Contents

1 Getting Started with Oracle Cloud Infrastructure
   Objectives   1-2
   Oracle Cloud Infrastructure Strategy   1-3
   Oracle Cloud Infrastructure: Overview   1-4
   Regions and Availability Domains   1-5
   Off-box Network Virtualization   1-6
   Key Differentiators   1-7
   Summary   1-8

2 Identity and Access Management Service
   Objectives   2-2
   Identity and Access Management Service   2-3
   IAM Service Resources - Tenants, Compartments   2-4
   IAM Service Resources - Users, Groups   2-5
   IAM Service   2-6
   Policies   2-7
   IAM Service   2-8
   Policies   2-9
   Policy Examples   2-10
   IAM Service resources are global   2-11
   Resource Locations   2-12
   Signing Up for Oracle Cloud Infrastructure Services   2-14
   Signing In to the Console   2-15
   Resource Identifier   2-16
   Resource Identifier – tenancy OCID   2-17
   Resource Identifier – instance OCID   2-18
   Security Credentials   2-19
   Summary   2-20
   Practice 2: User, Group, and Policy Management   2-21

3 Virtual Cloud Network Service
   Objectives   3-2
   Virtual Cloud Network (VCN)   3-3
   Basic Networking   3-4
   Subnet   3-5
IGW, DRG 3-6
Security Lists, Route Table 3-7
Stateful Security Lists 3-8
Stateless Security Lists 3-9
Default Security List 3-10
Default VCN components 3-11
Public Subnet 3-12
Private Subnet with a VPN 3-13
DNS Choice 3-15
DHCP Configuration 3-16
FastConnect 3-17
Off-box Network Virtualization 3-18
Bandwidth and Latency between BM instances 3-19
Summary 3-20
Practice 3: Network Management 3-21

4 Compute Service
Objectives 4-2
Compute: Bare Metal & Virtual Machines 4-3
Shape: Processor and Memory Resources 4-4
Available Shapes 4-5
NVMe SSD Devices 4-6
Protecting NVMe SSD Devices 4-7
BM.HighIO1.512 Options 4-8
BM.DenseIO1.512 Options 4-9
Images 4-10
Custom Images 4-11
Launching a Compute Instance 4-12
Creating a Key Pair 4-13
Choosing a Compartment 4-14
Using a Virtual Cloud Network 4-15
Launching an Instance 4-16
Getting the Public IP Address 4-17
Using a Block Volume 4-18
Attaching Volume to an Instance 4-19
Summary 4-20
Practice 4: Instance Management 4-21

5 Block Volume and Object Storage Service
Objectives 5-2
Storage Services 5-3
Overview of Block Volume Service 5-4
Block Volume Service Components 5-5
How Can I Use Block Storage with My Instance? 5-6
Creating and Attaching a Block Volume Using the Console 5-7
Managing Block Storage Volumes 5-8
Backup and Restoration 5-9
What’s a Mount Point? 5-10
Detaching and Deleting Block Volumes 5-11
Performance Benchmark 5-12
Overview of Object Storage Service 5-13
Object Storage Elements 5-15
Object Storage Service Features 5-16
Managing Buckets and Objects 5-17
Managing Multipart Uploads 5-18
Summary 5-20
Practice 5: Storage Management 5-21

6 Load Balancing Service
Objectives 6-2
Load Balancing Service 6-3
Public/Private LB 6-4
All LB Concepts 6-5
Load Balancing Service: Shapes 6-6
Load Balancing Service: Protocol Support 6-7
Public Load Balancer example configuration 6-8
Summary 6-9
Practice 6: Implementing Public Load Balancer and High Availability 6-10

7 Database Service
Objectives 7-2
Oracle Cloud Infrastructure – Database Service 7-3
Use Cases 7-4
Exadata DB Systems 7-5
Exadata System Configuration 7-6
What’s New with Exadata DB Systems? 7-7
Database Backup to IaaS Object Store 7-8
Scaling Exadata DB Systems 7-9
Bare Metal Database System 7-10
Shapes for Bare Metal Database Systems 7-11
Storage 7-12
Managing the Database Systems 7-13
Getting Started with Oracle Cloud Infrastructure
September 2017
Objectives

After completing this lesson, you should be able to:

- Describe Oracle Cloud Infrastructure
- Explain typical use cases for Oracle Cloud Infrastructure
Our strategy is to give customers the best cloud applications and platform, partner with a broad and open ecosystem, and run these technologies on the best infrastructure, either in the cloud or on-premises, or both.
Oracle Cloud Infrastructure combines the elasticity and utility of public cloud with the granular control, security, and predictability of on-premises infrastructure to deliver high-performance and cost-effective infrastructure services.

Oracle Cloud Infrastructure is the first cloud platform to implement off-box network virtualization. The off-box network virtualization takes network and IO virtualization out of the software stack and puts it in the network. As a result, customers can provision truly elastic, self-service, pay-as-you-go physical, dedicated hosts with no hypervisor overhead, noisy neighbors or shared resources with a full software-defined layer 3 network topology.

In addition, the off-box network virtualization enables you to run bare metal hosts side-by-side with any class of systems – from Virtual Machines (VMs) to Engineered Systems such as Exadata, all using the same set of APIs. This implies that you can leverage Exadata hardware (such as InfiniBand) and software (such as smart scan, flash cache, columnar compression) features for your applications while leveraging the cloud-native security and governance capabilities of a layer 3 virtual cloud network.
Disaster recovery is a salient feature of cloud computing. In the case of Oracle Cloud Infrastructure, while the availability domains provide the facility for high availability, regions provide the basis for disaster recovery. Regions are completely independent of other regions and can be separated by vast distances—across countries or even continents. Generally, you would deploy an application in the region where it is most heavily used, since using nearby resources is faster than using distant resources. However, you can also deploy applications in different regions to:

- Mitigate the risk of region-wide events, such as large weather systems or earthquakes
- Meet varying requirements for legal jurisdictions, tax domains, and other business or social criteria
Generally the network virtualization is rendered by relying on the hypervisor [the hardware virtualization layer]. However, with Off-box Virtualization, the hypervisor layer is removed and network virtualization is run on the hardware directly. This increases network performance and more importantly gives a higher level of security by providing isolation. So that even if the hypervisor layer is breached, the attack remains localized to that single virtual network and does not permeate to other virtual networks.
Key Differentiators

Enterprise IaaS Architecture

• Industry’s first Bare Metal Cloud Services w/ support for key enterprise apps
• Off-Box Network Virtualization (w/ support for plugging Exadata appliances)
• Robust Security and Governance capabilities
• Flexibility and control (Bare Metal and VMs share the same set of APIs)

Industry Leading Price Performance

• Lower compute costs than AWS EC2 compute
• Fast, predictable block storage with no additional cost for IOPS; multiple X cheaper than AWS
• Bandwidth costs cheaper than AWS bandwidth by 85%
• Non-oversubscribed network, predictable performance with low latency and high throughput
• Industry leading 25 Gb/s network fabric (to be launched at OOW17)
Summary

In this lesson, you should have learned how to:

• Describe Oracle Cloud Infrastructure
• Explain typical use cases for Oracle Cloud Infrastructure
Identity and Access Management Service
September 2017
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
Identity and Access Management Service

- Identity and Access Management Service (IAM) lets you control who has access to your cloud resources
- A Resource is a cloud object that you create and use in Oracle Cloud Infrastructure Service
  - Example: Compute instances, block storage volumes, Virtual Cloud Networks (VCNs), subnets, route tables, and so on are resources
- IAM concepts – Tenancy, Compartments, Users, Groups, Policies
IAM Service Resources - Tenants, Compartments

**Tenancy**
- Equivalent of an account; tenancy contains all of your Bare Metal Cloud Services 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 hierarchical; permissions in a parent compartment are inherited by child compartments (*currently compartments are only one level deep*)
- Compartments are global and logical; distinct from *physical containers* like Regions and Availability Domains
- Resources can be connected/shared across compartments
IAM Service Resources - Users, Groups

**Users**
- Users can be created and given console passwords to use the web console and/or API signing keys to use the REST API and SDKs
- User must be placed in groups to be given access to cloud resources
  - A new user has no permissions until you place the user in one or more groups and there's at least one policy that gives that group permission to either the tenancy or a compartment
- Users can be members of multiple groups

**Groups**
- Used to grant privileges to cloud resources
- A group has no permissions until you write at least one policy that gives that group permission to either the tenancy or a compartment
Policies

