

Amazon Elastic Block Store (EBS)

Overview

This lab introduces you to Amazon Elastic Block Store (EBS) using the AWS Management Console.

Amazon EBS offers persistent storage for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon EBS volumes are network-attached and persist independently from the life of an instance. Amazon EBS volumes are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instance boot partition or attached to a running Amazon EC2 instance as a standard block device.

When used as a boot partition, Amazon EC2 instances can be stopped and subsequently restarted, enabling you to pay only for the provisioned storage resources while maintaining the instance's state. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated within the Availability Zone to prevent data loss due to failure of a single piece of hardware.

For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon Simple Storage Service (Amazon S3) and automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes and can protect your data for long-term durability. You can also easily share these snapshots with co-workers and other AWS developers.

This lab guide explains basic concepts of Amazon EBS in a step-by-step fashion. However, it can only give a brief overview of Amazon EBS concepts. For further information, see the [Amazon EBS documentation](#).

Topics covered

When you finish the lab, you will be able to:

- Create an Amazon EBS volume.
- Attach the EBS volume to an Amazon EC2 instance.
- Create a file system on the EBS volume.
- Create a Snapshot of the EBS volume.
- Restore a snapshot as a new EBS volume.
- Modify an EBS volume.

Pre-Task

- Launch a new Linux EC2 instance with
Key: name
Value: Lab

Task 1: Create an Elastic Block Store volume

In this task, you create a new Amazon EBS volume. Amazon EBS volumes are like hard drives in a computer. The data on them persists through the lifetime of the volume and can be transported between instances as needed.

- In the **AWS Management Console**, on the **Services** menu, choose **EC2**.
- At the top right of the screen, if you see **New EC2 Experience** toggle to use the new UI, if it is not enabled by default
- In the left navigation pane, under **ELASTIC BLOCK STORE**, choose **Volumes**.

Notice the existing volume that has a state of **in-use**. This volume is connected to an Amazon EC2 instance **Lab** that you created above

- Point to the **Name** column for the volume being displayed. A pencil icon appears.
- Choose the pencil icon, change the name of the volume to , and then press **Enter** or choose the checkmark to save your changes.

This adds a name to the EBS Volume. It is recommended that you name and tag your EBS volumes to keep track of their content and purpose.

- Take note of the **Availability Zone** of the volume. You use this value in an upcoming step.
- Choose **Create Volume**

You are presented with the **Create Volume** dialog box, which contains various configuration options:

- **Volume Type** specifies whether to use magnetic disks or SSD. Each volume type has its own capabilities that can meet specific use-cases.
- **Size** can be up to 16TB.
- **IOPS** defines the speed of the disk in *Inputs/Outputs per Second*.
- **Availability Zone** identifies where to create the Volume, which should be in the same AZ as the instance that will use the volume.
- **Snapshot ID** allows a previous snapshot to be restored onto the new volume.
- **Encryption** chooses whether the contents of the volume should be automatically encrypted when stored on disk.
- In the **Create Volume** dialog box, configure:
 - **Volume Type**: General Purpose SSD (GP2)
 - **Size (GiB)** :
 - **Availability Zone**: Use the Availability Zone noted in the earlier step (same as the AZ of the instance)
- Choose **Add Tag** and then configure:
 - **Key**:
 - **Value**:
- Choose **Create Volume**

A **Volume created successfully** message displays.

- Choose **Close**

Your new EBS volume appears in the volume list. Wait for the newly created volume to change to a **State** of *available*.

If your newly created volume shows the **State** as *creating* or does not appear, choose the refresh icon at the top-right corner of the page.

Congratulations! You have successfully created a new Amazon EBS volume.

You can now attach the volume to an Amazon EC2 instance, which allows the instance to use the volume. Next, you select the Amazon EC2 instance that you want to use the volume.

Task 2: Attach the EBS volume to an Amazon EC2 instance

In this task, you attach the new EBS volume to a Linux-based Amazon EC2 instance.

- Select your volume named *My volume*.
- In the **Actions** drop-down list, choose **Attach Volume**.
- Choose the **Instance** field and then select the *running* instance that appears. It should be named *Lab*.
- Make a note of the path in the **Device** field. The default should be */dev/sdf*. You use it in a future task when formatting and mounting the volume.
- Choose **Attach**

Notice the **Status** of the volume changes to *in-use*.

