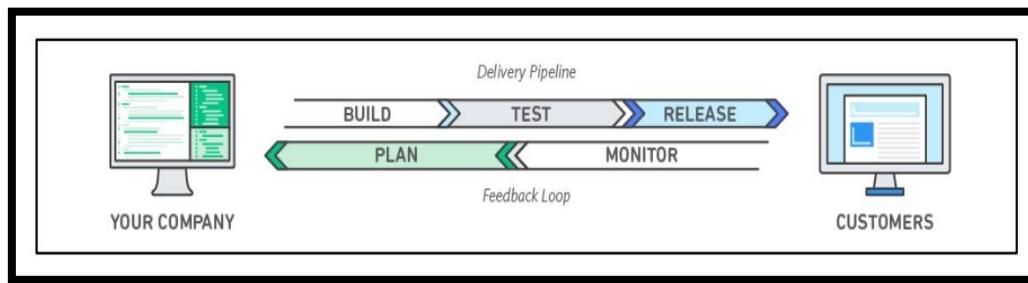


EXPERIMENT – O1

Q1. What is DevOps ?

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.



Q2. What is AWS EC2?Why EC2 ?

Amazon Elastic Compute Cloud (EC2) is a part of Amazon.com's cloud-computing platform, Amazon Web Services, that allows users to rent virtual computers on which to run their own computer applications. Amazon EC2 provides the following features:

- ▲ Virtual computing environments, known as instances.
- ▲ Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software).
- ▲ Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types.
- ▲ Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place).

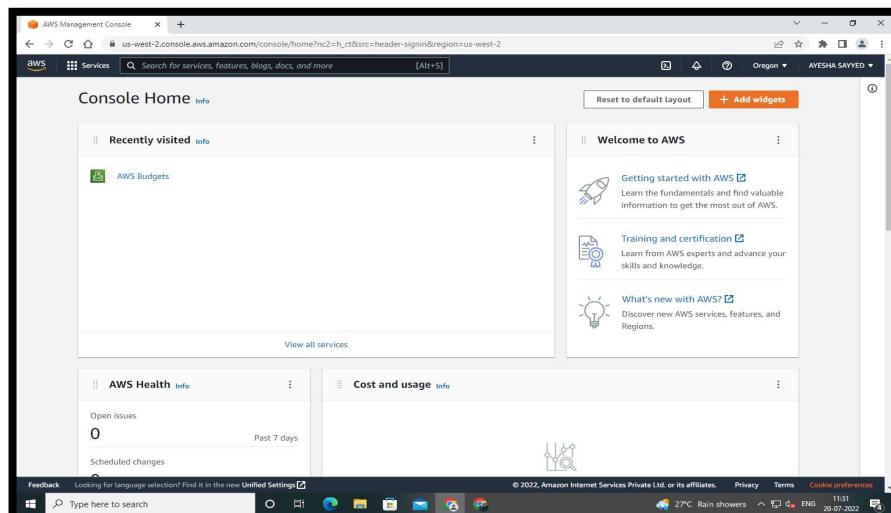
- Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as instance store volumes .
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes.
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones.
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups.
- Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses.
- Metadata, known as tags, that you can create and assign to your Amazon EC2 resources.
- Virtual networks you can create that are logically isolated from the rest of the AWS Cloud, and that you can optionally connect to your own network, known as virtual private clouds (VPCs).

Q3. Launch two instances of AWS EC2, one windows another ubuntu . Get connected to instances using RDP and MobaXterm client software. Explain each step of EC2 creation and launching with the help of screenshots.

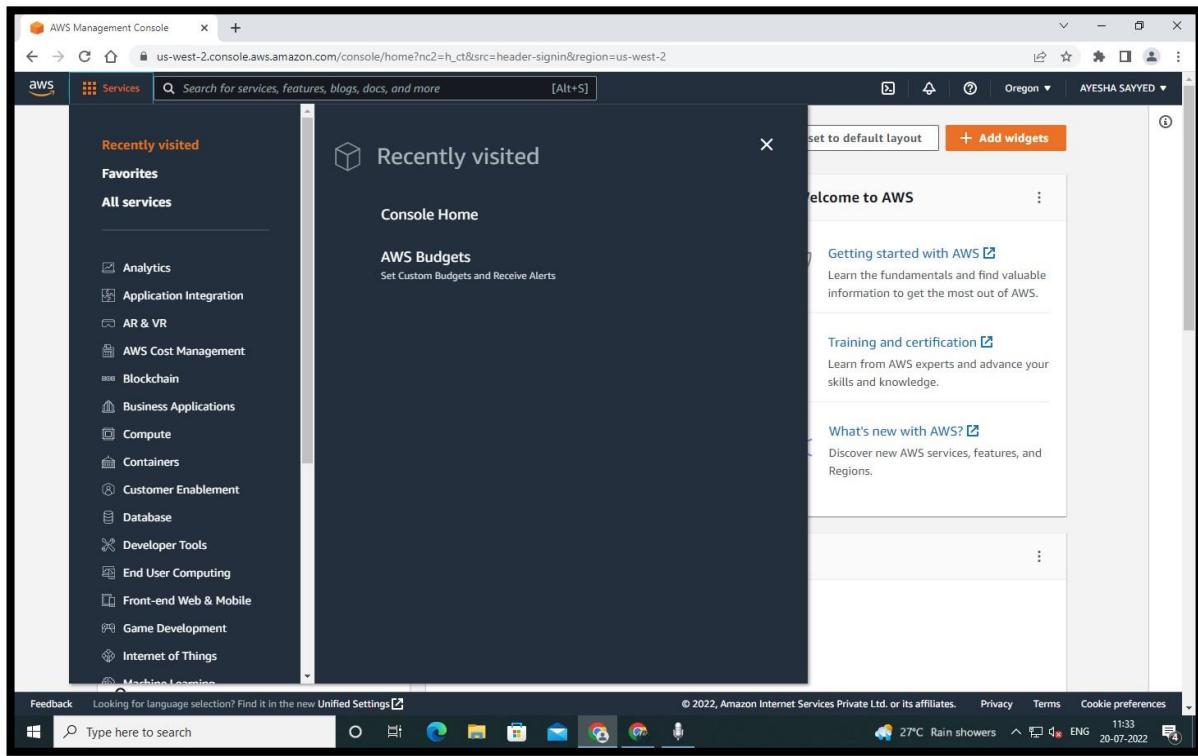
Open google.com from both the instances ,search your own name .

→[A] [WINDOWS INSTANCE]

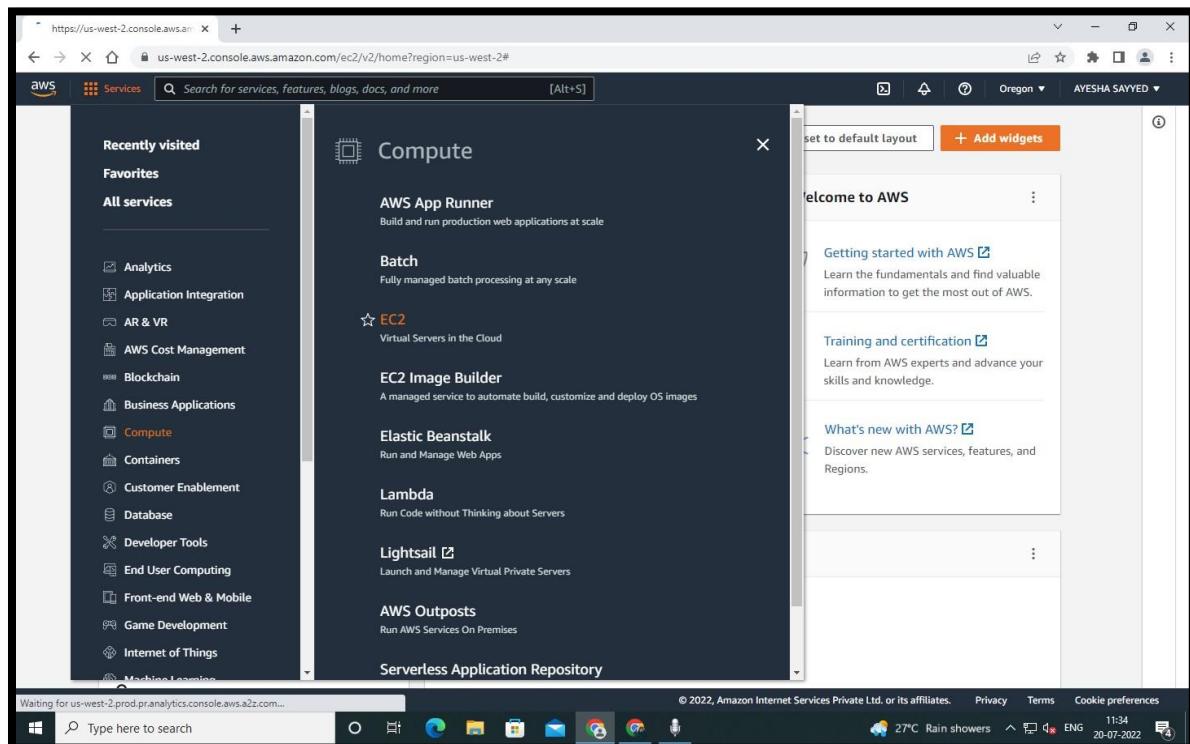
STEP 1 : AWS Management Console Dashboard



STEP 2 : Click on Services.



STEP 3 : Now CLICK on Compute -> EC2.



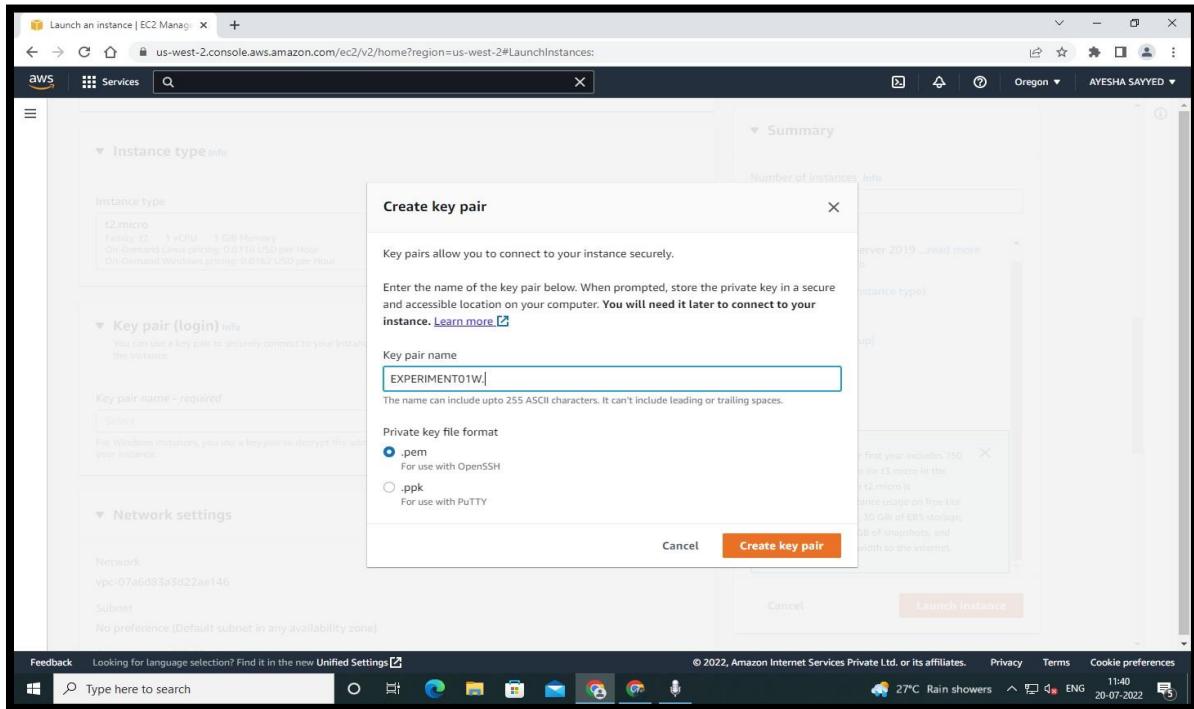
STEP 4 : Launch Instance

The screenshot shows the AWS EC2 Management Console with the URL <https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard>. The left sidebar is expanded to show 'Instances' and 'Launch Templates'. The main area displays 'Resources' and 'Account attributes'. A central box titled 'Launch instance' contains a large orange 'Launch instance' button. To the right, there's a 'Service health' section showing 'Region: US West (Oregon)' and 'Status: This service is operating normally'. A sidebar on the right lists 'Explore AWS' options like 'Save Up to 45% on ML Inference' and 'Get Up to 40% Better Price Performance'. The bottom status bar shows the date '20-07-2022' and time '11:35'.

