ORACLE UNIVERSITY

Integrated Cloud Applications & Platform Services

Oracle Cloud Infrastructure Fundamentals

Student Guide

D100804GC10

Edition 1.0 | February 2018 | D102534

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

ORACLE

Authors

Brendan Howes

Sardar Jamal Arif

Rohit Rahi

Renu Bhatt

Technical Contributors and Reviewers

Brendan Howes

Sardar Jamal Arif

Rohit Rahi

Renu Bhatt

Editor

Arijit Ghosh

Publishers

Asief Baig

Giri Venugopal

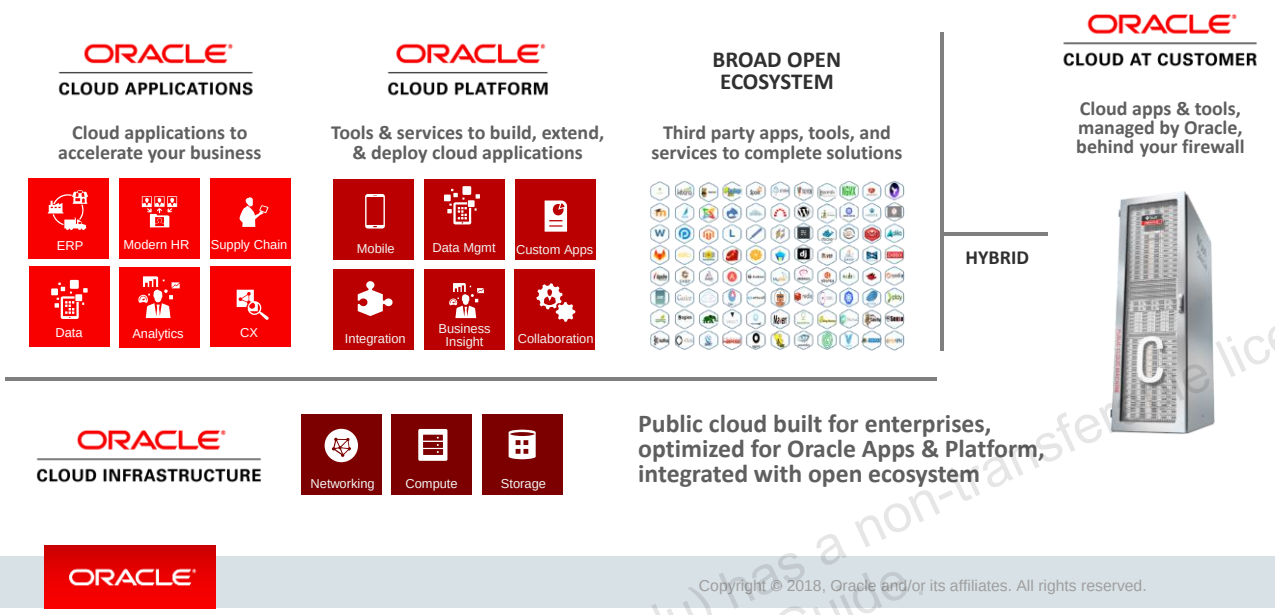
Jobi Varghese

Contents

1

Getting Started with Oracle Cloud Infrastructure

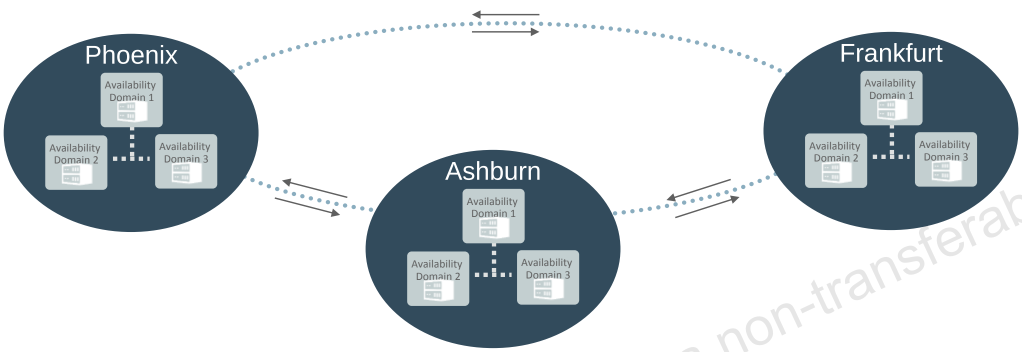
Oracle Cloud Infrastructure Strategy



Our vision is that Oracle Cloud Infrastructure is the foundation on which our core Infrastructure, Platform and SaaS services will run. In short, OCI is the very foundation of Oracle Cloud services. As you can see here, Oracle Cloud Infrastructure is optimized to run Oracle DB and apps but it is also an open platform with a broad ecosystem that supports 3rd party apps, tools and frameworks. Customers who want to run these exact same services behind a corporate firewall can leverage Cloud @ Customer

Regions + Availability Domains + Backbone Network

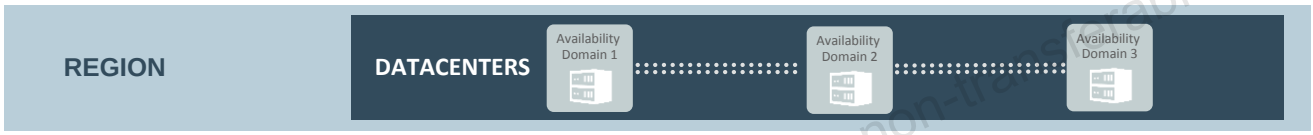
- Regions serve different geographies – provide Disaster Recovery capability
- Availability Domains – provide a High Availability foundation within a region
- Backbone Network + Peering – private connectivity between regions and direct peering



Let us discuss the core concepts briefly. We operate in regions – a region is a metropolitan area; inside the regions we have multiple data centers that are called Availability domains. 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 and 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

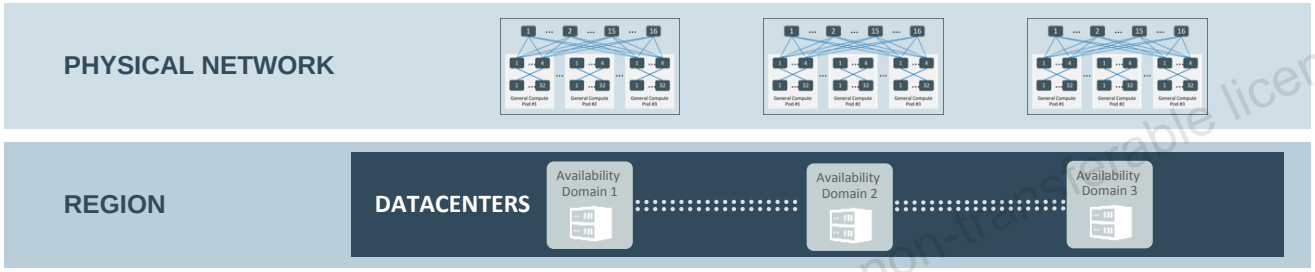
- Multiple fault-decorrelated, completely independent datacenters: ADs
- Predictable low latency & high speed, encrypted interconnect between ADs
  - < 500µs expected one-way latency, 1Tb/s bandwidth
- Enables zero-data-loss architectures (e.g. Oracle MAA) and high availability scale-out architectures (e.g. Cassandra)



Again, regions are constructed of insolated fault domains. You can see some latency numbers here < 500 micro seconds one-way latency.

Inside an AD – High Scale, High Performance Network

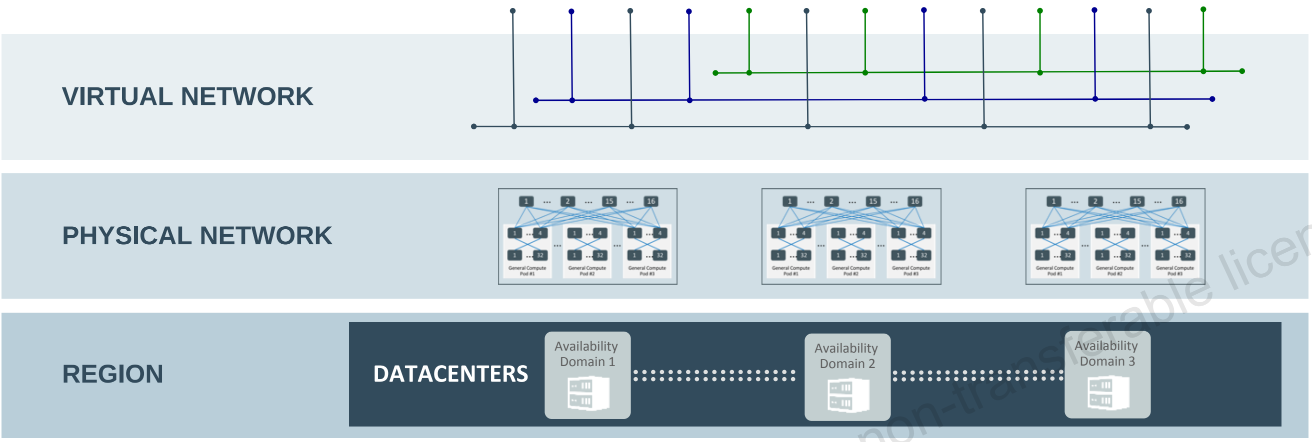
- Non-oversubscribed network – flat, fast, predictable
- Very high scale – ~1 million network ports in an AD
- Predictable low latency & high speed interconnect between hosts in an AD
  - < 100µs expected one-way latency, 2 x 25Gb/s bandwidth



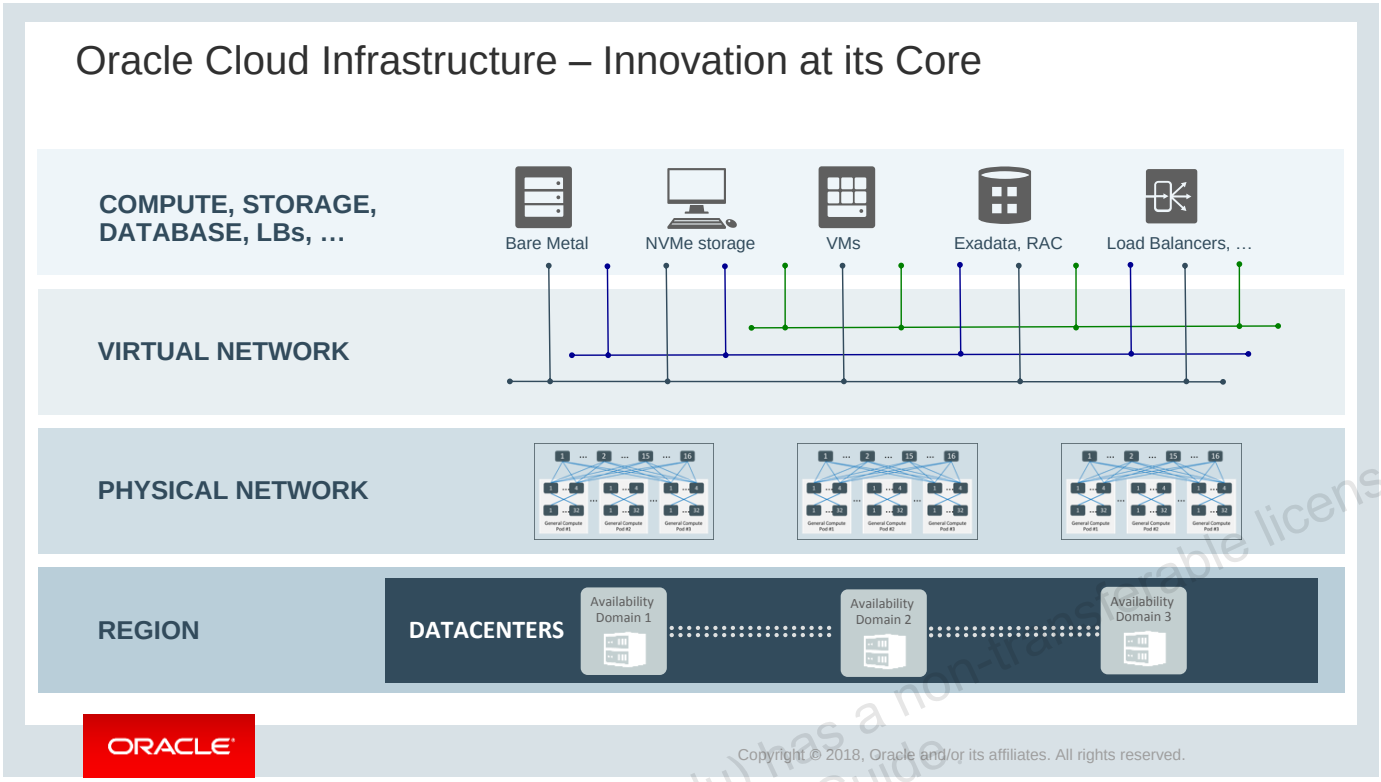
Inside of these ADs, we have built one of the best public cloud networks. It is big, flat and fast. It runs on a high scale to the tune of 1 million network works per AD. By flat, I mean it is not oversubscribed, so we get tremendously good latency. And fast means we support 25 Gbps network bandwidth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

Comprehensive Virtual Network with Off-box Virtualization

Highly configurable private overlay networks – moves management and I/O out of the hypervisor and enables lower overhead and bare metal instances

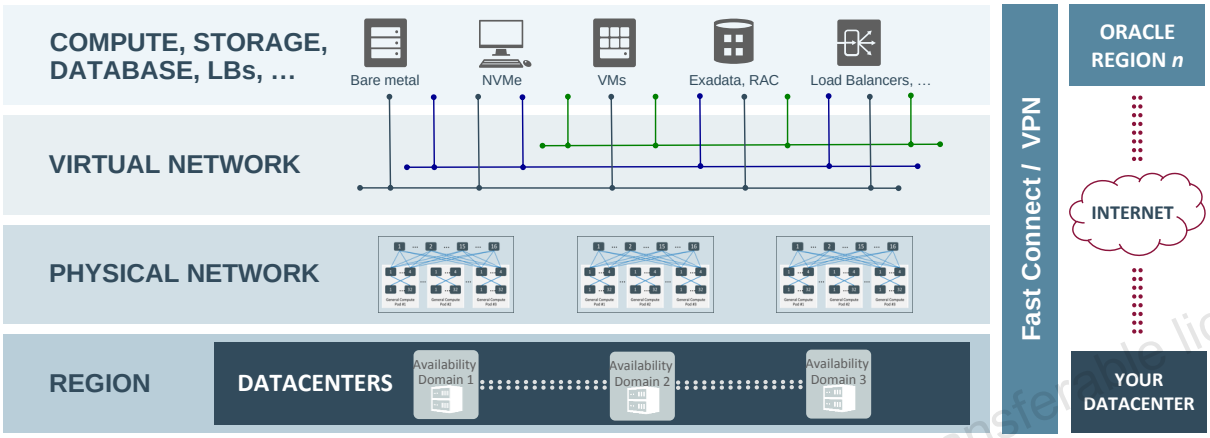


We have made some drastic changes into how virtual networking is done. We call if off-box virtualization. As the name implies, we pulled all the virtualization out into the network, including storage and network IO virtualization. Generally, this enables the next layer up – so we can take any physical form-factor and plug that in to 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 the classes of services and meet our goals around performance and security.

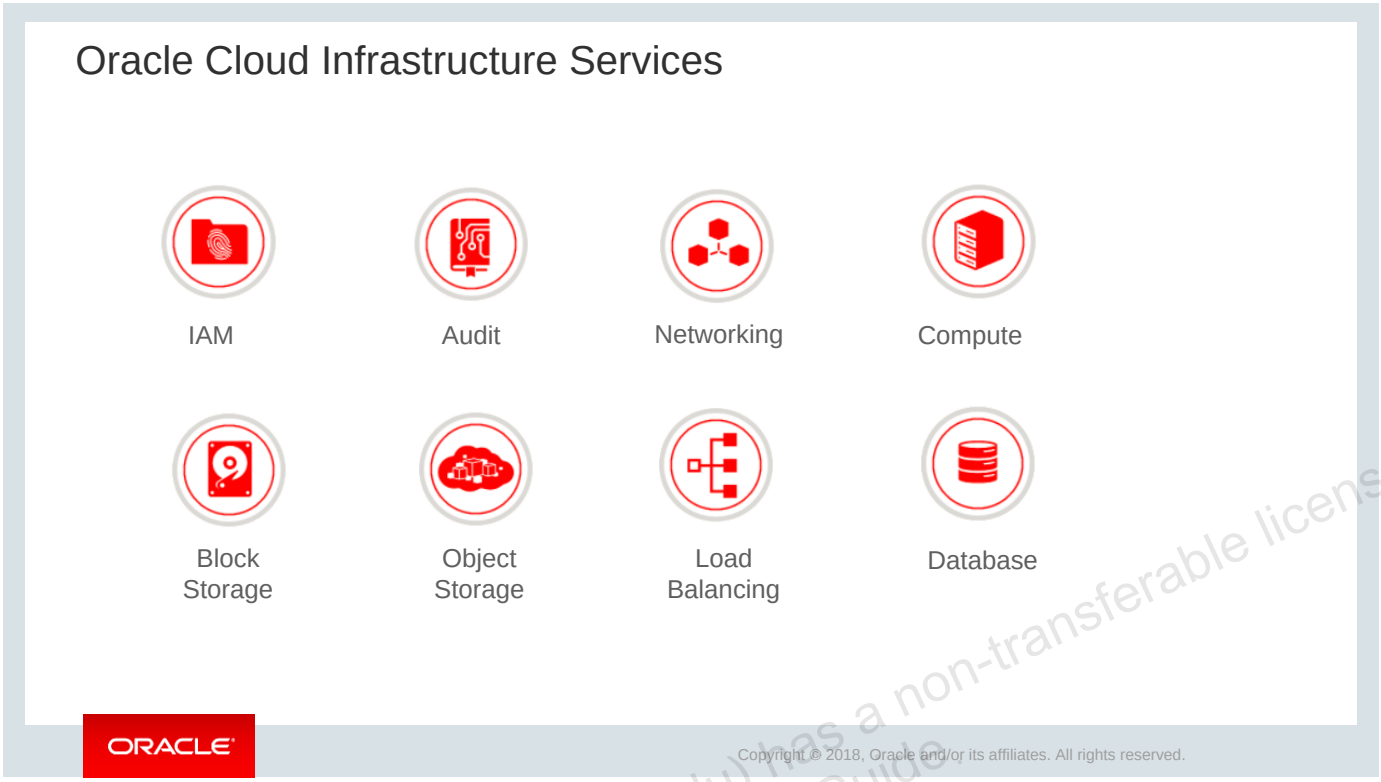


So this is what the fully integrated picture looks like. We have the largest class of compute and storage options plugged into a highly differentiated virtualized environment built using off-box network virtualization

Oracle Cloud Infrastructure – Innovation at its Core



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



Ok, 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. 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 - Infrastructure Heavy Workloads that demand high scale/high performance
  - Best hardware, best performance, best price
  - Big Data, HPC, Machine Learning
- Cloud Native workloads: 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 like SAP etc. 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 like Terraform and run these apps on OCI.

Key Differentiators

Oracle Apps and support for Enterprise IaaS Architecture

- Best place to run Oracle Database and key enterprise Oracle Apps
- Industry's first Bare Metal Cloud Services
- Flexibility and control (Bare Metal and VMs share the same set of APIs)
- Off-Box Network Virtualization (w/ support for plugging Exadata appliances)
- Non-oversubscribed network, predictable performance with low latency and high throughput
- Robust Security and Governance capabilities

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 throughout and low latency. We also have a unique approach to security and governance through the use of compartments, which we'll talk about in the later modules.

Course outline

- Lesson 2: Identity and Access Management Service
- Lesson 3: Virtual Cloud Network Service
- Lesson 4: Compute Service
- Lesson 5a: Block Volume Service
- Lesson 5b: Object Storage Service
- Lesson 6: Load Balancing Service
- Lesson 7: Database Service
- Lesson 8: DNS service

In the next 8 modules, we'll cover the core OCI services starting with Identity and access management, networking, compute, storage, Database and DNS. That's all for this lecture.

2

Identity and Access Management Service

Objectives

After completing this lesson, you should be able to:

- Describe the concepts and terms used in IAM service
- Log in and navigate through the web console
- Configure users and groups
- Create compartments and Policies

In this lecture, we'll describe the key concepts and terms used in the IAM service

Identity and Access Management Service

- Identity and Access Management Service (IAM)
  - control who can access your OCI account
  - what services and resources they can use
  - how they can use these resources
- Resource is a cloud object that you create and use in OCI (e.g. compute instances, block storage volumes, Virtual Cloud Networks)
- IAM uses traditional identity concepts such as Principals, Users, Groups, Policies
- OCI IAM introduces a new feature called Compartments

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 etc. There are 5 concepts you need to know in IAM – Principals, Users, Groups, Policies and Compartments. The fifth one compartment is a unique OCI feature that we'll look into in the later slides.

Principals

- A principal is an IAM entity that is allowed to interact with OCI resources
- Three types of Principals – root users, IAM users and Instance Principals
- Root User
  - when customers sign-up for an OCI account, the first IAM user is called the root user
  - root user is persistent and has complete administrative access to all OCI resources
- IAM Users/Groups
  - Users are persistent identities setup through the IAM service to represent individual people or applications
    - Users enforce principle of least privilege
      - user has no permissions until placed in one (or more) groups and
      - group having at least one policy with permission to tenancy or a compartment
  - Group is a collection of users who all need the same type of access to a particular set of resources
  - Same users can be member of multiple groups

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.

Instance Principals

- Instance Principals can make API calls against other OCI services without storing credentials in a configuration file
- Instance Principals are implemented in OCI with Dynamic Groups
- Membership in the dynamic group is determined by a set of matching rules. When you set up a dynamic group, you also define the rules for membership in the group.
- Resources that match the rule criteria are members of the dynamic group
- Dynamic Groups also need Policies to access OCI resources

ORACLE

Suppose that an application running on an OCI compute instance needs to access an OCI Object Storage bucket. A policy granting permission to read and write that bucket can be created and assigned to an IAM

user, and the application can use the config file and key for that IAM user to access the object bucket. The problem with this approach is that the private key for the user must be accessible to the application, probably by storing it in some sort of configuration file. The process for obtaining the private key and storing it in the configuration is usually complicated and a hindrance to agile development. So the solution is to use Instance principals to make API calls against OCI services without storing credentials in a config file.

Instance Principals is discussed in detail in the advanced IAM module.

Tenants and Compartments

- Tenancy
  - Equivalent of an account; tenancy contains all of your OCI resources
  - Provisioned with a single, top-level compartment called the ‘root compartment’; you can create other compartments
- Compartment
  - Logical container used to organize and isolate cloud resources; each resource is in exactly one compartment
  - Compartments are global and logical; distinct from physical “containers” like Regions and Availability Domains
  - Resources can be connected/shared across compartments
  - Compartments can be renamed (but not deleted or moved)
  - Currently, compartments are only one level deep, but in future, will be hierarchical (permissions in a parent compartment inherited by child compartments)

Tenancy Concept equivalent to an account. Compartments are logical containers , these are logical not physical, which are used to organize and isolate cloud resources. It is a useful concept in the case of ‘shadow IT’

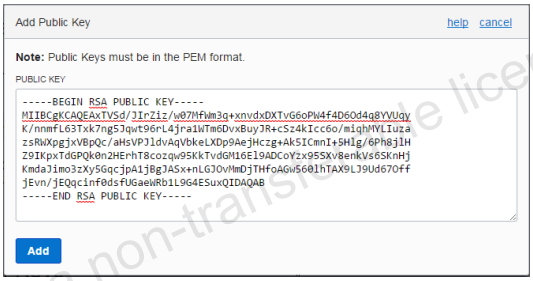
Common Questions:

- How do you do billing? Tagging Feature
- Hierarchical Compartments : Roadmap item

Authentication

Two ways IAM service authenticates a Principal

- User name, 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 API in conjunction with the SDK
  - 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 user name, password combination. A program that accesses the API with an IAM user or root user uses an API signing key

Authorization

- Process of specifying what actions an authenticated Principal can perform is called authorization
- Authorization in IAM service done by defining specific privileges in policies and associating them with principals
- Supports 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 <tenancy_name>
  - Allow group <group_name> to <verb> <resource-type> in compartment <compartment_name> [where <conditions>]
  - E.g. Allow group ProjectA_Admins to manage all-resources in compartment ProjectA_compartment

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

Policies

Allow group *<group_name>* to *<verb>* *<resource-type>* in tenancy *<tenancy_name>*

verb	Type of access
inspect	Read only access without access to any user-specified metadata
read	Read only access, plus the ability to get user-specified metadata
use	Update existing resources, but not create or delete
manage	Includes all permissions for the resource

Aggregate resource-type	Individual resource type
all-resources	
database-family	db-systems, db-nodes, db-homes, databases
instance-family	instances, instance-images, volume-attachments, console-histories
object-family	buckets, objects
virtual-network-family	vcn, subnet, route-table, more
volume-family	Volumes, volume-attachments, volume-backups

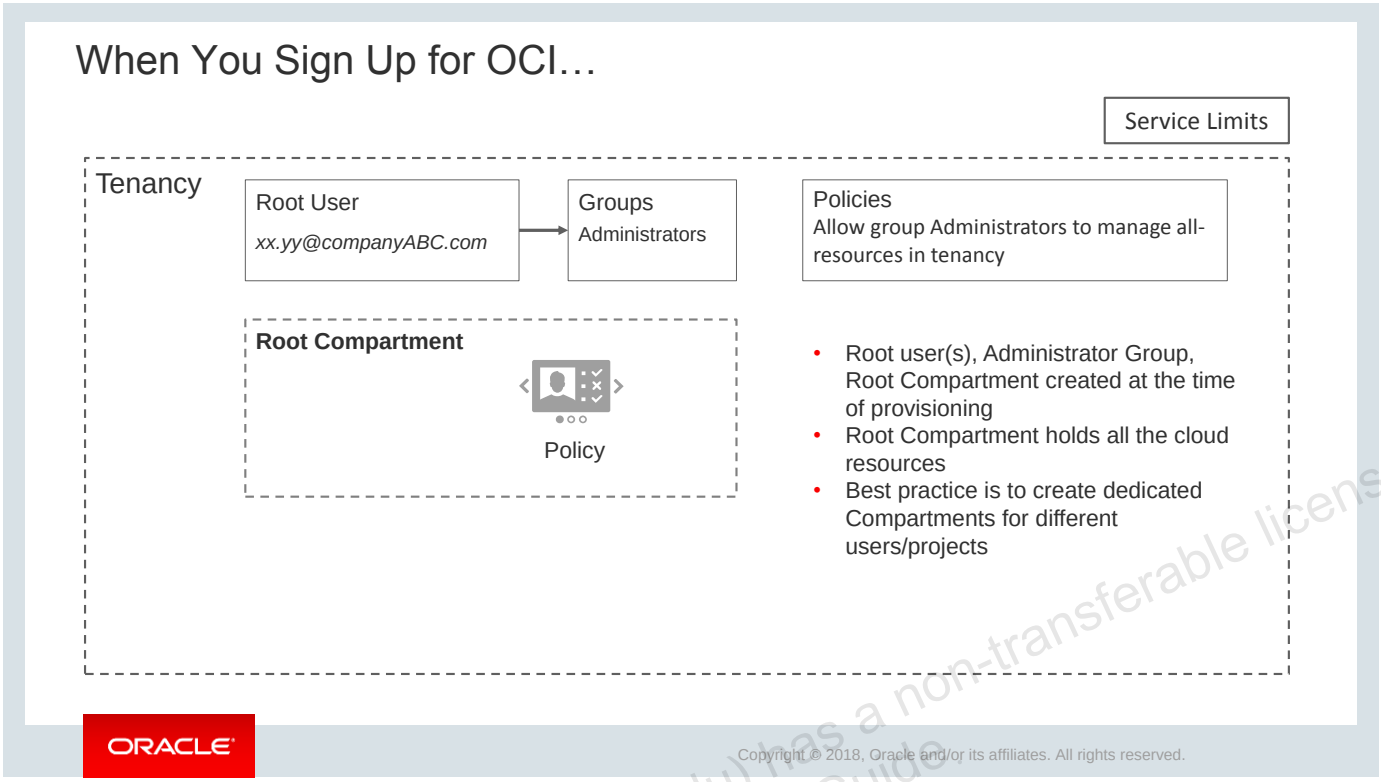
The IAM Service has no family resource-type, only individual ones; Audit and Load Balancer have individual resources (load-balancer, audit-events)

This slide shows you the syntax of Policies. There are three things needed for policies – an action or verb, resource type and whether the policy is at the tenancy or compartment level. Furthermore, IAM allows granular policies, so they can be applied at either the aggregate level or individual resource level. Polices can also include one more conditions

Policy Examples

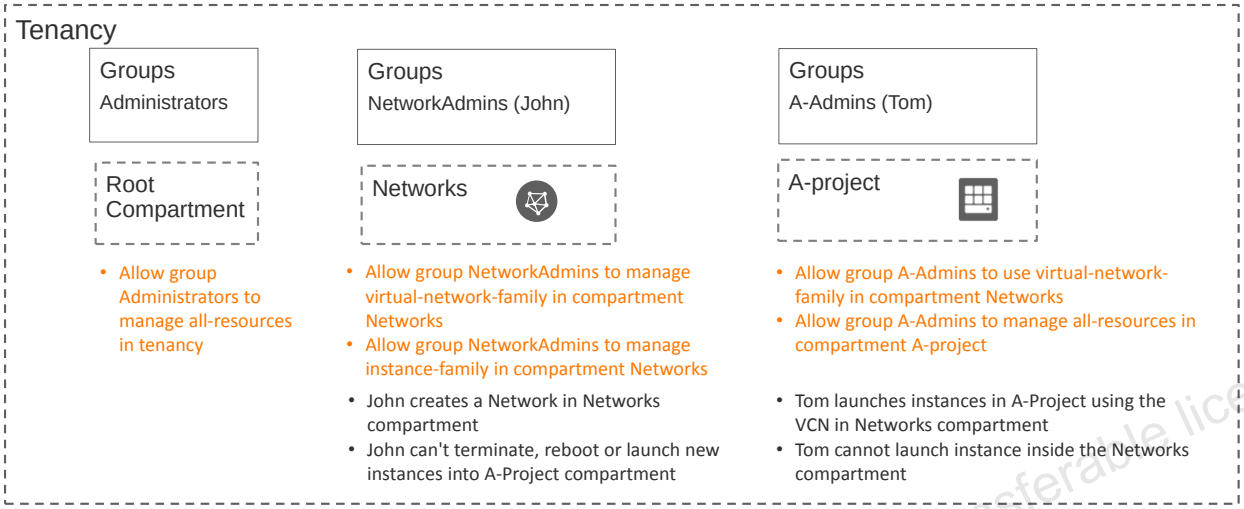
Aggregate Resource Types	Individual Resource Types
allow group Admins to manage all-resources in tenancy	
allow group NetworkAdmins to manage virtual-network-family in tenancy	allow group NetworkAdmins to manage subnet in compartment Marketing
allow group HRAdmins to use instance-family in compartment HR	allow group HRAdmins to use console-histories in compartment HR
allow group ServerAdmins to inspect object-family in tenancy	allow group ServerAdmins to read objects in compartment IT

Same as the bullet points



When you sign up for OCI, a root user, administrator group and a root compartment is created by default.

