**CHAPTER 5**

**Storage**

In this chapter, you will learn how to

•   Create and manage block storage volumes

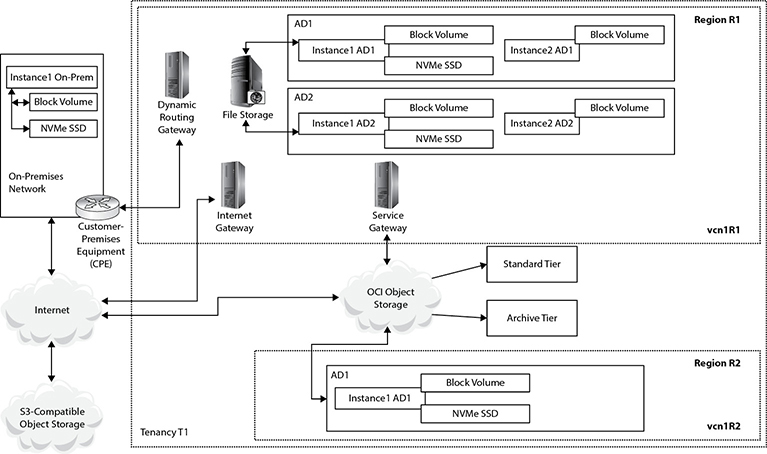
•   Use object storage

•   Explain file storage service

Storage is an essential ingredient in the cloud computing puzzle and unsurprisingly a comprehensive array of storage options is available on OCI. Instead of relying on teams of storage engineers to prepare SAN (storage area network), NAS (network attached storage), DAS (direct attached storage), and other storage types for your enterprise, you can rapidly provision equally diverse storage options on OCI with ease. Each storage option serves a different purpose.

The fastest, most expensive storage options available in OCI are NVMe SSD storage drives attached locally to a compute instance. This storage is typically used in high performance computing where high IO speeds are required, such as an important transactional database, and provides terabyte scale capacity. This is not durable storage because it is not replicated to other ADs, unlike block volumes that are durable and fast as they are based on NVMe SSD storage in a storage server and provide petabyte scale storage. The slowest and cheapest storage option is OCI object storage, also providing petabyte scale storage. Archive tier object storage buckets are appropriate for long-term storage when some data must be kept safely but IO speeds are not important, such as keeping several years of financial record backups for audit or compliance purposes. The File Storage Service (FSS) provides network file systems (NFSv3) that provide shared storage to instances in the same region and offers exabyte scale storage.

[Figure 5-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig1) contextualizes the most common OCI storage offerings within a tenancy spanning two regions.



**Figure 5-1**   OCI storage topology

Region 1 consists of two ADs, each hosting two compute instances. Instance1 AD1 has block volume and NVMe SSD storage attached. As discussed in [Chapter 4](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4), when a compute instance is provisioned a boot volume is chosen. This boot volume resides on OCI block volume storage and behaves like a regular disk volume in your PC or server. If this was a database server, the Oracle binary files ($ORACLE\_HOME) could be placed on the block volume, while the database files would reside on the high-speed local NVMe SSD disk drives attached to Instance1 AD1. Instance2 AD1 has only a block volume attached. Instance1 On-Prem also has a block volume (regular disk drives) and local NVMe SSD storage.

Instance1 AD1 and Instance1 AD2 can concurrently mount network attached storage over NFS (network file system protocol) through the File Storage Service (FSS). FSS is versatile shared storage that is slower than block volumes but faster than object storage. FSS is typically used for file server type applications. Through the CPE-DRG bridge, Instance1 On-Prem may access the file storage service if the correct permissions are in place.

Object storage is a relatively new construct based on tiered storage buckets. These are akin to folders or directories in a typical file system but there is no hierarchy. Object storage is accessible over the Internet. A bucket may be either standard or archive tier. These tiers are differentiated by performance and cost. Instance1 On-Prem may access the same object storage bucket over the Internet as Instance2 AD2 through an OCI service gateway.

With this 10,000-foot view of the primary storage options, you are ready to dive deeper into the storage intricacies available on OCI.

**Block Storage**

Block volumes are provided to your compute instances by the OCI block volume service. This service manages and carves out block storage volumes as per your requirement. Block volumes may be created, attached, connected, or detached from compute instances. In fact, a block volume may be detached from one compute instance and attached to another instance in the same AD, thereby moving the volume. They may be grouped with other block volumes to form a logical entity known as a volume group. Volume groups may be backed up together to form a consistent point-in-time, crash-consistent backup that is also useful for cloning.

A boot volume is a special type of block volume because it contains a boot image. OCI differentiates between boot volumes and block volumes. When you create a compute instance, you may either create a new boot volume, which hosts the operating system boot image, or use an existing available unused boot volume. Boot volumes are discussed in [Chapter 4](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4). Additional block volumes may be created and attached to your instances to expand available storage. Block volumes may be used for database storage or for hosting general purpose file systems.



**EXAM TIP**    There are two types of block storage volumes. A boot volume is used as the image source for a compute instance while a block volume allows dynamic expansion of storage capacity of an instance.

When you create a block volume, you specify several attributes, including the following:

•   **Name**   This is the block volume description.

•   **Compartment**   The logical container to which the block volume belongs.

•   **AD**   Only instances in the same AD can use this volume.

•   **Size**   Can be between 50GB and 32TB, at the time of this writing.

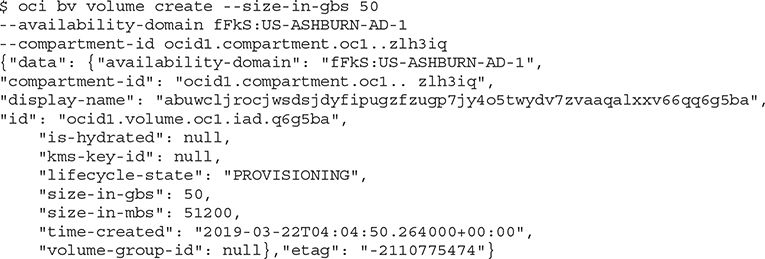
•   **Backup Policy**   OCI offers three optional block volume backup policies: bronze, silver, and gold.

•   **Encryption**   All block volumes are encrypted using either Oracle-managed keys or customer-managed keys stored in an OCI key management vault.

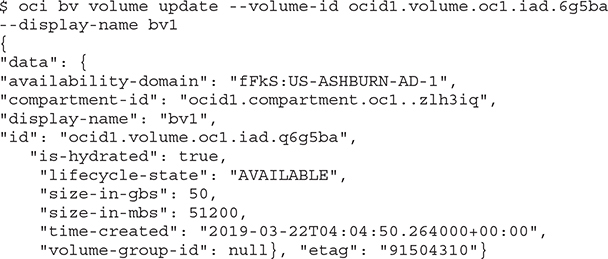
This section describes typical block volume administration activities including creating, listing, using, deleting, and backup options.

**Create Block Volumes**

The following OCI CLI command creates a 50GB block volume in AD1 in a specific compartment.



The system-generated block volume display-name property may be updated, for example to bv1, with the following command:



One of the block volume attributes to note is lifecycle-state, which may have one of the following values:

•   **PROVISIONING**   Block volume is being created.

•   **AVAILABLE**   Block volume is ready to be attached, detached, connected, or disconnected.

•   **TERMINATING**   Block volume is being deleted.

•   **TERMINATED**   Block volume has been deleted and is no longer available.

•   **FAULTY**   Block volume requires diagnosis.

•   **RESTORING**   Block volume is being restored from a backup.



**NOTE**    A block volume belongs to a specific AD. This makes sense if you consider that it is physically manifested across several disk drives in a data center somewhere.

**Exercise 5-1: Create a Block Volume**

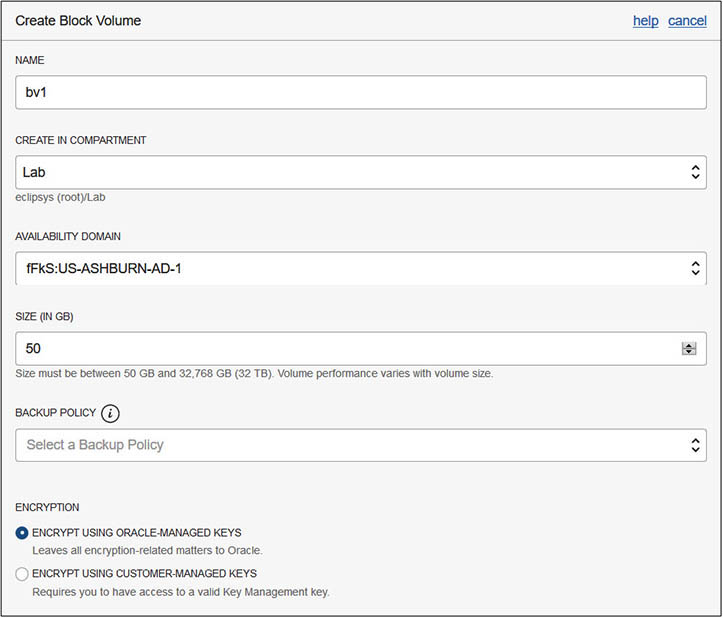
In this exercise, you will create a block volume in the same AD as compute instance web1. If you completed all the exercises in [Chapter 4](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4), you should have the required instance.

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

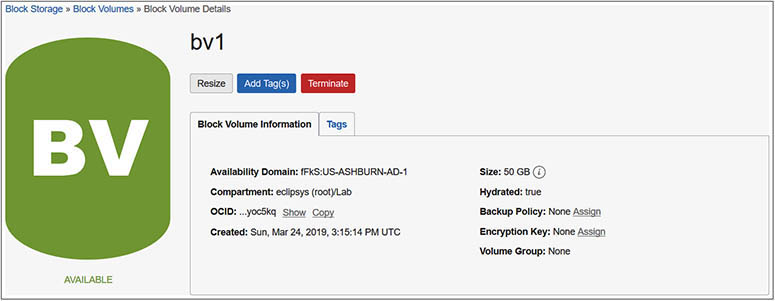
**2.**   Navigate to Compute | Instance and choose the web1 instance. Take note of the AD in which it resides.

**3.**   Navigate to Block Storage | Block Volumes and choose Create Block Volume.

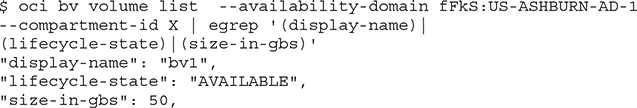
**4.**   Provide a name (for example bv1), choose the compartment to which it will belong, and choose the same AD as compute instance web1. Specify a size—for example, 50GB. Leave the backup policy and encryption options at their default settings. Provide a tag as desired, and choose Create Block Volume.



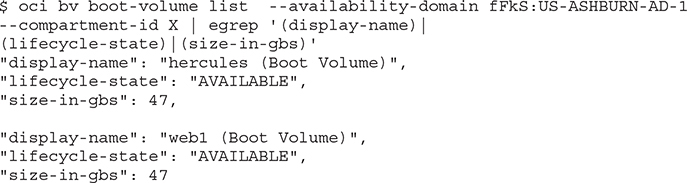
**5.**   The block volume provisioning process begins and after a few minutes its status changes to AVAILABLE.



You may list all block volumes in an AD that belong to a compartment with the following OCI CLI command:



Boot volumes in the same AD may be listed by querying the block volume service with the following OCI CLI commands. Note the difference between the previous command and this command. Both call the block volume server (oci bv) but block volumes use the volume API while boot volumes use the boot-volume API.



**Attaching Block Volumes**

Once a block volume has been provisioned and is at the AVAILABLE lifecycle-state, you may start using it by attaching it to a compute instance. There are two types of volume attachments:

•   **iSCSI**   Often pronounced “i-scuzzy,” it connects the block volume to the instance using an TCP/IP network connection. It may be useful to think of iSCSI as an interface dedicated to routing IO packets to a storage subsystem over TCP/IP. iSCSI is an established storage communications protocol and is supported on bare-metal and VM instances. Once a block volume is attached with iSCSI, it must be connected to the instance.