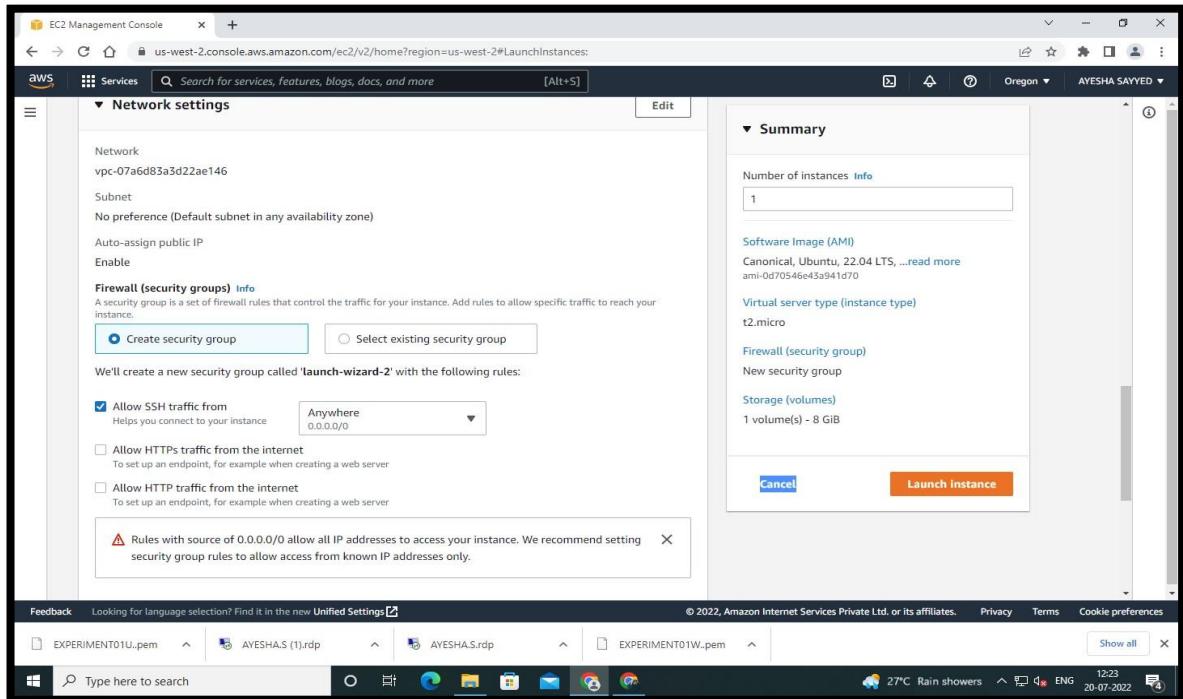
STEP 5 : Select Windows

The screenshot shows the 'Launch an instance' wizard with the URL <https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstances>. The left sidebar shows 'EC2 > Instances > Launch an instance'. The main area has a 'Summary' section with 'Number of instances: 1'. Below it, under 'Application and OS Images (Amazon Machine Image)', there's a search bar and a 'Quick Start' section with icons for Amazon Linux, Ubuntu, Windows, Red Hat, and SUSE Linux. A callout box highlights the 'Free tier' information: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' The bottom status bar shows the date '20-07-2022' and time '11:37'.

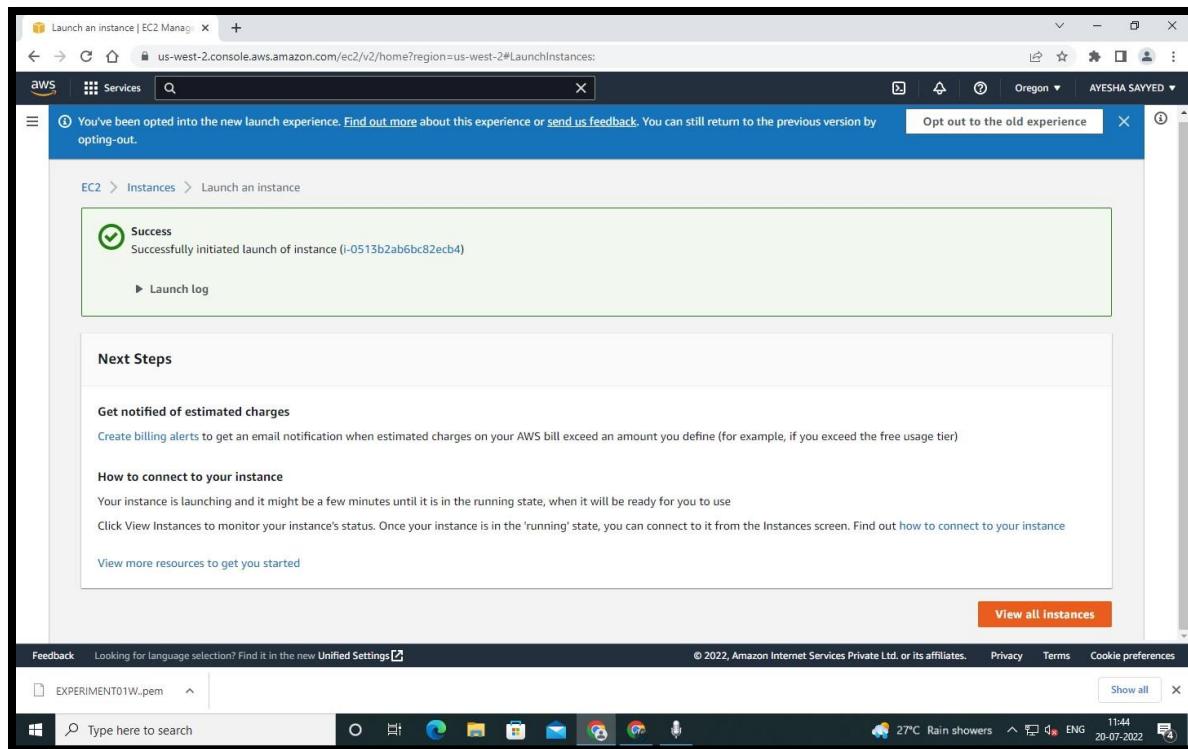
STEP 6: Create key pair. A .pem file will be downloaded which will be later used to connect to the instance.



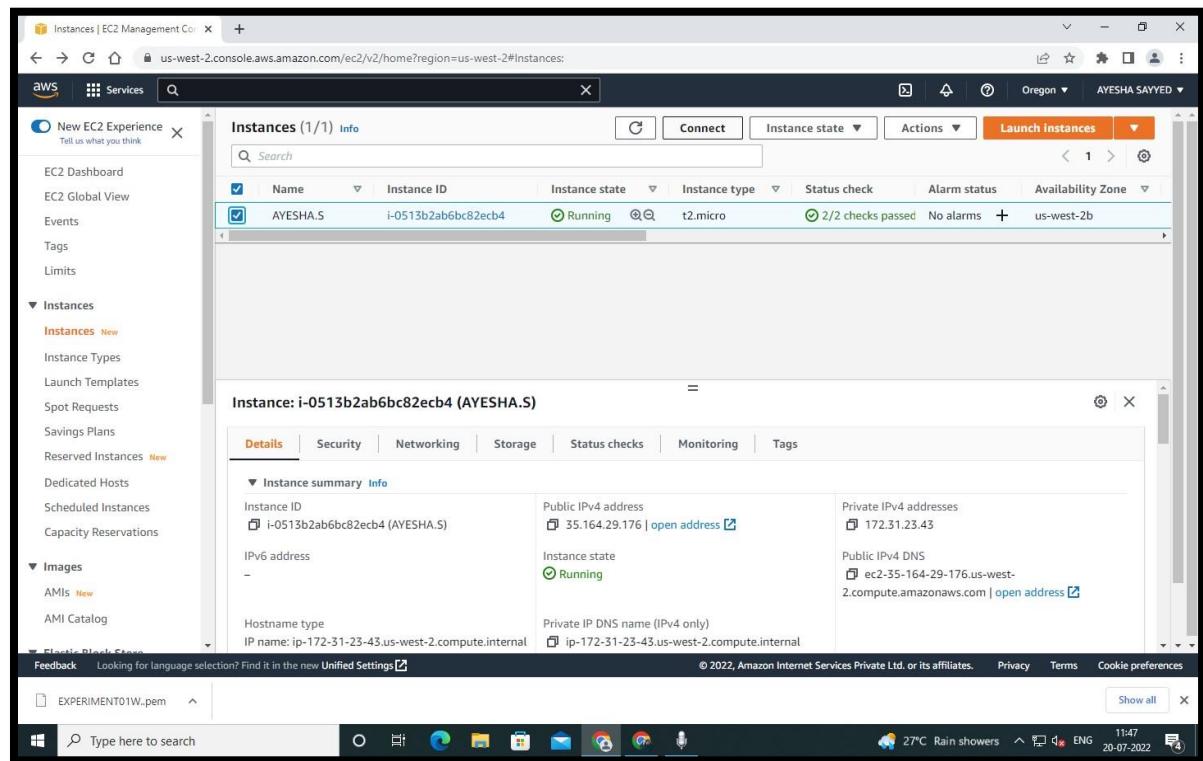
STEP 7: Network Settings: Select ‘Allow RDP Traffic from’ Anywhere. Then Launch Instance.



