analysis, and discovery
- Describe how to create dashboards, reports, and notebooks

ORACLE

Practice 9: Overview

- Practice 9-1: Creating Projects and Workspaces in Oracle Machine Learning
- Practice 9-2: Creating and Running Notebooks in Oracle Machine Learning
- Practice 9-3: Creating SQL Scripts in Oracle Machine Learning
- Practice 9-4: Running SQL Statements in Oracle Machine Learning
- Practice 9-5: Scheduling Jobs in Oracle Machine Learning

ORACLE

10

Monitoring and Managing Autonomous Database

Objectives

After completing this lesson, you should be able to:
- Monitor performance in an Autonomous Database
- Manage concurrency and priorities
- Manage run-away SQL statements
- Manage Optimizer statistics and hints

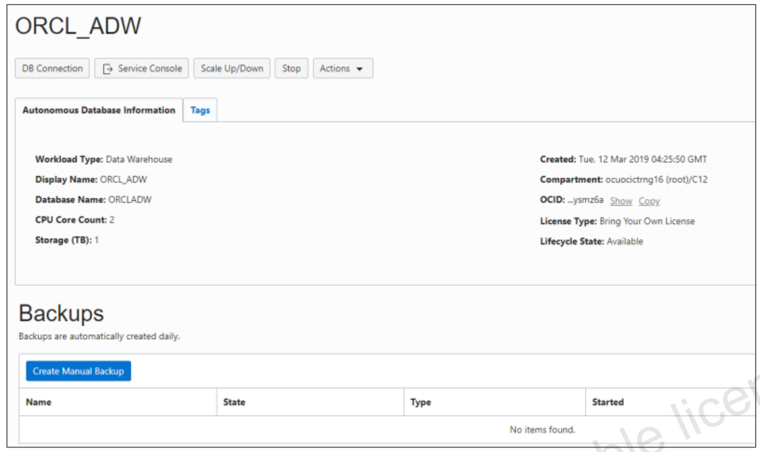
Monitoring Performance in Autonomous Database

ORACLE AUTONOMOUS DATABASE

ORACLE

Automated Management

- Oracle automates end-to-end management of data warehouse:
  - Provisioning new database instances
  - Growing and shrinking storage and/or compute
  - Patching and upgrading
  - Backup and recovery
- The full life cycle is managed using Service Console.

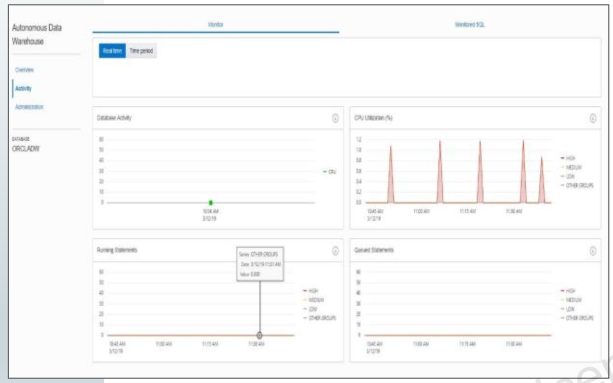


The **Overview** and **Activity** tabs in the Service Console provide information about the performance of the service. The **Activity** tab also shows past and current monitored SQL statements and detailed information about each statement.

Automated Tuning

“Load and go”

- Define tables, load data, and run queries:
  - No tuning required
  - No special database expertise required
  - No need to worry about tablespaces, partitioning, compression, in-memory, indexes, and parallel execution
- Fast performance out of the box with zero tuning
- Simple web-based monitoring console
- Built-in resource-management plans



ORACLE

Monitoring Activity and Utilization

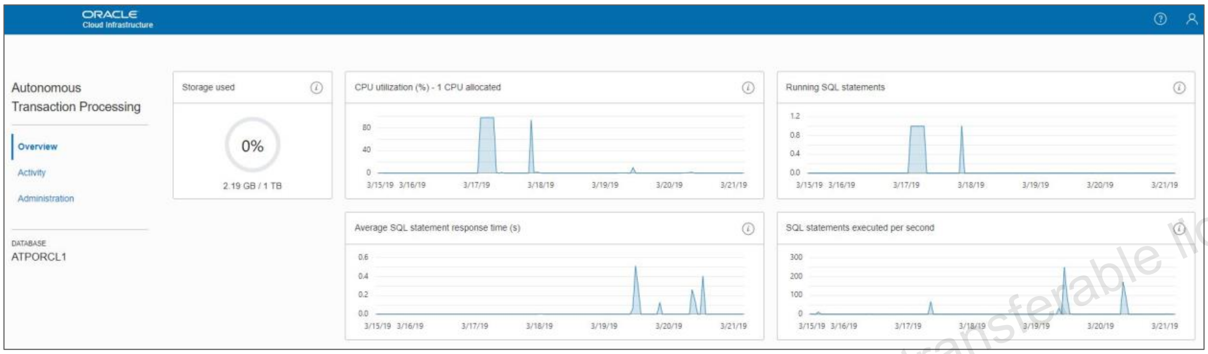
To monitor activity and utilization, perform the following steps:

1. Sign in to cloud.oracle.com.
2. On the OCI page, click ATP or ADW in the left navigation pane.
   – The Overview tab shows the general information about the utilization of the service.
3. From the Autonomous Database landing pad, select an instance.
4. On the Details page, click Service Console.
   – The Service Console button displays the login prompt.
5. At the prompt, enter `ADMIN` as the username and enter the password for the `ADMIN` user.

ORACLE

Monitoring Overview: ATP Workload

- Simplified monitoring using the web-based service console
- Historical and real-time performance charts
- Real-time SQL monitoring to monitor running and past SQL statements
- Historical data load monitoring



ORACLE

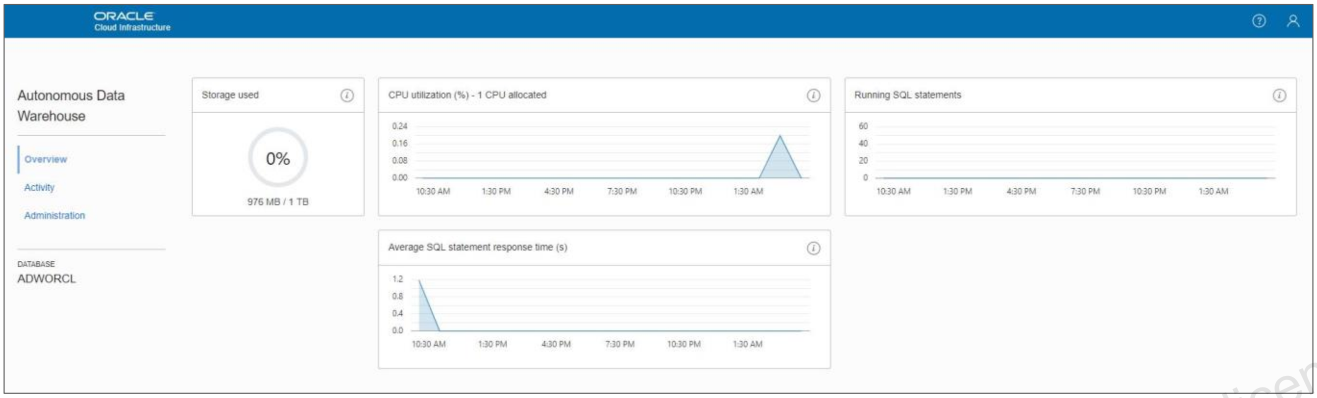
The charts on this page are:

- **Storage:** Shows the total and used storage capacity of the service. It indicates what percentage of space is currently in use.
- **CPU utilization (%):** Shows the historical CPU utilization of the service
- **Running SQL statements:** Shows the average number of running SQL statements historically
- **Average SQL statements response time:** Shows the average response time of SQL statements historically
- **SQL statements executed per second:** Shows the SQL statements executed per second

The default retention period for performance data is eight days. So, the CPU utilization, running statements, and average SQL response time charts show data for the last eight days by default.

The retention time can be changed by modifying the Automatic Workload Repository retention setting with the PL/SQL procedure `DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS`. Be aware that increasing the retention time will result in more storage usage for performance data.

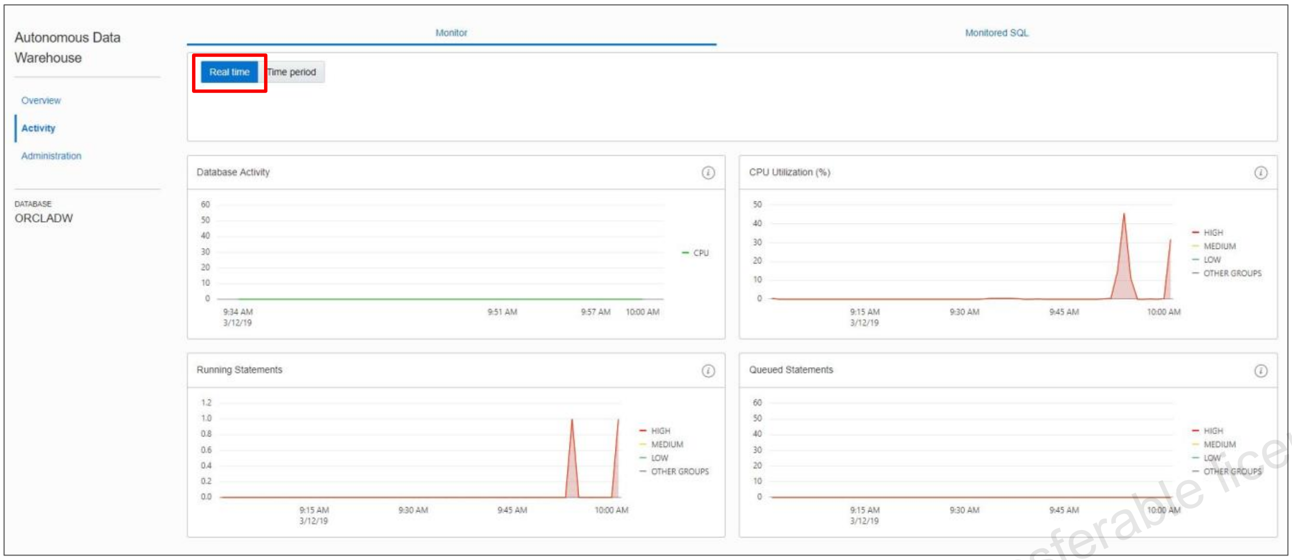
Monitoring Overview: ADW Workload



The charts on this page are:

- **Storage:** Shows the total and used storage capacity of the service. It indicates what percentage of space is currently in use.
- **CPU utilization (%):** Shows the historical CPU utilization of the service
- **Running SQL statements:** Shows the average number of running SQL statements historically
- **Average SQL statements response time:** Shows the average response time of SQL statements historically

Autonomous Database: Activity Dashboard



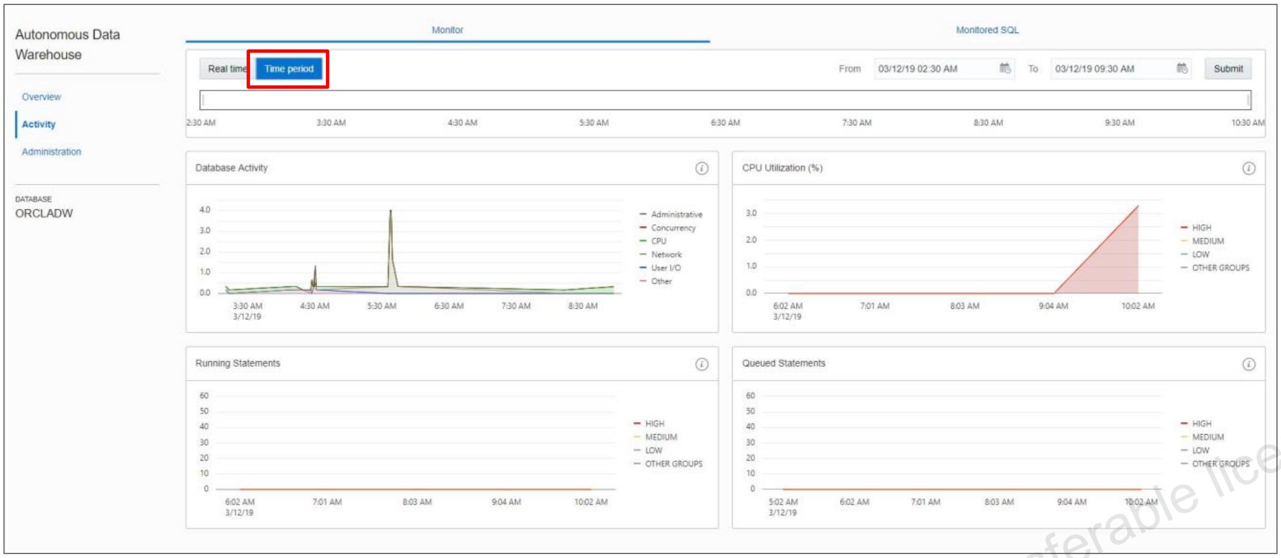
ORACLE

The following charts are on this page:

- **Database Activity:** Shows the average number of sessions in the database using CPU or waiting on a wait event
- **CPU Utilization:** Shows the CPU utilization of each consumer group
- **Running Statements:** Shows the average number of running SQL statements in each consumer group
- **Queued Statements:** Shows the average number of queued SQL statements in each consumer group

The default view on this tab is real time. This view shows performance data for the last hour.

Autonomous Database: Activity Dashboard



To see earlier data, click "Time period." The default retention period for performance data is eight days. So, this view shows information for the last eight days, by default.

In the time period view, you can use the calendar to look at a specific time in the past eight days. You can also use the time slider to change the period for which performance data is shown.

Monitoring SQL Statements

- The Monitored SQL tab on the Activity tab shows information about current and past monitored SQL statements.
- To see detailed SQL Monitor report for a statement, select a statement and click Show Details.
- The Overview tab in the popup shows general information about that statement.
- The Plan Statistics tab will show the runtime execution plan of the statement.
- The Parallel tab will show information about the parallel processes, if the statement uses parallelism.

Managing Concurrency and Priorities in an Autonomous Database

ORACLE AUTONOMOUS DATABASE

ORACLE

Managing Concurrency and Priorities for ADW

- Concurrency and prioritization of user requests are determined by the database service the user is connected with.
- Users are required to select a service when connecting to the database. The service names are in the following format:
  - `databasename_low`
  - `databasename_medium`
  - `databasename_high`
- Services map to the `LOW`, `MEDIUM`, and `HIGH` consumer groups.

ORACLE

Managing Concurrency and Priorities for ADW

- Basic characteristics of consumer groups are:
  - `HIGH`
    - Highest resources, lowest concurrency
    - Queries run serially unless you specify a parallel degree.
  - `MEDIUM`
    - Less resources, higher concurrency
    - Queries run serially unless you specify a parallel degree.
  - `LOW`
    - Least resources, highest concurrency
    - Queries run serially unless you specify a parallel degree.

ORACLE

Managing Concurrency and Priorities for ATP

- Concurrency and prioritization of user requests are determined by the database service the user is connected with.
- Users are required to select a service when connecting to the database. The service names are in the following format:
  - `databasename_tpurgent`
  - `databasename_tp`
  - `databasename_high`
  - `databasename_medium`
  - `databasename_low`
- Services map to the `TPURGENT`, `TP`, `HIGH`, `MEDIUM`, and `LOW` consumer groups.

Managing Concurrency and Priorities for ATP

- **`TPURGENT`**: The highest priority application connection service for time critical transaction processing operations. This connection service supports manual parallelism.
- **`TP`**: A typical application connection service for transaction processing operations. This connection service does not run with parallelism.
- **`HIGH`**: A high priority application connection service for reporting and batch operations. All operations run in parallel and are subject to queuing.
- **`MEDIUM`**: A typical application connection service for reporting and batch operations. All operations run in parallel and are subject to queuing. Using this service, the degree of parallelism is limited to four (4).
- **`LOW`**: A lowest priority application connection service for reporting or batch processing operations. This connection service does not run with parallelism.

ORACLE

Managing Concurrency and Priorities

Idle Time Limits

- The maximum idle time is 60 minutes for all consumer groups.
- An idle session will be terminated after 60 minutes.
- A session can be terminated if it stays idle for more than 5 minutes and the resources used are needed by other users.
- Other active sessions will proceed without waiting for the idle session.

ORACLE

The maximum idle time for all consumer groups is 60 minutes; if a session stays idle for more than 60 minutes, it will be terminated.

A session may also be terminated if it stays idle for more than five (5) minutes and the resources it consumes are needed by other users. This allows other active sessions to proceed without waiting for the idle session.

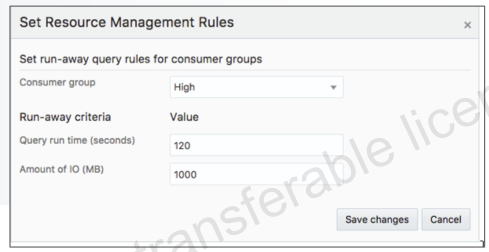
Managing Run-Away SQL Statements on Autonomous Transaction Processing

ORACLE AUTONOMOUS DATABASE

ORACLE

Managing Run-Away SQL Statements

- Configure ADB to terminate SQL statements automatically based on their run time or the amount of I/O they are doing.
- Rules can be set by using the service console or the `cs_resource_manager` PL/SQL package.
- To set rules by using the service console:
  1. In the Service Console, click the **Administrations** tab.
  2. Click **Set Resource Management Rules** to open the page to set run-away query rules.
  3. Select the **Consumer Group.**
  4. Set run-away criteria values, query run time (seconds), and amount of I/O (MB).
  5. Click **Save changes**.



ORACLE

Managing Run-Away SQL Statements

- Use the `cs_resource_manager.update_plan_directive` procedure to set query run time and Amount of I/O for SQL statements.
- The following command can be used to set the limits:

```
BEGIN
  cs_resource_manager.update_plan_directive(consumer_group =>
'HIGH',   io_megabytes_limit => 1000, elapsed_time_limit => 120);
END;
/
```

- To reset the values to lift the limits, set the values to null:

```
BEGIN
  cs_resource_manager.update_plan_directive(consumer_group =>
'HIGH', io_megabytes_limit =>   null, elapsed_time_limit => null);
END;
/
```

Managing Optimizer Statistics and Hints in an Autonomous Database

Managing Optimizer Statistics on ADW Workload

Optimizer stats gathering

- Stats are gathered automatically during direct load operations.
- Users can gather stats manually if they want.

Optimizer hints

- Hints are ignored by default.
- Users can enable hints explicitly.

Result cache configuration

- The result cache is enabled by default for all queries.

ORACLE

Managing Optimizer Statistics on ATP Workload

Optimizer stats gathering:

- Stats are gathered automatically.
- Automatic statistics gathering is enabled by default.
- It runs in a standard maintenance window.

Optimizer hints:

- Optimizer and Parallel hints are honored by default
- Optimizer and Parallel hints can be disabled by setting the `OPTIMIZER_IGNORE_HINTS` and `OPTIMIZER_IGNORE_PARALLEL_HINTS` parameters to `TRUE`.

Summary

In this lesson, you should have learned how to:

- Monitor performance in an Autonomous Database
- Manage concurrency and priorities
- Manage run-away SQL statements
- Manage Optimizer statistics and hints

ORACLE

Practice 10: Overview

- Practice 10-1: Monitoring and Managing Autonomous Database with ADW Workload

ORACLE

11

Backing Up and Restoring an Autonomous Database

Objectives

After completing this lesson, you should be able to:

- Back up an Autonomous Database
- Take manual backup of Autonomous Databases
- Restore an Autonomous Database

Back Up and Restore an Autonomous Database

ORACLE AUTONOMOUS DATABASE

ORACLE

Back Up and Restore an Autonomous Database

- OCI automatically backs up Autonomous Databases.
- The retention period for backups is 60 days.
- A database can be restored and recovered to any point-in-time in the retention period.
- Automatic backups are weekly full backups and daily incremental backups.
- Manual backups can be taken to supplement automatic backups.
- Manual recovery can be initiated from the cloud console.

