1. Create a New EBS Volume: in the AWS Managemen

Go to the EC2 Dashboard t Console.

* In the navigation pane, under "Elastic Block Store", click "Volumes".
* Click "Create Volume". Choose the volume type, size, and availability zone (which should be the same as your EC2 instance). Click "Create".

2. Attach the EBS Volume to Your EC2 Instance:

* After the volume is created, select it, and click "Actions", then "Attach Volume".
* Search for your EC2 instance by instance ID or name and attach the volume to it.
* /dev/sdf

On Your RHEL System

After attaching the EBS volume to your EC2 instance, you need to make the operating system recognize the new volume, format it, and mount it:

1. Log in to Your EC2 Instance:

- Use SSH to connect to your EC2 instance running RHEL.

2. Identify the New EBS Volume:

- Use the `lsblk` command to list all available disk devices and their mount points to identify the newly attached EBS volume (e.g., `/dev/xvdf`).

lsblk

3. Check if the Volume Contains a File System:

- Use the `file -s` command on your new device to check if there's a file system. If the output shows "data", it means there is no file system.

sudo file -s /dev/xvdf

4. Create a File System:

- If there's no file system, create one. For example, to create an ext4 file system, use the following command:

sudo mkfs -t ext4 /dev/xvdf

5. Create a Mount Point:

- Create a directory to mount your new volume.

sudo mkdir /mnt/my-new-volume

6. Mount the Volume:

- Mount the volume to the directory you've just created.

sudo mount /dev/xvdf /mnt/my-new-volume

7. Ensure the Volume Mounts on Reboot:

- Edit `/etc/fstab` to add an entry for the device to ensure it mounts on reboot.

- Get the UUID of the device using the `blkid` command:

sudo blkid /dev/xvdf

- Open the `/etc/fstab` file and add a new line with the UUID. It should look something like this:

UUID=your-uuid /mnt/my-new-volume ext4 defaults,nofail 0 2

- Replace `your-uuid` with the actual UUID you got from the `blkid` command.

8. Test the Configuration:

- To make sure everything is set up correctly, test the mount configuration:

sudo mount -a

Data Partitioning

1. Unmount the disk if it is mounted:

sudo umount /mnt/my-new-volume

1. Start the fdisk utility on your disk:

sudo fdisk /dev/xvdf

1. **Create the First Partition:**

* Enter n for a new partition.
* Choose p for a primary partition.
* Choose the partition number as 1.
* Press Enter to accept the default first sector.
* When asked for the last sector, type +5G to create a 5GB partition.

1. **Create the Second Partition:**

* Enter n for a new partition.
* Choose p for a primary partition.
* Choose the partition number as 2.
* Press Enter to accept the default first sector.
* Press Enter again to accept the default last sector, which should automatically allocate the rest of the space (around 5GB).

1. Write Changes: After creating both partitions, press w to write the changes and exit fdisk.
2. Format the Partitions:

* You'll need to format the new partitions with a file system:
* sudo mkfs -t ext4 /dev/xvdf1
* sudo mkfs -t ext4 /dev/xvdf2

1. Mount the Partitions:

* Create mount points and mount the partitions:
* sudo mkdir /mnt/partition1
* sudo mkdir /mnt/partition2
* sudo mount /dev/xvdf1 /mnt/partition1
* sudo mount /dev/xvdf2 /mnt/partition2

1. Update /etc/fstab:

* Get the UUIDs of the new partitions:
* sudo blkid
* Edit /etc/fstab and add the following lines, replacing uuid-of-xvdf1 and uuid-of-xvdf2 with the respective UUIDs you just obtained:
* UUID=uuid-of-xvdf1 /mnt/partition1 ext4 defaults,nofail 0 2
* UUID=uuid-of-xvdf2 /mn/partition2 ext4 defaults,nofail 0 2

**Or**

Edit /etc/fstab and add the following lines

/dev/xvdf1 /mnt/partition1 ext4 defaults,nofail 0 2

/dev/xvdf2 /mnt/partition2 ext4 defaults,nofail 0 2

Sudo mount -a