Success message will be shown after successful creation of instance

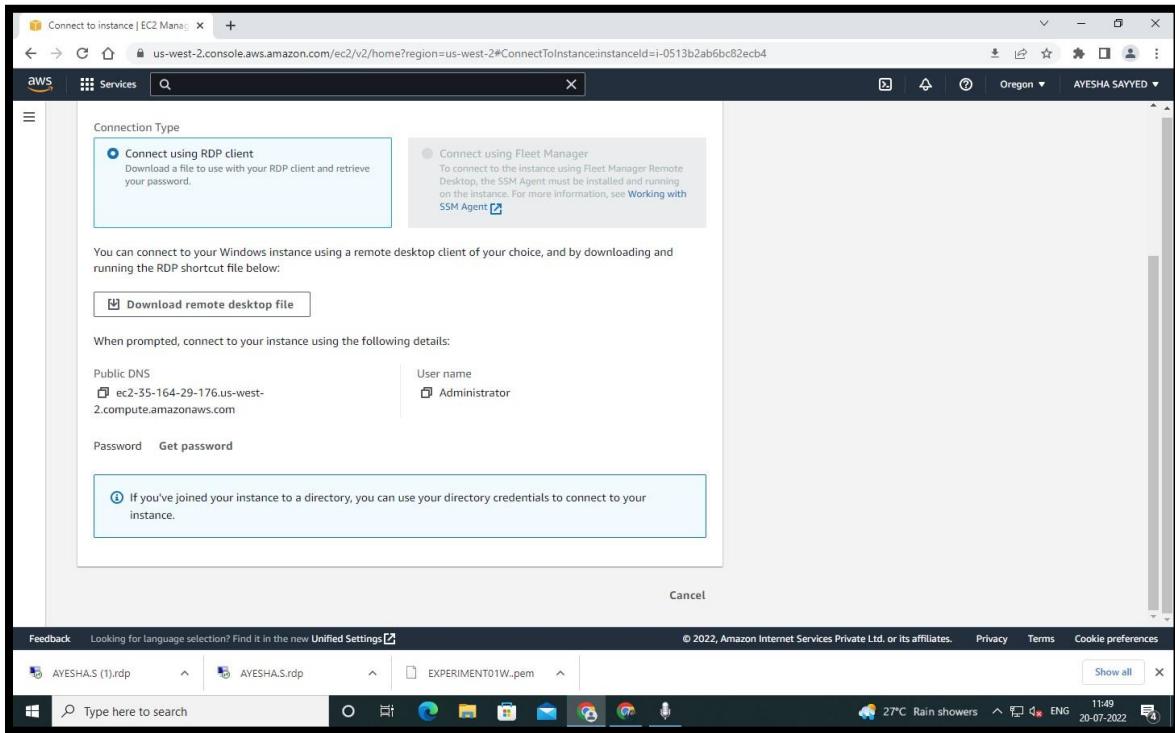


STEP 8: RDP Connection We have to download Remote Desktop File to connect to the instance.

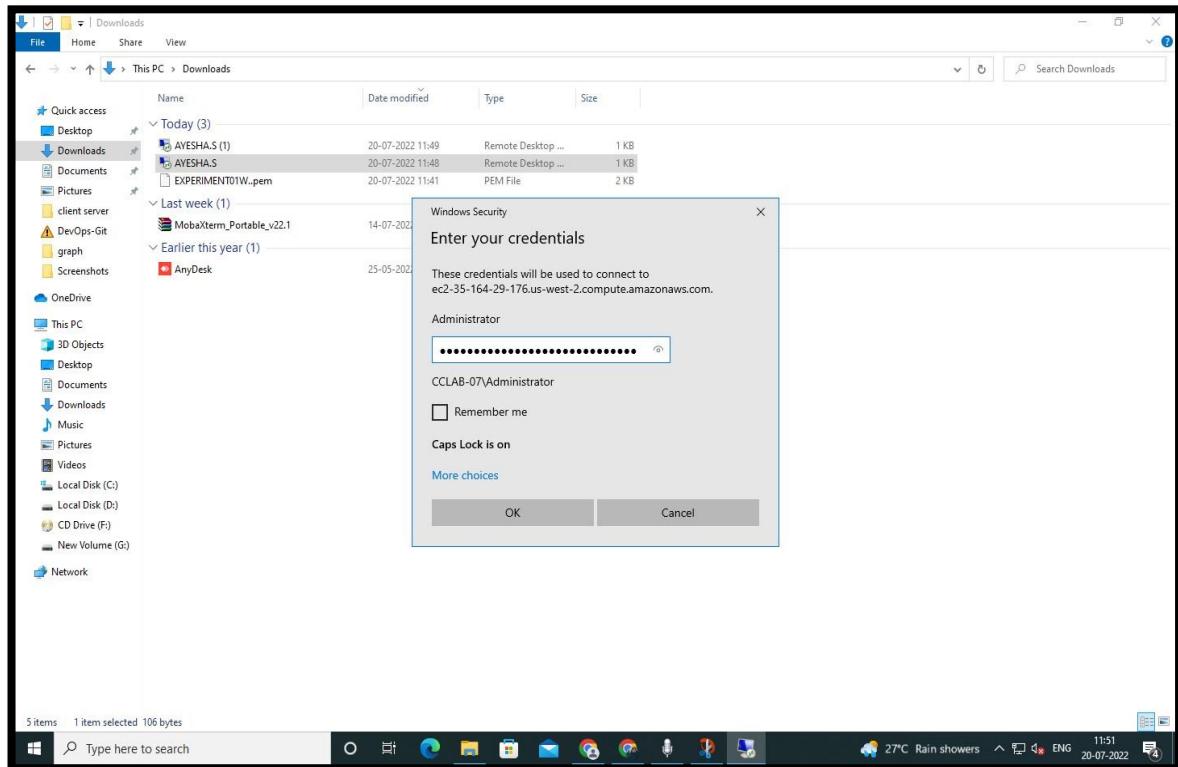


Go to RDP Client to download the Remote Desktop File

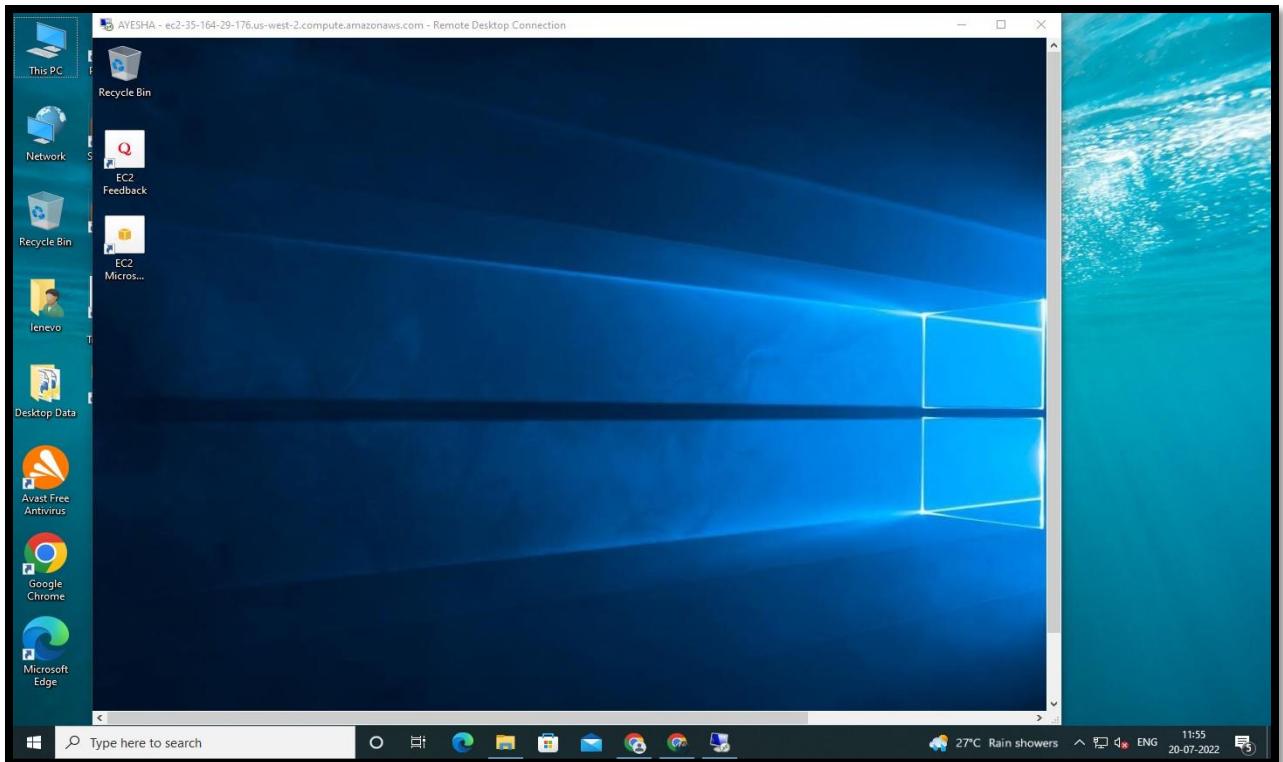
Using the .pem file downloaded earlier, decrypt the password and copy it.



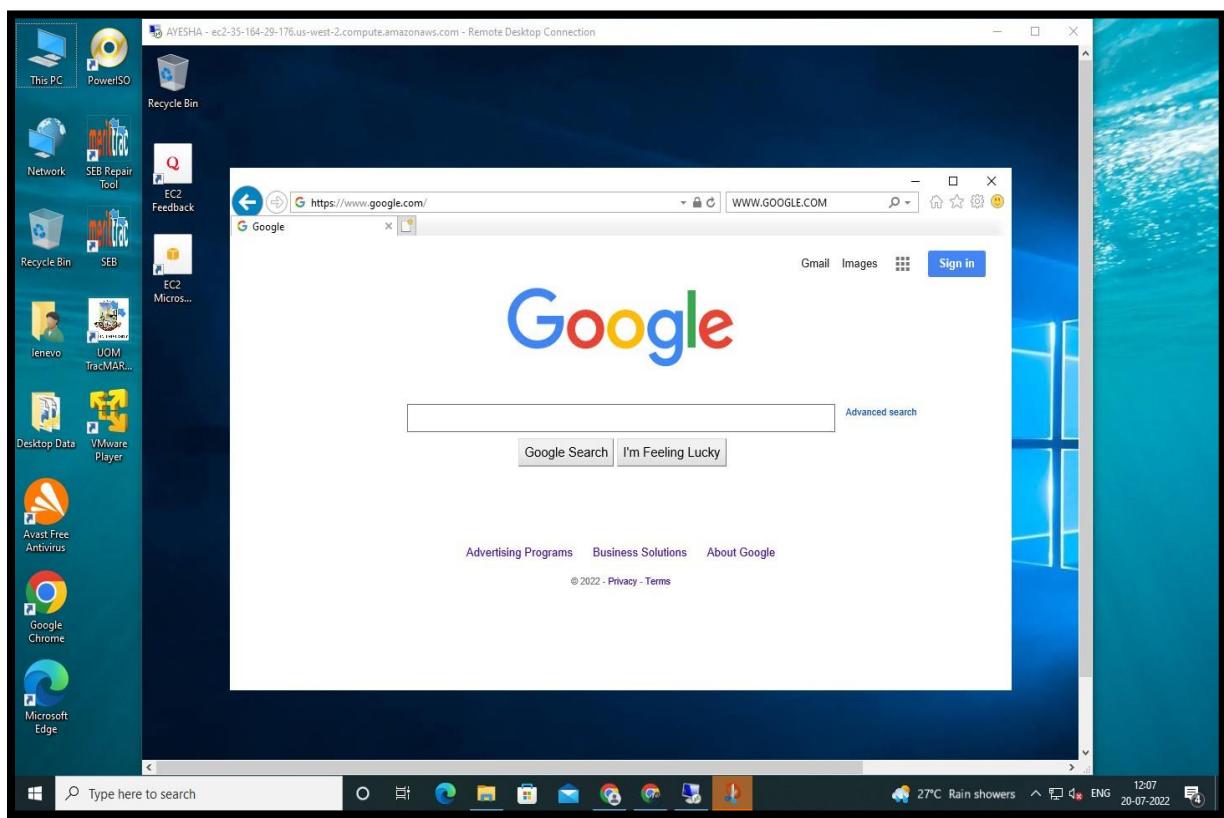
Now, run the Remote Desktop File and put the decrypted password copied earlier