•   **Paravirtualized**   This attachment type is available only on VMs and adds an extra IO virtualization layer. However, the block volume attached using the paravirtualized attachment type is ready to use and does not require further connection intervention.



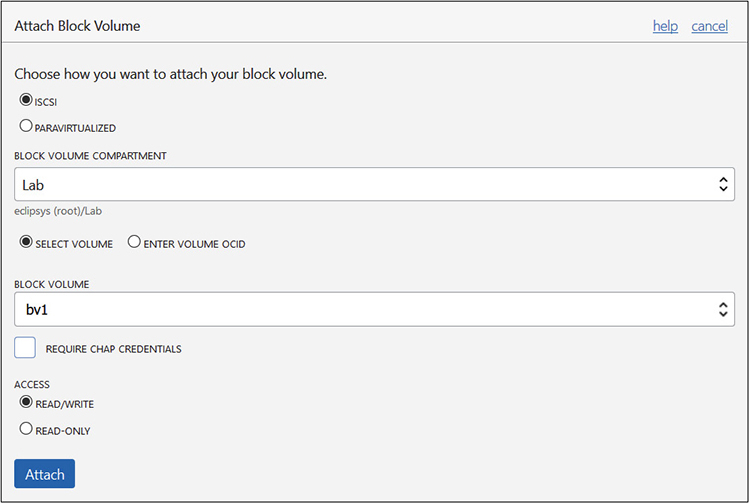
**CAUTION**    iSCSI attachments are the only option when connecting block volumes to bare-metal instances, Windows and Linux VM instances based on Oracle-provided images published before February 2018 and December 2017 respectively.

When attaching a block volume to your instance, you also choose the access type. An attached block volume may be accessed in read/write or read-only modes. Read/write mode is the default access type, allowing instances to change data on the volume, while read-only attached volumes are protected from modification. To change the access type of a block volume, you need to detach the volume and reattach it with the new access type. You may also require that iSCSI-attached block volumes be authenticated using CHAP (Challenge Handshake Authentication Protocol) when connecting the attached volumes to the instance.



**EXAM TIP**    Boot volumes are always attached with read/write access. To obtain information from the boot volume—for example, while troubleshooting an instance boot-up issue—the boot volume may be detached from an instance and attached with read-only access to another instance as a regular block volume.

[Figure 5-2](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig2) shows the Attach Block Volume console interface where you specify the attachment type (iSCSI or paravirtualized), the block volume compartment, the volume to be attached, whether CHAP credentials are required or not, and the access type (read/write or read-only).



**Figure 5-2**   Attach a read/write block volume using iSCSI

Some instances created using Oracle-provided Linux images support consistent device paths. When attaching the volume to an instance, you may have an additional option to choose a consistent device path—for example, /dev/oracleoci/oraclevdb. Consistent device paths remain constant between instance reboots. This enables you to refer to a consistent path when performing several operations on a volume, including partitioning, creating file systems, mounting file systems, and specifying automatic mounting options in the Linux /etc/fstab file. These operations are possible with instances that do not support consistent device paths by referencing the volume by its unique block identifier.



**NOTE**    iSCSI-attached volumes have higher IOPS performance than paravirtualized-attached volumes, which have additional overheads due to IO virtualization.

**Exercise 5-2: Attach a Block Volume to a Linux Instance**

In this exercise, you will attach the previously created block volume to the web1 compute instance.

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

**2.**   Navigate to Compute | Instance and choose the web1 instance.

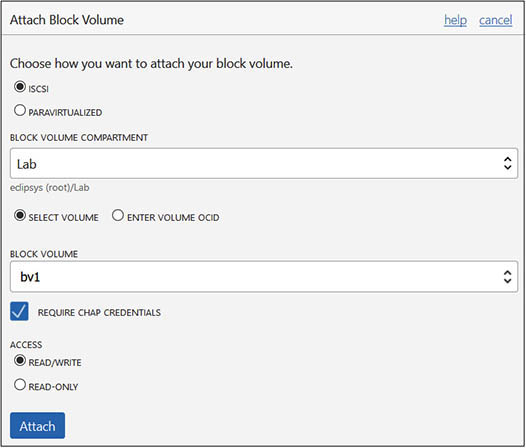
**3.**   Choose Attached Block Volumes and click Attach Block Volumes.

**4.**   Choose iSCSI as the attachment type.

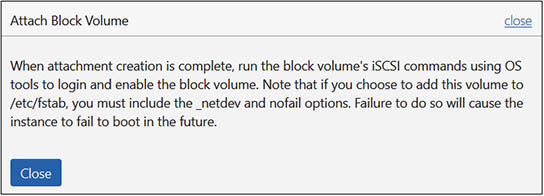
**5.**   Choose the relevant compartment and newly created block volume.

**6.**   Check the Require CHAP Credentials checkbox.

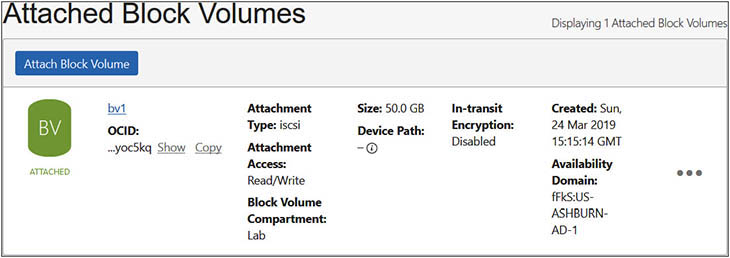
**7.**   Ensure that read/write access is selected and choose Attach.



**8.**   As the attachment process begins, the console notifies you to run the block volume’s iSCSI commands to connect and enable the volume. There is also information of mounting options required when specifying this volume in the Linux /etc/fstab file. This is discussed in the next section.



**9.**   When the block volume is attached, the instance information page in the console shows the newly attached block volume.



**Connecting Block Volumes**

Once a block volume is attached to an instance using iSCSI, it must be connected to the instance. The connection procedure varies based on two fundamental criteria:

•   **Operating system image**   Block volume connection procedure is different on Windows and Linux.

•   **CHAP requirements**   Additional connection steps are required if CHAP credentials are required to connect the block volume to the instance.

**Exercise 5-3: Connect a Block Volume to Your Linux Volume Using iSCSI and CHAP**

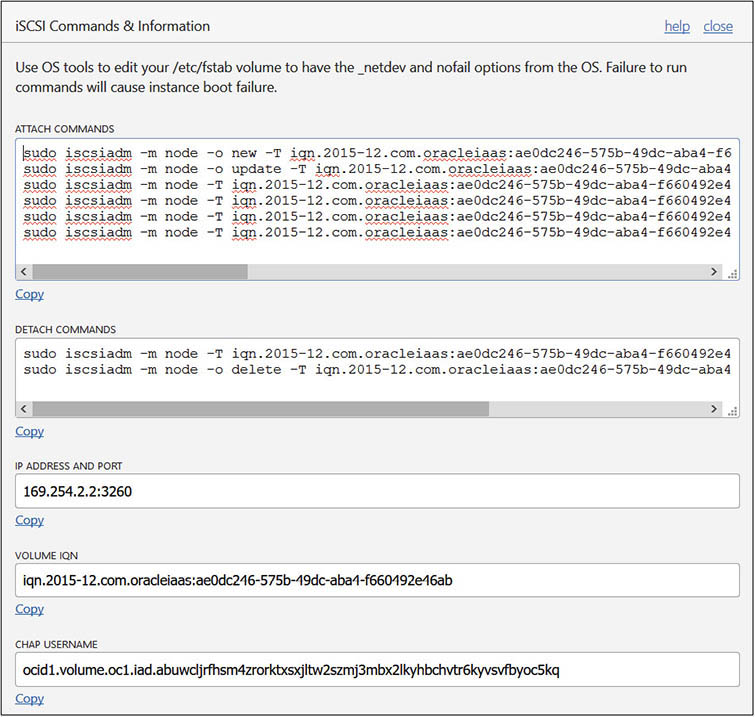
In this exercise, you will connect the previously attached block volume to the web1 compute instance.

**1.**   Sign in to the OCI console, navigate to Compute | Instance, choose the web1 instance, and click Start it if it is not yet started.

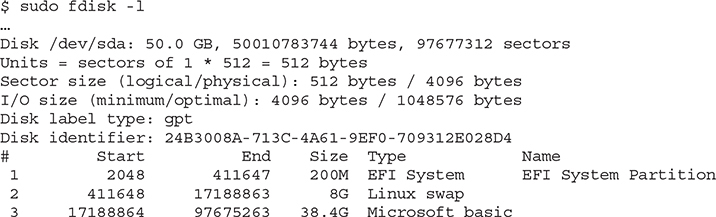
**2.**   Navigate to the Attached Block Volumes section, hover your mouse over the ellipses on the right side of the bv1 block volume, and choose iSCSI Commands & Information.



**3.**   Copy the attach commands to a text editor. You will need this to connect the volume to the instance from the Linux shell. Note that the attach commands include the block volume IP address and port number, the volume IQN (iSCSI qualified name), and the CHAP username and password. These commands are used in Steps 5–10.



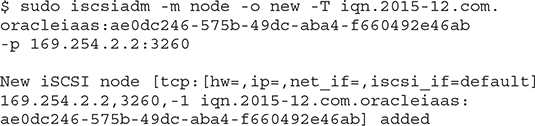
**4.**   Make an SSH connection to the web1 instance and describe the attached disks with the fdisk -l command. There is a single disk, the boot volume, that was attached when the instance was created.



**5.**   Use the following attach command to register the newly attached volume with the instance:

Images

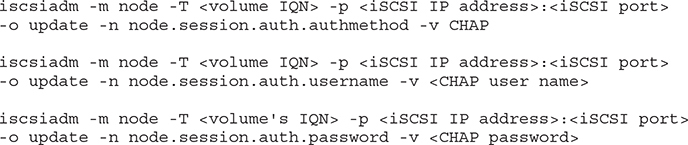
**6.**   When the newly attached volume is registered, the output should resemble the following:



**7.**   For the registration to persist reboots, issue the following command, which returns nothing when it completes successfully.

Images

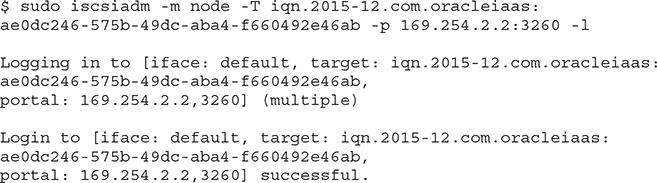
**8.**   Because you chose that CHAP credentials were required to connect to the volume when creating the block volume attachment, issue the following commands. Notice there is no output when the iscsiadm commands complete successfully.



**9.**   Log in to iSCSI using this command:

Images

**10.**   When you are successfully connected, the output resembles the following:



**11.**   You may run fdisk -l again to confirm the new block volume is listed as follows:



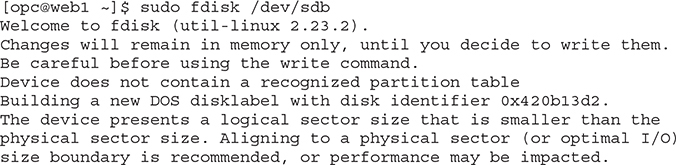
Once a block volume is connected and visible to an instance, it may be used for ASM disk storage for databases (discussed in [Chapter 6](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch6.xhtml#ch6)) or for file systems, or for pretty much anything you would use a disk volume for on a regular on-premises server to which a disk has been presented.

**Exercise 5-4: Format a Block Volume, Create a File System, and Mount the Volume**

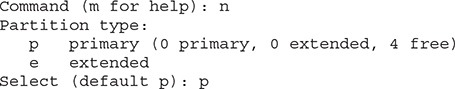
In this exercise, you use the previously connected block volume in the web1 compute instance. A directory (/app) is created to mount the block volume. The volume is partitioned with fdisk. An ext4 file system is created and the volume is mounted on the /app directory. To persist this mapping through reboots, the /etc/fstab file is updated with the UUID of the block volume mapped to the /app directory. The web1 instance used in this exercise does not support consistent device paths.