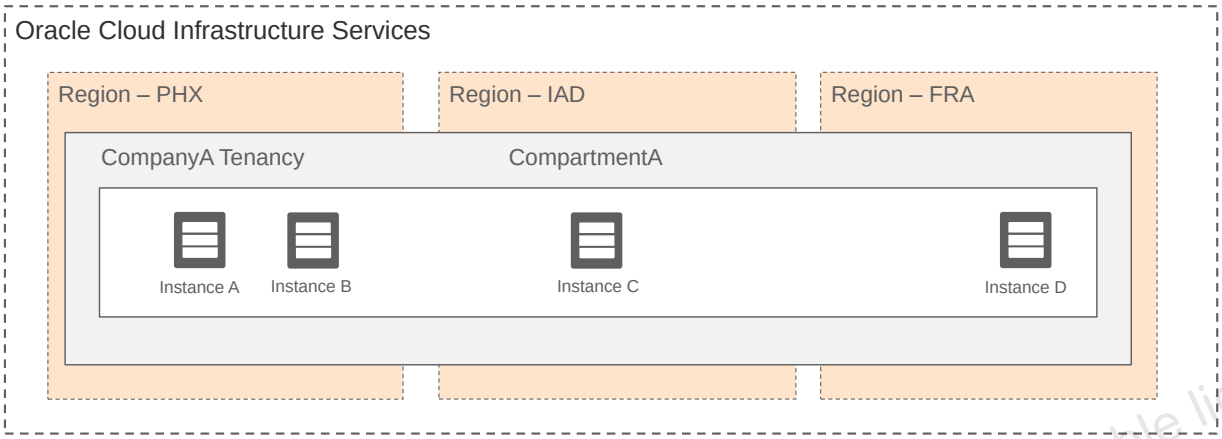
VCN Network Topology v/s Access Control Using Compartments



The instances Tom launched reside in the VCN from a network topology standpoint but from an access standpoint, they're in the A-Project compartment, not the Networks compartment where the VCN is

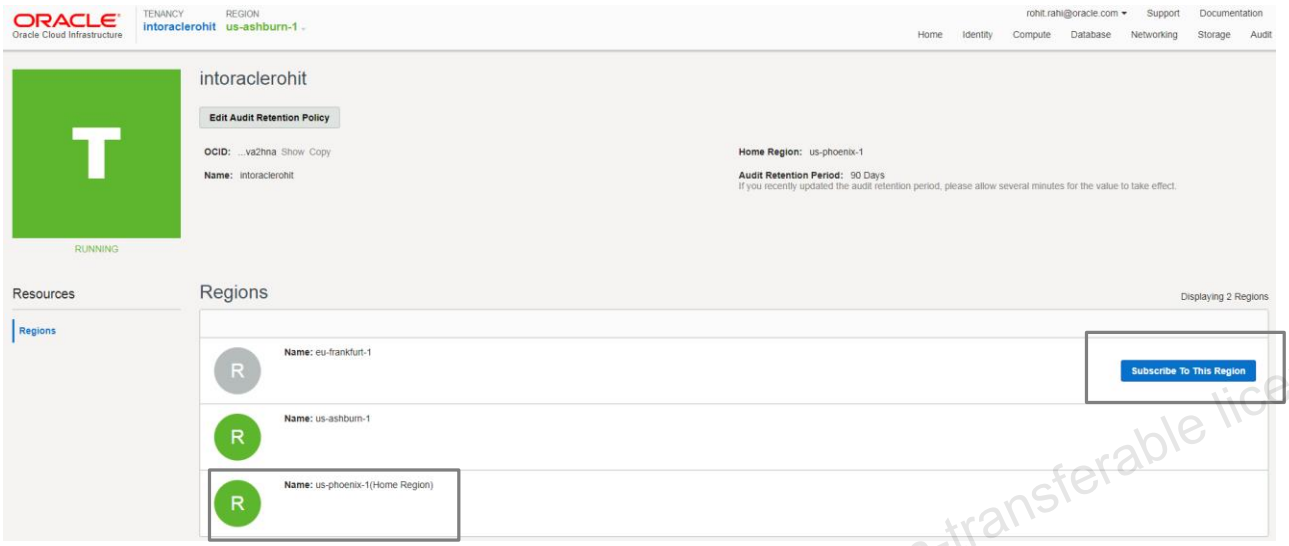
Same as bullet points

IAM Service Resources Are Global



- IAM Service resources (compartments, users, groups, and policies) are global, so you can access them across all regions
- Home region is where you sign-up and your subscription resides (but can always subscribe to other regions)

IAM Service Resources Are Global



Resource Locations

Service	Resource	Location	
IAM	Users, Groups, Policies, Compartments, API Signing Keys	Global	
Compute	Images	Regional	
	Instances	Availability Domain	Instances can be attached only to volumes in the same AD
	Volumes	Availability Domain	
	Volume backup	Region	Backups can be restored as new volumes to any AD within the same region
Database	DB Systems	Availability Domain	
Network	Virtual Cloud Network (VCN)	Region	
	Subnet	Availability Domain	
	Security Lists, Route Table	Region	
	Dynamic Routing Gateway (DRG)	Region	
	Customer Premises Equipment (CPE), Internet Gateway	Region	

Resource Locations

Service	Resource	Location	
Load Balancer	Load Balancer	Region	
Storage	Buckets	Region	Bucket is a regional resource but it can be accessed from any location as long as correct region-specific URL is used

Signing Up for Oracle Cloud Infrastructure Services

You can sign up for Oracle Cloud Infrastructure Services in the following ways:

- Contact your Oracle sales representative
- Visit Oracle Store, https://shop.oracle.com and sign up for the Oracle Cloud Infrastructure Services
- Sign up for a free trial at http://cloud.oracle.com/tryit

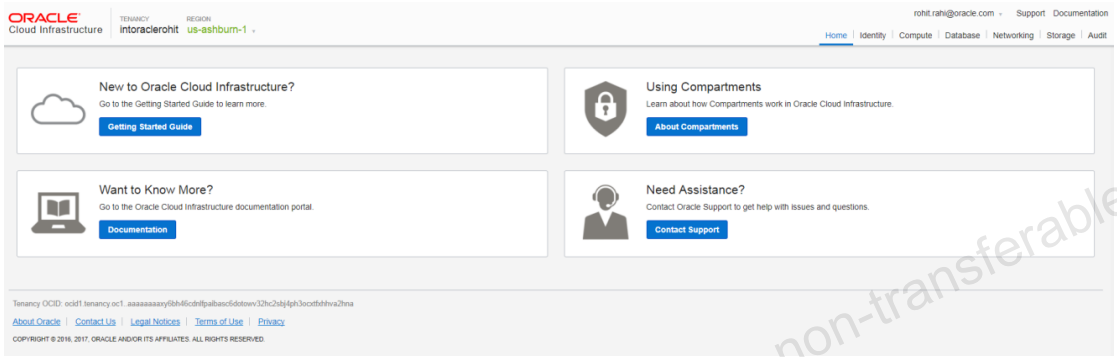
You can sign up for Oracle Cloud Infrastructure Services in the following ways:

- Contact your Oracle sales representative: Your Oracle sales representative can provide you information about the pricing options available to your company. Your sales representative will collect some information from you and initiate the registration process.
- Go to the Oracle Store: Visit https://shop.oracle.com/ and sign up for the Oracle Cloud Infrastructure Services.
- Sign up for a free trial at http://cloud.oracle.com/tryit

When your registration process is completed, you will be provisioned a “Tenancy” in Oracle Cloud Infrastructure Services. Oracle will send you a notification email with instructions to sign in to the web console for the first time. There is no charge until you start using the service.

Signing In to the Console

- Region based URL for the web-based console (e.g. Ashburn region): https://console.us-ashburn-1.oraclecloud.com
- Use the console to access and manage your Oracle Cloud Infrastructure services.
- The services you can use depend on: Service Limits set for your tenancy, permissions granted by administrator.



Console is the web-based user interface that you use to access and manage Oracle Cloud Infrastructure Services.

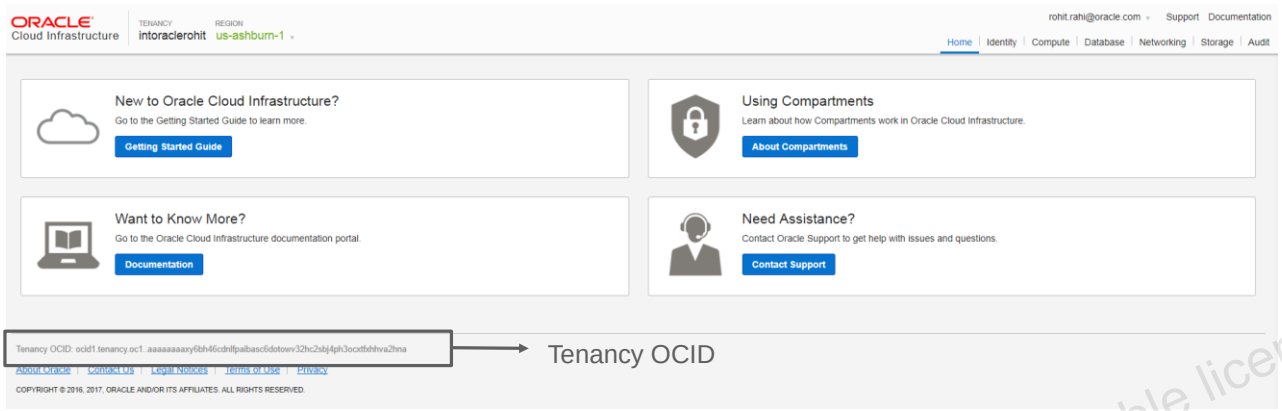
- The supported browsers include the latest versions of Google Chrome, Firefox, Microsoft Edge, and Internet Explorer 11.
- When you sign in to the web console, you'll see the home page.
- Use the service tabs in the upper right to create, manage, and view your cloud resources.
- Links to the documentation and to Oracle Support give you quick access to help and detailed information for using the services.

Resource Identifier

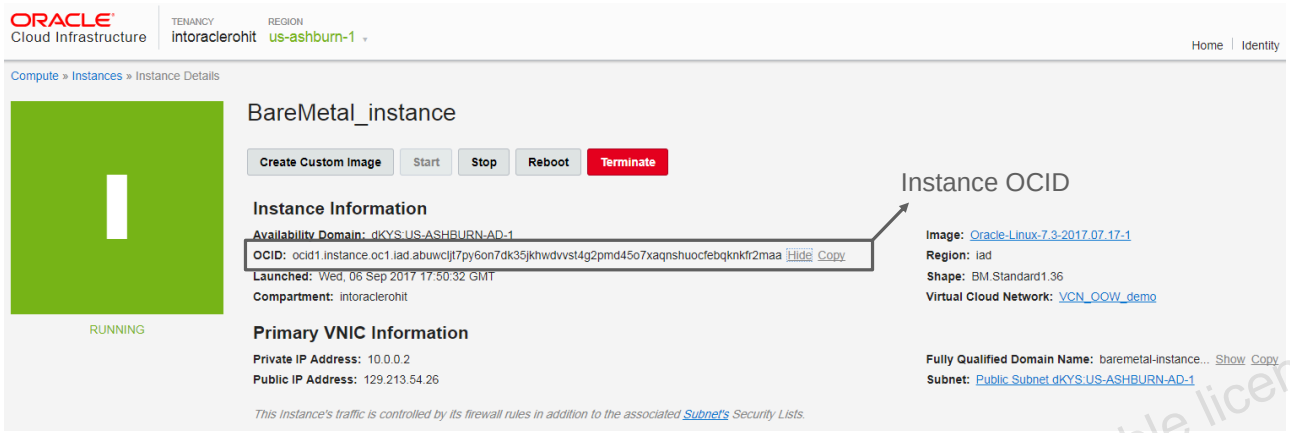
- Oracle Cloud Identifier (OCID) - Oracle-assigned unique ID to every resource
- ocid1.<RESOURCE TYPE>.<REALM>. [REGION][.FUTURE USE].<UNIQUE ID>
  - ocid1: literal string indicating the version of the OCID
  - Resource type: type of the resource (vcn, instance..)
  - Realm: currently oc1, realm is the set of regions that share entities
  - Future use: reserved for future use
  - Unique ID: unique portion of the ID
- Examples
  - tenancy: ocid1.tenancy.oc1..aaaaaaaaxy6bh46cdnlfpaibasc6dotowv32hc2sbj4ph3ocxtfxhhva2hna
  - instance: ocid1.instance.oc1.iad.abuwcljtwfk7f5e2o3q6ircgpdty6rg52itdyg72tgdtbiwqlujt7vm5h3da

Everything is a resource in the cloud, every resource gets a unique ID, we call them for short OCID.

Resource Identifier – Tenancy OCID



Resource Identifier – Instance OCID



Summary

In this lesson, you should have learned how to:

- Describe the concepts and terms used in IAM service
- Log in and navigate through the web console
- Configure users and groups
- Create compartments and Policies

3

Virtual Cloud Network Service

Objectives

After completing this lesson, you should be able to:

- Describe key Virtual Cloud Network (VCN) concepts
- Manage your cloud network components, such as:
  - Route Table, Security List
  - Internet Gateway, Dynamic Routing Gateway
- Evaluate the different options of connecting to the Internet
- Describe key advanced Features of VCN
- Describe NAT, VCN Peering, Multiple vNICs and Secondary IPs

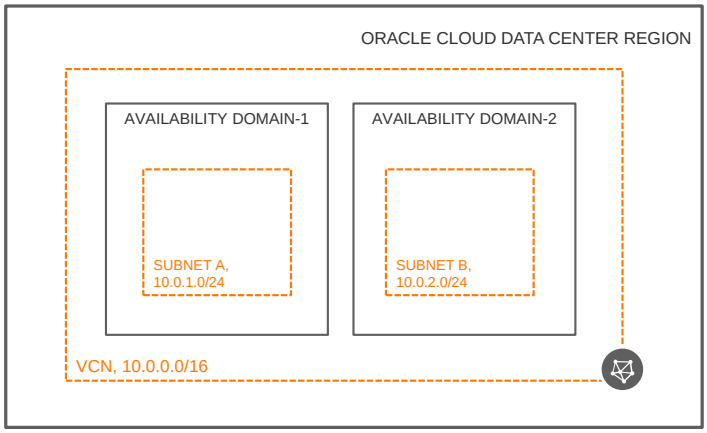
In this lecture, we'll go through the various virtual cloud network components, look at what these components do and discuss a few advanced concepts such as NAT and Peering

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.
- A VCN covers a single, contiguous IPv4 CIDR block of your choice.
- Oracle recommends using one of the private IP address ranges in RFC 1918 (10.0.0.0/8, 172.16/12, and 192.168/16) for VCN address space. However, you can use a publicly routable range.
- Allowable VCN size range is from /16 to /30 (VCN reserves the first two IP addresses and the last one in each subnet's CIDR).

A VCN within oracle cloud infrastructure if 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.

Subnet



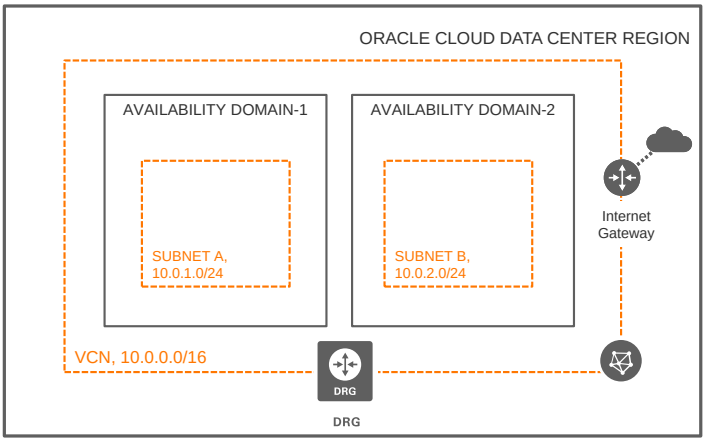
A VCN resides within a single region but can cross multiple Availability Domains (AD).

Subnet: Each VCN network is subdivided into subnets, and each subnet is contained within a single Availability Domain.

- You can have more than one subnet in an AD for a given VCN.
- Each subnet has a contiguous range of IPs, described in CIDR notation. Subnet IP ranges may not overlap.
- Subnets can be designated as either Public or Private.
- Instance draw their internal IP address and network configuration from their subnet.

Availability Domain Construct

Internet Gateway



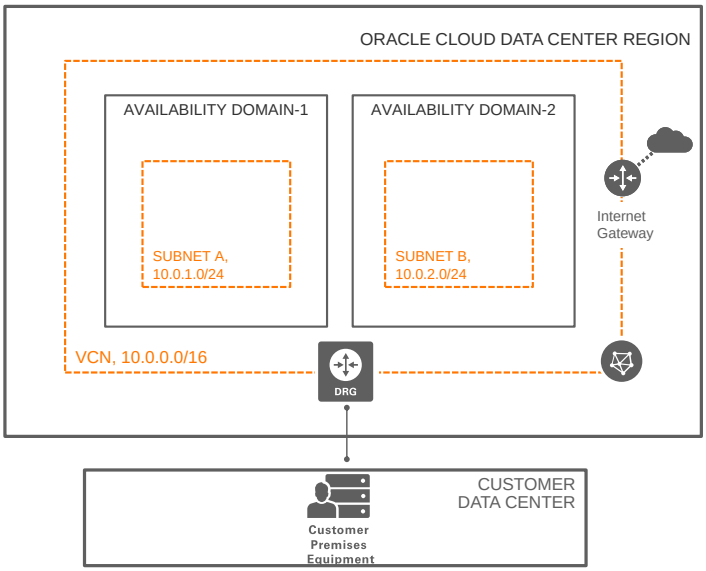
Internet Gateway:

- Internet Gateway provides a path for network traffic between your VCN and the internet
- After creating an Internet Gateway, you must add a route for the Gateway in the VCN's route table to enable traffic flow

Regional Construct

Internet gateways provide a path for network traffic between the VCN and the internet. When an instance on a public subnet is assigned a public IP you need to have a route table and internet gateway in place for the traffic to make it to and from the internet.

Dynamic Routing Gateway



Dynamic Routing Gateway (DRG):

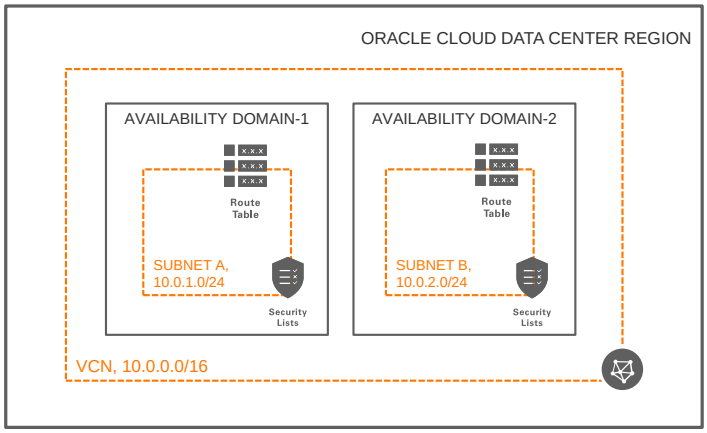
- A virtual router that provides a single point of entry for remote network paths coming into your VCN.
- You can use it to establish a connection with your on-premises network via IPSec VPN or FastConnect.
- After attaching a DRG, you must add a route for the DRG in the VCN's route table to enable traffic flow.

Regional Construct

Dynamic Routing Gateway - DRG - a virtual router that provides a point of entry for remote networks coming into the VCN. Peers can establish connection directly with your DRG instances either via Fastconnect, a connection mechanism for directly connecting other clouds or your physical locations or via IPSEC VPN.

After adding either a DRG or IGW to your VCN, you then must associate the gateway to your VCN and to add the appropriate routes in order to enable traffic flow

Security Lists, Route Table

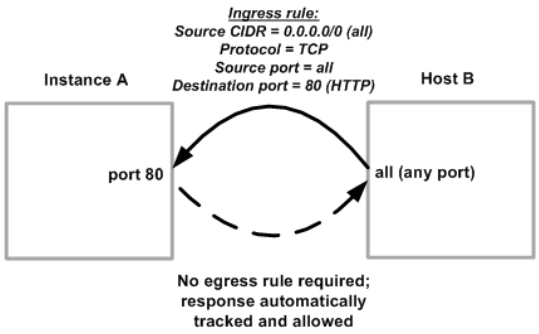


Security List: A common set of firewall rules associated with a subnet and applied to all instances launched inside the subnet

- Security lists provide ingress and egress rules that specify the types of traffic allowed in and out of the instances
- You can choose whether a given rule is stateful or stateless

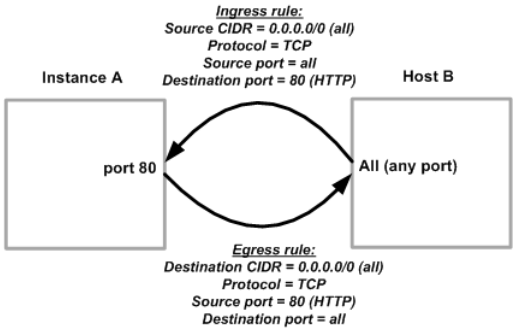
Route Table: A set of route rules that provide mapping for the traffic from subnets via gateways to destinations outside the VCN

Stateful Security Lists



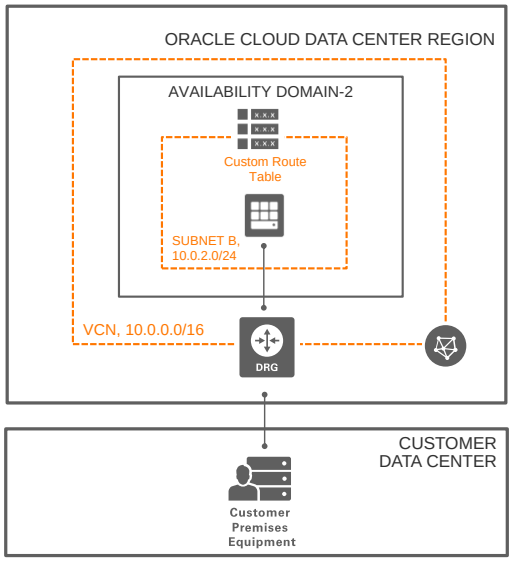
- Connection Tracking: when an instance receives traffic matching the stateful ingress rule, the response is tracked and automatically allowed regardless of any egress rules
- Similarly for sending traffic from the host
- Default Security Lists are stateful

Stateless Security Lists



- With stateless rules, response traffic is not automatically allowed.
- To allow the response traffic for a stateless ingress rule, you must create a corresponding stateless egress rule.
- If you add a stateless rule to a security list, that indicates that you do NOT want to use connection tracking for any traffic that matches that rule.
- Stateless rules are better for scenarios with large numbers of connections.

OCI VPN



- OCI VPN securely connects on-premises network to OCI VCN through an IPSec VPN connection
- OCI provisions redundant VPN tunnels located on physically and logically isolated tunnel endpoints
- Currently, only static routes are supported (BGP is not supported)
- OCI VPN supports only IKEv1 using a shared secret
- DRG: VPN headend at OCI end of the IPSec VPN
  - Same gateway used for IPSec VPN or FastConnect circuits to connect your on-premises network and VCN
  - Each VCN can have only a single DRG
- CPE: Actual VPN router in your on-premises network (hardware or software)

Oracle implements site to site VPN capability to the DRG via IPSEC. This enables the DRG when acting as a route target to forward traffic to remote sites.

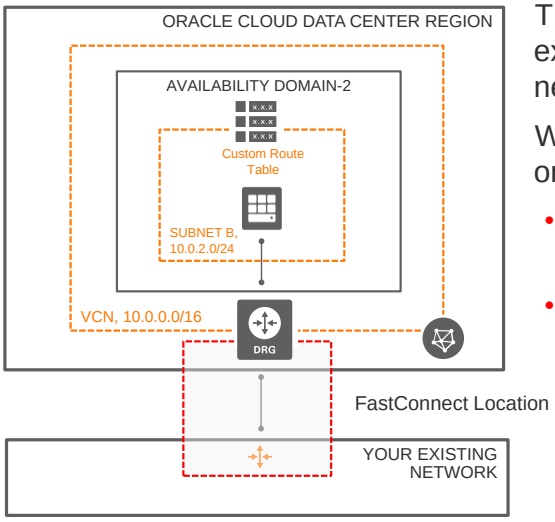
By default every request for a site to site ipsec connection will create 3 IP addresses that will connect to your CPE or your CPE's virtual IP address to create redundant and possibly load balanced VPN connections back to oracle cloud.

Currently routing via the IPSEC VPN tunnel is done with static routes, BGP support is coming but is not implemented as of yet.

Currently OCI VPN supports IKEv1 using a shared secret. DRG is the head end of the IPSEC VPN .

Each VCN can only have one DRG associated with it at a time.

FastConnect



The general concept of a connection between your existing network and your VCN over a private physical network instead of the internet.

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

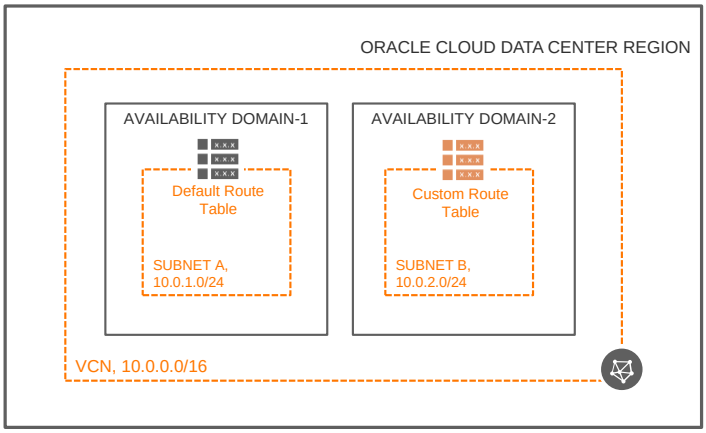
- Colocation: By co-locating with Oracle in a FastConnect location
- Provider: By connecting to a FastConnect provider

Fast connect provides private, dedicated bandwidth via MPLS via oracle partners who are collocated in our datacenters. With this you can link up to other clouds or to your on premise networks via these fast connect providers.