- Supports security principle of least privilege; by default, users are not allowed to perform any actions
- Policies are comprised of 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>]
  - Example: Allow group ProjectA_Admins to manage all-resources in compartment ProjectA_compartment
Tenancy

**Users**
- User_1
- User_2

**Groups**
- group_X
- group_Y

**Policies**
- **PolicyA**: Allow group_X to manage all-resources in compartmentA
- **PolicyB**: Allow group_Y to manage all-resources in compartmentB
Policies

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

<table>
<thead>
<tr>
<th>verb</th>
<th>Type of access</th>
<th>Aggregate resource-type</th>
<th>Individual resource type</th>
</tr>
</thead>
<tbody>
<tr>
<td>inspect</td>
<td>Read only access without access to any user-specified metadata</td>
<td>all-resources</td>
<td>db-systems, db-nodes, db-homes, databases</td>
</tr>
<tr>
<td>read</td>
<td>Read only access, plus the ability to get user-specified metadata</td>
<td>database-family</td>
<td>instances, instance-images, volume-attachments, console-histories</td>
</tr>
<tr>
<td>use</td>
<td>Update existing resources, but not create or delete</td>
<td>instance-family</td>
<td></td>
</tr>
<tr>
<td>manage</td>
<td>Includes all permissions for the resource</td>
<td>object-family</td>
<td>buckets, objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>virtual-network-family</td>
<td>vcn, subnet, route-table, more</td>
</tr>
<tr>
<td></td>
<td></td>
<td>volume-family</td>
<td>Volumes, volume-attachments, volume-backups</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Aggregate Resource Types</th>
<th>Individual Resource Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow group <strong>Admins</strong> to <strong>manage all-resources</strong> in tenancy</td>
<td>allow group <strong>NetAuditors</strong> to <strong>manage subnet</strong> in compartment IT</td>
</tr>
<tr>
<td>allow group <strong>HRAdmins</strong> to <strong>use all-resources</strong> in compartment <strong>HR</strong></td>
<td>allow group <strong>CompSec</strong> to <strong>use console-histories</strong> in tenancy</td>
</tr>
<tr>
<td>allow group <strong>NetAuditors</strong> to <strong>read virtual-network-family</strong> in tenancy</td>
<td>allow group <strong>ServerAdmins</strong> to <strong>read instances</strong> in compartment IT</td>
</tr>
<tr>
<td>allow group <strong>ServerAdmins</strong> to <strong>inspect instance-family</strong> in tenancy</td>
<td>allow group <strong>VolumeBackupAdmins</strong> to <strong>inspect volumes</strong> in tenancy</td>
</tr>
</tbody>
</table>
IAM Service resources are global

IAM Service resources (compartments, users, groups, and policies) are global, so you can access them across all regions.
## Resource Locations

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

<table>
<thead>
<tr>
<th>Service</th>
<th>Resource</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Balancer</td>
<td>Load Balancer</td>
<td>Region</td>
</tr>
<tr>
<td>Storage</td>
<td>Buckets</td>
<td>Region</td>
</tr>
</tbody>
</table>

Bucket is a regional resource but it can be accessed from any location as long as the correct region-specific URL is used.
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

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.
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..aaaaaaaaaxy6bh46cdnlfpaibasc6dotowv32hc2sbj4ph3ocxtfxhhva2hna
  - instance:
    ocid1.instance.oc1.iad.abuwcljtwfk7f5e2o3q6ircgdty6rg52tdyg72tgdtbiwqlujt7vm5h3da
<table>
<thead>
<tr>
<th>Resource Identifier – instance OCID</th>
</tr>
</thead>
</table>

![Instance OCID Image](image-url)

**Instance OCID**

**BM**

**Instance Information**

- **Availability Domain:** dKYS.US.Ashburn.NA.1
- **OCID:** ocs1.instance.oc1.iad.abcwzdj2fks2d9qqlgd8y2tbgfr2g2t8bs3h8bqy77wm9x3da
- **Launched:** Fri, 07 Jul 2017 21:12:16 GMT
- **Compartment:** intercalifith

**Primary VNIC Information**

**Image:** Oracle-Linux:7.4.2017.06.23
**Region:** iad
**Shape:** VM.Standard1.16
**Virtual Cloud Network:** VCN-Ashburn

---

*Oracle Cloud Infrastructure Fundamentals 2 - 18*
Security Credentials

Access to different interfaces requires appropriate credentials

- **Console 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.

Console Password:

- Sign in to the web console the first time with the one-time password, and change the password, when prompted. Password requirements are shown in the console.
- The one-time password expires in 7 days. You can change the password later.
- Also, you or an administrator can reset the password using the console or the API. Resetting the password creates a new one-time password that you'll be prompted to change the next time you sign in to the console. If you're blocked from signing in to the console because you've tried 10 times in a row unsuccessfully, contact your administrator.

API Signing Key:

- After you've uploaded your first API key in the console, you can use the API to upload any additional ones you want to use. If you provide the wrong kind of key (for example, your instance SSH key, or a key that isn't at least 2048 bits), you'll get an InvalidKey error.

You can upload your PEM public key in the Console:

- Open the Console, and sign in.
- Click your username in the top-right corner of the Console, and then click User Settings.
  - If you're an administrator doing this for another user, instead click Identity, click Users, and then select the user from the list.
- Click Add Public Key. Paste the contents of the PEM public key in the dialog box and click Add.
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
This is the first practice session. In this session, you explore the Oracle Cloud Infrastructure environment that you have been provided. You will also set up users, group, and security policy rules that you will use to build your highly available WordPress environment.

All participants share one tenancy. Each participant will work in their own compartment to create and configure resources to set up the application in a highly available configuration.
Virtual Cloud Network Service
September 2017
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
Virtual Cloud Network (VCN)

- A Virtual Cloud Network is a virtual version of a traditional network—including subnets, route tables, and gateways—on which your instances run. A cloud network resides within a single region but can cross multiple Availability Domains.
- A VCN covers a single, contiguous IPv4 CIDR block of your choice.
- Recommend using one of the private IP address ranges in RFC 1918 (10.0.0.0/8, 172.16/12, and 192.168/16). However, you can use a publicly-routable range.
- Allowable VCN size range: /16 to /30.
- VCN reserves the first two IP addresses and the last one in each subnet's CIDR.
Basic Networking

- CIDR notation: an IP address and its associated routing prefix
- 0.0.0.0/0 = entire IPv4 range (whole Internet)
- x.x.0.0/16 = class B network (65,536 IP addresses)
- 10.0.0.0/16 = 65,536 IP addresses (10.0.1.0 – 10.1.255.255)
- IP address 192.168.0.15 w/ netmask 255.255.255.0 in CIDR notation = 192.168.0.15/24 (the first 24 bits of the IP address given are considered significant for the network routing)
- RFC 1918 IP Blocks – Private IP address blocks
  - 10.0.0.0 - 10.255.255.255
  - 172.16.0.0 - 172.31.255.255
  - 192.168.0.0 - 192.168.255.255
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.
- Instances draw their internal IP address and network configuration from their subnet.
Internet Gateway: Internet Gateway provides a path for network traffic between your VCN and the internet.

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 through IPSec VPN or FastConnect.

After creating an IGW or attaching a DRG, you must add a route for the IGW/DRG in the VCN's route table 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 through 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.
Here are some characteristics of the default security list:

- **Stateful ingress**: Allow TCP traffic on destination port 22 (SSH) from source 0.0.0.0/0 and any source port. This rule makes it easy for you to create a new cloud network and public subnet, launch a Linux instance, and then immediately connect through SSH to that instance without needing to write any security list rules yourself.
  - The default security list does not include a rule to allow Remote Desktop Protocol (RDP) access. If you're using Windows images, make sure to add a stateful ingress rule for TCP traffic on destination port 3389 from source 0.0.0.0/0 and any source port.

- **Stateful ingress**: Allow ICMP traffic type 3 code 4 from source 0.0.0.0/0 and any source port. This rule makes it easy to receive Path MTU Discovery fragmentation messages if you're using jumbo frames.

- **Stateful ingress**: Allow ICMP traffic type 3 (all codes) from your VCN's CIDR [Classless Inter-Domain Routing] IPs and any source port. This rule makes it easy for your instances to receive connectivity error messages from other instances within the VCN.

