**CHAPTER 4**

**Compute Instances**

In this chapter, you will learn how to

•   Describe compute service components

•   Create and manage compute instances

•   Explain advanced compute concepts

All roads in cloud computing lead to the compute instances. These are the actual computers or servers that run system and application software. This is a huge driver for the existence of public clouds. Instead of managing your own equipment, you pay for just the equipment you need while it is managed by a cloud vendor in one or more data centers.

Compute instances are used for desktop and server class machines. Some organizations no longer provide desktop PCs or laptops to employees. Instead, they have opted to provide staff with dumb terminal access via web browsers to PC desktops that are actually compute instances running in a cloud. The flexibility of this offering is staggering. When staff leave, their VM access needs only to be terminated. If they require more computing resources, these VMs can be upgraded on-the-fly. Backups, upgrades, patching, and corporate software management of end-user PCs may be orchestrated through a common set of commands and processes, ensuring a secure and consistent desktop ecosystem.

This flexibility extends to larger compute instances often used as file, database, middleware, and directory servers. In fact, there is no distinction based on the size of the compute instances. There are just different shapes, either running as bare-metal or virtual compute instances on either a predefined or custom image. It is a very compelling proposition that you can right-size your compute footprint based on your actual requirements and scale as your organizational needs change, without needing to prophesy your requirements over the next five years in terms of both hardware infrastructure and human capital.

This chapter discusses shapes, images, and custom images before exploring the administrative tasks of creating and managing instances. The notions of BYOI (Bring Your Own Image) and BYOH (Bring Your Own Hypervisor) are described before the chapter closes with a discussion on boot volumes and fault domains.

**Compute Service Components**

The compute service fundamentally provides compute instances as a service, which means you interact with OCI through APIs or the console to provision a computing host or instance. OCI provides a hypervisor layer that accepts API calls from the console, CLI, and other SDKs. A hypervisor is the lowest-level operating system software that is installed on bare-metal servers. A compute instance runs on top of the hypervisor layer and interacts with the physical hardware through the hypervisor.

A compute instance resides in a single availability domain (AD). Instances are either virtual machines (VMs) or bare-metal machines (BMs). VMs and BMs reside on physical equipment localized in a data center or AD. A VM is defined as an independent computing environment executing on physical hardware. Multiple VMs may share the same physical hardware. BM instances execute on dedicated hardware, providing strong isolation and highest performance. In other words, the hypervisor on a particular x86 server may host multiple VMs or a single BM instance. Both machine types are available in many shapes and are based on x86 hardware, and so are capable of running a variety of Linux and Windows operating systems.

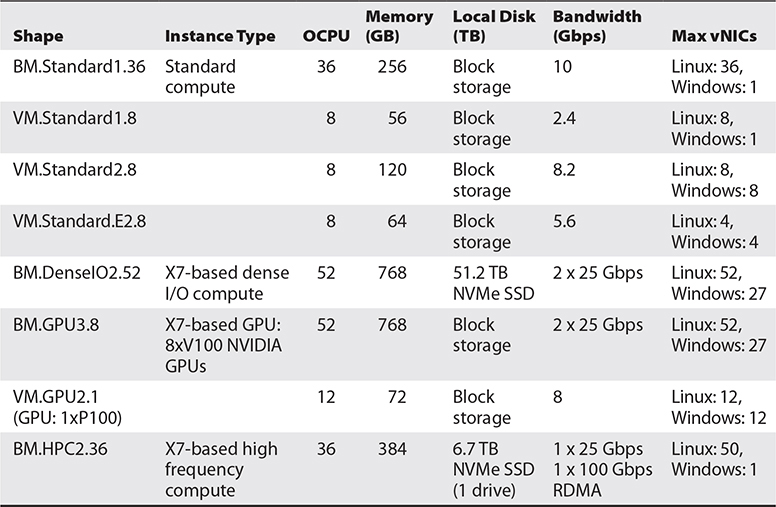
When a new compute instance is created, many options may be specified, including a name, the AD it resides in, the boot volume (which may be an Oracle-provided image such as Oracle Linux 7.7), and the shape.



**CAUTION**    Within OCI, when the word “instance” is used without additional description, it is almost always referring to compute instances and not Oracle database instances.

**Compute Shapes**

A compute shape is a predefined bundle of computing resources, primarily differentiated by Oracle Compute Units (OCPUs), memory, network interfaces, network bandwidth, and support for block and NVMe local storage. An OCPU is equivalent to a hyperthreaded core. Therefore, each OCPU corresponds to two hardware execution threads, known as vCPUs. [Table 4-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4tab1) lists a subset of available compute shapes across both VM and BM instance types. New compute shapes are added as new hardware is added to availability domains.



**Table 4-1** VM and BM Compute Shapes

Shape names contain several useful identifiers. VM and BM refer to the broad category of instance type. Standard means that only block storage is available, while DenseIO means local NVMe drivers are present to support NVMe SSD disk drives. GPU shapes are based on servers with NVIDIA GPUs (Graphical Processing Units) while HPC shapes offer supercomputer high performance compute power. The two numbers at the end of the shape name refer to the generation of the hardware and the number of OCPUs.

Consider the shapes VM.Standard1.8 and VM.Standard2.8, described in [Table 4-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4tab1). Both have eight OCPUs, but they are based on first- and second-generation hardware (in this case, X5 and X7). Other differences between these two VM shapes are the available memory (56GB vs 120GB), the network bandwidth supported (2.4 Gbps vs. 8.2 Gbps), and the support for multiple network interfaces when running Windows operating systems (1 vs. 8).



**CAUTION**    Not all shapes are available in all regions. For example, the first-generation shapes are only available to certain tenancies in the Phoenix, Ashburn, and Frankfurt regions. When you create a compute instance, there may already be service limits imposed on your account. These may be elevated but require a service request to be opened by an administrator through the OCI console.

Consider the shapes VM.Standard2.8 and VM.Standard.E2.8, described in [Table 4-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4tab1). These shapes are almost identically named, except for the additional “.E” in the latter shape. This identifies the underlying CPU as an AMD E-series microprocessor (EPYC CPU) as opposed to the standard Intel Xeon–based microchips found in other shapes. Other differences between the VM.Standard2.8 and VM.Standard.E2.8 shapes include the available memory (120GB vs. 64GB), the network bandwidth supported (8.2 Gbps vs. 5.6 Gbps), and the number of available network interfaces (8 vs. 4). Similarly, the GPU3 and GPU2 shapes are based on NVIDIA P100 and V100 chipsets respectively.



**NOTE**    The underlying physical server that hosts multiple VMs or a single BM instance is an Oracle bare-metal server like an X7-2 x86 machine you could purchase for your on-premises data center. VMs share the physical infrastructure, whereas BM instances have dedicated access to the CPU cores, memory, and network interfaces (NICs) on the physical hosts. A wide range of compute shapes is currently available, and the variety will continue to increase as demand grows. It is important to know your options to understand the costs and benefits of each shape. Unsurprisingly, bare-metal shapes cost more than their VM counterparts, and larger machines cost more than smaller machines. This will inform prudent decision-making that balances scalability and performance requirements with financial constraints.

**Compute Images**

When creating compute instances, a key decision is to determine the operating system image. You may choose from the following:

•   **Platform images**   Pre-built OCI-provided images with an operating system

•   **Oracle images**   Pre-built OCI images with applications pre-installed as well

•   **Partner images**   Trusted pre-built third-party images published by partners

•   **Custom images**   Images you have generated from other OCI instances or imported into OCI

•   **Boot volumes**   Previously created boot volumes

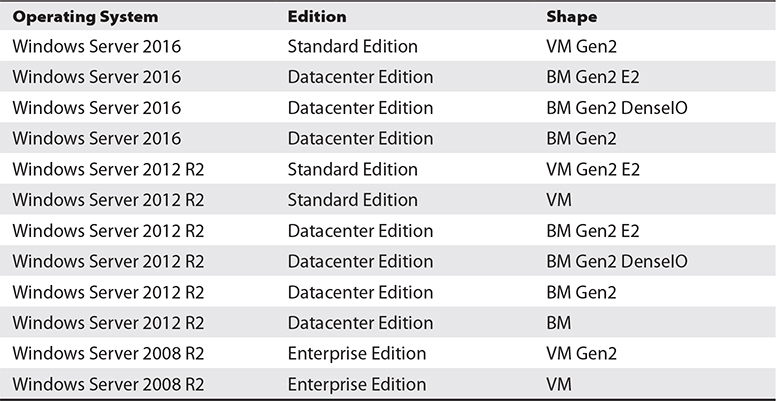
•   **Image OCID**   A specific version of an image (example: an image location provided from the OCI Marketplace)

•   **BYOH**   Bring Your Own Hypervisor

These image types are discussed next.

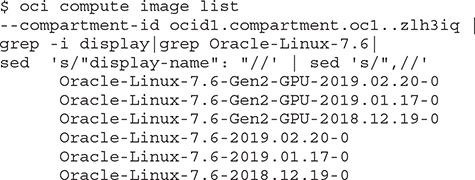
**Platform Images**

OCI offers several pre-built Linux and Windows images complete with the appropriate drivers to rapidly provision your instance. Available OCI preconfigured images include these operating systems in various shape-related editions. [Table 4-2](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4tab2) shows several images for several Windows Server images.

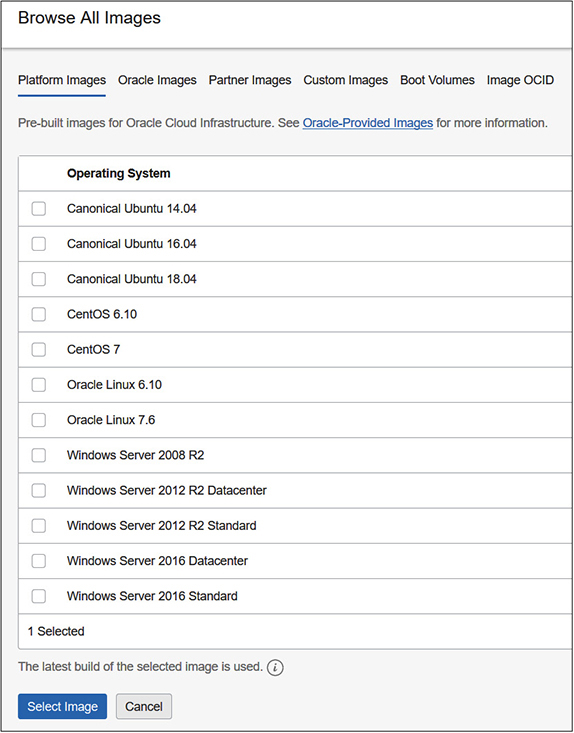


**Table 4-2** Several Preconfigured Windows Images and Supported Shape Type

Using the CLI, you may also list the current set of images:



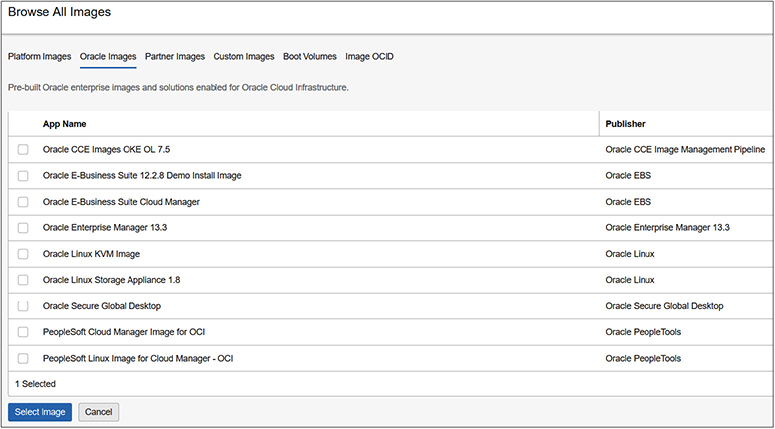
This list shows six builds of the Oracle Linux 7.6 image. The Gen2-GPU images are specific to the second-generation GPU shapes. When creating an instance, the console shows the latest builds for all preconfigured images. Other available builds are visible through the Advanced Options section discussed later in the chapter or through the CLI and API. [Figure 4-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig1) shows a typical list of platform images available through the console. This list evolves as newer versions are released and older versions are desupported.