ORACLE

Oracle Autonomous Database automatically backs up your database for you. The retention period for backups is 60 days. You can restore and recover your database to any point-in-time in this retention period.

Automatic Backups

Autonomous Database automatic backups provide weekly full backups and daily incremental backups.

Manual Backups

You do not have to do any manual backups because Autonomous Database backs up automatically. You can do manual backups using the cloud console; for example, if you want to take a backup before a major change to make restore and recovery faster. The manual backups are put in your Cloud Object Storage bucket. When you initiate a point-in-time recovery, Autonomous Database decides which backup to use for faster recovery.

Recovery

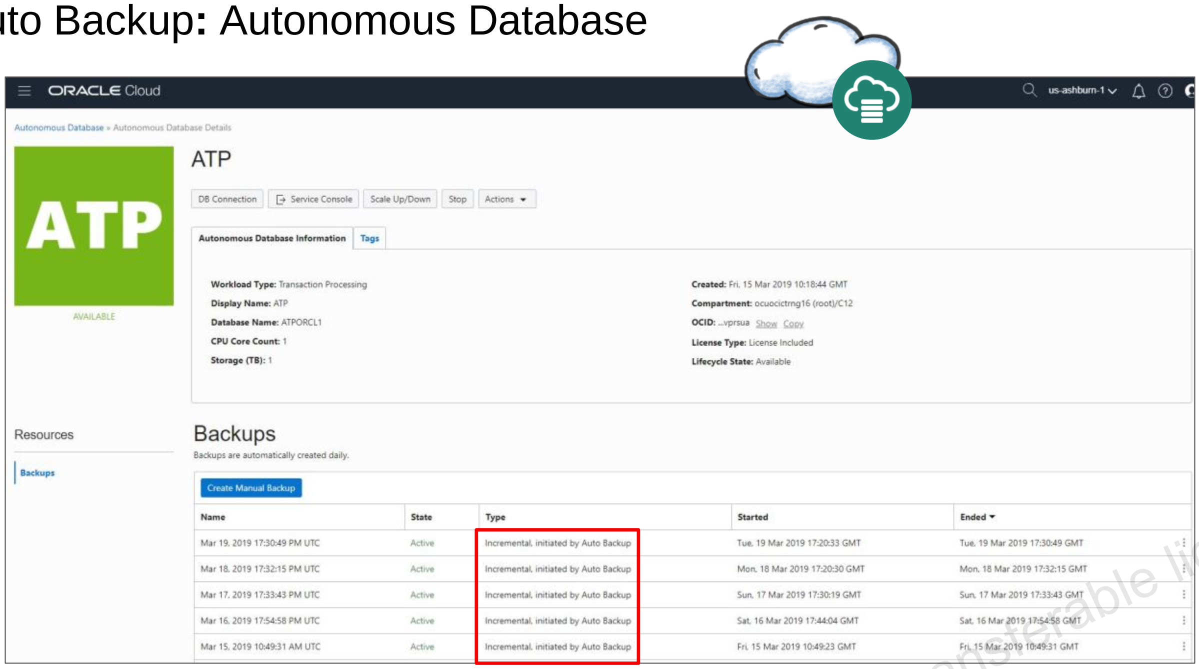
You can initiate recovery for your Autonomous Database using the cloud console. Autonomous Database automatically restores and recovers the database to the point-in-time you specify.

Back Up and Restore an Autonomous Database

- Autonomous Data Warehouse Cloud automatically backs up your database for you.
- The retention period for backups is 60 days.
- You can restore and recover your database to any point-in-time in this retention period.
  - Automatic Backups
    - Autonomous Data Warehouse Cloud automatic backups provide weekly full backups and daily incremental backups.
  - Manual Backups
    - Manual backups are done using the cloud console.
    - Manual backups are put in Cloud Object Storage.

ORACLE

Auto Backup: Autonomous Database



ORACLE

Manual Backup: Autonomous Database

- The Autonomous Database Service Console enables you to initiate manual backups.
- To do this, you need to:
  - Enter the Oracle Cloud Infrastructure Object Storage credentials and your Oracle Cloud Infrastructure Object Storage tenancy URL
  - Create a bucket to hold the backups

ORACLE

In addition to automatic backups, Autonomous Database also allows you take manual backups to your Oracle Cloud Infrastructure Object Storage.

The Autonomous Database console allows you to initiate manual backups. To be able to do this, you need to define your Oracle Cloud Infrastructure Object Storage credentials and your Oracle Cloud Infrastructure Object Storage tenancy URL. You also need to create a bucket to hold the backups. This is a one-time operation. After you define your credentials and your tenancy URL, you can initiate manual backups without doing the same operations again.

Note: Each manual backup creates a full backup on your Oracle Cloud Infrastructure Object Storage bucket and the backup can be used only by the Autonomous Database instance when you initiate a point-in-time-recovery.

Manual Backup: Autonomous Database

To take a manual backup:

1. Set the database `default_bucket` property of Oracle Cloud Infrastructure Object Storage tenancy URL as the `ADMIN` user.

   ```
   ALTER DATABASE PROPERTY SET
   default_bucket='https://swiftobjectstorage.us-phoenix-
   1.oraclecloud.com/v1/test';
   ```

2. On OCI Object Storage, create a folder to hold the backups. The format of the bucket name is `backup_databasename`, where `databasename` is in lowercase.

   ```
   https://swiftobjectstorage.us-phoenix-
   1.oraclecloud.com/v1/atpc/backup_test
   ```

3. Create a credential for OCI Object Storage account using `DBMS_CLOUD.CREATE_CREDENTIAL`.
4. Set the database property `default_credential` to the credential created in the previous step.

   ```
   ALTER DATABASE PROPERTY SET default_credential =
   'ADMIN.DEF_CRED_NAME';
   ```

ORACLE

To define your Oracle Cloud Infrastructure Object Storage credentials and tenancy URL, and to create the bucket for manual backups, perform the following steps:

1. Set the database default_bucket property to your Oracle Cloud Infrastructure Object Storage tenancy URL. The format of the tenancy URL is `https://swiftobjectstorage.region.oraclecloud.com/v1/tenant`. Note that you need to do this as the `ADMIN` user. For example:

   ```
   ALTER DATABASE PROPERTY SET
   default_bucket='https://swiftobjectstorage.us-phoenix-
   1.oraclecloud.com/v1/atpc';
   ```

2. On your Oracle Cloud Infrastructure Object Storage, create a folder to hold the backups. The format of the bucket name is backup_*databasename* (where *databasename* is in lowercase).

   ```
   https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/atpc/backup_atpc1
   ```

3. Create a credential for your Oracle Cloud Infrastructure Object Storage account using `DBMS_CLOUD.CREATE_CREDENTIAL`

   ```
   BEGIN
         DBMS_CLOUD.CREATE_CREDENTIAL(
         credential_name => 'DEF_CRED_NAME',
         username => 'atpc_user@oracle.com',
         password => 'password' );
    END;
   /
   ```

4. Set the database property `default_credential` to the credential you created in the previous step. For example:

```
ALTER DATABASE PROPERTY SET default_credential = 'ADMIN.DEF_CRED_NAME';
```

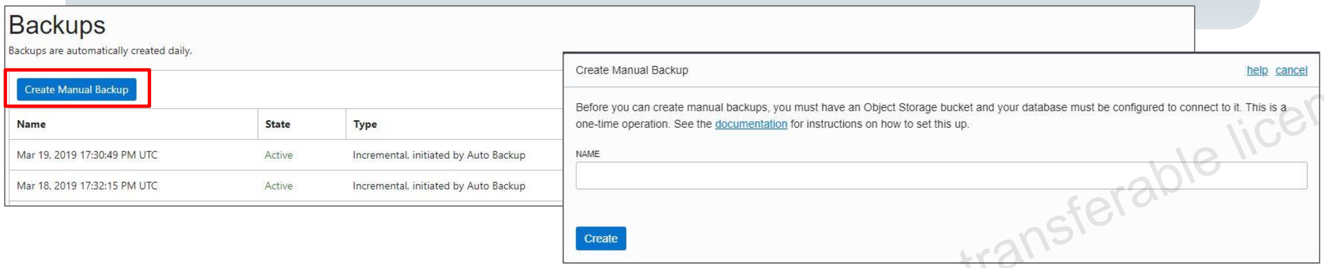
To list the current value for the default bucket, run the following command:

```
SELECT PROPERTY_VALUE from database_properties WHERE
PROPERTY_NAME='DEFAULT_BUCKET';
```

Manual Backup: Autonomous Database

To initiate a manual backup:

1. Sign in to Oracle Cloud account.
2. On the home page, click on the top-left corner, then click **ATP** or **ADW**.
3. Select an instance from the links in the **Name** column.
4. On the details page, under **Backups**, click **Create Manual Backup.**
5. In the **Create Manual Backup** dialog box, enter a name in the **Name** field and click **Create**.



ORACLE

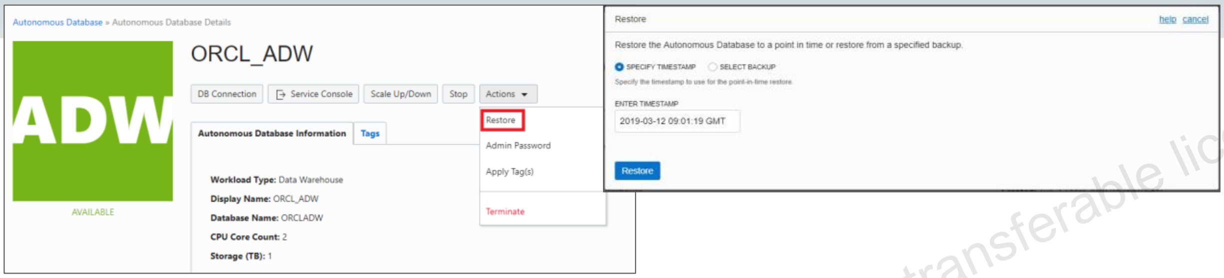
After completing the steps in the previous slides, you can take manual backups any time you want. To initiate a manual backup:

1. Sign in to your Oracle Cloud Account.
2. On the Oracle Cloud Infrastructure page, click on the top left navigation list of the page, then click **ATP or ADW**, and select a Compartment.
3. Select an Autonomous Database instance from the links in the **Name** column.
4. On the details page, under **Backups**, click **Create Manual Backup**.
5. In the **Create Manual Backup** dialog box, enter a name in the **Name** field.
6. In the **Create Manual Backup** dialog box, click **Create**.

Restore: Autonomous Database

To restore a database to a point-in-time or to a specified backup:

1. Sign in to Oracle Cloud Account.
2. From the OCI page, go to **ATP** or **ADW** as per requirement.
3. Select an instance from the links in the **Name** column to restore.
4. On the details page, click **Restore** on the **Actions** tab.
5. In the Restore dialog box, select **SPECIFY TIMESTAMP** or **SELECT BACKUP.**



Summary

In this lesson, you should have learned how to:

- Back up an Autonomous Database
- Take manual backup of Autonomous Databases
- Restore an Autonomous Database

ORACLE

Practice 11: Overview

- Practice 11-1: Restoring an Autonomous Database Instance from an Automatic Backup

ORACLE

12

Starting, Stopping, Scaling, and Terminating an Autonomous Database Instance

Objectives

- Starting and Stopping Autonomous Database instance
- Scaling Autonomous Database resources
- Terminating Autonomous Database instance

ORACLE

Start and Stop an Autonomous Database Instance

ORACLE AUTONOMOUS DATABASE

ORACLE

Starting and Stopping Autonomous Database Instance

Autonomous database instances can be started or stopped by logging into cloud console. When you stop your Autonomous Database, billing stops for CPU usage. Billing for storage continues even when the database is stopped.

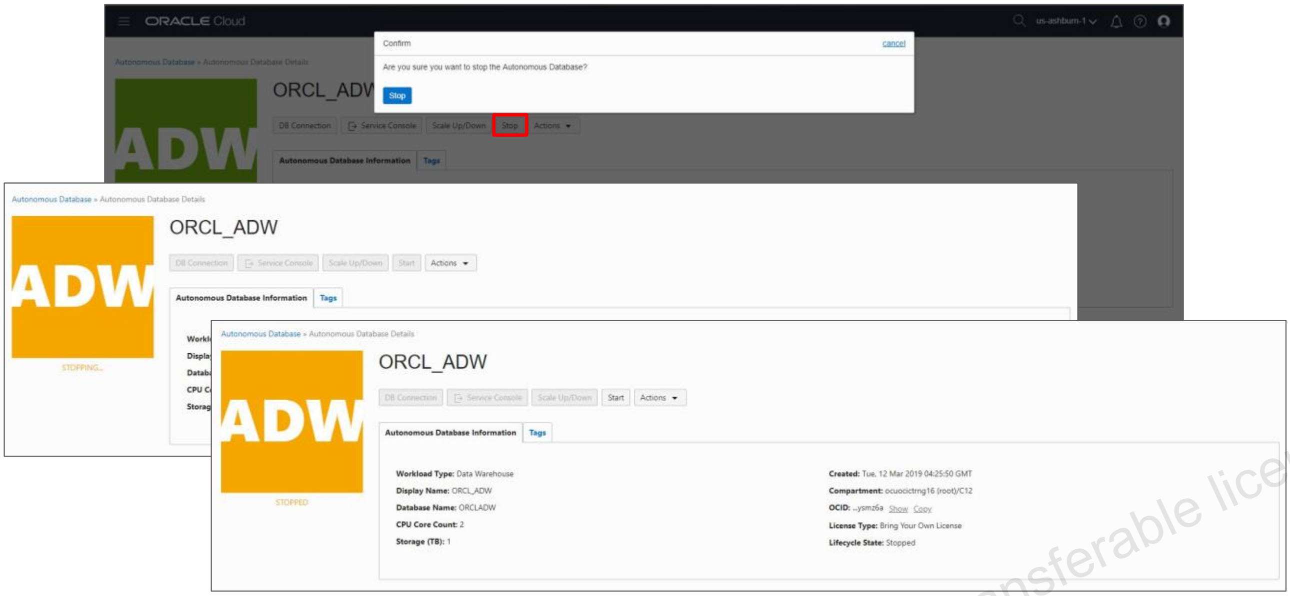
To start and stop an instance:

1. Sign in to your Oracle Cloud Account.
2. From the OCI page, go to ATP or ADW.
3. In the list of Autonomous Databases, click the display name of the database that you want to administer.
4. Click Stop (or Start).
5. Click stop or start an Autonomous Database in the Confirmation dialog box.

ORACLE

You will perform these operations as part of the practice session.

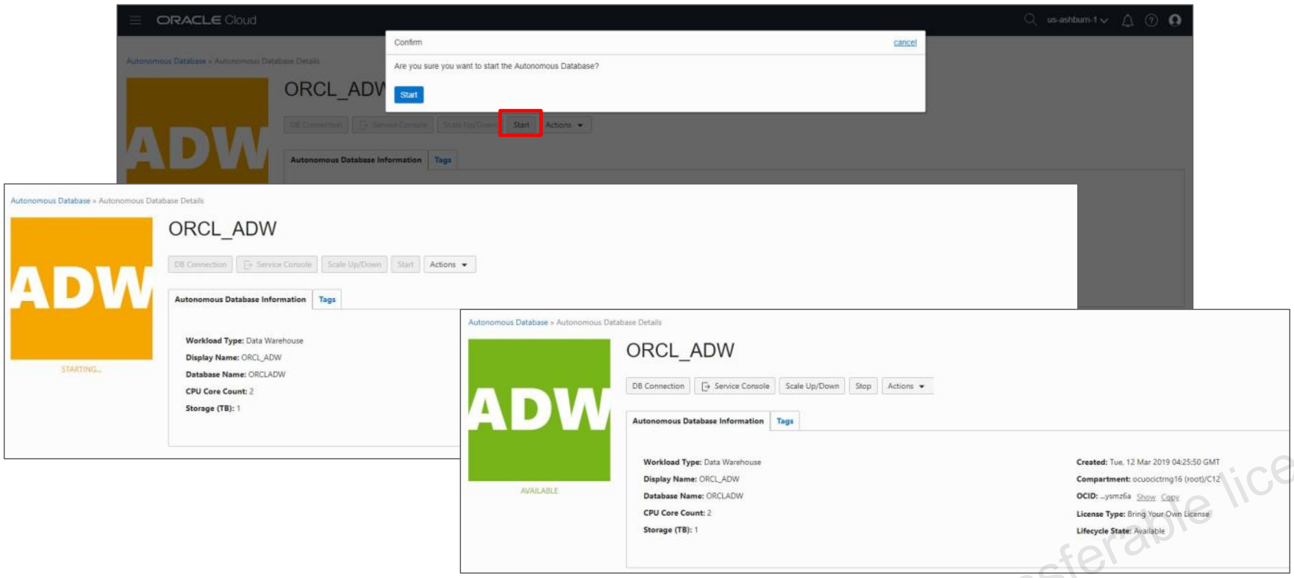
Stopping an Autonomous Database Instance



When an Autonomous Database instance is stopped, the following details apply:

- Tools are no longer able to connect to a stopped instance.
- Autonomous database in-flight transactions and queries are stopped.
- Autonomous database CPU billing is halted based on full-hour cycles of usage.

Starting and Autonomous Database Instance



ORACLE

Stopping an Autonomous Database Instance

Lifecycle State:	Stopping... - The Autonomous Transaction Processing operation failed due to an unknown error. Refer to JobId e8ad01df-e479-445c-b8e4-5552ef0a8b99 and WorkRequestId 0f886418-b17a-4fd0-8c16-5db39f1edb6f when opening a Service Request at My Oracle Support.

Any error, while stopping an Autonomous Database, requires you to raise an SR with Oracle along with your CSI details, JOBID, and workRequestId. This information is gathered when “Stopping instance” is clicked.

ORACLE

Scale Up/Down Autonomous Database Resources

Scaling Up and Down Resources

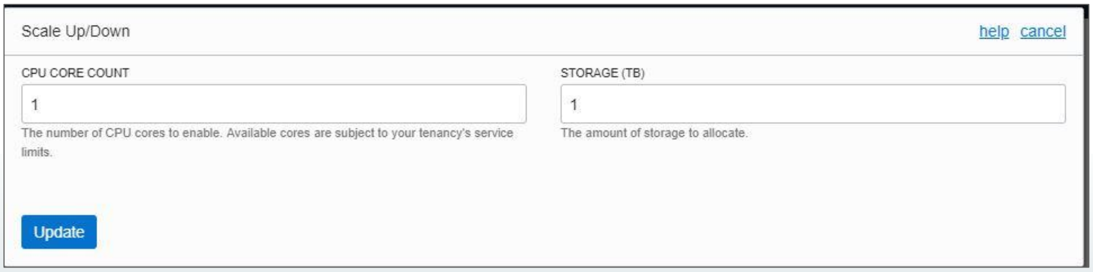
The Autonomous Database console enables you to add or remove CPU and storage resources as per your need.

- CPU Core Count or Storage is between 1 and 128. The number you enter represents the desired final value for your environment for CPU Core Count or Storage.
- The number of available cores is subject to your tenancy's service limits. An Autonomous Database can have a maximum of 128 cores and 128 TB of storage. Scaling the CPU Core Count will affect your CPU billing.