- **Stateful egress**: No rules defined to allow all traffic. This allows instances to initiate traffic of any kind to any destination. Notice that this means the instances can talk to any Internet IP address if the cloud network has an Internet Gateway. And because stateful security rules use connection tracking, the response traffic is automatically allowed regardless of any ingress rules.
Default VCN components

Your VCN automatically comes with some default components:
- Default route table
- Default security list
- Default set of DHCP options

You can’t delete these default components; however, you can change their contents (for example: individual route rules). And you can create more of each kind of component in your cloud network (for example: additional route tables).
Public Subnet

- Create a VCN, provide a CIDR range
- Create an Internet Gateway
- Create a Route Rule with traffic to Internet Gateway (for all IP addresses, 0.0.0.0/0)
- Create Security List rules that allow the traffic (and each instance’s firewall must allow the traffic)
- Create a Public Subnet within a specific AD with the Route Table and Security List
- Create an instance with a public IP address within the Subnet
Private Subnet with a VPN

- Create a VCN, provide a CIDR range
- Create a Dynamic Routing Gateway (DRG); attach it to the VCN
- Create a new Route Table so its default route is directed toward DRG and thus to the VPN
- Create a Route Rule with traffic to DRG - add a CIDR block of 0.0.0.0/0 (all non-intra-VCN traffic that is not already covered by other rules in the route table will go to the DRG)
- Create Security List rules that allow the traffic (for example: port 1521 for Oracle databases)
- Create a Private Subnet within a specific AD with the Route Table and Security List
Private Subnet with a VPN

- Create an IPSec connection for VPN
- Data center admin must configure the on-premises router before network traffic can flow between your on-premises network and VCN
- At your end of the IPSec VPN is the actual router in your on-premises network (hardware or software). A virtual representation of the router in Bare Metal Cloud Services is referred to as Customer-Premises Equipment (CPE)
If you choose to use the default option of DNS, that is, Internet and VCN Resolver with DNS Hostnames Across the VCN, then all instances in the VCN can communicate with each other without knowing their IP addresses. Make sure to assign a DNS label to the VCN and every subnet. Then make sure to assign every instance a host name (or at least a display name) at launch. The instances can then communicate with each other using FQDNs instead of IP addresses. If you also set the Search Domain DHCP option to the VCN domain name, the instances can then communicate with each other using just <hostname>.<subnet DNS label>.<VCN DNS label>.oraclevcn.com instead of the FQDN.

If you use Custom DNS Servers to Resolve DNS Hostnames, then you can set up an instance to be a custom DNS server within your VCN and configure that instance to resolve the hostnames for your instances. You must configure the servers to use 169.254.169.254 as the forwarder for the VCN domain.
Your cloud network uses DHCP options to automatically provide configuration information to the instances when they boot up. Each cloud network comes with a default set of DHCP options with an initial value that you can change. If you don’t specify otherwise, every subnet will use the VCN’s default set of DHCP options.

You can’t change which set of DHCP options is associated with a subnet after the subnet is created. If you don’t want to use the default set, make sure to create your desired set of DHCP options before creating the subnet. However, remember that you can also change the values for the options. Whenever you change the value of one of the DHCP options, you need to either restart the DHCP client on the instance, or reboot the instance, for the change to take effect on existing instances in the subnets associated with that set of DHCP options.

Be sure to keep the DHCP client running so you can always access the instance. If you stop the DHCP client manually or disable Network Manager, the instance can’t renew its DHCP lease and will become inaccessible when the lease expires (typically within 24 hours). Do not disable Network Manager unless you use another method to ensure renewal of the lease. Stopping the DHCP client might remove the host route table when the lease expires. Also, loss of network connectivity to your iSCSI connections might result in loss of the boot drive.
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
We use ‘Off Box Network Virtualization’. Note that the virtualization layer is well isolated from the Bare-Metal nodes and as a result, it is much harder for a bad actor to compromise the virtualization layer.
### Bandwidth and Latency between BM instances

<table>
<thead>
<tr>
<th>[opc@iperf-client ~]$ sudo iperf3 -c 10.0.0.5</th>
<th>Connecting to host 10.0.0.5, port 5201</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ID] Interval</td>
<td>Transfer</td>
</tr>
<tr>
<td>00-01.00</td>
<td>sec 1.13 GBytes 9.67 Gbits/sec</td>
</tr>
<tr>
<td>00-03.00</td>
<td>sec 1.15 GBytes 9.85 Gbits/sec</td>
</tr>
<tr>
<td>00-04.00</td>
<td>sec 1.15 GBytes 9.86 Gbits/sec</td>
</tr>
<tr>
<td>00-05.00</td>
<td>sec 1.15 GBytes 9.87 Gbits/sec</td>
</tr>
<tr>
<td>00-06.00</td>
<td>sec 1.15 GBytes 9.87 Gbits/sec</td>
</tr>
<tr>
<td>00-07.00</td>
<td>sec 1.15 GBytes 9.87 Gbits/sec</td>
</tr>
<tr>
<td>00-08.00</td>
<td>sec 1.15 GBytes 9.87 Gbits/sec</td>
</tr>
<tr>
<td>00-09.00</td>
<td>sec 1.15 GBytes 9.86 Gbits/sec</td>
</tr>
<tr>
<td>00-10.00</td>
<td>sec 1.15 GBytes 9.87 Gbits/sec</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>[opc@iperf-client ~]$ sudo iperf3 -c 129.213.56.64</th>
<th>Connecting to host 129.213.56.64, port 5201</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ID] Interval</td>
<td>Transfer</td>
</tr>
<tr>
<td>00-01.00</td>
<td>sec 666 MBytes 5.59 Gbits/sec</td>
</tr>
<tr>
<td>00-02.00</td>
<td>sec 462 MBytes 3.88 Gbits/sec</td>
</tr>
<tr>
<td>00-03.00</td>
<td>sec 462 MBytes 3.88 Gbits/sec</td>
</tr>
<tr>
<td>00-04.00</td>
<td>sec 461 MBytes 3.87 Gbits/sec</td>
</tr>
<tr>
<td>00-05.00</td>
<td>sec 462 MBytes 3.88 Gbits/sec</td>
</tr>
<tr>
<td>00-06.00</td>
<td>sec 476 MBytes 3.99 Gbits/sec</td>
</tr>
<tr>
<td>00-07.00</td>
<td>sec 491 MBytes 4.12 Gbits/sec</td>
</tr>
<tr>
<td>00-08.00</td>
<td>sec 486 MBytes 4.08 Gbits/sec</td>
</tr>
<tr>
<td>00-09.00</td>
<td>sec 480 MBytes 4.03 Gbits/sec</td>
</tr>
<tr>
<td>00-10.00</td>
<td>sec 482 MBytes 4.05 Gbits/sec</td>
</tr>
</tbody>
</table>
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
Practice 3: Network Management

In this practice, each participant uses their assigned compartment and:

- Creates a virtual cloud network (VCN)
- Creates a subnet within the VCN

![Diagram showing network management practices with compartments and subnets]
Compute Service

September 2017
Objectives

After completing this lesson, you should be able 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
- Add block volume to a compute instance
Latency: Same Random and Sequential: ~90 μsec Read, ~20 μsec Write
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.

Oracle Compute Cloud Service enables you to select from a range of predefined shapes that determine the number of CPUs available in an instance and the amount of RAM available in an instance.

Several predefined shapes are available for both bare metal and virtual machine instances.
In the 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.

<table>
<thead>
<tr>
<th>Shape</th>
<th>Instance type</th>
<th>OCPU</th>
<th>RAM (GB)</th>
<th>Local Disk (TB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM.Standard1.36</td>
<td>Standard compute capacity</td>
<td>36</td>
<td>256</td>
<td>Block Storage only</td>
</tr>
<tr>
<td>BM.HighIO1.36</td>
<td>High I/O compute capacity</td>
<td>36</td>
<td>512</td>
<td>12.8 TB NVMe SSD</td>
</tr>
<tr>
<td>BM.DenseIO1.36</td>
<td>Dense I/O compute capacity</td>
<td>36</td>
<td>512</td>
<td>28.8 TB NVMe SSD</td>
</tr>
<tr>
<td>VM.Standard1.1</td>
<td>Standard</td>
<td>1</td>
<td>7</td>
<td>Block Storage only</td>
</tr>
<tr>
<td>VM.Standard1.2</td>
<td>Standard</td>
<td>2</td>
<td>14</td>
<td>Block Storage only</td>
</tr>
<tr>
<td>VM.Standard1.4</td>
<td>Standard</td>
<td>4</td>
<td>28</td>
<td>Block Storage only</td>
</tr>
<tr>
<td>VM.Standard1.8</td>
<td>Standard</td>
<td>8</td>
<td>56</td>
<td>Block Storage only</td>
</tr>
<tr>
<td>VM.Standard1.16</td>
<td>Standard</td>
<td>16</td>
<td>112</td>
<td>Block Storage only</td>
</tr>
<tr>
<td>VM.DenseIO1.4</td>
<td>Dense I/O compute capacity</td>
<td>4</td>
<td>60</td>
<td>3.2 TB NVMe SSD</td>
</tr>
<tr>
<td>VM.DenseIO1.8</td>
<td>Dense I/O compute capacity</td>
<td>8</td>
<td>120</td>
<td>6.4 TB NVMe SSD</td>
</tr>
<tr>
<td>VM.DenseIO1.16</td>
<td>Dense I/O compute capacity</td>
<td>16</td>
<td>240</td>
<td>12.8 TB NVMe SSD</td>
</tr>
</tbody>
</table>
NVMe SSD Devices