**Figure 4-1**   A variety of pre-built platform images

**Oracle Images**

Oracle has published images with preinstalled applications known as Oracle images. [Figure 4-2](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig2) shows how applications, including the Oracle E-Business Suite Demo environment, Oracle Enterprise Manager, or PeopleSoft Cloud Manager, can be deployed by simply selecting an image from a list of prebuilt applications.



**Figure 4-2**   Enterprise images pre-built for OCI



**CAUTION**    Not all Oracle images are available in all regions. The repository of canned images is expanding and images are released to different regions at different times.

When you create a compute instance from an Oracle image, you have to accept some legal terms and conditions before the instance is provisioned.

**Partner Images**

OCI provides a cloud marketplace where third-party vendors proffer their application software to OCI users. Partner images are pre-built images that include an operating system and application deployment from a third-party provider. These images have been vetted by Oracle and are considered trusted images. Upon choosing a partner image on which to base your compute instance, you are required to accept legal terms and conditions from both Oracle and the third-party vendor that governs your use of the image.

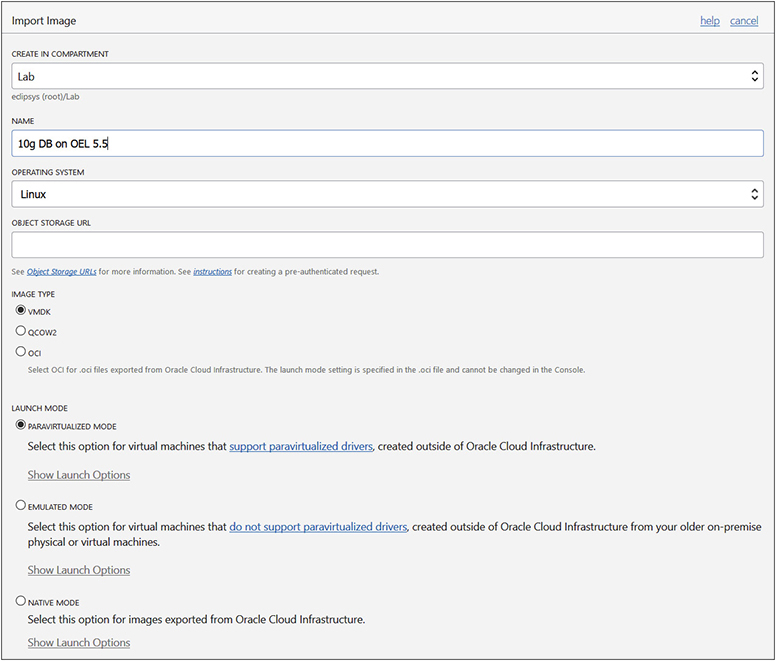
**Custom Images**

OCI provides an interface for you to create your own images from existing compute instances and save these as custom images to be used as the basis for future compute instance deployments. Custom images may be based on OCI platform, oracle, or partner images that have been customized in some way, or they may be imported from an external image that meets several requirements.

Some organizations create a master image, sometimes known as a gold image, that is rolled out to a specific user community. You can create a custom image, ensuring that all security, network, and antivirus settings are in place with appropriate patches. These could be desktop images preloaded with office productivity applications or pretty much any software.

Another situation that lends itself to custom images is related to autoscaling, discussed later in this chapter. Imagine a website hosted on a single compute instance running an HTTP server. You can create a custom image from this compute instance and use it as the basis for automatically provisioning additional preconfigured web server compute instances when your website is overloaded.

[Figure 4-3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig3) shows the OCI console interface used for importing pre-existing external images as custom images.



**Figure 4-3**   Import custom images for compute instances.

Existing virtual machines may be exported as OCI, VMDK, or QCOW2 format images. These exports are uploaded to object storage (discussed in [Chapter 5](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5)). For example, you may have a legacy Oracle 10*g* database on Oracle Enterprise Linux 5.5 hosting a production system on-premises in a virtualized environment. You could lift and shift this environment to OCI by exporting the virtualized image as a VMDK or QCOW2 format image type, uploading the exported image files to object storage, importing the image as a custom image in OCI, and creating a compute instance based on this image.

As [Figure 4-3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig3) shows, custom images must be launched in one of three modes:

•   **Native mode**   Drivers in the image communicate directly with underlying hypervisor.

•   **Paravirtualized mode**   The guest image is modified to hook directly to the underlying hypervisor for certain tasks.

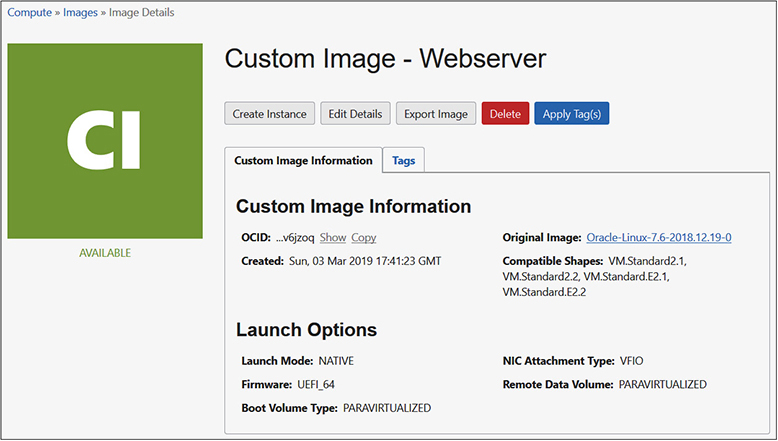
•   **Emulated mode**   The guest image is fully virtualized and runs without modification on the OCI hypervisor.

The launch mode is determined by the compatibility of the underlying image with the hardware hosting the virtual machines. Custom images imported from OCI format exports may be launched in native mode because these images already have system drivers for the underlying hardware. Images created outside of OCI may be launched in either emulated mode or paravirtualized mode, depending on whether the operating systems in these images have support for the underlying hardware. Older operating systems typically do not have drivers for modern hardware and are likely to launch in emulated mode only.



**NOTE**    The performance of virtual machines is related to its launch mode. If the compute shapes are the same, a VM launched in native mode performs better than one launched in paravirtualized (PV) mode. An instance launched in PV mode will perform better than one in emulated mode. It is therefore preferable to migrate older systems to newer natively supported images.

You may create a custom image from an existing compute instance. The compute instance is stopped (though a manual shutdown beforehand is usually advisable) for the duration of the image creation and started up automatically once the imaging is completed. [Figure 4-4](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig4) shows a custom image created from an existing instance. This image has an OCID and is compatible with several compute shapes. It may be launched in native mode and is based on an Oracle Linux 7.6 image.

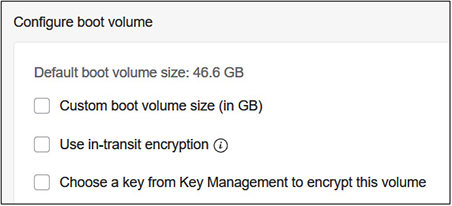


**Figure 4-4**   Custom image created from a running compute instance

**Boot Volumes**

When you create a compute instance, you may configure the boot volume for the instance. This is a special block volume that stores the operating system and boot loader required to launch the compute instance. [Figure 4-5](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig5) shows several optional boot volume configuration options.

The boot volume size may be increased to provide headroom for future growth of the boot volume. The default boot volume size depends on the image chosen. Linux images usually require a significantly smaller boot volume than Windows images. Access to boot volumes is provided to compute instances through the OCI Block Volume service. Storage blocks traverse a network fabric between the underlying storage and the compute instance. You may choose to encrypt the data in-transit. Boot volumes may also be encrypted at rest using predefined keys you have configured using the Key Management vault.



**Figure 4-5**   Boot volume sizing and encryption configuration options



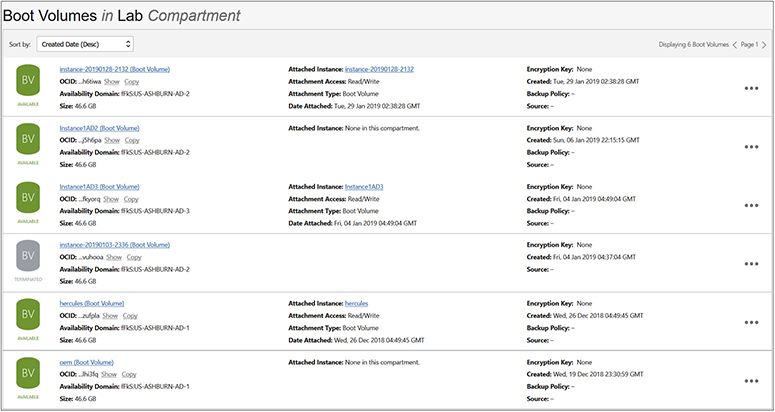
**NOTE**    Choosing a custom boot volume size increases the size of the block volume presented to the OS but not the size of the file systems built upon it. You need to manually extend the volume to take advantage of the larger size.

A boot volume is attached to the instance but may be detached and retained upon termination of the instance. It may also be cloned and backed up. A detached boot volume, which may be a clone or previously attached boot volume, may be used as the image for a new compute instance.



**EXAM TIP**    A boot volume used as the image source for a compute instance must be available in the same AD chosen to host the compute instance.

[Figure 4-6](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig6) lists six boot volumes in the Lab compartment. Three are attached to compute instances, one is terminated, and two are not attached. It is these detached volumes that are available to reuse as boot volumes in new compute instances. Notice that each boot volume has an OCID and resides physically in storage in a particular AD. Therefore, to create a new compute instance in AD1 in the Ashburn region (for this tenancy), only the OEM boot volume is available for reuse as a boot volume for compatible shapes.

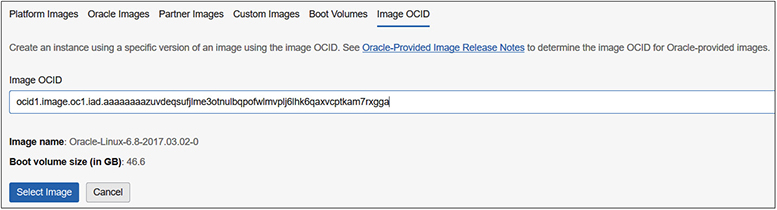


**Figure 4-6**   Attached and detached boot volumes

Apart from imaging compute instances, a boot volume clone may be taken when troubleshooting a problematic instance that cannot boot up or to recover data. A useful technique is to clone the boot volume of the problematic instance and attach the clone to another instance as another block volume (discussed in [Chapter 5](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5)). By mounting the cloned volume as a secondary volume, you can expose the file systems on the cloned volume for further investigation.

