# **Working with Amazon Elastic Block Store (Amazon EBS)**

## **Lab overview**

****

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

## **Topics covered**

After completing this lab, you will be able to:

* Create an 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.

## **Other AWS services**

During your access time in this lab, an AWS Identity and Access Management (IAM) policy has disabled any AWS services other than the ones needed for this lab. In addition, the capabilities of the services used in this lab are limited to what the lab requires and in some cases are even further limited as an intentional aspect of the lab design. Expect errors when accessing other services or performing actions beyond those that the lab guide provides.

## **What is Amazon EBS?**

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

When used as a boot partition, EC2 instances can be stopped and subsequently restarted so that you pay for the storage resources used only while maintaining your instance's state. EBS volumes offer greatly improved durability over local EC2 instance stores because 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 EBS volumes and can protect your data for long-term durability. You can also easily share these snapshots with coworkers and other AWS developers.

This lab guide explains the basic concepts of Amazon EBS. However, it gives only a brief overview of Amazon EBS concepts. For further information, see the [Amazon EBS documentation](https://aws.amazon.com/ebs/).

## **EBS volume features**

EBS volumes deliver the following features:

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

## **Duration**

This lab takes approximately **45 minutes** to complete.

## **Accessing the AWS Management Console**

1. At the top of these instructions, select Start Lab to launch your lab.

A Start Lab panel opens displaying the lab status.

1. Wait until you see the message **Lab status: ready**, and then select the **X** to close the Start Lab panel.
2. Select the Details dropdown menu above these instructions, and then select Show

Copy all the lab details, such as **PublicIP**, **AvailabilityZone**, etc., and save them in a text editor such as [Atom](https://atom.io/), [Sublime Text](https://www.sublimetext.com/), or [Visual Studio Code](https://code.visualstudio.com/).

This lab refers to this information that you have saved as **Lab Details**.

1. At the top of these instructions, select AWS

This step opens the AWS Management Console in a new browser tab. The system automatically logs you in.

**Tip**: If a new browser tab does not open, a banner or icon at the top of your browser typically indicates that your browser is preventing the site from opening pop-up windows. Select the banner or icon, and choose **Allow pop-ups**.

1. Arrange the AWS Management Console tab so that it displays alongside these instructions. Ideally, you should be able to see both browser tabs at the same time to make it easier to follow the lab steps.

**Do not change the Region unless instructed to do so**.

## **Task 1: Creating a new EBS volume**

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

1. In the AWS Management Console, on the **Services** menu, choose **EC2**.
2. In the left navigation pane, choose **Instances**.  
   An EC2 instance named **Lab** has already been launched for your lab.
3. Note the **Availability Zone** of the instance. It looks similar to **us-west-2a**.
4. In the left navigation pane, choose **Volumes**.  
   You see an existing volume that the EC2 instance is using. This volume has a size of 8 GiB, which makes it easy to distinguish from the volume you create next, which is 1 GiB in size.
5. Choose **Create Volume**, and then configure the following:
   * **Volume Type:** Select **General Purpose SSD (gp2)**
   * **Size (GiB):** Enter 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 the following:
     + **Key:** Enter Name
     + **Value:** Enter My Volume
6. Choose **Create Volume**, and then choose **Close**Your new volume appears in the list and moves from the **creating** state to the **available** state. You may need to choose **Refresh** to see your new volume.

## **Task 2: Attaching the volume to an instance**

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

1. Select the check box next to **My Volume**.
2. In the **Actions** menu, choose **Attach Volume**.
3. Choose the **Instance** field, and then select the instance that appears (Lab).  
   Note that the **Device** field is set to **/dev/sdf**. You use this device identifier in a later task.
4. Choose **Attach**

The volume state is now **in-use**.

## **Task 3: Connecting to your EC2 instance**

### **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](https://labs.vocareum.com/web/1469070/423601.0/ASNLIB/public/docs/lang/en/README.html#ssh-MACLinux).