Fast connect is offered starting at 1gbps and is priced very low to encourage the use of these circuits. These circuits can scale up to 60gbps per second, possibly higher, depending on the provider. When using a partnered oracle fast connect provider circuits are provisioned quite quickly and again are priced such as to encourage their use. Routing is handled via BGP.

So basically this is a secure cost competitive option to connect your on premise networks with oracle cloud infrastructure.

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 (e.g. individual route rules). And you can create more of each kind of component in your cloud network (e.g. additional route tables).

DNS Choice

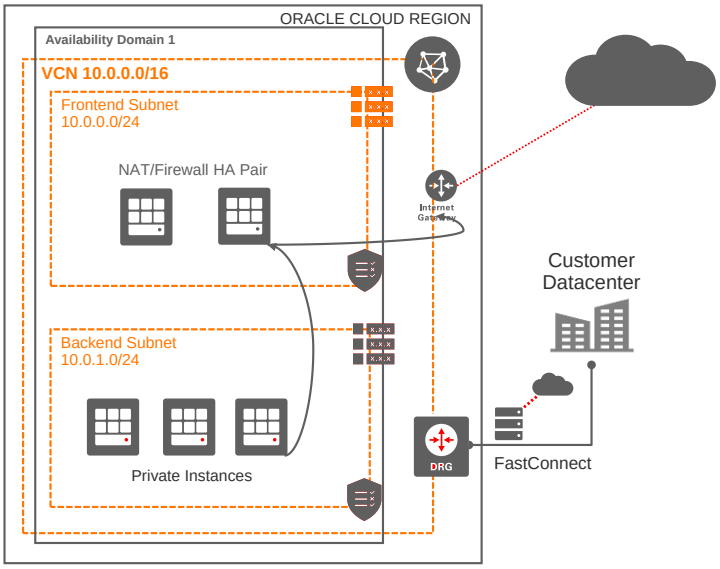
- The Domain Name System (DNS) enables lookup of other computers using host names.
- You choose the DNS for each subnet in the cloud network. Default Choice is Internet and VCN Resolver. This is an Oracle-provided option that includes two parts:
  - Internet Resolver: Lets instances use host names that are publicly published on the Internet. The instances do not need to have Internet access by way of either an IGW or an IPSec VPN DRG.
  - VCN Resolver: Lets instances use host names (which you can assign) to communicate with other instances in the VCN.
- Custom Resolver: Use your own DNS servers. These could be Internet IP addresses for DNS servers in your VCN, or DNS servers in your on-premise network, which is connected to your VCN by way of an IPSec VPN connection.

Instance FQDN: <hostname>.<subnet DNS label>.<VCN DNS label>.oraclevcn.com
(you can specify VCN, Subnet and hostname DNS labels)

DNS enables lookup using host names – remember this isn't public DNS service, but DNS for VCN and subnets. The default choice is Internet and VCN resolver which lets the instances use hosts 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.

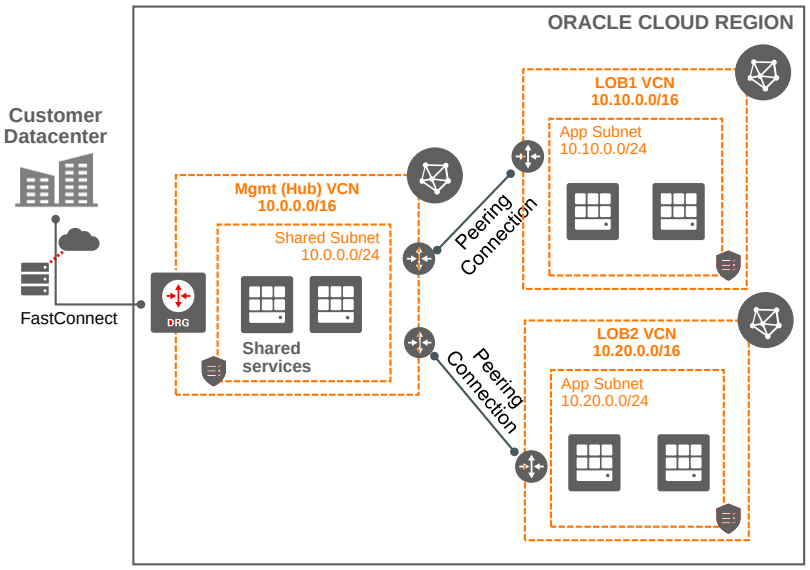
Advanced Features of VCN

Private IP as Route Target – NAT



- Ability to use a private IP as the target of a route rule in situations where you want to route a subnet's traffic to another instance.
- Use Cases
  - To implement NAT in VCN
  - To implement a virtual network function (such as a firewall or intrusion detection)
  - To manage an overlay network on the VCN, which lets you run container orchestration workloads

VCN Peering



- VCN peering is the process of connecting multiple virtual cloud networks (VCNs)
- VCN Peering is currently between same tenants and region
- With peering, instances in two VCNs communicate as if they are part of the same network
- Two VCNs in a peering relationship must not have overlapping CIDRs
- Use Cases: Business groups or LOBs may use separate VCNs, but shared services are deployed into the central VCN, which is peered with multiple other VCNs

A given VCN can have a maximum of 10 peerings at a time.

The two VCNs in the peering relationship must not have overlapping CIDRs. However, if VCN-1 is peered with three other VCNs, those three VCNs can have overlapping CIDRs with each other. You would set up the subnets in VCN-1 to have route rules that direct traffic to the appropriate peered VCN.

Multiple VNICs for Instances

- Ability to add secondary VNICs to an instance after it's launched.
- The secondary VNIC can be in a subnet in the same VCN as the primary VNIC or a different VCN.
- All the VNICs must be in subnets in the same Availability Domain as the instance
- Support for Both Bare metal and Virtual Machines (Linux Only)
- Use Cases
  - Bring your own Hypervisor (BYOH) scenarios
  - Connecting instances to multiple subnets
  - Connecting instances to multiple VCNs

There's a limit to how many VNICs can be attached to an instance, and it varies by shape.

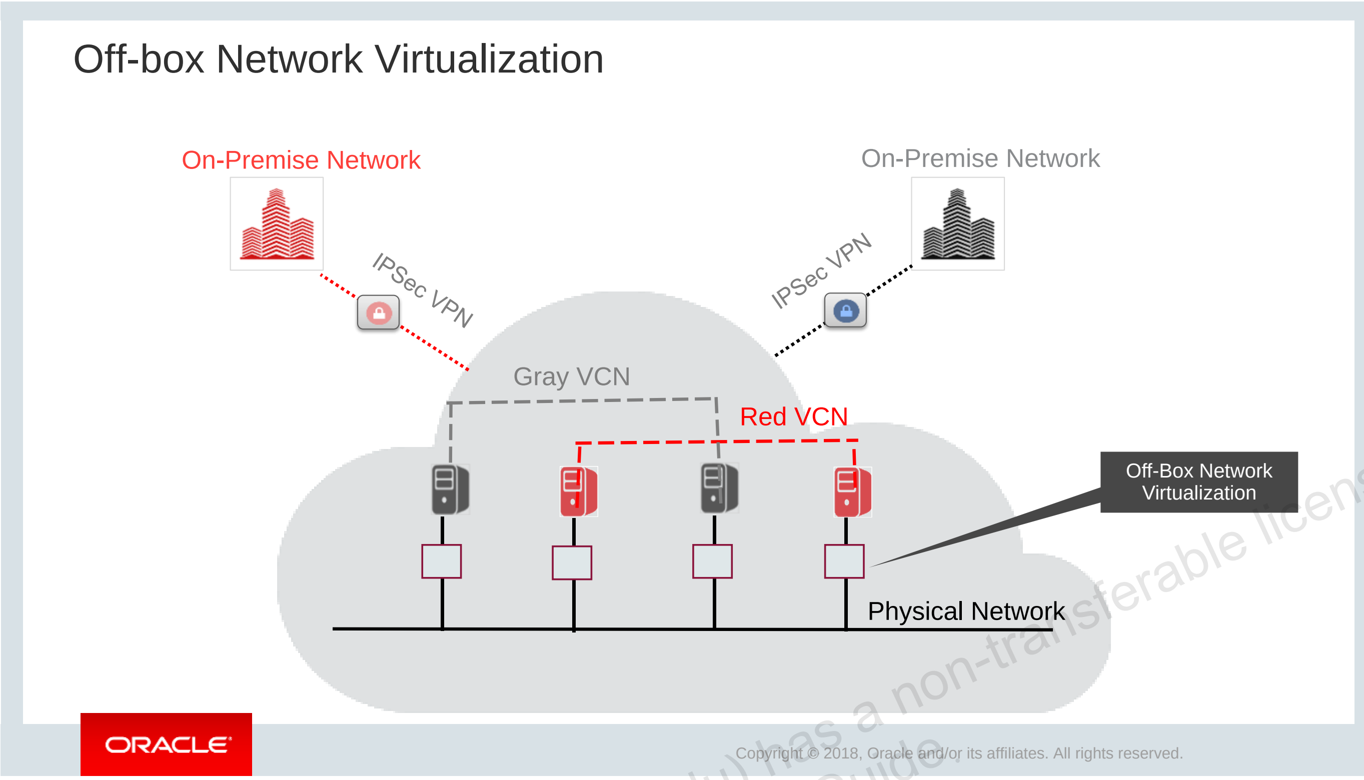
The instance's bandwidth is fixed regardless of the number of VNICs attached. You can't specify a bandwidth limit for a particular VNIC on an instance

Secondary IP Addresses

- Ability to add secondary IP addresses to an instance after it's launched.
- It can be added to either the primary VNIC or a secondary VNIC on the instance. The secondary private IP address must come from the CIDR of the VNIC's subnet.
- Can move a secondary private IP from a VNIC on one instance to a VNIC on another instance if both VNICs belong to the same subnet.
- Use Cases
  - Instance failovers
  - Run multiple services or endpoints on a single instance

You assign a secondary private IP to an instance. Then if the instance has problems, you can easily reassign that secondary private IP to a standby instance in the same subnet.

For example, you could have multiple container pods running on a single instance, and each uses an IP address from the VCN's CIDR. The containers have direct connectivity to other instances and services in the VCN. Another example: you could run multiple SSL websites with each one using its own IP address.



We have made some drastic changes into how virtual networking is done. We call if off-box virtualization. As the name implies, we pulled all the virtualization out into the network, including storage and network IO virtualization. Generally, this enables the next layer up – so we can take any physical form-factor and plug that in to 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 the classes of services and meet our goals around performance and security.

Summary

In this lesson, you should have learned how to:

- Describe key Virtual Cloud Network (VCN) concepts
- Manage your cloud network components, such as:
  - Route Table, Security List
  - Internet Gateway, Dynamic Routing Gateway
- Evaluate the different options of connecting to the Internet
- Describe key advanced Features of VCN
  - NAT, VCN Peering, Multiple VNICs and Secondary IPs

4

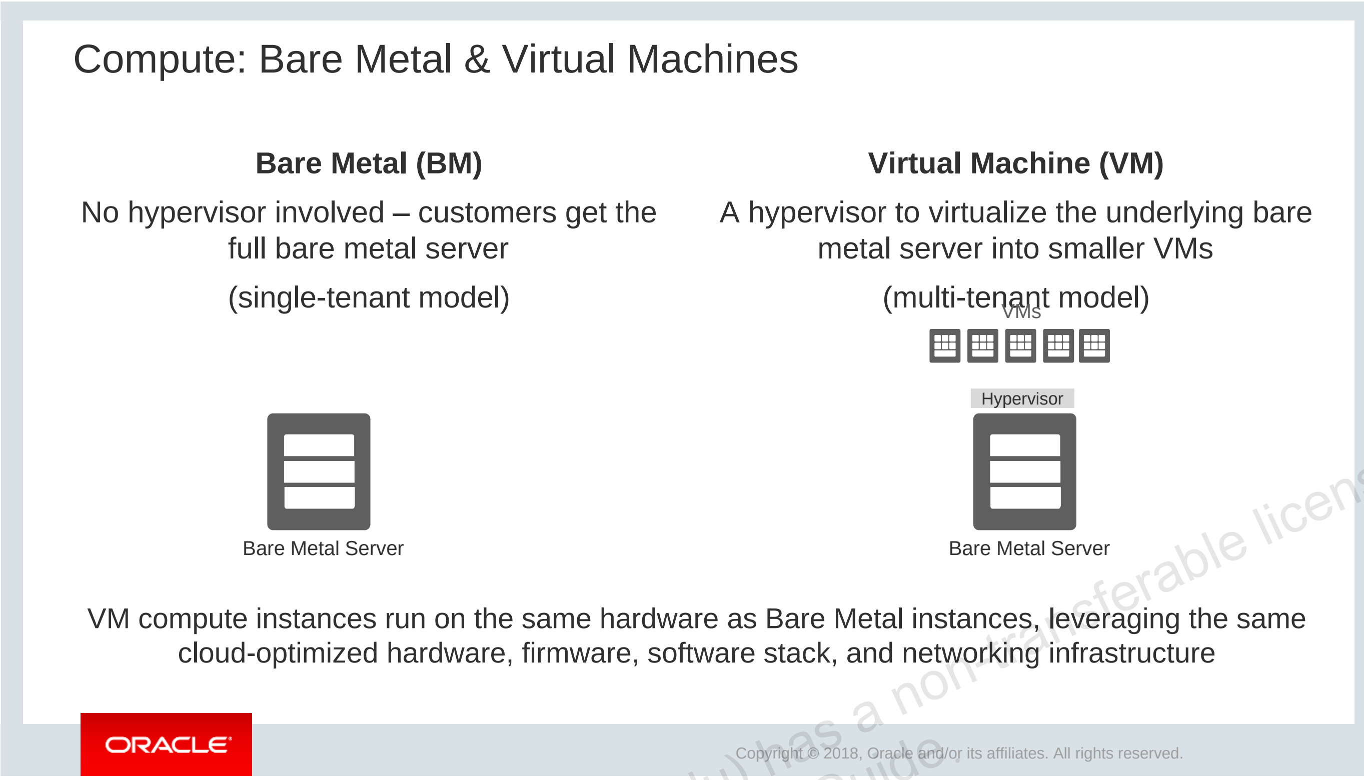
Compute Service

Objectives

After completing this lesson, you should be able to:

- Describe the OCI Compute Service
- Describe images, shapes, local storage
- Create and launch a compute instance
- Set up the credentials necessary for accessing the compute resource

In this lecture we'll dive deeper into the OCI compute service, talk about concepts and look at various features of the OCI compute service



First thing is that OCI is the only public cloud which supports bare metal and VMs using the same set of APIs, hardware, firmware, software stack and networking infrastructure. You can see the two models on the slide – Bare Metal instances are instances where customers get the full server. This is also referred to as 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 multi-tenant model. The advantage here is flexibility in regards 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.

While 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.

Available Shapes – Bare Metal

Shape	Instance type	OCPU	RAM (GB)	Local Disk (TB)	Network Bandwidth	Max vNICs	List Price (OCPU Per Hour)
BM.Standard2.52	X7 based Standard compute	52	768	Block Storage only	2 X 25 Gbps	24	$0.0638
BM.DenseIO2.52	X7 based Dense I/O compute	52	768	51.2 TB NVMe SSD	2 X 25 Gbps	24	$0.1275
BM.Standard1.36	X5 based Standard compute	36	256	Block Storage only	10 Gbps	16	$0.0638
BM.HighIO1.36	X5 based High I/O compute	36	512	12.8 TB NVMe SSD	10 Gbps	16	$0.102
BM.DenseIO1.36	X5 based Dense I/O compute	36	512	28.8 TB NVMe SSD	10 Gbps	16	$0.1275
BM.GPU2.2	X7-based GPU: 2 P100 NVIDIA GPUs	28	192	Block Storage only	2 X 25 Gbps	24	$1.275

- Pricing Info: https://cloud.oracle.com/infrastructure/pricing
- 2 x 25 Gbps implies two NIC cards with 25 Gbps bandwidth
- x7 shapes are only available in IAD region

Network bandwidth is based on expected bandwidth for traffic within a VCN.

X7 based shapes are available only in IAD.

X7 shapes have 2 in them and X5 have 1.

In case of standard VM instances, NVMe storage is not available. For all the shapes, Block Volume storage is offered.

The *Dense I/O* instances are configured with 28.8 TB of local NVMe storage and are ideal for extreme transactional workloads that work on large datasets and require low latency and high throughput, such as Big Data and High Performance Compute (HPC) applications.

Available Shapes – VMs (current gen)

Shape	Instance type	OCPU	RAM (GB)	Local Disk (TB)	Network Bandwidth	Max vNIC	List Price (OCPU Per Hour)
VM.Standard2.1	Standard	1	15	Block Storage only	1 Gbps	2	$0.0638
VM.Standard2.2	Standard	2	30	Block Storage only	2 Gbps	2	$0.0638
VM.Standard2.4	Standard	4	60	Block Storage only	4.1 Gbps	2	$0.0638
VM.Standard2.8	Standard	8	120	Block Storage only	8.2 Gbps	4	$0.0638
VM.Standard2.16	Standard	16	240	Block Storage only	16.4 Gbps	8	$0.0638
VM.Standard2.24	Standard	24	320	Block Storage only	24.6 Gbps	12	$0.0638
VM.DenseIO2.8	Dense I/O	8	60	6.4 TB NVMe SSD	8.2 Gbps	4	$0.1275
VM.DenseIO2.16	Dense I/O	16	240	12.8 TB NVMe SSD	16.4 Gbps	8	$0.1275
VM.DenseIO2.24	Dense I/O	24	320	25.6 NVMe SSD	24.6 Gbps	12	$0.1275

In case of standard VM instances, NVMe storage is not available. For all the shapes, Block Volume storage is offered.

The *Dense I/O* instances are configured with 28.8 TB of local NVMe storage and are ideal for extreme transactional workloads that work on large datasets and require low latency and high throughput, such as Big Data and High Performance Compute (HPC) applications.

Available Shapes – VMs – Previous Gen

Shape	Instance type	OCPU	RAM (GB)	Local Disk (TB)	Network Bandwidth	Max vNIC	List Price (OCPU Per Hour)
VM.Standard1.1	Standard	1	7	Block Storage only	Upto 600 Mpbs	2	$0.0638
VM.Standard1.2	Standard	2	14	Block Storage only	Upto 1.2 Gbps	2	$0.0638
VM.Standard1.4	Standard	4	28	Block Storage only	1.2 Gbps	2	$0.0638
VM.Standard1.8	Standard	8	56	Block Storage only	2.4 Gbps	4	$0.0638
VM.Standard1.16	Standard	16	112	Block Storage only	4.8 Gbps	8	$0.0638
VM.DenseIO1.4	Dense I/O	4	60	3.2 TB NVMe SSD	1.2 Gbps	2	$0.1275
VM.DenseIO1.8	Dense I/O	8	120	6.4 TB NVMe SSD	2.4 Gbps	4	$0.1275
VM.DenseIO1.16	Dense I/O	16	240	12.8 TB NVMe SSD	4.8 Gbps	8	$0.1275

In case of standard VM instances, NVMe storage is not available. For all the shapes, Block Volume storage is offered.

The *Dense I/O* instances are configured with 28.8 TB of local NVMe storage and are ideal for extreme transactional workloads that work on large datasets and require low latency and high throughput, such as Big Data and High Performance Compute (HPC) applications.

NVMe SSD Devices

- Locally attached SSDs are not protected.
- OCI provides no RAID, snapshots, backup capabilities for these devices.
- Customers are responsible for the durability of data on the local SSDs.

Instance type	NVMe SSD Devices
BM.HighIO1.512	4 drives = 12.8TB raw
BM.DenseIO1.512	9 drives = 28.8TB raw
VM.DenseIO1.4	1 drive = 3.2 TB raw
VM.DenseIO1.8	2 drives = 6.4 TB raw
VM.DenseIO1.16	4 drives = 12.8 TB raw

```
ubuntu@nvme:~$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0    0 46.6G  0 disk
├─sda1    8:1    0 46.5G  0 part /
├─sda14   8:14   0    4M  0 part
└─sda15   8:15   0  106M  0 part /boot/efi
nvme0n1 259:4    0  2.9T  0 disk
nvme1n1 259:5    0  2.9T  0 disk
nvme2n1 259:3    0  2.9T  0 disk
nvme3n1 259:6    0  2.9T  0 disk
nvme4n1 259:7    0  2.9T  0 disk
nvme5n1 259:8    0  2.9T  0 disk
nvme6n1 259:1    0  2.9T  0 disk
nvme7n1 259:0    0  2.9T  0 disk
nvme8n1 259:2    0  2.9T  0 disk
```

ORACLE

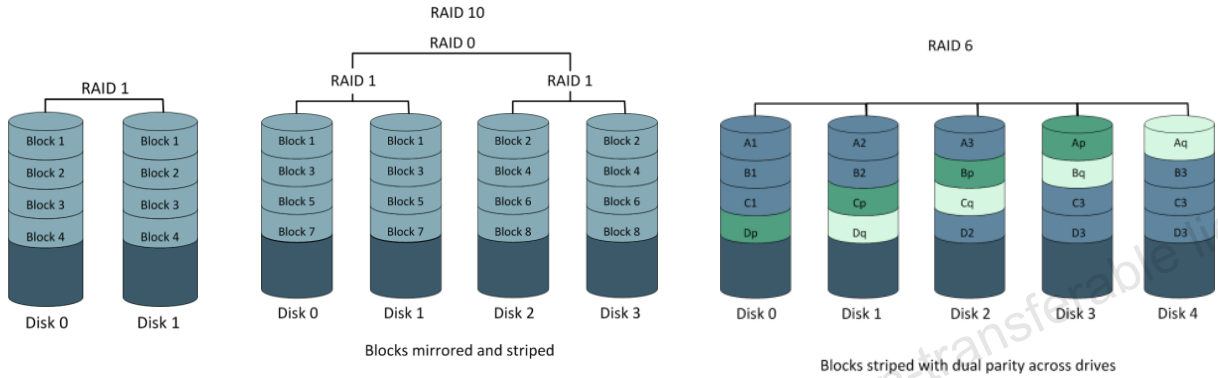
Some instance shapes in Oracle Cloud Infrastructure include locally attached NVMe devices. These devices provide extremely low latency, high performance storage that is ideal for big data, OLTP, and any other workload that can benefit from high-performance storage. Note that these devices are not protected in any way; they are individual devices locally installed on your instance. Oracle Cloud Infrastructure does not take images, back up, or use RAID or any other methods to protect the data on NVMe devices. It is your responsibility to protect and manage the durability the data on these devices.

Protecting NVMe SSD Devices

RAID 1: An exact copy (or mirror) of a set of data on two or more disks

RAID 10: Stripes data across multiple mirrored pairs. As long as one disk in each mirrored pair is functional, data can be retrieved.

RAID 6: Block-level striping with two parity blocks distributed across all member disks.



A protected RAID array is the most recommended way to protect against an NVMe device failure. There are three RAID levels that can be used for the majority of workloads:

RAID 1: An exact copy (or mirror) of a set of data on two or more disks; a classic RAID 1 mirrored pair contains two disks, as shown here.

RAID 10: Stripes data across multiple mirrored pairs. As long as one disk in each mirrored pair is functional, data can be retrieved

RAID 6: Block-level striping with two parity blocks distributed across all member disks

BM.DenseIO1.512 Options

- **RAID 6 across all nine SSDs**
  Single LUN with ~23.8TB of usable space that will survive the failure of any two devices
- **Four device RAID 10 and five device RAID 6 arrays**
  Results in two arrays with isolated I/O (data and log files) with 6.4TB and 9.6TB of usable space
- **RAID 10 array across 8 devices**
  Single LUN with ~12.8TB of space that will survive the failure of any one device and a hot spare
- **Two RAID 10 arrays of 4 devices each**
  Two LUNs, each with ~6.4TB of space and a global hot spare

```
$ sudo yum install mdadm -y

$ sudo mdadm --create /dev/md0 --raid-devices=9 --level=6 /dev/nvme0n1 /dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1 /dev/nvme4n1 /dev/nvme5n1 /dev/nvme6n1 /dev/nvme7n1 /dev/nvme8n1

$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf >> /dev/null
```

ORACLE

Images

A template of a virtual hard drive that determines the operating system and other software for an instance. Images can be Oracle-provided, custom, or BYOI. Oracle-provided images -

Image	Name
Oracle Linux 7 UEK Release 4	Oracle-Linux-7.x-
Oracle Linux 6 UEK Release 4	Oracle-Linux-6.x-
CentOS 7	CentOS-7-x
CentOS 6	CentOS-6.x-
Ubuntu 16.04 LTS	Canonical-Ubuntu-16.x-
Ubuntu 14.04 LTS	Canonical-Ubuntu-14.x-
Windows Server 2012 R2 – Bare Metal (BM)	Windows-Server-2012-R2-Standard-Edition-BM-Gen2, DenseIO Gen2, Standard Edition-BM
Windows Server 2012 R2 - Virtual Machine (VM)	Windows-Server-2012-R2-Standard-Edition-VM-Gen2, Standard Edition-VM
Windows Server 2008 R2 - Virtual Machine (VM)	Windows-Server-2008-R2-Standard-Edition-VM-

All Oracle-provided images include rules that allow only "root" on Linux instances or "Administrators" on Windows instances to make outgoing connections to the iSCSI network endpoint (169.254.0.2:3260) that serves the instance's boot and block volumes.

Oracle recommends that you do not reconfigure the firewall on your instance to remove these rules. Removing these rules allows non-root users or non-administrators to access the instance’s boot disk volume. Oracle recommends that you do not create custom images without these rules unless you understand the security risks.

Custom Images

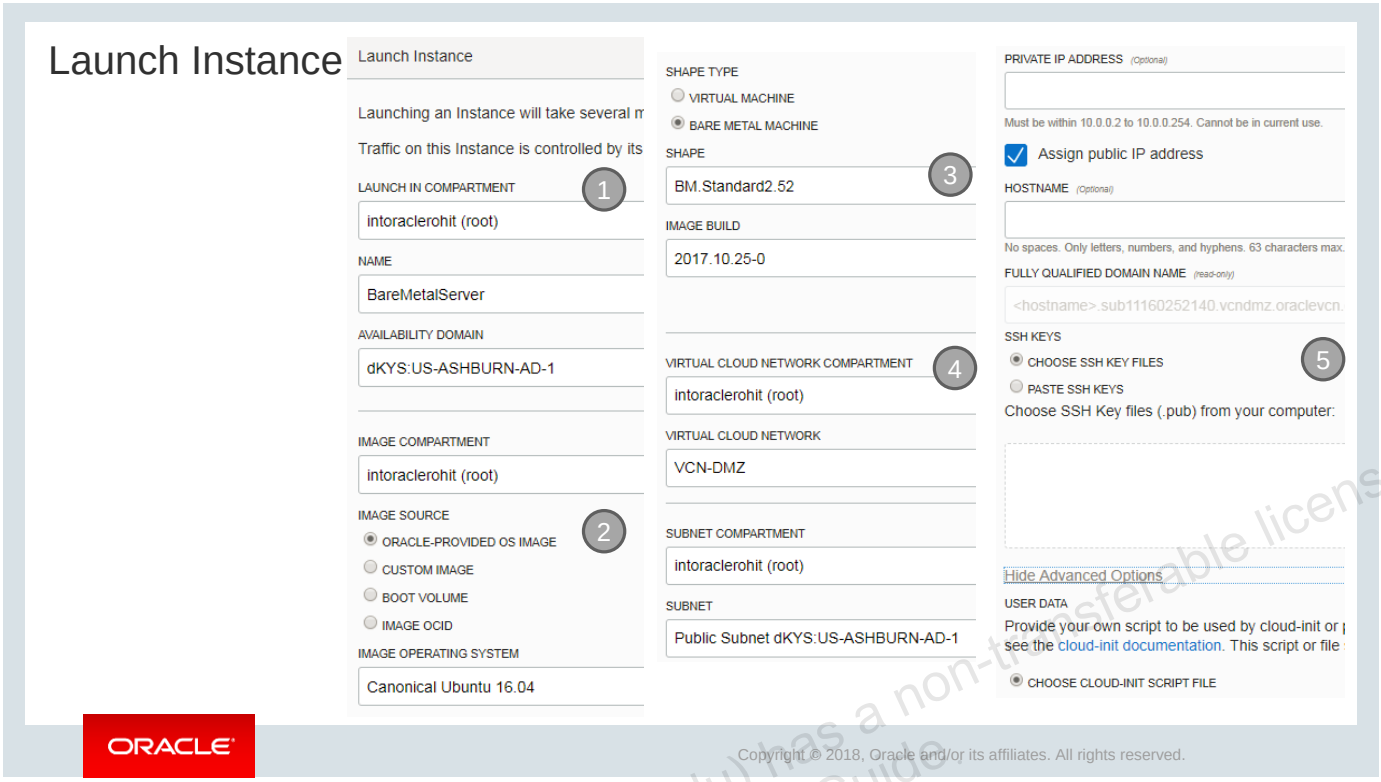
- Possible to create a custom image of an instance's boot disk and use it to launch other instances.
- Instances you launch from your image include customizations, configuration, and software installed when you created the image.
- When you create an image of a running instance, the instance shuts down and remains unavailable for several minutes. When the process completes, the instance restarts.
- Custom images do not include the data from any attached block volumes.
- Custom images cannot be > 50 GB in size.
- Custom images cannot be downloaded or exported.
- Support Generalized and Specialized images for Windows
  - Generalized image - generalized OS disk, cleaned of computer-specific information.
  - Specialized image - OS disk that is already fully installed, and a copy of the original BM or VM

Launching a Compute Instance

The steps to launch a compute instance are:

1. Choose a Compartment.
2. Specify image source and OS.
3. Choose a form-factor (VM/BM) and shape.
4. Create a Virtual Cloud Network (choose one if already created) and subnet.
5. Launch a compute instance using the Key Pair (created earlier).
6. Optional – cloud-init script, public IP or not, assign private IP, hostname.