- Locally attached SSDs are not protected
- Bare Metal Cloud Service provides no RAID, snapshots, backups capabilities for these devices
- Customers are responsible for the durability of data on the local SSDs

<table>
<thead>
<tr>
<th>Instance type</th>
<th>NVMe SSD Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM.HighI0.1.512</td>
<td>4 drives = 12.8TB raw</td>
</tr>
<tr>
<td>BM.DenseI0.1.512</td>
<td>9 drives = 28.8TB raw</td>
</tr>
<tr>
<td>VM.DenseI0.1.4</td>
<td>1 drive = 3.2 TB raw</td>
</tr>
<tr>
<td>VM.DenseI0.1.8</td>
<td>2 drives = 6.4 TB raw</td>
</tr>
<tr>
<td>VM.DenseI0.1.16</td>
<td>4 drives = 12.8 TB raw</td>
</tr>
</tbody>
</table>

```bash
ubuntui@nvme:~$ lsblk
NAME   MAJ:MIN  RM  SIZE RO TYPE  MOUNTPOINT
sda     8:0     0 46.6G 0 disk
├─sda1   8:1     0  4.6G 0 part /
│ └─sda14 8:14   0  4.6G 0 part
└─sda15  8:15   0  100M 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
```
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.
BM.HighIO1.512 Options

RAID 10 across all 4 SSDs with 6.4 TB usable space, can survive the failure of one device; fast performance

RAID 6 across all 4 SSDs with 6.4 TB usable space, but can survive the failure of two devices; slower, but higher durability
### 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

```bash
$ 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
```
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.

<table>
<thead>
<tr>
<th>Image</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Linux 7 Unbreakable Enterprise Kernel Release 4</td>
<td>Oracle-Linux-7.x-</td>
<td>The UEK is Oracle's optimized operating system kernel</td>
</tr>
<tr>
<td>Oracle Linux 6 Unbreakable Enterprise Kernel Release 4</td>
<td>Oracle-Linux-6.x-</td>
<td>The UEK is Oracle's optimized operating system kernel</td>
</tr>
<tr>
<td>CentOS 7</td>
<td>CentOS-7-x</td>
<td>CentOS is a free, open-source Linux distribution</td>
</tr>
<tr>
<td>CentOS 6</td>
<td>CentOS-6.x-</td>
<td>CentOS is a free, open-source Linux distribution</td>
</tr>
<tr>
<td>Ubuntu 16.04 LTS</td>
<td>Canonical-Ubuntu-16.x-&lt;date&gt;&lt;number&gt;</td>
<td>Ubuntu is a free, open-source Linux distribution</td>
</tr>
</tbody>
</table>
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.
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.

## Launching a Compute Instance

The steps to launch a compute instance are:

1. Create a Key Pair
2. Choose a Compartment
3. Create a Virtual Cloud Network
4. Launch an Instance
5. Connect to Your Instance
6. Add a Block Volume
Creating a Key Pair

- Instances use an SSH key pair instead of a password to authenticate a remote user.
- A key pair file contains a private key and public key.
- You keep the private key on your computer and provide the public key every time you launch an instance.
- To create key pairs, you can use a third-party tool such as:
  - OpenSSH on UNIX-style systems (including Linux, Solaris, BSD, and OS X)
    ```
    ssh-keygen -t rsa -N "" -b 2048 -C "<key_name>" -f <path/root_name>
    ```
  - PuTTY Key Generator on Windows

While use of PuTTY is shown in the slide for accessing from Windows environments, you could also install a bash shell – such as the Ubuntu based bash shell or Git bash – in a Windows environment. When you use a bash environment, the Linux commands work the same way in bash shell in Windows environment.
As we already mentioned, compartment is a collection of related resources that can be accessed only by those groups that have permission. For example, one compartment could contain all the instances and storage volumes that make up the production version of your company's Human Resources system. Only users with permission to that compartment can manage those instances and volumes.

The compute instances, or any resource for that matter, once created in a compartment cannot be moved to another compartment. You can however create an image and using that image clone a compute instance to another compartment within your tenancy.
Before you can launch an instance, you need to have a Virtual Cloud Network (VCN). In the VCN, you launch the instance into a subnet. A subnet is a subdivision of your VCN that you define in a single Availability Domain. The subnet directs traffic according to a route table. The subnet also uses a security list to control traffic in and out of the instance.

When you created a VCN, you would have noted the details of the VCN that you just created. The VCN has the following resources and characteristics:

- **CIDR block range of 10.0.0.0/16**
- **An Internet Gateway**
- **A route table with a default route rule to enable traffic to and from the Internet Gateway**
- **A Default Security List that allows specific ingress traffic to and all egress traffic from the instance**
- **A public subnet in each Availability Domain.**
- **The VCN will automatically use the Internet and VCN Resolver for DNS.**
The instance is displayed in the Console in a provisioning state. Expect provisioning to take a few minutes before the status changes to Running. Do not refresh the page. Once the instance is running, wait a few more minutes for the operating system to boot before you attempt to connect.

The shape you select determines the number of CPUs, memory to be allocated to your Compute instances.
The public IP address of your instance is what you need to connect to the instance and configure other resources within that instance.

Use the following SSH command to access the instance. Enter the passphrase welcome1 when prompted.

```
$ ssh opc@<public-ip-address>
```

•  <public-ip-address> is your instance IP address that you retrieved from the Console.
Using a Block Volume

- In the Console, click Storage > Block Volumes.
- Click Create Block Volume.
- In the Create Block Volume dialog box, enter the following:
  - **Create in Compartment:** Select the compartment in which you want to create the volume.
  - **Name:** Enter a user-friendly name.
  - **Availability Domain:** Select the same Availability Domain that you selected for your instance.
  - **Size:** Select an appropriate size.
- Click Create Block Volume.

Block Volume Service provides network storage to use with your Compute instances. After you create, attach, and mount a volume to your instance, you can use it just as you would a physical hard drive on your computer. A volume can be attached to a single instance at a time, but you can detach it from one instance and attach to another instance, keeping your data intact.
Challenge-Handshake Authentication Protocol (CHAP) is a security protocol. When you set up your production environment, Oracle recommends that you use CHAP credentials.

Attaching Volume to an Instance

• In the Console, click Compute and then click Instances.
• Click your instance name to view its details.
• Click Attach Block Volume.
• In the dialog box, enter or select the following:
  – Block Volume Compartment: Select the compartment where you created the block volume.
  – Block Volume: Select the block volume from the list.
  – Require CHAP Credentials
• Click Attach.
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
- Add block volume to a compute instance
Practice 4: Instance Management

In this practice, each participant uses their assigned compartment and:
• Launches an Oracle Linux VM
• Attaches the block volume created in the previous practice
• Mounts the block volume and transfers some content
• Customizes the instance and deploys the LAMP stack followed by WordPress
Block Volume and Object Storage Service
September 2017
Objectives

After completing this lesson, you should be able to:

• Create, attach, configure, and mount block volumes
• Back up and restore block volumes
• Detach and delete block volumes
• Describe concepts and uses of object storage
• Create object storage
• Upload objects to object storage
• Upload multipart objects to object storage
Storage Services

Oracle Cloud Infrastructure offers two main storage services

- **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.

- **Object Storage Service**
  - Object storage is independent of a server and accessed over the Internet
  - Data is managed as objects using an API built on standard HTTP verbs
  - It is an ideal storage platform to store very large amounts of data
A common usage of Block Volume Service is to add storage capacity to an instance. To use a block storage volume, you should:

- Create a block storage volume through the console or the API
- Attach the volume to an instance using a volume attachment
- Connect to the volume from your instance's guest OS using iSCSI
- Mount the volume and use within your instance

A Block Volume Service volume can be detached from an instance and moved to a different instance without loss of data. This data persistence allows you to easily migrate data between instances and ensures that your data is safely stored, even when it is not connected to an instance. Any data will remain intact until you reformat or delete the volume.