1. Read through the three bullet points in this step before you start to complete the actions because you will not be able see these instructions when the **Details** panel is open.
   * Choose the Details dropdown menu above these instructions you are currently reading, and then choose Show.  
      A **Credentials** window opens.
   * Choose the **Download PPK** button, and save the **labsuser.ppk** file. Typically, your browser saves it to the Downloads directory.
   * To exit the **Details** panel, choose the **X**.
2. Download the needed software.
   * You use **PuTTY** to connect to the EC2 instance by using SSH. If you do not have PuTTY installed on your computer, [download it here](https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe).
3. Open **putty.exe**.
4. Configure PuTTY to not timeout:
   * Choose **Connection**.
   * Set **Seconds between keepalives** to 30
5. This step allows you to keep the PuTTY session open for a longer period of time.
6. Configure your PuTTY session:

* Choose **Session**.
* For **Host Name (or IP address)**, copy and paste the public IPv4 address for the instance. To find it, return to the EC2 console, and choose **Instances**. Select the box next to the instance, and in the **Details** tab, copy the **Public IPv4 address** value.
* In PuTTy, in the **Connection** list, expand **SSH**
* Choose **Auth** (don't expand it).
* Choose **Browse**.
* Browse to and select the **labsuser.ppk** file that you downloaded.
* Choose **Open** to select it.
* Choose **Open**.

1. Choose **Yes** to trust the host and connect to it.
2. When prompted with **login as**, enter ec2-user  
   This connects you to the EC2 instance.
3. Windows users can [skip ahead to the next task](https://labs.vocareum.com/web/1469070/423601.0/ASNLIB/public/docs/lang/en/README.html#ssh-after).

### **macOS and Linux Users**

These instructions are for Mac/Linux users only. If you are a Windows user, [skip ahead to the next task](https://labs.vocareum.com/web/1469070/423601.0/ASNLIB/public/docs/lang/en/README.html#ssh-after).

1. Read through all the instructions in this one step before you start to complete the actions because you will not be able see these instructions when the **Details** panel is open.
   * Choose the Details dropdown menu above these instructions you are currently reading, and then choose Show.  
      A Credentials window opens.
   * Choose the **Download** button, and save the **labsuser.pem** file.
   * To exit the **Details** panel, choose the **X**.
2. Open a terminal window, and change the cd directory to the directory where the **labsuser.pem** file was downloaded.  
   For example, run the following command if the file was saved to your Downloads directory:

cd ~/Downloads

1. To change the permissions on the key to be read only, run the following command:

chmod 400 labsuser.pem

1. Return to the AWS Management Console, and in the EC2 service, choose **Instances**.  
   The **Lab** instance should selected.
2. In the **Description** tab, copy the **Public IPv4 address** value.
3. In the command below, replace **<public-ip>** with the public IP address you copied in the previous step. Return to the terminal window, and run the adjusted command:

ssh -i labsuser.pem ec2-user@<public-ip>

1. Enter yes when prompted to allow a first connection to this remote SSH server.  
   Because you are using a key pair for authentication, you are not prompted for a password.

## **Task 4: Creating and configuring your file system**

In this task, you 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.

1. Run the following command to view the storage available on your instance:

df -h

1. You should see output similar to the following:

Filesystem Size Used Avail Use% Mounted on  
devtmpfs 469M 0 469M 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 1.5G 6.6G 18% /  
tmpfs 96M 0 96M 0% /run/user/1000

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

The path **/dev/nvme0n1p1** may seem unfamiliar to you. This is because the EC2 instance created for you is an Amazon Linux instance that is built on the AWS Nitro System, which enables high performance. On the Nitro System, EBS volumes are exposed as **NVMe** block devices. The devices names that you specify in a block device mapping, such as **/dev/sdf**, are renamed using NVMe device names in the format **/dev/nvme[0-26]n1**.

1. Run the following command to list the device that was previously created for your EBS volumes:

ls -l /dev/sdf

1. You should see an output similar to the following:

lrwxrwxrwx 1 root root 7 May 4 08:47 /dev/sdf -> nvme1n1

In Task 2, the EBS volumes was attached to the **/dev/sdf** device. As you can see, Amazon Linux also creates a symbolic link from the device name in the block device mapping (for example, **/dev/sdf**) to the NVMe device name (**/dev/nvme1n1**).

1. On the Nitro System, you can use the **lsblk** command to verify which device is mounted. Run the following command:

lsblk

1. You should see an output similar to the following:

​  
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 259:5 0 1G 0 disk  
​

The output of **lsblk** removes the /dev/ prefix from full device paths. Here you can see that the **nvme1n1** volume is not mounted yet.