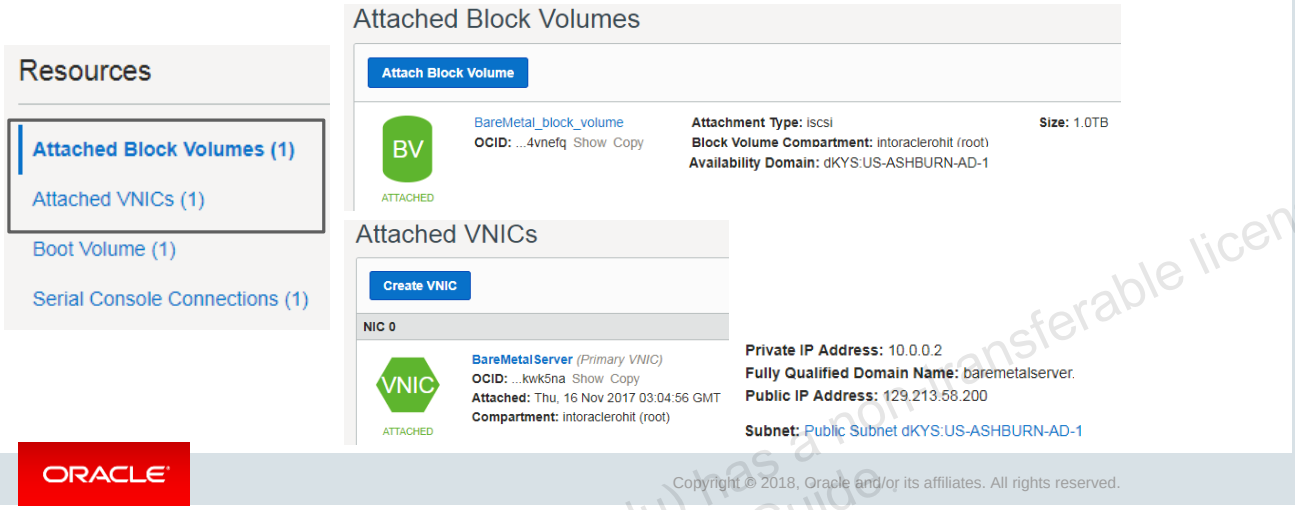
The creation of a compute instance is referred to as Launching an Instance. To create an instance irrespective of the type of image, you must follow the sequence of steps. In the previous lessons you have already gained familiarity with the compartment and virtual network.



Instance (running)

Block Volumes and Secondary vNICs

- Block volumes are high-performance iSCSI LUNs that can be used to provide persistent and durable storage for instance.
- Every instance has >1 vNIC. Secondary vNICs can be added for BYOH, network appliances or if the instance needs to connect to multiple VCNs.



Oracle Cloud Infrastructure offers high-performance remote block (iSCSI) LUNs that are redundant and can be backed up

Boot Volume and Serial Console Access

- A boot volume is created automatically when you launch a compute instance. You can detach and keep the boot volume at instance termination and instantiate a new instance from it.
- Serial console access to instances is provided to remotely troubleshoot malfunctioning instances.

A boot volume is created automatically when you launch a compute instance, and instance becomes available within a minute.

Compute service provides serial console access to instances, enabling you to remotely troubleshoot malfunctioning instances such as instance not starting up properly.

Instance life cycle

- Start – Restarts a stopped instance. After the instance is restarted, the Stop action is enabled
- Stop – Shuts down the instance. After the instance is powered off, the Start action is enabled
- Reboot – Shuts down the instance, and then restarts it
- Terminate – Permanently delete instances that you no longer need (vNICs and volumes are auto detached; boot volume is deleted, but you can preserve it).
- Resource Billing
  - Standard VM and BM instances, billing pauses in a STOP state
  - High I/O BM instances and dense I/O BM and VM instances, billing continues even in STOP state

Summary

In this lesson, you should have learned how to:

- Describe Compute Service
- Describe images, shapes, local storage
- Create and launch a compute instance
- Set up the credentials necessary for accessing the compute resource

5a

Object Storage Service

Objectives

After completing this lesson, you should be able to:

- Describe concepts, use cases, and Tiers of Oracle Object Storage Service
- Manage buckets and objects
- Explain the features
  - Upload multipart objects to object storage
  - Amazon S3 Compatibility API

As part of this lesson, we will go over major concepts related to Object storage Service, what are the basic tiers available to the end users, and also learn how to manage buckets and objects in an object storage.

At the end, we will also talk about Object Storage multi part upload feature and AWS S3 Compatibility API.

Object Storage Service

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

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, HTTP PUT methods.

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 object storage is completely independent of the a 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

- **Big Data/Hadoop Support**
  - Use as a primary data repository for big data
  - **HDFS connector** provides connectivity to various big data analytic engines like Apache Spark and MapReduce
- **Content Repository**
  - Highly available and durable content repository for data, images, logs, and 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, e.g. pharmaceutical trials data, genome data, and Internet of Things (IoT)

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

The 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

---

Lets talk about common use cases for object storage.

Object storage service offers a scalable storage platform which enables you to not only store large data sets but also operates 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.

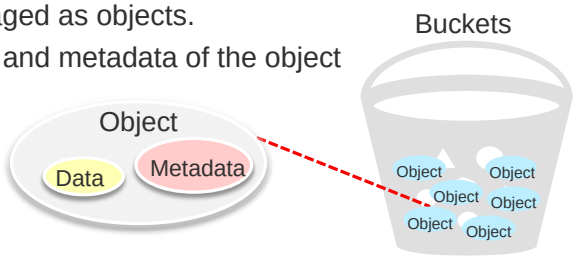
Another common use case is back up or archive, where data is typically written once and read many times. Or the data is to archived for longer periods of time. The durability and low-cost characteristics of object storage tiers make it a perfect platform to store long living data.

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.
  - Each object is composed of the object itself and metadata of the object
- **Bucket**
  - A logical container for storing objects
  - Each object is stored in a bucket
- **Namespace**
  - Each tenant is associated in one default namespace that spans all compartments
  - Buckets names within a namespace are unique
  - Buckets and objects exist in flat hierarchy,
- **Compartment**
  - Buckets can only exist in a one compartment



Within the object storage service, there are some key resources.

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

A bucket is a logical container created by the user, and can contain 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 buckets names are unique. Buckets and objects in a namespace are in flat hierarchy, but you can create directory structures for your ease.

Object Storage Service Features

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

So what are the object storage top features.

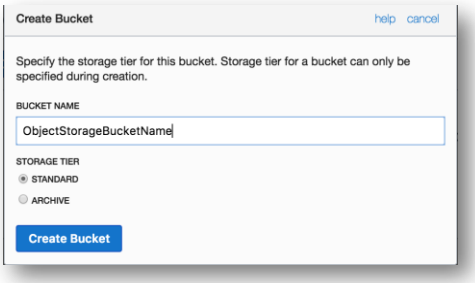
Object Storage has a strong consistency mode, which means when there is a read request made, you are guaranteed a consistent most recent copy of the data that has been completely written in the system. We have a high performance network where lookups for data are fast and consistent.

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 then 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
  - 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
  - Minimum retention requirement for Archive Storage is 90 days
  - Objects need to be restored before download
  - Archive Bucket can't be upgraded to Standard storage tier
  - Time To First Byte (TTFB) after Archive Storage restore request is made: 4 Hours



Currently object storage provides two different storage tiers options when creating a bucket.

Standard storage tiers is for the 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. Once a standard storage bucket is created, it can't be downgraded to an archive storage tier.

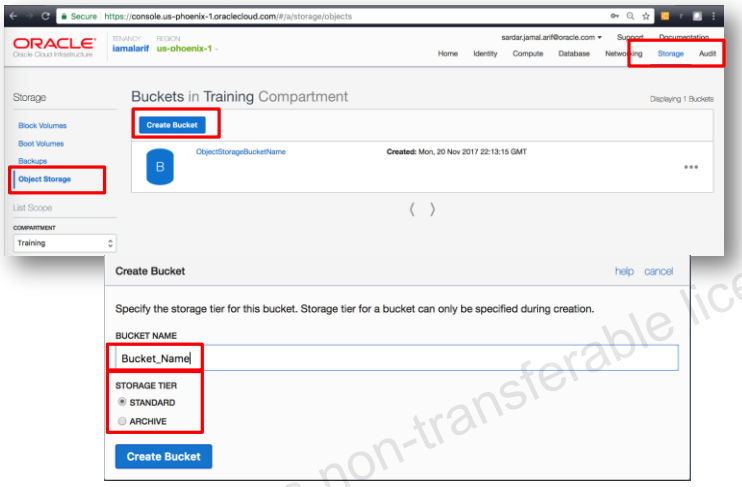
The other option is for Archive Storage tier. This is for data that is not frequently accessed, but must be preserved for longer periods of time.

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.

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

Managing Buckets and Objects (1)

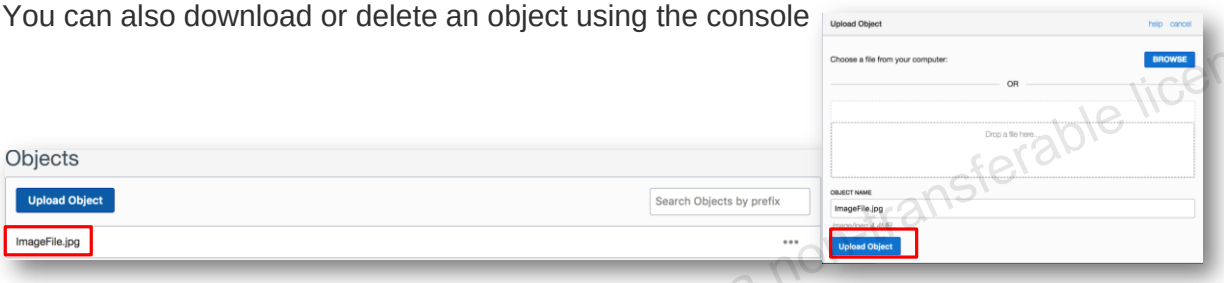
- Then navigate to **Storage > Object Storage**
  - **Create bucket.**
    - Name
    - Storage Tier
      - Standard
      - Archive



Creating a bucket is a simple process. Login into the tenant and choose a region where you want to create a bucket. Navigate to the Storage Service and choose object storage. Within Object Storage, click on Create Bucket. In create bucket window, provide a name (remember bucket names are unique within a namespace or tenant) and choose a storage tier , standard or archive. And click create bucket to create a bucket in that region.

Managing Buckets and Objects (2)

- To upload an object:
  - Click the bucket name. A list of objects in the bucket is displayed.
  - Click Upload Object.
    - Then click Browse, navigate to and select the file you want to upload, and then click Open.
    - If you want to change the name of the object, edit the name in the Object Name field.
  - Click Upload Object. The object is uploaded and displayed in the list of objects.
- You can also download or delete an object using the console



For uploading an object. Click the bucket you just created, and it shows you the list of all objects in that bucket. It also shows additional info of the bucket, like if it's a public bucket or a private one, what is the namespace of the bucket, etc. Now click on upload object, and browse to the file you want to upload. Here you can also change the name of the object. Click Upload object and the object is uploaded.

It is possible to also delete or download objects as well.

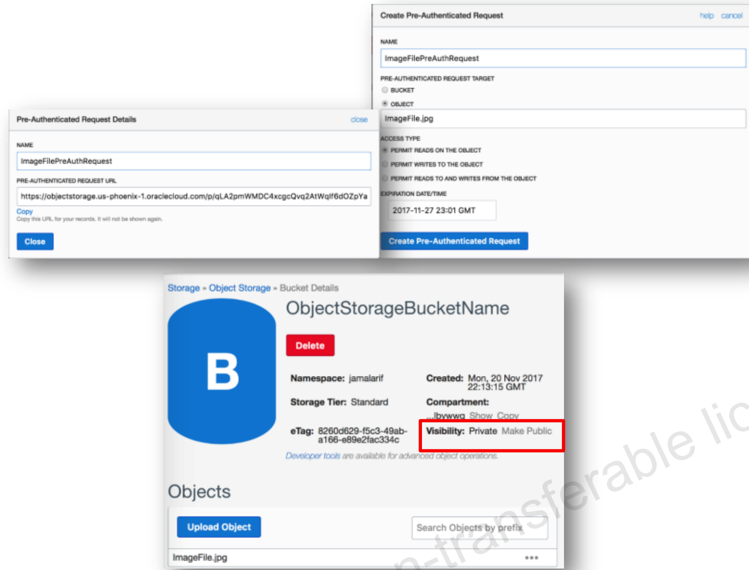
Managing Access and Authentications

Pre-Authenticated Requests

- Provides a way to let users access a bucket or an object without having their own credentials
- Can access via a unique URL

Public Buckets

- By default, access to a bucket and its contents requires authentication
- Support of anonymous, unauthenticated access to a bucket via Public Buckets



Pre-authenticated requests allow users to access the bucket or its objects via a unique time bound URL. For e.g., if you want to create a pre-authenticated request for the file that we just uploaded in the last step, you will be provided with a unique URL and you will also choose the time till it can remain active. You can create, delete, or list pre-authenticated requests using the Console, using the CLI, or by using an SDK to access the API.

Another concept is of the public buckets. When ever you create a bucket, it is by default private. However we support anonymous, unauthenticated access to a bucket via public buckets. You can make a bucket public by enabling read access to the bucket. Users would then be able to list the contents of a public bucket or download bucket objects.

Managing Multipart Uploads

- Object Storage Service supports multipart uploads for more efficient and resilient uploads, especially for large objects.
- You can use the retry feature to upload only the failed upload.
- You can use multipart upload RESTAPI calls or the Java Software Development Kit (SDK) to manage multipart uploads, but not the Console.
- Oracle recommendation: multipart upload for objects greater than 100 MiB.
- Maximum size for an uploaded object is 10 TiB. Object parts must be no larger than 50 GiB

Since a bucket can have unlimited number of objects. One common ask is how to upload such huge chunks of files. For that Object storage services offers a multipart upload feature to efficiently upload large files.

You can segment your file into smaller parts and then upload these parts in parallel. Object storage supports up to 10,000 parts per file and a part can be as small as 10 MB.

Using multipart upload not only significantly reduces the upload time but also minimizes the impact of network failures. If a part fails to upload for any reason you can retry to upload the parts instead of having to upload the entire file.

This is currently offered via API and SKD. Oracle recommends that for any data that is larger than 100MB, we should use multipart. The maximum size for an uploaded object is 10 TiB and the Object parts must be no larger than 50 GiB.

Object Storage: Amazon S3 Compatibility API

- Set of Object Storage APIs that let you build products and services that interoperate with other storage services, such as Amazon S3
- Following highlights the differences between Object Storage API and S3 Compatibility API
  - **Compartments**
    - any buckets created using the Amazon S3 Compatibility API are created in the root compartment of the Oracle Cloud Infrastructure tenancy.
  - **Global bucket namespace**
    - Object Storage doesn't use a global bucket namespace.
    - Bucket names must be unique within the context of a namespace, but bucket names can be repeated across namespaces.
  - **Encryption**
    - Encrypted data at rest by default.
    - Encryption can't be turned on or off using the API.
  - **Object Level Access Control Lists (ACLs)**
    - Oracle Cloud Infrastructure does not use ACLs for objects. Instead, IAM policies are used to manage access to compartments, buckets, and objects.

This is one of the new features of the Object Storage Service. Customers can now use the Amazon S3 Compatibility API in addition to the OCI Native API to manage OCI Object Storage. So any tools and applications that integrate with the Amazon S3 API can be migrated to the Oracle Cloud Infrastructure without having to rip and replace APIs. This also helps promote an open cloud platform where the customer's data will never be locked-in.

There are differences which are highlighted and should be considered during design.

Summary

In this lesson, you should have learned how to:

- Describe concepts, use cases and Tiers of Oracle Object Storage Service
- Manage buckets and objects
- Explain features
  - Upload multipart objects to object storage
  - Amazon S3 Compatibility API

That sums up our Object storage lesson. Provided is a quick summary of what we learned in this lesson.

5b

Block Volume Service

Objectives

After completing this lesson, you should be able to:

- Create, attach, configure, and mount block volumes
- Back up and restore block volumes
- Block Volume Cloning Service
- Boot Volumes

In this lesson, you will learn about the OCI block volume features, managing a block volume and its recent feature announcements.

Block Volume Service:

- Block storage operates at the raw storage device level and manages data as a set of numbered, fixed-size blocks using protocols such as iSCSI.
- Block Volume Service lets you dynamically provision and manage block storage volumes.
- You can create, attach, connect, and move volumes, as needed, to meet your storage and application requirements.
- Typical Scenarios
  - Expand an Instance's Storage
  - Persistent and Durable Storage
  - Instance Scaling

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 lifecycle. Using this service you can create block volumes and attach them to your compute instance.
The Oracle cloud infrastructure block volumes service delivers a simple, scalable block volume service that fulfills all your work load performance needs. The service let's 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.

Once attached and connected to an instance you can use the volume like a regular hard drive. The service also let 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. Once any compute instance are launched, you can attach block volumes to the instance, increasing the instances 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 the date 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 to scale the instance CPUs keeping the data secure in the process. We will talk about it later in the lesson.

One important thing to remember is that Block volume is associated always with an instance, therefore its always created with in an availability domain. That is why it's an AD construct.

Block Volume Service

- Elastic block storage volumes are configurable from 50 GB to 16 TB.
- Total block storage for an instance up to .5 PB (16 TB x 32 volumes).
- The service offers 60 IOPS per GB and scales linearly up to 250 K IOPs.
- Data is encrypted at rest in both volumes and backups.
- All volumes are automatically replicated to protect against data loss.

So continuing some additional features of the block volume service. Block volume sizes start from 50GB and can go upto 16TB with 1GB increments. And you can attach a maximum of 32 block volumes to an instance.

From performance perspective, IOPS and bandwidth scale linearly per GB volume size up to the service maximums. So for instance, a 16 TB volume: gives 25,000 IOPS at 320 MB/s throughput. The performance you observe for a block volume is basically limited by the network bandwidth of your compute instance.

Block volumes are highly durable and secure. Multiple copies of data are stored redundantly across multiple storage servers, and there is a backup service available for taking block volumes backups.

Moreover Your data is encrypted at rest on a block volume. Backups that you create are also encrypted for increased security.

Block Volume Service Components

The components required to create a volume and attach it to an instance are briefly described as follows:

- **Instance:**
  - An Oracle Cloud Infrastructure compute host
- **iSCSI:**
  - A TCP/IP-based standard used for communication between the instance and the attached volume
- **Volume:**
  - **Block Volume:** A detachable block storage device that allows you to dynamically expand the storage capacity of an instance
  - **Boot volume:** A detachable boot volume device that contains the image used to boot a compute instance.

Lets talk about the block volume service components.

An instance is a compute resource in OCI (VM or Baremetal)

As discussed earlier, block volumes uses the standard iSCSI protocol for communication between the host and the block volumes.

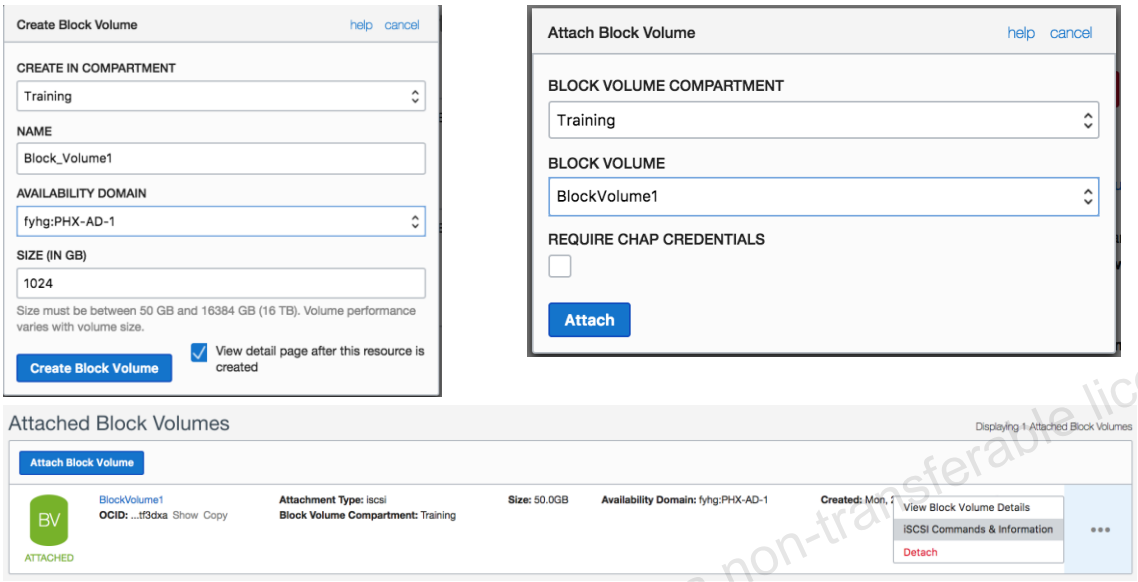
The block volume has two types of block volumes!

Block volume is a detachable block storage device that you attach to an instance once the instance is launched.

The boot volume is a recently announced feature of block volume service. When ever you launch a baremetal or virtual instance with an oracle provided image or custom image, a new boot volume is automatically launched in the same compartment containing the image used to boot that instance. This boot volume is associated with the instance which is launched.

We will talk more about boot volume later in the deck.

Creating and Attaching a Block Volume Using the Console



Lets take a look at what are the steps of managing a block volume service.

You log in to the console, and navigate to the storage service tab. There you choose block volume, and click on create block volume

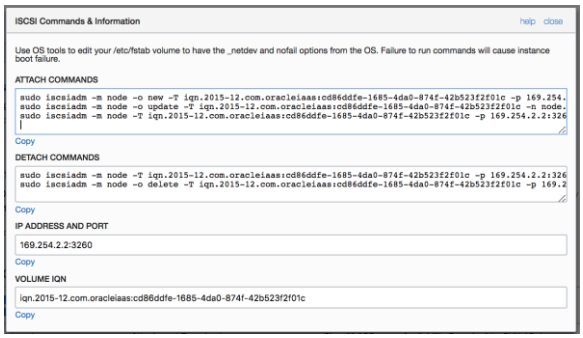
While creating block volume, you have to provide the name of the block volume, the compartment you are creating it in, the AD, and the size of the block volume.

Compartment and AD are important, because it is tightly coupled with an instance that you are going to attach it to. And compute instances are launched with in specific AD. Therefore you also choose during creating which AD and compartment you want to create this block volume in.

Once created, you navigate to the compute instance tab, and attach it.

Managing Block Storage Volumes

- Use iSCSI protocol to connect, configure, and mount the block volume to use it like a normal hard drive.
- On the console, click the Actions icon (…) on attached block volume, and click iSCSI Commands and Information.
- Use that information to configure and mount the volume to the instance.



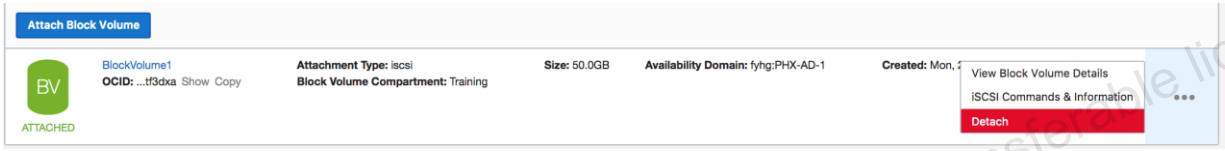
Once the volume is attached, you log on to the instance and use the iscsiadm command-line tool to configure the `iSCSI` connection in your instance. Once block volume is configured, you can mount it and use it like a regular hard drive.

Block volumes also work well with windows instances, so if a windows instance is present, a different set of copy-paste iscsi commands will be provided.

By default these commands persist the connection, so they should be there after a reboot and not need re-attaching.

Detaching and Deleting Block Volumes

- When an instance no longer requires a block volume, you can disconnect and then detach it from the instance without any loss of data.
- When you attach the same volume to another instance or to the same instance, DO NOT FORMAT the disk volume. Otherwise, you will lose all the data on the volume.
- When the volume itself is no longer needed, you can delete the block volume.
- You cannot undo a delete operation. Any data on a volume will be permanently deleted once the volume is deleted.



Lets talk more on what are some of management options available for block volumes.

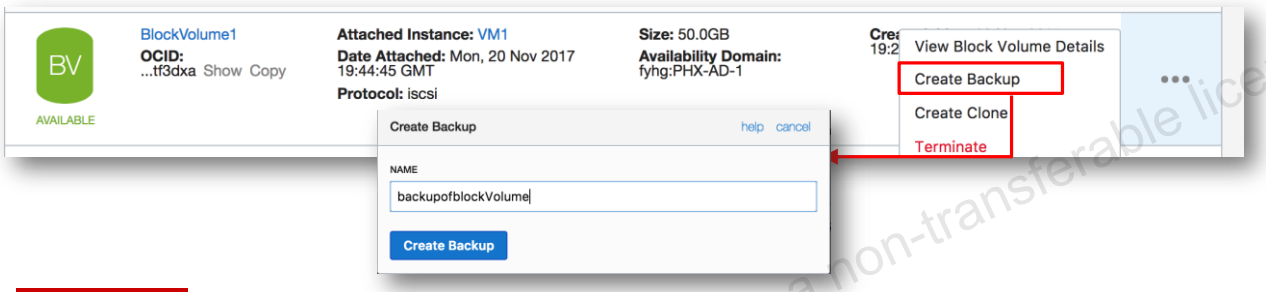
Like we discussed earlier, you can detach a Block Volume once its no longer needed without any loss of data. You can detach, and reattach to a different instance.

Make sure when you reattach, not to format the disk as it would delete the data on the disk.

You can also delete the block volumes. The delete operation is final and can't be undone.

Backup and Restoration

- Backup is a complete point-in-time complete snapshot copy of your block volumes.
- Backups are encrypted and stored in the Object Storage Service, and can be restored as new volumes to any Availability Domain within the same region.
- This capability provides you with a spare copy of a volume and gives you the ability to successfully complete recovery within the same region.
- Restoration of volume from a backup takes less than a minute regardless of the volume size.



ORACLE

Block Volumes allows you to take snapshots and point in time backups any time. And an interesting thing about volume backups is they are differential and automatically managed, so a number of volume snapshot backups will chain to form the most recent backup, saving you space, time and cost. And the service manages these incremental backups for the users.

When it comes time to restore, these backup/snapshots are automatically maintained so that any restorations will match the volume state at the time of the last snapshot. Users don't have to keep track of those incremental backups.