Scale your database on demand without tedious manual steps:

- Independently scale compute or storage
- Resizing occurs instantly, fully online
- Memory, I/O bandwidth, concurrency scales linearly with CPU
- Close your database to save money when not used
- Restart instantly

Adding/Removing CPU or Storage Resources



Navigate to the Autonomous Database instance administration page:

1. On the details page, click **Scale Up/Down**.
2. On the Scale Up/Down prompt, select the change in resources for your scale request.
3. Click **Update** to change your resources.

ORACLE

You will perform these operations as part of the practice session.

Terminating an Autonomous Database

ORACLE AUTONOMOUS DATABASE

ORACLE

Terminating an Autonomous Database

- Terminating an Autonomous Database permanently deletes it.
  - The database data, including automatic backups, will be lost when the system is terminated.
  - Manual backups remain in Object Storage and are not automatically deleted when you terminate an Autonomous Database.
- Oracle recommends that you create a manual backup before terminating.

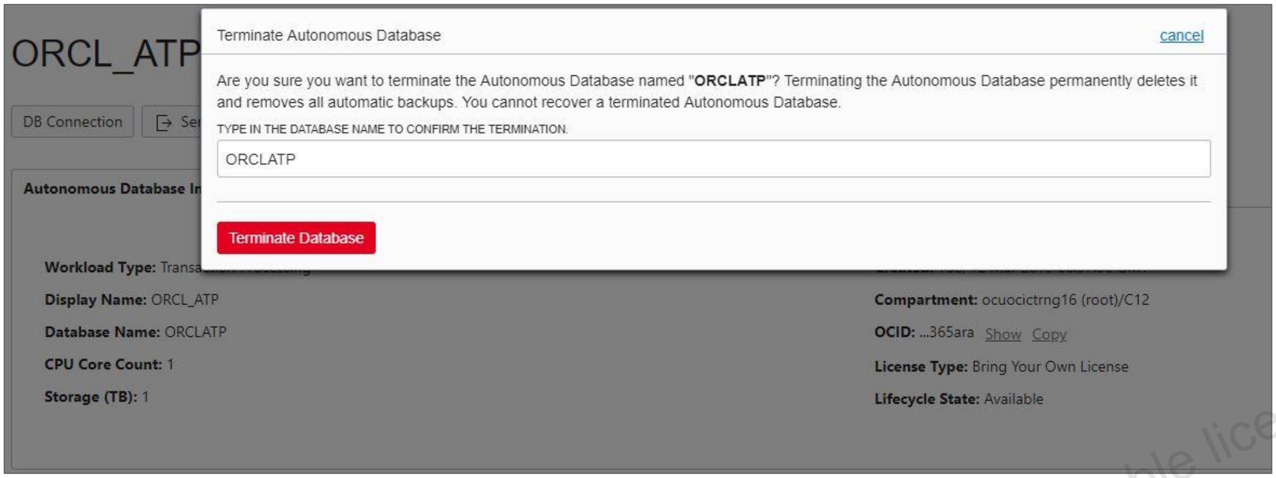
ORACLE

Terminating an Autonomous Database

1. Sign in to your Oracle Cloud Account.
2. From the OCI page, go to ATP or ADW.
3. From the list of Autonomous Databases, click on the display name of the database you want to administer.
4. On the Details page, select **Terminate**.
5. Enter the name of the Autonomous Database in the dialog box.
6. Click **Terminate Database** to delete the instance.

ORACLE

Terminating Autonomous Transaction Processing



ORACLE

When you initiate instance termination, a warning dialog box appears as shown in the slide. You will be asked to ***type in your database name***, to confirm your action to terminate the instance. If you want to proceed with terminating your instance, review the message, enter your instance name, and click Terminate Database.

You will perform this operation as part of the practice session.

Note: Display Name and Database Name are different.

Summary

In this lesson, you should have learned how to:

- Start and stop Autonomous Database instance
- Scale Autonomous Database resources
- Terminate Autonomous Database

ORACLE

Practice 12: Overview

- Practice 12-1: Start and Stop an Autonomous Database Instance
- Practice 12-2: Scale Up CPU and Storage Resources
- Practice 12-3: Scale Down CPU and Storage Resources
- Practice 12-4: Terminating an Autonomous Database Instance

ORACLE

13

Oracle Cloud Infrastructure Security

ORACLE

Objectives

After completing this lesson, you should be able to:

- Identify key features of Identity and Access Management Service
- Identify the security features of Oracle Database on Oracle Cloud Infrastructure
- Describe how Oracle Cloud Infrastructure resources are secured
- Explain why security is a shared responsibility

ORACLE

OCI Security Features: Overview of Database Service

No	Security Capability	OCI DBCA Security Feature
1	Instance security isolation	OCI bare metal (BM) instance
2	Network security and access control	VCN, VCN Security Lists, VCN public and private subnets, VCN route table
3	Secure and highly-available connectivity	VPN DRGs
4	User authentication and authorization	IAM tenancy, compartments and security policies, console password, API signing key, SSH keys
5	Data encryption	DBaaS TDE, RMAN encrypted back-ups, storage and object encryption at rest
6	End-to-end TLS	LBaaS with TLS1.2, customer-provided certificates
7	Auditing	OCI API audit logs

ORACLE

Security Features of the OCI Database Service

This includes instance security isolation, network security and total isolation, IPSEC VPNs and FastConnect dedicated network circuits, granular controls for users, in addition to secure methods for API, and SSH access. We also have TDE on by default, with all backups and block and object storage encrypted by default.

Identity and Access Management Service

- Identity and Access Management (IAM) service enables you to control who can do what in your OCI account:
  - Control who can access your OCI account
  - What services and resources they can use
  - How they can use these services and resources
- Resource is a cloud object that you create and use in OCI (for example, compute instances, block storage volumes, and Virtual Cloud Networks).
- IAM uses traditional identity concepts, such as Principals, Users, Groups, and Policies.
- OCI IAM introduces a new feature called Compartments.

ORACLE

IAM service basically lets you control who can access your OCI account, what services and resources they can use, and how they can use these resources. In a cloud, a resource is an object that you create like compute instances, and so on. There are five concepts in IAM – Principals, Users, Groups, Policies, and Compartments. Compartment is a unique OCI feature that we'll look into in later slides.

Principals

- A principal is an IAM entity that is allowed to interact with OCI resources.
- There are two types of principals:
  - IAM Users/Groups
    - When customers sign up for an OCI account, the first IAM user is the default administrator.
    - The default administrator sets up other IAM users and groups.
    - Users are persistent identities set up through IAM service to represent individual people or applications.
    - Users enforce the security principle of least privilege.
      - A user has no permissions until placed in one (or more) group.
      - A group has at least one policy with permission to tenancy or a compartment.
    - A group is a collection of users who all need the same type of access to a particular set of resources.
    - A user can be a member of multiple groups.
  - Instance Principals
    - Instance Principals let instances (and applications) make API calls against other OCI services removing the need to configure user credentials or a configuration file.

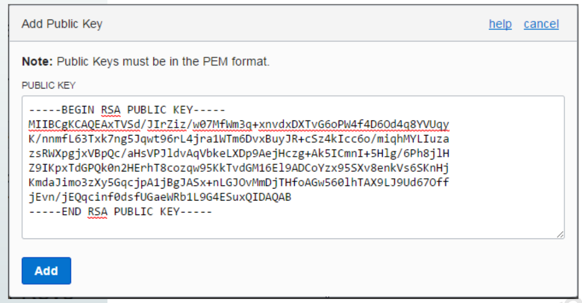
ORACLE

A principal is an IAM entity that is allowed to interact with OCI resources. The three principals that can authenticate and interact with OCI resources are root users, IAM users, and groups and instance principals. The root user is associated with the actual OCI account and cannot be restricted in any way. IAM users and groups are persistent identities that can be controlled through the IAM service.

Authentication

IAM service authenticates a principal by:

- Username, Password
  - You use the password to sign in to the web console.
  - An administrator will provide you with a one-time password when setting up your account.
  - At your first log in, you are prompted to reset the password.
- API Signing Key
  - The API Signing Key is required when using the OCI API in conjunction with the SDK/CLI.
  - The key is an RSA key pair in the PEM format (minimum 2048 bits required).
  - In the interfaces, you can copy and paste the PEM public key.



When you log in to the OCI console as a root user or IAM user, you use a username password combination. A program that accesses the API with an IAM user or root user uses an API signing key.

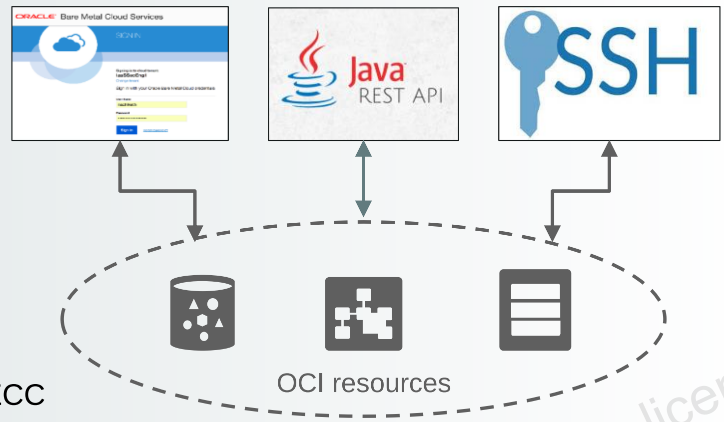
Authorization

- Authorization is the process of specifying what actions an authenticated principal can perform.
- Authorization in IAM service is done by defining specific privileges in policies and associating them with principals.
- Authorization supports the security principle of least privilege; by default, users are not allowed to perform any actions. (Policies cannot be attached to users, but only groups.)
- Policies comprise one or more statements, which specify what groups can access what resources and what level of access users in that group have.
- Policies are written in human-readable format:
  - Allow group <group_name> to <verb> <resource-type> in tenancy
  - Allow group <group_name> to <verb> <resource-type> in compartment <compartment_name> [where <conditions>]
- Policies can be attached to a compartment or the tenancy. Where you attach it controls who can then modify it or delete it.

Note that in OCI, policies cannot be attached directly to IAM users, but only groups.

User Authentication: OCI Security Credentials

- Console password
  - Access to OCI console
- API signing key
  - Access to OCI REST APIs
  - Signed API calls over TLS1.2
  - 2048-bit RSA key pair
- SSH key pair
  - Access to OCI instances
  - 2048-bit RSA or DSA, 128-bit ECC



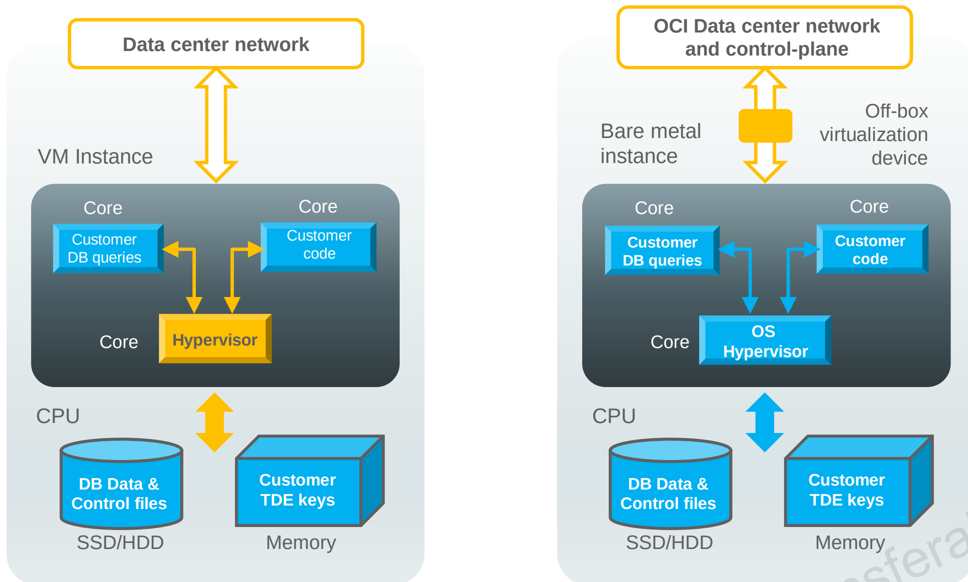
Console access is done by a password.

Two-factor authentication is on its way as well.

Access to the OCI API is done via a 2048-bit key pair. This is done on a per-user basis. These keys can be used to access OCI not only with the REST API, but also CLI, the SDKs, the chef knife plug-in, and with terraform.

Instances by default allow access only to SSH over the network. The instances are configured for SSH access with key-based authentication only. Password login, root login, and weak keys are disabled by default. This includes Ubuntu, Centos, and Oracle Linux as well as the database nodes.

Instance Isolation: OCI Database Bare Metal (BM) Instance



For every instance we have isolation in place.

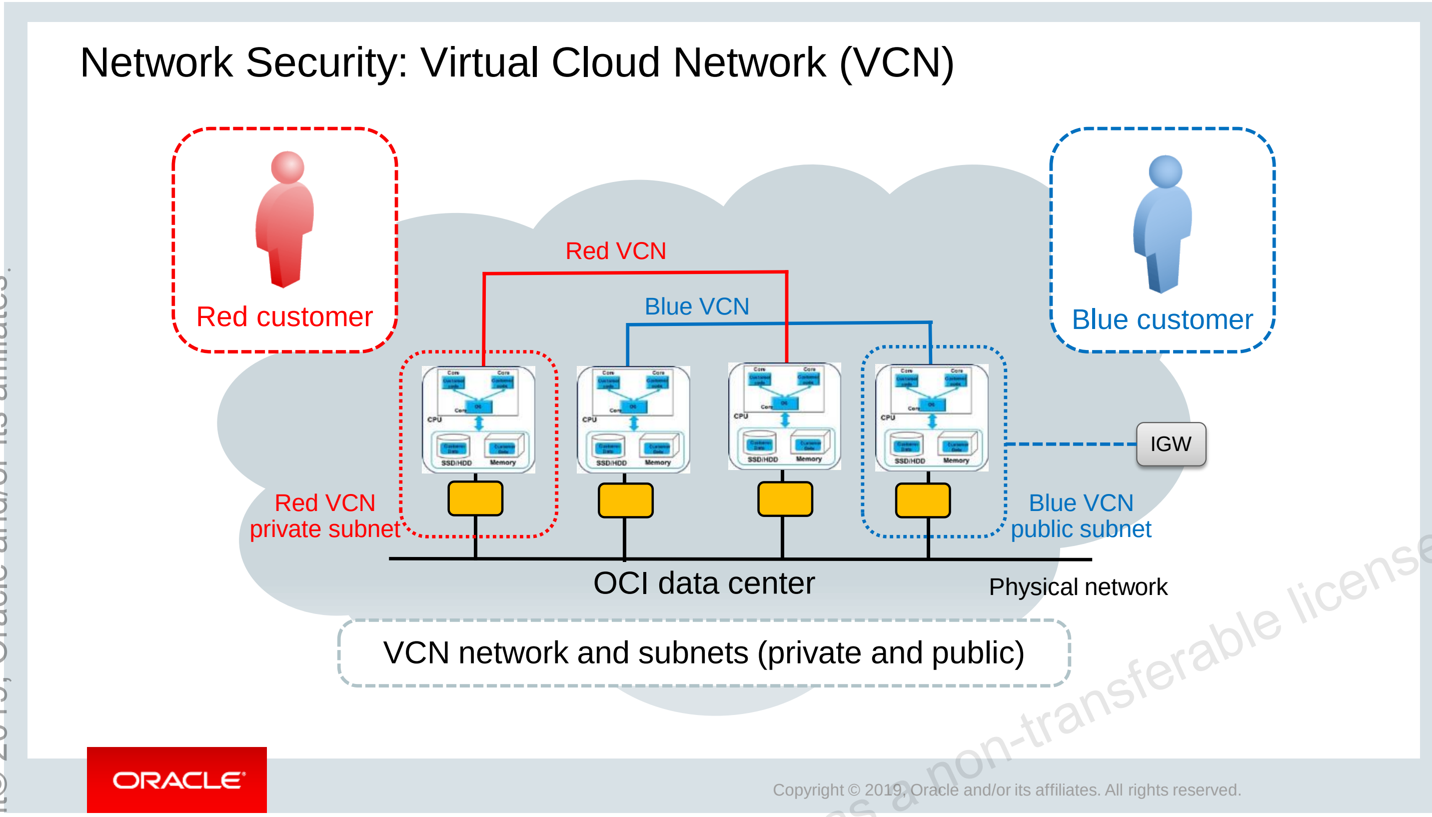
For VM shapes, the hypervisor separates instances from each other by default. Memory, CPUs, and storage data is never accessible by another tenant. For VMs, the network is virtualized off-box so there is no shared network or vswitch isolation issues on the host.

On VMs, the storage is done to block, which is highly secure—rotating encryptions keys and data to the service is encrypted in flight over dedicated networks—not shared with the VM networks. Block also stores the data at rest encrypted.

On bare metal database instances including RAC, there is no shared tenancy model at all. The machines and the network are fully dedicated to the tenant. There is nothing shared at all. The entire CPU and memory is dedicated to the tenant, the network is fully virtualized off-box, and storage is fully encrypted.

For Exadata, quarter- and half-rack shapes can be shared but never on the same physical storage or compute nodes. There is a virtual context in each of the compute nodes to isolate operators from common resources. Administrators still have full access to the context but cannot make system-wide changes; just changes that affect their own databases, compute, and storage nodes.

Networking and cluster networking is divided such that it is nonblocking and storage and compute nodes are dedicated to a user's shape. The cluster network is partitioned so that no inter-tenant data leakage or contention is possible.



Networking in OCI is completely virtualized. Access can be restricted between the tenancies and within the tenancy itself.

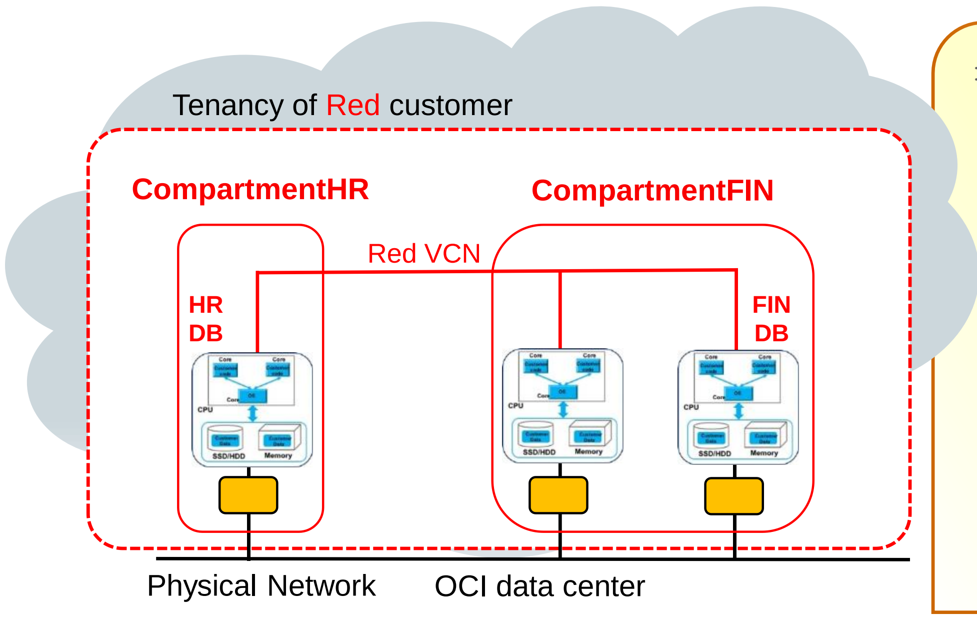
Tenancies are completely network isolated. Permissions can be used to isolate networking between compartments and between VCNs.