To move your volume to another instance, unmount the drive from the initial instance, terminate the iSCSI connection, and attach it to the second instance. From there, you simply connect and mount the drive from that instance's guest OS to instantly have access to all of your data. Additionally, Block Volume Service volumes offer a high level of data durability compared to standard, attached drives. All volumes are automatically replicated for you, helping to protect against data loss.
The Internet Small Computer System Interface (iSCSI) is an IP-based standard for connecting storage devices. iSCSI encapsulates SCSI commands in IP network packets, which allows data transfer over long distances and sharing of storage by client systems. As iSCSI uses the existing IP infrastructure, it does not require the purchase and installation of fiber-optic cabling and interface adapters that are needed to implement Fibre Channel (FC) storage area networks.

Oracle Linux supports iSCSI initiator functionality in software. The kernel-resident device driver uses the existing network interface card (NIC) and network stack to emulate a hardware iSCSI initiator. As the iSCSI initiator functionality is not available at the level of the system BIOS, you cannot boot an Oracle Linux system from iSCSI storage.
A storage volume is a virtual disk that provides persistent block storage for Compute instances. You can use storage volumes to store data and applications.

Block Volume Service, a part of Oracle Cloud Infrastructure, allows you to:

- Create block storage volumes and attach them to your Compute instances. When you create a storage volume, you can specify the capacity that you need.
- Attach one or more storage volumes to an instance either while creating the instance or later, while the instance is running.
- Scale up or scale down the block storage capacity for the instance by attaching or detaching storage volumes even while the instance is running. Also, remember that, when a storage volume is detached from an instance or when the instance is deleted, data stored on the storage volume is not lost.
Creating and Attaching a Block Volume Using the Console

To create a Block Volume:
In the console, click Storage.
• Click Create Block Volume.
• Fill in the required volume information:
  - **Name**: A user-friendly name or description.
  - **Domain**: Must be in the same Availability Domain as the instance.
  - **Size**: Can be between 50 GB to 2TB.
• Click Create.
The volume will be ready to attach once its icon no longer lists it as PROVISIONING in the volume list.

To Attach a Block Volume:
In the Console, click Compute.
• In the Instances list, select the instance you want to attach to the volume.
• Click the name of the instance to display the instance details.
• Click Attach Volume and select the volume you want from the Volume drop-down menu.
• Click Attach.
You can connect to the volume once the volume's icon no longer lists it as Attaching.

To Connect to the Block Volume:
The Console provides the commands required to configure, authenticate, and log on to iSCSI.
Managing Block Storage Volumes

- You use the iSCSI protocol to connect to and configure the block volume.
- After you configure the volume, you can mount and use it like a normal hard drive.
- When you attach a block volume to an instance, the console provides the volume information. Click the Actions icon (…) on your volume's row, and then click iSCSI Commands and Information.
- You can use that information to configure and mount the volume to the instance.

You use the iSCSI protocol to attach a volume to an instance. Once the volume is attached, you log on to the instance and use the iscsiadm command-line tool to configure the iSCSI connection. After you configure the volume, you can mount it and use it like a normal hard drive.
Backup and Restoration

- You can take point-in-time complete image backups 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.

**To take a backup:**
- In the console, click Storage.
- Click Backups.
- Click the block volume for which you want to create a backup.
- Click Create Backup.
- Enter a name for the backup, and then click Create Backup.

The backup will be completed once its icon no longer lists it as CREATING in the volume list.

**To restore a new volume from a backup:**
- In the Console, click Storage, and then click Backups.
  - A list of the block volumes in the compartment you’re viewing is displayed. If you don’t see the one you’re looking for, make sure you’re viewing the correct compartment.
- Select the block volume backup you want to restore.
- Click Create Block Volume.
- Enter a name for the block volume and choose the Availability Domain in which you want to restore it.
- Click Create.

The volume will be ready to attach once its icon no longer lists it as PROVISIONING in the volume list.
About Mount Points

The process of associating a storage volume with an operating system is called **mounting**.

A **mount point** is the place in the current system’s directory hierarchy where the storage volume and its file system will be attached. The mount point is always a normal directory. The mount point doesn’t have to be created directly at the root (/); it can be created anywhere in the hierarchy of the system.

**The `/etc/fstab` File System Table**

Linux systems maintain a list of file systems and options in the `/etc/fstab` file. This is a plain text file. To ensure that your storage volumes are mounted at boot time, add the mount point details as an entry in the `/etc/fstab` file.
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.
Performance Benchmark

- Create 1TB volume
- Attach to BM compute instance
- Run sample performance benchmarks per volume
- IOPS
  
  ```
  sudo fio --filename=/dev/sdb --direct=1 --rw=randwrite --bs=4k --ioengine=libaio --iodepth=64 --runtime=30 --numjobs=16 --time_based --group_reporting --name=client-max
  ```

- Throughput
  
  ```
  sudo fio --filename=/dev/sdb --direct=1 --rw=randwrite --bs=256k --ioengine=libaio --iodepth=64 --runtime=30 --numjobs=4 --time_based --group_reporting --name=client-max
  ```

- Latency
  
  ```
  sudo fio --filename=/dev/sdb --direct=1 --rw=randrw --bs=4k --ioengine=libaio --iodepth=1 --runtime=30 --numjobs=1 --time_based --group_reporting --name=client-max
  ```
Object storage is where data is handled as an object, also known as unstructured data. The main differences between object storage and traditional storage (also known as block storage), are listed as follows:

- Stored data contains customized metadata.
- Data is indexed, allowing for much faster search results.
- Data can be located by using pointers instead of finding its location based on tracks and sectors on the hard disk (that is, the standard file system that we have used for many years).

This type of storage is used as an essential part of cloud services, in data centers, and it is normally integrated with virtual machines.

Because object storage allows for additional attributes as part of the “bundle,” applications, programs and storage devices are able to better manipulate data.

Nearly any file type can be stored in the form of object storage. Some popular files include media files (images, videos, music, and photos), documents, PDFs, backups, archives, and so on.

Multiple users can access the data.

With Object Storage Service, you can safely and securely store or retrieve data directly from the Internet or from within the cloud platform. Object Storage Service is agnostic to data content type. It enables a variety of use cases and works equally well with them. The Object Storage Service is a regional service. It is not tied to any specific compute instance. You can access data from anywhere within or outside the context of the Oracle Cloud Infrastructure, as long as you have Internet connectivity and can access the Object Storage Service API endpoint.

(HDFS Connector: https://docs.us-phoenix-1.oraclecloud.com/Content/Object/Tasks/hadoopsupport.htm)
The Object Storage Service resources are:

- **Object**: Any type of data, regardless of content type, is stored as an object. The object is composed of the object itself and metadata about the object. Each object is stored in a bucket.

- **Bucket**: A logical container for storing objects. Buckets are created by users or systems as needed. A bucket is associated with a single compartment which, in turn, has policies that indicate what actions a user can perform on a bucket and all the objects in the bucket.

- **Namespace**: The logical entity that lets you own your personal bucket names. Bucket names need to be unique within the context of a namespace, but bucket names can be repeated across namespaces. Each tenant is associated with one default namespace (tenant name) that spans all compartments. Within a namespace, buckets and objects exist in flat hierarchy, but you can simulate directories to help navigate a large set of objects (for example, guitars/fender/stratocaster.jpg, guitars/gibson/lespaul.jpg).

- **Compartment**: Compartments help you organize resources to make it easier to control access to them. A bucket can only exist in one compartment.
A bucket is a user-created resource, which can hold an unlimited number of objects.

When using this form of storage, data is treated as an object. Think of an object as a document file. Users can add additional attributes to each object such as: notes about the file, location where the file was created, compatibility options, and so on.

Traditional data storage (block storage) does not support additional metadata and attributes. Additionally, the file location must be specified by the user; this way, the operating system calls up that file from the hard drive directly.

Object storage allows for searchable metadata, automatic indexing, multiple copies/backups of stored data, and the ability to access storage nodes found in different parts of the world.

- If the storage container is about to reach its capacity limit, a new storage node is created to allow the user to continue adding data.

