Date 14-04-2-2024





A white background with black text

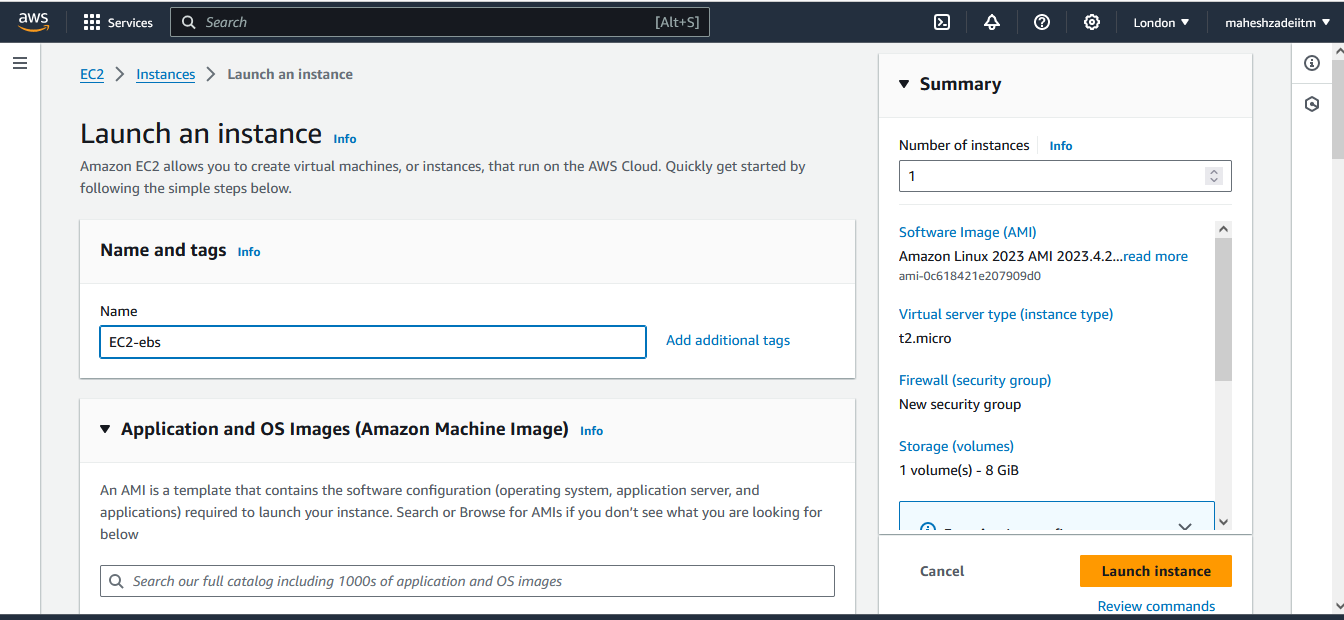
Description automatically generated

In the fast-paced world of cloud computing, AWS has emerged as a dominant force, offering a wide array of services and tools to cater to the ever-evolving needs of businesses. One such crucial feature is Elastic Block Store (EBS) volumes, which allow you to attach additional storage to your Amazon EC2 instances.

In this blog post, I'll walk you through the process of launching a Linux EC2 instance, creating an EBS volume (let's say 1 GiB), and resizing it (let's say 2 GiB).

**Task 1: Launch a Linux EC2 Instance**

1. Log in to the AWS console and Click on EC2.



1. Enter a name for the instance EC2-ebs

3) Select the Linux AMI that best suits your needs. For example, you might choose "Amazon Linux 2" or "Ubuntu Server." I'm choosing Ubuntu OS from the suggestions.

A screenshot of a computer

Description automatically generated

4.Select the instance type that matches your requirements in terms of CPU, memory, and other resources. You can choose the free tier-eligible options or scale up for more power. I keep the default instance type (t2.micro)

A screenshot of a computer

Description automatically generated

5 . Click on Create key pair and download the .pem file. Later, we’ll use PuTTYgen to create a .ppk file to access the EC2 instance.

A screenshot of a computer

Description automatically generated

6. In the network settings, select the check boxes corresponding to SSH and HTTP Traffic.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

7. Configure the amount of storage you want for your instance. The default 8 GB of gp2 volume should be enough for a basic web server.

A screenshot of a computer

Description automatically generated

8. Review the summary of the EC2 instance and click on the Launch Instance. It’ll take a moment to initiate the instance.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

9. The newly created instance is launched in the availability zone in the US-East-1 Region. We’ll now open PuTTY & copy this Public IP to access the instance from PuTTY.