**1.**   Make an SSH connection to the web1 instance.

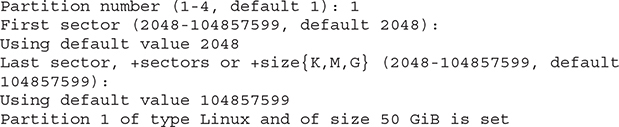
**2.**   Use the fdisk command with the device path (for example, /dev/sdb) for the newly connected block volume.



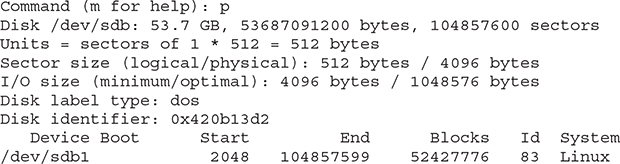
**3.**   Choose n to create a new partition and choose p for primary partition type.



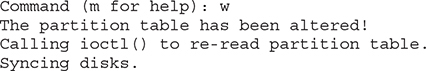
**4.**   Accept the default partition number 1 and the default settings for the first and last sectors because you are allocating all the space on this disk to this partition.



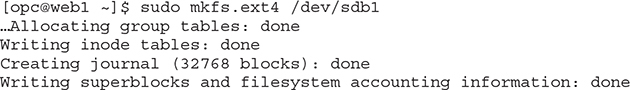
**5.**   Choose p to print the configuration to the screen.



**6.**   Choose w to write or save the configuration.



**7.**   Create an ext4 file system on this new partition.



**8.**   Create a directory to mount the file system.

Images

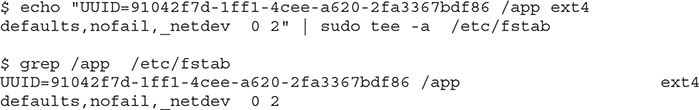
**9.**   Mount the file system to the /app directory and confirm that it has been mounted.

Images

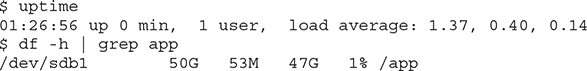
**10.**   Your block volume is ready to be used. However, once the compute instance restarts, the partition must be manually mounted again unless you configure automounting by adding an entry into the /etc/fstab file. Because this instance does not support consistent device paths like /dev/oracleoci/oraclevdb, you have to use the disk partition unique identifier (UUID), which you get from the blkid command.

Images

**11.**   An entry must be placed in the /etc/fstab to mount the /app mount point to the UUID along with the mount type (ext4 in this case) and several mount options to ensure the instance does not hang because of this volume mount. The nofail option basically lets the operating system know that it should ignore any failures associated with this mount, whereas the \_netdev option indicates there is a dependency on the networking services that should be started on the instance prior to mounting this volume. These options allow the instance startup routine to get services started in the correct sequence and to proceed with the startup routine even if there are errors when mounting this volume.



**12.**   Test the persistence of this setting by rebooting the instance. You should find the /app volume when the instance restarts.

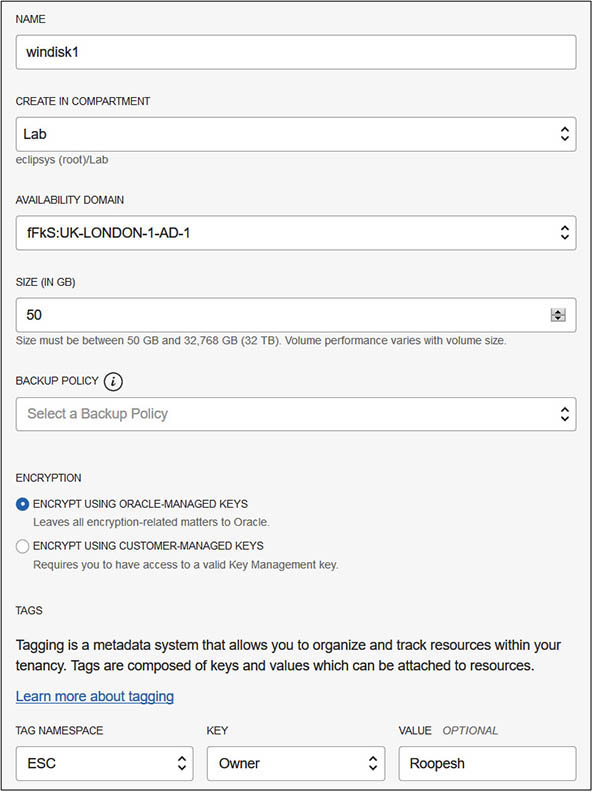


The steps to create and attach a block volume to Linux and Windows compute instances are identical. Connecting the volume to the instance is different because of the different operating system tools required for iSCSI and disk management.

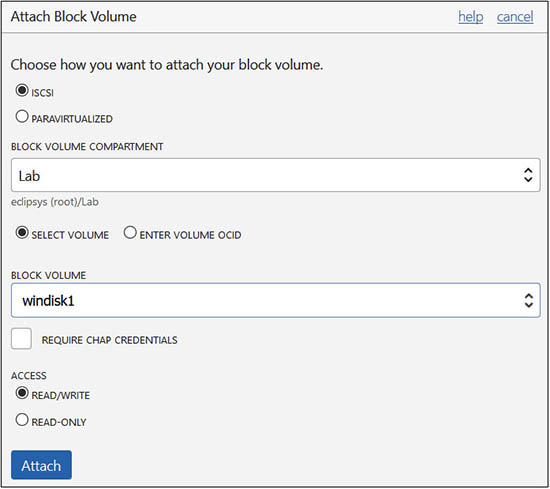
**Exercise 5-5: Present a Block Volume to a Windows Instance**

To complete this exercise, you need a Windows compute instance—for example, one based on the Oracle-provided Windows 2016 image.

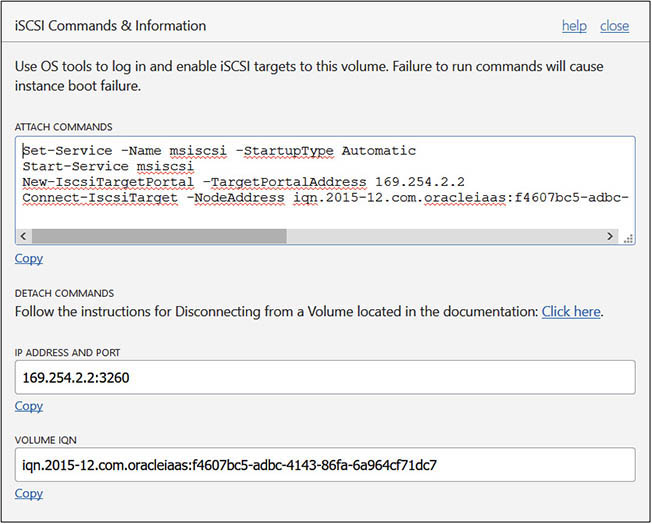
**1.**   Create a 50GB block volume in the same AD as your Windows instance.



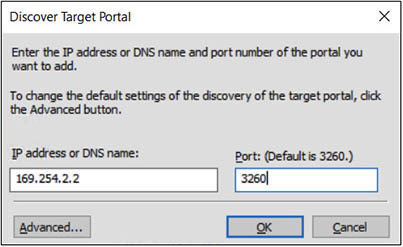
**2.**   Attach the block volume to your instance using iSCSI. Choose a read/write access mode and select Attach. You will be reminded to run the block volume’s iSCSI commands to log in and enable the volume once the attachment is created.



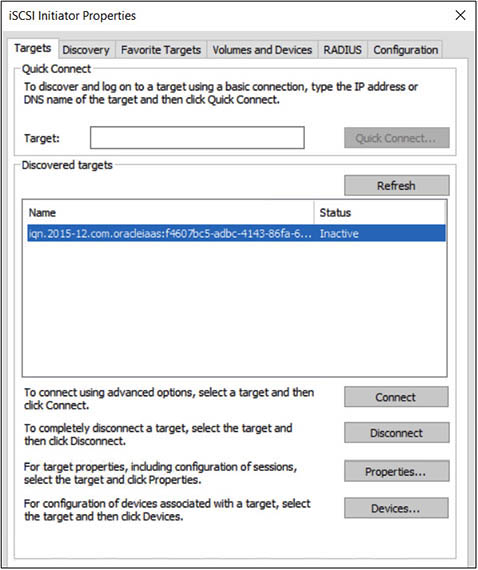
**3.**   Viewing the iSCSI Commands & Information on the attached block volume yields Windows Powershell commands, the IP address and port for the volume, as well as the volume IQN.



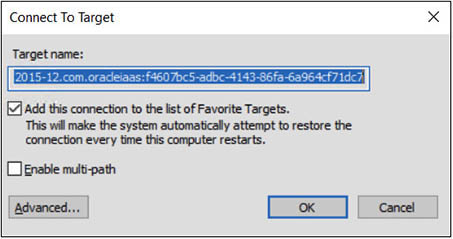
**4.**   Connect to the Windows instance with a remote desktop tool, search the Windows menu for iSCSI Initiator, and launch the tool. Navigate to the Discovery tab, choose Discover Portal, specify the IP address and port number, and select OK.



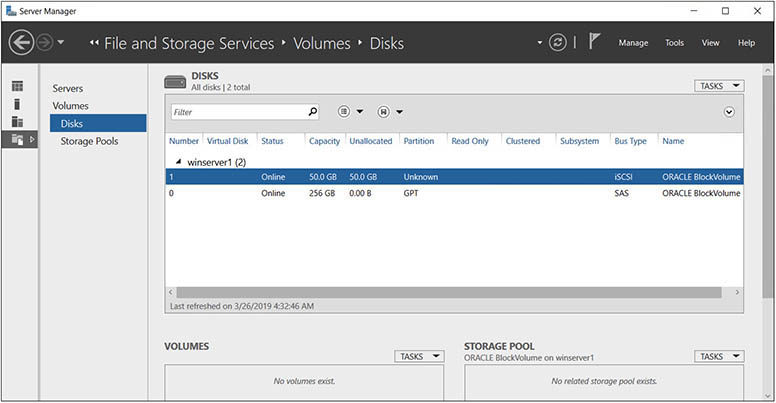
**5.**   Navigate to the Targets tab, and the discovered volume should be listed under discovered targets (though marked as inactive). Highlight the new target and then choose Connect.



**6.**   Ensure that the Add This Connection To The List Of Favorite Targets checkbox is selected before choosing OK. The volume is now connected to the instance.



**7.**   In Windows Server Manager, use the File And Storage Services | Volumes | Disks interface to mount the newly added volume.



**Block Volume Backup Options**

Backups protect your data and are an essential component of any business continuity and disaster recovery strategy. Point-in-time backups of data on a block volume may be taken manually or by assigning a policy that specifies a backup schedule. Block volume backups are encrypted and stored in object storage buckets (discussed later in this chapter) and may be restored as new volumes to any AD within the region they are stored.

Block volume backups may be either

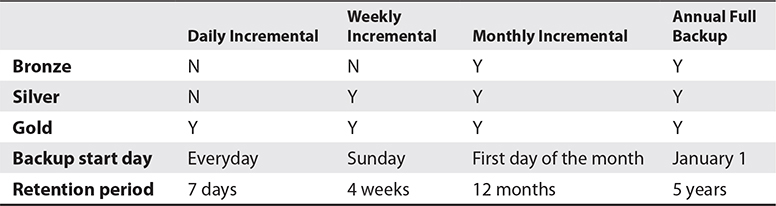
•   **Full**   All changes since the volume was created are backed up.

•   **Incremental**   Only changes since the last backup are saved.

The rate of changes on the block volume determines the size of the incremental backup, but it is generally smaller and faster than a full backup.

**Block Volume Backup Policies**

A block volume may be backed up manually or be assigned one of the automatic backup policies listed in [Table 5-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5tab1). These policies specify an increasingly aggressive backup and retention schedule as they progress from Bronze to Silver to Gold.