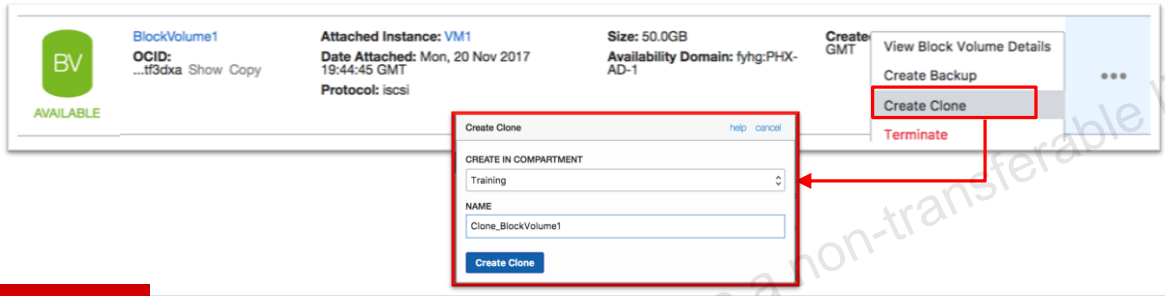
Backups go into the backup section of the storage service - which is based on object storage –

And they can be restored to the same or a different availability domain and re-attached to the original instance or a new one.

I do want to mention here that this would not be the right mechanism to use to backup any databases. The database instances do not use block volumes for their data. They use local and object storage for backups.

Volume Cloning

- A block volume cloning feature which creates a point-in-time direct disk-to-disk deep copy of a source volume without a backup
- A clone can only be created in the same AD with no need of detaching the source volume before cloning it.
- Volume Cloning and Backup operation are mutually exclusive operations.
- A clone can be attached and used as regular volume when its lifecycle state changes from ”PROVISIONING” to "AVAILABLE", usually within seconds (At this time, the data is being copied in the background).



ORACLE

Block Volume service also allows you to clone a block volume. This enables you to make a copy of an existing block volume without needing to go through the backup and restore process

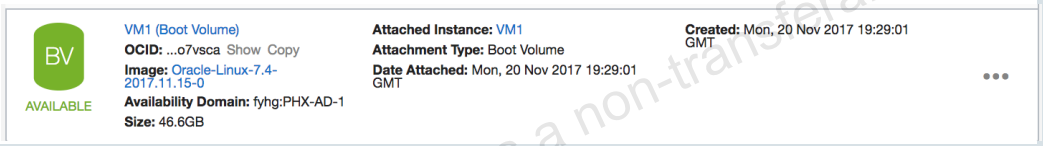
And it’s a point-in-time direct disk-to-disk deep copy of the source volume, so all the data that is in the source volume when the clone is created is copied to the clone volume. Any subsequent changes to the data on the source volume are not copied to the clone. Since the clone is a copy of the source volume, you cannot change the size, it will be the same size as the source volume

You can only create a clone for a volume within the same region, Availability Domain and tenant. You can create a clone for a volume between compartments as long as you have the required access permissions for the operation.

Boot Volumes: Manageable Boot Disks for Compute Instances

All Oracle Cloud Infrastructure compute instances are now launched with manageable boot volumes, provided by the Block Volumes service, as their system boot disks. Boot Volumes offer the following features:

- Ability to preserve your boot disk content by keeping it when you terminate a compute instance
- High durable boot disks by creating multiple replicas across the AD
- Compute instance scaling via boot volumes
- Faster launch times for Linux and Windows VMs
- All boot volumes are encrypted by default
- Ability to troubleshoot and repair your boot disks and OS images by using Block Volumes Detach/Attach features



ORACLE

So we briefly discussed about boot volumes early.

When any instance is launched (virtual machine or a bare metal) on an oracle provided image or a custom image, a new boot volume for the instance is created in the same compartment. That boot volume is associated with that instance until you terminate the instance. When you terminate the instance, you have the option of preserving the boot volume and its data. This feature gives you more control on the boot volumes of your compute instance. For instance .

It gives you the ability to preserve your boot disk content by keeping it when you terminate a compute instance: You can use the preserved boot volume for new instance creation.

Just like block volumes are replicated across ADs, the boot volumes are also highly durable as they are replicated across ADs automatically.

Boot volumes can also help in instance scaling. Since you can preserve the boot volume when terminating an instance, the preserved boot volume can be used with a new instance of different shape, which can have more OCPUs.

The launch times are much faster than earlier.

All boot volumes are encrypted at rest like block volumes

And it also helps us in troubleshooting or repairing boot disks.

To use boot volumes, there is nothing special that one needs to do. Moving forward all instances that are launched will be done using boot volumes having all the features we talked earlier.

Summary

In this lesson, you should have learned how to:

- Create, attach, configure, and mount block volumes
- Back up and restore block volumes
- Block Volume Cloning Service
- Boot Volumes

That sums up our block storage service. A quick summary is provided on what we learned.

5c

Data Transfer Service

Objectives

After completing this lesson, you should be able to:

- Describe the need for Data transfer service
- Create Transfer Job, Transfer Device, and Transfer package

Data Transfer Service

- Data Transfer service is an offline data transfer solution that lets you migrate large volumes of data to Oracle Cloud Infrastructure.
- Using Data Transfer service you can transfer hundreds of TB of data on commodity hard disk drives (HDDs) and ship these drives to an Oracle transfer site.
- Data Transfer service provides significantly faster data upload compared to over-the-wire data transfer.

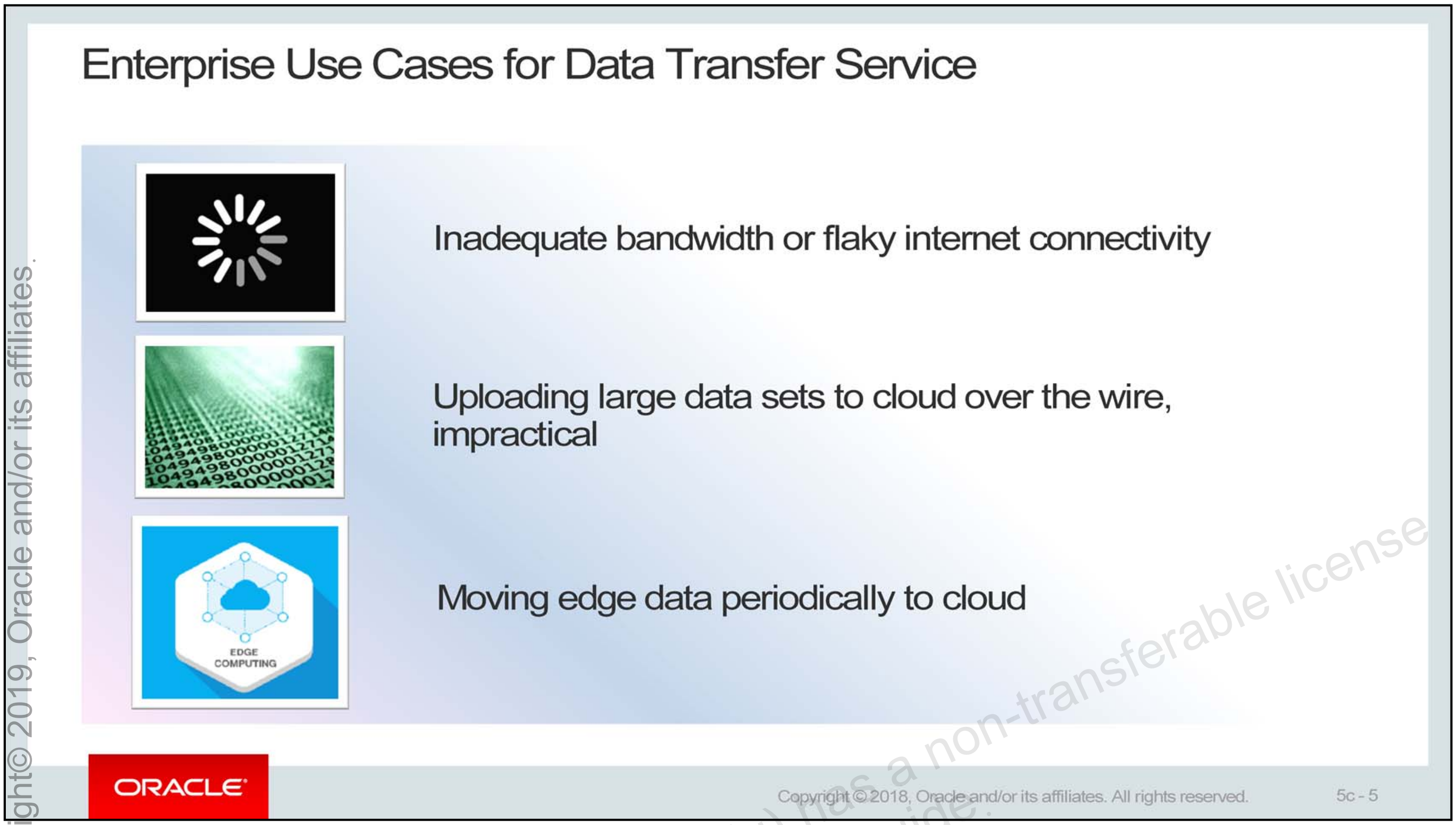
Interstate Highway Bandwidth >> Internet Bandwidth

	Approximate Data Upload Time				
Dataset Size	**10 Mbps**	**100 Mbps**	**1 Gbps**	**10 Gbps**	**Using DTS**
10 TB	92 Days	9 Days	22 Hrs	2 Hrs	**~1 week**
100 TB	1,018 Days	101 Days	10 Days	24 Hrs	**~1 week**
500 TB	5,092 Days	509 Days	50 Days	5 Days	**~ 1 week**
1 PB	10,185 Days	1,018 Days	101 Days	10 Days	**~ 2 weeks**

Sweet Spot for DTS

ORACLE

You can use data transfer service for moving large and small datasets. When you should really use data transfer service depends on how large your dataset is and what your internet connectivity looks like. It doesn't make sense to use the data transfer service all the time. If you are trying to move 10 TB of dataset and you have 10 Gbps connectivity, say via FastConnect, It will take two hours to move the dataset. Having said that, there is a sweet spot for data transfer service as shown in the picture. Using data transfer service you can move your 10 TB or even a PB within 2 weeks' time.



Let's talk about the use cases for this service. The first one is obvious. You have a large amount of data to move but don't have the bandwidth to do so. The second use case is that you have a good bandwidth but can't dedicate the full bandwidth for data transfer due to other application workload that requires bandwidth in the datacenter. In that case, data transfer service should be evaluated. The last one is to move edge data periodically to the cloud. If you have multiple remote sites with average or poor internet connectivity, you move data from edge to the cloud.

Typical Enterprise Data Transfer Requirements

- Security
- Speed of Data Transfer
- Chain of Custody
- Ease of Use

When it comes to data migration, there are four high level requirements that customers ask for. First is the data security. When we are moving their data to the cloud, we need to provide assurance to the customer that their data is secured at all times. Second one is related to the performance. Along with security, they want data to be transfer as quickly as possible. Third requirement is chain of custody. They always want to know where data is. They always want to know at what stage data transfer is so they can track where the data is and when its going to get in the cloud. And finally ease of use. If we have all, Security, Speed and chain of custody but if its not easy to use, then it defeats the purpose.

OCI Data Transfer Service

- Migrate data to the cloud using off-the-shelf hard drives
  - Support SATA II/III 2.5“, 3.5" or USB 2.0/3.0 drives (coming soon)
- Easy to use –
  - Use service console or Data Transfer Utility
- Fully automated with self-service
  - Self service workflows. No SRs required to initiate or manage data transfer.
- Full visibility into chain of custody of data
  - Status maintained and surfaced at every stage of the transfer process
- Data fully encrypted at each stage of the process
  - AES 256 bit encryption (dm-crypt and LUKS utilities used to encrypt device)

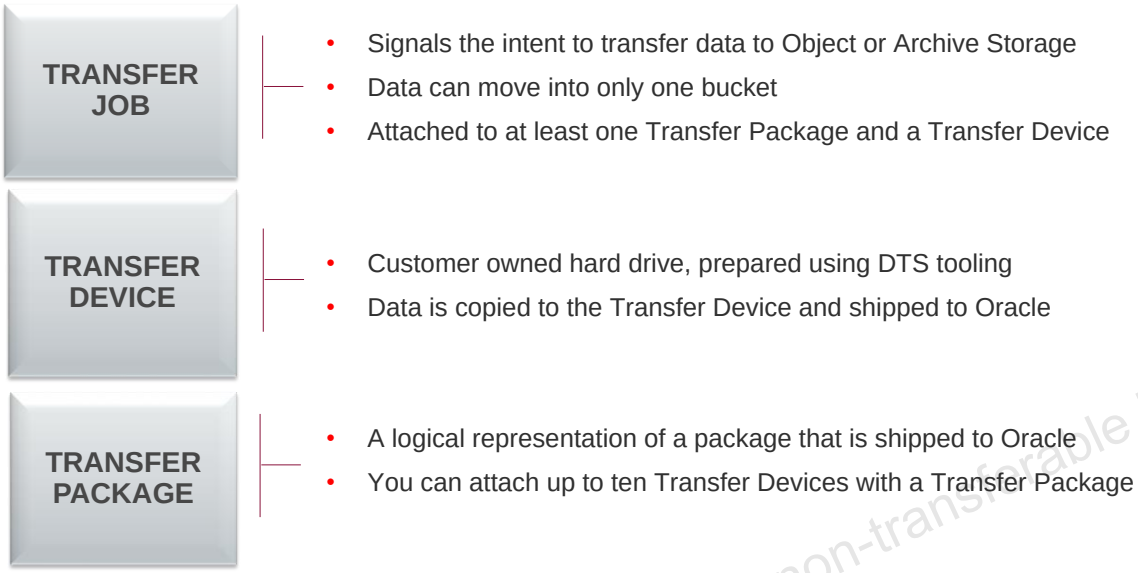
Let’s look at details of data transfer service. It is a solution to migrate data to the cloud. Earlier, customers used hard drives to move data to the cloud. Today we support ATA drive 2.5” and 3.5”. USB 2.0/3.0 will be supported very soon.

It’s very easy to use and completely self served. There is no need to talk to anyone at Oracle or even create an SR. The workflow is pretty straightforward.

Any given time, you have full visibility into where your data is. We will update the status of your package at every stage. When you ship the data, it is encrypted by the utility you used. Once we receive the data, we upload it to your Object Storage account where data is encrypted at rest.

So, your data is always secured.

OCI Data Transfer Service: Terminology



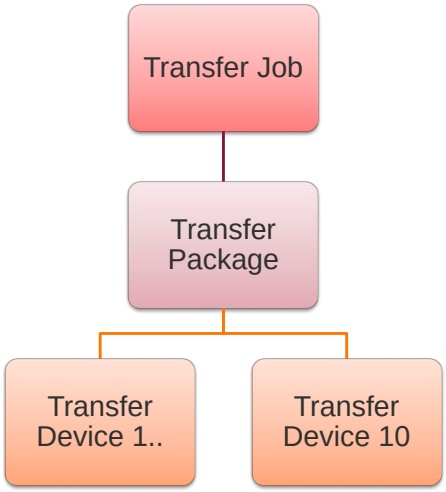
Let's look at the terminology that we will be using going forward.

A transfer job is just an entry in the system to let us know that you would like to transfer data to your Object or Archive Storage. Remember that one transfer job equals one bucket only. What it means is when you create a transfer job, all the data can only be transferred into one single bucket. In order to transfer data to multiple buckets, you need to create multiple transfer jobs.

A transfer device is an HDD that is used to send the data to Oracle. The customer is responsible to procure this HDD. They can use data transfer utility to prepare it for data transfer.

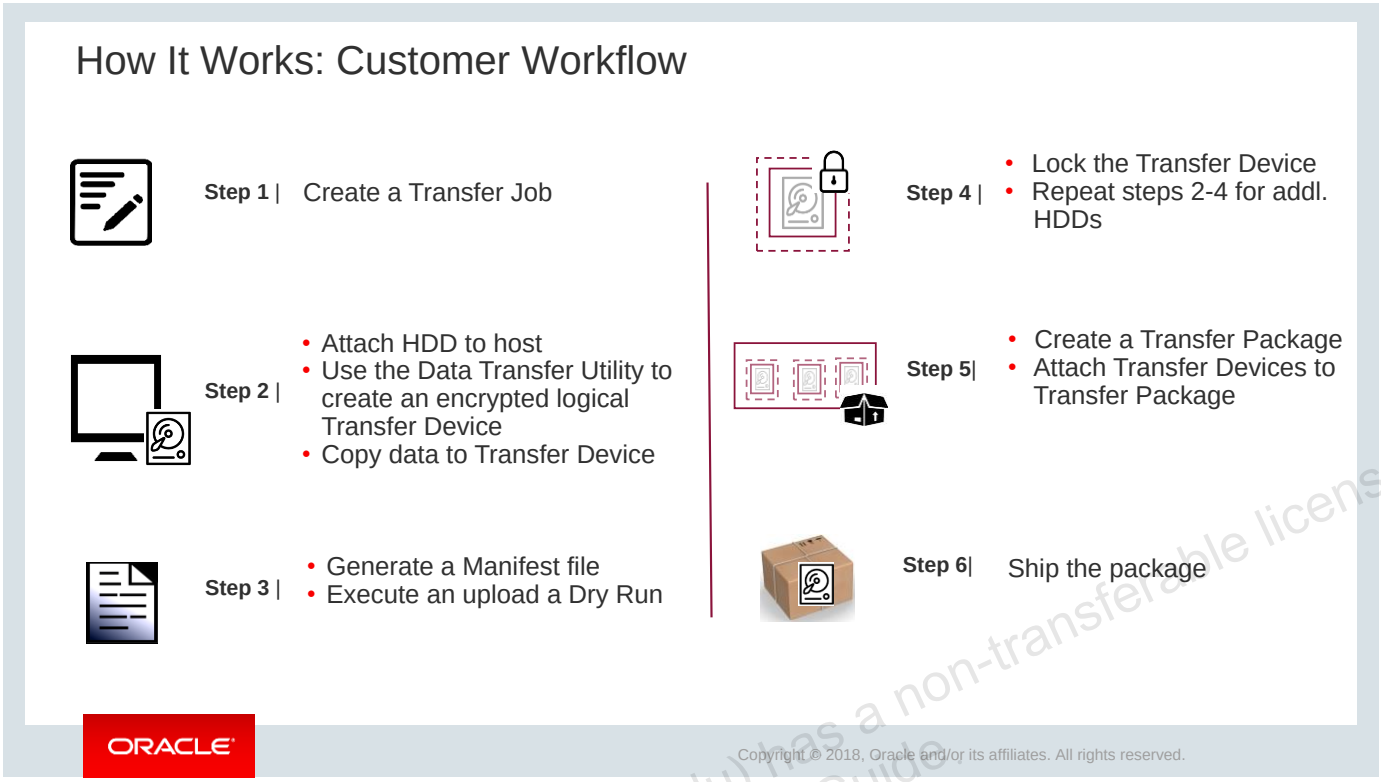
Transfer Package is a logical grouping of Transfer Devices that are used to transfer data to Oracle. You an attach up to ten Transfer Devices per transfer package.

OCI Data Transfer Service: Terminology



- A Transfer Job can have one or more Transfer Packages
- A Transfer Package can attach up to 10 Transfer Devices
- Each Transfer Device holds a portion of the total data set

Each Transfer Job consists of at least one Transfer Package. Each Transfer Package can have up to ten Transfer Devices. Remember, even though you can use ten Transfer Devices, all the data will move to a single bucket and maximum data allowed is 1 PB.



As mentioned earlier, steps are pretty straightforward.

Step 1 – You need to create a Transfer Job using the console or data transfer utility. While selecting a Transfer Job, you are asked to select a bucket in your compartment.

Step 2 – To create Transfer Device, you need to use the data transfer utility. Attach the HDD to a host. Make sure that there are no partitions on the HDD already. If there are partitions, the utility will not allow you to use it in order to avoid any data deletion by mistake.

It will create a volume and mount it for copying the data. You can use OS commands to transfer data to this volume. Data is encrypted.

Step 3 – Once you are done with creating all the Transfer Devices, you need to generate a Manifest File using the data transfer utility. The manifest file contains an index and hash information for all the files. It is used to make sure that data is not corrupted during transportation. It is also used to create an Upload Dry Run report. During an Upload Dry Run, the data transfer utility compares all the files in the container with Manifest File to make sure that there is no naming clash. If there is, the customer will have to delete the appropriate files and run the report again.

Step 4 – Finally you run the lock command using the data transfer utility to lock the Transfer Device. You can repeat steps 2-4 for creating additional Transfer Devices.

sudo dts device lock --job-id <job> --device-label <device> --block-device <block_device>

Step 5 – Finally you create a Transfer Package and select the Transfer Devices you would like to send to Oracle.

Step 6 – Ship the package with your preferred vendor.

For steps 1 and 5, you can use either console or data transfer utility. But for steps 2-4, you can use only the data transfer utility.

How It Works: Backend Workflow

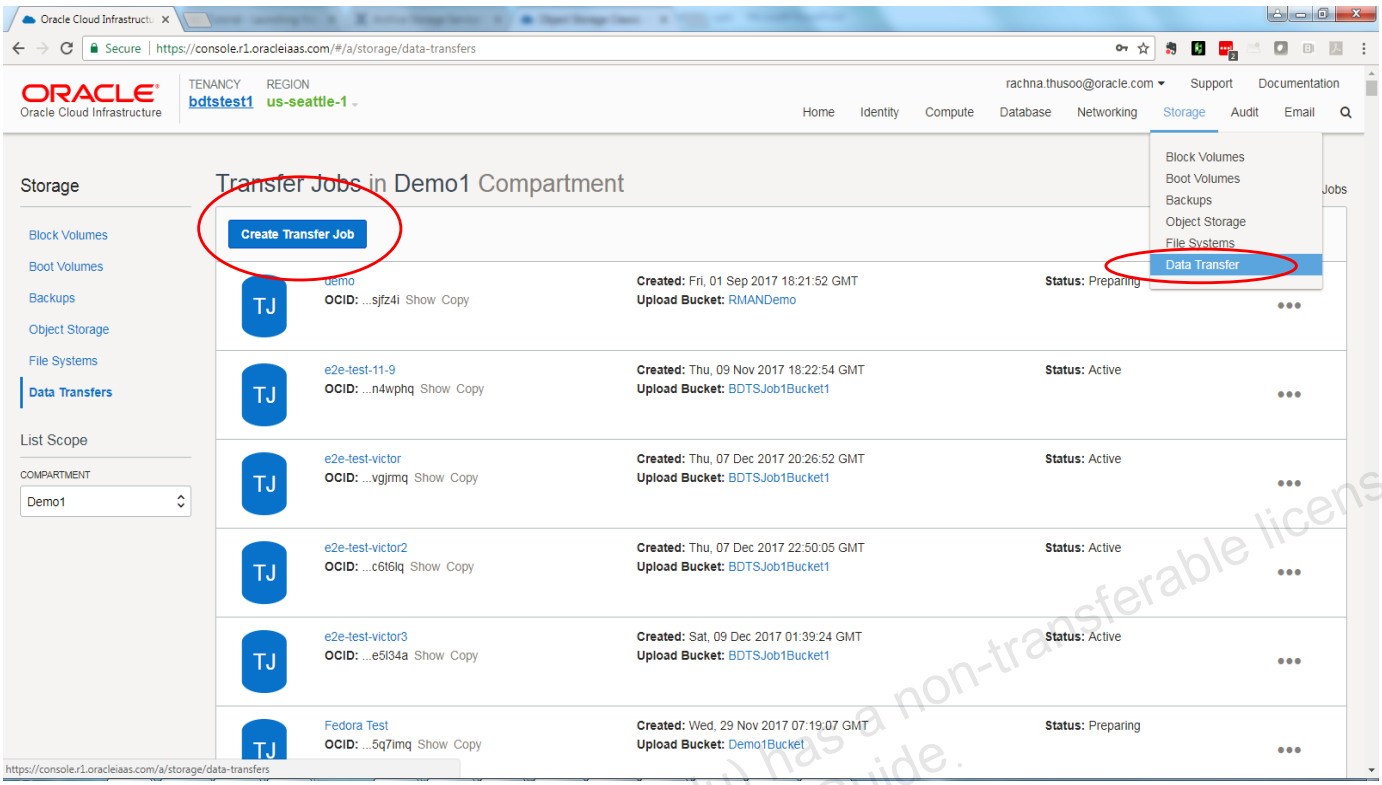
Step 1 | Proactively monitor package status

Step 2 | Upload data from Transfer Device to Object Storage

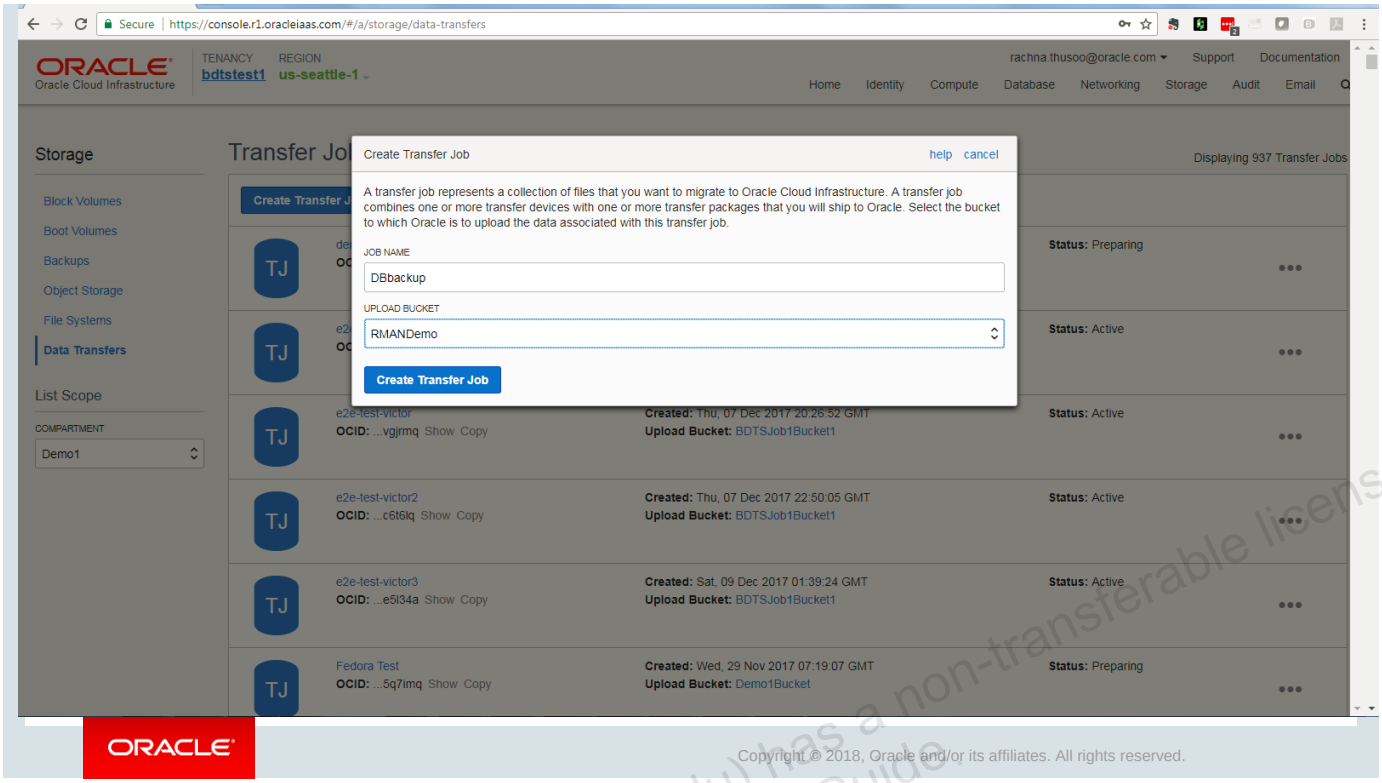
Step 3 | Post upload summary to customer's object storage bucket

Step 4 | Package Transfer Devices in its encrypted state; Ship package back to customer

We proactively monitor the status of the package. Once we receive the package, we upload the data to the customer's Object Storage bucket that was selected while creating the Transfer Job. Once we finish uploading, we post the upload summary to the customer's Object Storage bucket. We then ship the package back to the customer. Remember, the customer is responsible for sending us the return label.



Here is how it looks in the console. As I mentioned earlier, you can either use the console console or the data transfer utility to create a transfer job.



While creating a Transfer Job, you need to specify the Object Storage bucket where you want the data to be transferred to. Make sure you create the bucket before executing this step.