A screenshot of a computer

Description automatically generated

10.Open PuTTY & paste the copied Public IP in the Host Name (or IP address).

A screenshot of a computer

Description automatically generated

Then go to, Connection -> SSH -> Auth -> Credentials.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

11.Now we need a putty private key file for authentication to access the instance. So we’ll now open PuTTYgen & generate the .ppk file from the .pem file.

**Generating .ppk from .pem:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

12 .Select the generated .ppk file and then click on Open.

A screenshot of a computer

Description automatically generated

13. It is successfully connected to the instance. Click on Accept.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Enter user name as **ubuntu** to log in to the instance.

A screenshot of a computer

Description automatically generated

The instance is ready for use.

A screenshot of a computer

Description automatically generated

Run the command to update the system



Install and Configure Apache



Copy public IP from EC2 ubuntu instance and paste in browser to check Apache2 is install

Public IP : 13.40.3.202

A screenshot of a computer

Description automatically generated

>>>>> Another way of connecting EC2 ubuntu instance by ssh command

Before the go the path where the private key is available.( here is in download folder)

Type the command

ssh -i london.pemkeypair.pem ubuntu@13.40.3.202



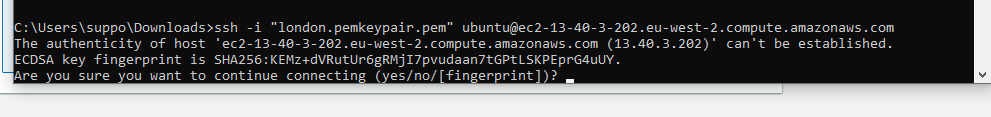
A screenshot of a computer

Description automatically generated

Another way of connecting EC2 Instacne by Public DNS

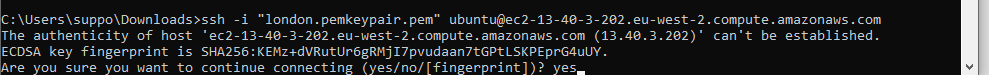
A screenshot of a computer

Description automatically generated



Below Command

ssh -i "london.pemkeypair.pem" [ubuntu@ec2-13-40-3-202.eu-west-2.compute.amazonaws.com](mailto:ubuntu@ec2-13-40-3-202.eu-west-2.compute.amazonaws.com)



A screenshot of a computer

Description automatically generated

Another way connecting EC2 Instance by EC2 Instance connect Method

A screenshot of a computer

Description automatically generated

Click on connect

A screenshot of a computer

Description automatically generated

**Task 2: Create an EBS Volume and Attach It to the instance**

1.Now we’ll create an EBS volume with 1 GB of storage and attach it to the created EC2 instance. In this same EC2 page, scroll down to Elastic Block Store (EBS) and click on Volumes.

A screenshot of a computer

Description automatically generated

2.Currently, it is showing the Root Volume Device of the instance launched in the previous task. Click on Create Volume

A screenshot of a computer

Description automatically generated

1. Enter 1 GiB for the size and select the availability zone that our current instance is running from.
2. A screenshot of a computer

   Description automatically generated

A screenshot of a computer

Description automatically generated

Click on Create Volume

Keep the remaining options as it is and click on Create Volume.

A screenshot of a computer

Description automatically generated

1 Gib Volume created

A screenshot of a computer

Description automatically generated

4 Now, we’ll attach this volume to our instance. Go to Actions -> Attach Volume

A screenshot of a computer

Description automatically generated

1. Select the instance to which we want to attach our newly created volume and click on attach volume.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Volume is successfully attached to our instance. Now we’ll go to our instance.

A screenshot of a computer

Description automatically generated

Type **lsblk** and see the available volumes and we can see that there is no mount point for **xvdf**.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

LSBLK command

1. We'll run the **file** command. It detected the device **xvda** as a volume of root partition and **xvdf** as just data.

If result says “/dev/xvdf: data“, it means your volume is empty.

Pls cross verify it xvdb

A screenshot of a computer

Description automatically generated

The output of the file command is different for both volumes (**xvda** & **xvdf**). So, we’ll make xvdf a file system.