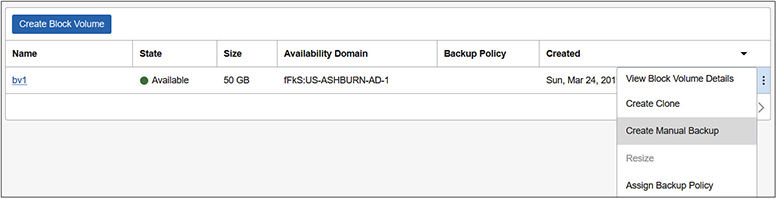


**Table 5-1** Volume Backup Policies

A block volume may be assigned a policy at any time in its lifecycle. A volume assigned a Silver policy is incrementally backed up weekly on a Sunday, monthly on the first day of the month, and annually on January 1. Once a policy-based backup expires (the backup is older than the policy retention period), it is automatically deleted. If you want to retain a backup for longer than its retention period, a manual backup must be performed.

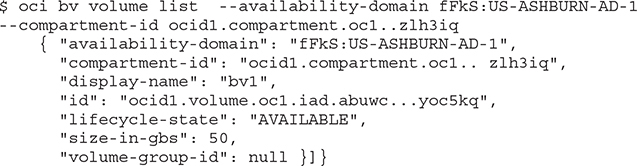
**Manual Block Volume Backups**

You may back up block volume using the OCI CLI, the API, or the console. [Figure 5-3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig3) shows the console interface used to create a manual backup of a volume or to assign a backup policy to a volume.

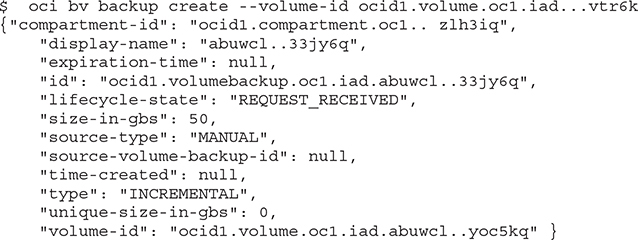


**Figure 5-3**   Block volume backup options

The following OCI CLI command lists the block volumes in a specific compartment. Note that some of the OCI CLI output has been removed for brevity.



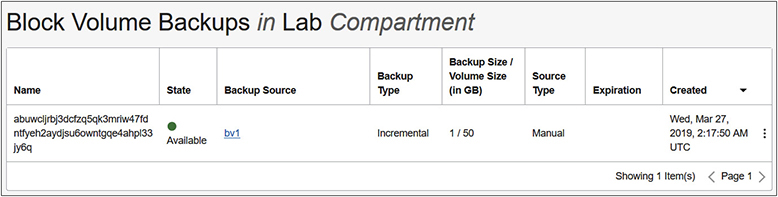
Using the volume\_id, you can create a manual backup using the OCI CLI as follows:





**CAUTION**    If you do not specify the backup type to either incremental or full, the OCI CLI defaults to an incremental backup.

[Figure 5-4](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig4) lists the block volume backup made with the earlier OCI CLI command.



**Figure 5-4**   Block volume backup list

**Exercise 5-6: Create a Full Backup of a Block Volume**

You may use the console, the CLI, or the API to back up a volume. Unless you use a predefined policy, you may want to create backup scripts that run on your preferred schedule. Hence, this exercise focuses on backups using the CLI and assumes there is a block volume available to be backed up like the one created in Exercise 5-1.

**1.**   List the available block volumes in your compartment with the CLI command, taking note of the volume-id for the block volume you wish to back up.

Images

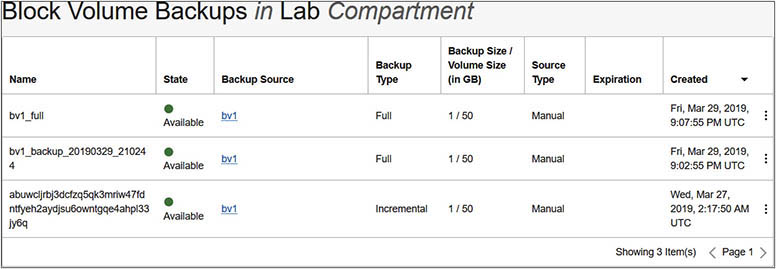
**2.**   Create a FULL backup by specifying the –type parameter, and provide a display name if desired. The CLI command to create a block volume backup requires the volume-id returned by the previous command.

Images

**3.**   The volume backup begins with the life-cycle property being set to "REQUEST\_RECEIVED" before changing to "AVAILABLE".

**4.**   To take a manual backup using the console, navigate to Block Storage | Block Volume, choose the ellipses menu adjacent to the block volume, and choose Create Manual Backup (see [Figure 5-3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig3)). This menu also lets you assign a backup policy (Bronze, Silver, or Gold) to the volume.

**5.**   To view the available block volume backups, navigate to Block Storage | Block Volume Backups. The full backup created earlier in the exercise should be listed.



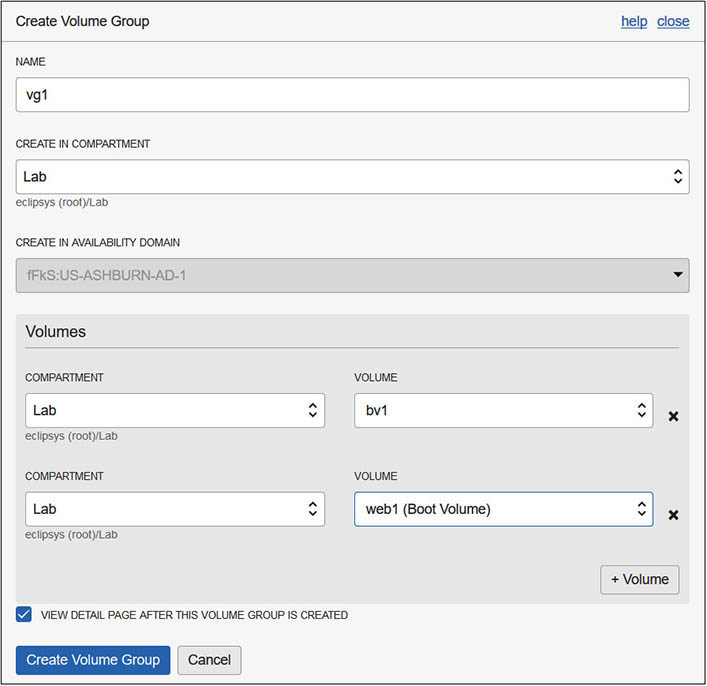
**Volume Groups**

Multiple volumes including boot and block volumes may be grouped together as a volume group. Volume groups may be backed up and restored as a consistent point-in-time set or cloned to create new consistent environments.

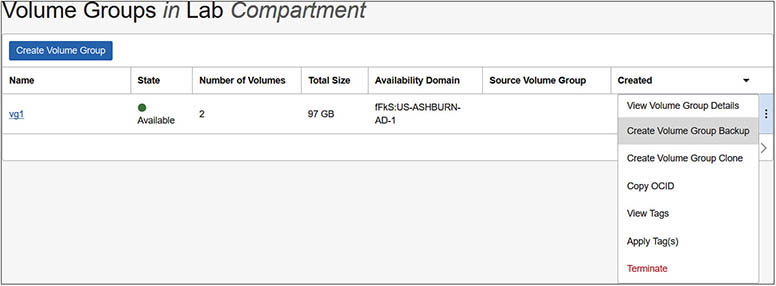
**Exercise 5-7: Create and Back Up a Volume Group**

**1.**   In the console, navigate to Block Storage | Volume Groups, and choose Create Volume Group.

**2.**   Provide a volume group name like vg1, and choose the compartment and AD for the volume group. Add multiple blocks or boot volumes to the volume group. These volumes usually belong to the same instance or have some dependency on one another for data consistency. The bv1 block volume attached to the web1 instance and its boot volume are chosen. Use the +Volume button to add additional volumes to the group, and choose Create Volume Group.

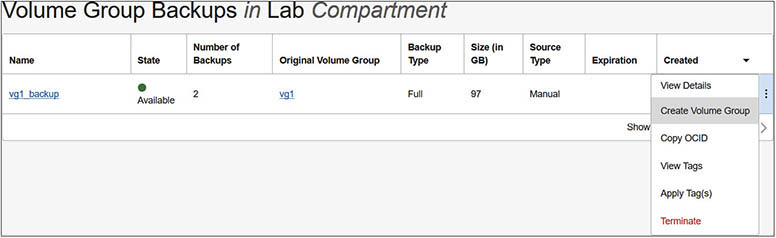


**3.**   Seconds later the volume group detail page should appear. The volume group is a logical grouping of volumes. It has no data but is rather a metadata description of the volumes it comprises. Navigate to Block Storage | Volume Groups and choose the ellipses menu adjacent to the new volume group.



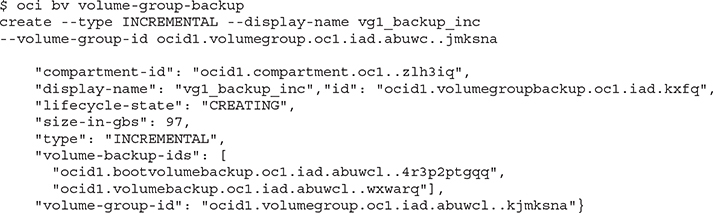
**4.**   You can create a volume group backup or a clone in another compartment. Choose Create Volume Group Backup, specify a volume group backup name like vg1\_backup, and choose Full Backup (the other type is Incremental).

**5.**   Navigate to Block Storage | Volume Group Backups to see the list of volume group backups.



**6.**   This volume group backup is a group of boot and block volume backups. These backups are also accessible through the listings of individual boot and block volume backups with display names like web1 (Boot Volume)\_backup\_20190329\_210244 and bv1\_backup\_20190329\_210244 respectively. Note the identical suffix backup\_20190329\_210244 between them indicating they are part of a volume group backup set.

**7.**   It is likely to be more practical to back up volume groups using scripts with an OCI CLI command like the following:



**8.**   Note that the OCIDs of the boot and block volume are listed in the volume-backup-ids property. You may view the status of the volume group backup by navigating to Block Storage | Volume Group Backups.

**Delete and Recover Block Volumes**

When instances are no longer required, you may reclaim the block storage by detaching block volumes from instances. Detached block volumes may be deleted or terminated. The following OCI CLI command deletes a block volume no longer required.

Images



**NOTE**    You always pay for OCI block storage, whether it is attached to a running or stopped instance. To avoid unnecessary costs, block volumes that are no longer required should be deleted.

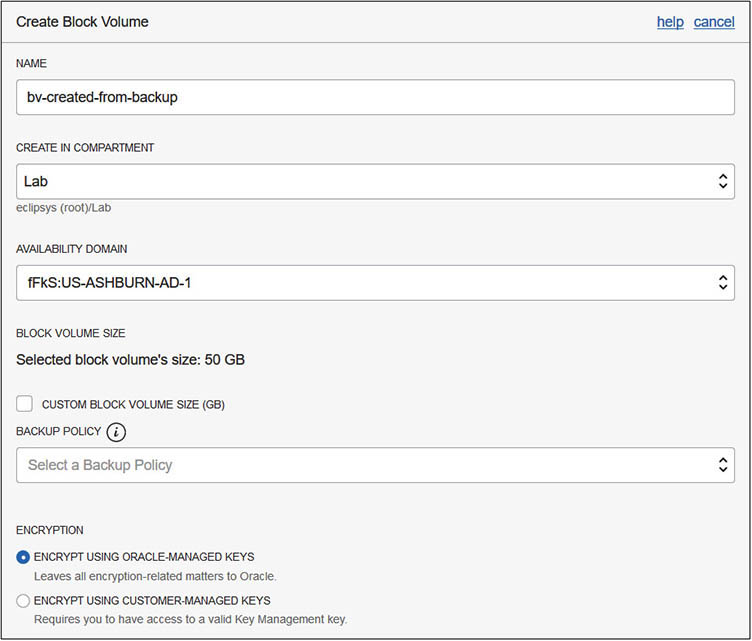
**Block Volume Recovery**

A block volume may be damaged during its lifespan. A logical error such as some data corruption, or data being accidentally removed or overwritten, may have damaged the volume. If a backup is available, it may be restored to a new block volume.