Within a given VCN, subnets can be created on different availability domains. Subnets can either be public—where instances can optionally receive a public IP address—or private where it is not possible for instances to receive a public IP address and will have to rely on networking from something other than the Internet Gateway.

Because the networking is done off-box, no amount of rooting or compromising the hypervisor on shared hosts or abuse on bare metal hosts will give access to other networks. It's not isolated at the vswitch, it's done off host.

The networking in OCI provides a security environment to create tiers and provide security to keep public traffic away from the database as well as keeping the database service away from friendly-fire incidents within the tenancy with compartments and policy.

User Authorization: OCI IAM




```
IAM Security Policies:

  Allow group
  DatabaseAdmins to
  manage databases
  in tenancy

  Allow group
  HR to read
  databases in
  CompartmentHR

  Allow group
  Finance (FIN) to
  read/write
  databases in
  CompartmentFIN
```

ORACLE

With IAM, policies can isolate who can interact with the database service.

We can give access across the entire tenancy for a group of database administrators to fully manage all databases. However, groups that enabled networking or other permissions would not be able to make changes to the service.

In addition to that, we can create policies that isolate users of their own compartments from one another or keep users in their own compartments from managing their own database instances. A policy could be defined to allow users in the HR group to perform read-only operations on database service instances in their compartments, and another can be made to allow those database users in the finance compartment. Remember that these are restrictions to the OCI Database Service itself, and each service has a definition of what inspect, read, user, and manage can do.

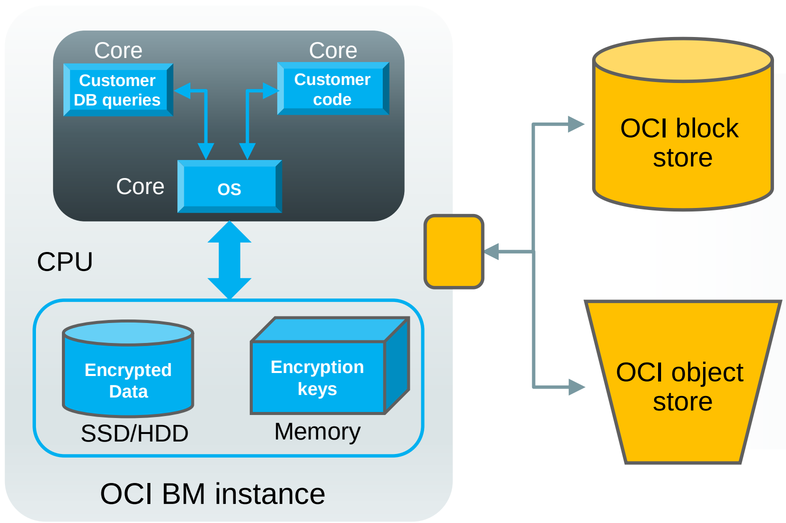
For the database, service policies can be implemented on:

- db-systems
- db-nodes
- db-homes
- Individual databases

Permissions are given as follows: inspect > read > use > manage. This model is done with least-privilege. No statement for a group to a given resource means no access.

Permissions are as granular to include individual database, database backup operations, database home management, db node management (for example, start, stop, restart), general backup operations including managing database backups, and DataGuard failover operations.

Data Encryption: OCI Storage Encryption

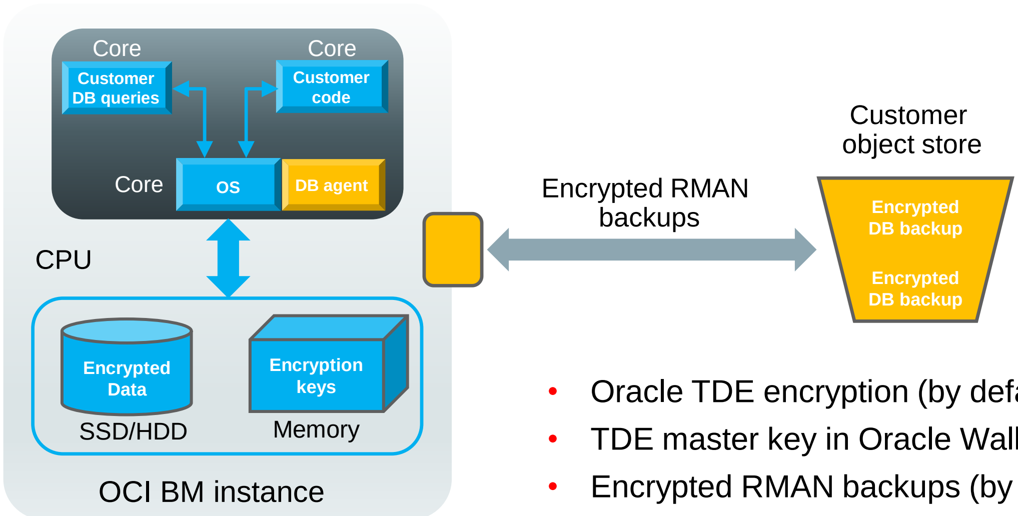


- Customer encryption
  - Client-side encryption
  - Customer-controlled keys
- OCI block storage
  - Encryption at rest
- OCI object store
  - Per-object encryption

OCI block storage has in-flight data encryption as well as encryption at rest. Data is persisted to a minimum of three copies in an availability domain and is durable and secure.

OCI object storage has a per-object security and encryption mechanism. Authentication requests to objects can be done at the bucket level or at the object level. The object store itself is a region and maintains a minimum of three copies of an object all times (more is common) and spreads them throughout the region. Access to the object is secure and encrypted and these objects are encrypted at rest as well.

Data Encryption: OCI Database Service TDE



- Oracle TDE encryption (by default)
- TDE master key in Oracle Wallet on BM instance
- Encrypted RMAN backups (by default)

The OCI database service is set by default to enable TDE on all databases by default.

The keys are managed by the service as well; so, if the system or databases are restarted, the keys are automatically used to reopen / restart the databases.

TDE can be disabled if more performance is needed, but TDE is on by default.

The administrator of the OCI database system can get to the mount point, which contains all the wallets for all the databases and uses tools such as `orapki` and Wallet Manager to manage those keys.

By default all of the backups are also encrypted. If automatic backups are selected, the full backups to the object will be fully encrypted. Restore operations require the correct password or manual key management.

In the Cloud, Security Is a Shared Responsibility

Security Managed by Oracle

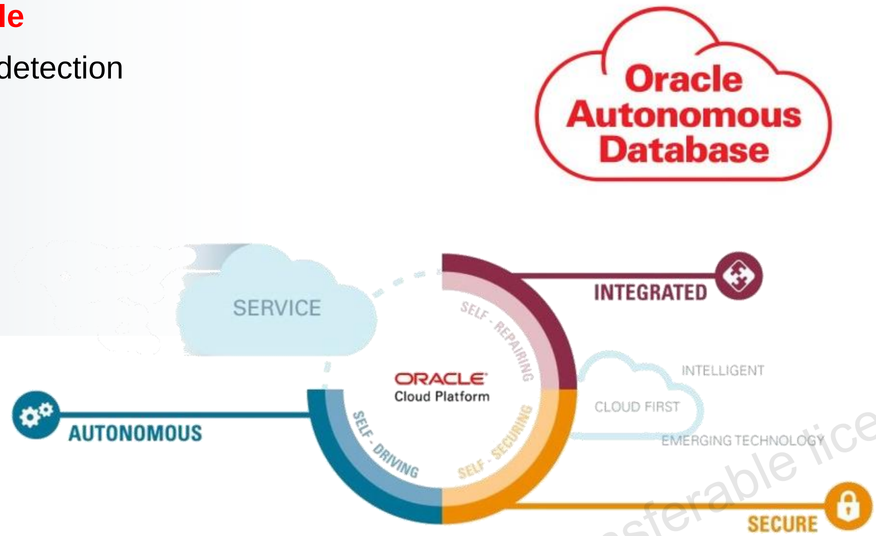
Network security and threat detection

Strong platform security

Automatic database patches

Strict administrative control

Data encryption by default



ORACLE

Security

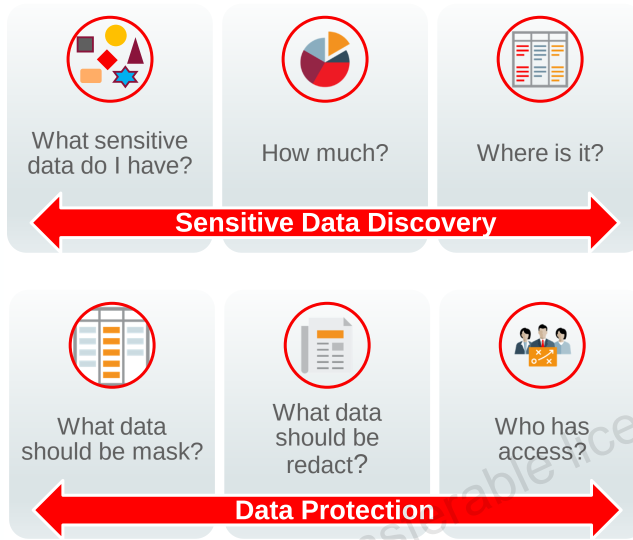
What Can I Do to Prepare?

Identify your assets:

- Know what data you have and where

Secure all databases:

- Make sure that there are no insecure settings, default passwords, and so on.
- Remove unnecessary privileges.
- Determine what data should be masked in dev and test environments.
- Determine what data should be redacted or dynamically masked in applications.



ORACLE

Security

What Can I Do to Prepare?

Encrypt your data:

- All cloud services encrypt data.

Back up your data:

- Make sure you can recover from backups.

Work with application development:

- See if redaction is applicable to current applications.

ORACLE

Summary

In this lesson, you should have learned to:

- Identify key features of Identity and Access Management Service
- Identify the security features of Oracle Database on Oracle Cloud Infrastructure
- Describe how Oracle Cloud Infrastructure resources are secured
- Explain why security is a shared responsibility

ORACLE

Practice 13: Overview

There are no practices for this lesson.

14

Managing an Autonomous Database Using Command Utilities and APIs

ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe the Oracle Cloud Infrastructure REST APIs
- Describe Oracle Cloud Infrastructure - Command Line Interface (OCI CLI)
- Explain Command Line Syntax
- Install and configure OCI CLI
- Manage autonomous databases by using OCI CLI
- Use Terraform to manage autonomous databases

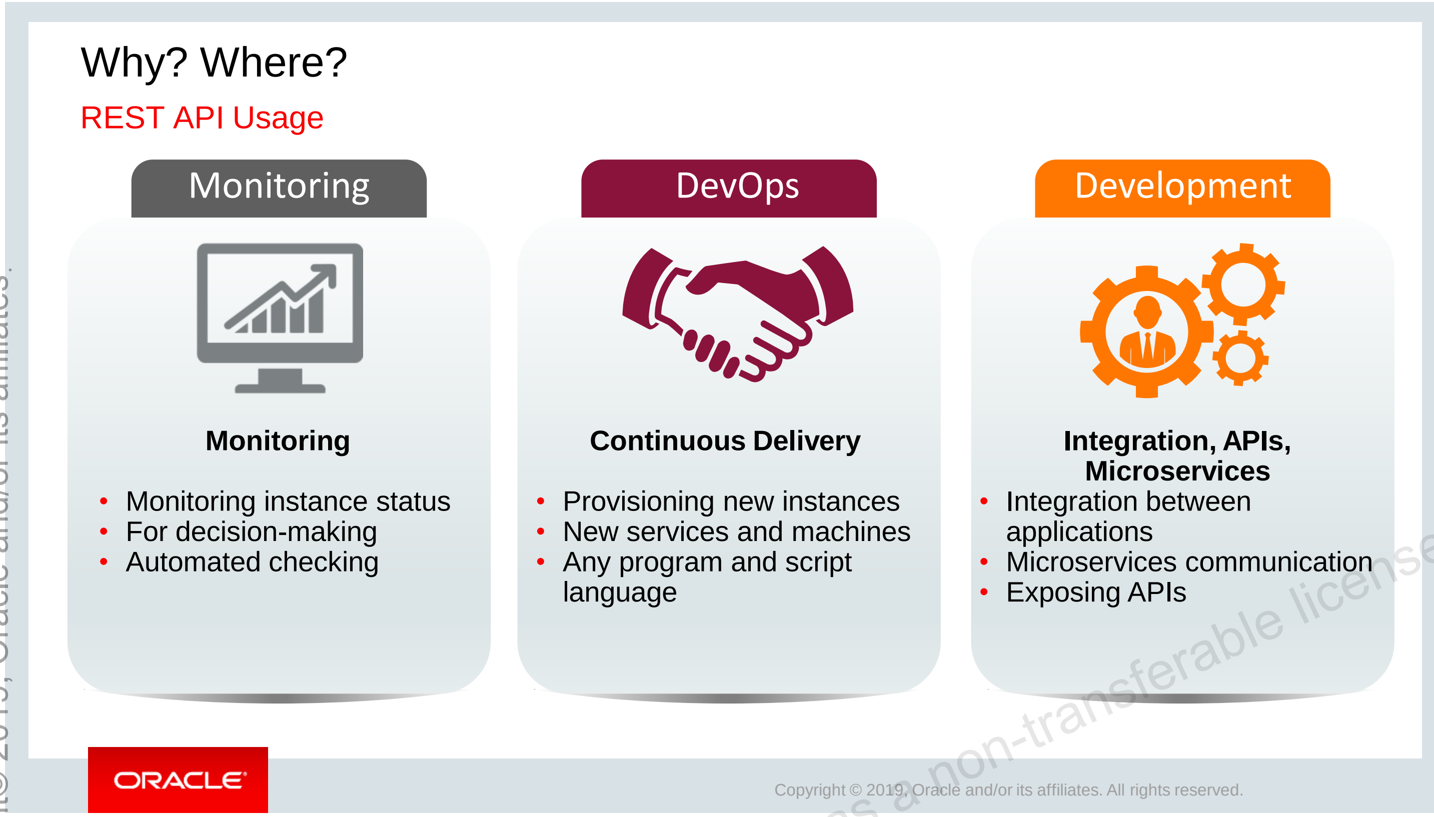
ORACLE

OCI REST APIs

- Alternate method for managing OCI resources
- Provide secured access to cloud resources via command line
- Use Web protocols to programmatically control OCI resources

The Oracle Cloud Infrastructure (OCI) APIs are typical REST APIs that use HTTPS requests and responses.

Note that the Database Service APIs include functions that are common to all Database Services on OCI.



The main use cases for REST APIs are:

- **Development:** In development, these APIs can be used for communication between applications. REST APIs are increasingly being used instead of traditional XML-based SOAP web services. They allow developers to expose their APIs by using REST and allow third-party applications to consume the APIs.

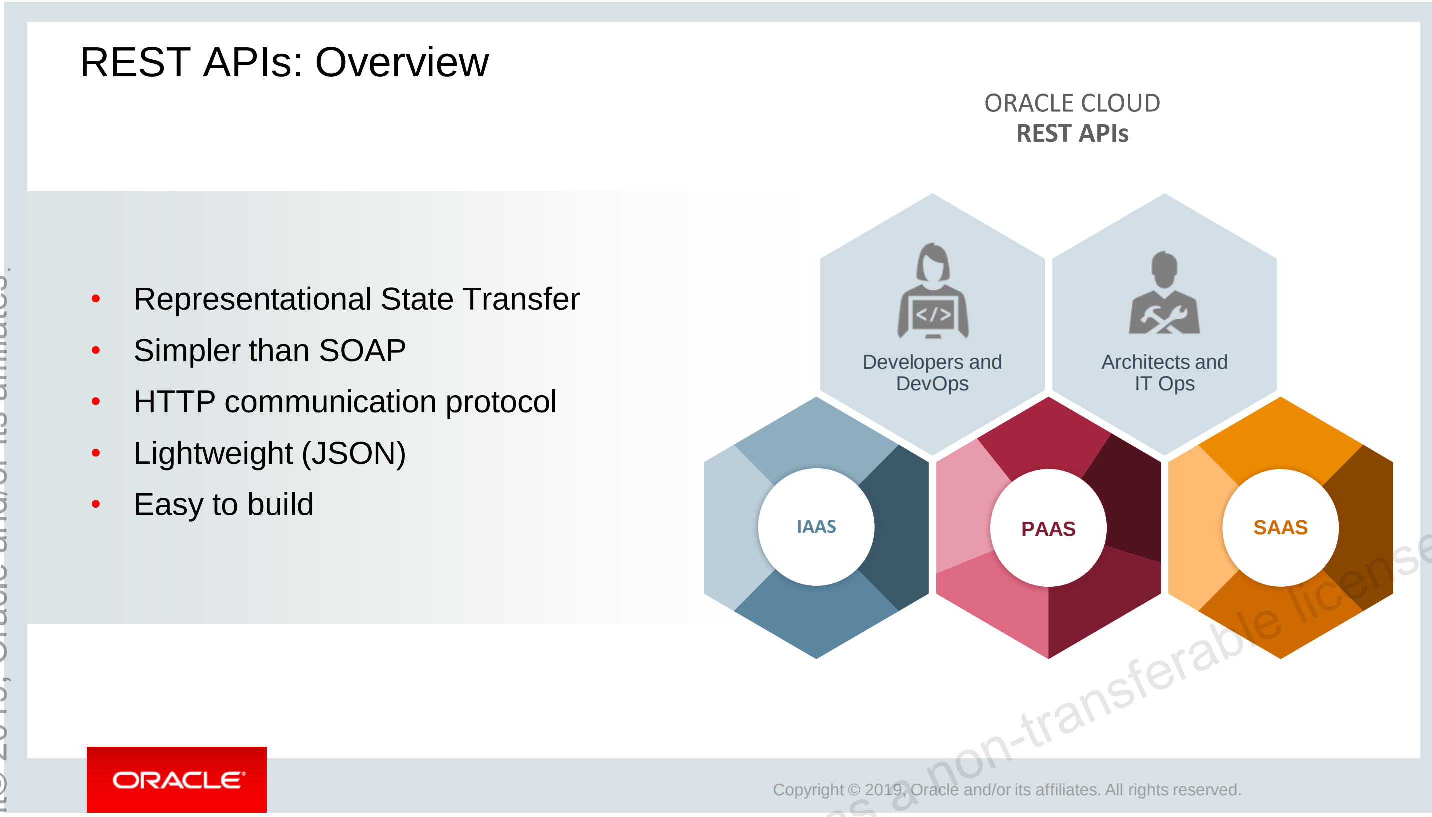
  The micro-service architecture assumes that communication between applications, and general communication is mainly made by using REST APIs.
- **DevOps:** Most of the DevOps operations can be performed by using REST APIs, such as creating new instances and connecting to the cloud environment. REST APIs can be easily consumed by simpler script languages such as shell scripting. For example, for cost efficiency, you can have scripts that shut down the instances of your database cloud every day. (Dev, Test, and UAT environments can usually be shut down after working hours.)
- **Monitoring:** Some REST APIs can be used by monitoring tools to check the status of services and applications, and depending on the status, actions can be triggered by using other REST APIs. For example, if an instance is down during the morning check of your database instances, you can call a REST API to start the instance.

REST APIs

- REST stands for "Representational State Transfer."
- It is an architectural style for distributed hypermedia systems.
- RESTful web services represent a different approach to web services than SOAP-based services.
- The tools (JAX-RS) and best practices related to REST are still evolving.
- REST is designed to limit system behavior to only what is required.

A REST API is a type of web service that is deployed on the Internet and that can be accessed by using the HTTP protocol.