FILE command

1. Use **mkfs** to create a filesystem.

A screenshot of a computer program

Description automatically generated

**A screenshot of a computer

Description automatically generated**

## To format the volume to the ext4 filesystem

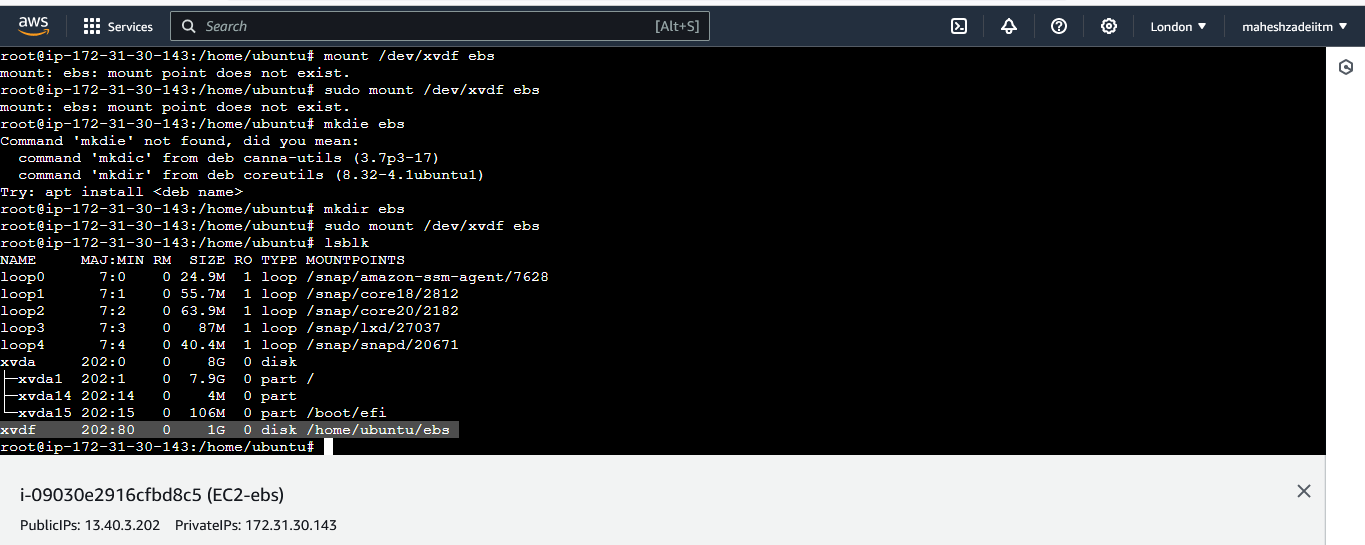
sudo mkfs -t ext4 /dev/xvdf

**A black and white text

Description automatically generated**

**A black and white screen with white text

Description automatically generated**

****

1. Now, there is a mount point for **xvdf**.

The mount command

Go to the newly created directory **ebs** and create a new text file.

**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1.txt is created. Now we have a new EBS volume attached to our EC2 instance.

**Step 3: Resize the Attached Volume and make sure it reflects in the connected instance**

Go to AWS Console -> EC2 -> Elastic Block Storage -> Volumes -> Modify Volume. Change the size from 1 Gib to 2 GiB and click on Modify.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Increase the Volume**

**1 to 2 Gib**

**A screenshot of a computer

Description automatically generated**

**Click on Modify**

**A screenshot of a computer error

Description automatically generated**

**Volume modified**

**A screenshot of a computer

Description automatically generated**

Now, we’ll check the device information. Here **lsblk** is showing 2 GiB but **df** is still showing 1 GiB.

**A screen shot of a computer

Description automatically generated**

## To check if the newly attached volume is having any data

sudo file -s /dev/xvdf

This step is for only verification

 The df command

 We’ll use the **resize2fs** command to reflect the changes.

**A screenshot of a computer program

Description automatically generated**

## To check the disk space

df -h

## To list available block devices

lsblk

**A screenshot of a computer

Description automatically generated**

Now **df** is showing 2 GiB and we can also see the 1.txt that was created in the previous step.

The resize2fs program

But if you restart your machine - this volume will be automatically unmount (detach)

To make it automount even after reboot, follow these steps

## Take the backup of the /etc/fstab file

sudo cp /etc/fstab /etc/fstab.bak

A black and white background with white text

Description automatically generated

##Actual Format ???

<device\_name> <mount\_point> <file\_system\_type> <fs\_mntops> <fs\_freq> <fs\_passno>

## Final run to check if there are any errors

sudo mount -a

If no errors - you have successfully done

Restart your machine and cross verify

And that's it! We've successfully launched a Linux EC2 instance, attached an EBS volume, and resized it to meet our requirements.