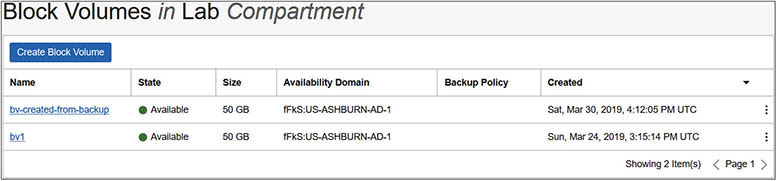
Remember this new block volume has the data as of the time the last backup was taken. You can attach this new block volume to an instance to recover lost data or use it as a replacement volume if the original volume must be replaced. In this case, detach the original volume from its instance and attach the new volume created from the backup.

**Exercise 5-8: Restore a Block Volume Backup to a New Block Volume**

**1.**   In the console, navigate to Block Storage | Block Volume Backups, choose the ellipses menu adjacent to one of your block volumes, and choose Create Block Volume. Provide a name like bv-created-from-backup, a compartment and AD for the new block volume, a backup policy if required, and encryption options, and choose Create Block Volume.



**2.**   You can view the newly created block volume restored from the backup in the console by navigating to Block Storage | Block Volumes.



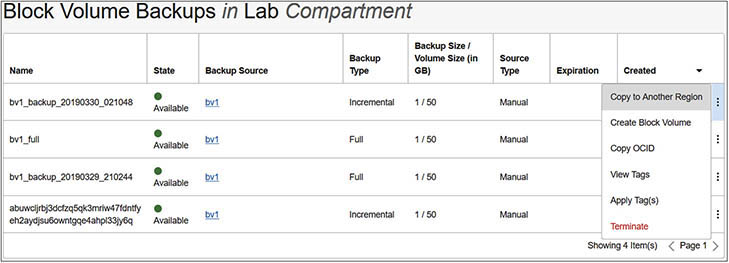
**3.**   This block volume is ready to be attached to an instance or to be cloned and used like a regular block volume.

**Copy Block Volume Backup to Another Region**

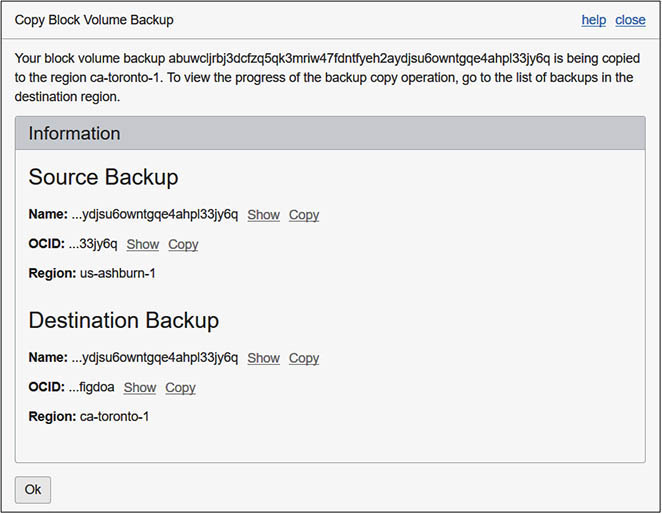
Volume backups may be copied to other regions. This may be useful to clone environments in other OCI regions.

**Exercise 5-9: Recover a Block Volume Backup in a Different Region**

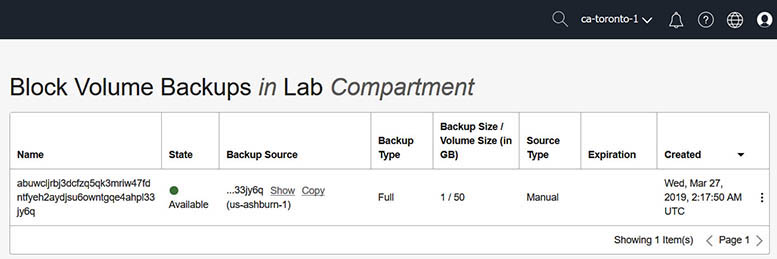
**1.**   In the console, navigate to Block Storage | Block Volume Backups, choose the ellipses menu adjacent to one of your block volumes, and choose Copy To Another Region.



**2.**   Provide a destination region and choose Copy Block Volume Backup. Your tenancy must be subscribed to at least two regions before you can choose a destination region.



**3.**   Once the backup has been copied, you can change your region and navigate to Block Storage | Block Volume Backups to locate the foreign backup. Note that the Backup Source column describes the source region.

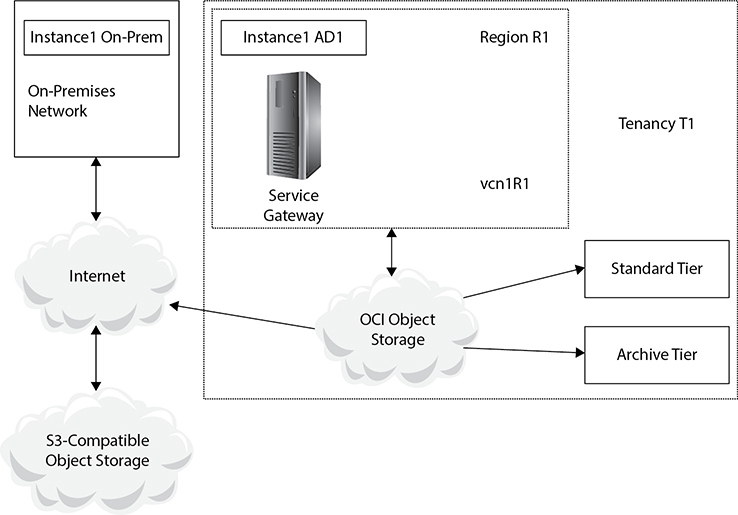


**4.**   You may create block volumes in the new region from this backup as in the previous exercise.

**Object Storage**

Object storage is a relatively new resilient storage type that has become a standard for general purpose file storage in the cloud. The object storage system is Internet accessible, and you control the permissions and whether a bucket is publicly accessible or not. OCI object storage integrates with OCI’s Identity and Access Management (IAM) to control permissions on object storage.

Object storage is not suitable for high-speed computing storage requirements (such as those required to run databases) but provides flexible and scalable options for unstructured data storage and sharing as well as being great for big data and content repositories, backups and archives, log data, and other large datasets. Object storage is also not bound to an instance or an AD but is a region-level construct that resides in a compartment. [Figure 5-5](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig5) situates OCI object storage within a tenancy. Instances in your tenancy may read and write to object storage through a service gateway. Instances on-premises connect either through the VCN (if this connection is set up) or through the Internet to object storage if sufficient permissions have been granted. Both OCI and on-premises instances connect to non–OCI S3–compatible object storage through the Internet.



**Figure 5-5**   Object storage location within a tenancy

**Buckets and Objects**

A bucket is a logical container for objects that reside in a compartment. As the name suggests, you store objects of any data type in a bucket. You may create up to 1,000 buckets per compartment per region and store an unlimited number of objects in a bucket, as of this writing. Buckets may not be nested. This is different from traditional file systems. A bucket may not contain other buckets. A single uneditable namespace is provided to a tenancy that serves as the top-level root container for all buckets and objects. This is a system-generated string and may be queried with this CLI command:

$ oci os ns get

For some older tenancies, the namespace string may be a lowercase version of the tenancy name, but this is now a system-generated string.



**EXAM TIP**    Bucket names must be unique within a namespace. The same bucket name may be used in a separate tenancy, unlike several other mainstream cloud object storage vendors. Bucket names are case sensitive, may not be longer than 256 characters, and may only contain letters, numbers, hyphens, underscores, and periods.

A bucket may exist at one of two tiers:

•   **Standard tier**   Objects stored in a standard tier bucket may be accessed frequently, and your data is immediately available. This tier of storage has good performance but is more expensive than archive tier storage.

•   **Archive tier**   Objects that are infrequently accessed but that must be retained and preserved for a long time are well suited to this tier. There is a longer lead time to access objects in archive tier buckets than in standard tier buckets.

Objects in buckets are encrypted automatically using either keys from your key management system or with OCI-provided keys. Object storage is also replicated across multiple storage servers in a region providing high data durability.

Object storage may be accessed in several ways including the following:

•   **OCI console**   This is a simple browser-based interface well suited to working with a relatively small number of object storage artifacts.

•   **CLI**   This is a fully functional command-line interface that is easily integrated into scripts and well suited for working with a relatively large number of object storage artifacts. Most API calls are wrapped by the CLI simplifying the interaction with object storage without the need for programming.

•   **REST APIs and SDKs**   These offer the most functionality but require some programming expertise.

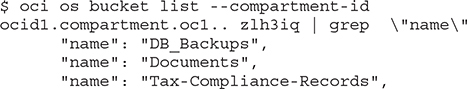
When objects are placed in a bucket, they are uploaded. When they are retrieved, they are downloaded. You may upload and download objects to a bucket using the mechanisms listed previously. If a bucket is designated as a public bucket or if a bucket or object is shared with a pre-authenticated request, users may interact with object storage, performing reads and writes based on the sharing permissions, using HTTPS-based tools including web browsers and the curl and wget utilities. These mechanisms are discussed later in the chapter.

When an object is downloaded, the most recently written copy of the object is served by the object storage service, ensuring strong data consistency.



**EXAM TIP**    An object storage bucket can exist in only one compartment but can also be moved between compartments.

Using the following CLI command, you may list the bucket names in your compartment:



**Archive Tier Buckets**

Objects in standard tier buckets may be downloaded immediately, whereas objects in archive tier buckets must be restored before they are downloadable. The restore time can take several hours so it is important to store appropriate data in archive tier buckets.

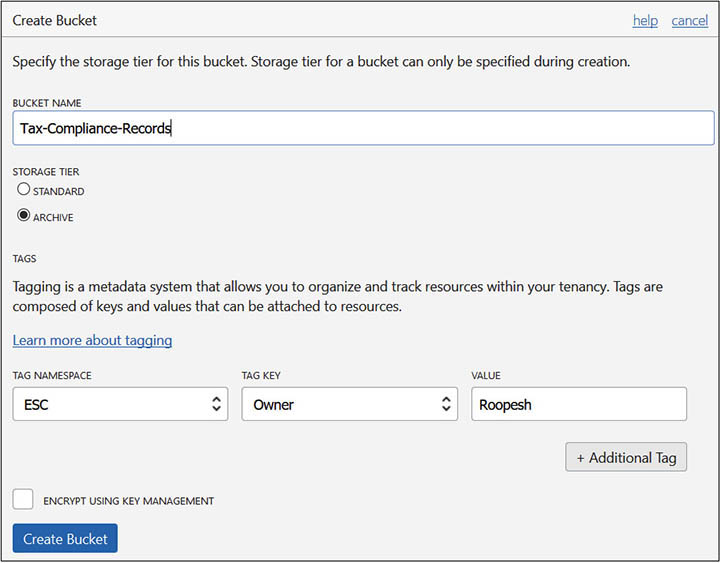
**Exercise 5-10: Upload, Restore, and Download Using an Archive Tier Bucket**

In this exercise, you will create an archive tier bucket named Tax-Compliance-Records, upload a file, restore the object, and download it to your local machine.

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

**2.**   Navigate to Object Storage | Object Storage, and choose Create Bucket.

**3.**   Name the bucket Tax-Compliance-Records, choose the archive storage tier, provide appropriate tag and encryption information, and choose Create Bucket.



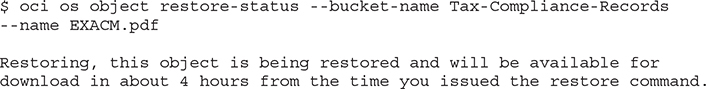
**4.**   Navigate to the Tax-Compliance-Records bucket and choose Upload Object. Browse your local machine and choose a file less than 2GB in size. The file upload process begins, and after a short while, your file is stored in archive storage. You can use the console, API, or the following CLI command to list the file uploaded into this bucket as follows:

Images

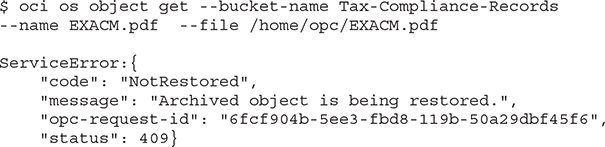
**5.**   To retrieve the file uploaded to this archive tier bucket, you must first initiate an object restore request. Using the console, navigate to the bucket and choose Restore Object from the ellipses menu adjacent to the object name. You are asked to optionally specify the time available to download the object once it is restored. It is available for download by default for 24 hours, but you may specify a download availability time from 1 to 240 hours. The following CLI command requests a restore of an object named EXACM.pdf from the Tax-Compliance-Records bucket.