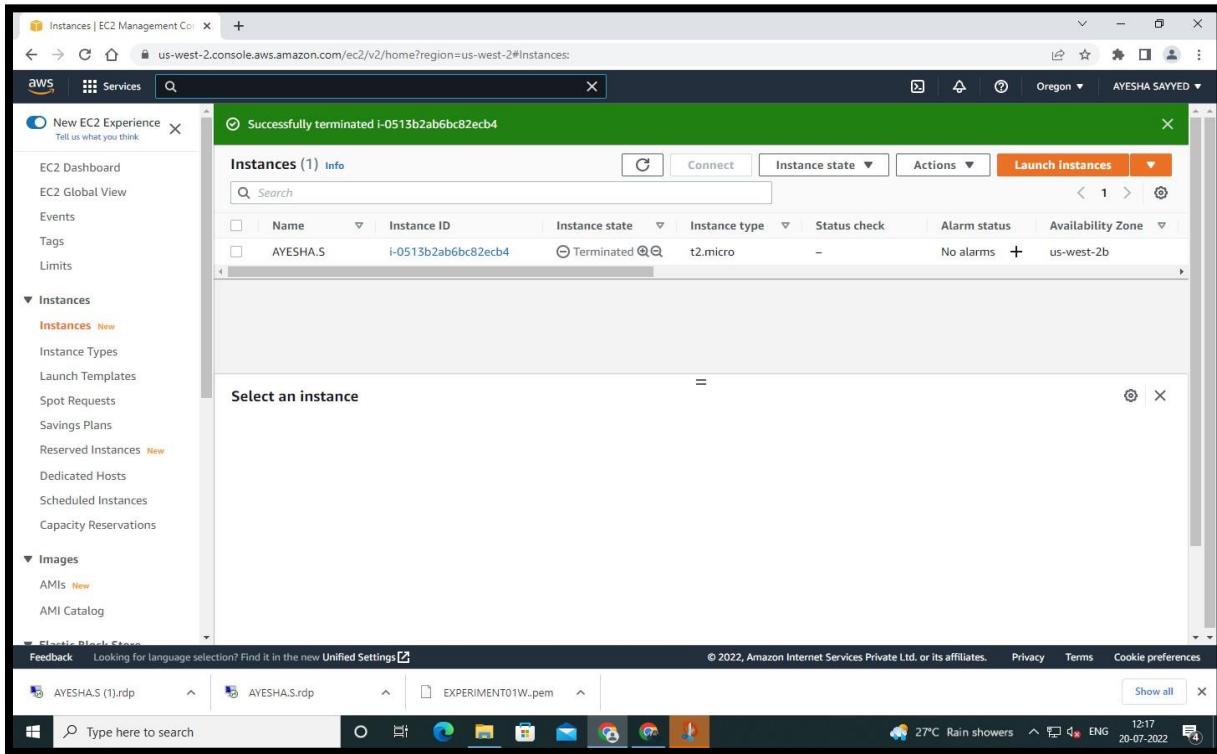
Your Windows Instance will start running.



STEP 9 : Open google.com to search your name.

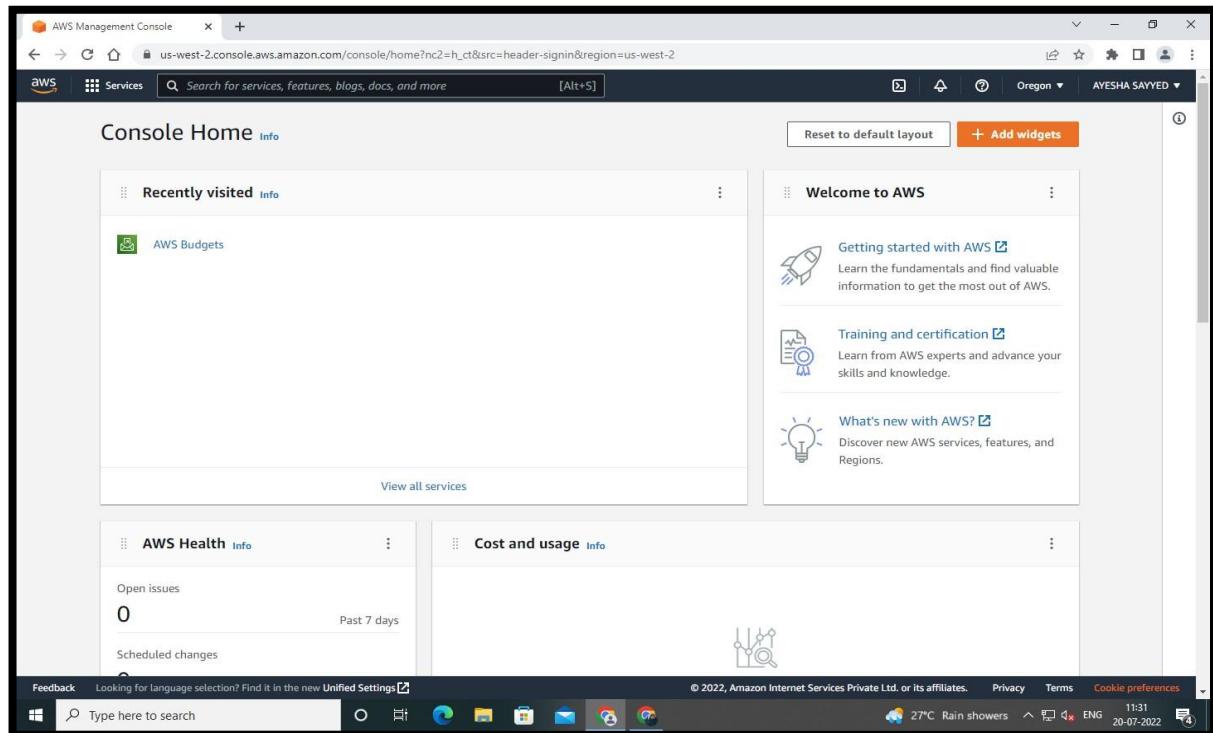


STEP 10 : Terminate the instance if you don't wish to use it again.



→[B] [UBUNTU INSTANCE]

STEP 1: AWS Management Console Dashboard



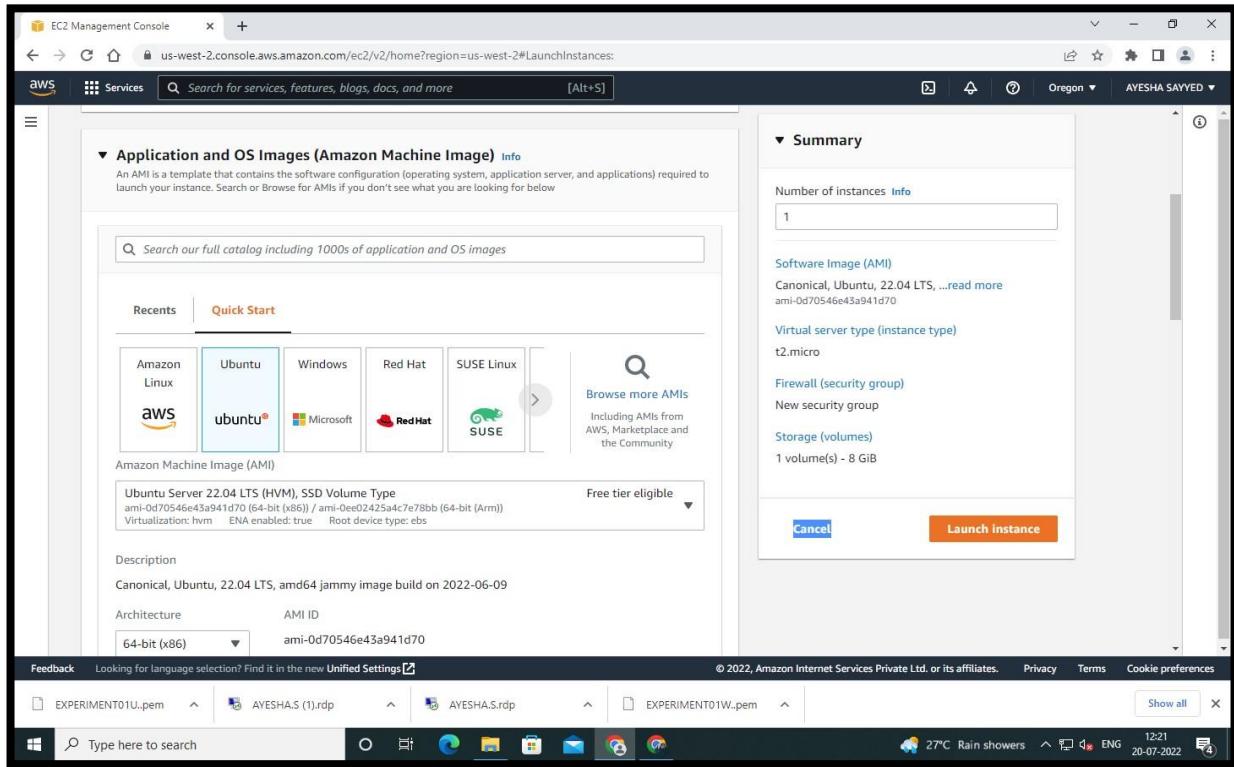
STEP 2 : CLICK on Services ->Compute ->EC2.

The screenshot shows the AWS Management Console with the URL <https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#>. The left sidebar is open, showing 'Recently visited', 'Favorites', and a list of 'All services' under the 'Compute' category. The main content area is titled 'Compute' and lists several services: AWS App Runner, Batch, EC2, EC2 Image Builder, Elastic Beanstalk, Lambda, Lightsail, AWS Outposts, and Serverless Application Repository. A 'Welcome to AWS' sidebar on the right provides links to 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. The bottom status bar shows the date as 20-07-2022 and the time as 11:34.