Next, let's examine the two volumes that are now attached to the EC2 instance.

- In the left navigation pane, under **INSTANCES**, choose **Instances**.
- Select the **Lab** instance from the list.
- Click **Storage** tab and scroll down to the **Block devices** section.
- Notice *Delete on termination* is *Yes* for the device name */dev/xvda*, which means that this Amazon EBS volume will be permanently deleted if the Amazon EC2 instance is terminated.
- Notice *Delete on termination* is *No* for the device name */dev/sdf*, which means that this Amazon EBS volume will persist after the Amazon EC2 instance is terminated. The volume could then be attached to a different instance at a later time. Volumes that are not deleted with the termination of an instance can be deleted from the Volumes page.

Congratulations! You have successfully attached an Amazon EBS volume to an instance.

Task 3: Create and configure your file system

In this task, you create an ext3 file system on the EBS volume under the /mnt/data-store mount point.

- Connect to your instance to proceed
- In the EC2 Instance Connect session, enter the following command to view the storage available on your instance:

```
df -h
```

You should see output similar to:

```
[ec2-user@ip-10-10-1-61 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        462M   0    462M   0% /dev
tmpfs           479M   0    479M   0% /dev/shm
tmpfs           479M 392K   479M   1% /run
tmpfs           479M   0    479M   0% /sys/fs/cgroup
/dev/nvme0n1p1  8.0G  1.1G   7.0G  14% /
tmpfs           96M   0     96M   0% /run/user/1000
```

The command output shows the 8 GiB disk volume that was created with the Amazon EC2 instance. Your new Amazon EBS volume is not yet shown.

- Enter the following command to create an ext3 file system on the new Amazon EBS volume. Replace **/dev/sdf** with the value of **Device** from the previous task, if it's different.

```
sudo mkfs -t ext3 /dev/sdf
```

The output should look similar to:

```
[ec2-user@ip-10-10-1-61 ~]$ sudo mkfs -t ext3 /dev/sdf
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

- Enter the following command to create a new directory for mounting the new storage volume:

```
sudo mkdir /mnt/data-store
```

- Enter the following command to mount the new storage volume:

```
sudo mount /dev/sdf /mnt/data-store
```

To configure the Linux instance to mount this volume whenever the instance is started, you must add modify the `/etc/fstab` configuration file.

- Enter the following command to add the appropriate line to the end of `/etc/fstab`:

```
echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab
```

- Enter the following command to view the `/etc/fstab` configuration file:

```
cat /etc/fstab
```

Verify the line you entered in the previous command is shown at the end of the file.

- Enter the following command to view the available storage again:

```
df -h
```

The output now contains an additional line, `/dev/nvme1n1`, which is the EBS volume you created. Notice that the **Mounted on** path is the folder you created previously.

```
[ec2-user@ip-10-10-1-61 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        462M   0    462M   0% /dev
tmpfs           479M   0    479M   0% /dev/shm
tmpfs           479M 392K   479M   1% /run
tmpfs           479M   0    479M   0% /sys/fs/cgroup
/dev/nvme0n1p1  8.0G  1.1G   7.0G  14% /
tmpfs           96M   0     96M   0% /run/user/1000
/dev/nvme1n1     9.8G  23M   9.2G   1% /mnt/data-store
```

- Enter the following command to create a new text file, containing some sample text, on the newly mounted volume:

```
sudo sh -c "This is a test data > /mnt/data-store/file.txt"
```

- Enter the following command to view the contents of the text file you just created and verify that the text has been written to your volume.