A common analogy to better understand object storage is valet parking:

- Even though you do not know where the car is parked—or if it has been relocated multiple times while you were away—when you are ready to leave, your ticket number is used to trace your car and return it to you. The car is the object; the ticket number is the object’s unique identifying number that provides the location of the car; and the valet’s parking lot is the container where the vehicles are parked in a flat area.
Following are some salient features of object storage:

- **Strong Consistency**: When a read request is made, the Object Storage Service always serves the most recent copy of the data that was written to the system. The Object Storage Service also offers a high-performing, high-bandwidth network.

- **Durability**: Data is stored redundantly across multiple storage servers across multiple Availability Domains. Data integrity is actively monitored using checksums and corrupt data is detected and auto repaired. Any loss of data redundancy is actively managed by recreating a copy of the data from the redundant copy.

- **Performance**: The Compute Service and the Object Storage Service are co-located on the same network. This means that instances running on the Compute Service can expect very high, non-blocking network bandwidth to the object store.

- **Custom Metadata**: You can define your own extensive metadata as key-value pairs for any purpose. For example, you can create descriptive tags for objects, retrieve those tags, and sort through the data.

- **Hadoop Support**: You can use the Object Storage Service as the primary data repository for big data. The HDFS connector provides connectivity to various big data analytic engines. This connectivity enables the analytics engines to work directly with data stored in the Object Storage Service.

- **Encryption**: The Object Storage Service employs 256-bit Advanced Encryption Standard (AES-256) to encrypt object data on the server. Each object is encrypted with its own key and object keys are encrypted with a master encryption key that is frequently rotated. Encryption is enabled by default and cannot be turned off.
Managing Buckets and Objects

- A bucket is a container for storing objects in a compartment within a namespace.
- In the console, access a compartment. Then navigate to Storage > Object Storage and click Create bucket. Enter a name and click Create.
- 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.

A bucket is associated with a single compartment. The compartment has policies that indicate what actions a user can perform on a bucket and all the objects in the bucket.

An object is a file or unstructured data such as: multimedia files, data backups, static web content, or logs that you upload to a bucket within a compartment within a namespace. Objects are processed as a single entity. You can't edit or append data to an object, but you can replace the entire object.

Note: In this course, while you can create a bucket and upload data as objects, we will not use object storage resources in the hands-on labs and practices.
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 REST API calls or the Java Software Development Kit (SDK) to manage multipart uploads, but not the Console.

With multipart uploads, individual parts of an object can be uploaded in parallel to reduce the amount of time you spend uploading. Multipart uploads can also minimize the impact of network failures by letting you retry a failed part upload instead of requiring you to retry an entire object upload. Oracle recommends that you perform a multipart upload to upload objects larger than 100 MB. The maximum size for an uploaded object is 10 TB. Object parts must be no larger than 50 GB. For very large uploads, a multipart upload also offers you the flexibility of pausing and resuming at your own pace.

A multipart upload consists of the following steps:
- Initiating an upload
- Uploading object parts
- Committing the upload

In the initiating step, you should create the parts to upload. The Object Storage Service provides API operations for the remaining steps. The service also provides API operations for listing in-progress multipart uploads, listing the object parts in an in-progress multipart upload, and aborting in-progress multipart uploads.

Creating Object Parts

With multipart upload, you split the object you want to upload into individual parts. Individual parts can be as large as 50 GB or as small as 10 MB. (The Object Storage Service waives the minimum part size restriction for the last uploaded part.) Decide what part number you want to use for each part. Part numbers can range from 1 to 10,000. You do not need to assign contiguous numbers, but the Object Storage Service will construct the object by ordering part numbers in ascending order.
Initiating an Upload: After you finish creating object parts, initiate a multipart upload by making a CreateMultipartUpload REST API call. Provide the object name and any object metadata. The Object Storage Service responds with a unique upload ID that you must include in any requests related to this multipart upload. The Object Storage Service also marks the upload as active. The upload remains active until you explicitly commit it or abort it.

Uploading Object Parts: Make an UploadPart request for each object part upload. In the request parameters, provide the namespace, bucket name, upload ID, and part number. In the request body, include the object part. Object parts can be uploaded in parallel and in any order. When you commit the upload, the Object Storage Service uses the part numbers to sequence object parts. Part numbers do not have to be contiguous. If multiple object parts are uploaded using the same upload ID and part number, the last upload overwrites the part and is committed when you call the CommitMultipartUpload API.

- The Object Storage Service returns an ETag value for each part uploaded. You need both the part number and corresponding ETag value for each part when you commit the upload.
- In the event of network issues, you can restart a failed upload for an individual part. You do not need to restart the entire upload. If, for some reason, you cannot perform an upload all at once, multipart upload lets you continue uploading parts at your own pace. While a multipart upload is still active, you can keep adding parts as long as the total number is less than 10,000.
- You can keep track of an active multipart upload by listing all parts that have been uploaded. (You cannot list information for an individual object part in an active multipart upload.) The ListMultipartUploadParts operation requires the namespace, bucket name, and upload ID. The Object Storage Service will respond with information about the parts associated with the specified upload ID. Parts information includes the part number, ETag value, MD5 hash, and part size (in bytes).

Committing the Upload: When you have uploaded all object parts, complete the multipart upload by committing it. Use the CommitMultipartUpload request parameters to specify the namespace, bucket name, and upload ID. Include the part number and corresponding ETag value for each part in the body of the request. When you commit the upload, the Object Storage Service constructs the object from its constituent parts. The object is stored in the specified bucket and namespace. You can treat it like you would any other object. Garbage collection will release storage space occupied by any part numbers you uploaded, but did not include in the CommitMultipartUpload request.

- You cannot list or retrieve parts from a completed upload. You cannot append or remove parts from the completed upload either. If you want, you can replace the object by initiating a new upload.
- If you decide to abort a multipart upload instead of committing it, wait for in-progress part uploads to complete and then use the AbortMultipartUpload operation. If you abort an upload while part uploads are still in progress anyway, the Object Storage Service cleans up both completed and in-progress parts. Upload IDs from aborted multipart uploads cannot be reused.
Summary

In this lesson, you should have learned how to:

- Create, attach, configure, and mount block volumes
- Back up and restore block volumes
- Detach and delete block volumes
- Describe concepts and uses of object storage
- Create object storage
- Upload objects to object storage
- Upload multipart objects to object storage
Practice 5: Storage Management

In this practice, each participant uses their assigned compartment and creates a block volume of 256 GB.
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
You want a single entry point to your application cluster. Load Balancing Service, a part of Oracle Cloud Infrastructure, offers you an IP-based load balancer that is highly available across availability domains within a region. The Load Balancing Service is primarily a regional service and offers a public IPv4 address within your VCN.

The service provides a load balancer with a public IP address, provisioned bandwidth, and high availability. Load Balancing Service provisions the public IP address across two subnets within your VCN to ensure accessibility even during an Availability Domain outage. You can configure multiple listeners for the IP address to load balance transport Layer 4 and Layer 7 (TCP and HTTP) traffic.
Public/Private LB

- **Public Load Balancer**
  - Requires 2 subnets, each in a separate AD
  - subnet1 – primary LB; subnet2 – stand-by LB for high availability in case of an AD outage
  - Public IP attached to subnet1; LB and IP switch to subnet2 in case of an outage
  - Service treats the two LB subnets as equivalent and you cannot denote one as "primary"

- **Private Load Balancer**
  - Private IP address that serves as the entry point for incoming traffic
  - Requires only 1 subnet – local to AD; no HA in case of any AD outage
• Backend Server – application server responsible for generating content in reply to the incoming TCP or HTTP traffic
• Backend Set – logical entity defined by a list of backend servers, a load balancing policy, and a health check policy
• Health Checks – a test to confirm the availability of backend servers; supports TCP & HTTP health checks
• Listener – an entity that checks for incoming traffic on the load balancer's IP address
• Load Balancing Policy – tells the load balancer how to distribute incoming traffic to the backend servers (round-robin, IP hash, least connection)
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:

<table>
<thead>
<tr>
<th>100 Mbps</th>
<th>400 Mbps</th>
<th>8000 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process 100 Mbps total bandwidth when multiple clients connected</td>
<td>Process 400 Mbps total bandwidth when multiple clients connected</td>
<td>Process 8000 Mbps total bandwidth when multiple clients connected</td>
</tr>
<tr>
<td>Key characteristics: Up to 1K SSL handshakes per sec with cipher (ECDHE-RSA2K)</td>
<td>Key characteristics: Up to 4K SSL handshakes per sec with cipher (ECDHE-RSA2K)</td>
<td>Key characteristics: Up to 40K SSL handshakes per sec with cipher (ECDHE-RSA2K)</td>
</tr>
</tbody>
</table>