Images

**6.**   You may check the progress of the restore request using the CLI as follows:



**7.**   Once the restore is complete, you may download the object. If you try to download the object using the CLI before the restore is completed, you will receive an error similar to the following:





**NOTE**    The object storage service uses the 134.70.0.0/17 CIDR block IP range for all regions.

**Standard Tier Buckets**

Standard tier buckets are created by default and are often referred to as object storage while archive tier buckets are often known as archive storage. These are general purpose and support fast uploads and immediate downloads with no delays as you wait for data to be restored. Oracle is continuously improving the underlying physical storage infrastructure supporting the object storage service to provide a high performing storage option at a lower cost than block storage. Standard tier buckets are resilient and durable and are a good option for storing recent backups, big data repositories, images, videos, log files, and other content. OCI provides an HDFS connector enabling Apache Spark and Hadoop jobs to run against data in the OCI object storage service. Internet of Things (IoT) data and other large application data sets are good candidates for standard tier buckets.

**Exercise 5-11: Upload, Restore, and Download Using a Standard Tier Bucket**

In this exercise, you will create a standard tier bucket named Documents, upload a file, and download it to your local machine.

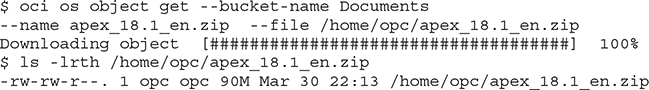
**1.**   Sign in to the OCI console and choose your compartment.

**2.**   Navigate to Object Storage | Object Storage, and choose Create Bucket.

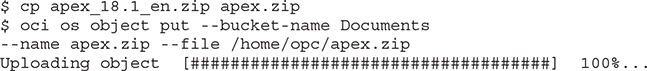
**3.**   Name the bucket Documents, choose the standard storage tier, provide appropriate tag and encryption information, and choose Create Bucket.

**4.**   Once the bucket is created choose Upload Object on the Document bucket details page, browse your local machine, and choose a file less than 2GB in size. The file upload process begins, and after a short while, your file is stored in an object storage bucket.

**5.**   To retrieve the file uploaded to this standard tier bucket using the console, navigate to the bucket and choose Download from the ellipses menu adjacent to the object name. You are prompted to choose a destination location and filename on your local machine for this object. The following CLI command requests a download of an object named apex\_18.1\_en.zip from the Documents bucket to a file named /home/opc/ apex\_18.1\_en.zip.

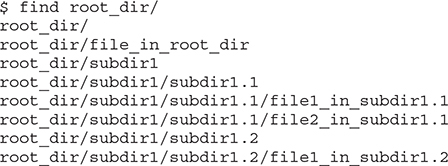


**6.**   In the following example, the downloaded file is renamed apex.zip and uploaded to this bucket using the CLI:



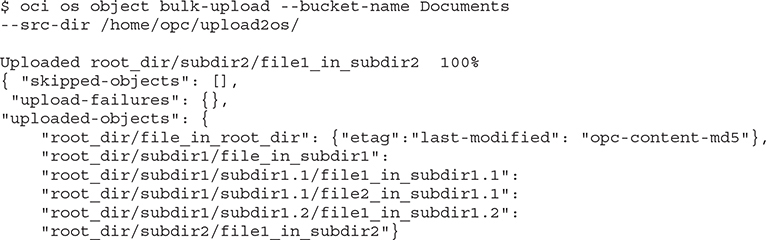
**Pseudo-Hierarchies in Object Storage**

The flat structure in a bucket may simulate a traditional hierarchical file system by naming objects with a trailing slash (/). Consider the following hierarchical structure on a typical Linux system:

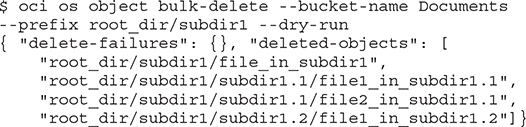


Images

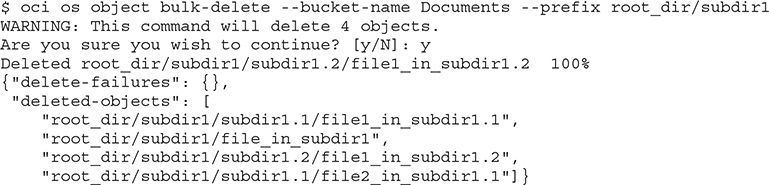
The entire directory may be uploaded to an object storage bucket using the CLI bulk-upload command. The etag, last-modified, and opc-content-md5 attributes have been omitted for brevity.



These objects are actually all files in the same flat bucket with names that make them appear like a traditional hierarchical directory. To manage these as if they were hierarchical, you can use matching patterns. For example, to delete all the objects with names beginning with root\_dir/subdir1, you may use the CLI bulk-delete command with the prefix option. This command also allows you to perform a dry-run to ensure you delete what you mean to delete.

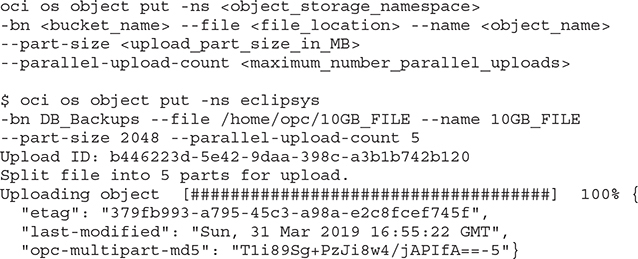


Run the command without the dry-run option to remove these objects.

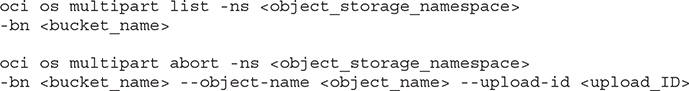


**Multipart Uploads for Large Objects**

Objects may be uploaded to buckets using the console, but there is a 2GB limit per object. The CLI, SDKs, or API may be used to upload larger objects up to 10TB by performing a multipart upload and parallelizing the upload to reduce the overall upload time. Using the API, you are required to split the object into multiple parts, upload the parts, and commit the upload, which allows the object storage service to reconstruct the large object from its constituent parts. When using the CLI, you are not required to split the object into parts manually as the splitting, upload, and commit are done automatically by the utility. You can choose the size of the component parts and the maximum number of parts to be uploaded in parallel (the default is three). The following CLI command allows you to initiate the multipart upload. A worked example shows a 10GB file being split into 2GB parts and uploaded using five parallel threads.



To list the progress of the upload and to abort the upload, you may use the following CLI commands:



**Pre-Authenticated Requests**

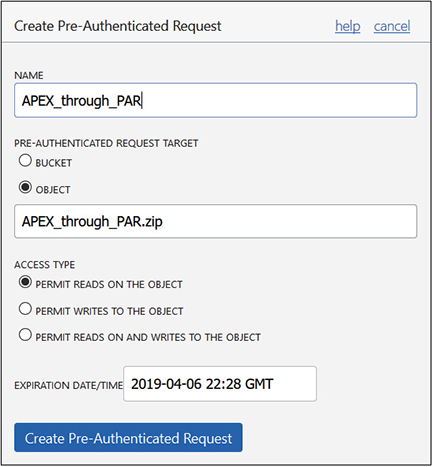
OCI provides several options to share objects or buckets. You can designate a bucket’s visibility as public, which allows anyone to access your bucket without requiring authentication. You should use this option cautiously and carefully evaluate whether you need to make a bucket publicly visible. A safer option is to set up a pre-authenticated request (PAR) that exposes a bucket or an object for a limited time.

A pre-authenticated request may be created for a bucket or an object as follows:

•   **Bucket**   PARs permit writes.

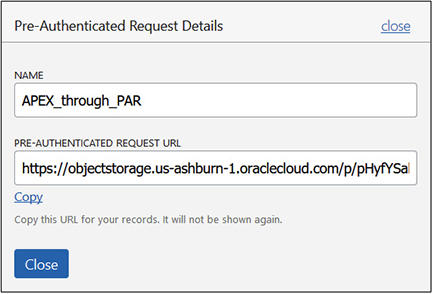
•   **Object**   PARs permit either reads or writes or both reads and writes.

Using the console to create PARs illustrates the options, as shown in [Figure 5-6](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig6). Navigate to the object storage bucket list and choose Create Pre-Authenticated Request from the ellipsis menu adjacent to a bucket or object. Provide a name for the PAR and choose whether you wish to expose a bucket or a specific object, the access type, and the expiration date and time after which the PAR is no longer valid.



**Figure 5-6**   Create pre-authenticated request on an object

Once the PAR is created, a request URL is provided, as shown in [Figure 5-7](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig7). This URL must be saved as it is not shown again by the console. You may use commands such as wget and curl on Linux to access the bucket or objects exposed through the PAR. The following command reads the apex.zip file through the PAR with no authentication required:



**Figure 5-7**   Pre-authenticated request details

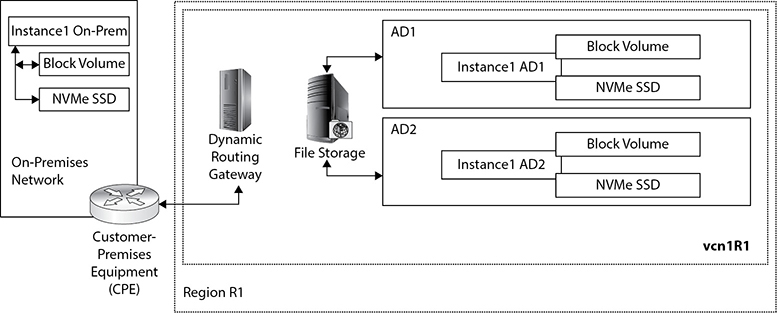
Images

The following command writes a file to the object storage bucket using a PAR:

Images

**File Storage Service**

The file storage service (FSS) is a network-based storage system that allows multiple instances to mount a shared file system. Many applications such as Oracle EBS (E-Business Suite) require a shared file system and many organizations use NFS (network file system) mountpoints to share and store data on a remote file system. The FSS is a regional service available to instances in all ADs in a region. [Figure 5-8](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig8) situates the file storage service in a VCN in a region. Instances in your on-premises network may be permitted to use the file storage service file systems and are also known as NFS clients.

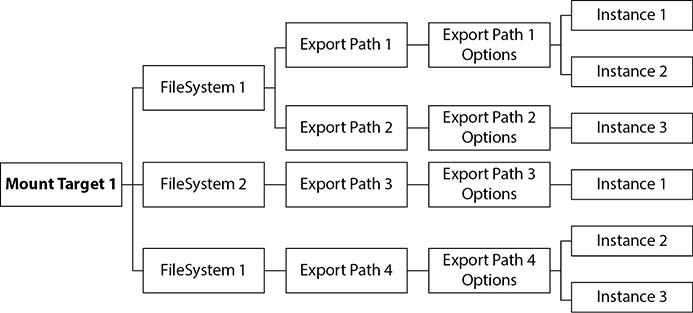


**Figure 5-8**   File storage as a regional service

FSS provides NFSv3–compatible file systems supporting full POSIX semantics similar to NFS file systems that have been available on traditional networks for decades. If you are familiar with NFS, then many of the concepts discussed in the following section will be familiar. There are several nuances to be aware of, however, so this is well worth the read.

**FSS Concepts**

FSS provides network-based file systems in a region. These file systems are physically located on storage servers in an AD and are replicated to other ADs or fault domains providing high durability. [Figure 5-9](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig9) expresses the relationship between three FSS concepts and instances in your region.



