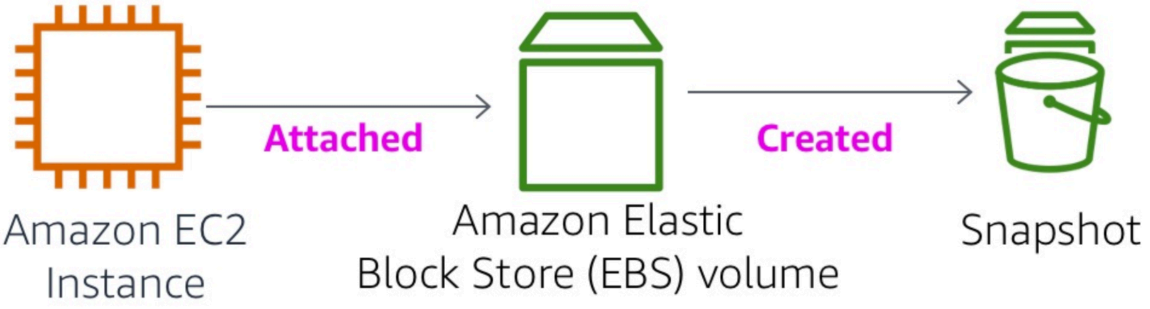
Lab Overview



This lab focuses on Amazon Elastic Block Store (Amazon EBS), a key underlying storage mechanism for Amazon EC2 instances. In this lab, you will learn how to create an Amazon EBS volume, attach it to an instance, apply a file system to the volume, and then take a snapshot backup.

Topics covered

By the end of this lab, you will be able to:

- Create an Amazon EBS volume
- Attach and mount your volume to an EC2 instance
- Create a snapshot of your volume
- Create a new volume from your snapshot
- Attach and mount the new volume to your EC2 instance

Lab Pre-requisites

To successfully complete this lab, you should be familiar with basic Amazon EC2 usage and with basic Linux server administration. You should feel comfortable using the Linux command-line tools.

Duration

This lab will require approximately **30 minutes** to complete.

What is Amazon Elastic Block Store?

Amazon Elastic Block Store (Amazon EBS) offers persistent storage for 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 instances 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 storage resources used while maintaining your instance's state. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone).

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 the 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.

Amazon EBS Volume Features

Amazon EBS volumes deliver the following features:

- **Persistent storage:** Volume lifetime is independent of any particular Amazon EC2 instance.
- **General purpose:** Amazon EBS volumes are raw, unformatted block devices that can be used from any operating system.
- **High performance:** Amazon EBS volumes are equal to or better than local Amazon EC2 drives.
- **High reliability:** Amazon EBS volumes have built-in redundancy within an Availability Zone.
- **Designed for resiliency:** The AFR (Annual Failure Rate) of Amazon EBS is between 0.1% and 1%.
- **Variable size:** Volume sizes range from 1 GB to 16 TB.
- **Easy to use:** Amazon EBS volumes can be easily created, attached, backed up, restored, and deleted.

Task 1: Create a New EBS Volume

In this task, you will create and attach an Amazon EBS volume to a new Amazon EC2 instance.

1. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
2. In the left navigation pane, choose **Instances**.
   An Amazon EC2 instance named **Lab** has already been launched for your lab.
3. Note the **Availability Zone** of the instance. It will look similar to *us-east-1a*.
4. In the left navigation pane, choose **Volumes**.
   You will see an existing volume that is being used by the Amazon EC2 instance. This volume has a size of 8 GiB, which makes it easy to distinguish from the volume you will create next, which will be 1 GiB in size.
5. Choose **Create volume** then configure:
   - **Volume Type:** *General Purpose SSD (gp2)*
   - **Size (GiB):** `1`. **NOTE**: You may be restricted from creating large volumes.
   - **Availability Zone:** Select the same availability zone as your EC2 instance.
   - Choose **Add Tag**
   - In the Tag Editor, enter:
     - **Key:** `Name`
     - **Value:** `My Volume`
6. Choose **Create Volume**
   Your new volume will appear in the list, and will move from the *Creating* state to the *Available* state. You may need to choose **refresh** to see your new volume.