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
Public Load Balancer example configuration

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

• Create a public load balancer
• Create a backend set with health check
• Add backend servers to your backend set
• Create a listener
• 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
Summary

In this lesson, you should have learned how to:

• Describe Oracle Cloud Infrastructure Load Balancers concepts
• Create and test a Public Load balancer
Practice 6: Implementing Public Load Balancer and High Availability

In this practice, each participant:

- Creates a new subnet in another Availability Domain
- Launches another instance using the custom image
- Creates a Public Load Balancer
- Configures security rules
- Configures the two Instances as backend servers
- Verifies access through Load Balancer, and checks high availability
Objectives

After completing this lesson, you should be able to:

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

- Oracle Cloud Infrastructure - Database Service provides Oracle Database systems in the cloud.
- Database system has these features:
  - Bare Metal compute instance for high performance
  - 10 gigabit network connection
  - Local NVMe storage is two-way and three-way mirrored for redundancy
  - Oracle Transparent Data Encryption is enabled by default
- You can increase or decrease your licensed cores dynamically as per your requirement.
- The two types of Database Systems offered by Oracle Cloud Infrastructure are:
  - Bare Metal DB Systems
  - Exadata DB Systems

The Oracle Cloud Infrastructure - Database Service lets you quickly launch an Oracle Database System (DB System) and create one or more databases on it. You have full access to the features and operations available with Oracle Database, but Oracle owns and manages the infrastructure. The Database Service supports several types of DB Systems, ranging in size, price, and performance.

Customers control and manage software that directly affects their application

- Database, OS, Clusterware

Oracle manages underlying infrastructure

- Facilities, servers, storage, storage software, networking, firmware, hypervisor, etc.

Customers have administrator privileges for compute VMs and databases so they can configure and run the system as they like

- Customers initiate automated database update script when it is convenient for them
- Can be run rolling across nodes to avoid database down time
Use Cases

- Mission Critical Production Databases
  - Very large databases (VLDB)
  - Database consolidation
  - OLTP, Data Warehousing, Analytics, Reporting
  - AppsU (EBS, JDE, PSFT)
- Test, Development, Certification, Try before Buy
- Disaster Recovery
An Exadata DB System consists of a quarter rack, half rack, or full rack of compute nodes and storage servers, connected by a high-speed, low-latency InfiniBand network and intelligent Exadata software. You can configure automatic backups, optimize for different workloads, and scale up the system to meet increased demands.
Exadata DB Systems are offered in quarter rack, half rack or full rack configurations, and each configuration consists of compute nodes and storage servers. The compute nodes are each configured with a Virtual Machine (VM). You have root privilege for the compute node VMs, so you can load and run additional software on them. However, you 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 by using Oracle Net Services from outside the Oracle Bare Metal Cloud Services. You are responsible for database administration tasks such as creating tablespaces and managing database users. You can also customize the default automated maintenance set up, and you control the recovery process in the event of a database failure.

Exadata System Configuration

The following table outlines the system resources based on your choice of configuration:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Quarter Rack</th>
<th>Half Rack</th>
<th>Full Rack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Compute Nodes</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Total Minimum (Default) Number of Enabled CPU Cores</td>
<td>22</td>
<td>44</td>
<td>88</td>
</tr>
<tr>
<td>Total Maximum Number of Enabled CPU Cores</td>
<td>84</td>
<td>168</td>
<td>336</td>
</tr>
<tr>
<td>Total RAM Capacity</td>
<td>1440 GB</td>
<td>2880 GB</td>
<td>5760 GB</td>
</tr>
<tr>
<td>Number of Exadata Storage Servers</td>
<td>3</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>Total Raw Flash Storage Capacity</td>
<td>38.4 TB</td>
<td>76.8 TB</td>
<td>153.6 TB</td>
</tr>
<tr>
<td>Total Raw Disk Storage Capacity</td>
<td>288 TB</td>
<td>576 TB</td>
<td>1152 TB</td>
</tr>
<tr>
<td>Total Usable Storage Capacity</td>
<td>84 TB</td>
<td>168 TB</td>
<td>336 TB</td>
</tr>
</tbody>
</table>
What’s New with Exadata DB Systems?

- Identity and Access Management (IAM) service
- Secure access with tenants, compartments, and resources
- Database backup service to IaaS object store
- Network Infrastructure improvements
- Virtual Cloud Network (VCN) use cases

• Network service components
  - Virtual Cloud Network (VCN) with private subnets and Availability Domain
  - Internet Gateway - routing between VCN and Internet
  - Dynamic Routing Gateway – private routing between VCN and on-premise network
• New connection use cases
  - Public subnets, private subnets with VPN, public and private subnets with VPN
• New hardware
  - Faster network, faster servers
• New tooling
  - New backup service
  - IaaS object store
Data is stored redundantly across multiple storage servers across multiple Availability Domains. Data integrity is actively monitored using checksums and corrupt data is automatically repaired. Any loss of data redundancy is actively managed by recreating a copy of the data from the redundant copy.

What you need

- VCN with Internet Gateway
- Object Storage Services bucket (additional purchase)
- Swift password
- User with tenancy-level access to object storage

Special notes

- Backup traffic is contained in VCN (no Internet traffic)

Backup to Object Storage doc: https://docs.us-phoenix-1.oraclecloud.com/Content/Database/Tasks/backingupOS.htm
Scaling within: You can scale up the number of enabled CPU cores in the system if an Exadata DB System requires more compute node processing power. For a non-metered Exadata DB System, you can temporarily modify the compute node processing power (bursting) or add compute node processing power on a more permanent basis. For a metered Exadata DB System, you can simply modify the number of enabled CPU cores.

Scaling across: Exadata DB System configurations enables you to move to a different system configuration. This is useful when a database deployment requires:

- Processing power that is beyond the capacity of the current system configuration
- Storage capacity that is beyond the capacity of the current system configuration
- A performance boost that can be delivered by increasing the number of available compute nodes
- A performance boost that can be delivered by increasing the number of available Exadata Storage Servers

Scaling from a quarter rack to a half rack, or from a half rack to a full rack, requires that the data associated with your database deployment is backed up and restored on a different Exadata DB System, which requires planning and coordination between you and Oracle.
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 server 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

The available shapes for Bare Metal Database Systems are:

- **BM.HighIO1.36:**
  - One-node DB System with one Bare Metal server, up to 36 CPU cores, 512 GB memory, and four 3.2 TB locally attached NVMe drives (12.8 TB total)

- **BM.DenseIO1.36:**
  - One-node DB System with one Bare Metal server, 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:**
  - Two-node RAC DB System with two Bare Metal servers, up to 36 CPU cores on each node (72 total per cluster), 512 GB memory, and direct attached shared storage with twenty 1.2 TB SSD drives (24 TB total)

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).
The shape you choose for a DB System determines its total raw storage, but other options, like 2- or 3-way mirroring and the space allocated for data files, affect the amount of usable storage on the system.

---

### Storage

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

<table>
<thead>
<tr>
<th>Shape</th>
<th>Raw Storage</th>
<th>Usable Storage with Normal Redundancy (2-way Mirroring)</th>
<th>Usable Storage with High Redundancy (3-way Mirroring)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM.HighIO1.36</td>
<td>12.8 TB NVMe</td>
<td>DATA 3.5 TB RECO 740 GB</td>
<td>DATA 2.3 TB RECO 440 GB</td>
</tr>
<tr>
<td>BM.DenseIO1.36</td>
<td>28.8 TB NVMe</td>
<td>DATA 9.4 TB RECO 1.7 TB</td>
<td>DATA 5.4 TB RECO 1 TB</td>
</tr>
<tr>
<td>BM.RACLocalStorage1.72</td>
<td>24 TB SSD</td>
<td>DATA 8.6 TB RECO 1.6 TB</td>
<td>DATA 5.4 TB RECO 1 TB</td>
</tr>
</tbody>
</table>
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.

To launch a database system, open the Console, click Database, choose your Compartment, and then click Launch DB System. In the Launch DB System dialog enter or select the appropriate values and click Launch. While the task of launching a database is quite simple, you should plan your database implementations with your database architect.
Provision Exadata CS in the Bare Metal Cloud
To start, enter your User Name, Password and Tenant ID.
Then, click the Sign In button.
Before we can create our Exadata Cloud Service, we first need to set up our Virtual Cloud Network (VCN).