Roy Fielding was one of the authors of the HTTP specification and a co-founder of the Apache HTTP Server project. The web is the largest implementation of the REST architecture. By applying RESTful design to web services, you are, therefore, attempting to mimic the architecture of a proven Internet-scale, distributed hypermedia system.



As mentioned earlier, RESTful services are simpler than traditional XML SOAP communications.

REST APIs also use the HTTP communication protocol but mainly use JSON (which is human readable) for communication (as payloads).

Moreover, JSON is easy to build and easy to read.

Using Oracle Cloud Infrastructure REST APIs

Example: Display information about an Exadata Cloud Service instance (DB System)

Request Method | Resource URL | Request Signature

```
$ curl -X GET -sS 'https://database.uk-london-1.oraclecloud.com/20160918/dbSystems/ocid1.dbsystem.oc1
  .uk-london-1.abwgiljsbydgbr2ecxa74zeqokts6yku2h2rlnbujm4cprn75mrc6r5m5knq' -H "date: Thu, 31 Jan 20
  19 05:44:00 GMT" -H 'Authorization: Signature version="1",keyId="ocid1.tenancy.oc1..aaaaaaaab72ycc6
  5ujcqftrkbfizd4bqddd2o4y6d25kgk46txrjt6phhgyq/ocid1.user.oc1..aaaaaaaa7tpw4a5sqpeunha6oucd2wqns7kco
  zeonlu6uhndimx6w33macwq/fd:ab:93:97:2c:ac:7d:24:38:a4:1a:c5:eb:eb:bb:e9",algorithm="rsa-sha256",hea
  ders="(request-target) date host",signature="FOxM5uFyDh6UOG97ZMvs3Y/0X39xmrNeqrxHy1ifjjTvlMAvuFuNap
  w3mWXg9fdFHsHAZT+iFLF3BklsRNZg8ZzZbdwkftN7FBcqsN/URxgNg27P+gVkeN5mQEE0ZzgqaCnMLNHiKOxKU8W7lvbDiBeRJ
  sYRWkvdRs86c41CanYK+zrvVOVJkapmGaVYP+AzoNPuwo075sAhc1c4PGYncc48Mguo0nEsrXrXqcTgPdeu3V2rMBD6R1D0yh81
  OYwSQA4YiA5xCUA5gC8huMCUuECmx8z9VhYpdQZVKekFKGKavk/7RsW6V9Ohfwgfei39bHlkMuI/eH9LwrSLzaf3zw=="'
{
  "availabilityDomain" : "uwRT:UK-LONDON-1-AD-1",
  "backupSubnetId" : "ocid1.subnet.oc1.uk-london-1.aaaaaaaailf7uwp3exr6nk7ex5aqpp7pekp4wpvunrpvsmfaezhqvuudd6ga",
  "clusterName" : "Cluster1",
  "compartmentId" : "ocid1.compartment.oc1..aaaaaaaaevi5ite3fdznhelejrelqbg2oegcfzkiepvio6o5robcafbwd2xa",
  "cpuCoreCount" : 16,
...
  "vipIds" : [ "ocid1.privateip.oc1.uk-london-1.abwgiljsljhbsedjuwxcwhq4mgrxeg4qxnu62fckr5atay5pwncrzkwwndpq",
  "ocid1.privateip.oc1.uk-london-1.abwgiljscmusbso4wlrc3vuwe3omdeyreorwvmodusqzmxn2mtodndhxdjsq" ]
}
```

ORACLE

The example in the slide shows an Oracle Cloud Infrastructure (OCI) REST call, which is run using the cURL utility. The example makes a REST call that displays information about a specific Exadata Cloud Service instance. From an OCI perspective, an Exadata Cloud Service instance is also known as a DB System. In this example, we use an Exadata DB System, but it can also be any other DB System in OCI.

In the example cURL command, the request method is `GET`. A `GET` request typically returns information about a resource.

The resource URL identifies the resource that is the subject of the request. It contains two main parts:

- The first part of the resource URL (`https://database.uk-london-1.oraclecloud.com`) specifies the REST API endpoint. In this case, the endpoint in use is the OCI Database API endpoint for the `uk-london-1` region, which contains the Exadata Cloud Service instance.
- The next part (`/20160918/dbSystems/ocid1.dbsystem.oc1.uk-london-1.abwgiljsbydgbr2ecxa74zeqokts6yku2h2rlnbujm4cprn75mrc6r5m5knq`) specifies the API path. In this example, the path identifies the API version (`20160918`) and the OCID for the service instance being interrogated (`ocid1.dbsystem.oc1.uk-london-1.abwgiljsbydgbr2ecxa74zeqokts6yku2h2rlnbujm4cprn75mrc6r5m5knq`).

The request signature contains information about the tenancy, user, and RSA key, which are all used to authenticate the request. Note that the signature information is only valid for a short period of time. So, in practice, it must be regenerated for every REST call. For detailed information and sample code relating to request signatures, see https://docs.cloud.oracle.com/iaas/Content/API/Concepts/signingrequests.htm.

The complexity of the request signature means that users are unlikely to make REST calls outside of REST orchestration environments, which contain facilities to programmatically generate the request signature. Additionally, Oracle provides the `oci` Command Line Interface (CLI), which provides a simple interface to the OCI REST APIs. The `oci` CLI is detailed later in the lesson.

OCI Command Line Interface

The **oci** Command Line Interface:

- Is easy to use and lightweight to install
- Runs on Mac, Windows, or Linux
  - Uses Python (version 2.7.5, or 3.5, or later)
- Provides the same core functionality as the Web console
  - Calls OCI REST APIs
- Supports the following OCI services:
  - Announcements
  - Audit
  - Container Engine for Kubernetes
  - Core Services (including Networking, Compute, Block Volume)
  - Database
  - DNS
  - Email Delivery
  - File Storage
  - IAM
  - Key Management
  - Load Balancing
  - Object Storage
  - Search

ORACLE

The `oci` Command Line Interface (CLI) is an easy to use, small footprint tool that you can use on its own or in conjunction with the OCI console to complete OCI monitoring and administration tasks. The CLI provides the same core functionality as the Web console. It also contains some additional commands, such as the ability to run scripts, which extend the functionality of the OCI console.

The CLI is built on Python (version 2.7.5, or 3.5, or later), running on Mac, Windows, or Linux. The Python code makes calls to Oracle Cloud Infrastructure REST APIs to provide the functionality for the various services.

The list in the slide outlines the services supported by the CLI.

For more details, see the website at:

https://docs.cloud.oracle.com/iaas/Content/API/Concepts/cliconcepts.htm

Installing and Configuring OCI CLI

Using the installer script and the setup command is the fastest way to get started with OCI CLI.

Tasks involved in getting started with OCI CLI:

1. Address the requirements and install the CLI. It's supported on Windows, MacOS, Linux, and UNIX.
2. Provide necessary details to the installation script prompts.
3. After successful installation, set up the Config File.

ORACLE

Note: Oracle recommends that you avoid using string values that include confidential information while working with OCI CLI.

The installer script automatically installs the CLI and its dependencies, Python and virtualenv. Before running the installer, ensure that you address the requirements.

For information about installing OCI CLI:

- https://docs.cloud.oracle.com/iaas/Content/API/SDKDocs/cliinstall.htm

For the list of requirements, refer to:

- https://docs.cloud.oracle.com/iaas/Content/API/Concepts/cliconcepts.htm#Requirements

Configuring OCI CLI Config File

- After successful installation, set up the Config File by using the **`oci setup config`** command.
- You will be prompted to enter the following when you run the setup command:
  - Location of the config file
  - User OCID
  - Tenancy OCID
  - OCI Region
  - Location of your private key file
- You can use the OCI Web Console to gather some of this information and keep it handy.

ORACLE

You will configure OCI CLI as part of the practice session.

Getting Started with OCI Command Line Interface

Quickstart configuration:

1. Download and install:

```
$ bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh)"
```

   – Follow the prompts to specify installation directories and configure environment settings.

2. Configure:

```
$ oci setup config
```

   – Follow the prompts to specify configuration attributes: user OCID, tenancy OCID, and OCI region.
   – Specify an RSA key pair; generate new keys or use existing keys.

3. Upload the API key:
   – Register the RSA public key as an API key in the OCI console.

See also Command Line Interface (CLI) in *Oracle Cloud Infrastructure Documentation*

ORACLE

You must download, install, and configure the `oci` command line interface (CLI) before you can use it. The slide outlines the quickstart configuration process. This is a one-time installation and configuration process that most users will use. Note the following details:

1. **Download and install**: Use the command shown in the slide to download and install the `oci` CLI and required software packages, including Python. During installation, you are prompted to specify where you would like to install the `oci` binaries. You are also given the option to update your `$PATH` environment setting and enable shell/tab completion.
2. **Configure**: Use the command shown in the slide to configure the `oci` CLI. During configuration, you are prompted to specify the location where the configuration file is stored. You are also prompted to supply the user OCID, tenant OCID, and region that are associated with your Exadata Cloud Service instance. These values can be determined by examining the OCI Web console. Finally, you are prompted to specify an RSA key pair to use for request authentication. You can allow the configuration process to generate a new key pair, or you can specify an existing key if you have one.
3. **Upload API key**: In the OCI Web console, you must register the RSA public key that is associated with your `oci` CLI configuration. You must register the public key by using the API Keys section on the OCI console User Details page.

For further details and information about alternative manual configuration options, see Command Line Interface (CLI) in *Oracle Cloud Infrastructure Documentation*.

Getting Started with OCI CLI: Considerations

- After the setup is complete, you can get started with OCI CLI.
- The setup is specific to your OCI user account and the details you provide during the setup.
- You need to reconfigure the Config File by using the **`oci setup config`** command if you need to change any of the details you provided during the setup.
- Setup for a given set of details is a one-time activity, that is, you don't have to run the setup if the details remains the same.

You can get help for any CLI command using the following and get started:

```
oci --help
```

Command Line Syntax

- The basic command line syntax is:

  `oci <service> <type> <action> <options>`

- The following command will list the Autonomous Database instances in a given compartment:

```
$ oci db autonomous-database list -c
ocid1.compartment.oc1..aaaanuielrxczjfoani77wb7fpzbjbbjqp73b45odtr4
kfa
```

- In this example, the syntax is applied as follows:
  - **db** is the <service>.
  - **db autonomous-database** is the resource <type>.
  - **list** is the <action>.
  - Rest of the command string consists of <options>; in this case, the compartment OCID.

ORACLE

Most commands must specify a service, followed by a resource type and then an action.

We have common set of OCI CLI syntax for Oracle Autonomous Database, and they work for both ADW and ATP instances.

For information about available OCI CLI commands for the Database Service, see:

- https://docs.cloud.oracle.com/iaas/tools/oci-cli/latest/oci_cli_docs/cmdref/db.html

You will execute some of these commands in the practice session.

Using Terraform to Manage Autonomous Databases

- Terraform enables you to safely and predictably create, change, and improve infrastructure.
- Terraform is "infrastructure-as-code" software that allows to define infrastructure resources in the files that can be persisted, versioned, and shared.
- Files describe the steps required to provision infrastructure and maintain its desired state.
- Terraform's configuration and execution building blocks are modules, which are self-contained configuration packages.
- To use Terraform for OCI, download Terraform from HashiCorp, and then download the OCI Terraform provider.

Terraform: Overview

Terraform is "infrastructure-as-code" software that allows you to define your infrastructure resources in files that you can persist, version, and share. These files describe the steps required to provision your infrastructure and maintain its desired state; it then executes these steps and builds out the described infrastructure.

Terraform's configuration and execution building blocks are modules, which are self-contained configuration packages. You can use these modules to organize your code and to create reusable components. HashiCorp, the developer of Terraform, provides a library of open-source Terraform modules "out of the box" to support many common tasks.

To use Terraform for Oracle Cloud Infrastructure, you must download two components: Terraform from HashiCorp, and then the Oracle Cloud Infrastructure Terraform provider.

Terraform is an open-source orchestration tool that codifies APIs into declarative configuration files that can be shared amongst team members, treated as code, edited, and reviewed,

Summary

In this lesson, you should have learned to:

- Describe the Oracle Cloud Infrastructure REST APIs
- Describe Oracle Cloud Infrastructure - Command Line Interface (OCI CLI)
- Explain Command Line Syntax
- Install and configure OCI CLI
- Manage autonomous databases by using OCI CLI
- Use Terraform to manage autonomous databases

ORACLE

Practice 14: Overview

- Practice 14-1: Configure OCI CLI
- Practice 14-2: Managing Autonomous Database Instance using OCI CLI

15

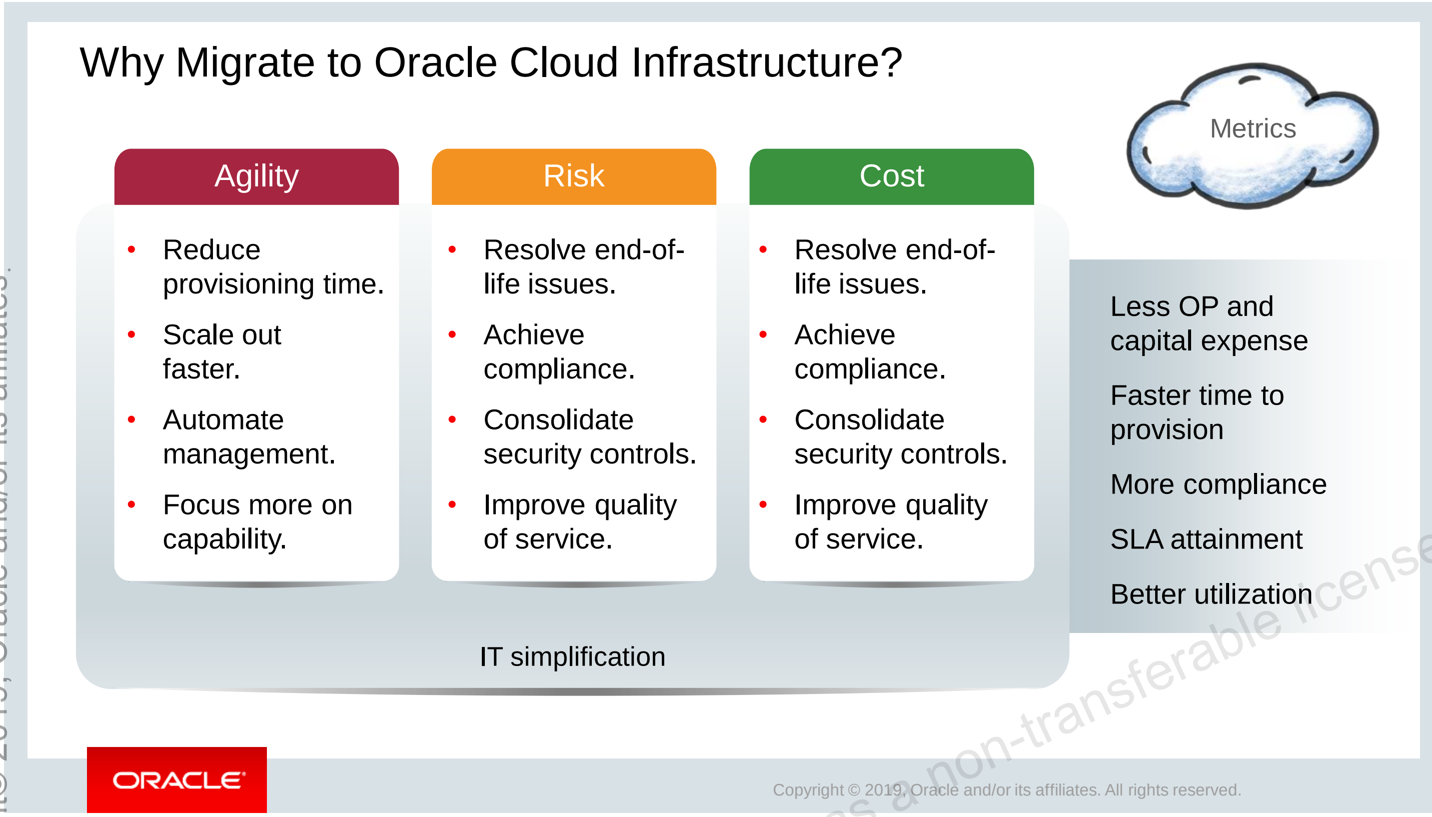
Migrating Oracle Databases to Autonomous Database: Overview

ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe the benefits of migrating to Oracle Cloud
- Explain database management in the cloud as opposed to on-premises
- Identify what can be migrated
- Getting started with Cloud Database migration
- Explain the available migration methods

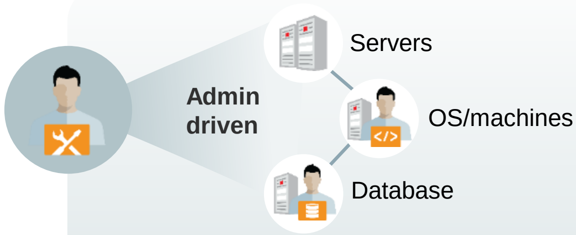


This slide shows the benefits of using Oracle Cloud Service in terms of:

- Lowering operational costs
- Lowering capital expenditure
- Reducing time to provision
- Maintaining service-level agreements by adhering to the **compliance** and **utilization** metrics

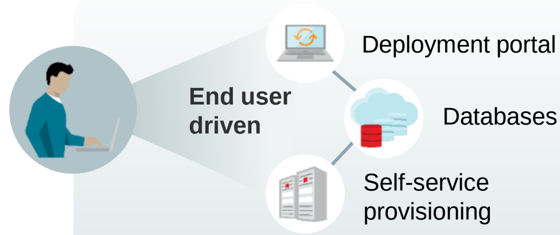
Managing Oracle Database: On-Premises Versus Oracle Cloud Infrastructure

Traditional Database Deployment



- Specify and procure hardware.
- Configure hardware.
- Deploy hardware.
- Add hardware and reconfigure stack as demand grows.
- Deploy database.

OCI DB Systems



- Request database deployment via the cloud.
- Adjust capacity as demand changes.
- Retire database when not needed.

ORACLE

Traditional IT operations are very administration-driven, customized for the application environment, and slow due to the transitions between different administrative teams, such as hardware, storage, database, and so on.

Managing Oracle Database: On-Premises Versus Cloud

Operation type	On-Premises Database	Cloud Database
Backing up of database and system configuration files	Database files + database control files + database SPFILE	All database files + SPFILE + password files and others from `/home/oracle/bkup/dbcfg.spec` and `/home/oracle/bkup/oscfg.spec` files
Database backup	Local storage or cloud	• Local compute node storage • Oracle Storage Cloud Service container • Both Cloud Storage and Local Storage
Scheduling of database backups	Manual scheduling using RMAN> `backup`	Automatic or Manual Scheduling: `bkup_api`
Database installation	Manual • Oracle Database 11g, 12c or 18c • Database creation	Automatic • Oracle Database 11g ,12c or 18c • Pre-created database
Location for database files and backups	Manual	Automatic
Housekeeping of database logs and diagnostics files	Manual	Automatic, using a configuration file or Manual
Tools used for database monitoring	EM Express, EM Cloud Control, SQL Developer	EM Express, Oracle Management Cloud, EM Cloud Control, SQL Developer, SQL Developer Web
Oracle Database Architecture	Non-CDBs and CDBs	Only CDBs from 12c onward
Patch compliance	• None • Oracle Support • EM Cloud Control	• GUI Tool: Oracle Database Cloud Service console • `dbaascli dbpatchm`

ORACLE

The table in the slide lists the main differences (based on type of DBA operations) between on-premises databases and cloud database deployments.