**Image OCID**

Custom and platform images are storage resources with OCIDs. You can create a compute instance by referencing an available image by its OCID. This image is then used to clone a new boot volume for your compute instance. [Figure 4-7](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig7) shows an older-platform image (Oracle Linux 6.8) that is no longer available through the platform images list, being referenced using its OCID.



**Figure 4-7**   Create a compute instance using an image OCID

The Oracle-Provided Image Release Notes link on the Image OCID page in the OCI console takes you to the online documentation where families of operating system images and their OCIDs are listed. Custom images may also be created using their image OCID.



**CAUTION**    An important distinction must be made between images and boot volumes. Platform, Oracle, Partner, and Custom images as well as images referenced by their OCIDs are templates that are cloned onto new boot volumes when a compute instance is created. Boot volumes and boot volume clones are actually attached to new compute instances. These may be cloned to instantiate a new boot volume to attach to another new compute instance.

**Bring Your Own Hypervisor**

Strictly speaking, BYOH follows the same principles as Oracle and custom images discussed earlier. These offer significant value so are covered discretely. Moving existing virtualized infrastructure becomes a simple lift and shift exercise because BYOH allows you to use a wide range of operating systems, including many legacy systems. OCI provides support for installing several hypervisors on bare-metal instances. Supported hypervisors include the following:

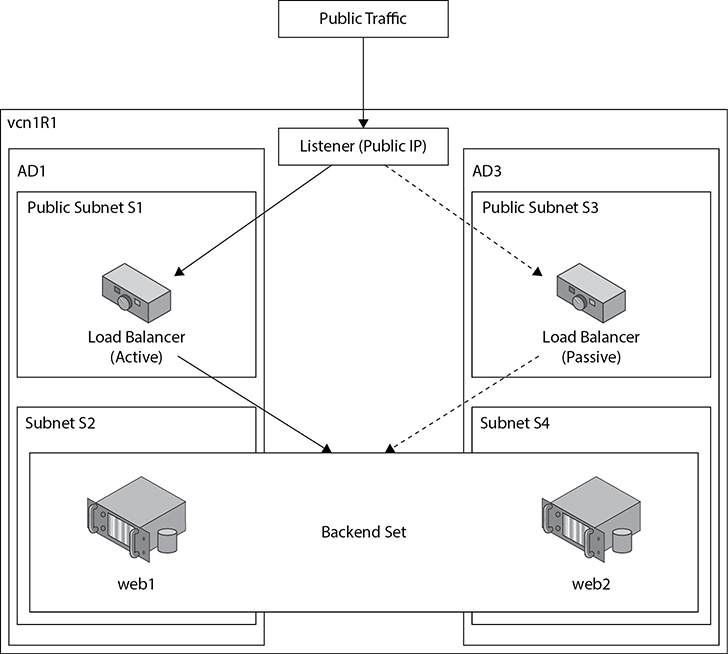
•   **Kernel-based VM (KVM)**   This entails imaging a bare-metal instance with an Oracle Linux image and installing the KVM server software. After configuring the networking on the bare-metal instance to leverage multiple vNICs and provisioning additional block storage, KVM guests may be imported from existing images or installed from ISO images. You may choose an Oracle Linux KVM Image from the Oracle image list in the OCI console.

•   **Oracle VM (OVM)**   You set up OVM Manager on a VM or BM instance and OVM Server on bare-metal instances using Image OCIDs provided by Oracle. OVM may be used as on-premises for hosting guest VMs. OVM manager behaves as a bastion host to the guest VMs.

•   **Hyper-V**   Older Windows operating system images may be deployed as guest VMs of Hyper-V deployed on bare-metal instances.

**Instance Management**

It is about time we get into the thick of things and create some compute instances. Prerequisites include having a VCN in place and deciding on an architecture. If you have worked through exercises in the earlier chapters, you should have several VCNs in place in your OCI tenancy and should be good to go. This section consists of several exercises geared to producing a solution based on the architecture in [Figure 4-8](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig8). The second part in this section discusses management of these compute instances.



**Figure 4-8**   Two compute instances serving as web servers as part of a backend set



**NOTE**    While the console is great for deploying relatively small solutions, it does not scale well to large solutions. Treating infrastructure as a service and deploying hundreds or thousands of compute instances, network, storage, and security infrastructure is only practical by leveraging automation tools such as Terraform, OCI CLI scripts, or the APIs through the SDKs. Automation tools are discussed in [Chapter 7](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch7.xhtml#ch7).

**Create Compute Instances**

This section consists of five exercises that entail

•   Creating an SSH key pair to use to connect to the Linux compute instances

•   Creating a compute instance to use as a web server to serve a simple Hello World static HTML page

•   Creating a custom image based on this compute instance and using this to spawn a new compute instance

•   Creating a public load balancer to round robin HTTP traffic to these web servers

•   Creating a Windows instance

To connect securely to your Linux compute instances, you need an SSH key pair. Basically, you generate a key pair consisting of a private key and a public key. You keep the private key safe and do not share or post publicly. You provide the public key at compute instance create time.

When OCI provisions a new Oracle Linux or CentOS compute instance, a user named opc is created. This is the initial user used to connect to the instance. The opc (Oracle Public Cloud) user has privileges to gain root access through the sudo command. The public key is added into the authorized\_keys file in a hidden subdirectory in the opc users home directory (/home/opc/.ssh). For Ubuntu Linux, the initial user created is named ubuntu.

When you try to connect to the public IP address of the instance using the SSH protocol on port 22 (open by default in the firewall), you provide the private key. Through a series of encrypted message exchanges between your SSH client and the compute instance, the client is authenticated and a connection is established.



**NOTE**    SSH key pairs may be created using a variety of encryption formats including RSA, DSS, and DSA and may be created using tools such a ssh-keygen from the UNIX OpenSSH package or the PuTTY Key Generator on Windows. Key pairs may be used multiple times, but this presents the same risk as setting the same root password on multiple systems. Whether the key pair is used only once or used in multiple situations, the private key must be kept secret. It is good practice to rotate these keys periodically.

**Exercise 4-1: Create an SSH Key Pair**

If you are using Windows, download a tool that is capable of generating SSH keys, such as PuTTY Key Generator, and follow Steps 1–6. If you are using a Unix-style system (including MacOS), ensure that a tool such as ssh-keygen is available and follow Step 7 or 8 in this exercise.

**1.**   Launch PuTTY Key Generator (sometimes labeled as PuTTYgen).

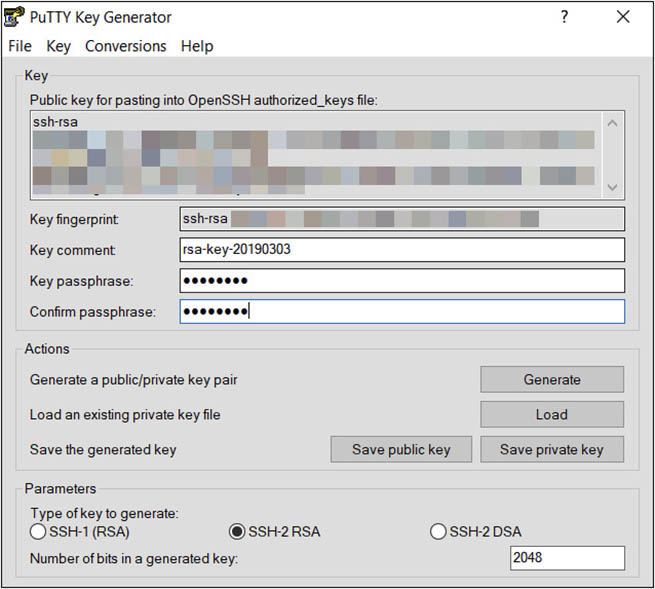
**2.**   Choose SSH2-RSA as the key type and specify 2048 as the number of bits in this key.

**3.**   Choose Generate and move your mouse cursor over the blank area to generate some random seed that will be used to generate the keys.

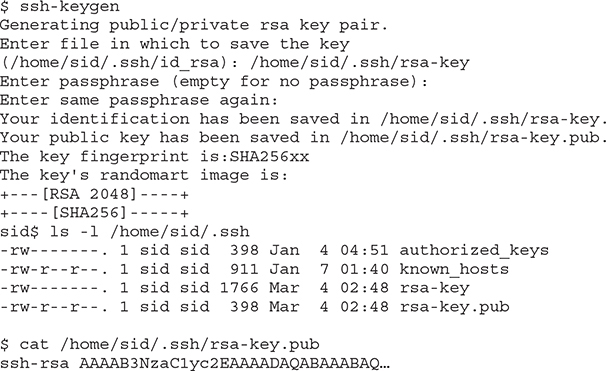
**4.**   Provide a key passphrase as an additional security measure.

**5.**   Copy the public key in OpenSSH authorized\_keys file format and paste it into a text file, typically saved with a .pub extension. In this exercise, the file with the public key is named rsa-key.pub.

**6.**   Choose Save Private Key and save the PuTTY format private key as rsa-key.ppk. It is usually advisable to export the private key via Conversions | Export OpenSSH key into RSA format if you intend to use the same key pair on a Unix-style computer (including MacOS). Alternatively, the PuTTY format private key can later be converted using PuTTYgen on Windows or Linux via the putty-tools package.



**7.**   You can run ssh-keygen in interactive mode or through a single command. In interactive mode, simply run ssh-keygen at the Unix prompt. You will be prompted to provide a filename (/home/sid/.ssh/rsa-key) and a passphrase.



**8.**   To run ssh-keygen as a single command, you are required to specify several parameters including -t <key type>, -N <passphrase>, -b <number of bits in key>, -C <key name>, and -f <filename>. For example:



**9.**   You have successfully generated your SSH key pair. If you specified a passphrase, keep this safe as it will be required in the next exercise when connecting to your compute instance.

**Exercise 4-2: Create a Compute Instance to Use as a Web Server**

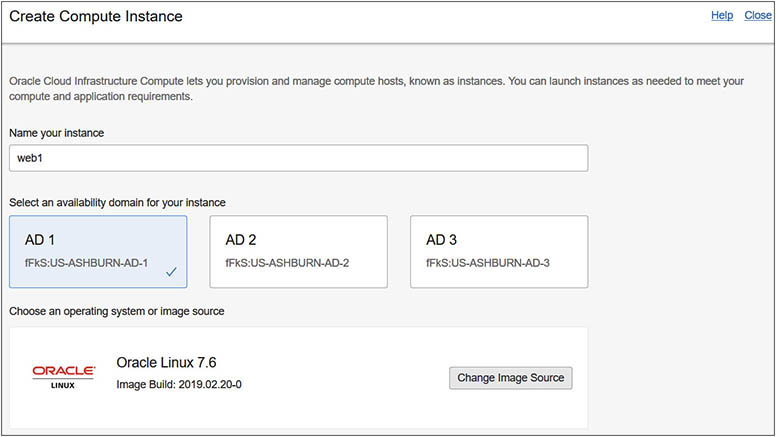
In this exercise, you will create a compute instance in your home region in a VCN with at least two subnets. If you completed all the exercises in [Chapter 3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch3.xhtml#ch3), you should have a VCN similar to VCN1R1. A security ingress rule allowing TCP traffic in through port 80 must be added to the relevant security list used by the public subnet you plan to use for your web server compute instance.