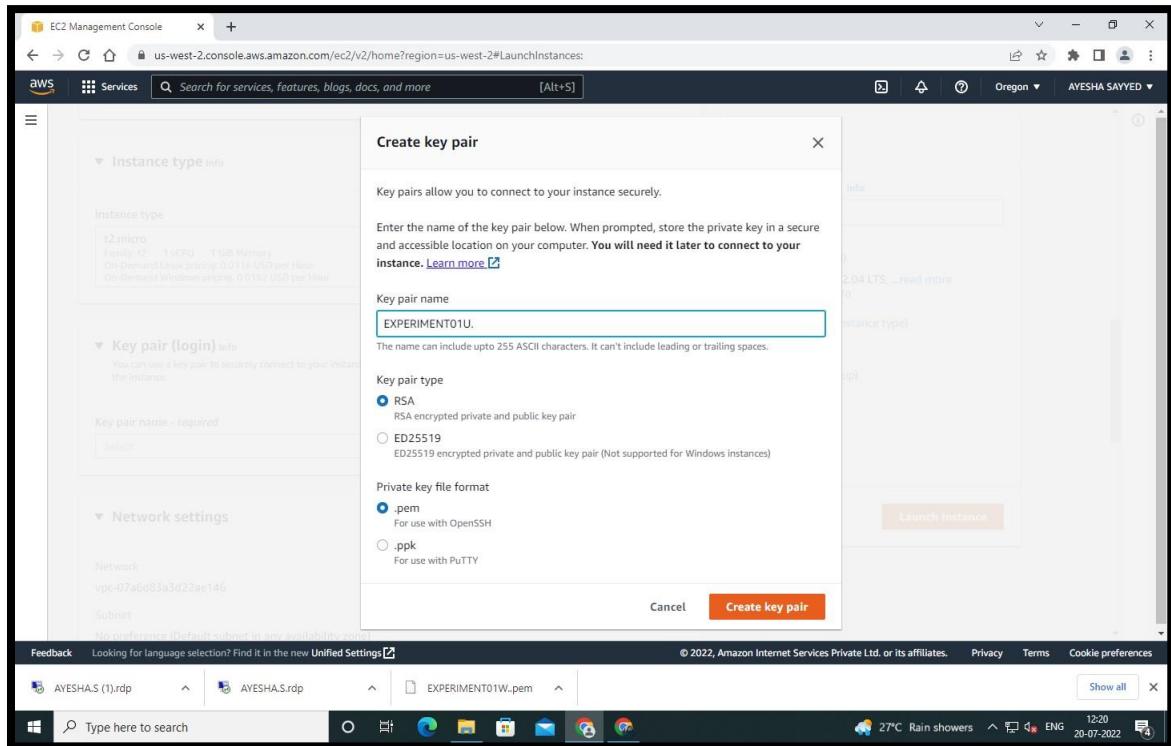
STEP 3 : Launch Instance.

The screenshot shows the AWS EC2 Management Console with the URL <https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Home>. The left sidebar is open, showing 'New EC2 Experience' (with a 'Tell us what you think' link), 'EC2 Dashboard' (with 'EC2 Global View', 'Events', 'Tags', 'Limits'), 'Instances' (with 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations'), 'Images' (with 'AMIs', 'AMI Catalog'), and 'Elastic Block Store' (with 'Volumes'). The main content area has three main sections: 'Resources' (listing 0 instances running, 0 dedicated hosts, 0 elastic IPs, 0 instances, 0 key pairs, 0 placement groups, 0 security groups, 0 snapshots, 0 volumes), 'Launch instance' (with a large 'Launch instance' button and a note about launching in the US West (Oregon) Region), and 'Service health' (showing the region as US West (Oregon) and a status message: 'This service is operating normally'). A 'Account attributes' sidebar on the right lists supported platforms (VPC), default VPC (vpc-07a6d83a3d22ae146), settings (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments), and explore AWS offers (Save Up to 45% on ML Inference, Get Up to 40% Better Price Performance). The bottom status bar shows the date as 20-07-2022 and the time as 11:35.

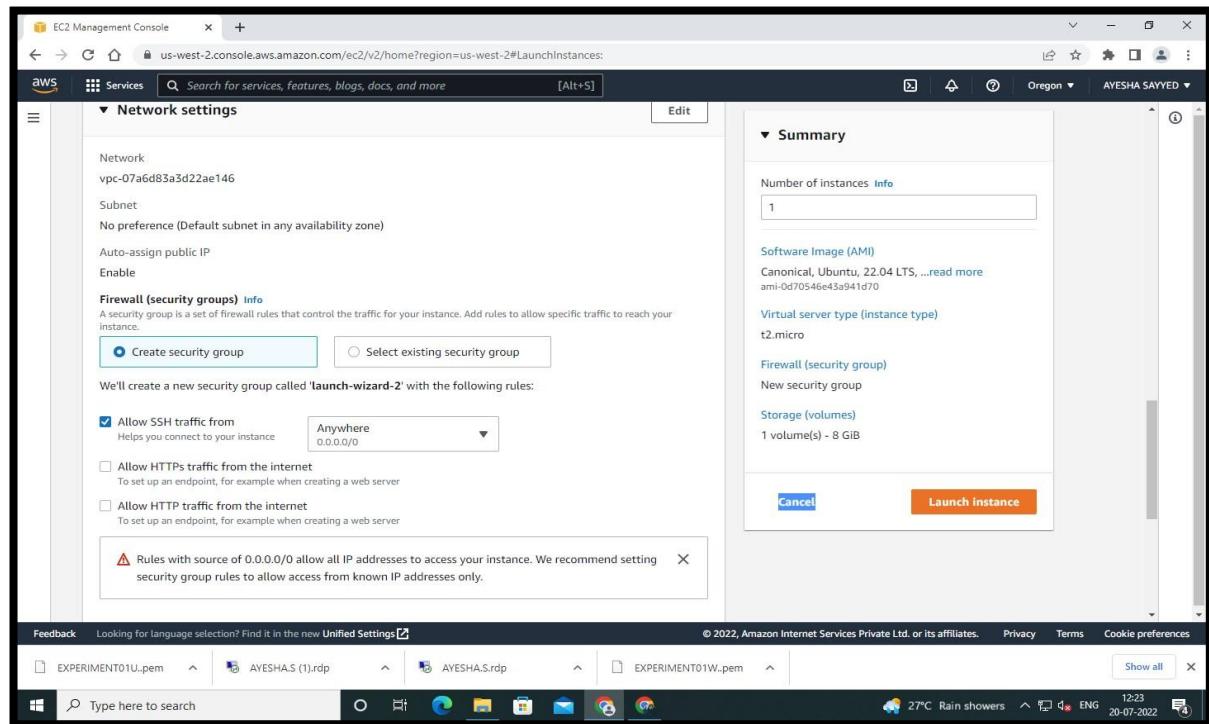
STEP 4 : Select Ubuntu



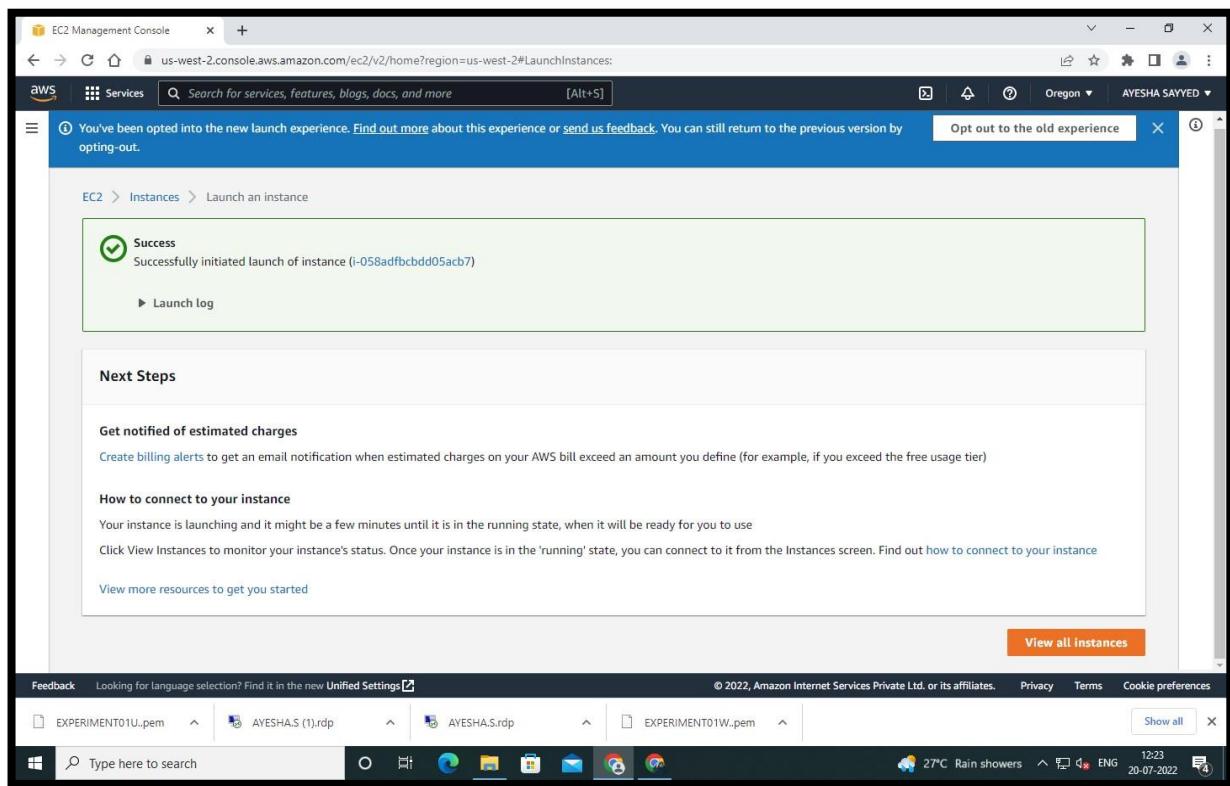
STEP 5 : Create key pair. A .pem file will be downloaded which will be later used to connect to the instance.



STEP 6 : Network Settings: Select ‘Allow SSH Traffic from’ Anywhere. Then Launch Instance.



Success message will be shown after successful creation of instance.



STEP 7: Launch Ubuntu instance to get the remote host and username for SSH in MobaXterm.

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with Instances selected), Images, AMIs, and Classic Block Store. The main area shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
S.AYESHA	i-058adfbcbdd05acb7	Running	t2.micro	Initializing	No alarms	us-west-2b
AYESHA.S	i-0513b2ab6bc82ecb4	Terminated	t2.micro	-	No alarms	us-west-2b

Below the table, a detailed view for instance i-058adfbcbdd05acb7 (S.AYESHA) is displayed. It shows the following details:

- Instance ID: i-058adfbcbdd05acb7 (S.AYESHA)
- Public IPv4 address: 34.221.202.91 | open address
- Private IP4 addresses: 172.31.20.159
- Public IPv4 DNS: ec2-34-221-202-91.us-west-2.compute.amazonaws.com | open address
- IPv6 address: -
- Instance state: Running
- Hostname type: IP name: ip-172-31-20-159.us-west-
- Private IP DNS name (IPv4 only): ip-172-31-20-159.us-west-2.compute.internal

Go to SSH Client to get the remote host and username.