1. Run the following command to verify that there is no file system attached to the device:

sudo lsblk -f

1. You should see an output similar to the following:

NAME FSTYPE LABEL UUID MOUNTPOINT  
 nvme0n1  
 ├─nvme0n1p1 xfs / 7b35... /  
 └─nvme0n1p128  
 nvme1n1

As you can see, the **nvme1n1** device linked to the **/dev/sdf** device is not formatted yet.

1. Run the following command to create an ext3 file system on the new volume:

sudo mkfs -t ext3 /dev/sdf

1. Run the following command to create a directory for mounting the new storage volume:

sudo mkdir /mnt/data-store

1. Run the following command to mount the new volume:

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

1. To configure the Linux instance to mount this volume whenever the instance is started, you need to add a line to **/etc/fstab**. Run the following command:

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

1. Run the following command to view the configuration file to see the setting on the last line:

cat /etc/fstab

1. Run the following command to view the available storage again:

df -h

1. The output now contains an additional line: **/dev/nvme[x]n1**:

Filesystem Size Used Avail Use% Mounted on  
.......  
/dev/nvme1n1 976M 1.3M 924M 1% /mnt/data-store  
.......  
​

1. You can also use the **lsblk** command. Enter:

lsblk -f

1. The output will be :

NAME FSTYPE LABEL UUID MOUNTPOINT   
 nvme0n1  
├─nvme0n1p1 xfs / 7b... /  
└─nvme0n1p128  
nvme1n1 ext3 a2b... /mnt/data-store

1. On your mounted volume, run the following command to create a file and add some text to it:

sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"

1. Run the following command to verify that the text has been written to your volume:

cat /mnt/data-store/file.txt

## **Task 5: Creating an EBS snapshot**

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

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

1. In the AWS Management Console, choose **Volumes**, and select the check box next to **My Volume**.
2. In the **Actions** menu, choose **Create Snapshot**.
3. Choose **Add Tag**, and then configure the following options:
   * For **Key**, enter Name
   * For **Value**, enter My Snapshot
   * Choose **Create Snapshot**, and then choose **Close**
4. Your snapshot is listed in the **Snapshots** console.
5. In the left navigation pane, choose **Snapshots**.  
   Your snapshot is displayed. It starts with a state of **pending**, which means that the snapshot is being created. It then changes to a state of **completed**. Only used storage blocks are copied to snapshots, so empty blocks do not take any snapshot storage space.
6. In PuTTY, run the following command to delete the file that you created on your volume in your remote SSH session:

sudo rm /mnt/data-store/file.txt

1. Run the following command to verify that the file has been deleted:

ls /mnt/data-store/

1. Your file has been deleted.

## **Task 6: Restoring the EBS snapshot**

If you ever want to retrieve data stored in a snapshot, you can restore the snapshot to a new EBS volume.

### **Create a volume using your snapshot**

1. In the **AWS Management Console**, select the check box next to **My Snapshot**.
2. In the **Actions** menu, choose **Create Volume**.
3. For **Availability Zone**, select the same Availability Zone that you used earlier.
4. Choose **Add Tag**, and then configure:
   * For **Key,**, enter Name
   * For **Value,**, enter Restored Volume
   * Choose **Create Volume**
   * Choose **Close**
5. 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**

1. In the left navigation pane, choose **Volumes**.
2. Select the check box next to **Restored Volume**.
3. In the **Actions** menu, choose **Attach Volume**.
4. Choose the **Instance** field, and then select the instance that appears (Lab).

Note that the **Device** field is set to **/dev/sdg**.

1. Choose **Attach**The volume state is now **in-use**.

### **Mount the restored volume**

1. In PuTTY, run the following command to create a directory for mounting the new storage volume:

sudo mkdir /mnt/data-store2

1. Run the following command to mount the new volume:

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

1. Run the following command to verify that the volume you mounted has the file that you created earlier:

ls /mnt/data-store2/

1. You should see the **file.txt** file.

## **Conclusion**

Congratulations! You have completed the lab.

Select End Lab at the top of this page, and then select **Yes** to confirm that you want to end the lab.

​

1. A panel appears indicating that **DELETE has been initiated... You may close this message box now.**
2. Select the **X** in the top-right corner to close the panel.