**1.**   Sign in to the OCI console and choose your compartment.

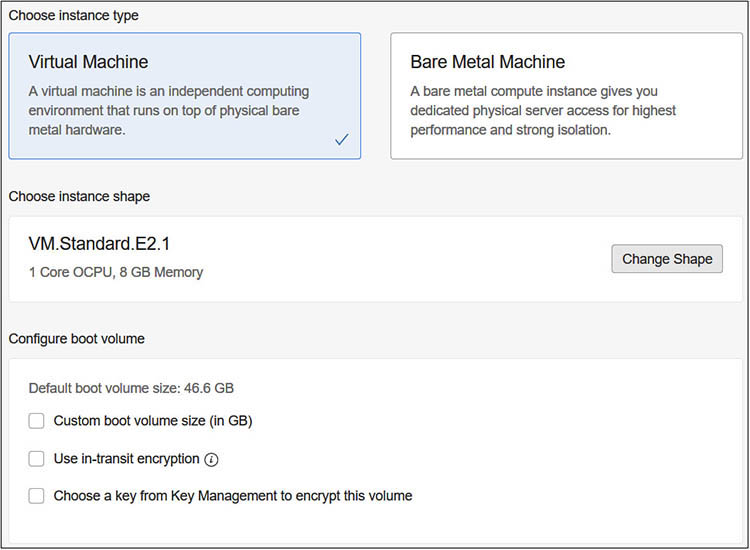
**2.**   Navigate to Compute | Instances and choose Create Instance.

**3.**   Name your instance (for example, web1) and select an AD for your instance (for example, AD1).

**4.**   For operating system or image source, choose Change Image Source, navigate to Platform Images, and select an operating system (for example, Oracle Linux 7.*x*).

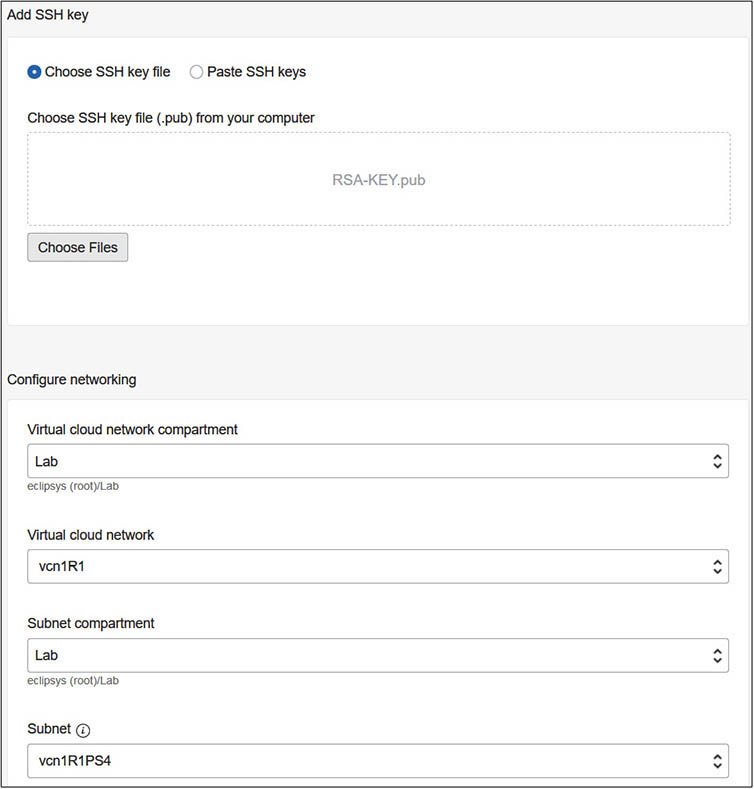


**5.**   Select a virtual machine instance type, click Change Shape, and choose a shape (for example, VM.StandardE2.1). Change any boot volume configuration options as required or accept the default settings.



**6.**   Add the public SSH key file you generated in the previous exercise. You may choose the key file or paste the SSH key from the public key file. You may add multiple public keys to the compute instance at this stage if required by choosing to paste multiple public SSH keys.

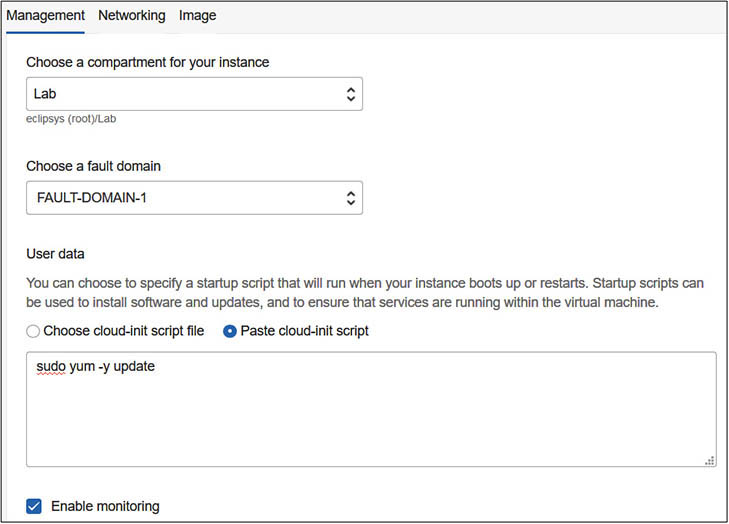
**7.**   Configure networking for your instance by choosing the compartment, VCN, subnet compartment, and public subnet.



**8.**   Click Show Advanced Options. Ensure that your compartment is correct. Choose a fault domain (FAULT-DOMAIN-1). Fault domains are sets of fault-tolerant isolated physical infrastructure within an AD. By choosing different fault domains for two VM instances, you ensure these are hosted on separate physical hardware, thus increasing your intra-availability domain resilience.

**9.**   Insert the following commands into the User data cloud-init script area. This script will run when the newly provisioned compute instance boots up. The following script updates the package repository.

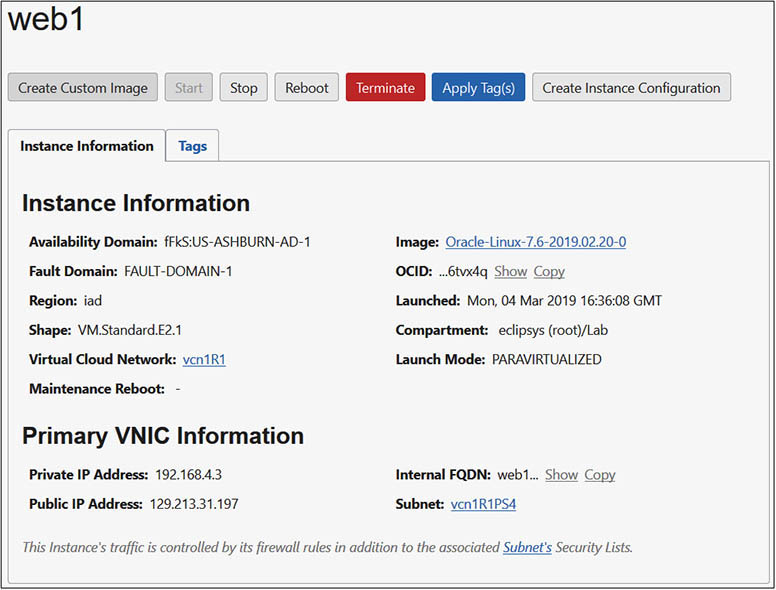
sudo yum -y update



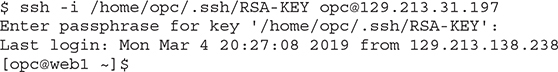
**10.**   Choose the Networking menu in the Advanced options and confirm the VCN and subnet settings. Additionally, check that the Assign Public IP Address option is checked. Finally, enter a hostname, for example, web1 and click Create.



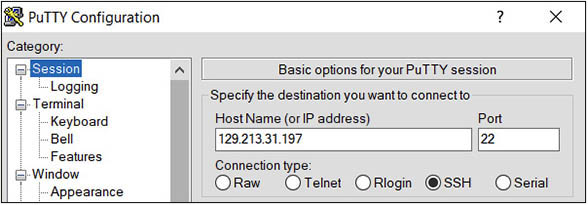
**11.**   After a few minutes, the status of your new compute instance will change from PROVISIONING to RUNNING. RUNNING does not mean the instance has fully booted up. You may have to wait a few minutes for the instance to boot up and run the cloud-init startup script.



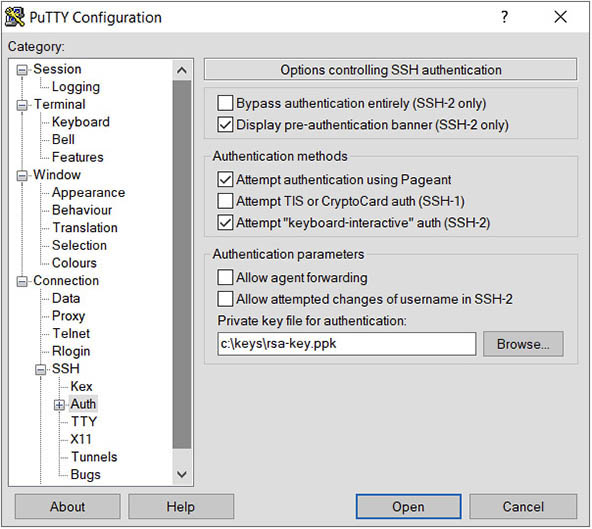
**12.**   You may connect to the compute instance using ssh and the private key, which matches the public key you provided when provisioning the instance. If you are using a Unix-style system, ensure that the ssh program is available and connect to the instance as follows: ssh -i <*path to private key file*> opc@<*public IP address of compute instance*>. For example:



**13.**   If you are using a Windows system, launch a terminal emulator such as PuTTY, provide the public IP address in the Session category, and choose SSH as the connection type.



**14.**   In PuTTY, navigate to the Connection | Data section and provide opc as the auto-login username. Navigate to the Connection | SSH | Auth section, and browse to the private key file created earlier.



**15.**   Navigate to the Session category and provide a name for this connection, such as web1, and choose Save. Choose Open to connect to the instance.

**16.**   You should now be at the linux prompt connected as the opc user to your new web1 instance. Ensure that you can connect to the Internet, specifically the [yum.oracle.com](http://yum.oracle.com/) server. If you have difficulties with network connectivity, ensure that an Internet gateway is attached to your VCN and that there are no security list or route table rules preventing the connection.

**17.**   The following commands install the Apache web server (httpd), add a firewall rule to allow incoming traffic on TCP port 80, reload the firewall daemon, create a simple index.html file with the message Hello World from <*hostname*>, and start up the Apache HTTP daemon.



**18.**   In a browser, navigate to your website by using the URL http://<*public IP address of web1*> and you should see the Hello World from: web1 message.

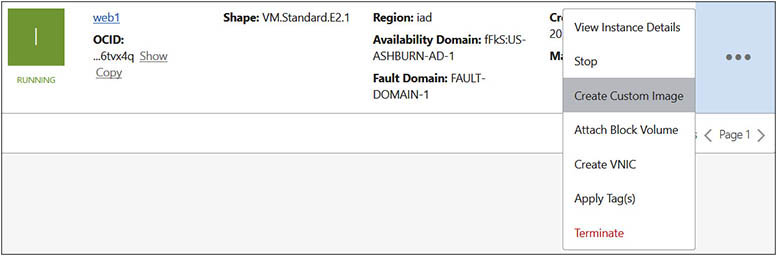


**Exercise 4-3: Create and Use a Custom Image**

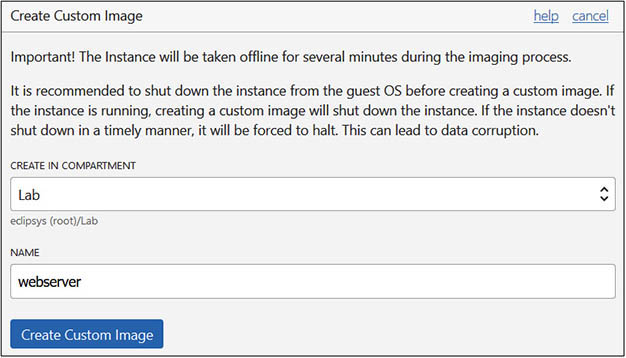
In this exercise, you will create a custom image from the web1 compute instance created in the previous exercise. You then create a new compute instance named web2 using the custom image.