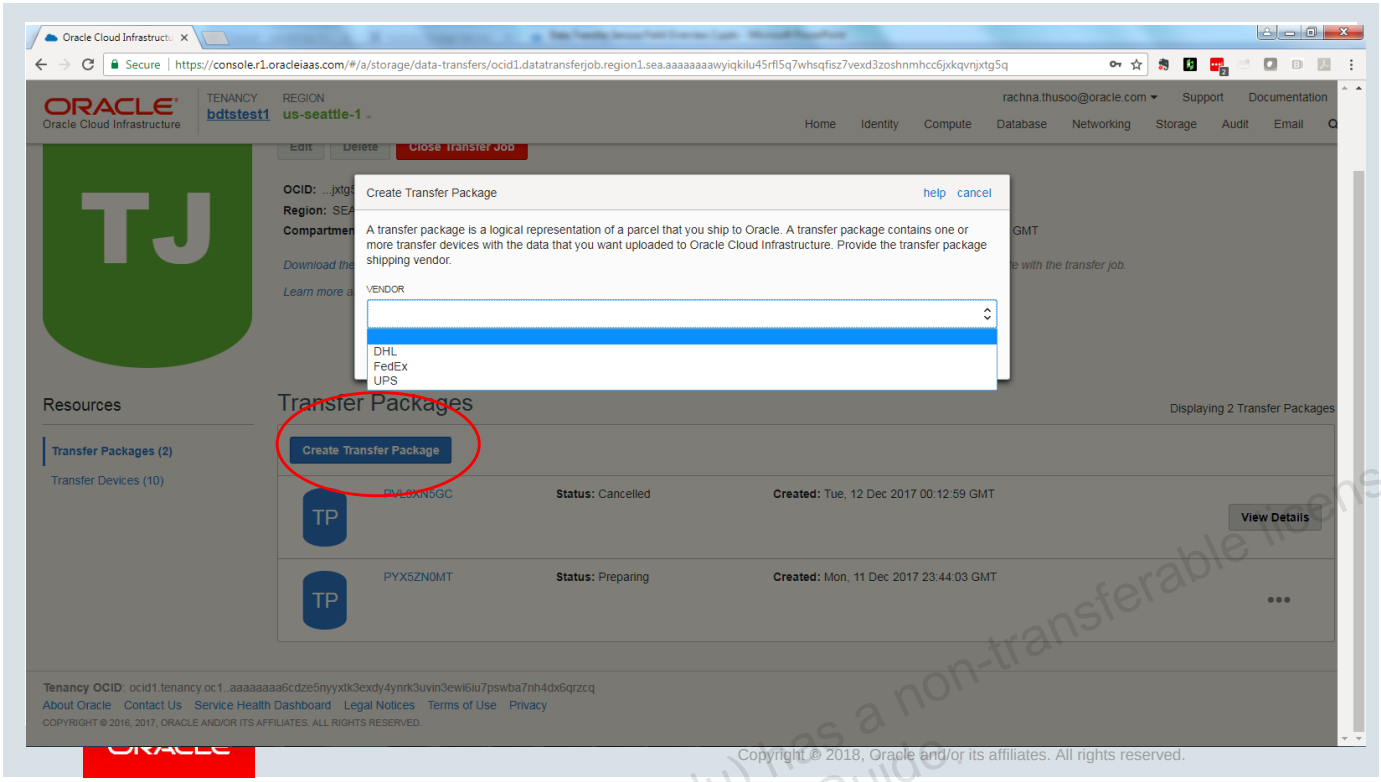
Create Transfer Devices

(You can use only the data transfer utility to create Transfer Devices)

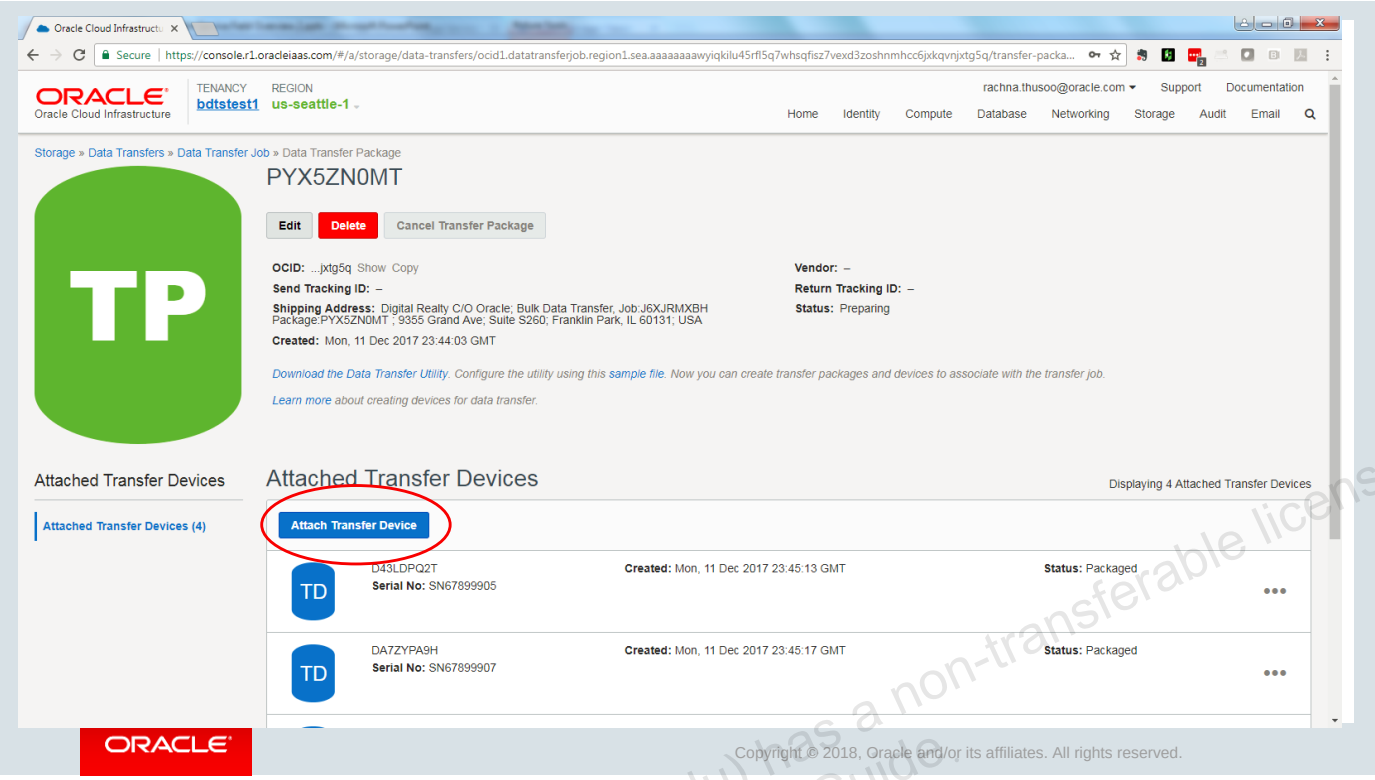
Steps

1. Attach HDD/s to the Host.
2. Create a Transfer Device.
   - Sets up the HDD for encryption using the passphrase
   - Creates a file system on the HDD
   - Mounts the file system at /mnt/orcdts_<device_label>
3. Copy Data to the Transfer Device.
4. Generate a Manifest File.
5. Generate a Dry-Run Report of the transfer.
6. Lock the transfer Device.
7. The Data Transfer utility also encrypts and copies the config_upload_user configuration file to the transfer device.

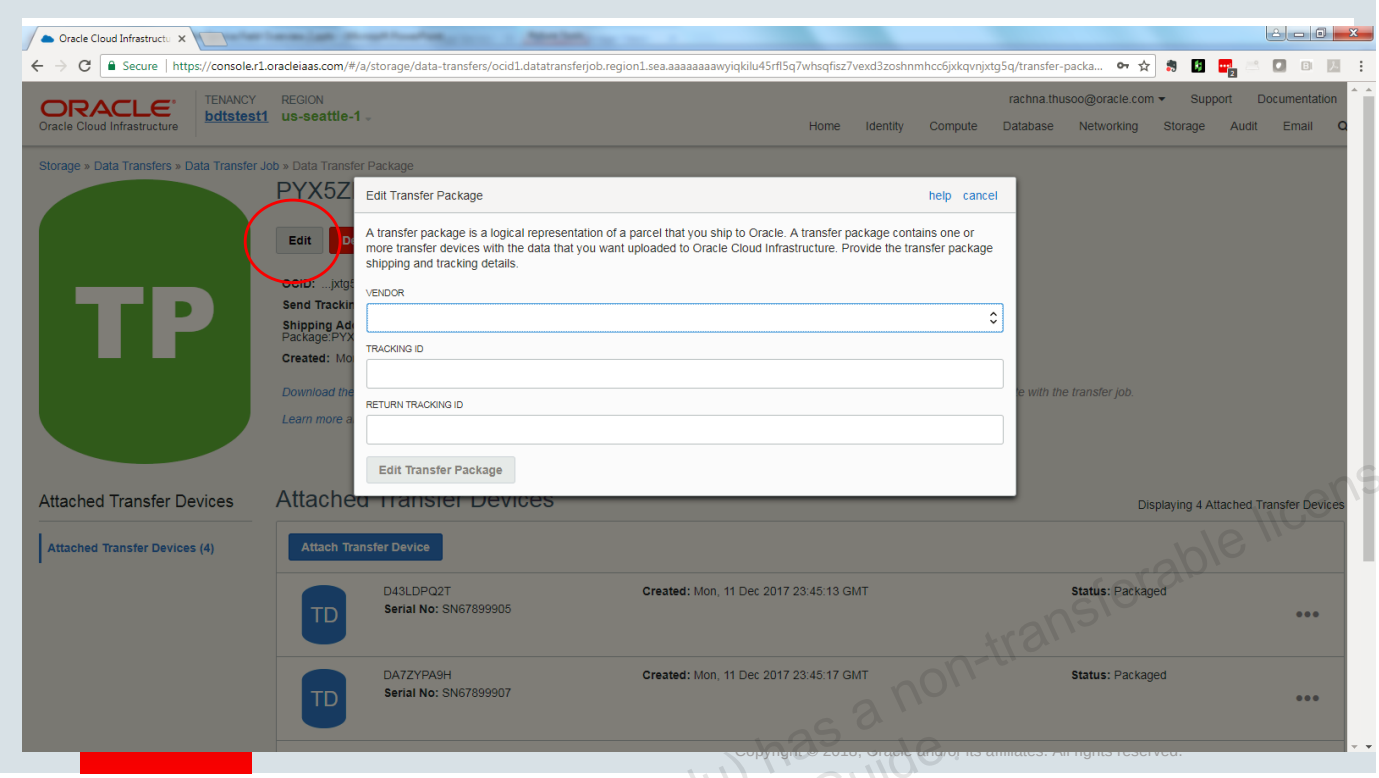
We went through these steps earlier. Remember that Transfer Devices can only be created using data transfer utility.



Once the Transfer Devices are created, you can create Transfer Package.



You can add up to 10 Transfer Devices per Transfer Package.



Once you ship the package, you are responsible for updating the Transfer Package information with Tracking Id and Return Tracking Id.

Pricing and Packaging

- Free for all Oracle cloud customers
- Normal Object Storage charges apply after data is uploaded to OCI
- Data Transfer Setup costs
  - Cost of procuring hard drives – customer is expected to purchase the hard drives. The drives are returned to the customer
  - Shipping costs – customer pays for shipping to and from the Oracle Transfer Site

Oracle does not charge anything for Data Transfer Service but the customer will be charged for Object Storage usage. Also, customer is responsible for procuring the HDDs for data transfer and shipment cost.

Summary

In this lesson, you should have learned how to:

- Describe the need for Data transfer service
- Create Transfer Job, Transfer Device, and Transfer package

6

Load Balancing Service

Objectives

After completing this lesson, you should be able to:

- Describe Oracle Cloud Infrastructure Load Balancing Service concepts
- Create and test a Public Load Balancer

In this lesson, we will talk about load balancing service and its core concepts. We will also take a brief look at how a public LB is managed in OCI.

Load Balancing Service

- Provides automated traffic distribution from one entry point to multiple servers in VCN
- Improves resource utilization, facilitates scaling, and helps ensure high availability
- Regional Load Balancer for your VCN; redundant across two ADs (No single point of failure)
- Supported Protocols – TCP, HTTP/1.0, HTTP/1.1, HTTP/2, WebSocket
- SSL Offloading – SSL Termination, End to End SSL, SSL Tunneling
- Key differentiators
  - Private or Public Load Balancer and Public or Private IP address
  - Provisioned Bandwidth – 100 Mbps, 400 Mbps, 8 Gbps
  - Single LB for TCP and HTTP protocols

Oracle Cloud infrastructure Load Balancing Service provides an automated traffic distribution from one entry point into multiple backend 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.

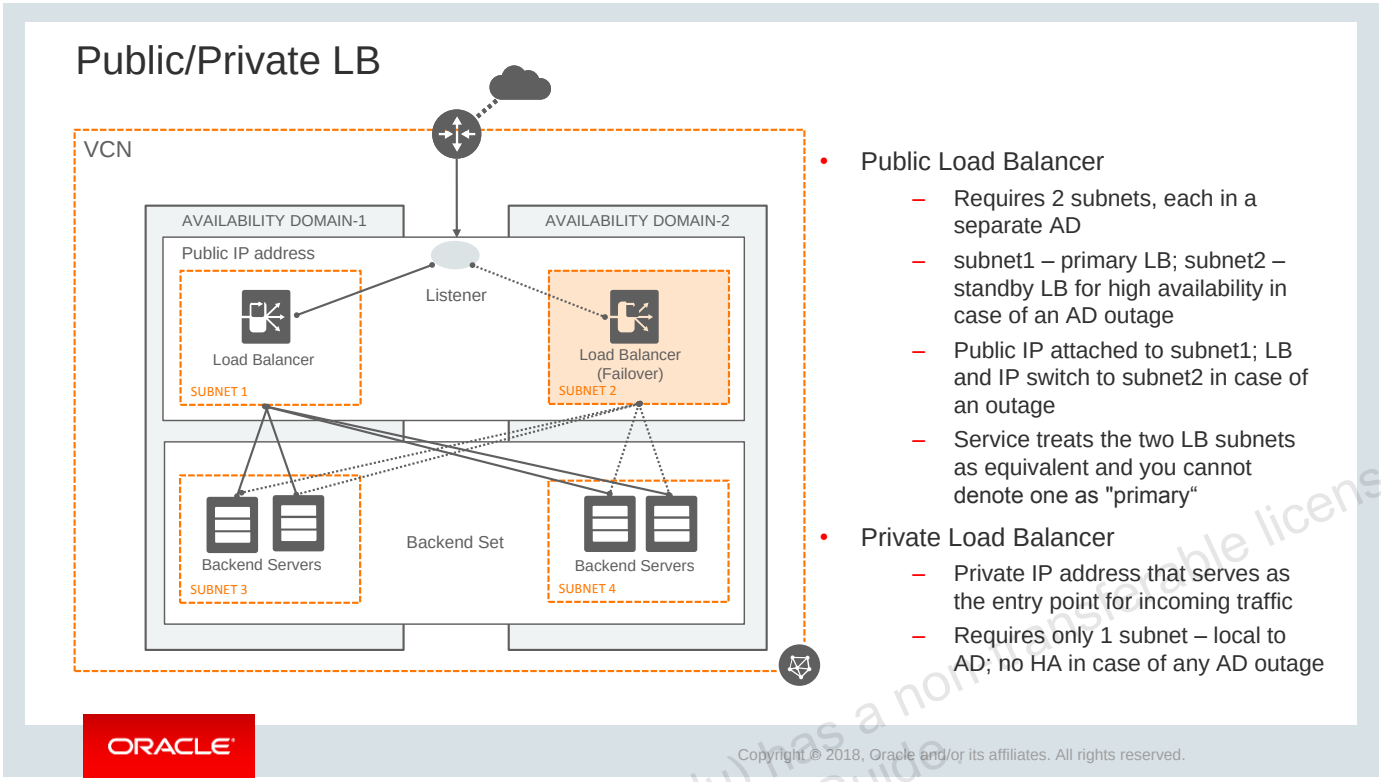
The OCI load balancer supports TCP and the usual HTTP protocols, as well as HTTP/2 and websocket, supporting things like Data Compression, Server Push, Multiplexing of requests. All of these features are supported.

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.
   We can also use 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: 100 Mbps, 400 Mbits, and 8 Gbits. These sizes are for aggregate throughput. The nice thing about having this much capacity provisioned is that it's 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.



Let’s move forward and discuss how the LB service works.

There are two kinds of LBs, a Public LB and a private LB. Let’s first talk about the Public LB.

When you create a Public LB you select two ADs for the LB to reside in; in this case this LB lives in AD1 and AD2. Because OCI is going to create two copies of the LB to make the service highly available, you need to have two subnets (its subnet 1 and subnet 2). After creation, the Public load balancer sits at the edge of a VCN.

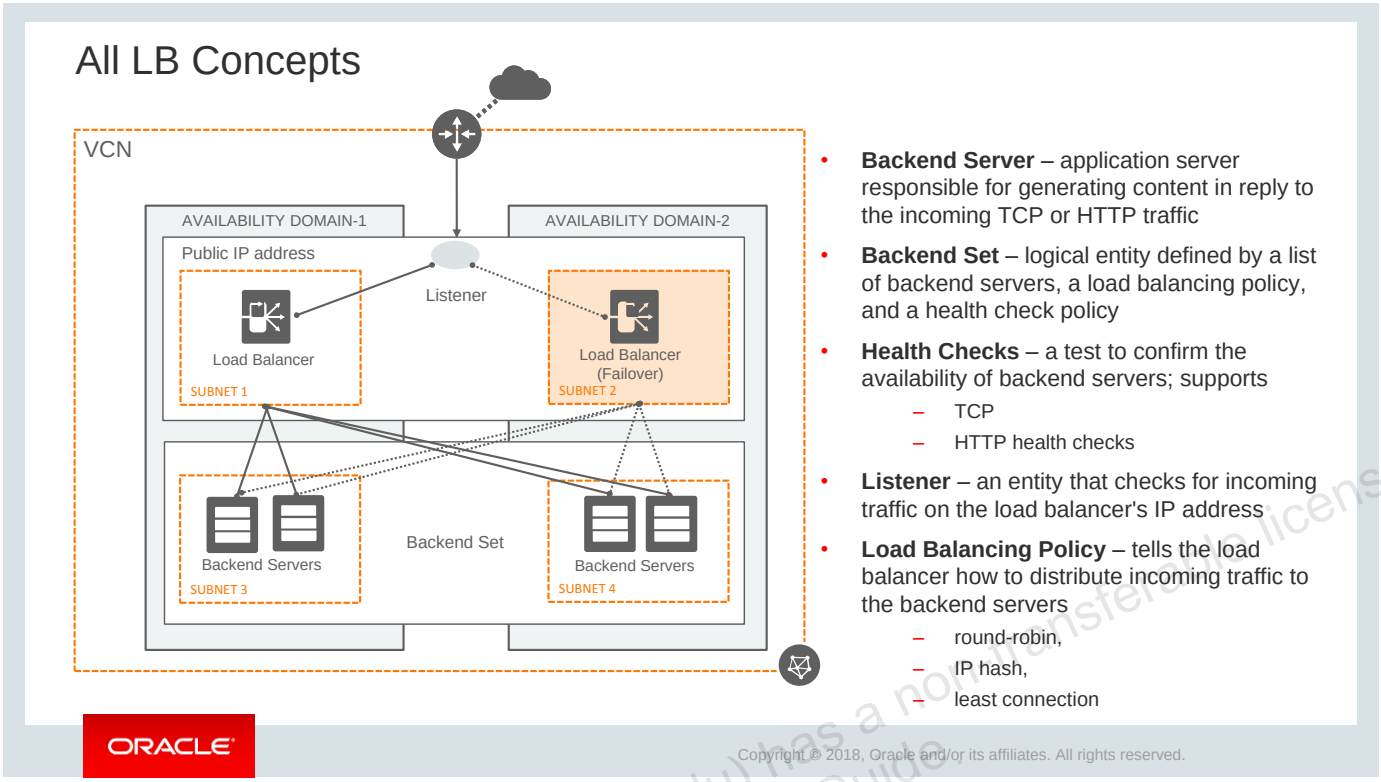
There is a primary load balancer selected automatically to hold the public IP, and a secondary load balancer in an active/standby configuration. This is completely invisible to the user. There is no requirement or capability to designate primary or secondary LB. Next we have a listener. This is the public IP address and the service ports that are opened up to sit between the internet and your backend servers.

In case one of the ADs goes down, the listener will fail over to the other availability domain automatically and where we see a dotted line up at the top will be the new path for the traffic.

This HA is built-in, the user doesn’t have to manage that HA. Remember that there is no way or reason to change which LB is acting as the primary load balancer. It is all managed by the service itself.

The second type of load balancer is a private LB.

For the private load balancer, the implementation is a bit different. Two copies of the load balancer go into a single subnet in a single AD. So it doesn't give you HA in case of the AD outage. However, other than this, all other capabilities are the same.



Let's move forward and talk about the different components within the LB service.

So with any load balancer we need to have a backend server. It's the actual application server which is responsible for generating responses to any incoming requests, HTTP, TCP traffic, etc.

Generally with the load balancer you would want to see two or more application servers or backend servers per availability domains/subnets being load balanced to.

Within a given AD/subnet you can have a backend set. This would be the server in that AD that are available to answer incoming requests. Backend sets are logical entities that simply contain a list of available backend servers and a corresponding health check policy. Note that backend sets can span across ADs.

Within each backend set, you also need to define a load balancing policy, which is how the next server will be chosen. This can be based on IP Hash, least number of connections, and weighted round robin.

There is also a health check policy to check the backend servers and makes sure the service is up. If the service has failed per the parameters in the health check policy, the backend server will be removed until the service returns.

Another basic concept is the listener, which is a logical entity bound to the Load balancer's IP address on which will we create the virtual LB service.

So these are the basic five concepts to keep in mind when creating a load balancer.

One thing to remember here is that though a private load balancer has both of its active and passive entities in a single AD, it doesn't mean the backend servers can't be in all three availability domains.

Load Balancing Service: Shapes

A template that determines the load balancer's total pre-provisioned maximum capacity (bandwidth) for ingress plus egress traffic. Available shapes are:

100 Mbps	400 Mbps	8000 Mbps
Process 100 Mbps total bandwidth when multiple clients connected	Process 400 Mbps total bandwidth when multiple clients connected	Process 8000 Mbps total bandwidth when multiple clients connected
Key characteristics: Up to 1K SSL handshakes per sec with cipher (ECDHE-RSA2K)	Key characteristics: Up to 4K SSL handshakes per sec with cipher (ECDHE-RSA2K)	Key characteristics: Up to 40K SSL handshakes per sec with cipher (ECDHE-RSA2K)

As we discussed before, these are the three shapes available for the Load Balancer service. The sizes are for aggregate capacity, and you can also see the scaling SSL handshakes.

ECDHE is Elliptic Curve Diffie-hellman key Exchange, an encrypted key exchange standard.

Load Balancing Service: Protocol Support

HTTP Load Balancer

- Operates at higher app layer
- HTTP/1.x, WebSocket, HTTP/2 protocol support for incoming HTTP traffic
- SSL Termination, End-to-End SSL
- Traffic Shaping Policy:
  - (Weighted) Round-Robin/Least-Connection/IP-Hash
  - Mark Backend Servers as Drain/ Backup for maintenance window
  - Supports X-forwarded-for header
- Health Check Policy:
  - Application-specific check with response code/body match

TCP Load Balancer

- Operates at intermediate transport layer
- SSL Termination, End-to-End SSL
- Traffic Shaping Policy:
  - (Weighted) Round-Robin/Least-Connection/IP-Hash
  - Mark Backend Servers as Drain/ Backup for maintenance window
  - Use IP-Hash Load Balancing policy for client-IP persistence
- Health Check Policy:
  - Standard TCP Ping-based health check

This shows the protocol support for Load balancing service, for each HTTP and TCP protocol support.

In case of a TCP, LB deals with delivery of messages only with no regard to the content of the messages (layer 4).

We support SSL termination and end-to-end SSL tunnels for both cases. Traffic shapes policies are also similar for both.

Using the API, there is an ability to mark backend servers for Backup, Drain, or Offline.

In case of backup, the load balancer forwards ingress traffic to this backend server only when all other backend servers not marked as "backup" fail the health check policy. This configuration is useful for handling disaster recovery scenarios.

In case of Drain, the load balancer stops forwarding new TCP connections and new non-sticky HTTP requests to this backend server so an administrator can take the server out of rotation for maintenance purposes.

And with marked off as Offline, the load balance forwards no ingress traffic to this backend server.

In case for HTTP LB, we also support X-forwarded-for, which allows the backend server to learn the IP address of the original client.

Health checks: in case of HTTP, health check can check for the server, response codes, and use a regular expression to look for a match in the body/content.

And with TCP, we have ping-based health checks.

Load Balancer: Health Check API

There are four levels of health status indicators.

- Ok (green)
  - No attention required.
  - The resource is functioning as expected
- Warning (yellow) Some reporting entities require attention.
  - The resource is not functioning at peak efficiency or the resource is incomplete and requires further work
- Critical (red)
  - Some or all reporting entities require immediate attention.
  - The resource is not functioning or unexpected failure is imminent.
- unknown (grey)
  - Health status cannot be determined.
  - The resource is not responding or is in transition and might resolve to another status over time.
- Health Check is activated for
  - Load Balancer
  - Backend Servers & Backend Sets

The Load Balancing service provides health status indicators that use your health check policies to report on the general health of your load balancers and their components. You can see health status indicators on the Console List and Details pages for load balancers, backend sets, and backend servers. You also can use the Load Balancing API to retrieve this information.

There are four levels of health status indicators, as shown here. OK, Warning, Critical, and unknown.

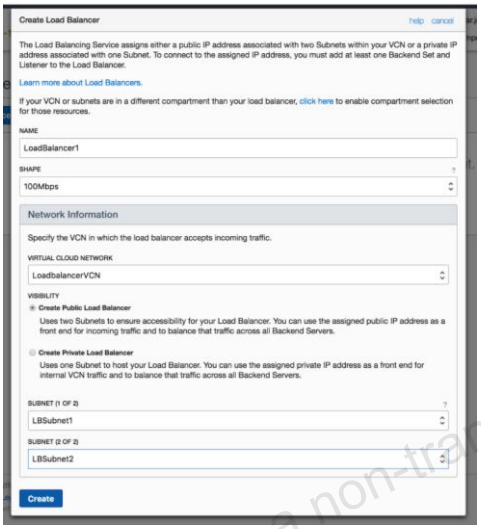
The precise meaning of each one differs for each component (LB, Backend sets, Backend servers).

Current limitation is that health Check is updated every 3 minutes, finer granularity is not unavailable, and it doesn't provide any historical data.

Public Load Balancer Example Configuration (1)

To create and test a public load balancer, complete the following steps:

- Create a public load balancer
  - Name
  - Shape
  - Virtual Cloud Network
  - Visibility
    - Public
    - Private
  - Subnet 1
  - Subnet 2

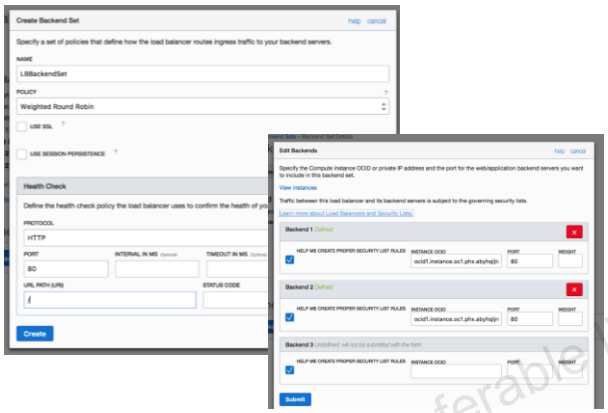


This shows typical steps that are required to create a Public Load Balancer.

Public Load Balancer Example Configuration (2)

To create and test a public load balancer, complete the following steps:

- Create a Backend Set
  - Name
  - Policy
  - SSL
  - Health Check
- Add backend Servers to backend set
  - Instance
  - Port
  - Weight



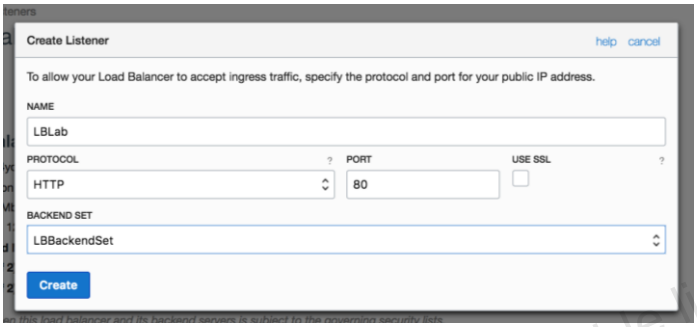
Rules are added to LB Security Lists and Backend Server Security Lists automatically to allow traffic between them. A list is published for the user.