Managing Oracle Database: On-Premises Versus Cloud

Operation Type	On-Premises Database	Cloud Database
Database recovery	Manual: Using RMAN> `recover`	Automated using Database Service console or `dbaascli orec`
Storage allocation	Manual: Using UNIX commands	GUI tool: Oracle Database Cloud Service console
Tablespace encryption	None by default	Default encryption for user-defined tablespaces: Initialization parameter `encrypt_new_tablespaces = cloud_only`
Types of server connection	All types (password, SSH ...)	SSH – Pair Key based
Database user and group	`oracle` user and `oinstall` group	`oracle` and `opc` users, and `oinstall` group

ORACLE

What Can You Migrate to Oracle Database Cloud?

- Tables or partitioned tables
- Schemas
- PL/SQL objects
- Tablespaces
- Noncontainer databases
- Container database (CDB)
- Pluggable databases (PDBs)

ORACLE

Note: All the on-premises database objects listed in the slide (output of the query `select distinct object_type from dba_objects`) can be migrated to the cloud. A few commonly used ones will be discussed in detail in subsequent lessons.

Considerations for Choosing a Migration Method

- 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

ORACLE

Various migration methods exist, and each migration method has different benefits, opportunities, requirements, and limitations.

For example, Oracle Cloud Service uses a little-endian platform, so if you are migrating from a big-endian platform, some physical migration approaches are not feasible, or require extra processing to achieve. Also, the use of specific database features, such as materialized views or object data types, may impose restrictions on some migration methods.

For more details on migrating to OCI Database, refer to Oracle University training ***Migrating Your Oracle Database to Oracle Cloud Infrastructure.***

Migration: Information Gathering

- 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

ORACLE

Determining Applicable Methods

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

- Database version of your source database:
  - Oracle Database 11*g* Release 2 version lower than 11.2.0.3
  - Oracle Database 11*g* Release 2 version 11.2.0.3 or higher
  - Oracle Database 12*c* Release 1 version lower than 12.1.0.2
  - Oracle Database 12*c* Release 1 version 12.1.0.2 or higher
  - Oracle Database 12*c* Release 2 version 12.2.0.1
- For Oracle Database 12*c* Release 1/ Oracle Database 12*c* Release 2 source databases, the architecture of the database:
  - Multitenant container database (CDB)
  - Non-CDB
- Your source database host platform and endian format:
  - Query `V$DATABASE` to identify the platform name for your source database. Platforms are either little endian or big endian depending on the byte ordering that they use.
  - Query `V$TRANSPORTABLE_PLATFORM` to view all platforms that support cross-platform tablespace transport, along with the endian format of each platform.
  - The Oracle Cloud Infrastructure Database uses the Linux platform, which is little endian.

The database character set of your source database and the Oracle Cloud Infrastructure Database database:

- By default, databases are configured to use the AL32UTF8 database character set on Oracle Cloud Service. You can select the required character set during provisioning.
- The target database version to which you are migrating on Oracle Cloud Service:
  - Oracle Database 11*g* Release 2
  - Oracle Database 12*c* Release 1
  - Oracle Database 12*c* Release 2
  - Oracle Database 18c

Migration: Analysis and Planning

- DOWN TIME
- DATA SIZE
- NETWORK
- SIZING
- ENCRYPTION
- VERSION AND EDITION
- PATCHES
- CROSS PLATFORM

ORACLE

- **Down time:** Determine from your business what the downtime service level agreements (SLAs) are and how much down time, if any, the business can accommodate. You can also review Recovery Time Objective (RTO) and Recovery Point Objective (RPO) SLAs to see how much down time is acceptable according to your disaster recovery (DR) and business continuity (BC) guidelines.
- **Database Size:** Determine the data volume. Typically, the size of the database is based on two factors: whether the physical or logical migration method is considered, and whether all or part of the data will be migrated to the target database.
- **Network Bandwidth:** Determine the available network bandwidth between the source and target databases. In addition to available bandwidth, network reliability is also important. Based on the data transfer method, network interruption might require you to restart the data transfer job.
- **Cross-Platform Migration:** Determine the endianness of the source and target platforms. Oracle Cloud Infrastructure databases are little-endian. If your source database is big-endian, you can either select the logical migration method, which is typically slower, or use Oracle Data Guard or RMAN cross-platform features for the cross-platform migrations.
- **Database Character Set:** Determine the database character set for the source and target databases. For most migration methods, the target database character set must be a superset of the source database character set. Some methods might need the exact same character set to avoid data loss.
- **Data Encryption:** Determine whether the source database uses Transparent Data Encryption (TDE). TDE is mandatory for all Oracle Cloud Infrastructure databases. If TDE is not used at the source, enable it either at the source or at the target. Be sure to back up and restore the required TDE wallets from the source to the target.

- **Database Version, Edition, and Options:** Determine the database version, edition, and options for the source and target databases. Based on the migration method, the target and source database version and edition must be compatible. For the Oracle Cloud Infrastructure 12*c* database target, the multitenant architecture is mandatory, so ensure that the selected migration method can accomplish the migration into the CDB/PDB, as needed.
- **Databases Patches:** Determine the patch level for the source and target databases. Ensure that the source and target are at the same or compatible Patch Set and Release Upgrade level. Apply any required patches at the source to minimize any discrepancies during or after the migration. Also, as necessary, apply any one-off patches at the target.
- **DB Name:** Determine the database name used at the source database. For full database restore methods, it is mandatory to create the target database by using the same database name as used at the source database. However, use the DB Unique Name of the target as created by the Oracle Cloud Infrastructure tooling.
- **DB Block Size:** Determine the database block size used at the source database. For partial restore methods like transportable tablespaces, it might be necessary to adjust the cache size parameters based on the target database.
- **DB Time Zone:** Determine the database time zone used at the source database. It might be necessary to adjust the database time zone at the target database.
- **DB Users, Privileges, and Objects:** Determine the database users, privileges, and objects, like DB Links, from the source database that might also need to be created at the target database.
- **Sizing:** Determine the source database sizing and consider future growth to size the target database. In addition to CPU and Memory, ensure that the sizing meets your IOPS and Network Bandwidth requirements.
- **Target Database:** To ensure the target database will have all the required metadata for OCI tooling to work, create the target database using one of the supported methods like OCI Console, OCI CLI or Terraform OCI provider. This target database will be cleaned to be used as a shell for the migration, as needed.

Migration: Data Transfer Options (Online and Sync)

Transfer Option	Transfer Mode	Options for Copying Data
Public Internet	Online	OCI CLI, OCI API, OCI Console, rclone
IPSec VPN	Online	OCI CLI, OCI API, OCI Console, rclone
FastConnect	Online	OCI CLI, OCI API, OCI Console, rclone
Storage Gateway	Sync	cp/scp to NFS mount points

ORACLE

Based on your data volume, network bandwidth, and network reliability, use one of the following options to upload the backups to Oracle Cloud Infrastructure (OCI) Object Storage.

Public Internet Online OCI CLI, OCI API, OCI Console, rclone

IPSec VPN Online OCI CLI, OCI API, OCI Console, rclone

https://docs.cloud.oracle.com/iaas/Content/Network/Tasks/managingIPsec.htm

FastConnect Online OCI CLI, OCI API, OCI Console, rclone

https://docs.cloud.oracle.com/iaas/Content/Network/Concepts/fastconnect.htm

Storage Gateway Sync cp/scp to NFS mount points

https://docs.cloud.oracle.com/iaas/Content/StorageGateway/Concepts/storagegatewayoverview.htm

Migration: Data Transfer Options (Offline)

Data Transfer Disk

- Send your data to an Oracle Data Transfer site (US or Frankfurt).
- Oracle will upload the data for you over fast network connections.
- Data is wiped off the disks and shipped back after data is uploaded to Oracle Cloud.

Data Transfer Appliance

- Rent a Data Transfer Appliance from Oracle to migrate PB scale datasets to Oracle Cloud.
- Each Transfer Appliance can migrate up to 150 TBs.
- Use multiple appliances to migrate large datasets.
- Keep the Transfer Appliance onsite for up to 30 days.
- It is available for use in US and European Union countries.

ORACLE

For more information, refer to:

- https://docs.cloud.oracle.com/iaas/Content/DataTransfer/Concepts/overview.htm

Migration: Security Considerations

- Transparent Data Encryption (TDE) is mandatory for all OCI databases.
- Enable it on the source or target database.
- Make sure you back up and restore your TDE wallets from the source to the target.

Migration Options

Migration Option	Autonomous	BM/VM/Exadata	Database Versions	Benefits
Restore from Object Store (Online or sync data transfer)	✓ Data Pump	✓ RMAN based Restore	All	Supports all database editions and platforms along with full and incremental backups
Restore from Object Store (Offline data transfer using Data Transfer Service)	✓	✓	All	Same as Restore from Object Store. Additionally, supports low bandwidth/high data volume scenarios
Data Guard	NA	✓	Enterprise Edition and above	Minimal down time
Oracle SQL Developer	✓	NA	All	Quick and simple; suitable for migrating database objects of small to medium size

ORACLE

Apart from the options listed in the slide, there are a bunch of standard Oracle options available for migration, which are discussed in the Oracle University training *Migrating Your Oracle Database to Oracle Cloud Infrastructure.*

You can also refer to:

- https://docs.cloud.oracle.com/iaas/Content/Database/Tasks/migrating.htm
- https://www.oracle.com/technetwork/database/bi-datawarehousing/adw-technical-faq-public-5069016.pdf

Replicate Data to Autonomous Database Using GoldenGate

- You can configure the Autonomous Database instance as a **target database** for Oracle GoldenGate On Premises.
- The source for replicating to Autonomous Databases can be:
  - Oracle GoldenGate On Premises releases 12.3.0.1.2 and later
  - Oracle Database Cloud Service on Oracle Cloud and Oracle Cloud at Customer
  - Oracle Exadata Cloud Service on Oracle Cloud and Oracle Cloud at Customer
- You cannot set up Oracle Autonomous Database as a source database for Oracle GoldenGate Cloud Service.

ORACLE

You can configure an Oracle Autonomous Database instance as a **target database** for Oracle GoldenGate Cloud Service.

Autonomous Database supports all standard SQL and business intelligence tools and deliver scalable analytic query performance. Autonomous Database provides all of the performance of the market-leading Oracle Database in a fully-managed environment that is tuned and optimized. Oracle GoldenGate Cloud Service captures from on-premises Oracle Databases and sends data to a GoldenGate Cloud Service instance, and vice versa is not possible because you cannot set up Oracle Autonomous Database as a source database for Oracle GoldenGate Cloud Service.

For more information about the replication, refer to:

- https://docs.oracle.com/goldengate/c1230/gg-winux/GGODB/replicating-data-oracle-autonomous-data-warehouse-cloud.htm#GGODB-GUID-660E754E-B9A6-48DD-AA66-0D6B66A022CD

Summary

In this lesson, you should have learned how to:

- Describe the benefits of migrating to Oracle Cloud
- Explain database management in the Cloud as opposed to on-premises
- Identify what can be migrated
- Getting started with Cloud Database migration
- Explain the available migration methods

ORACLE

Practice 15: Overview

There are no practices for this lesson.

16

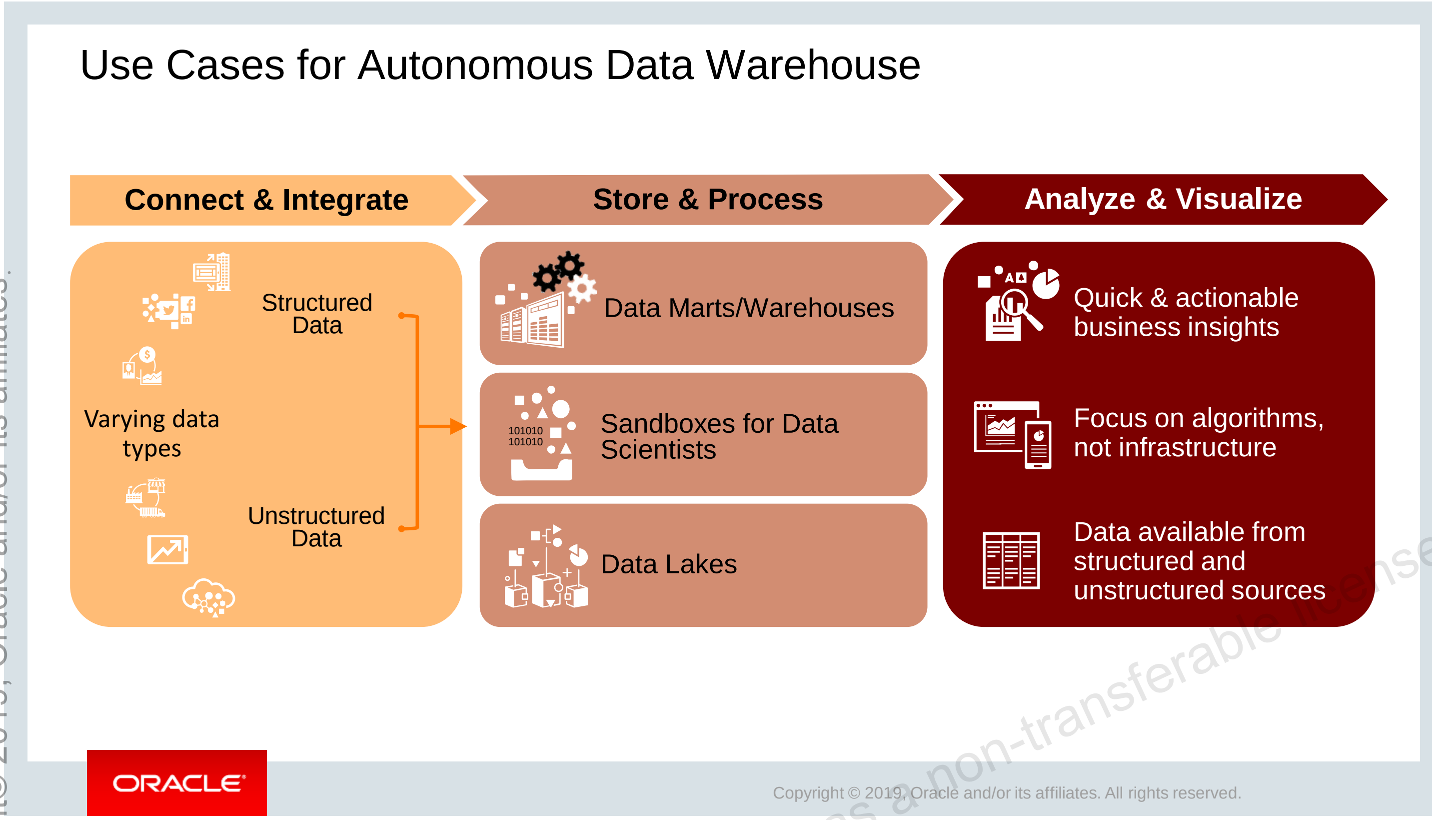
Oracle Autonomous Database: Use Cases

ORACLE

Objectives

- Use Cases

Use Cases



There are three key use cases that you see many of your customers engaging or wanting to engage in:

New Analytical Data Marts and/or Data Warehouse

Benefits:

- Enable business teams to get answers quickly
- Fast query performance
- Integration with leading business-analytics tools

Example:

In the retail industry, there is a need to explore multiple sets of data to determine the cause of sudden drop in gross margin despite increasing sales. The CFO needs insights within one week and the analyst can't afford to wait the usual two or three months to sign off and get a new DW/mart set up and loaded with data.

With ADWC, you can set up a new data mart from scratch (that is, signing up a new account and provisioning DW) within few minutes. Use Oracle DIPC to load varying data types into ADWC (built-in connectors). A tuned, ready-to-go, out-of-the-box data mart built on the Exadata platform ensures data can be accessed and analytical queries run significantly faster – up to 100x than before.

Using OAC can run analytics report that tell meaningful stories in a clear presentable visual format that's highly customizable for the user, thus enabling business leaders to get rapid insights and facilitates their ability to make smarter decisions.

Sandboxes for Data Scientists

- Benefits:
  - Enables data scientists to focus on algorithms, not infrastructure
  - In-database machine-learning models
  - Built-in notebook

Data Lakes

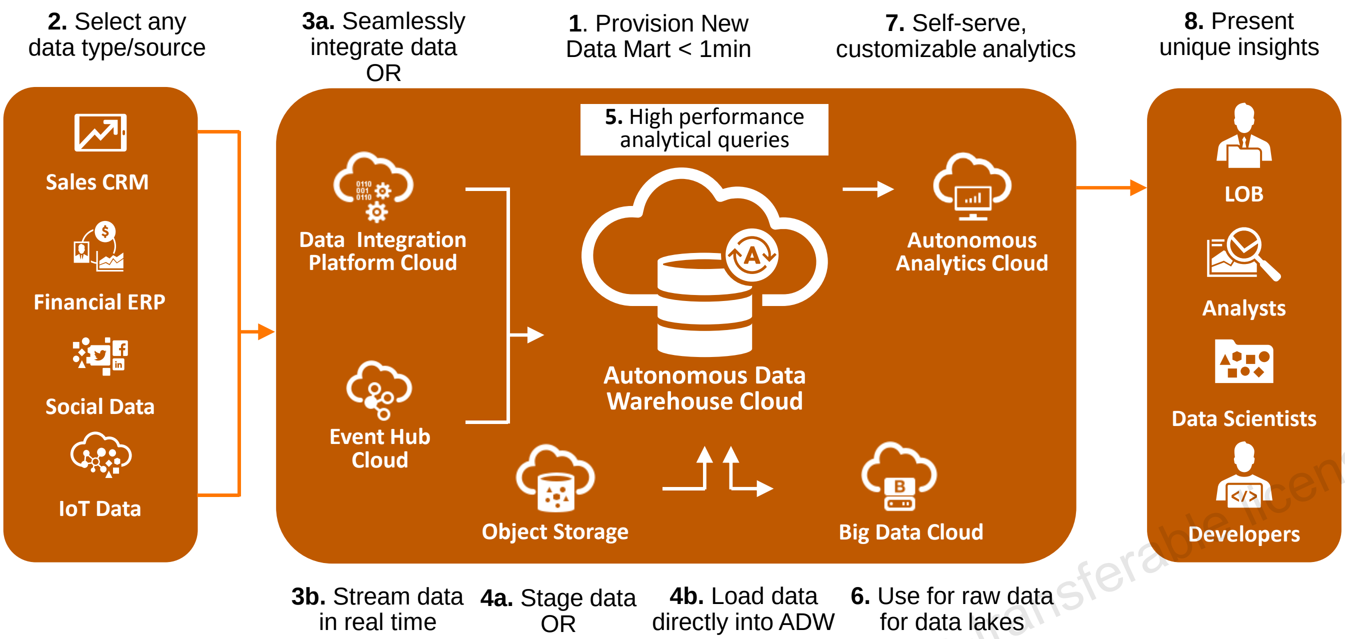
- Benefits:
  - Makes broadest range of data available to all analysts
  - Fast SQL queries over cloud object stores
  - Seamlessly combine data from data marts, data warehouses, and lakes

Example:

Think of the data lake as complementary to your data warehouse. There are many business problems that require data beyond what is typically stored in your data warehouse. With predictive maintenance, for example, we would need raw sensor data (stored in the data lake) to complement official maintenance and purchase records (stored in the data warehouse). When trying to determine the next best action for a given customer, we would want to work with both customer purchase records (in the data warehouse) and customer web browsing or social media usage (details of which would most likely be stored in the data lake). Having more data to work with enables new or more accurate insights and is essential for many use cases in all industries from manufacturing (e.g. yield optimization) to healthcare (personalized treatment), from financial services (fraud detection) to the public sector (smart cities and infrastructure).

An object storage–based data lake is the cheapest, most cost-effective place for storing this kind of multi-structured data. In the data lake, it can be processed, transformed, or used as input to machine learning algorithms (using Big Data Cloud). Intermediate or final results can be stored back in the lake and can be queried by, or loaded into, the data warehouse as needed.

