**Step-by-Step Guide to Using Elastic Block Store (EBS) with Linux EC2 Instances**

**Part 1: Creating a Linux EC2 Machine with Multiple Volumes**

1. **Login to AWS Console**
   * Open the **AWS Management Console**.
   * Navigate to **EC2 Dashboard**.
2. **Launch a New Linux EC2 Instance**
   * Click **Launch Instance**.
   * Choose **Amazon Linux 2** (or any preferred Linux AMI).
   * Select an **instance type** (e.g., t2.micro for free tier).
   * Configure security group (allow SSH access, port 22).
3. **Add an Additional EBS Volume**
   * In **Step 4: Add Storage**, click **Add New Volume**.
   * Set **Volume Type** to **General Purpose SSD (gp2)**.
   * Set **Size** (e.g., 10 GiB).
   * Check **Delete on Termination** (optional).
4. **Launch the Instance**
   * Review settings and click **Launch**.
   * Select or create a key pair for SSH access.
   * Click **Launch Instances**.

**Part 2: Checking & Mounting the Additional EBS Volume**

**1. Connect to the EC2 Instance using SSH**

* Use **Putty** (Windows) or ssh command (Linux/Mac):
* ssh -i "your-key.pem" ec2-user@<your-instance-public-ip>

**2. Verify Disk Space Usage**

* Run the following command:
* df -hT

**3. Check Available Disk Devices**

* Run:
* lsblk
* Identify the new volume (e.g., /dev/xvdb).

**4. Verify the EBS Volume**

* Run:
* sudo file -s /dev/xvdb
* If it says **"data"**, it means the volume is not yet formatted.

**5. Format the Volume**

* Create an **ext4** file system:
* sudo mkfs -t ext4 /dev/xvdb

**6. Create a Mount Directory**

* Run:
* mkdir test

**7. Mount the Volume**

* Mount the EBS volume:
* sudo mount /dev/xvdb test

**8. Verify the Mount**

* Change directory:
* cd test
* Create sample files:
* sudo touch file1 file2 file3
* Check the files:
* ls
* Check disk space again:
* df -hT

**Part 3: Unmount and Detach the Volume**

**1. Unmount the EBS Volume**

* Move out of the directory:
* cd
* Unmount the volume:
* sudo umount test

**2. Detach the Volume in AWS Console**

* Go to **EC2 Dashboard > Volumes**.
* Select the **EBS Volume**.
* Click **Actions > Detach Volume**.
* Confirm **Detach**.

**Part 4: Attach the EBS Volume to Another EC2 Instance**

**1. Create Another Linux EC2 Instance**

* Repeat **Step 2** to launch another EC2 instance in **the same availability zone**.

**2. Attach the Volume**

* Go to **EC2 Dashboard > Volumes**.
* Select the **detached EBS Volume**.
* Click **Actions > Attach Volume**.
* Select the **new EC2 instance**.
* Click **Attach**.

**Part 5: Mount the EBS Volume on the Second EC2 Instance**

**1. Connect to the Second EC2 Instance**

* Use SSH:
* ssh -i "your-key.pem" ec2-user@<new-instance-public-ip>

**2. Verify Disk Space Usage**

* Run:
* df -hT

**3. Check Available Disk Devices**

* Run:
* lsblk
* Identify the new volume (e.g., /dev/xvdf).

**4. Verify the Attached Volume**

* Run:
* sudo file -s /dev/xvdf
* It should now show a valid **ext4 file system**.

**5. Create a New Mount Directory**

* Run:
* mkdir training

**6. Mount the Volume**

* Run:
* sudo mount /dev/xvdf training

**7. Verify the Files**

* Change to the directory:
* cd training
* List the files:
* ls
* You should see **file1, file2, file3** from the previous EC2 instance.

**Final Verification**

* Run:
* df -hT
* Check that /dev/xvdf is mounted under /training.

**Conclusion 🎉**

1. **Created an EC2 instance with multiple EBS volumes.**
2. **Mounted and formatted an additional EBS volume.**
3. **Detached the volume and reattached it to another EC2 instance.**
4. **Mounted it on the new instance and verified the data.**

Now, your EBS volume can be used across different EC2 instances, preserving data even after the instance is terminated. 🚀