This shows typical steps that are required to create a Public Load Balancer.

Public Load Balancer Example Configuration (3)

To create and test a public load balancer, complete the following steps:

- Create a Listener
  - Name
  - Protocol, Port, SSL
  - Backend Set



This shows typical steps that are required to create a Public Load Balancer.

Public Load Balancer Example Configuration (4)

To create and test a public load balancer, complete the following steps:

- Update the public load balancer subnet security list to allow Internet traffic to the listener
- Verify your public load balancer
- Update rules to protect your backend servers

This shows typical steps that are required to create a Public Load Balancer.

Summary

In this lesson, you should have learned how to:

- Describe Oracle Cloud Infrastructure Load Balancers concepts
- Create and test a Public Load balancer

7

Database Service

Database Service: Objectives

After completing this lesson, you should be able to:

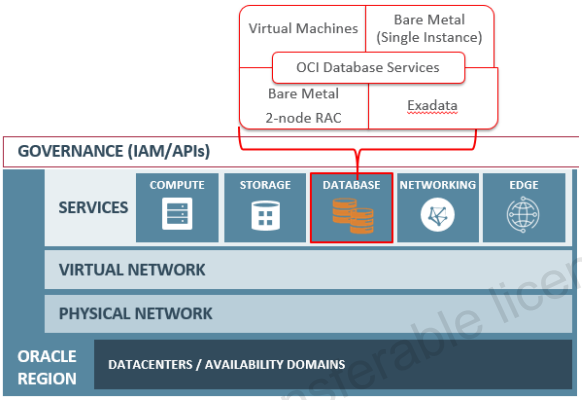
- Describe the options of database systems available with Oracle Cloud Infrastructure
- Features of Database Service
- Launch a one-node database system

For this lesson we will look into fundamentals of the OCI Database System instances on Oracle Cloud Infrastructure.

We will go over the various types of DB systems, some of the characteristics of the various OCI Database Systems, and finally go over how to launch an instance.

Oracle Cloud Infrastructure: Database Service

- Mission-critical enterprise grade cloud database service with comprehensive offerings to cover all enterprise database needs
  - VM, BareMetal, Exadata
- Complete Lifecycle Automation
  - Provisioning, Patching, Backup, Restore, Replicate, Clone, Complete flexibility
- High Availability and Scalability
  - Robust Infrastructure
  - Robust Database Options
  - Dynamic CPU and Storage scaling
- Security
  - Infrastructure (IAM, Security Lists, Audit logs)
  - Database (Transparent Data Encryption)
- Bring Your Own License (BYOL)



At a very high level there are three types of DATABASE systems offered by Oracle cloud infrastructure. First, the bare metal database systems come in two flavors, database systems come in a single node shape along with a two-node RAC shape. The second type of system constitutes the VM-based shapes, which support single and two node cluster operations. The third type of Oracle database system on Oracle cloud infrastructure is Exadata, which comes in quarter, half, and full rack shapes.

Oracle Database Service is backed by a robust infrastructure and it's capable of handling mission-critical production workloads.

This includes three Availability Domains and multiple regions. Currently active redundancy can be implemented with features like dataguard configured to operate across availability domains.

The networking that backs these database systems, along with every other system in OCI, is fully non-blocking, fully contextualized (multi-tenant with full isolation between networks). Speeds go from a minimum of 10 gigabit up to dual 25 gigabit per host along with dedicated Infiniband for cluster and storage networking for RAC and Exadata shapes.

Isolation is accomplished through off-box networking, which allows bare metal hosts along with database systems like Exadata to participate in virtual networks without needing vswitch software installed on host.

For multi-node shapes, the cluster networking is dedicated Infiniband.

Database Systems are protected by 2 or 3-way mirroring.

The Database Systems can be brought up standalone or in a RAC cluster, which is entirely configured and managed by the Database Service. In addition to RAC, Exadata Systems are also available.

Because the systems are fully managed they are MAA (maximum availability architecture) compliant.

Dynamic CPU and Storage Scaling features are available as well as the ability to upsize Exadata deployments across shapes. CPU core usage can be changed hourly to right-size the Database System.

For security there are a number of features and capabilities.

There are as part of the identity service users, groups, compartments and policies which can share or isolate the database system with fine-grained, roles-based controls.

There is also networking security, implicit isolation, off-box network virtualization as well as security lists and on-host firewalls in place.

Along with the policies and network security there is a complete auditing service which tracks all actions of the users whether through the API or Console.

At the Database level there is encryption on by default. Data at rest is transparently encrypted. Backups done to the object store are encrypted and communications with the Database service are encrypted by default.

Licensing flexibility is also available with BYOL - either use the database service with included licenses or bring existing Oracle licenses to the host for use on the cloud.

All of the Database Systems in OCI can be managed by tools such as Enterprise Manager, SQL Developer, etc., just as a regular on-premises database.

==========

==========

Robust Infrastructure

3 Availability Domain – Region architecture

Fully redundant and non blocking Clos Networking Fabric

3 way mirrored storage (optional 2 way mirroring) for Database

Redundant Infiniband Fabric (Exadata, 2 Node RAC) for Cluster networking

Robust Database Options

Database RAC Option

Automated DataGuard

- Within the AD and across AD

MAA Certified Deployment

Automated CPU and Storage Scaling

Infrastructure

Comprehensive IAM Resource Security Model

- Users, Group, Policy and Resource Compartments

Security List – IP Firewall

Audit logs for IaaS/DBaaS API

Database

Default TDE encryption for at rest data

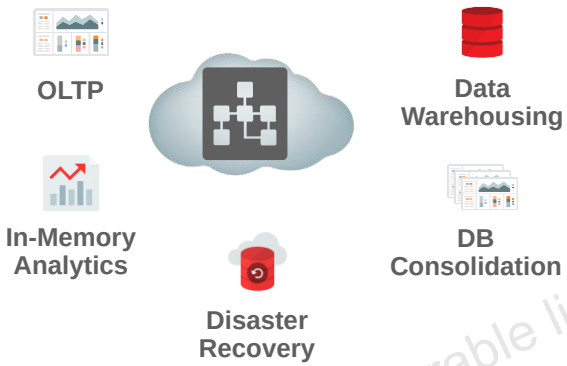
Encrypted backup in Object Store

Secure Client communication through SQLNet

Mention bring your own license.

Database Service: Use Cases

- Mission Critical Production Databases
  - Very large databases (VLDB)
  - Database consolidation
  - OLTP, Data Warehousing, Analytics, Reporting
  - Apps Unlimited (EBS,JDE,PSFT, Siebel)
- Test, Development, Certification, Try before you Buy
- Disaster Recovery
- Migration of Database to Cloud



The database service is suitable for a wide range of workloads and use cases.

- Anything mission critical can be brought to an Oracle Cloud Infrastructure Database System.
- Very large databases, with scaling in Exadata currently going to 8 nodes/336 cores, 12 storage servers, 5.7TB of RAM, 150 TB of flash, 1.1 PB of raw disk and 330TB of usable storage with 3-way mirroring.
- Database consolidation, with containerized DBs - CDBs and PDBs; the database has been written with database consolidation in mind.
- OLTP, Date warehouse, analytics, and reporting
- The Database service is ideal to bring Applications Unlimited to the cloud - E-business Suite, JD Edwards, PeopleSoft and Siebel. These applications all have a growing set of tools to assist customers to Lift and Shift and Move and Improve on-premises applications to OCI.
- Smaller shapes are ideal for Test, development and certification efforts. In addition, it's possible to test out very large Database Systems shapes without having to deal with procurement to see how performance would be on an Exadata.
- Database Systems on OCI can be managed with existing tools such as enterprise manager / cloud control - same as on-premises systems. DB Systems can be configured with Dataguard, Data Pump, GoldenGate, work with RMAN, backup to object storage, etc.
- There's flexibility.

==========

100% Oracle Database compatibility makes migration easy and low risk

Logical Migration: allows reorganization and optimization

Data Pump, GoldenGate Replication

Physical Migration: simplest, byte-to-byte copy

RMAN backup, Transportable technologies, Data Guard

Restore from backup on Oracle Public Cloud

Data Movement Options:

Use public internet

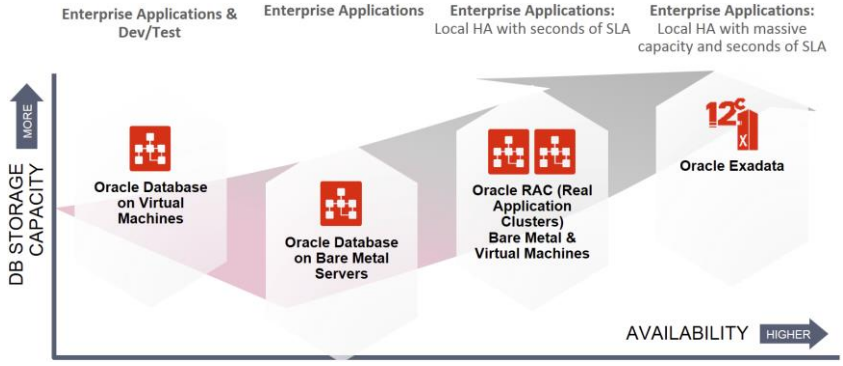
Private high bandwidth virtual network (FastConnect)

Data Transfer Services

==========

MAA Migration Best Practices "Best Practices for Migrating to Exadata Database Machine"

Database Service: Platform Offerings



- Database Service on VMs
- Database Service on Bare Metal Servers
  - Standalone and RAC
- Exadata Database Service
  - Quarter, Half and Full Rack
- Consistent API Across all Platforms
- Compatible with Existing and Familiar tools

The three types of DATABASE system platforms are:

- DB on VM - as both single node or 2 node shapes.
- In addition to VMs, there are bare metal hosts which come as standalone systems or in a RAC configuration.
- The third type of platform is Exadata - scaling from quarter to full rack.

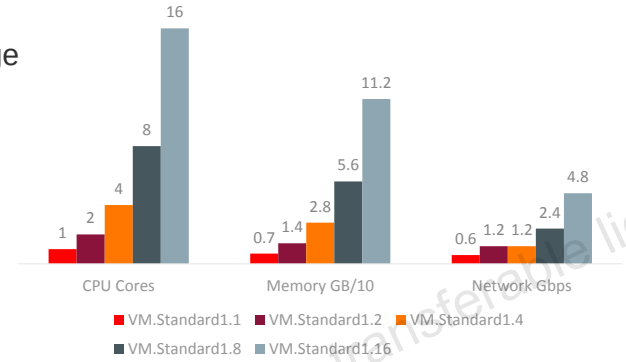
All of these shapes operate under the same API - no special invocations for the various shapes. When orchestrating with tools like terraform, the resource type is the same.

You can manage these systems by using the Console, API, Enterprise Manager, Enterprise Manager Express, SQL Developer, and the dbcli CLI.

Database Service on Virtual Machines

Platform	CPU Core	Memory	Storage	Network	RAC Interconnect	Nodes
VM	1 -16	7-112 GB	256GB -48 TB	0.6- 4.8 Gbps	0.6-4.8 Gbps (Shared)	1-2

- Single Instance or 2 Node RAC
- 3 - way mirrored networked block storage
- Very high performance SR-IOV based network interface
  - Separate interfaces for database client and RAC interconnect
- Scale storage from 256GB to 40 TB



ORACLE

First we can look at the Database Service on VMs.

Database on VMs offers a wide range of flexibility for the Database Service.

Not all workloads need dedicated bare metal servers. Customers asked for a cost effective, easy, and durable database option well suited for a variety of workloads ranging from proof of concept, dev/test environments, to production applications.

VM-based Database Shapes can accommodate these workloads.

The Database Service on VMs is fully featured - while these instances are run-on VMs, the software can be configured with Standard, Enterprise, High and Extreme Editions. Database service on VMs is built on the same high performance, enterprise secure grade, highly durable, and available Cloud infrastructure used by all Oracle Cloud Infrastructure Services.

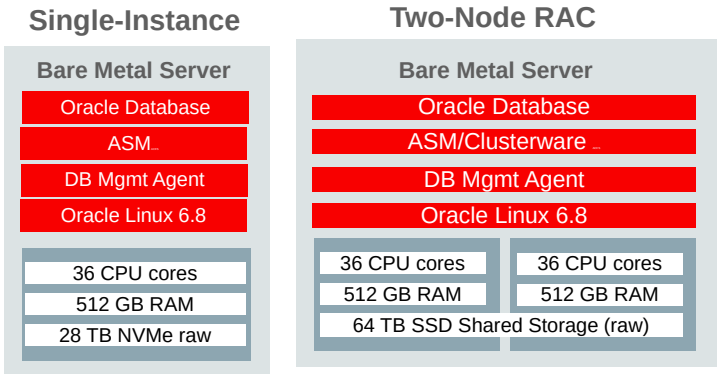
Database Service on Virtual Machines: Benefits

- Cost Effective & Flexible
  - 1 → 16 CPU VMs
  - Pay only for compute and storage used
- Ease of Getting Started
  - Ability to create fully featured Oracle databases
  - Native 2 node RAC configuration available
- Built-in HA constructs
  - Easy deployment of a 2-node RAC configuration with different VM shapes
- Durable & Scalable Storage
  - Remote storage starting 256GB up to 40 TB
  - Dynamic storage scaling
- Secure
  - OCI IAM management control, VCN Security Lists

Benefits of the Database Service on VMs:

- Customers can start as low as a single OCPU VM up to 16 OCPUs. Customers pay only for OCPUs and Storage used.
- Customers can easily create Oracle Certified, full featured, fully supported 11*g*, 12*c* (both 12.1 & 12.2) databases with choice of any database edition. Native 2-node RAC configuration is supported. Use familiar tools to manage your database.
- Customers can easily deploy 2-node RAC configurations with all the VM shapes. For example: Easily deploy a 2-node RAC configuration with 2 core Virtual Machines and shared block storage of up to 40 TB.
- Customers can use remote storage starting 256 GB up to 40 TB. Storage can scale up with no downtime.
- Customers still get all the advantages of our OCI IAM for management control and VCN Security lists for ACL'ing down their database environments like before.

Bare Metal Database System



- Bare Metal Database Systems rely on bare metal servers running Oracle Linux. Two types of BM DB systems:
- One-node database systems:
  - Single bare metal server
  - Locally attached NVMe storage
  - Recommended for Test and Dev
- Two-node RAC database:
  - Two bare metal servers in RAC configuration
  - Direct attached shared storage
  - Supports high performance, recommended for production

There are two types of Bare Metal Database Systems:

- 1-node DB Systems consist of a single bare metal server running Oracle Linux 6.8, with locally attached NVMe storage. This is the least expensive type of system and is recommended for test and development environments. If the node fails, you can simply launch another system and restore databases from current backups.
- 2-Node RAC DB Systems consist of two bare metal servers running Oracle Linux 6.8, in a RAC configuration, with direct-attached shared storage. The cluster provides automatic failover. This system supports only Enterprise Edition - Extreme Performance and is recommended for production applications.

You can manage these systems by using the Console, API, Enterprise Manager, Enterprise Manager Express, SQL Developer, and the dbcli CLI..

Shapes for Bare Metal Database Systems

Platform	CPU Core	Memory	Storage	Network	RAC Interconnect	Nodes
Bare Metal	2-72	512 GB	3.2-9.6 TB	10 Gbps	1 x 40 GbE IB (Dedicated)	1-2

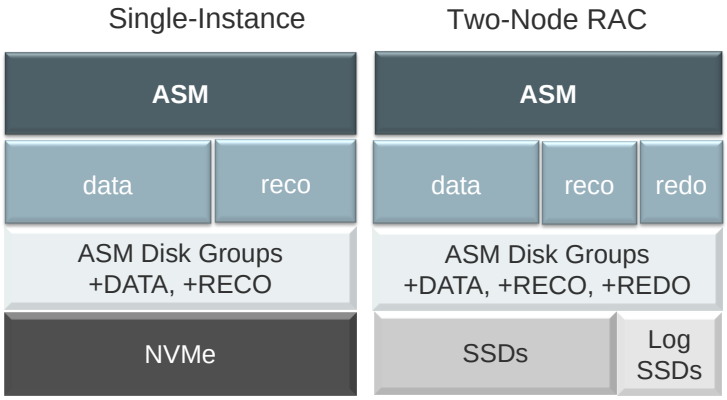
- High IO
  - 1 x x86 Server
  - 36 Cores
  - 512 GB Memory
  - 12.8 TB SSD (4 x 3.2 NVMe)
  - Single Instance
  - Capacity on demand, 2- 36 Cores
- Dense IO
  - 1 x x86 Server
  - 36 Cores
  - 512 GB Memory
  - 28.8 TB SSD (9 x 3.2 NVMe)
  - Single Instance
  - Capacity on Demand, 2- 36 Cores
- 2 Node RAC
  - 2 x x86 Server
  - 72 Cores
  - 512 GB Memory
  - 24.8 TB - 64TB SSD
  - RAC
  - Infiniband Interconnect
  - Capacity on demand, 4- 72 Cores

Shapes for 1- and 2-Node RAC DB Systems

When you launch a DB System, you choose a shape, which determines the resources allocated to the DB System. The available shapes are:

- BM.HighIO1.36: Provides a 1-node DB System (one bare metal server), with up to 36 CPU cores, 512 GB memory, and four 3.2 TB locally attached NVMe drives (12.8 TB total) to the DB System.
- BM.DenseIO1.36: Provides a 1-node DB System (one bare metal server), with up to 36 CPU cores, 512 GB memory, and nine 3.2 TB locally attached NVMe drives (28.8 TB total) to the DB System.
- BM.RACLocalStorage1.72: Provides a 2-node RAC DB System (two bare metal servers), with up to 36 CPU cores on each node (72 total per cluster), 512 GB memory, direct attached shared storage with twenty 1.2 TB SSD drives (24 TB total).

Database System: Storage Architecture



- Tracks the layout, configuration, and status of storage
- Monitors the disks for hard and soft failures
- Proactively off lines disks that failed, are predicted to fail, or performing poorly
- Performs corrective actions if possible
- These actions ensure highest level availability and performance at all times

Storage in OCI Database systems.

ASM directly interfaces with the disks.

Disks are not mounted on ACFS or another file system providing maximum IO. Some resources such as wallets are mounted in a common store along with database homes (binaries) but the DATA and RECOVERY areas are within ASM.

Storage is continuously monitored for any failures with the disks - these disks refer to NVME and SSDs. In the case of VM shapes block volume is used - which is NVME based - and multiple block volumes are brought in and managed the same way as these disks.

Any disks that fail will be managed. Space is reserved for rebalancing so the amount of free space is actually calculated based on that reservation. Whenever the shapes list a maximum amount of usable space in DATA and RECO, these reservations for rebalancing are already taken into account.

The root user has complete control over the Storage subsystem so customization and tuning is possible but the service sets these up by default in an optimal way.

Bare Metal Database Storage Options

The following table outlines the storage used based on the shape and options of Bare Metal Database System:

Shape	Raw Storage	Usable Storage with Normal Redundancy (2-way Mirroring)	Usable Storage with High Redundancy (3-way Mirroring)
BM.HighIO1.36	12.8 TB NVMe	DATA 3.5 TB RECO 740 GB	DATA 2.3 TB RECO 440 GB
BM.DenseIO1.36	28.8 TB NVMe	DATA 9.4 TB RECO 1.7 TB	DATA 5.4 TB RECO 1 TB
BM.RACLocalStorage1.72 (IAD)	24 TB SSD	DATA 8.6 TB RECO 1.6 TB	DATA 5.4 TB RECO 1 TB
BM.RACLocalStorage.72 (PHX, FRA)	64 TB SSD	DATA 23 TB RECOR 4.2 TB	DATA 14.4 TB RECO 2.6 TB

ORACLE

The shape you choose for a DB System determines its total raw storage. Options, like the percentage of the DISK use for RECOVERY (FRA, RECO, REDO) (either 20% or 60%), 2- or 3-way mirroring, and the space allocated for data files, affect the amount of usable storage on the system.

Since users have full control over Database Systems, you can log in and see exactly how all the disks are partitioned, allocated, and used.

Utilities like ASMCMD are available to the grid user to see the state of the DISK groups. You can also run SQL against any database instance and get the ASM information that lets you know how much space is left.

========================

============================

Disk Required Mirror Free MB : Space needed to rebalance after loss of single or double disk failure (for normal or high redundancy)

Disk Usable File MB : Usable space available after reserving space for disk failure and accounting for mirroring

PCT Util : Percent of Total Diskgroup Space Utilized

```
ASM Version is 12.2.0.1
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
|           |          |     |     |             |                 |
|                 |Disk Req'd      |                 |       |    |
|           |DG        |Num  |Num  |Disk Size    |DG Total         |DG Used
|DG Free          |Mirror Free     |Disk Usable      |PCT    |    |
|DG Name    |Type      |FGs  |Disks|MB           |MB               |MB
|MB               |MB              |File MB          |Util   |DFC |
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
|DATA       |NORMAL    |     9|     9|   2,441,888|       21,976,992|
25,932|       21,951,060|       1,977,929|       9,986,565|    .1%|PASS|
|RECO       |NORMAL    |     9|     9|     610,468|        5,494,212|
25,772|        5,468,440|         494,479|       2,486,980|    .5%|PASS|
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
```

SQL to dump the current usage and reservations:

```
SET SERVEROUTPUT ON
SET LINES 155
SET PAGES 0
SET TRIMSPOOL ON

DECLARE
   v_space_reserve_factor NUMBER := 0.15;
   v_num_disks    NUMBER;
   v_group_number   NUMBER;
   v_max_total_mb   NUMBER;
   v_max_used_mb        NUMBER;
   v_fg_count   NUMBER;

   v_required_free_mb   NUMBER;
   v_usable_mb      NUMBER;
   v_cell_usable_mb   NUMBER;
   v_one_cell_usable_mb   NUMBER;
   v_enuf_free      BOOLEAN := FALSE;
   v_enuf_free_cell   BOOLEAN := FALSE;
```

```
   v_req_mirror_free_adj_factor   NUMBER := 1.10;
   v_req_mirror_free_adj         NUMBER := 0;
   v_one_cell_req_mir_free_mb     NUMBER  := 0;

   v_disk_desc      VARCHAR(10) := 'SINGLE';
   v_offset      NUMBER := 50;

   v_db_version   VARCHAR2(8);
   v_inst_name    VARCHAR2(1);

   v_cfc_fail_msg VARCHAR2(500);

BEGIN

   SELECT substr(version,1,8), substr(instance_name,1,1)    INTO v_db_version,
v_inst_name    FROM v$instance;

 DBMS_OUTPUT.PUT_LINE('------ DISK and CELL Failure Diskgroup Space Reserve Requirements
------');
    DBMS_OUTPUT.PUT_LINE(' This procedure determines how much space you need to survive a
DISK or CELL failure. It also shows the usable space ');
    DBMS_OUTPUT.PUT_LINE(' available when reserving space for disk or cell failure (loss
of cell is rare and not usually a concern).  ');
    DBMS_OUTPUT.PUT_LINE(' These required mirror and usable space assume space utilized
to full capacity - a worst case condition.');
    DBMS_OUTPUT.PUT_LINE(' Please see MOS note 1551288.1 for more information.  ');
    DBMS_OUTPUT.PUT_LINE('.  .  .');
    DBMS_OUTPUT.PUT_LINE(' Description of Derived Values:');
    DBMS_OUTPUT.PUT_LINE(' Disk Required Mirror Free MB     : Space needed to rebalance
after loss of single or double disk failure (for normal or high redundancy)');
    DBMS_OUTPUT.PUT_LINE(' Disk Usable File MB              : Usable space available
after reserving space for disk failure and accounting for mirroring');
    DBMS_OUTPUT.PUT_LINE(' PCT Util                         : Percent of Total Diskgroup
Space Utilized');
    DBMS_OUTPUT.PUT_LINE(' DFC                              : Disk Failure Coverage Check
(PASS = able to rebalance after loss of single disk)');

   DBMS_OUTPUT.PUT_LINE('.  .  .');

   DBMS_OUTPUT.PUT_LINE('ASM Version is '||v_db_version);
```

```
-- Set up headings
      DBMS_OUTPUT.PUT_LINE('--------------------------------------------------------------------------
------------------------------------------------------------------------------');
      DBMS_OUTPUT.PUT('|          ');
      DBMS_OUTPUT.PUT('|         ');
      DBMS_OUTPUT.PUT('|     ');
      DBMS_OUTPUT.PUT('|     ');
      DBMS_OUTPUT.PUT('|            ');
      DBMS_OUTPUT.PUT('|                ');
      DBMS_OUTPUT.PUT('|                ');
      DBMS_OUTPUT.PUT('|                ');
      DBMS_OUTPUT.PUT('|Disk Req''d      ');
      DBMS_OUTPUT.PUT('|                ');
      DBMS_OUTPUT.PUT('|       |');
      DBMS_OUTPUT.PUT_LINE('    |');
      -- next row
      DBMS_OUTPUT.PUT('|          ');
      DBMS_OUTPUT.PUT('|DG       ');
      DBMS_OUTPUT.PUT('|Num  ');
      DBMS_OUTPUT.PUT('|Num  ');
      DBMS_OUTPUT.PUT('|Disk Size   ');
      DBMS_OUTPUT.PUT('|DG Total        ');
      DBMS_OUTPUT.PUT('|DG Used         ');
      DBMS_OUTPUT.PUT('|DG Free         ');
      DBMS_OUTPUT.PUT('|Mirror Free     ');
      DBMS_OUTPUT.PUT('|Disk Usable     ');
      DBMS_OUTPUT.PUT('|PCT    |');
      DBMS_OUTPUT.PUT_LINE('    |');
      -- next row
      DBMS_OUTPUT.PUT('|DG Name   ');
      DBMS_OUTPUT.PUT('|Type     ');
      DBMS_OUTPUT.PUT('|FGs  ');
      DBMS_OUTPUT.PUT('|Disks');
      DBMS_OUTPUT.PUT('|MB          ');
      DBMS_OUTPUT.PUT('|MB              ');
      DBMS_OUTPUT.PUT('|MB              ');
      DBMS_OUTPUT.PUT('|MB              ');
      DBMS_OUTPUT.PUT('|MB              ');
      DBMS_OUTPUT.PUT('|File MB         ');
      DBMS_OUTPUT.PUT('|Util   ');
      DBMS_OUTPUT.PUT_LINE('|DFC |');
      DBMS_OUTPUT.PUT_LINE('--------------------------------------------------------------------------
------------------------------------------------------------------------------');
```

```
 FOR dg IN (SELECT name, type, group_number, total_mb, free_mb, required_mirror_free_mb
FROM v$asm_diskgroup ORDER BY name) LOOP


      v_enuf_free := FALSE;


      -- Find largest amount of space allocated to a cell
      SELECT sum(disk_cnt), max(max_total_mb), max(sum_used_mb), count(distinct
failgroup)
     INTO v_num_disks,v_max_total_mb, v_max_used_mb, v_fg_count
      FROM (SELECT failgroup, count(1) disk_cnt, max(total_mb) max_total_mb, sum(total_mb
- free_mb) sum_used_mb
      FROM v$asm_disk
     WHERE group_number = dg.group_number and failgroup_type = 'REGULAR'
     GROUP BY failgroup);


   -- Amount to reserve depends on version and number of FGs
   IF  ((v_db_version like '12.2%') or (v_db_version like '18%')) THEN
     IF v_fg_count < 5 THEN
             v_space_reserve_factor := 0.15 ;
     ELSE
       v_space_reserve_factor := 0.09 ;
     END IF;
   ELSIF ( (v_db_version like '12.1%' ) or (v_db_version like '11.2.0.4%') ) THEN
       v_space_reserve_factor := 0.15 ;
   ELSE
       v_space_reserve_factor := 0.15 ;
   END IF;


   v_required_free_mb := v_space_reserve_factor * dg.total_mb;
   IF dg.free_mb > v_required_free_mb THEN v_enuf_free := TRUE; END IF;


        IF dg.type = 'NORMAL' THEN


                -- DISK usable file MB
                v_usable_mb := ROUND((dg.free_mb - v_required_free_mb)/2);
```

```
 ELSIF dg.type = 'HIGH' THEN
                 -- HIGH redundancy
                 -- DISK usable file MB
                 v_usable_mb := ROUND((dg.free_mb - v_required_free_mb)/3);

        ELSIF dg.type = 'EXTEND' THEN
                 -- EXTENDED redundancy for stretch clusters

                 -- DISK usable file MB
                 v_usable_mb := ROUND((dg.free_mb - v_required_free_mb)/4);

         ELSE
                -- We don't know this type...maybe FLEX DG - not enough info to say
                v_usable_mb := NULL;

        END IF;

       DBMS_OUTPUT.PUT('|'||RPAD(dg.name,v_offset-40));
       DBMS_OUTPUT.PUT('|'||RPAD(nvl(dg.type,'  '),v_offset-41));
       DBMS_OUTPUT.PUT('|'||LPAD(TO_CHAR(v_fg_count),v_offset-45));
       DBMS_OUTPUT.PUT('|'||LPAD(TO_CHAR(v_num_disks),v_offset-45));
       DBMS_OUTPUT.PUT('|'||TO_CHAR(v_max_total_mb,'999,999,999'));
       DBMS_OUTPUT.PUT('|'||TO_CHAR(dg.total_mb,'999,999,999,999'));
       DBMS_OUTPUT.PUT('|'||TO_CHAR(dg.total_mb - dg.free_mb,'999,999,999,999'));
       DBMS_OUTPUT.PUT('|'||TO_CHAR(dg.free_mb,'999,999,999,999'));
       DBMS_OUTPUT.PUT('|'||TO_CHAR(ROUND(v_required_free_mb),'999,999,999,999'));
       DBMS_OUTPUT.PUT('|'||TO_CHAR(ROUND(v_usable_mb),'999,999,999,999'));

      -- Calc Disk Utilization Percentage
       IF dg.total_mb > 0 THEN
          DBMS_OUTPUT.PUT('|'||TO_CHAR((((dg.total_mb -
dg.free_mb)/dg.total_mb)*100),'999.9')||CHR(37));
       ELSE
          DBMS_OUTPUT.PUT('|       ');
       END IF;
```

```
      IF v_enuf_free THEN
         DBMS_OUTPUT.PUT_LINE('|'||'PASS|');
      ELSE
         DBMS_OUTPUT.PUT_LINE('|'||'FAIL|');
      END IF;



   END LOOP;

     DBMS_OUTPUT.PUT_LINE('-------------------------------------------------------
-------------------------------------------------------------------------------
----------');
   <<the_end>>

   IF v_cfc_fail_msg is not null THEN
      DBMS_OUTPUT.PUT_LINE('Cell Failure Coverage Freespace Failures Detected. Warning
Message Follows.');
      DBMS_OUTPUT.PUT_LINE(v_cfc_fail_msg);
   END IF;

   DBMS_OUTPUT.PUT_LINE('.  .  .');
   DBMS_OUTPUT.PUT_LINE('Script completed.');

END;
/

WHENEVER SQLERROR EXIT FAILURE;
```

Oracle Database Exadata Cloud Service

Full Oracle Database with all advanced options on the fastest and most available database cloud platform

- Scale-Out Compute, Scale-Out Storage, Infiniband, PCIe flash
- Complete Isolation of tenants with no overprovisioning

All Benefits of Public Cloud

- Fast, Elastic, Web Driven Provisioning
- Oracle Experts Deploy and Manage Infrastructure

In addition to VMs, Bare Metal hosts, and Bare Metal RAC and VM, Exadata is available on OCI.