**1.**   Sign in to the OCI console and choose your compartment.

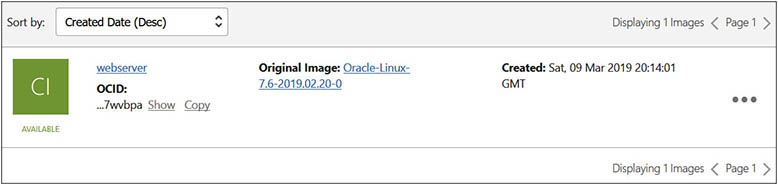
**2.**   Navigate to Compute | Instances and hover over the ellipsis adjacent to the web1 compute instance, and choose Create Custom Image. An alternative navigation path is to choose the Create Custom Image button on the Instance Details page.



**3.**   Name your custom image (for example, webserver), select a compartment for this resource, and choose Create Custom Image. The compute instance is offline during image creation and its status changes to Creating Image. It is good practice to properly shut down the instance before using it to create a custom image. This ensures data consistency when it is used to build future instances. If you do not stop the instance prior to creating a custom image, OCI shuts it down cleanly or forcibly if necessary, risking data corruption, before creating the custom image and rebooting the source instance.



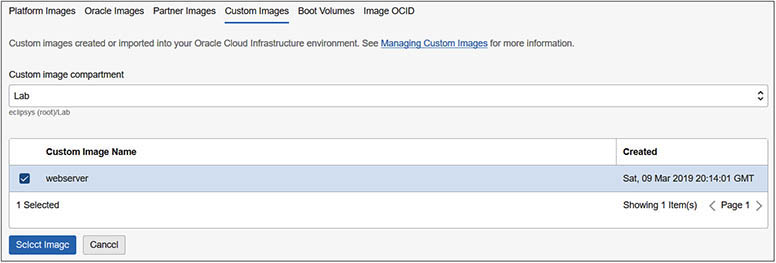
**4.**   Once the custom image is created, navigate to Compute | Custom Images to see the list of custom images including the web server image you have just created.



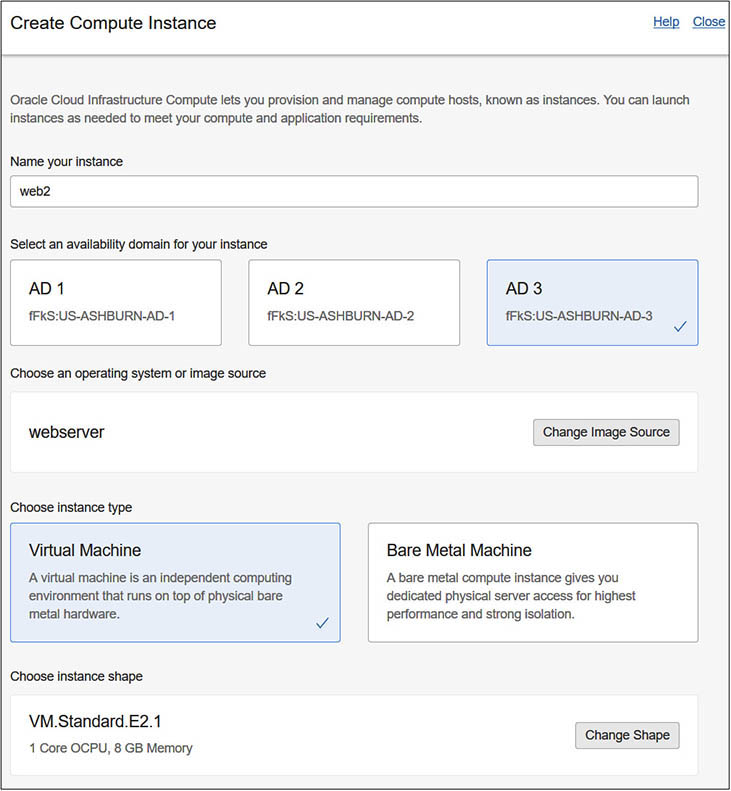
**5.**   Before using this custom image to create a new instance, ensure that your VCN has two separate public subnets.

**6.**   Navigate to Compute | Instances and choose Create Instance. Name your instance (for example, web2) and select an AD for your instance (for example, AD3).

**7.**   For operating system or image source, choose Change Image Source, navigate to Custom Images, select webserver, and choose Select Image.

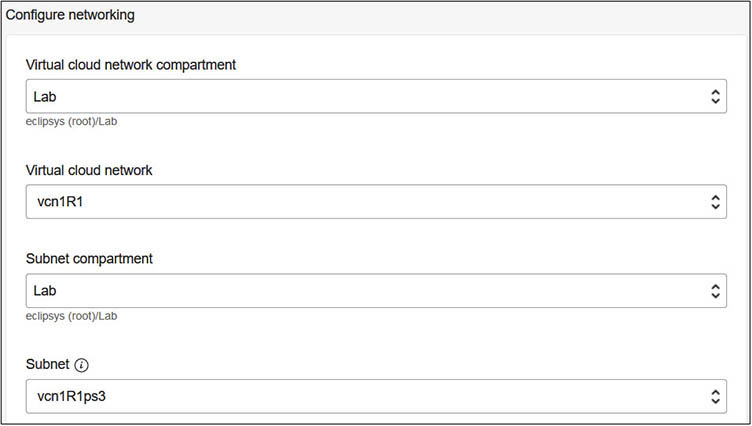


**8.**   Choose an instance shape (for example, VM.Standard.E2.1). Change any boot volume configuration options as required or accept the default settings.

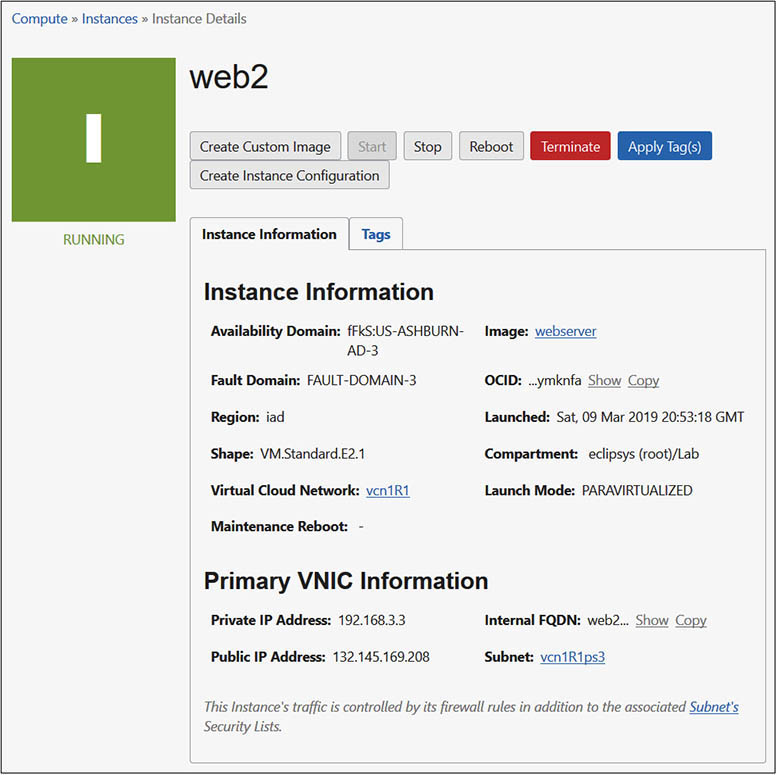


**9.**   Because you are creating this instance from a custom image, the public key used when creating the web1 instance is already present in the .ssh/authorized\_hosts file. You need only add a public key here if you generate a new key pair.

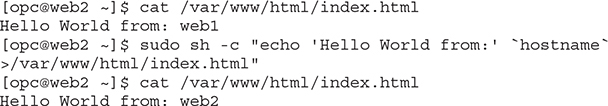
**10.**   Configure networking for your instance by choosing the compartment, VCN, Subnet compartment, and public subnet. Choose a different subnet from the one being used by the web1 compute instance.



**11.**   Click Show Advanced Options. Ensure that your compartment is correct. If your web2 instance resides in the same AD as web1, choose a fault domain (for example, FAULT-DOMAIN-3) that is different from the fault domain being used by web1. Choose Create to build the instance.



**12.**   Because the web2 instance is based on your web server custom image, the index.html file must be updated. Once the instance is up and running, connect using ssh and the matching private key, and run the following commands.



**13.**   In a browser, navigate to your web page by using the URL http://<*public IP address of web2*>, and you should see the Hello World from: web2 message.





**NOTE**    There seems to be a shift in thinking about compute instances and treating them as disposable commodities rather than as individuals. For example, if a web server like web2 has errors or failures, a system administrator may decide that instead of investing several hours in troubleshooting the error, it may be simpler and faster to clone a new web server from a custom image.

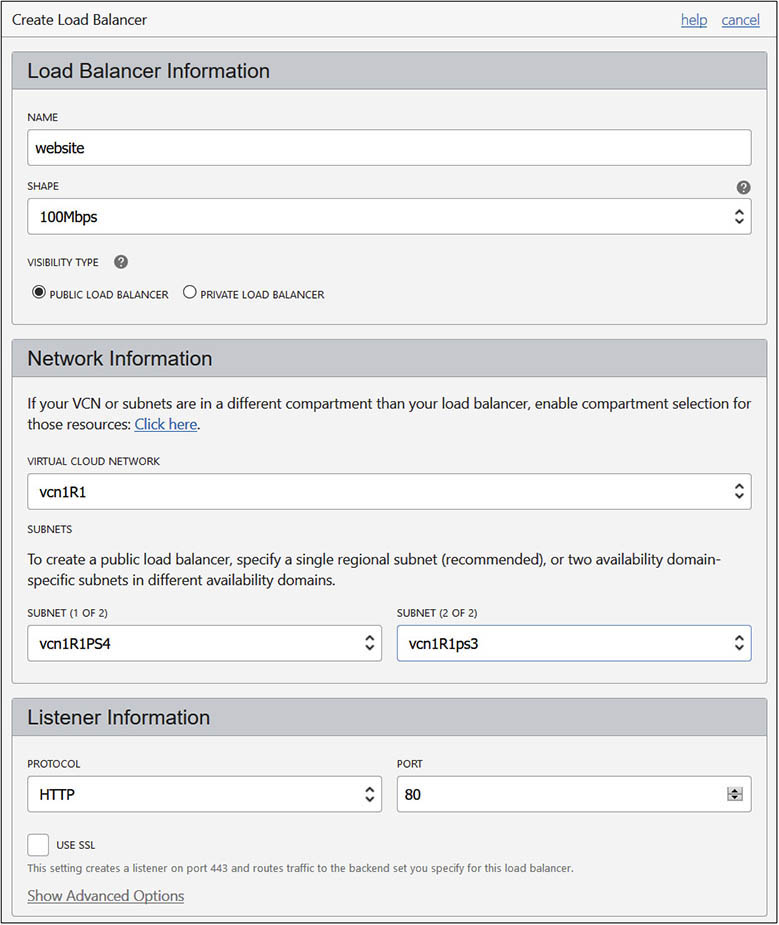
**Exercise 4-4: Create a Load Balancer to Route Traffic to Web Servers**

In this exercise, you will create a public load balancer to direct HTTP traffic to a backend set comprising the web1 and web2 compute instances.