Task 2: Attach the Volume to an Instance

You can now attach your new volume to the Amazon EC2 instance.

7. Select **My Volume**.
8. In the **Actions** menu, choose **Attach volume**.
9. Choose the **Instance** field, then select the instance that appears (Lab).
   Note that the **Device** field is set to */dev/sdf*. You will use this device identifier in a later task.
10. Choose **Attach volume**

    The volume state is now *In-use*.

Task 3: Connect to Your Amazon EC2 Instance (Skip if you already have)

Windows Users: Using SSH to Connect

These instructions are for Windows users only.

If you are using macOS or Linux, skip to the next section.

11. Download needed software.
    - You will use **PuTTY** to SSH to Amazon EC2 instances. If you do not have PuTTY installed on your computer, download it here.
12. Open **putty.exe**
13. Configure PuTTY to not timeout:
    - Choose **Connection**
    - Set **Seconds between keepalives** to `30`
14. This allows you to keep the PuTTY session open for a longer period of time.
15. Configure your PuTTY session:
    - Choose **Session**
    - **Host Name (or IP address):** Paste the *Public DNS or IPv4 address* of the Lab instance that you noted earlier.
    - Back in PuTTY, in the **Connection** list, expand **SSH**
    - Choose **Auth** and expand **Credentials**
    - Under **Private key file for authentication:** Choose **Browse**
    - Browse to the *.ppk* file that you downloaded, select it, and choose **Open**
    - Choose **Open** again
16. To trust and connect to the host, choose **Accept**
17. When prompted **login as**, enter: `ec2-user`
    This will connect you to the EC2 instance.
18. Windows Users: Choose here to skip ahead to the next task.

macOS and Linux Users

These instructions are for Mac/Linux users only. If you are a Windows user, skip ahead to the next task.

19. Open a terminal window, and change directory `cd` to the directory where the .pem file was downloaded.
    For example, run this command, if it was saved to your Downloads directory:

20. `cd ~/Downloads`
21. Change the permissions on the key to be read only, by running this command:
22. `chmod 400 .pem`
23. Return to the AWS Management Console, and in the EC2 service, choose **Instances**.
    The **Lab** instance should be selected.
24. In the *Details* tab, copy the **Public IPv4 address** value.
25. Return to the terminal window and run this command (replace **<public-ip>** with the actual public IP address you copied):
26. `ssh -i .pem ec2-user@<public-ip>`
27. Type `yes` when prompted to allow a first connection to this remote SSH server.
    Because you are using a key pair for authentication, you will not be prompted for a password.

Task 4: Create and Configure Your File System

In this task, you will add the new volume to a Linux instance as an ext3 file system under the /mnt/data-store mount point.

If you are using PuTTY, you can paste text by right-clicking in the PuTTY window.

28. View the storage available on your instance:
    a. `df -h`
29. You should see output similar to:

```
a. Filesystem      Size  Used Avail Use% Mounted on
b. devtmpfs        484M     0  484M   0% /dev
c. tmpfs           492M     0  492M   0% /dev/shm
d. tmpfs           492M  460K  491M   1% /run
e. tmpfs           492M     0  492M   0% /sys/fs/cgroup
f. /dev/xvda1      8.0G  1.5G  6.6G  19% /
g. tmpfs            99M     0   99M   0% /run/user/0
h. tmpfs            99M     0   99M   0% /run/user/1000
```

    This is showing the original 8GB disk volume. Your new volume is not yet shown.

30. Create an ext3 file system on the new volume:
    a. `sudo mkfs -t ext3 /dev/sdf`
31. Create a directory for mounting the new storage volume:
    a. `sudo mkdir /mnt/data-store`
32. Mount the new volume:
    a. `sudo mount /dev/sdf /mnt/data-store`

To configure the Linux instance to mount this volume whenever the instance is started, you will need to add a line to */etc/fstab*.