The screenshot shows the 'Connect to instance' dialog for instance i-058adfbcbdd05acb7 (S.AYESHA). The 'SSH client' tab is selected. The dialog provides the following instructions:

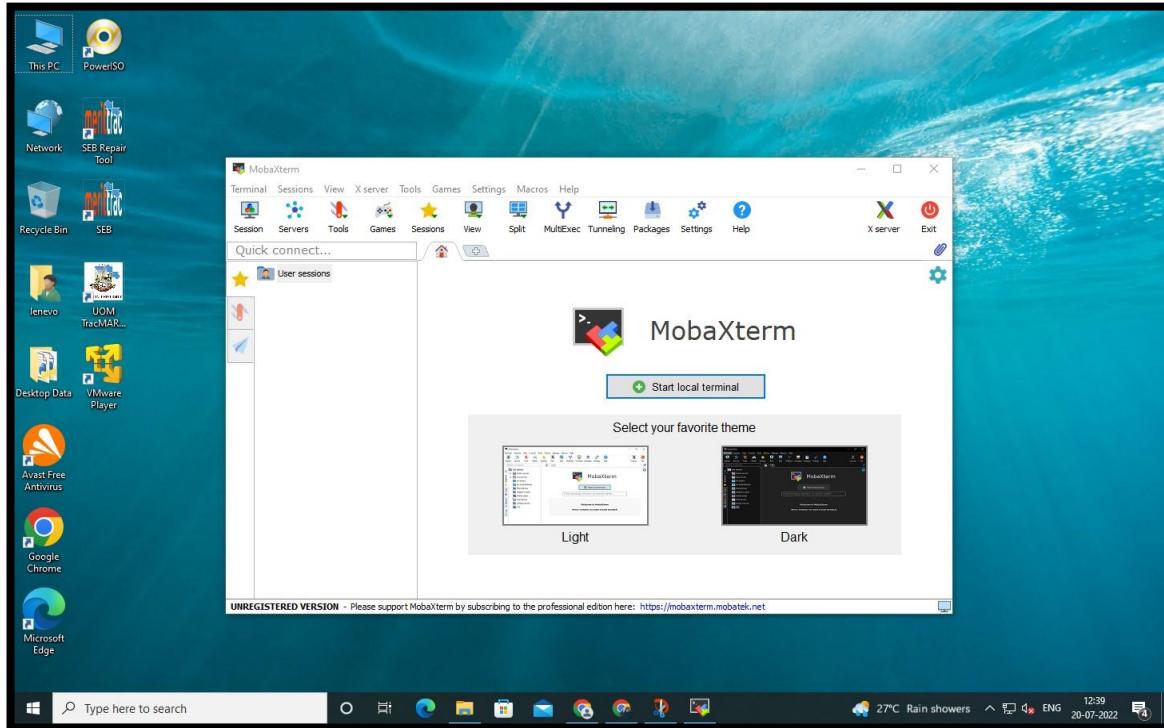
- Open an SSH client.
- Locate your private key file. The key used to launch this instance is EXPERIMENT01U.pem.
- Run this command, if necessary, to ensure your key is not publicly viewable:
chmod 400 EXPERIMENT01U.pem
- Connect to your instance using its Public DNS:
ec2-34-221-202-91.us-west-2.compute.amazonaws.com

Example command:
ssh -i "EXPERIMENT01U.pem" ubuntu@ec2-34-221-202-91.us-west-2.compute.amazonaws.com

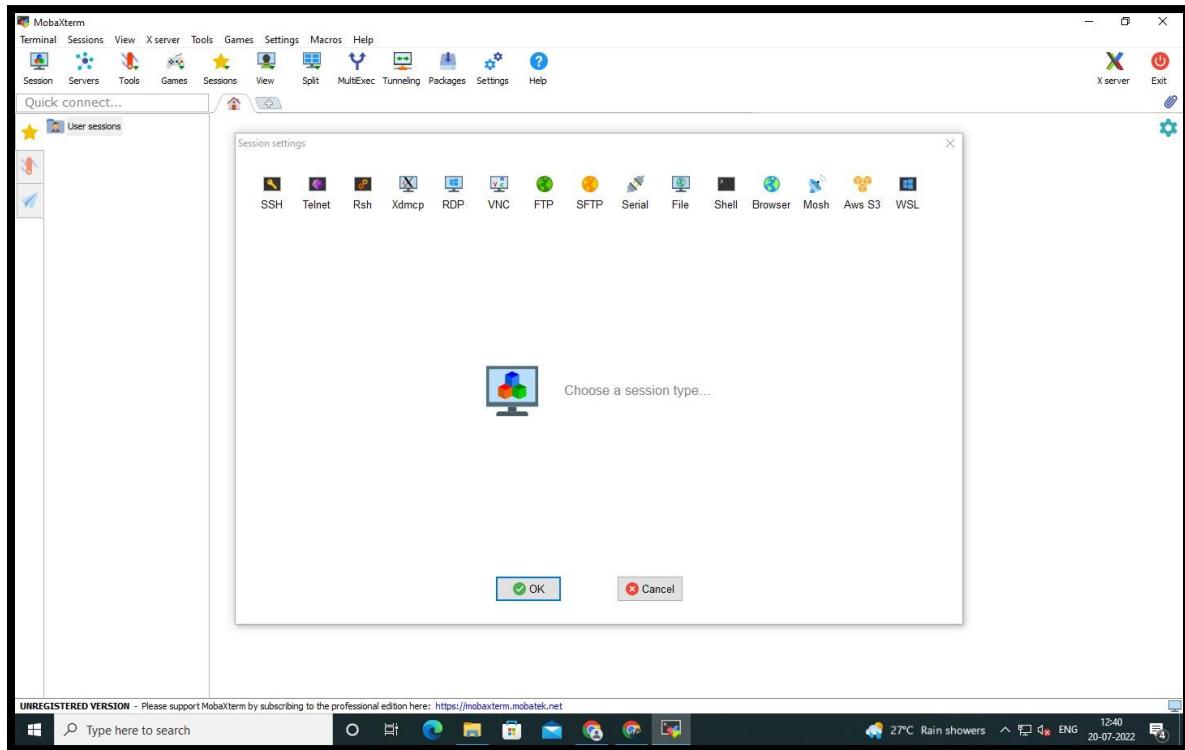
Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Copy the Public DNS which is your remote host and the username is the word before '@'

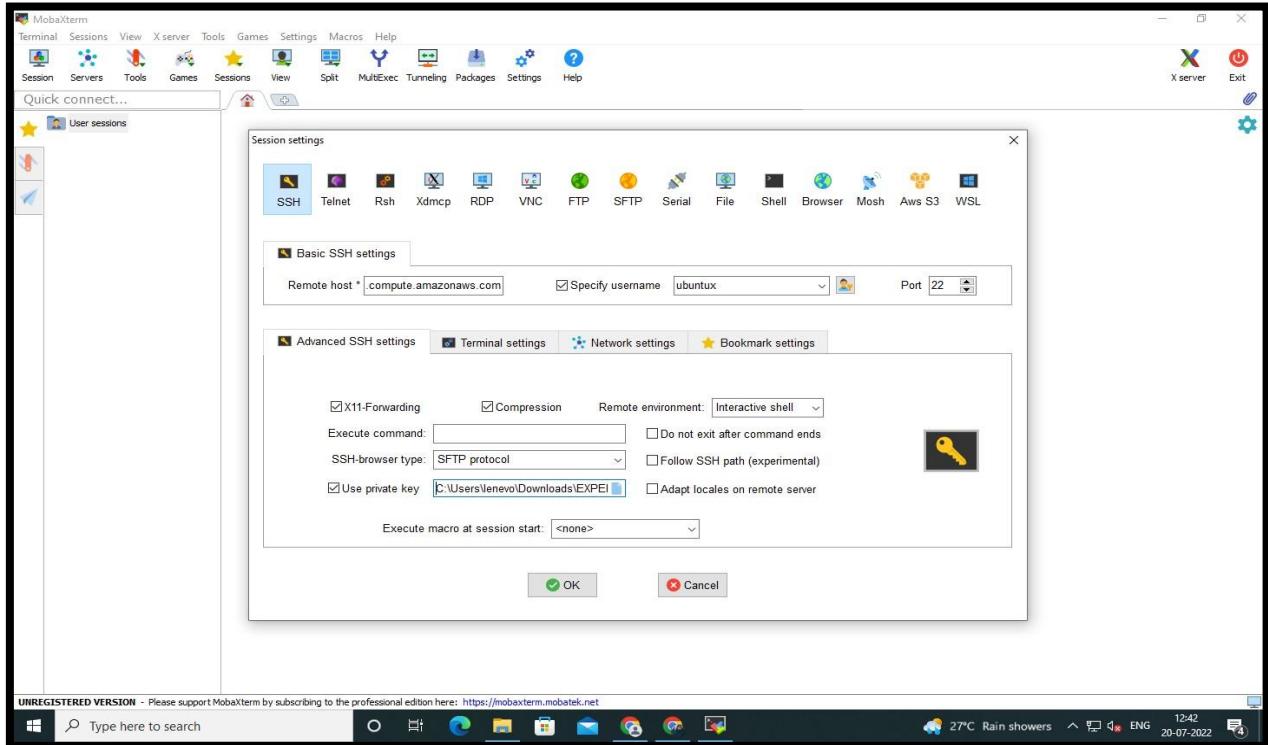
STEP 9 : Download MobaXterm to connect the instance



Run MobaXterm. Go to Sessions → New Session.



Then select SSH Fill the basic SSH settings and attach the .pem file downloaded earlier in advanced SSH settings ‘Use private key’ section → Then OK.



Your Ubuntu Instance will be running Execute few commands as follows – [sudo su sudo apt update].

```

MobaXterm Personal Edition v22.1 *
(SSH client, X server and network tools)

SSH session to ubuntu@ec2-34-221-202-91.us-west-2.compute.amazonaws.com
  • Direct SSH : ✓
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✓ (remote display is forwarded through SSH)
  • For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Wed Jul 20 07:16:17 UTC 2022

System load: 0.0 Processes: 100
Usage of /: 19.0% of 7.58GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.20.159
Swap usage: 0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth: file /home/ubuntu/.xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

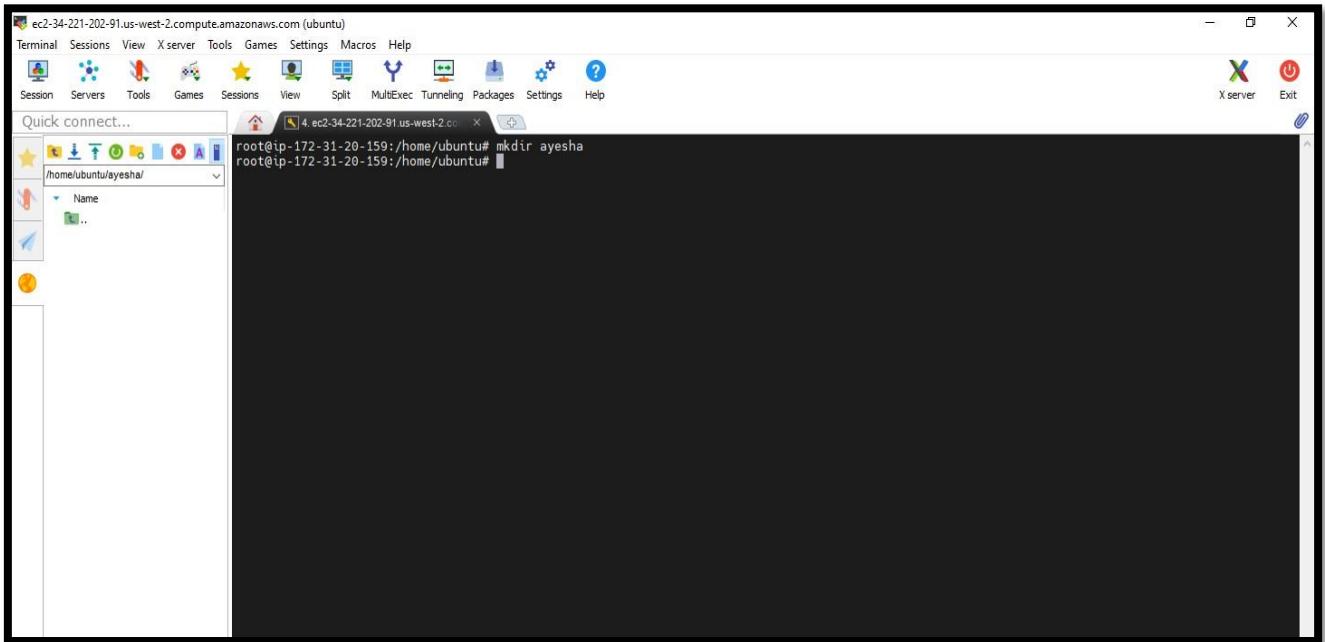
ubuntu@ip-172-31-20-159:~$ sudo su
root@ip-172-31-20-159:/home/ubuntu#