Start by clicking the Networking tab.
Then select Virtual Cloud Networks
Here on the Virtual Cloud Networks home page, we can create a new VCN for our Exadata Cloud Service to use.

Start by clicking the Create Virtual Cloud Network button.
Start by naming our VCN.

The Create Virtual Cloud Network modal is now shown.
Then add a CIDR block.
The VCN resolver allows you to add a single host name in your TNS entry file on the app servers. For example, a typical SCAN IP TNS entry would consist of 3 IP address in the file, say 10.0.0.1, 10.0.0.2 and 10.0.0.3. With the VCN Resolver, you can put a single hostname, say myexacs.bmcloud.com and that entry would round robin to the SCAN IPs automatically.

We have our choice of CIDR ranges to use. We can use 10.0, 172.16 or 192.168. Once we select our range, we can then add the selected range and the bits used for the network block we are referencing. This is indicated by the number after the 
.

In our page below, we have selected 172.16.0.0 as our range and 22 as the bit length. This would give us an IP range from 172.16.0.0 to 172.16.3.255, 1024 IP addresses we can use in this block for our VCN.

Leave this checked. The VCN Resolver for DNS allows hostname resolution at VCN level. When creating our ExaCS, we can set a hostname and reference that hostname on other compute or ExaCS nodes in our VCN.
Now create the VCN by clicking the Create Virtual Cloud Network button.
Our VCN is now created.

We now need to create two subnets for our Exadata Cloud Service: a data and a backup subnet. We click on our VCN to view the details and start this process.
On the VCN details page, we need to create the 2 subnets.

Start by clicking the Create Subnet button.
The Create Subnet modal is now open.

Start by giving the new subnet a name. Here we will call it data and use it for the client network of the ExaCS.
Next, we choose an Availability Domain.
Next, we choose an Availability Domain.
Now we create a CIDR block that is a subset of our VCN. Here we use a larger bit length to create a smaller range of IPs.
We are going to use the default route table.
As well as the default DHCP Options. You can use custom ones if we had previously created them.
When done, we can click the Create button.
We have now created our data subnet.
Using the same process as we did for the data subnet, we can create a backup subnet. We just use a slightly different IP range.

Now we can go create our Exadata Cloud Service.
Click the DB Systems menu item under the Database tab.
To start the Exadata Cloud Service provisioning and DB creation process, click the Launch DB System button.
Here is the Launch DB System Modal.

First we need to give the Exadata Cloud Service a name.
Next, we choose an Availability Domain. Be sure to choose the same AD as where we created the network.
Next, we choose an Availability Domain.
Here we pick the Exadata Cloud Service shape we want to provision.
We can also name the RAC cluster.
The CPU Core Count lets us choose the starting amount of OCPUs for this shape. It is automatically set to the minimum count for the shape chosen.
Here we add a public key to be used for securing the OS on the compute nodes.

Clicking on the Show Advanced Options exposes the following:
Here we set if we want to use local and cloud or cloud only backups for our ExaCS. Using 40%, we reserve 60% of the disk for backups and 40% for data, thus allowing local backups. Selecting 80%, we use 80% for data and 20% for backups, which would prevent local backups and enable cloud only backups.
### TOTAL NUMBER OF NODES

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU CORE COUNT</td>
<td>22</td>
</tr>
</tbody>
</table>

The number of CPU cores to enable on the DB System. Specify a multiple of 2, up to 84.

#### SSH PUBLIC KEY

```
AAAAB3NzaC1yc2EAADAiQABAAAbAQDK13y95xO2H23lmy92yN-1MD71246xK
```

#### DATA STORAGE PERCENTAGE

- **40%**
- High

High disk redundancy (3-way mirroring) is required for all Exadata shapes.

### Network Information

- **VIRTUAL CLOUD NETWORK**
- **SUBNET**
  - First select 'Availability Domain' and 'Virtual Cloud Network'.
- **BACKUP SUBNET**
  - First select 'Availability Domain' and 'Virtual Cloud Network'.

---

*Oracle Cloud Infrastructure Fundamentals 7 - 46*
Scrolling down the modal we have the Networking section.
For the Virtual Cloud Network, we can choose the one we created before we started the DB process.
We then choose the data subnet for the client subnet.
And the backup subnet we created for the backup subnet attribute.
Here we set the hostname for the DB service.
Next up is the Database Information details section.

Start by choosing either a new database or to create a database from a backup.
Here we give the database a name.
Next, choose a database version. We can pick from 11.2.0.4, 12.1.0.2 and 12.2.0.1.
If we pick 12.1 or 12.2, the Database name will be the CDB name and here we set the PDB name.
The following two fields are for the database password. The password must conform to the rules outlined below the fields.
The database workload section will allow us to choose the type of database we create, an OLTP or DSS database.
Depending on what we chose for the disk percentages, this section will allow us to choose a backup method for the database we are creating.
Opening the Advanced Options, we can choose the database character set and national character set.
Now, click the Launch DB System button to create this Exadata Cloud Service.
We see our newly created service provisioning in the database systems page.
And ready to use in a short while.

We can click on the database service name to see the details.
On the DB System details page we can see all instance level details, databases on the system as well as the networking details.
OCPU Bursting
Online Scale-up Through Compute Bursting

Grow/shrink compute capacity to meet peak or seasonal demands

- Dynamically add or reduce OCPUs as often as once an hour
- Hourly rates to lower costs – avoids the need to provision for peak
- Burst up to 2x the base number of OCPUs or max capacity (whichever is lower)
- GUI-based self-service
Click the Scale Up/Down button to enable bursting.
Exadata Cloud Service Management with EM
Enterprise Manager sees Exadata Cloud Service as a cluster.
Cluster databases on the Exadata Cloud Service

### Database Information

<table>
<thead>
<tr>
<th>Database Name</th>
<th>Percentage of Size</th>
<th>Percent of Capacity</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCSDK-ExclP</td>
<td>0.00%</td>
<td>0.00%</td>
<td>11.2.0.08</td>
</tr>
<tr>
<td>CRM-ExclP</td>
<td>0.00%</td>
<td>0.00%</td>
<td>10.1.0.06</td>
</tr>
<tr>
<td>ODS-ExclP</td>
<td>0.00%</td>
<td>0.00%</td>
<td>11.2.0.06</td>
</tr>
<tr>
<td>ORDS-ExclP</td>
<td>0.00%</td>
<td>0.00%</td>
<td>11.2.0.06</td>
</tr>
</tbody>
</table>

### Server Pools

<table>
<thead>
<tr>
<th>Name</th>
<th>Minimum Size</th>
<th>Maximum Size</th>
<th>Important Active Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>ODS</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>CRM</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>ODS</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>APP</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
Click the SALES database on the Exadata Cloud Service.
Enterprise Manager sees this database just as any other on-premises database.

We can even manage and monitor this database like an on-premises database, only difference is it’s on an Exadata Cloud Service.

Enterprise Manager even knows that this is a clustered database with 2 instances.
Click on a PDB to drill into its details.
Again, Enterprise Manager lets us manage and monitor all aspects of an Exadata Cloud Service database just like it was on-premises.
Availability of Advanced Database Features: Multitenant, In-Memory, etc.
Utilize the power of Exadata Cloud Service by loading large tables into memory with the database In-Memory feature of Database 12c. Enterprise Manager makes putting tables in-memory simple with a few clicks of the mouse.
Using the Schema menu select: Database Objects -> Tables
Click Edit button to put it into memory.
Use In-Memory tab to put tables into memory with a mouse click.
Just check this checkbox and the table is now in-memory.
Use Enterprise Manager’s In-Memory Central to manage and monitor in-memory tables.
Analyze In-Memory attributes such as the relative size and “hotness” of the various in-memory tables.
Security!! All Tablespaces Created Encrypted in Oracle Cloud
By default, all new tablespaces are encrypted in the Exadata Cloud Service.
To see this in action, use the Administration menu to select: Storage -> Tablespaces
Create a new tablespace called HR_DATA
Keep Encryption option unchecked
Tablespace created, click on name to bring up details.
Verify that even with the option unchecked, this tablespace has been created encrypted!
Summary

In this lesson, you should have learned how to:

- Describe the options of database systems available with Oracle Cloud Infrastructure
- Launch a one-node database system
Practice 7: Launching a Database System Instance

This practice is to be completed after the training. In this practice, each participant:

- Creates a new database instance
- Connects to the database machine using ssh
- Connects to the database using SQL Plus within the database machine
- Performs start and stop of the database