New Oracle Analytical Data Mart



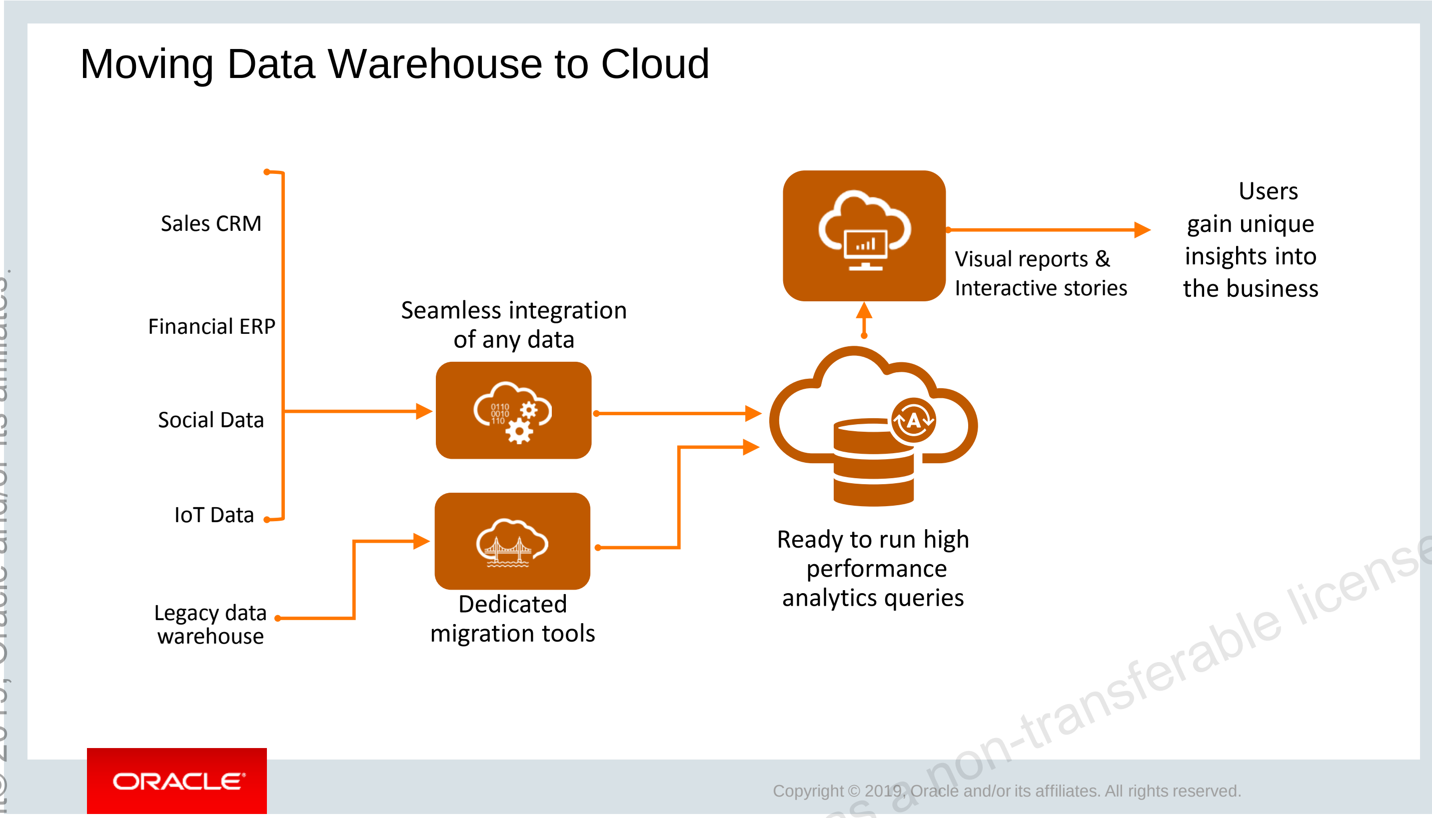
Create New Analytical Data Marts to Meet Your Business Needs

Business Analyst, Data Scientist

- Get results fast, plus gain unique insights into the business for smarter decisions
- Explore any data you want, when you want to – meet demanding goals
- Deliver visual reports that are easy-to-consume and help uncover valuable predictions and insights to meet demanding business goals
- Deliver actionable insights within a few hours; ADWC + AAC with embedded ML provides unique new insights & trends that help accelerate time to market

IT Director, DW Architect, DBA

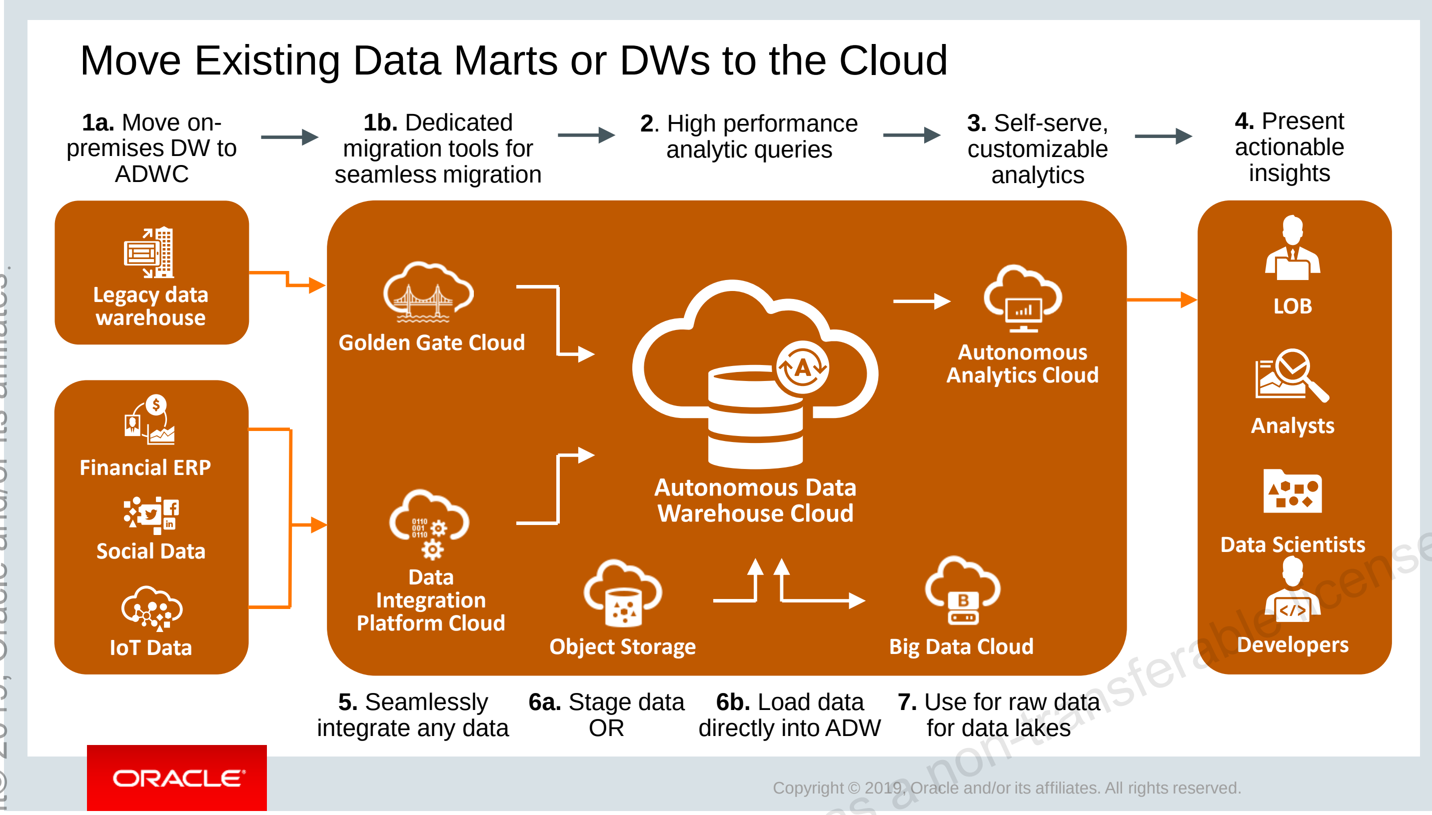
- Provide a platform that supports high concurrency whilst delivering fast analytical queries
- Reduce risk with a data warehouse that is self-securing, detecting, and applying security patches on its own
- Allow users to create a data mart and start running queries within minutes
- Support high concurrent user access allowing users to seamlessly access and process any data from anywhere



The first step is to migrate the existing on-premises data warehouse to Oracle ADWC. Using dedicated migration tools, such as Oracle Golden Gate, you can easily and rapidly migrate the data warehouse to the Oracle Cloud platform with zero down time. Supporting tools and technologies, such as Datapump, PDBs and transportable table spaces, enable you to simplify your data moving to cloud.

Thereafter, you can continue the DIPC services, which enable you to integrate any data source as needed – be it structured operations data (for example, Sales CRM and Financial ERP) or unstructured data (such as Social and IoT data) – all easily ingested into ADWC ready to run high-performance analytics for the next business data analysis project.

Using Oracle Analytics Cloud, you can extract and convert the workloads into meaningful highly presentable reports that provide unique insights into the business.



Move Existing Data Marts or DWs to the Cloud
1a. Move on-premises DW to ADWC
1b. Dedicated migration tools for seamless migration
2. High performance analytic queries
3. Self-serve, customizable analytics
4. Present actionable insights
Legacy data warehouse
Financial ERP
Social Data
IoT Data
Golden Gate Cloud
Autonomous Data Warehouse Cloud
Autonomous Analytics Cloud
Data Integration Platform Cloud
Object Storage
Big Data Cloud
LOB
Analysts
Data Scientists
Developers
5. Seamlessly integrate any data
6a. Stage data OR
6b. Load data directly into ADW
7. Use for raw data for data lakes
ORACLE
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Easy Migration to Autonomous Data Warehouse

Migration of existing Oracle systems

Data pump import using ADW-aware settings:

- It ensures compatible data import.
- Violating data structures will be ignored or converted.

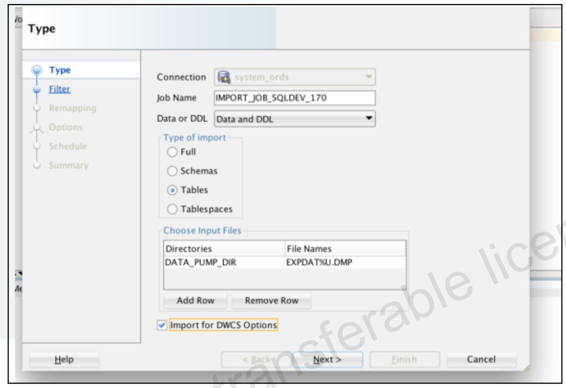
Migration of Redshift systems

End-to-end migration using migration workbench:

- Convert and create data warehouse schema.
- Unload Redshift data into S3.
- Load data into ADW.

Migration of other systems

- Migration workbench assistance or manual



ORACLE

Full Support of Data Warehouse Ecosystem

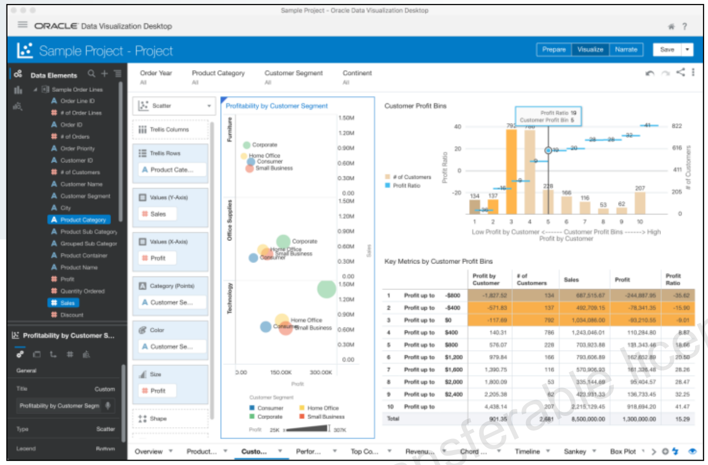
Autonomous Data Warehouse Cloud supports:

- Existing tools, running on-premises or in the cloud
  - Third-party BI tools
  - Third-party data-integration tools
  - Oracle BI and data-integration tools: BIEE, ODI, and so on
- Oracle Cloud services: Analytics Cloud Service, Golden Gate Cloud Service, Integration Cloud Service, and others
- Connectivity via SQL*Net, JDBC, ODBC

ORACLE

Oracle Data Visualization Desktop

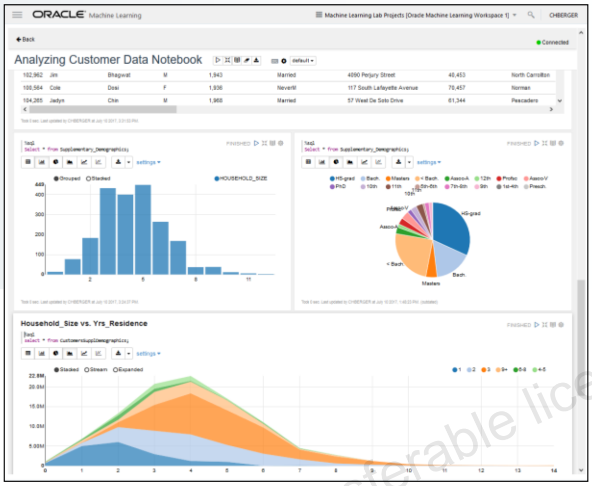
- Self-service data exploration for business users
- Rich, interactive visualizations
  - Included with Autonomous Data Warehouse
  - Also available with Oracle Analytics Cloud



ORACLE

OracleML: Built-in Notebook

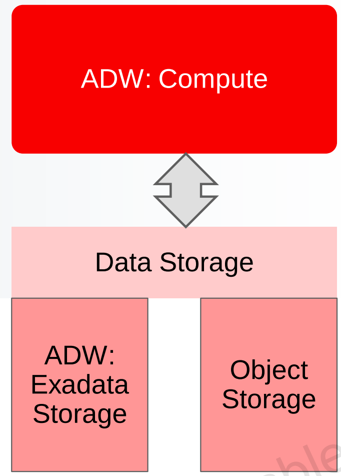
- Collaborative UI for data scientists
  - Easy access to shared notebooks, templates, permissions, scheduler, and so on
- Based on Apache Zeppelin
- Common UI for data scientists across multiple services



ORACLE

Seamless Access to Data Lakes

- ADW can query data within cloud object stores:
  - Oracle Object Store, AWS S3, or Azure Blob Storage
- Extend ADW to access data lakes:
  - High-performance SQL
  - Scalable joins between ADW and data lakes



ORACLE

Practice 16: Overview

- There are no practices for this lesson.

A

The DBMS_CLOUD Package and Autonomous Databases for Experienced Oracle Database Users

ORACLE

Objectives

After completing this appendix, you should be able to:

- Summarize `DBMS_CLOUD` subprograms
- Describe `DBMS_CLOUD` package file URI formats
- Describe `DBMS_CLOUD` package format options
- Explain the restrictions for:
  - Database initialization parameters
  - Database features
  - SQL commands
  - Data types

Summary of DBMS_CLOUD Subprograms

ORACLE AUTONOMOUS DATABASE

ORACLE

Summary of DBMS_CLOUD Subprograms

- The DBMS_CLOUD package is used with Oracle Autonomous Data Warehouse and Autonomous Transaction Processing to load, query, and save data from external data residing in the cloud.
- The DBMS_CLOUD package has several subprograms:
  - `COPY_DATA` procedure
  - `CREATE_CREDENTIAL` procedure
  - `CREATE_EXTERNAL_TABLE` procedure
  - `DELETE_FILE` procedure
  - `DROP_CREDENTIAL` procedure
  - `LIST_FILES` function
  - `PUT_OBJECT` procedure
  - `VALIDATE_EXTERNAL_TABLE` procedure
- The `CREATE_CREDENTIAL` procedure stores Cloud Object Storage credentials in the Autonomous Transaction Processing database. These stored credentials further help to perform load, query, and save data operations.

ORACLE

A credential can be stored in the ATP database by using the following command:

```
BEGIN
DBMS_CLOUD.CREATE_CREDENTIAL(
credential_name => 'DEF_BKP_CRED',
username => 'lab.user01',
password => ':9IvY3u}vh[C0GU;7IVB'
);
END;
/
```

Note

- `credential_name`: This can be any name. In this example, it is `DEF_BKP_CRED`.
- `username`: This is the Oracle Cloud (OCI) account username. In this example, it is `lab.user01`.
- `password`: This is not the password for your Cloud account username. It is the Auth token of the user, which is generated by clicking the user's `settings` → `Auth Tokens`.

The following command will drop the credential from the ATP database:

```
exec dbms_cloud.drop_credential('DEF_BKP_CRED');
```

Summary of DBMS_CLOUD Subprograms

Create an external table using the `DBMS_CLOUD.CREATE_EXTERNAL_TABLE` PL/SQL procedure. This procedure creates an external table in files in the cloud and allows you to run queries on external data from Autonomous Database.

```
begin
 dbms_cloud.create_external_table(
    table_name =>'CHANNELS_EXT',
    credential_name =>'OBJ_STORE_CRED',
file_uri_list =>
'https://swiftobjectstorage.usashburn1.oraclecloud.com
/v1/dwcsdemo/DEMO_DATA/chan_v3.dat',
    format => json_object('ignoremissingcolumns' value
'true', 'removequotes' value 'true'),
    column_list => 'CHANNEL_ID NUMBER,
         CHANNEL_DESC VARCHAR2(20),
         CHANNEL_CLASS VARCHAR2(20),
         CHANNEL_CLASS_ID NUMBER,  CHANNEL_TOTAL
VARCHAR2(13),    CHANNEL_TOTAL_ID NUMBER'
 );
end;
```

ORACLE

Query an External Table

```
BEGIN
DBMS_CLOUD.CREATE_CREDENTIAL(
credential_name => 'EXT_TAB',
username => 'lab.user01',
password => 'Fz+_QZ.wbfiL9U;x6DR;'
);
END;
/
```

Note

- **`password:`** This is not the password for your Cloud account username. It is the Auth token of the user, which is generated by clicking on the user's `settings` → `Auth Tokens`.

```
begin
 dbms_cloud.create_external_table(
    table_name =>'CHANNELS_EXT',
    credential_name =>'EXT_TAB',
    file_uri_list => 'https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/ocuocictrng23/ext_table/chan_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true'),
    column_list => 'CHANNEL_ID NUMBER,
     CHANNEL_DESC VARCHAR2(20),
     CHANNEL_CLASS VARCHAR2(20),
     CHANNEL_CLASS_ID NUMBER,
     CHANNEL_TOTAL VARCHAR2(13),
     CHANNEL_TOTAL_ID NUMBER'
 );
end;
/


select * from CHANNELS_EXT;
```

Summary of DBMS_CLOUD Subprograms

`DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE` validates the source files for an external table, generates log information, and stores the rows that do not match the format options specified for the external table in a *badfile* table on Autonomous Database. Before validating an external table, you need to create the external table.

To create an external table, use the `DBMS_CLOUD.CREATE_EXTERNAL_TABLE` procedure.