```
cat /mnt/data-store/file.txt
```

The output of the command should read:

```
some text has been written
```

Congratulations! You have successfully formatted the volume with a file system and written a file to it.

Task 4: Snapshot the EBS volume

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data. Each snapshot contains all of the information needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

In this task, you will create a snapshot of your EBS Volume.

- Return to your browser tab with the AWS Management Console.
- In the left navigation pane, under **ELASTIC BLOCK STORE**, choose **Volumes**.
- Select your volume named **My volume**.
- In the **Actions** drop-down list, choose **Create Snapshot**, and then configure:
 - **Description:** Snapshot of my volum
 - Choose **Add Tag** and then configure:
 - **Key:** Nam
 - **Value:** My snapshc
 - Choose **Create Snapshot**

A **Create Snapshot Request Succeeded** message displays.

- Choose **Close**
- In the left navigation pane, under **ELASTIC BLOCK STORE**, choose **Snapshots**.

Your snapshot appears in the list.

In the future, this snapshot can be used to create a new volume that will have the same contents as when the snapshot was created.

Congratulations! You have successfully created a snapshot of an Amazon EBS volume.

Task 5: Restore the Amazon EBS Snapshot

You can create Amazon EBS volumes from snapshots to clone or restore your volumes. When you create an Amazon EBS volume based on a snapshot, the new volume begins as an exact replica of the original volume that was used to create the snapshot. The replicated volume loads data in the background so that you can begin using it immediately. Amazon EBS snapshots can also be easily shared among AWS users or copied over AWS regions.

To retrieve data stored in a snapshot, you **Restore** the snapshot to a new EBS volume.

Create a volume using your snapshot

- Select **My snapshot**.

If the snapshot is still creating, wait 30 seconds and then choose the refresh icon at the top-right corner of the page. If necessary, repeat this process until the status of the snapshot changes to *completed*.

- In the **Actions** menu, choose **Create Volume**.

When restoring a snapshot to a new volume, you can also modify the configuration, such as changing the volume type, size, or Availability Zone.

- Change the **Size (GiB)** field to .
- For **Availability Zone**, select the same availability zone that you used in Task 1.
- Choose **Add Tag** and then configure:

Key:
Value:

- Choose **Create Volume**

A **Create Volume Request Succeeded** message displays.

- Choose **Close**

Attach the restored volume to your EC2 instance

- In the left navigation pane, under **ELASTIC BLOCK STORE** choose **Volumes**.
- Select **Restored volume**.

Ensure that *Restored volume* is the only volume selected.

- In the **Actions** menu, choose **Attach Volume**.
- Choose the **Instance** field and then select the *running* instance that appears. It should be named *Lab*.
- Make a note of the path in the **Device** field. The default should be */dev/sdg*. You use it in a future task to mount the restored volume.
- Choose **Attach**

Notice that the volume state changed to *in-use*.

Mount the restored volume

- To the left of these instructions, copy the **Ec2Instance** URL and paste it into a new web browser window.

If you still have the EC2 Instance Connect window open from a previous task, but it seems to be unresponsive, refresh the page to restore connectivity. The connection will time out after a few minutes of not being used.

- In the EC2 Instance Connect session, enter the following command to create a directory for mounting the restored storage volume:

```
sudo mkdir /mnt/data-store2
```

- Enter the following command to mount the new volume. Replace **/dev/sdg** with the value of **Device** from the previous task, if it's different.

```
sudo mount /dev/sdg /mnt/data-store2
```

Note that you did not have to format the restored volume with a file system prior to mounting it. This is because the volume you restored the snapshot of was already formatted when the snapshot was taken.

Before you can use the additional storage that you allocated to the Amazon EBS volume, you must extend the file system partition from within the operating system.

Before extending a file system that contains valuable data, it is best practice to create a snapshot of the volume, in case you need to roll back your changes.

- Enter the following command to display information about the block devices attached to your instance:

```
lsblk
```

The output should look similar to this:

```
[ec2-user@ip-10-10-1-61 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1      259:0    0   8G  0 disk
├─nvme0n1p1  259:1    0   8G  0 part /
└─nvme0n1p128 259:2    0   1M  0 part
nvme1n1      259:3    0  10G  0 disk /mnt/data-store
nvme2n1      259:4    0  15G  0 disk /mnt/data-store2
```

Notice that the size of the restored Amazon EBS volume you attached to the instance (nvme2n1) is 15 GiB, while the original volume (nvme1n1) is 10 GiB.

- Enter the following command to verify the size of the file system for each volume:

```
df -h
```

Notice the size of the file system on the restored volume is still approximately 10 GiB. You must extend the file system to take advantage of the increased volume size.

- Enter the following command to extend the file system to use the additional available capacity of the volume:

```
sudo resize2fs /dev/nvme2n1
```

The output should look similar to this:

```
[ec2-user@ip-10-10-1-61 ~]$ sudo resize2fs /dev/nvme2n1
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/nvme2n1 is mounted on /mnt/data-store2; on-line resizing
required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/nvme2n1 is now 3932160 blocks long.
```

- Enter the following command to verify the size of the file system for each volume:

```
df -h
```

The size of the file system should now be approximately 15 GiB.

Next, verify that the file you created on the original volume has been restored with the snapshot.

- Enter the following command to list the contents of the volume you just mounted:

```
ls /mnt/data-store2/
```

Verify that volume you mounted has the file that you created earlier. You should see *file.txt*.

- Enter the following command to verify the contents of *file.txt* are intact:

```
cat /mnt/data-store2/file.txt
```

The output should read *some text has been written*, which is the contents of the file you created in the previous task.

Congratulations! You have successfully restored a snapshot of an Amazon EBS volume and attached it to an instance.

Task 6: Modify the EBS volume

In this task, you modify the volume type and size of an Amazon EBS volume.

- Return to the AWS Management Console.
- In the left navigation pane, under **ELASTIC BLOCK STORE** choose **Volumes**.

- Select **My volume**.

Ensure that *My volume* is the only volume selected.

- In the **Actions** drop-down list, choose **Modify Volume**.

This dialog box allows you to select a different volume type or change the size of the volume.

- Change **Size (GiB)** to:
- Select the **Volume Type** drop-down menu and review the different volumes types that you could change this volume to, if you desired to do so. For this lab, you can keep the current volume type of **General Purpose SSD (gp2)**.

Refer to the *Additional resources* section at the end of this lab for more information about Amazon EBS volume types.

- Choose **Modify**

A confirmation dialog box appears asking if you are sure you want to modify the volume.

- Choose **Yes**

A **Modify Volume Request Succeeded** message displays.

- Choose **Close**

- Choose the refresh icon. Your volume should now have a size of **20 GiB**.

Extend the volume partition on the EC2 instance

Before you can use the additional storage that you allocated to the Amazon EBS volume, you must extend the file system partition from within the operating system.

Before extending a file system that contains valuable data, it is best practice to create a snapshot of the volume, in case you need to roll back your changes.

- To the left of these instructions, copy the **Ec2Instance** URL and paste it into a new web browser window.

If you still have the EC2 Instance Connect window open from a previous task, but it seems to be unresponsive, refresh the page to restore connectivity. The connection will time out after a few minutes of not being used.

- In the EC2 Instance Connect session, enter the following command to display information about the block devices attached to your instance:

```
lsblk
```

The output should look similar to this:

```
[ec2-user@ip-10-10-1-61 ~]$ lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
nvme0n1              259:0    0   8G  0 disk
├─nvme0n1p1          259:1    0   8G  0 part /
└─nvme0n1p128        259:2    0   1M  0 part
nvme1n1              259:3    0  20G  0 disk /mnt/data-store
nvme2n1              259:4    0  15G  0 disk /mnt/data-store2
```

Notice that the size of the first Amazon EBS volume you attached to the instance (nvme1n1) is now 20 GiB.

- Enter the following command to verify the size of the file system for each volume:

```
df -h
```

Notice the size of the file system on the volume is still approximately 10 GiB. You must extend the file system to take advantage of the increased volume size.

- Enter the following command to extend the file system to use the additional available capacity of the volume:

```
sudo resize2fs /dev/nvme1n1
```

The output should be similar to this:

```
[ec2-user@ip-10-10-1-61 ~]$ sudo resize2fs /dev/nvme1n1
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/nvme1n1 is mounted on /mnt/data-store; on-line resizing
required
old_desc_blocks = 1, new_desc_blocks = 2
The filesystem on /dev/nvme1n1 is now 5242880 blocks long.
```

- Enter the following command to verify the size of the file system for each volume:

```
df -h
```

The size of the file system should now be approximately 20 GiB.

Conclusion

Congratulations! You have successfully learned how to:

- Create an Amazon EBS volume in the Amazon Management Console.
- Attach an EBS Volume to an Amazon EC2 instance.
- Create a snapshot of an EBS volume.
- Restore a snapshot of an EBS volume.
- Modify an EBS Volume.