The Exadata systems are provided in three shapes and all of them have all the advanced options that Exadata provides turned on.

These are physical Exadata engineered systems - complete with Infiniband networking and scalable compute and storage nodes - that can be run on OCI without modification.

Complete isolation of tenants is facilitated - whenever partial shapes of Exadata are used, tenants are completely isolated.

Exadata on OCI gives all of the features, performance, and capabilities of on-premises Exadata but with the flexibility of cloud.

All the installation, from systems, to firmware, to OS install and maintenance to patching are managed by Oracle and presented as a public cloud service.

Exadata DB Systems

- The Exadata DB Systems let you leverage the power of Exadata within Oracle Cloud Infrastructure.
- Exadata Database Nodes are fully managed at the hardware level, and are completely available for the user
- Oracle manages Exadata infrastructure
  - Servers, storage, storage software, networking, firmware, hypervisor, etc.

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

Exadata DB Systems are offered in quarter-rack, half-rack, or full-rack configurations, and each configuration consists of compute nodes and storage servers.

You can see on this table the usable storage capacity and RAM for each of the configurations.

It is very nice to be able to try out Exadata for your database needs on the cloud without having to procure a physical Exadata. With Oracle Cloud infrastructure customers are able to try Exadata out and they like what they see.

Each compute node is configured so that users have root access to a virtual context running on the compute hosts.

You have root privilege to these compute nodes so you can load and run additional software on them.

However, users do not have administrative access to the Exadata infrastructure components, such as the physical compute node hardware, network switches, power distribution units (PDUs), integrated lights-out management (ILOM) interfaces, or the Exadata Storage Servers, which are all administered by Oracle.

You have full administrative privileges for your databases, and you can connect to your databases via public or private IPs or both.

Users are responsible for database administration tasks such as creating tablespaces and managing database users.

You can also customize the default automated maintenance setup, and you control the recovery process in the event of a database failure.

Exadata DB systems on Oracle cloud infrastructure benefit from having the IAM service which helps create policies on which users and groups can perform which actions on the Exadata and DB systems.

You can have compartments and VCNs for these database services and either isolate or share them.

All of the virtual cloud network capabilities and advantages are afforded to the DB and Exadata DBAAS system. You do not have to use a public IP for any of the instances if you do not want to.

You can use VPN and fastconnect to connect to your on-prem environments.

Because of the capabilities of the Oracle cloud infrastructure we can have the application tier seamlessly running on VMs while the database is running on bare metal.

Database Editions and Versions

	VM Database	BM Database	RAC Shapes (VM, BM)	Exadata	DB Versions
Standard Edition	Yes	Yes	No	No	11.2.0.4 12.1.0.2 12.2.0.1
Enterprise Edition	Yes	Yes	No	No	
High Performance	Yes	Yes	No	No	
Extreme Performance	Yes	Yes	Yes	Yes	
BYOL	Yes				

With OCI Database Service there are three main versions of the Database available, 11.2, 12.1, and 12.2.

Depending on the shape and clustering configuration, certain shapes are restricted to Extreme Performance.

All shapes have the ability for the user to BYOL.

All of the shapes can use multiple instances of the database and mix and match.

Complete Life Cycle Automation for Database Service

- Provisioning: Current bundle patch and critical one-off patches included
- Patching: Automated quarterly patching update
- Backup: Automated Backup to Object Store
- Restore: Full and point in time from backup
- Clone: Create database from a backup
- Complete flexibility
  - Full root access to host
  - All home grown automation/monitoring tools work

Provisioning and patching

Provisioning:

Patched with current Patchset, DB Bundle/PSU and all critical one-offs

PATCHING:

No need to research the required patches -- From firmware to database

No need to test end-to-end inter-operability -- Oracle thoroughly tests the entire stack

No need to have the knowledge to patch the components -- Automated patching process does it for you

1. Patches automatically queue up for database systems and instances.
2. software repository automatically updates
3. OS updates are applied
4. Grid infrastructure updates applied.
5. firmware applied
6. database instances are updated.

Backup:

The Database Service will provide managed backup and restore functionality for databases across all the different shapes, database versions, & editions offered on the platform .

Default Policy for Databases: Daily Incremental & Weekly full backups

On-Demand Full Backups: Ability to trigger a backup at any time.

RESTORE:

Restore From Backup: Restore from latest backup, point in time recovery (timestamp or SCN tag), restore from a specific backup

CLONE:

Create DB from Backup: Ability to create a new DB instance from a backup to another Database System.

Complete Flexibility: SQL*Plus, SQL Developer, toad, enterprise manager/cloud control, home grown tools - all work with OCI Database Systems.

Database Service: High Availability and Scalability

- Robust Infrastructure
  - 3 Availability Domain – Region architecture
  - Fully redundant and non-blocking Networking Fabric
  - 2-way or 3-way mirrored storage for Database
  - Redundant Infiniband Fabric (Exadata, 2 Node RAC) for Cluster networking
- Robust Database Options
  - Database RAC Option
  - Automated Data Guard within the AD and across ADs
  - MAA Configuration / Best Practices are built in
- Dynamic CPU and Storage Scaling

- The database service is built for availability and scalability.
- Backups are hosted on regional services like object storage which span across availability domains.
- Dataguard can be implemented across availability domains so that linked systems and backups can survive AD-level disruptions.
- Networking fabric is fully non-blocking with bare metal servers having 10 and 25 gigabit networking to the host. The generation2/x7 shapes have dual 25gbe networking to the bare metal hosts.
- Storage is set up to be highly available and has two options, NORMAL/2-way mirroring and HIGH/3-way mirroring, which guarantees that there are two or three copies of every extent.
- For RAC and Exadata shapes there is a dedicated Infiniband fabric for cluster networking.
- With RAC shapes the ability to create highly available database instances - from VMs, to Bare metal, to Exadata. With dataguard the ability to stretch the availability of the database pair across availability domains.
- The Database Systems are fully managed by Oracle and follow MAA/Maximum Availability Architecture - the supported and best practices are built in.
- It is also possible to scale database shapes from minimum to maximum CPU usage on the fly on a hourly basis. Within Exadata shapes there are options for users to grow out of smaller shapes to larger ones.
- Dynamic storage Scaling for VMs only
- Bare metal dynamic CPU storage

OCI Security Features Overview for Database Service

Security capability	OCI DB System security feature
Instance security isolation	OCI Bare-Metal (BM) instance
Network security and access control	VCN, VCN Security Lists, VCN public and private sub-nets, VCN route table
Secure and highly-available connectivity	VPN, Fastconnect, DRGs
User authentication & authorization	IAM tenancy, compartments and security policies, console password, API signing key, SSH keys
Data encryption	DBaaS TDE, RMAN encrypted back-ups, Storage and object encryption at rest
Auditing	OCI API audit logs

Overview of Security features for the Database Service

Instances are completely isolated on bare metal hosts. In the case of Exadata, the Exadata system is partitioned completely so there are is no exposure between tenants or performance implications.

In the case of shared infrastructure with Database on VMs, the DB Systems are completely isolated from one another via the host, and networking is being done with pass through, so networking is completely separated.

Networking is completely virtualized. Database systems can be completely isolated or put onto networks that are shared in the tenancy or through a gateway. In addition to the implicit networking isolation which can be restricted to completely private networks there is an ability to use security lists to further regulate what can communicate with the database instances.

For off-cloud networking, DRGs can connect the VCN/cloud networks to other locations via dedicated "FastConnect" circuits or via IPSEC VPN.

Identity and Access service can implement fine-grained, role-based controls that limit who in the tenancy can do what to the database service. Compartments offer a logical separation. It is possible to completely isolate database services within the tenancy so that dev, test and production can be in the same tenancy but never interfere with each other.

Data encryption is on by default, transparent data encryption is on by default, all backups - automated or manual - are encrypted and local, block and object storage are themselves encrypted.

OCI's auditing infrastructure provides a log of every action taken by every user via the console or the API. It is possible to get fine-grained details of which user performed what actions to any service at a very fine-grained level.

==========

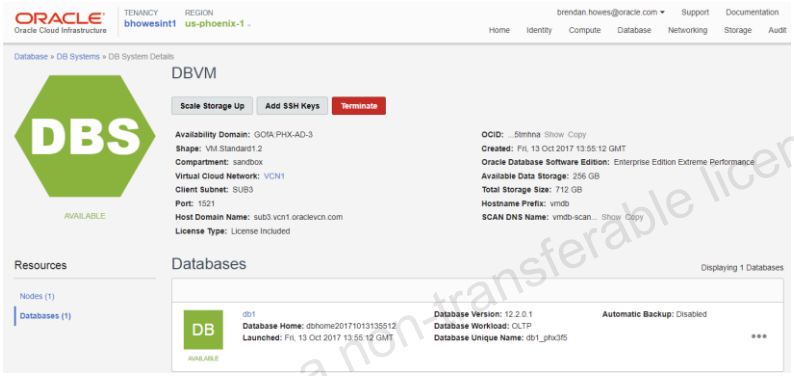
==========

Fastconnect = dedicated circuit, MPLS

Managing the Database Systems

You can use the console to perform the following tasks:

- Launch a DB System: You can create a database system.
- Check the status: You can view the status of your database creation and after that, you can view the runtime status of the database
- Start, stop, or reboot
- Scale: You can scale up the number of enabled CPU cores in the system.
- Terminate: Terminating a DB System permanently deletes it and any databases running on it



Now that an overview of the Database Service on OCI has been covered we can move on to the hands on lab.

Launching a database system is done through the console.

Open the Console, click Database tab, under DB Systems, make sure your compartment is set correctly and then start the Launch DB System process.

In the Launch DB System dialog enter or select the appropriate values:

DB System Information

DISPLAY NAME

AVAILABILITY DOMAIN

SHAPE

VM, BM, RAC, Exadata

ORACLE DATABASE SOFTWARE EDITION

CPU CORE COUNT

LICENSE TYPE (Included, BYOL)

SSH PUBLIC KEY

DATA STORAGE PERCENTAGE (DATA:RECO Split) (40/80)

DISK REDUNDANCY (Normal (2-way mirror)|HIGH (3-way mirror)

VIRTUAL CLOUD NETWORK

CLIENT SUBNET

HOSTNAME PREFIX

DATABASE NAME (CDB)

DATABASE VERSION

PDB NAME

DATABASE ADMIN PASSWORD

AUTOMATIC BACKUP ON|OFF

DATABASE WORKLOAD

ON-LINE TRANSACTION PROCESSING (OLTP) / DECISION SUPPORT SYSTEM (DSS)

CHARACTER SET

NATIONAL CHARACTER SET

While the task of launching a database is quite simple, you should plan your database implementations with your database architects.

Database Service: Summary

In this lesson, you should have learned how to:

- Describe the options of database systems available with Oracle Cloud Infrastructure
- Features of OCI Database Service
- Have the background needed to launch a one-node database system

8

OCI DNS Service

Objectives

After completing this lesson, you should be able to:

- Describe the OCI DNS Services available with Oracle Cloud Infrastructure
- Configure DNS for a Tenancy

After completing this lesson, you should be able to describe the basic OCI DNS services available on OCI.

You should also be able to configure DNS within a tenancy.

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

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.

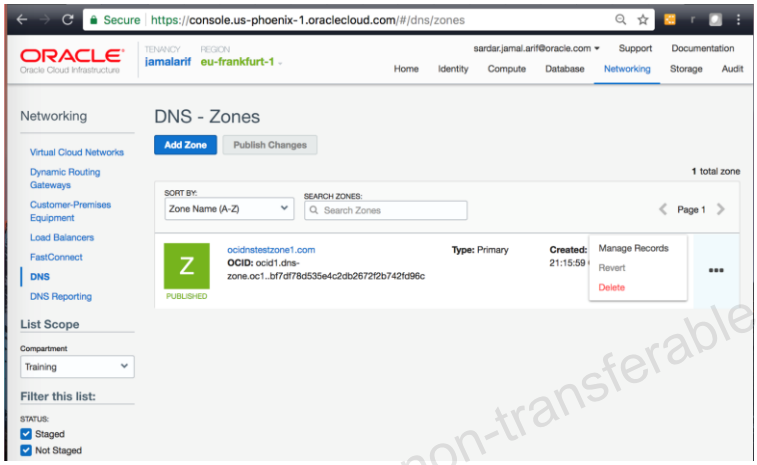
By configuring OCI DNS, enterprise and business customers can connect their DNS queries to various kinds of assets, such as OCI Compute, as well as to third-party and private assets. Operators can manage their own DNS records for both cloud and non-cloud resources. OCI DNS is used when zones need to be exposed to the internet for resolution.

Domains and zones can be both on 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 follows 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.
- Zone Transfer
- 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 also 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 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 in total.

DNS Record Types Supported by OCI DNS

Currently, OCI DNS supports the following DNS records:

- A (Address Record)
- AAAA (IPv6 Address Record)
- CNAME (Canonical Name Record)
- NS (Name Server record)
- MX (Mail Exchange Record)
- PTR (Pointer Record)
- SOA (Start of Authority Record)
- SRV (Service Locator)
- TXT (Text Record)

OCI DNS also supports an **ALIAS** record type which is specific to OCI DNS. It can be used to map to various OCI (Compute, Network, Database or Storage), third-party or private assets. The **ALIAS** record acts similar to a **CNAME** record but the difference is they can be at the 'apex' record of a Zone and not visible to external resolvers.

OCI DNS supports all the standard/major record types as is seen here.

OCI DNS via DYN can also register domain names.

Record types the basic A, QUAD-A, Cname records. Ttl defaults to one hour.

(NS) records are used to list all of the authoritative nameservers for your domain or subdomain. By default, any new primary zone in OCI DNS will automatically create NS records at the apex of the zone for your assigned nameservers. OCI DNS is capable of adding other NS records at the apex of the zone if you are using multiple DNS providers, such as a scenario if OCI DNS is primary and another provider is secondary.

MX - records are used to define the mail server accepting email for the domain. The Mail Exchanger record should be a fully qualified domain name (FQDN). (MX) records should not point to the CNAME or IP address as some email systems do not handle these.

PTR - pointer records are used for reverse mapping an IP address to a hostname, which is the opposite of an A Record that is forward mapping a hostname to an IP address. PTR records are found in Reverse DNS zones (in-addr.arpa for IPv4, ipv6.arpa for IPv6)

The Start of Authority (SOA) record is automatically created by OCI DNS when you create a zone. It specifies authoritative information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.

The SRV record allows administrators to use several servers for a single domain. This is useful for load-balancing and backup scenarios.

TXT records are used to hold descriptive, human readable text. May also include non-human readable content for specific uses. It is commonly used for Sender Policy Framework (SPF) records and Domain Key (DKIM) records that do require non-human readable text items.

ALIAS RECORD TYPE -

With enterprises increasingly moving their applications to cloud service providers and CDNs, there's the need to route traffic to named resources - for example - the DNS name of a cloud compute instance rather than its IP addresses takes on increasing importance.

Unfortunately, associating a named resource with the root (or APEX) of a DNS zone can lead to integration challenges with third-party systems.

OCI's Alias record type enables the association of a named resource with the root of their zones without introducing integration issues or any additional overhead.

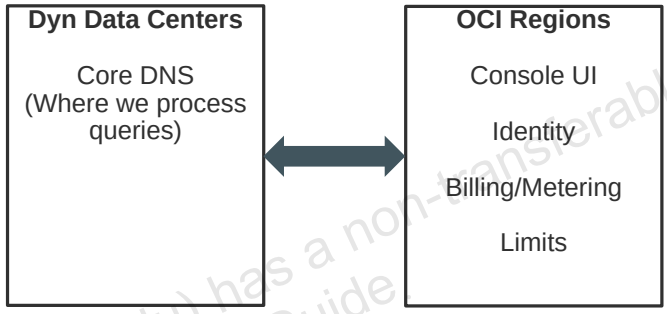
Additional record types supported by DYN include:

DHCID, DNSKEY, KEY, LOC, SSHFP, PX, DNAME, SPF, KX, IPSECKEY, CERT, TLSA, DS, CDS, CDNSKEY, CSYNC

Dyn DNS ⇔ OCI DNS Integration

Integrated Dyn's Global Authoritative DNS network into the OCI environment

- Identity – customers can use Oracle credentials (OCID)
- integration for metering/billing
- Limit Service integration
- OCI Console integration



When DYN merged its capabilities with OCI, OCI inherited all the DYN capabilities.

Currently a subset of DYN's capabilities are exposed through OCI DNS, but there is a rapid progression towards enabling all of the advanced features. This gives Oracle cloud infrastructure users direct access to DYN's network.

There is also integration with billing and metering - there is no separate "bill" for using OCI DNS. Service limits are also enforced by OCI along with API rate limiting.

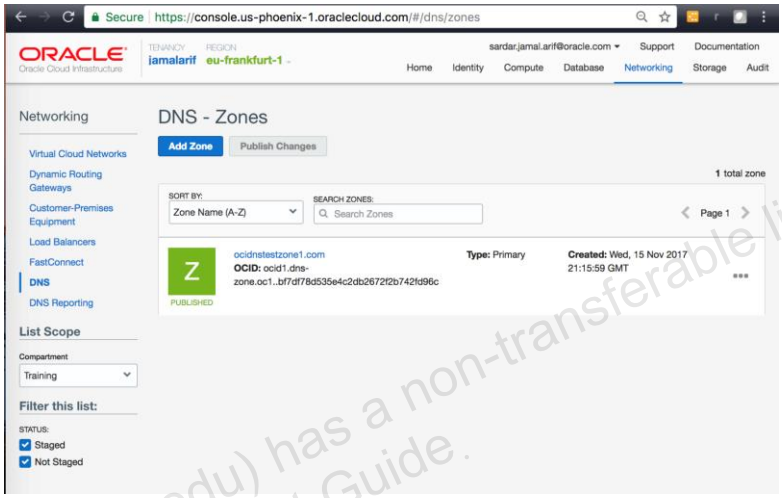
OCI console integration is complete for the initial capabilities and general API, SDK, and terraform are being built out for OCI DNS.

Using OCI DNS

OCI DNS is available in the OCI Console under the "Networking" tab

This will bring the user to the Managed DNS – Zones Screen

From here the user can create Zones to see that the service is working



OCI DNS landing page is the DNS zones tab.

From here existing zones can be managed or new zones can be created.

Zones can also be created from scratch or from a zone file.

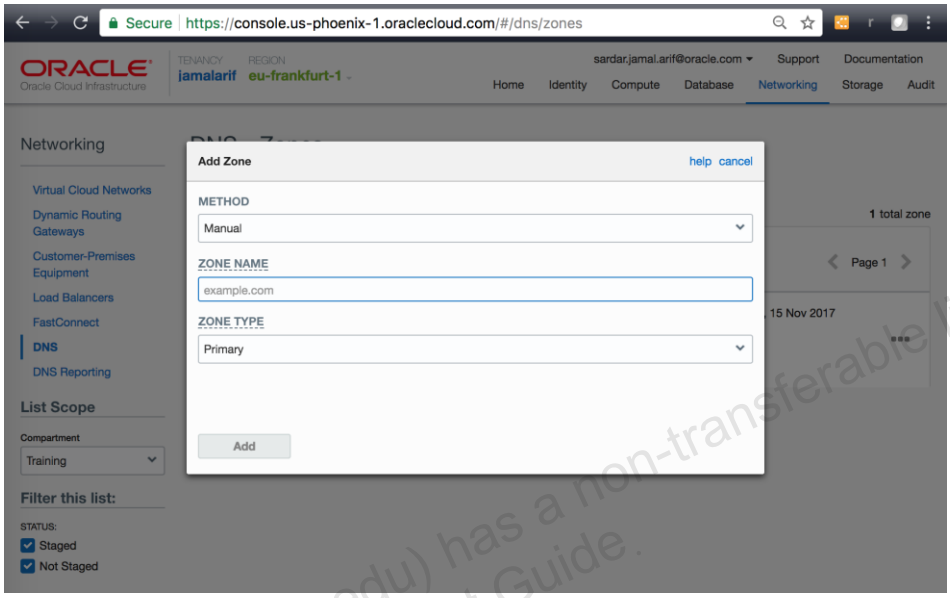
Existing zones can also be deleted or reverted to prior states.

Adding a Zone

From the Managed DNS – Zones page:

- Click “Add Zone”
- Select Method type of “Manual”
- Enter a “Zone Name”
  - i.e. SKUtest.net
- Select Zone Type of “Primary”
- Click “Add Zone”

Zone is created and can be verified from the Managed DNS – Zones page



Adding a zone is a matter of selecting the Add Zone button and selecting a name.

Zones that you are adding should be owned or controlled by you or the name should be unregistered.

A zone file can be used to create a zone - the file must be in RFC 1035 master file format as exported by BIND.

Zone creation takes a short period of time and the domain is available for editing.

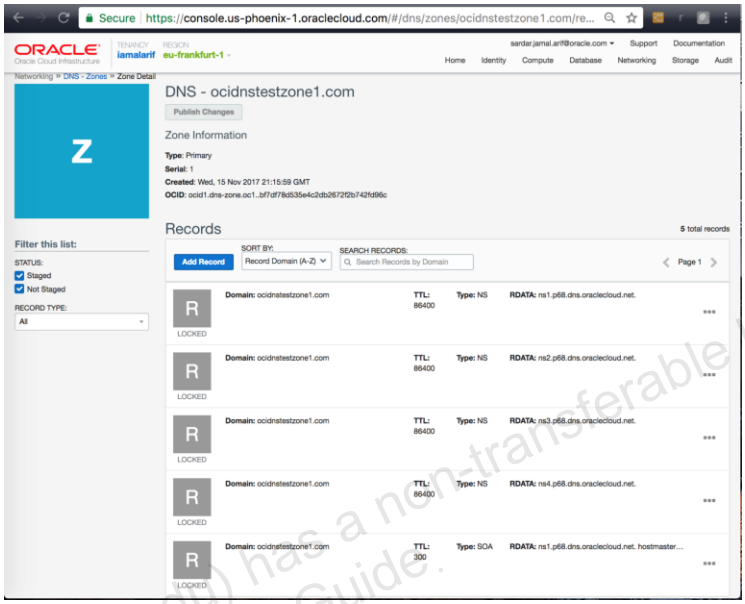
This is also where the decision is made to initially create the zone as primary or secondary.

View/Add Records

From the Managed DNS – Zones page:

- Select a zone to view record details for that zone
- Zone details will show the list of records for that zone
- Select Add Record to add new record
- ”Click “Publish Changes” to update Zone with new record details.

Default NS and SOA records are automatically generated when a Zone is created, so no new records need to be added to generate query data



Once the zone is created records can be added or viewed in the DNS-Zone Detail page.

If a record is selected it can be edited.

Certain records will be locked, like NS and SOA records. Those can be unlocked and relocked from here as well.

Unlocked records can be edited.

Each record type will have a different edit dialog box. Once edits are made they will be unpublished. To push them to a published state, select Publish and apply.

Oracle Cloud Infrastructure DNS: Private Pool and Vanity Nameserver

- **Private Pool** allows enterprises to host their domain names and DNS zones under a dedicated IP pool to segregate from those of other customers in order to reduce the risk of external issues affecting their websites.
  - Example: If multiple customers are in the same pool and one customer's Zones come under a DDoS attack, the other customers in the pool may have their DNS performance impacted until the DDoS is resolved.
- **Vanity Nameserver** allows enterprises to rename OCI Nameservers with their own branding
  - By default, all OCI customers will be hosted on the OCI nameservers. Using standard tools, users can determine that the customer's assets are hosted by OCI DNS.
  - Customers that are concerned about their brand can rebrand the nameservers
  - Example:
    - Default naming: ns1.pxx.dns.oraclecloud.net
    - Vanity naming: ns1.pxx.vanityname.net

OCI DNS is bringing a feature that allows for a private pool of IP addresses to be mapped to a customer. This makes it possible to not get a completely random IP when a new instance is created or an instance is terminated and re-created.

This capability also helps to keep the record editing a little less problematic as records in OCI DNS will at least be pointing to a customer-owned IP address if the instance is terminated.

Another capability of OCI DNS is the vanity nameserver where the forward and reverse names for the customer's hosted DNS nameservers can be customized.

This makes it look like the customer is running their own authoritative nameserver for their zones.

Benefits of OCI DNS

Some of the key benefits include:

- DNS Network operating for over 10+ years, leveraged by thousands of customers, large and small, Enterprise, Business and Web properties
- Support for OCI, other Cloud provider endpoints (AWS, Azure) and private assets, including Cloud, CDNs and Data Centers
- Consistently lowest query latency performance
- Industry leading propagation time to ensure fast response to DNS changes
- **Support for both Primary and Secondary DNS** services, unlike solutions from many Cloud Providers
- Industry's most accurate geolocation data set, created specifically for steering internet traffic
- **DDoS protection** built-in
- Most standards-compliant DNS platform

- Oracle OCI DNS is leveraging DYN's systems and network as-is - DYN is an established, industry-leading solution.
- DYN has over 10 years of runtime and thousands of large customers.
- Some of the larger customers have very large recognition and DNS load.
- OCI DNS can easily provide name services for other cloud providers and other third-party or on-premises assets.
- OCI DNS has excellent performance.
- Propagation times for changes is in the sub-one-minute category.
- A unique feature of OCI DNS is that it can function either as a primary or secondary DNS service unlike some competitive options.
- DDOS protection is built into OCI DNS.
- OCI DNS is the most standards-compliant DNS platform which adheres to the relevant RFCs as closely and interoperably as possible - OCI DNS is designed to play nice with everyone.

Summary

In this lesson, you should have learned the following:

- OCI DNS Service
- How to configure an OCI DNS Zone

In this lesson we went over the OCI DNS service, some of the OCI DNS service features, and learned how to configure and manage an OCI DNS zone.