**Figure 5-9**   Relationships between mount targets, file systems, exports, and instances

A mount target is an NFS endpoint that resides in a subnet in an AD or region and is given three IP addresses from that subnet by the file storage service. The mount target provides network access for file systems.



**CAUTION**    Locate the mount target in a subnet that has a CIDR range larger than /30 as each mount target requires three private IP addresses. NFS clients connect to the mount target. To avoid potential IP conflicts, it is good practice to place mount targets and subnets into dedicated subnets.

A file system is the primary resource for storing files in FSS. A file system is AD-specific and could be accessed via multiple export paths. Each export path is associated with a set of export options that determine which instances or NFS clients have access to the export path. Export options may specify the allowed source IP addresses and ports, whether access is read/write or read-only, and how Unix-style user and group access rights are reduced or squashed for the mounted network file system.

Multiple file systems may use the same mount target. The default soft limit is 100 file systems per mount target. Multiple export paths may be created for each file system to configure different export options for different NFS clients, on condition that the export paths in each file system are unique and non-overlapping. In [Figure 5-9](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig9), both FileSystem 1 and FileSystem 2 use the same mount target. FileSystem 1 exports three paths to the mount target. This is allowed on condition that these are unique and non-overlapping paths. For example, the export paths /exp1 and /exp1/p1 are overlapping and are not allowed to be exported to the same mount target.

Several network-related prerequisites must be met before using the file storage service. The security list associated with the subnet that contains the mount target must allow ingress TCP and UDP traffic on ports 111 and ports 2048, 2049, and 2050. [Figure 5-10](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig10) shows an example of four stateful ingress rules that allow traffic from any source IP address (0.0.0.0/0). You may wish to restrict the source range to specific CIDR ranges or even IP addresses, although it may become tedious to manage these virtual firewall rules at an IP address level for individual NFS clients. In a multitenant environment where multiple NFS clients access file systems through the same mount target, you can limit a client’s ability to connect to the file system and view or write data using export path options.



**Figure 5-10**   Stateful ingress security list rules required for mount target subnet

There are four layers of security to limit access to FSS:

•   **IAM service**   This uses OCI users, groups, and policies to permit OCI users to create and manage infrastructure for FSS, including creating instances (NFS clients), VCNs, and subnets; updating security lists; and creating OCI mount targets and file system objects.

•   **Security list**   This uses CIDR ranges to limit which instances can connect to the mount target.

•   **Export options**   This uses IP addresses, CIDR block ranges, access permissions, and root squash options to control access on a per-file system basis.

•   **NFSv3 Unix security**   This controls which Unix users can mount file systems and update or view files on the FSS file system.



**CAUTION**    Typical NFS mount options you may be used to may not be appropriate for mounting FSS file systems. Avoid specifying mount options such as nolock, rsize, or wsize when mounting FSS file systems to avoid performance and file locking issues.

**Create, Configure, and Mount a File Storage Service**

It is about time to get hands-on with the file storage service. In the upcoming exercise, you will complete the following steps:

•   Set up security list prerequisites. Set up stateful ingress rules, opening TCP and UDP ports 111 and 2048–2050, and egress rules allowing all traffic out in the security list used by the subnet where the mount target resides.

•   Create a file system and associate it with a new or existing mount target. If this is a new mount target, you choose the AD and subnet where this mount target resides.

•   Connect to an NFS client, mount the exported file system, and confirm it is mounted with read/write access.

**Exercise 5-12: Create a File System, Mount Target, and Mount with NFS Clients**

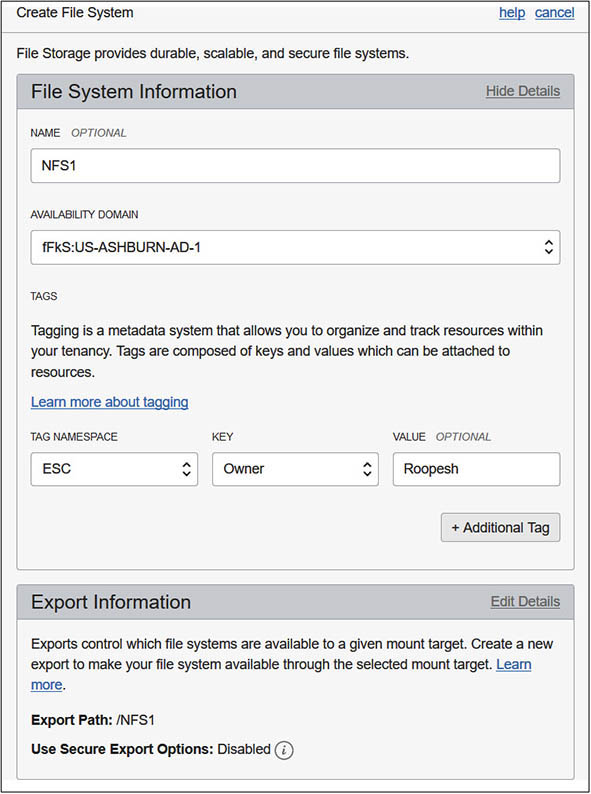
In this exercise, you will first set up the networking prerequisites that FSS depends on to expose the mount target to instances in your network. The exercise uses the virtual cloud network named vcn1R1, and a public subnet named PS1 in AD1 in the Ashburn region. Once the FSS artifacts are created, the web1 and web2 instances created in [Chapter 4](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch4.xhtml#ch4) are used to mount the shared network file system.

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

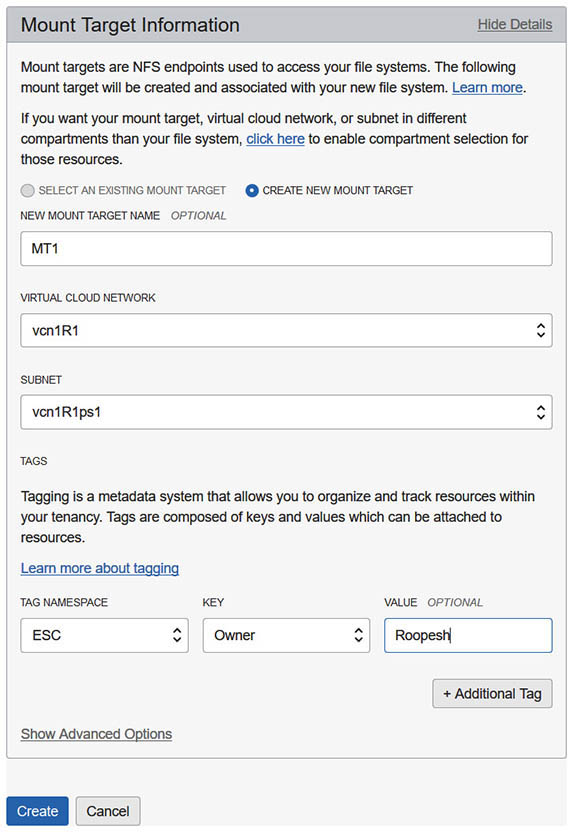
**2.**   Navigate to Networking | Virtual Cloud Networks, and choose vcn1R1 or a suitable VCN. Choose a subnet for your mount target and choose its security list. In this exercise, the subnet vcn1R1ps1 is chosen. It is a public subnet. You can choose a private subnet for the mount target; in fact, it’s good practice to do so, but additional configuration is required in that case.

**3.**   Edit the security list rules, adding four stateful ingress rules opening TCP and UDP destination ports 111 and 2048–2050, and an egress rule allowing all traffic out (if none exists). You can set the source CIDR range to limit the NFS client IP range, if desired. If you set the source CIDR range to 0.0.0.0/0, your ingress rules should resemble those listed in [Figure 5-10](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5fig10).

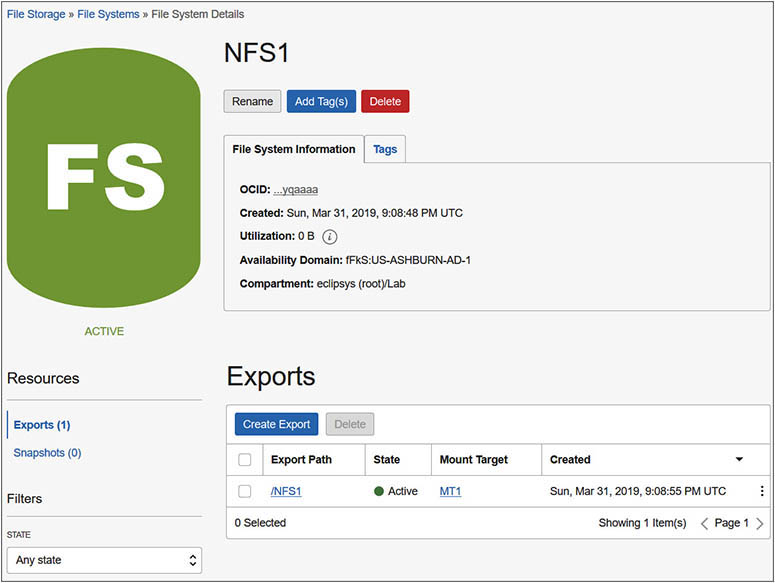
**4.**   Navigate to File Storage | File Systems and choose Create File System. Default file system, export information, and mount target information are provided. Choose Edit Details and provide a name for the file system, like NFS1. Choose an AD for the file system. The export path in the export information section defaults to /NFS1.



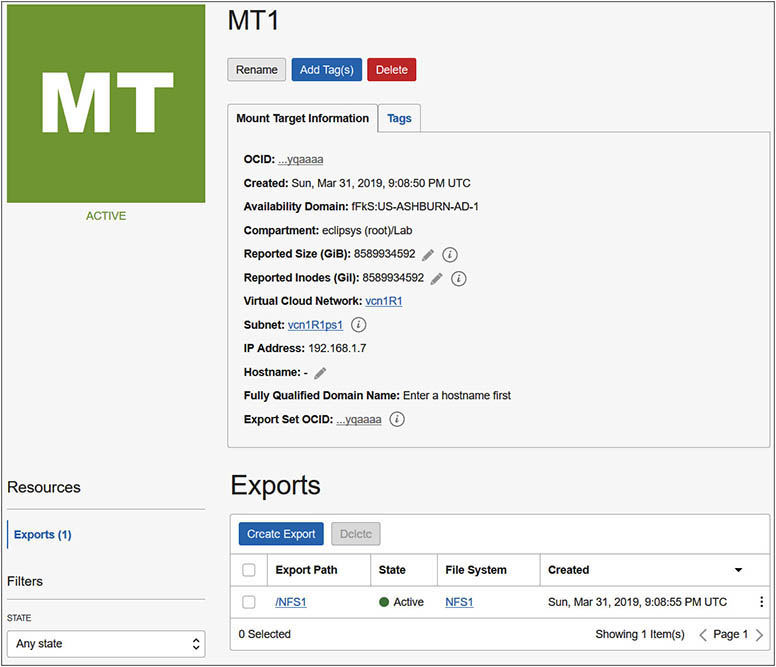
**5.**   In the mount target section, you may choose to associate the file system you are about to create with an existing mount target or create a new mount target. Choose Create New Mount Target and provide a name, like MT1. Choose a VCN, like vcn1R1, and a subnet, like vcn1R1ps1, and choose Create.



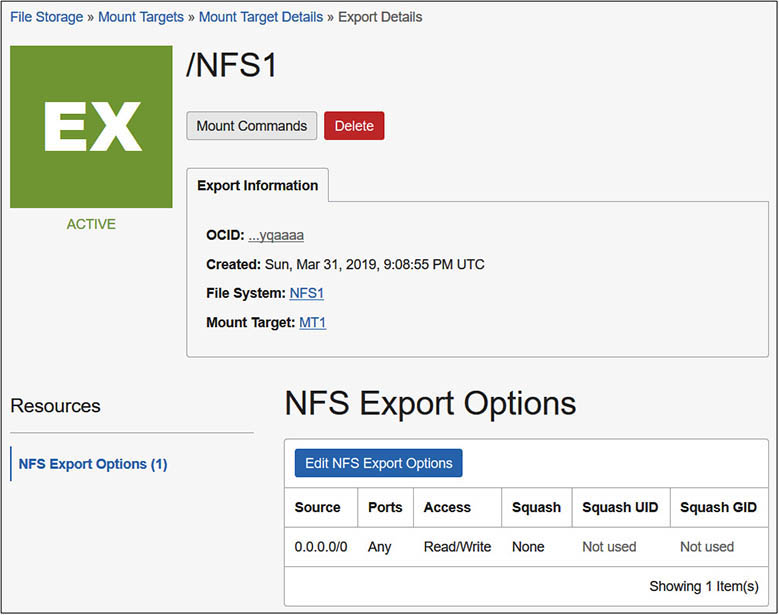
**6.**   Once the file system is created, examine the details page listing the OCID, AD, compartment, and utilization data. In the Exports section, note there is an active export path named /NFS1 associated with mount target MT1. You may add additional export paths to this file system as long as the export paths are unique and non-overlapping.