```

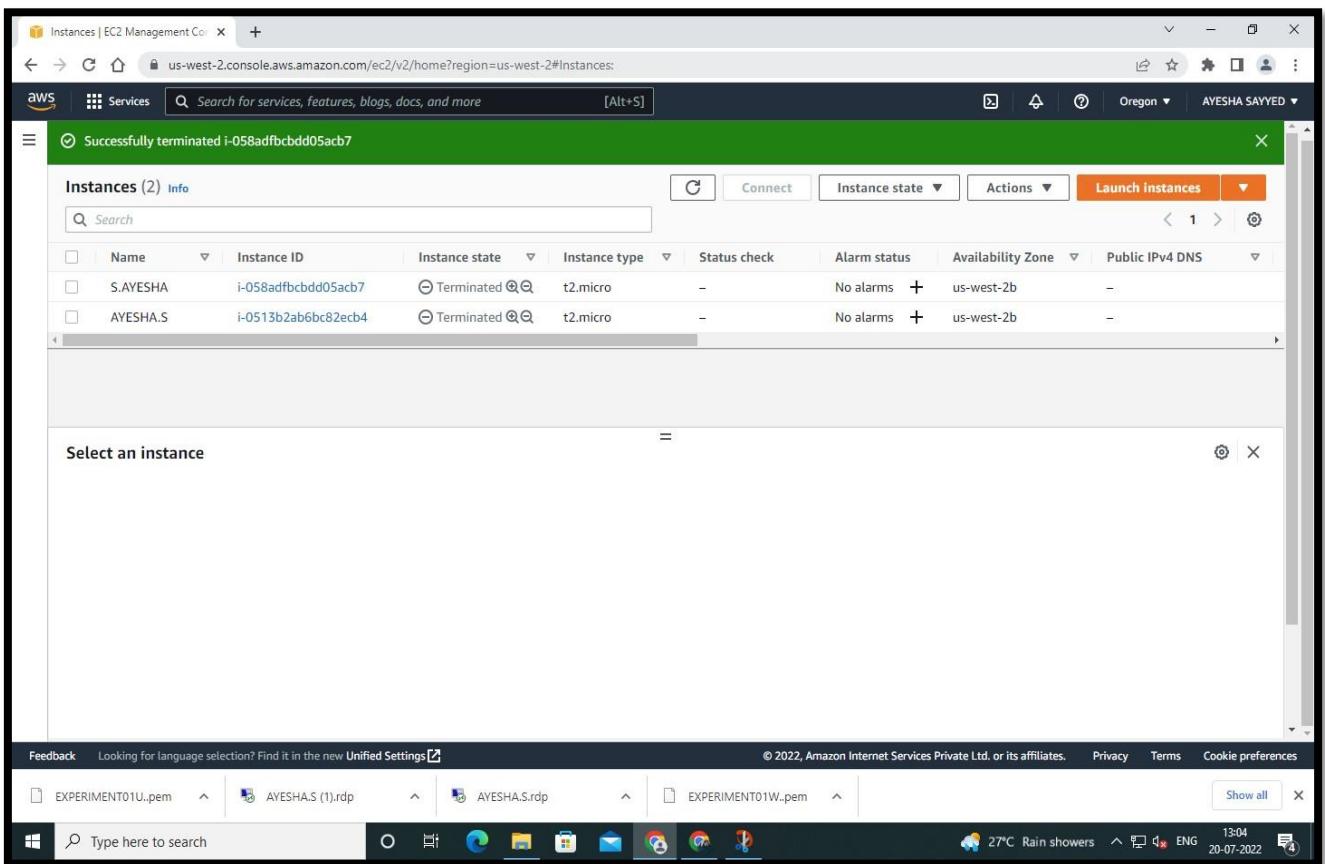
```
ec2-34-221-202-91.us-west-2.compute.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 4. ec2-34-221-202-91.us-west-2.compute.amazonaws.com (ubuntu)
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
E: Unable to locate package update
root@ip-172-31-20-159:/home/ubuntu# sudo apt update
sudo: apt: command not found
root@ip-172-31-20-159:/home/ubuntu# sudo apt update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [247 kB]
Get:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 kB]
Get:11 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [378 kB]
Get:12 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [94.8 kB]
Get:13 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [6688 kB]
Get:14 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [238 kB]
Get:15 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [36.4 kB]
Get:16 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 kB]
Get:17 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [171 kB]
Get:18 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [60.6 kB]
Get:19 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [4160 kB]
Get:20 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [7900 kB]
Get:21 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [2112 kB]
Get:22 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [420 kB]
Get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [112 kB]
Get:24 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 kB]
Get:25 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [5412 kB]
Get:26 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [8160 kB]
Get:27 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [236 kB]
Get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 kB]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [227 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [55.3 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [3564 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [203 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [30.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [512 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [93.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [33.0 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [2104 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4192 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [900 B]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [228 B]
Fetched 22.4 MB in 3s (6400 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
46 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-20-159:/home/ubuntu# sudo apt-get install chromium-browser
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
chromium-browser
0 upgraded, 1 newly installed, 0 to remove and 46 not upgraded.
Need to get 48.4 kB of archives.
After this operation, 164 kB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 chromium-browser amd64 1:85.0.4183.83~Ubuntu2 [48.4 kB]
Fetched 48.4 kB in 0s (544 kB/s)
Preconfiguring packages...
Selecting previously unselected package chromium-browser.
(Reading database ... 63612 files and directories currently installed.)
Preparing to unpack .../chromium-browser_1%3a85.0.4183.83~Ubuntu2_amd64.deb ...
⇒ Installing the chromium snap
⇒ Checking connectivity with the snap store
⇒ Installing the chromium snap
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search 27°C Rain showers 12:49 20-07-2022
```

```
ec2-34-221-202-91.us-west-2.compute.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 4. ec2-34-221-202-91.us-west-2.compute.amazonaws.com (ubuntu)
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
Get:18 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [60.6 kB]
Get:19 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [4160 kB]
Get:20 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [7900 kB]
Get:21 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [2112 kB]
Get:22 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [420 kB]
Get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [112 kB]
Get:24 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 kB]
Get:25 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [5412 kB]
Get:26 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [8160 kB]
Get:27 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [236 kB]
Get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 kB]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [227 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [55.3 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [3564 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [203 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [30.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [512 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [93.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [33.0 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [2104 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4192 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [900 B]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [228 B]
Fetched 22.4 MB in 3s (6400 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
46 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-20-159:/home/ubuntu# sudo apt-get install chromium-browser
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
chromium-browser
0 upgraded, 1 newly installed, 0 to remove and 46 not upgraded.
Need to get 48.4 kB of archives.
After this operation, 164 kB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 chromium-browser amd64 1:85.0.4183.83~Ubuntu2 [48.4 kB]
Fetched 48.4 kB in 0s (544 kB/s)
Preconfiguring packages...
Selecting previously unselected package chromium-browser.
(Reading database ... 63612 files and directories currently installed.)
Preparing to unpack .../chromium-browser_1%3a85.0.4183.83~Ubuntu2_amd64.deb ...
⇒ Installing the chromium snap
⇒ Checking connectivity with the snap store
⇒ Installing the chromium snap
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search 27°C Rain showers 12:50 20-07-2022
```

Create a directory of your name.



STEP 10: Terminate the instance if you don't wish to use it again.



EXPERIMENT - 02

Q1. What is S3? Explain Uses of S3 ?

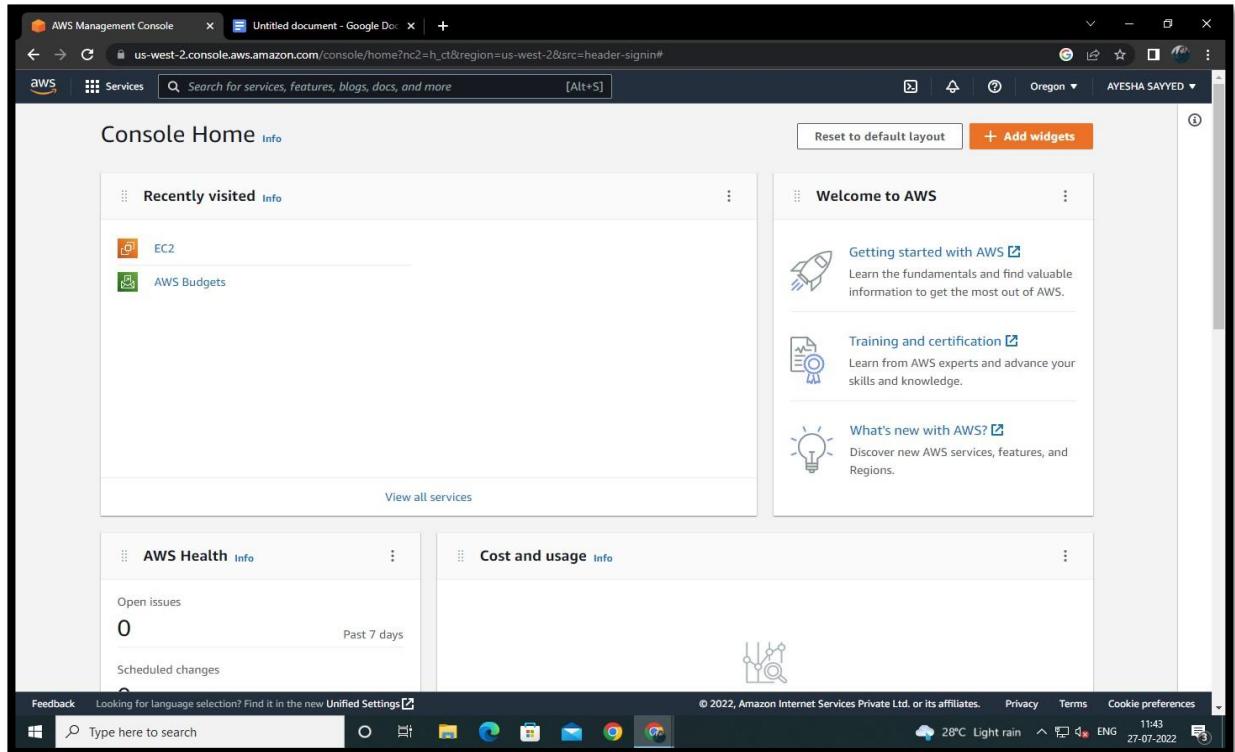
- ▲ Amazon S3 is an object storage service that stores data as objects within buckets.
- ▲ An object is a file and any metadata that describes the file. A bucket is a container for objects.
- ▲ Amazon Simple Storage Service (Amazon S3) is a scalable, high-speed, web-based cloud storage service.
- ▲ The service is designed for online backup and archiving of data and applications on Amazon Web Services (AWS).
- ▲ Amazon S3 was designed with a minimal feature set and created to make web-scale computing easier for developers.
- ▲ Amazon S3 can be used by organizations ranging in size from small businesses to large enterprises.
- ▲ S3's scalability, availability, security and performance capabilities make it suitable for a variety of data storage use cases.