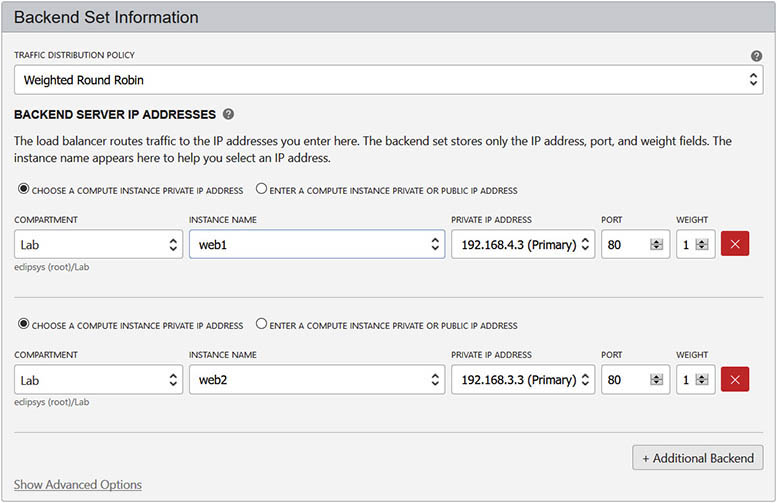
**1.**   Sign in to the OCI console, and choose your compartment.

**2.**   Navigate to Networking | Load Balancer, and choose Create Load Balancer.

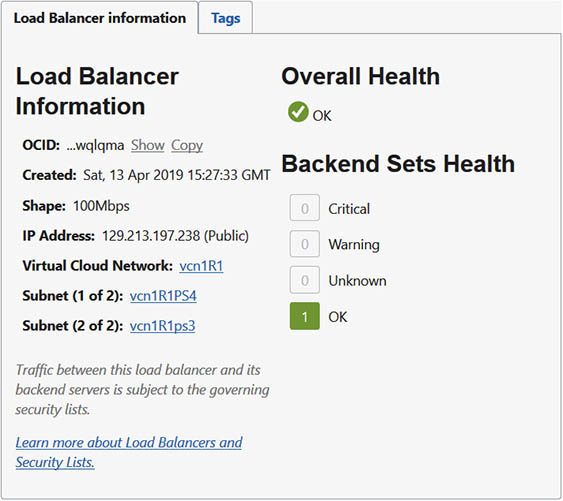
**3.**   Provide a load balancer name (for example, website) as well as a shape. Choose Public Load Balancer and select the VCN where compute instances web1 and web2 reside (for example, vcn1R1). If the region this is being created in has multiple ADs, choose two subnets in different ADs or choose a single regional subnet in a single AD region. Choose HTTP and port 80 for the listener.



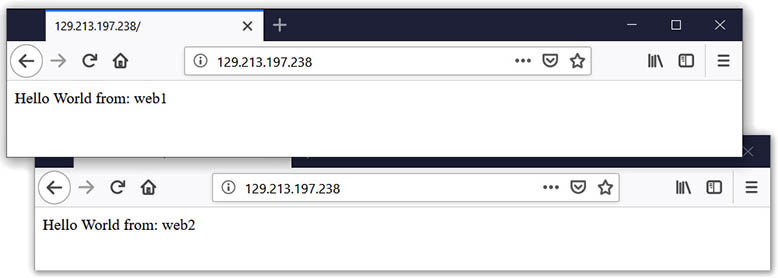
**4.**   Select Weighted Round Robin as the traffic distribution policy. Choose Enter A Compute Instance Private Or Public IP Address, and specify the public IP for web1. Add an additional backend server and specify the public IP for web 2. Both web1 and web2 are added as equally weighted backend servers. Choose Create to provision the load balancer.



**5.**   Once the load balancer is active, you may have to wait several minutes while the health check completes, and the Backend Sets Health shows up as OK.



**6.**   Navigate to the load balancer’s public IP address in your browser and repeatedly refresh the page. On consecutive HTTP requests, you should observe alternating traffic from web1 and web2.



Windows instances are slightly different from Linux instances. They do not use ssh key pairs, but rather use Windows credentials. Windows instances require an additional software license, and you are charged a runtime fee each time you use your Windows instance. You connect to the Windows instance using remote desktop protocol (RDP) on port 3389 as opposed to SSH on port 22, so the RDP port must be open for ingress traffic on the security list in the relevant VCN.

**Exercise 4-5: Create and Connect to a Windows Compute Instance**

In this exercise, you will create a Windows 2008 R2 server compute instance and connect to it using a remote desktop client over RDP.

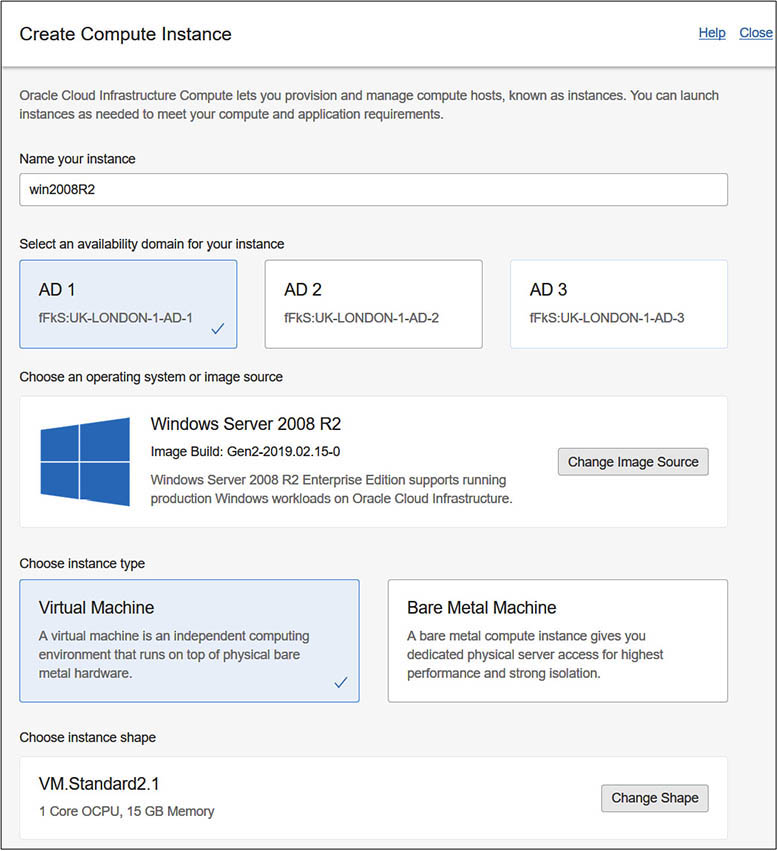
**1.**   Sign in to the OCI console and choose your compartment.

**2.**   Navigate to Compute | Instances and choose Create Instance.

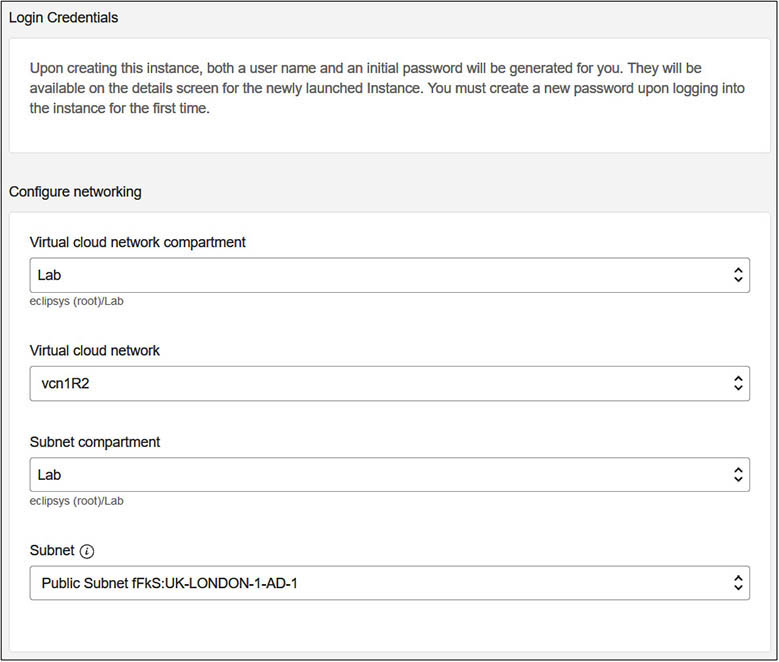
**3.**   Name your instance (for example, win2008R2) and select an AD for your instance (for example, AD1).

**4.**   For operating system or image source, choose Change Image Source and navigate to Platform Images, and select an operating system (for example, Windows Server 2008 R2). Check the box after you review and accept the terms of use and choose Select Image.

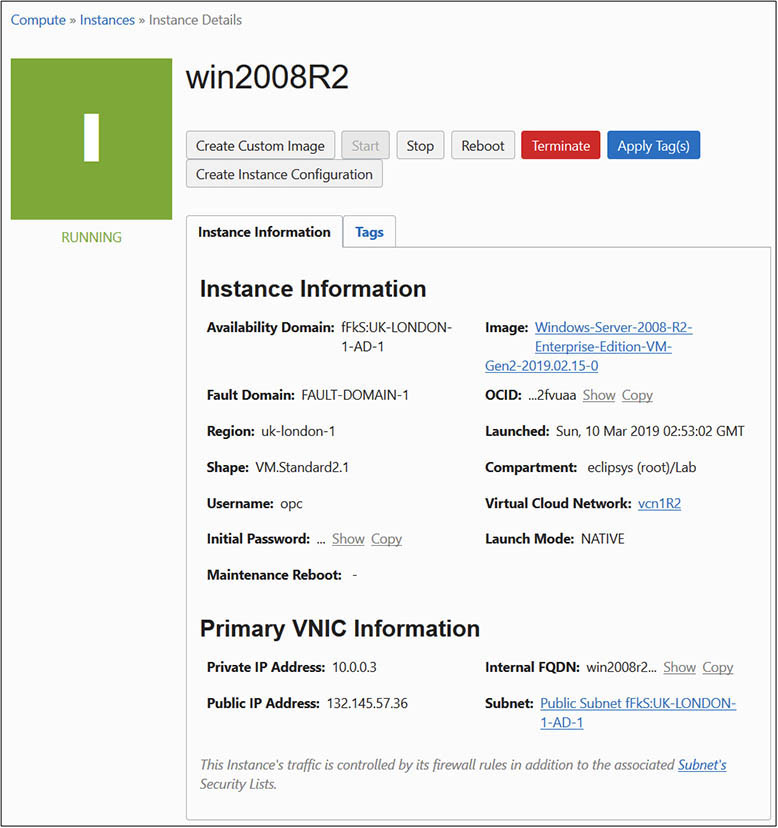
**5.**   Choose a shape for your Windows instance (for example, VM.Standard2.1). Change any boot volume configuration options as required or accept the default settings.



**6.**   Take note that upon creation of the Windows instance, both a username and an initial password are generated. Choose your VCN compartment, VCN, Subnet compartment, and Subnet, and click Create.