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

33. View the configuration file to see the setting on the last line:
    a. `cat /etc/fstab`
34. View the available storage again:
    a. `df -h`

The output will now contain an additional line - */dev/xvdf*:

```
b. Filesystem      Size  Used Avail Use% Mounted on
c. devtmpfs        484M     0  484M   0% /dev
d. tmpfs           492M     0  492M   0% /dev/shm
e. tmpfs           492M  460K  491M   1% /run
f. tmpfs           492M     0  492M   0% /sys/fs/cgroup
g. /dev/xvda1      8.0G  1.5G  6.6G  19% /
h. tmpfs            99M     0   99M   0% /run/user/0
i. tmpfs            99M     0   99M   0% /run/user/1000
j. /dev/xvdf       976M  1.3M  924M   1% /mnt/data-store
```

35. On your mounted volume, create a file and add some text to it.
    a. `sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"`
36. Verify that the text has been written to your volume.
    a. `cat /mnt/data-store/file.txt`

Task 5: Create an Amazon EBS Snapshot

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

You can create any number of point-in-time, consistent snapshots from Amazon EBS volumes at any time. Amazon EBS snapshots are stored in Amazon S3 with high durability. New Amazon EBS volumes can be created out of snapshots for cloning or restoring backups. Amazon EBS snapshots can also be easily shared among AWS users or copied over AWS regions.

37. In the **AWS Management Console**, choose **Volumes** and select **My Volume**.
38. In the **Actions** menu, select **Create snapshot**.
39. Choose **Add tag** then configure:
    - **Key:** `Name`
    - **Value:** `My Snapshot`
    - Choose **Create snapshot**

40. In the left navigation pane, choose **Snapshots**.
    Your snapshot is displayed. The status will first have a state of *Pending*, which means that the snapshot is being created. It will then change to a state of *Completed*.
    Note: Only used storage blocks are copied to snapshots, so empty blocks do not occupy any snapshot storage space.
41. In your remote SSH session, delete the file that you created on your volume.
    a. `sudo rm /mnt/data-store/file.txt`
42. Verify that the file has been deleted.
    a. `ls /mnt/data-store/`

    Your file has been deleted.

Task 6: Restore the Amazon EBS Snapshot

If you ever wish to retrieve data stored in a snapshot, you can **Restore** the snapshot to a new EBS volume.

Create a Volume Using Your Snapshot

43. In the **AWS Management Console**, select **My Snapshot**.
44. In the **Actions** menu, select **Create volume from snapshot**.
45. For **Availability Zone** Select the same availability zone that you used earlier.
46. Choose **Add tag** then configure:
    - **Key:** `Name`
    - **Value:** `Restored Volume`
    - Choose **Create volume**
47. Note: When restoring a snapshot to a new volume, you can also modify the configuration, such as changing the volume type, size or Availability Zone.

Attach the Restored Volume to Your EC2 Instance

48. In the left navigation pane, choose **Volumes**.
49. Select **Restored Volume**.
50. In the **Actions** menu, select **Attach volume**.
51. Choose the **Instance** field, then select the (Lab) instance that appears.
    Note that the **Device** field is set to */dev/sdg*. You will use this device identifier in a later task.
52. Choose **Attach volume**
    The volume state is now *in-use*.

Mount the Restored Volume

53. Create a directory for mounting the new storage volume:
    a.
    ```
    sudo mkdir /mnt/data-store2
    ```
54. Mount the new volume:
    a.
    ```
    sudo mount /dev/sdg /mnt/data-store2
    ```
55. Verify that volume you mounted has the file that you created earlier.
    a.
    ```
    ls /mnt/data-store2/
    ```

    You should see file.txt.

Conclusion

Congratulations! You now have successfully:

- Created an Amazon EBS volume
- Attached the volume to an EC2 instance
- Created a file system on the volume
- Added a file to volume
- Created a snapshot of your volume
- Created a new volume from the snapshot
- Attached and mounted the new volume to your EC2 instance
- Verified that the file you created earlier was on the newly created volume

Lab Complete

Congratulations! You have completed the lab.