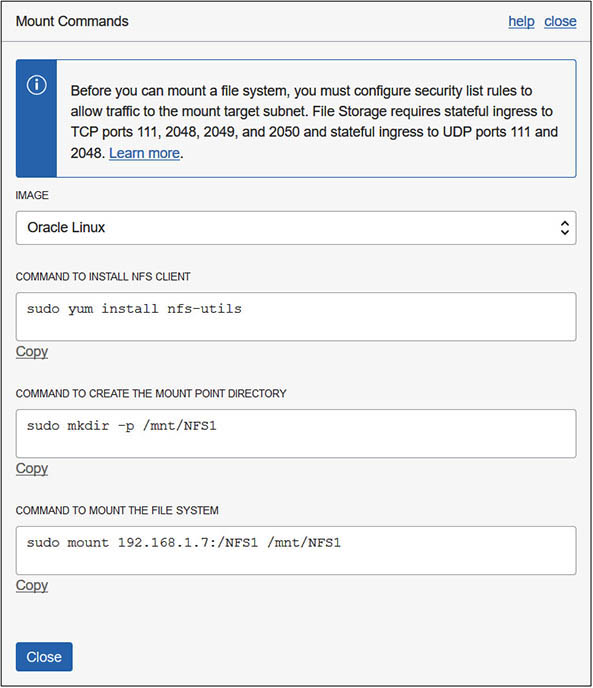
**7.**   Choose mount target MT1 and note the OCID, AD, compartment, reported size, and number of inodes in gibibytes, which is a decimal unit closely related to gigabyte. The subnet and IP address of the mount target is also listed, along with a facility to set a hostname. You may set the hostname and the fully qualified domain name (FQDN) field is updated, or you can use the IP address to mount the target, as discussed in subsequent steps.



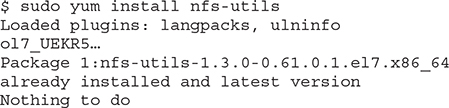
**8.**   Choose the /NFS1 export path to view the default NFS export options. The default options allow any instances (0.0.0.0/0) mounting this file system through this mount target to access the file system with read/write privilege. You may edit the NFS export options to restrict the CIDR range for NFS clients and to configure whether privileged source ports (1–1023) are required, whether the access is read-only or read/write, and whether identity squash or remapping of user IDs (UIDs) and group IDs (GIDs) to anonymous IDs is required.



**9.**   Navigate to the File Storage | Mount Targets page and choose MT1. Choose the ellipses menu adjacent to the /NFS1 export path in the Exports section and choose mount commands. Choose the image used to create your NFS client. Choose the appropriate OS image for your NFS client to get the mount commands. In this exercise, the web1 instance is designated as the NFS client and it uses Oracle Linux. Steps 10–12 are based on these mount commands.



**10.**   SSH to the compute instance that is your NFS client, such as the web1 instance, and install the NFS client library by copying the command from the mount command list provided in the console in the previous step. In this case, the nfs-util package is already installed.



**11.**   Make a directory to use as a mount point. You can copy the command from the OCI console if desired.

Images

**12.**   Mount the file system NFS1. You can copy the command from the console if desired. There is no acknowledgment message. If this hangs, it usually indicates a problem with the security list.

Images

**13.**   You may confirm that the FSS file system is mounted with the df command.

Images

**14.**   Create a file on the mount point to confirm that you have read/write permission.

Images

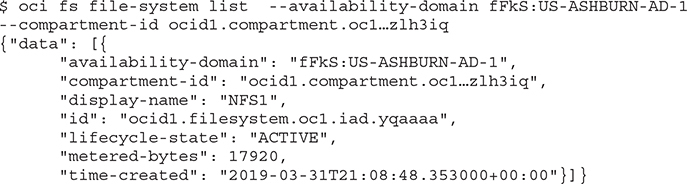


**NOTE**    For optimal performance, it is advisable to place the mount targets in the same AD as the instances using the mount point.

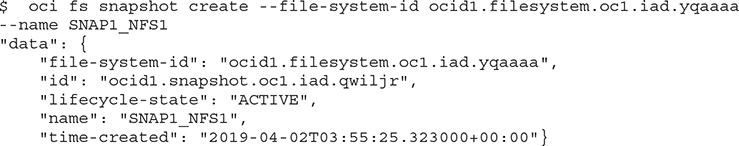
**FSS Snapshots**

FSS offers a convenient snapshot facility that takes a point-in-time backup of an FSS file system. Snapshots are read-only and are located in a hidden directory named .snapshot in the root directory of the FSS file system. Snapshots are incremental and are consequently very space efficient, backing up only files that have changed since the last snapshot. By default, you can take up to 10,000 snapshots per file system.

You can create snapshots through the console, API, or CLI. In this section, the CLI is used. Use the following CLI command to identify the file systems in an AD and compartment. Assuming you completed the previous exercise, you should see your file system listed here.



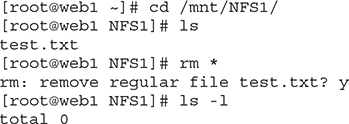
Take note of the file system OCID. The following CLI command creates a snapshot of the NFS1 file system called SNAP1\_NFS1.



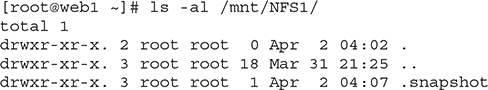
You may use the following command to list the snapshots available for your file system:

Images

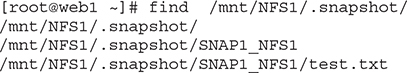
To test the snapshot, delete the /mnt/NFS1/test.txt file created in the previous exercise through the web1 instance.



Check for hidden files in the root directory:



Explore the contents of the snapshot:



Restore the deleted file by copying the backup in the snapshot back to the original location. The rsync command may be used for restoring multiple missing files.

Images

**Chapter Review**

This chapter discussed five OCI storage services, each suitable for different use cases.

Local NVMe SSD storage is temporary and has no durability. You have to ensure redundancy and protect against disk failures by creating RAID sets with adequate mirroring or set up other high-availability mechanisms. These are, however, the fastest storage available to bare-metal and VM instances as they are directly attached. Local NVMe SSD storage is suitable for high-performance workloads, including transactional databases.

Block storage provides boot volumes and block volumes and is most akin to classical on-premises SAN storage. Block volumes are durable, with multiple data block copies being made across multiple ADs or fault domains. Block volumes scale to petabytes of capacity and are well suited for databases and general-purpose storage offering good IO performance.

Object storage is exposed through buckets at either an archive tier or a standard tier. Archive tier buckets are often just referred to as archive storage while standard tier buckets are frequently referred to as object storage. Archive storage is suitable for long-term data retention and is highly durable and affordable. However, restoring data from archive storage can be a lengthy process requiring a restore operation before the data may be downloaded.

Standard tier buckets are the default object storage option and are suitable for most types of data, including backups, logfiles, photos, videos, and even big data and content repositories. Object storage scales to petabytes of capacity and is an attractive option for offloading legacy data sets from more expensive storage.

A significant portion of this chapter is dedicated to hands-on practical exercises enabling you to create and manage block storage, object storage, and file storage services.

The chapter closed with a discussion of the file storage service, an NFSv3-compatible shared network storage offering. FSS is durable and scales to exabytes of capacity and is critical for systems and applications that require a shared file system. FSS snapshots offer a convenient and simple option for incrementally backing up and restoring your FSS file systems.

With your network, compute instances, and storage options in place, you are ready to explore Oracle database options available in OCI, which is the subject of [Chapter 6](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch6.xhtml#ch6).

**Questions**

[**1**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans1)**.**   List the storage types available for provisioning on OCI VMs.

**A.**   Block storage

**B.**   NVMe SSD

**C.**   Object storage

**D.**   Flash storage

[**2**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans2)**.**   Which of the following are types of block storage?

**A.**   NVMe SSD

**B.**   Boot volumes

**C.**   Block volumes

**D.**   Object storage

[**3**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans3)**.**   When an instance starts up, what are the possible access options for attaching the boot volume?

**A.**   Read-only

**B.**   Read/write

**C.**   Copy-on-write (COW)

**D.**   Write-only

[**4**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans4)**.**   What are the different storage tiers available for buckets in object storage?

**A.**   Gold

**B.**   Archive

**C.**   Silver

**D.**   Standard

**E.**   Bronze

[**5**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans5)**.**   Which of the following statements are true?

**A.**   OCI bucket names must be unique within a namespace.

**B.**   OCI bucket names must be unique across all tenancies.

**C.**   OCI bucket names are not case-sensitive.

**D.**   OCI bucket names are case-sensitive.

[**6**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans6)**.**   When provisioning or configuring a block volume, you may specify which categories of backup policies?

**A.**   Gold

**B.**   Archive

**C.**   Silver

**D.**   Standard

**E.**   Bronze

[**7**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans7)**.**   Which of the following statements is true?

**A.**   An object storage bucket can only exist in one compartment.

**B.**   An object storage bucket can exist in multiple compartments.

**C.**   An object storage bucket is an AD-level resource.

**D.**   An object storage bucket is a regional-level resource.

[**8**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans8)**.**   Which of the following statements are true?

**A.**   Object storage has a flat structure.

**B.**   Object storage has a hierarchical structure.

**C.**   Multipart uploads can only be done for standard tier buckets.

**D.**   Multipart uploads are possible for all types of object storage.

[**9**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans9)**.**   File storage service snapshots are useful for making file system backups. What type of backup is taken with an FSS snapshot?

**A.**   FULL

**B.**   ROLLING

**C.**   INCREMENTAL

**D.**   CLONE

**E.**   NFSv3

[**10**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#ch5ans10)**.**   An important production system with a boot volume and two block volumes must be moved from the Ashburn (IAD) region to the Toronto (YYZ) region. Choose which options are feasible?

**A.**   Copy block storage to FSS file systems and mount on a new instance in Toronto.

**B.**   Copy a snapshot to the Toronto region and mount on a new YYZ instance.

**B.**   Use pre-authenticated requests to move the data without complex authentication.

**C.**   Create a volume group backup of the boot and block volumes, copy each of these volume backups to the YYZ region, and mount on a new YYZ instance.

**Answers**

[**1**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans1)**.**   **A, B, C.** Block storage, NVMe SSD, and object storage, as well as file storage services, may be provisioned on OCI.

[**2**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans2)**.**   **B, C.** Boot and block volumes are types of block storage.

[**3**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans3)**.**   **B.** Boot volumes are always attached with read/write access.

[**4**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans4)**.**   **B, D.** Object storage buckets are available at the standard and archive tiers.

[**5**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans5)**.**   **A, D.** OCI bucket names must be unique within a namespace and are case-sensitive.

[**6**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans6)**.**   **A, C, E.** Bronze, Silver, and Gold level backup policies may be configured for a block volume.

[**7**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans7)**.**   **A, D.** An object storage bucket can exist in only one compartment and is a regional-level resource.

[**8**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans8)**.**   **A, D.** Object storage has a flat structure, and multipart uploads are possible for all types of object storage.

[**9**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans9)**.**   **C.** A snapshot makes an incremental backup of an FSS file system.

[**10**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch5.xhtml#r_ch5ans10)**.**   **D.** The instance may be relocated by creating a consistent volume group backup of the boot and block volumes. These volume backups may be copied to the YYZ region and mounted on a new YYZ instance.