- The validate operation, by default, scans all the rows in source files and stops when a row is rejected.
- To validate only a subset of rows, use the `rowcount` parameter.
  - When the `rowcount` parameter is set, the validate operation:
    - Scans rows and stops when a row is rejected
    - Scans rows and stops when the specified number of rows are validated without errors

```
begin
DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE (table_name =>'CHANNELS_EXT');
end;
```

ORACLE

Summary of DBMS_CLOUD Subprograms

To see the result of a validate operation, query the `dba_load_operations` and `user_load_operations` tables:

- `dba_load_operations`: Shows all validate operations
- `user_load_operations`: Shows the validate operations in your schema

```
select * from dba_load_operations;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 1 in 0.231 seconds

	ID	TYPE	SID	SERIAL#	USERNAME	START_TIME	UPDATE_TIME	STATUS	TABLE_NAME	OWNER_NAME	FILE_URI_LIST	ROW...	LOGFILE_TABLE	BADFILE_TABLE	TEMPEXT_TABLE
1	1	VALIDATE	13787	63305	ADMIN	21-AUG-18 01.18.4...	21-AUG-18 01.18...	COMPLETED	CHANNELS_EXT	ADMIN	(null)	5	VALIDATE$1_LOG	VALIDATE$1_BAD	(null)

- The `LOGFILE_TABLE` column shows the name of the table you can query to look at the log of a validate operation.
  - The `BADFILE_TABLE` column shows the name of the table you can query to look at the rows that got errors during validation.
- Correct the error by dropping the external table using the `DROP TABLE` command.

Query the `LOGFILE_TABLE`

```
select * from validate$1_log;

 LOG file opened at 08/21/18 07:48:41

Total Number of Files=1

Data File: https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/ocuocictrng23/ext_table/chan_v3.dat

Log File: VALIDATE$1_204079.log


LOG file opened at 08/21/18 07:48:41

Total Number of Files=1

Data File: https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/ocuocictrng23/ext_table/chan_v3.dat

Log File: VALIDATE$1_204079.log
```

```
 LOG file opened at 08/21/18 07:48:41

Bad File: VALIDATE$1_204079.bad

Field Definitions for table CHANNELS_EXT
  Record format DELIMITED BY
  Data in file has same endianness as the platform
  Rows with all null fields are accepted

  Fields in Data Source:

    CHANNEL_ID                      CHAR (255)
      Terminated by "|"
    CHANNEL_DESC                    CHAR (255)
      Terminated by "|"
    CHANNEL_CLASS                   CHAR (255)
      Terminated by "|"
    CHANNEL_CLASS_ID                CHAR (255)
      Terminated by "|"
    CHANNEL_TOTAL                   CHAR (255)
      Terminated by "|"
    CHANNEL_TOTAL_ID                CHAR (255)
      Terminated by "|"
```

select * from validate$1_bad;

Should display 0 records to get successful validate

Summary of DBMS_CLOUD Subprograms

- `DBMS_CLOUD.LIST_FILES` lists the files and their sizes in the specified directory on Autonomous Transaction Processing. Currently, the specified directory is only `data_pump_dir`, which is a `dbfs` folder. You can add other directories pointing to the `dbfs` folder in the `LIST_FILES` function. Query `dba_directories` for the list off `dbfs` folders.
- To run `DBMS_CLOUD.LIST_FILES` with a user other than `ADMIN`, you need to grant read privileges on the `data_pump_dir` directory to that user:
  - `SELECT * FROM DBMS_CLOUD.LIST_FILES('DATA_PUMP_DIR’);`
- To find the value for `data_pump_dir`:
  - `select * from dba_directories;`
- `DBMS_CLOUD.PUT_OBJECT` copies a file from Autonomous Transaction Processing to the Cloud Object Storage. The maximum file size allowed in this procedure is 5 GB.

ORACLE

List the Files from `DATA_PUMP_DIR`

```
SELECT * FROM DBMS_CLOUD.LIST_FILES('DATA_PUMP_DIR');

Object_name         Bytes
dp.log                    129
VALIDATE$1_dflt.log 0
VALIDATE$1_dflt.bad 0
VALIDATE$1_204079.log      1110


copy the file called as dp.log from ATP to object Storage

begin
DBMS_CLOUD.PUT_OBJECT (
credential_name => 'EXT_TAB',
object_uri => 'https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/ocuocictrng23/ext_table/dp.log',
directory_name => 'DATA_PUMP_DIR',
file_name => 'dp.log');
end;
/
```

DBMS_CLOUD Package File URI Formats

DBMS_CLOUD Package File URI Formats

- **Oracle Cloud Infrastructure Object Storage URI Format**
  - https://swiftobjectstorage.<*region*>.oraclecloud.com/v1/<*tenant*>/<*bucket*>/<*filename*>
    For example, the following is the URI for the `channels.txt` file in the atpc_user bucket in the atpc tenancy of the Phoenix data center:
    - https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/atpc/atpc_user/channels.txt
- **Oracle Cloud Infrastructure Object Storage Classic URI Format**
  - **Service Permanent REST Endpoint URL**
    - This URL contains the auto-generated GUID for the account and remains *constant* for your account.
      - https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365
  - **Service Friendly REST Endpoint URL**
    - If you change the Storage Classic account name, the Service Friendly REST Endpoint URL and the authentication URL will also change. Ensure that you are using the appropriate and latest URL:
      - https://acme.storage.oraclecloud.com/v1/Storage-acme

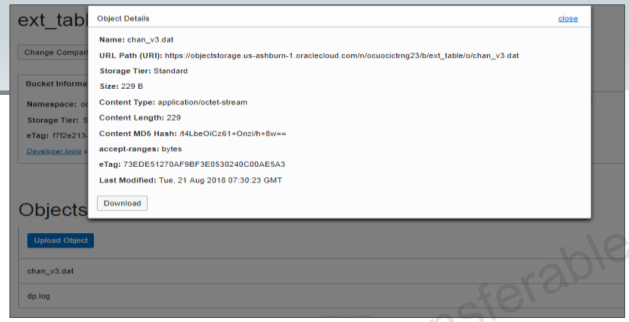
ORACLE

DBMS_CLOUD Package File URI Formats

- The REST API can be accessed from any application or programming platform that correctly and completely understands the Hypertext Transfer Protocol (HTTP) and has Internet connectivity. The REST API uses advanced facets of HTTP, such as secure communication over HTTPS, HTTP headers, and specialized HTTP verbs (PUT, DELETE).
- Accounts, containers, and objects in an Oracle Cloud Infrastructure Object Storage Classic instance are represented as REST resources and are accessible through HTTP uniform resource locators (URLs).
- **Amazon S3 URI Format**
  - For example, the following refers to the `channels.txt` file in the atpc bucket in the us-west-2 region: (This is called a region-specific endpoint.)
    - https://s3-us-west-2.amazonaws.com/atpc/channels.txt

DBMS_CLOUD Package File URI Formats

- **Azure Blob Storage URI Format**
  - For example, the following refers to the `channels.txt` file in the atpc container in the storage account atpc_user:
    - https://atpc_user.blob.core.windows.net/atpc/channels.txt
- Example of an Object Storage API:



ORACLE

DBMS_CLOUD Package Format Options

DBMS_CLOUD Package Format Options

The format argument in `DBMS_CLOUD` specifies the format of source files. These formats help to process exactly as has been mentioned in format_value. If a format_option is not used, then it is considered as the default format_value.

The two ways to specify the format argument are:

format => '{"***format_option***" : "***format_value***" }'

And:

format => json_object('***format_option***' **value** '***format_value***'))

For example:

format **=> json_object('type' VALUE 'CSV')**

format **=> json_object('ignoremissingcolumns' value 'true', 'removequotes' value 'true'),**

- **Ignoremissingcolumns:** If there are more columns in field_list than there are in the source files, the extra columns are stored as null. The default value is False.
- **Removequotes:** Removes any quotes that are around any field in the source file. The default value is False.
- **Recorddelimiter:** Specifies the record delimiter. The default is newline.
- **Delimiter:** Specifies the field delimiter. The default is | (pipe character).

ORACLE


```
begin
 dbms_cloud.create_external_table(
    table_name =>'CHANNELS_EXT',
    credential_name =>'EXT_TAB',
    file_uri_list => 'https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/ocuocictrng23/ext_table/chan_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true'),
    column_list => 'CHANNEL_ID NUMBER,
     CHANNEL_DESC VARCHAR2(20),
     CHANNEL_CLASS VARCHAR2(20),
     CHANNEL_CLASS_ID NUMBER,
     CHANNEL_TOTAL VARCHAR2(13),
     CHANNEL_TOTAL_ID NUMBER'
 );
end;
/
```

We have not used recorddelimiter and delimiter in this example because the `chan_v3.dat` file has the following type of data:

```
3|"Direct Sales"|"Direct"|12|"Channel total"|1|
9|"Tele Sales"|"Direct"|12|"Channel total"|1|
5|"Catalog"|"Indirect"|13|"Channel total"|1|
4|"Internet"|"Indirect"|13|"Channel total"|1|
2|"Partners"|"Others"|14|"Channel total"|1|
```

DBMS_CLOUD Package Format Options

- **Trimspaces:** Specifies how the leading and trailing spaces of the fields are trimmed. Valid values are `rtrim`, `ltrim`, `notrim`, and so on. `notrim` is the default value.
- **Ignoreblanklines:** Blank lines are ignored when set to true. The default is false.
- **Truncatecol:** If data in a file is longer than the field length, then instead of rejecting the row, the field will be truncated if set to true.
- **Dateformat and timestampformat:** The AUTO format option searches for:

```
J
MM-DD-YYYYBC
MM-DD-YYYY
YYYYMMDD HHMISS
YYMMDD HHMISS
YYYY.DDD
YYYY-MM-DD for dateformat
And
YYYY-MM-DD HH:MI:SS.FF
YYYY-MM-DD HH:MI:SS.FF3
MM/DD/YYYY HH:MI:SS.FF3 for timestampformat
```

Format options also have timestamptzformat and timestampltzformat.

Restrictions for Database Initialization Parameters

ORACLE AUTONOMOUS DATABASE

ORACLE

Restrictions for Database Initialization Parameters

- The respective initialization parameters are configured based on compute and storage values provided by you during instance configuration or when scaling up or down.
- You are not required to set any initialization parameters to start any service, but can modify a few of the initialization parameters. The following are the initialization parameters that can be modified:

APPROX_FOR_AGGREGATION	NLS_ISO_CURRENCY	OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES Allowed only with ALTER SESSION
APPROX_FOR_COUNT_DISTINCT	NLS_NCHAR_CONV_EXCP	OPTIMIZER_IGNORE_HINTS
APPROX_FOR_PERCENTILE	NLS_LANGUAGE	OPTIMIZER_IGNORE_PARALLEL_HINTS
AWR_PDB_AUTOFLUSH_ENABLED	NLS_LENGTH_SEMANTICS	PLSCOPE_SETTINGS
NLS_CALENDAR	NLS_NCHAR_CONV_EXCP	PLSQL_DEBUG
NLS_COMP	NLS_NUMERIC_CHARACTERS	PLSQL_WARNINGS
NLS_CURRENCY	NLS_SORT	PLSQL_OPTIMIZE_LEVEL
NLS_DATE_FORMAT	NLS_TERRITORY	PLSQL_CCFLAGS
NLS_DATE_LANGUAGE	NLS_TIMESTAMP_FORMAT	
NLS_DUAL_CURRENCY	NLS_TIMESTAMP_TZ_FORMAT	

ORACLE

Restrictions for Database Initialization Parameters

- Auto stats gathering is enabled in ATP and runs in the Standard maintenance window. This happens automatically in ATP.
- ATP honors optimizer hints and PARALLEL hints in SQL statements by default. This behavior can be disabled by setting `OPTIMIZER_IGNORE_HINTS` to `TRUE` at either the session or system level. In the same way, `OPTIMIZER_IGNORE_PARALLEL_HINTS` can be set to `TRUE`.
  - Example: `ALTER SESSION SET OPTIMIZER_IGNORE_HINTS=TRUE;`
  - `ALTER SESSION SET OPTIMIZER_IGNORE_PARALLEL_HINTS=TRUE;`

Restrictions for Database Features

Restrictions for Database Features

Autonomous Transaction Processing is built for mixed workloads and transaction processing. In some cases, features of Oracle Database Enterprise Edition are not available in Autonomous Transaction Processing. Additionally, database features that are designed for Oracle Database administration are not available in ATP. The following features have been removed from or are not available in ATP:

- Oracle Real Application Testing
- Oracle Database Vault
- Oracle OLAP
- Oracle R capabilities of Oracle Advanced Analytics
- Oracle Spatial and Graph
- Oracle Industry Data Models
- Oracle Tuning Pack
- Oracle Database Lifecycle Management Pack
- Oracle Data Masking and Subsetting Pack
- Oracle Cloud Management Pack for Oracle Database
- Oracle Application Express
- Oracle Multimedia
- Java in DB
- Oracle XML DB
- Context
- Oracle Workspace Manager

ORACLE

Restrictions for SQL Commands

Restrictions for SQL Commands

To ensure security and performance of Autonomous Transaction Processing, some SQL commands are restricted.

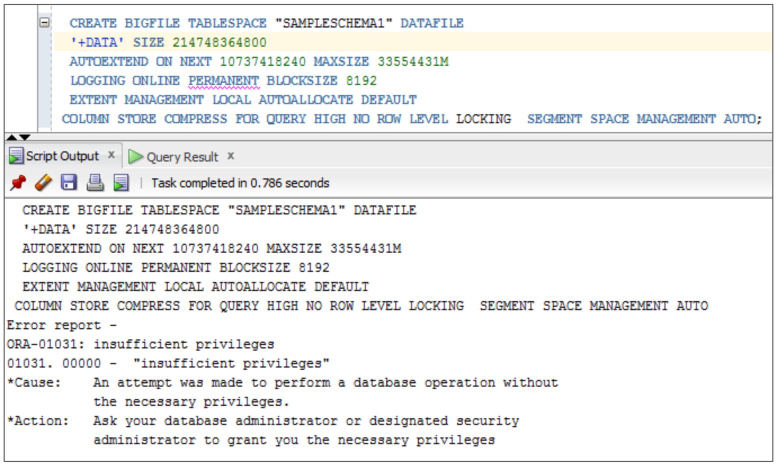
- If you try to use a restricted SQL command, the system reports:

  `ORA-01031: insufficient privileges`

  This error indicates that you are not allowed to run the SQL command in Autonomous Transaction Processing.
- The following SQL statements are not available in Autonomous Transaction Processing:

  `ADMINISTER KEY MANAGEMENT`, `ALTER PROFILE`, `ALTER TABLESPACE`, `CREATE DATABASE LINK`, `CREATE INDEX (BITMAP)`, `CREATE TABLESPACE`, `DROP TABLESPACE`

Restrictions for SQL Commands



ORACLE

Restrictions for SQL Commands

- The DDL statements that are available in Autonomous Transaction Processing with restrictions are:
  - **`ALTER PLUGGABLE DATABASE` and `ALTER DATABASE`**: Only the following clauses are allowed:
    `DEFAULT EDITION`, `SET TIME_ZONE`, `DATAFILE RESIZE`, `DATAFILE AUTOEXTEND ON`, `DATAFILE AUTOEXTEND OFF`
  - **`ALTER SESSION`**: Only the following clauses are allowed:
    `ADVISE COMMIT`, `ADVISE ROLLBACK`, `ADVISE NOTHING`, `CLOSE DATABASE LINK`, `ENABLE COMMIT IN PROCEDURE`, `DISABLE COMMIT IN PROCEDURE`, `ENABLE PARALLEL <QUERY|DDL|DML>`, `DISABLE PARALLEL <QUERY|DDL|DML>`, `FORCE PARALLEL <QUERY|DDL|DML>`, `ENABLE RESUMABLE`, `DISABLE RESUMABLE`, `SET EDITION`, `SET ROW ARCHIVAL VISIBILITY`, `SET DEFAULT_COLLATION`
- `ALTER SYSTEM` is not allowed except `ALTER SYSTEM SET` and `ALTER SYSTEM KILL SESSION`.
- `ALTER USER` and `CREATE USER`, the default TABLESPACE and PROFILE, are ignored.

ORACLE

Restrictions for SQL Commands

- Object tables and `XMLType` tables are not allowed.

The clauses that are not in this list are allowed.

Clause	CREATE TABLE Comment	ALTER TABLE Comment
`physical_properties`	Ignored	Ignored
`logging_clause`	Ignored	Ignored
`table_compression`	Ignored	Ignored
`inmemory_table_clause`	Ignored	Ignored
`ilm_clause`	Ignored	Ignored
`organization index`	Not allowed	Ignored
`organization external`	Not allowed	Ignored
`Cluster`	Ignored	Ignored
`LOB_storage_clause`	Ignored	Ignored
`zonemap_clause`	Not allowed	Ignored

Restrictions for Data Types

Restrictions for Data Types

The following data types are restricted in Autonomous Transaction Processing:

- Media types
- Spatial types


```
create table emp (id number, name varchar(100), address long)

select * from emp;

insert into emp values(1,'RAM',12344)
select * from emp

create table emp1 (id number, name varchar(100), address long raw);

CREATE OR REPLACE TYPE my_tab_t AS TABLE OF VARCHAR2(30);
CREATE TABLE nested_table (id NUMBER, col1 my_tab_t)
       NESTED TABLE col1 STORE AS col1_tab;

       INSERT INTO nested_table VALUES (1, my_tab_t('A'));
INSERT INTO nested_table VALUES (2, my_tab_t('B', 'C'));
INSERT INTO nested_table VALUES (3, my_tab_t('D', 'E', 'F'));

commit

SELECT * FROM nested_table;

SELECT id, COLUMN_VALUE FROM nested_table t1, TABLE(t1.col1) t2;
```

All the previous commands should work fine.

```
CREATE TABLE cola_markets (
  mkt_id NUMBER PRIMARY KEY,
  name VARCHAR2(32),
  shape MDSYS.SDO_GEOMETRY);

CREATE TABLE cola_markets (
  mkt_id NUMBER PRIMARY KEY,
  name VARCHAR2(32),
  shape MDSYS.SDO_GEOMETRY)
Error report -
ORA-00902: invalid datatype
00902. 00000 -  "invalid datatype"
*Cause:
*Action:
```

Sample Star Schema Benchmark (SSB) Queries and Analytic Views

Sample Star Schema Benchmark (SSB)

- The SSB schema contains the lineorder, customer, supplier, part, and dwdate tables.
- You can use sample queries and analytic views against the SSB schema.
- Both SH and SSB are provided as schema-only users, so you cannot unlock or drop those users or set a password.
- The storage of the sample data sets does not count toward your database storage.

ORACLE

Star Schema Benchmark Analytic Views

- Analytic views make it easy to extend a star schema with:
  – A hierarchical business model
  – Aggregation and measure calculation rules
  – Presentation and application-specific metadata
- The SSB schema includes an analytic view and four hierarchies that use the tables of the star schema.

SSB Analytic Views

Analytic views make it easy to extend a star schema with a hierarchical business model, aggregation and measure calculation rules, and presentation and application-specific metadata that can be used to enhance the content of a data set and simplify the development of business intelligence applications. The SSB schema includes an analytic view and four hierarchies that use the tables of the star schema. Use the following queries to query the analytic sample SSB view. Note that the analytic view is in the SSB schema.

The analytic view can be validated using VALIDATE_HIERARCHY and VALIDATE_ANALYTIC_VIEW.

ALL_ANALYTIC_VIEW_DIM_CLASS, ALL_ANALYTIC_VIEW_KEYS, and ALL_ANALYTIC_VIEW_LVLGRPS dictionary views can be used to get information related to analytic views.

Summary

In this appendix, you should have learned how to:

- Summarize `DBMS_CLOUD` subprograms
- Describe `DBMS_CLOUD` package file URI formats
- Describe `DBMS_CLOUD` package format options
- Explain the restrictions for:
  - Database initialization parameters
  - Database features
  - SQL commands
  - Data types

ORACLE