**7.**   Take note of the public IP address of the Windows compute instance, as well as the username and initial password once it is provisioned. Locate the RDP program you wish to use to connect to the instance. Many windows distributions have a Remote Desktop Connection program, and similar tools exist on Linux and Mac systems as well.



**8.**   Launch the RDP software and specify the public IP address of the Windows compute instance, and click Connect. If you are warned because the identity of the remote computer cannot be verified, choose to connect anyway. Choose the opc user and initial password. You are prompted to change your password and are now connected to your Windows instance.

Once created, compute instances require care and feeding just like regular on-premises compute instances. The next section discusses common management activities associated with compute instances.

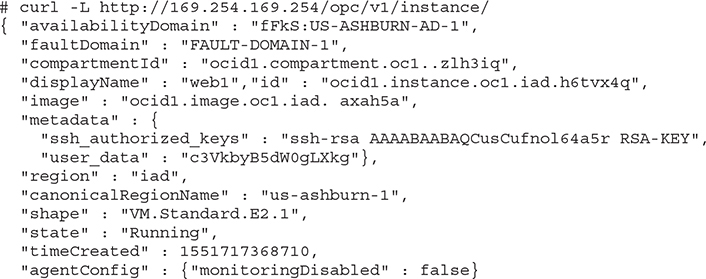
**Manage Compute Instances**

There are many APIs available for managing various aspects of compute instances, including stopping, starting, rebooting, and terminating the instance. Additional storage may be provisioned by adding block volumes. This is covered in detail in [Chapter 5](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5). You may also create secondary vNICs in some instances if it is supported by the chosen shape and platform. The APIs mentioned earlier are available through the console by using the menus adjacent to the instance or through scripts written using the CLI and SDKs.

When managing large infrastructure estates, scripted management is often more effective, so this section focuses on several common compute instance management tasks through the CLI.

**Instance Metadata**

Instance metadata on Oracle-provided images is retrieved by querying a special IP address while connected to an instance as follows. This provides instance metadata similar to the OCI console output.



**Instance Power Management**

Compute instance management tasks may be performed using the following OCI CLI command:

Images

ACTION commands relate to power management for the instance and include the following:

•   **START**   Power on

•   **STOP**   Power off

•   **RESET**   Power off and power on

•   **SOFTRESET**   Graceful shutdown and power on

•   **SOFTSTOP**   Graceful shutdown

**Multiple vNICs**

Secondary vNIC management is also supported through the OCI CLI. Current vNICs may be listed with the list-vnics command:



An additional vNIC may be added with the attach-vnic command:

Images

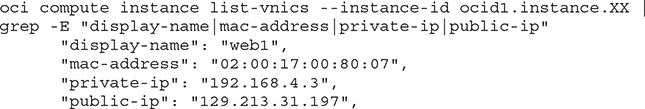
Use the list-vnics command, searching only for display-name, mac-address, and IPs:



Use the detach-vnic command to remove a vNIC:

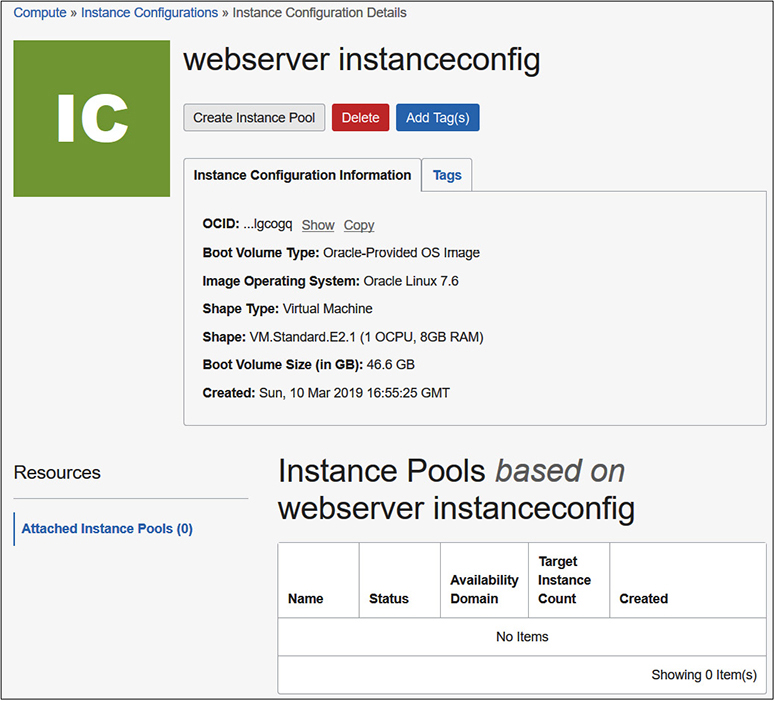
Images

Use the list-vnics command to confirm the detachment of the vNIC:



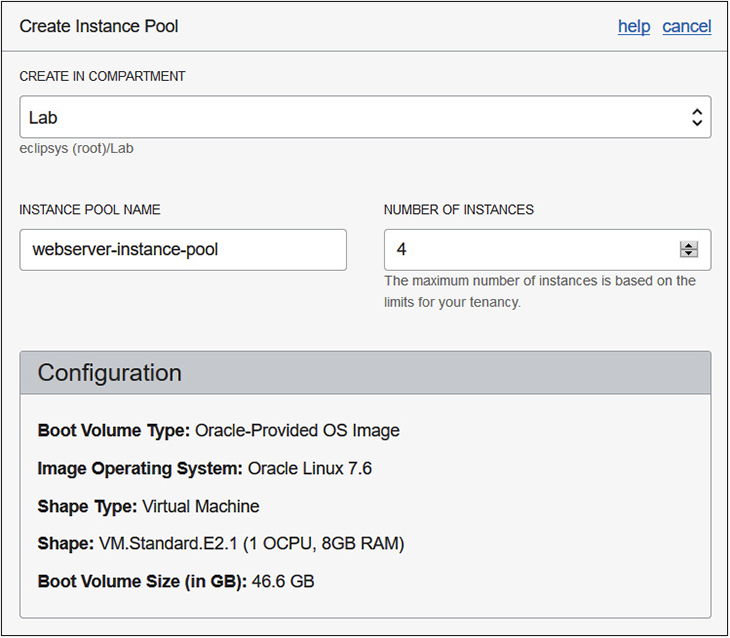
**Instance Configurations, Pools, and Autoscaling**

Instance configurations provide a system for creating configuration templates from existing compute instances. [Figure 4-9](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig9) shows an instance configuration created from the web1 instance documenting a set of configuration information including boot volume type and size, operating system, and compute shape.



**Figure 4-9**   Instance configuration information

Instance configurations form the basis for instance pools. These are pools of compute instances created using the instance configuration templates in a particular region. Instance configurations are different from custom images. Custom images are operating system images used as boot volumes for new instances whereas instance configurations contain a set of parameters to be used for instance pools. [Figure 4-10](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig10) shows how you can spawn many compute instances in a pool based a common instance configuration. In this example, up to four compute instances may be provisioned with a standardized instance configuration in this pool.

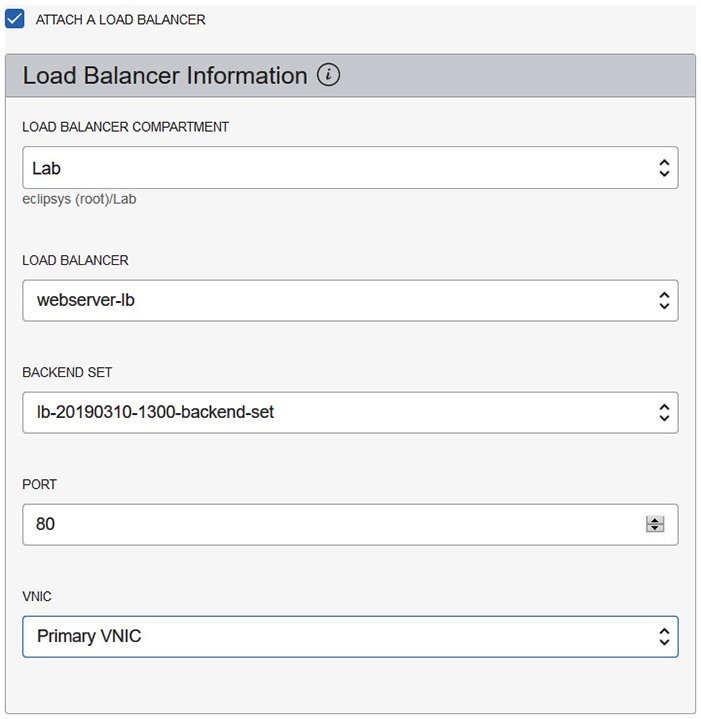


**Figure 4-10**   Instance pool configuration



**EXAM TIP**    Instances in an instance pool are provisioned in the same region but can be in multiple availability domains.

Load balancers may be attached to instance pools. [Figure 4-11](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig11) shows how provisioned instances in the pool may be added to the backend set associated with the load balancer. You may also specify multiple availability domains for the instance pool. As new instances are provisioned, these are created in the instance pool–specified ADs.



**Figure 4-11**   Attach a load balancer to the instance pool.

Instances in a pool may be managed together (or individually). For example, all instances in a pool may be reset with a single ResetInstancePool API operation. Other pool-level management APIs include StartInstancePool, StopInstancePool, and TerminateInstancePool. An instance pool may be in one of several states, including the following:

•   **Provisioning**   Initial creation of instances in the pool based on the instance configuration.

•   **Starting**   Instances are being launched.

•   **Running**   Instances are running.

•   **Stopping**   Instances are being shut down.

•   **Stopped**   All instances in the pool are shut down.

•   **Terminating**   All instances in the instance pool and their associated resources are being deprovisioned.

•   **Terminated**   The instance pool and its associated resources have been terminated.

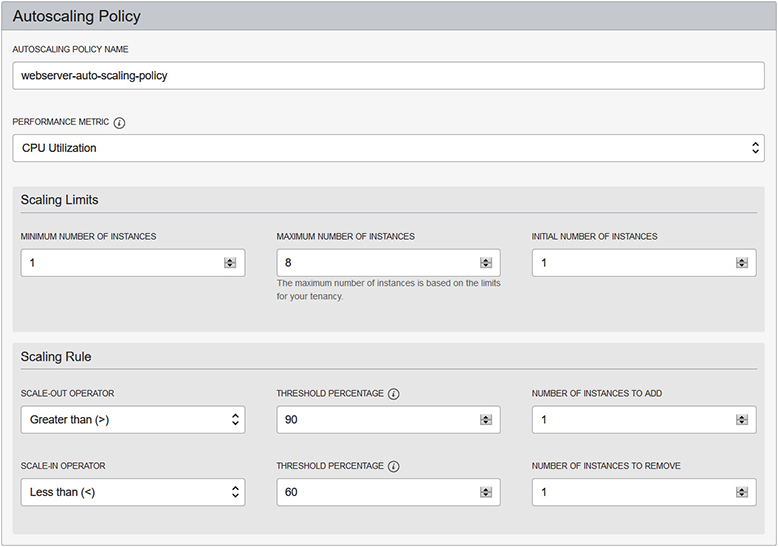
•   **Scaling**   The instance pool is being updated. Instances are being added or terminated.

Instance configuration and pools provide the basis for Autoscaling. Autoscaling refers to the dynamic addition or removal of instances from an instance pool based on an autoscaling policy. [Figure 4-12](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig12) shows an autoscaling configuration being created in a compartment for a specific instance pool.



**Figure 4-12**   Autoscaling configuration for an instance pool

The autoscaling adjustment is triggered by thresholds set in the autoscaling policy such as CPU or memory utilization. In [Figure 4-13](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig13), an autoscaling policy is defined based on the CPU utilization. Scaling limits and a scaling rule determine the autoscaling behavior.



**Figure 4-13**   Autoscaling policy defining the scaling metric, limits, and rule

Scaling limits define the minimum and maximum number of compute instances that may be automatically spawned in the instance pool, as well as the initial number of instances to start. The scaling rule defines the thresholds for scale-out, or adding instances, and for scale-in, or removing instances from the instance pool. The performance metrics of all instances in the pool are aggregated into one-minute intervals and averaged across the instance pool. When the average performance metric exceeds the given threshold in three consecutive one-minute intervals, an autoscaling event is triggered.

The cooldown period in [Figure 4-12](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig12) refers to the time spent evaluating the autoscaling thresholds after an adjustment is made. For example, the policy in [Figure 4-13](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig13) states that when the average CPU utilization across all instances in the pool exceeds 90 percent for three minutes in a row, then add one instance to the pool. Once the newly added instance is up and running, the pool must run for the cooldown period (300 seconds by default) before the averaged performance metrics are considered for another autoscaling event.

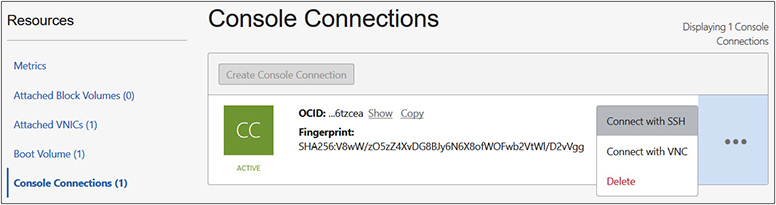


**EXAM TIP**    Autoscaling requires an instance pool. An instance pool requires an instance configuration. An autoscaling policy specifies scale-out and scale-in limits as well as the initial number of instances to start. The scaling rule defines the thresholds for scale-out and scale-in.

**Instance Console Connections**

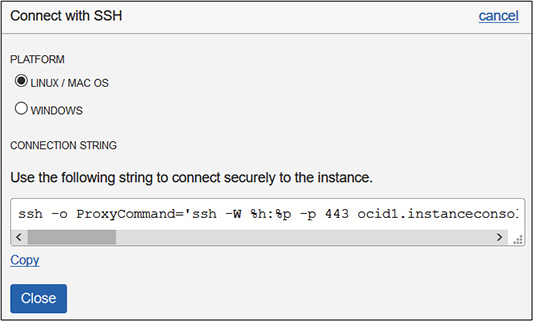
Compute instances are incarnated on servers in racks in data centers across the globe. There is no KVM (keyboard-video-mouse device) or monitor attached to the instance to see the console activity as you would if you were physically plugged into the server. Sometimes, an instance may not boot up properly or the SSH daemon on the operating system does not start so regular SSH access to the instance stops working. Troubleshooting is the most common reason you may wish to access the console. OCI provides a secure mechanism to connect to the instance console. For both Windows and Linux instances, you establish a secure SSH tunnel between your client machine and the compute instance. Your client machine can be running a Windows or Linux or Mac operating system. Once the tunnel is established, you make an SSH or VNC connection to the console of the compute instance. The VNC connection uses port forwarding to redirect a VNC connection to localhost:5900 to the compute instance.

To create a console connection to the web1 compute instance using the OCI console, navigate to Compute | Instances and choose the web1 instance. Choose Console Connections from the Resources area and select Create Console Connection. Provide an SSH public key you previously created and choose Create Console Connection. [Figure 4-14](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig14) shows the console connection resource that establishes the SSH tunnel to the console.



**Figure 4-14**   Console connection

You may connect over this tunnel with SSH or VNC to the compute instance. Choosing to Connect with SSH returns an interface, as shown in [Figure 4-15](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4fig15).

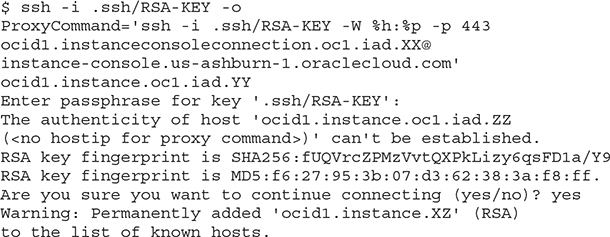


**Figure 4-15**   Console connection – Connect with SSH



**CAUTION**    The connection string provided assumes that the matching private key to the public key you provided when setting up the tunnel is available in the default location on your client machine. You may have to alter the provided connection string to reference your private key.

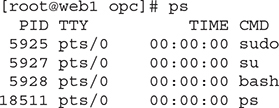
Notice that you may connect from a Linux, Mac, or Windows platform. You may copy the provided connection string to establish the console connection. If Windows is chosen as the client platform, the provided connection string consists of Powershell commands. Here is an excerpt from a Linux client connecting to the console of the web1 compute instance.





On Linux instances, you typically connect with SSH and work in a pseudo terminal session. In the previous excerpt from an SSH session, the ps (process) command is run, displaying, among other information, the PIDs of processes being run by the current user and the terminal associated with the process. Note that the terminal (TTY) is ttyS0. This means this is a serial console connection.

Consider the following excerpt taken when connected to the web1 compute instance using SSH and not using the console connection.



Here, all four listed processes run by the root user on the web1 compute instance are associated with the pseudo terminal pts/0.

**Chapter Review**

Compute instances are the quintessential IaaS components and OCI has done a great job simplifying the provisioning and management of these resources. This chapter discussed virtualized and bare-metal compute types and their associated shapes. The ever-expanding variety of available compute images will only improve the platform further. As the marketplace for partner images and Oracle images expands, provisioning once-complex software stacks will become much simpler.

A significant portion of this chapter is dedicated to hands-on practical exercises enabling you to create and manage several compute VM instances. This chapter also explored fault domains, especially important in single AD regions, as well as the ability to run legacy and custom operating systems on OCI through BYOH (Bring Your Own Hypervisor) and BYOI (Bring Your Own Image).

Autoscaling is a valuable feature for scaling systems through peak loads and down-scaling during off-peak times, and reduces costs of provisioning for the worst days along with reducing human effort. The chapter closed with a discussion on using instance console connections, typically for troubleshooting.

With your network and compute instances in place, you are ready to explore the available storage options, which is the subject of [Chapter 5](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5).

**Questions**

[**1**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans1)**.**   List the compute instance types available for provisioning on OCI.

**A.**   Virtual machines

**B.**   Paravirtualized machines

**C.**   Bare-metal machines

**D.**   OVM, KVM, and Hyper-V

[**2**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans2)**.**   Which of the following compute images may be used to create a compute instance?

**A.**   Oracle and Platform images

**B.**   Boot volumes and Partner images

**C.**   Paravirtualized machines

**D.**   OVA format images

[**3**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans3)**.**   A boot volume has been detached from an instance in AD1. Which of the following statements are true?

**A.**   A compute instance can be created in AD2 using the boot volume in AD1.

**B.**   A compute instance cannot be created in AD2 using the boot volume in AD1.

**C.**   The boot volume can be backed up in A1 and restored in AD2 and used to create a compute instance in AD2.

**D.**   The boot volume cannot be moved.

[**4**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans4)**.**   A compute instance is not starting up. You suspect a problem with the boot volume. Which of the following options may be used to troubleshoot this further?

**A.**   There is nothing further to do. The compute instance must be cloned and recreated.

**B.**   A console connection may be created to see if there is more information available on the console.

**C.**   The compute instance must just be reimaged from the same source image.

**D.**   The boot volume may be detached and attached to another working instance as a regular volume to access log files and examine configuration.

[**5**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans5)**.**   Which of the following statements is true?

**A.**   Instances in an instance pool may be provisioned across multiple VCNs.

**B.**   Instances in an instance pool may be provisioned across multiple regions.

**C.**   Instances in an instance pool are provisioned in the same AD.

**D.**   Instances in an instance pool are provisioned in the same region.

[**6**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans6)**.**   Instance metadata on Oracle-provided Linux images are retrieved by querying which special IP address while connected to an instance?

**A.**   <http://169.254.169.254/opc/v1/instance/>

**B.**   <http://127.0.0.1/opc/v1/instance/>

**C.**   <http://255.255.255.0/opc/v1/instance/>

**D.**   There is no such thing. Use the OCI console or CLI to get instance metadata.

[**7**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans7)**.**   Autoscaling requires which of the following resources in order to work? Choose all dependent resources.

**A.**   Multiple vNICs

**B.**   Instance pool

**C.**   Instance configuration

**D.**   Autoscaling policy

[**8**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans8)**.**   Which of the following statements is true?

**A.**   When autoscaling is enabled, OCI provisions as many compute instances as required, with no limit when scale-out thresholds are exceeded.

**B.**   When autoscaling is enabled, OCI provisions as many compute instances as required, limited by the autoscaling policy.

**C.**   When autoscaling is enabled, OCI provisions an equal number of compute instances as there currently are in the instance pool to ensure even load balancing.

**D.**   When autoscaling is enabled, OCI provisions zero compute instances. The cloud administrator must authorize the provisioning of new instances.

[**9**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans9)**.**   Which instance management ACTION commands are supported by the OCI CLI command: oci compute instance action --instance-id ocid1.instance.oc1.XX --action ACTION?

**A.**   TERMINATE

**B.**   CREATE

**C.**   STOP

**D.**   SOFTRESET

**E.**   HARDSTOP

[**10**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4ans10)**.**   Which hypervisors are supported by OCI on bare-metal instances?

**A.**   Kernel-based VM (KVM)

**B.**   Oracle VM (OVM)

**C.**   Oracle Virtual Box

**D.**   Oracle Solaris containers

**E.**   Hyper-V

**Answers**

[**1**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans1)**.**   **A, C.** VM and BM are the only available compute types.

[**2**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans2)**.**   **A, B.** Oracle, Platform, and Partner images and Boot volumes may be used to create a compute instance.

[**3**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans3)**.**   **B, C.** A boot volume used as the image source for a compute instance must be available in the same AD chosen to host the compute instance. A boot volume may be backed up and restored in a different AD.

[**4**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans4)**.**   **B, D.** Using a console connection to see if more information is available on the instance console is usually helpful as is detaching the boot volume from the problematic instance and attaching it to a working instance as a regular volume to access log files and examine configuration.

[**5**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans5)**.**   **D.** Instances in an instance pool are provisioned in the same region.

[**6**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans6)**.**   **A.** Instance metadata on Oracle-provided Linux images are retrieved by querying <http://169.254.169.254/opc/v1/instance/>.

[**7**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans7)**.**   **B, C, D.** Autoscaling requires an instance pool, instance configuration, and an autoscaling policy.

[**8**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans8)**.**   **B.** When autoscaling is enabled, OCI provisions as many compute instances as required, limited by the autoscaling policy.

[**9**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans9)**.**   **C, D.** ACTION commands relate to power management for the instance and include: START, STOP, RESET, SOFTRESET, and SOFTSTOP.

[**10**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#r_ch4ans10)**.**   **A, B, E.** OCI provides support for installing several hypervisors on bare-metal instances, including KVM, OVM, and Hyper-V.