Common use cases for S3 include the following: -

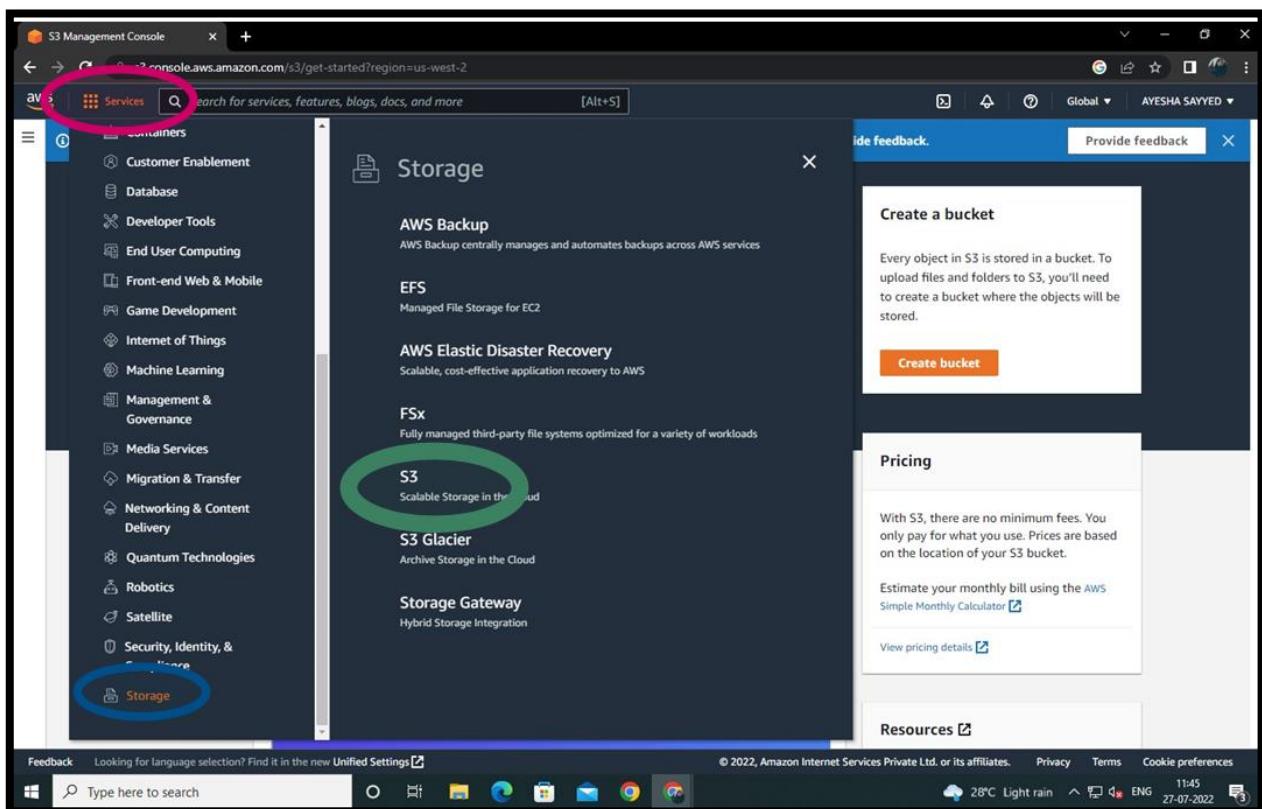
- Data storage
- Data archiving
- Application hosting for deployment, installation and management of web apps
- Software delivery
- Data backup
- Disaster recovery (DR)
- Running big data analytics tools on stored data
- Data lakes
- Mobile applications
- Internet of things (IoT) devices
- Media hosting for images, videos and music files ✓ Website hosting -- particularly well suited to work with Amazon CloudFront for content delivery.

Q2. Deployment of static web site on AWS S3 [Screen shots and steps are required].

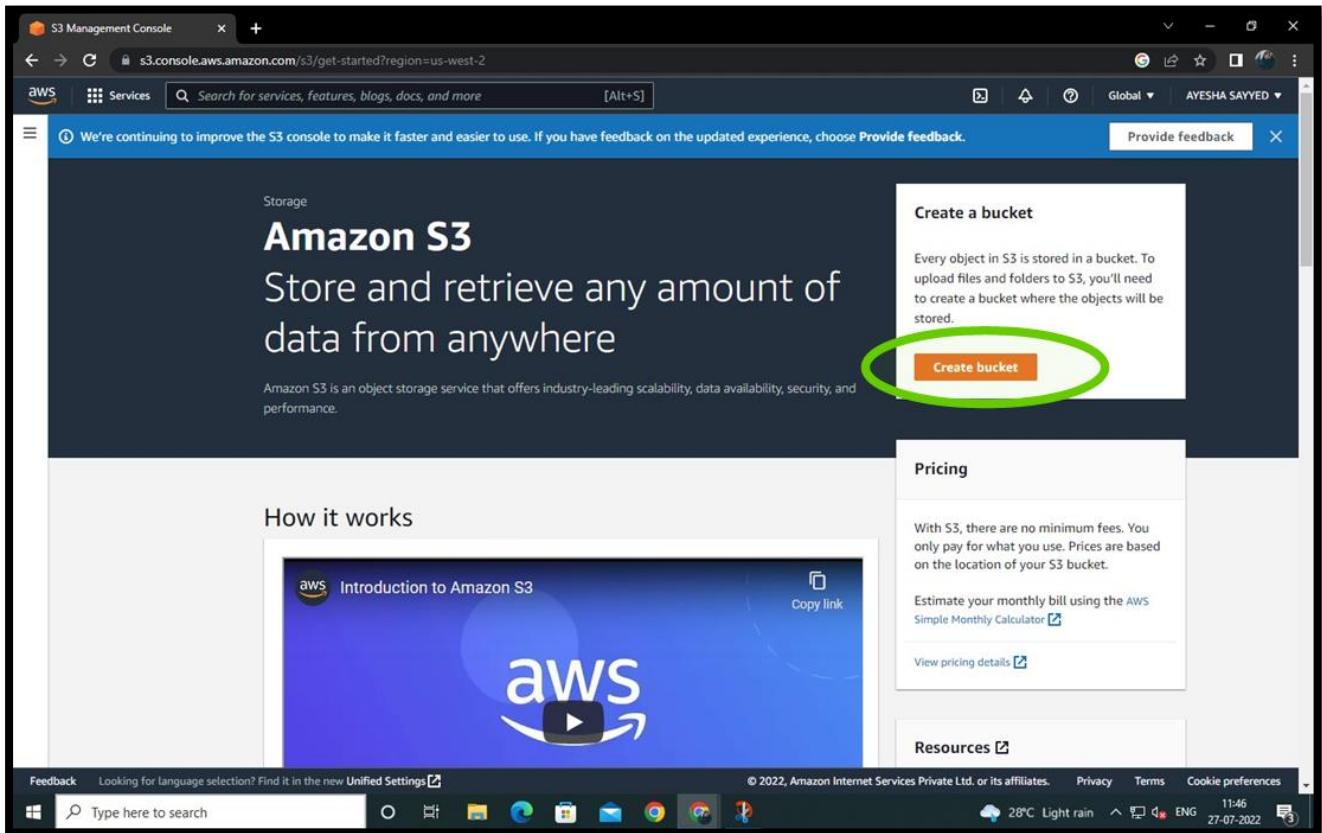
STEP 1: AWS Management Console Dashboard.



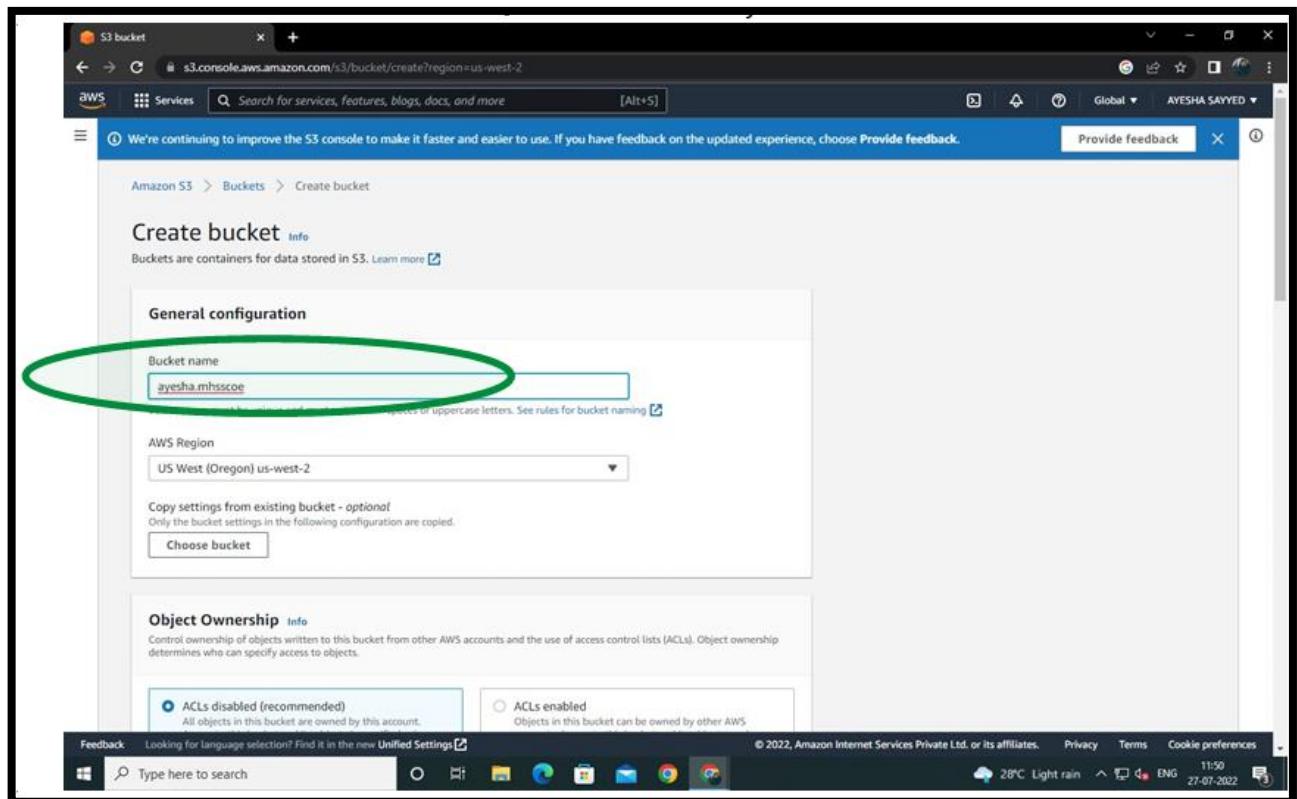
STEP 2: Click on Service -> Storage -> S3



STEP 3: Click on “Create a Bucket”.



STEP 4: Give name to your Bucket.



STEP 5: Uncheck the “Block all Public Access” checkbox.

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Warning: Turning off block all public access might result in this bucket and the objects within becoming public.
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

STEP 6: Select the Bucket

Buckets

Amazon S3 > Buckets

Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
ayesha.mhsscoe	US West (Oregon) us-west	Public	July 27, 2022, 11:50:55 (UTC+05:30)

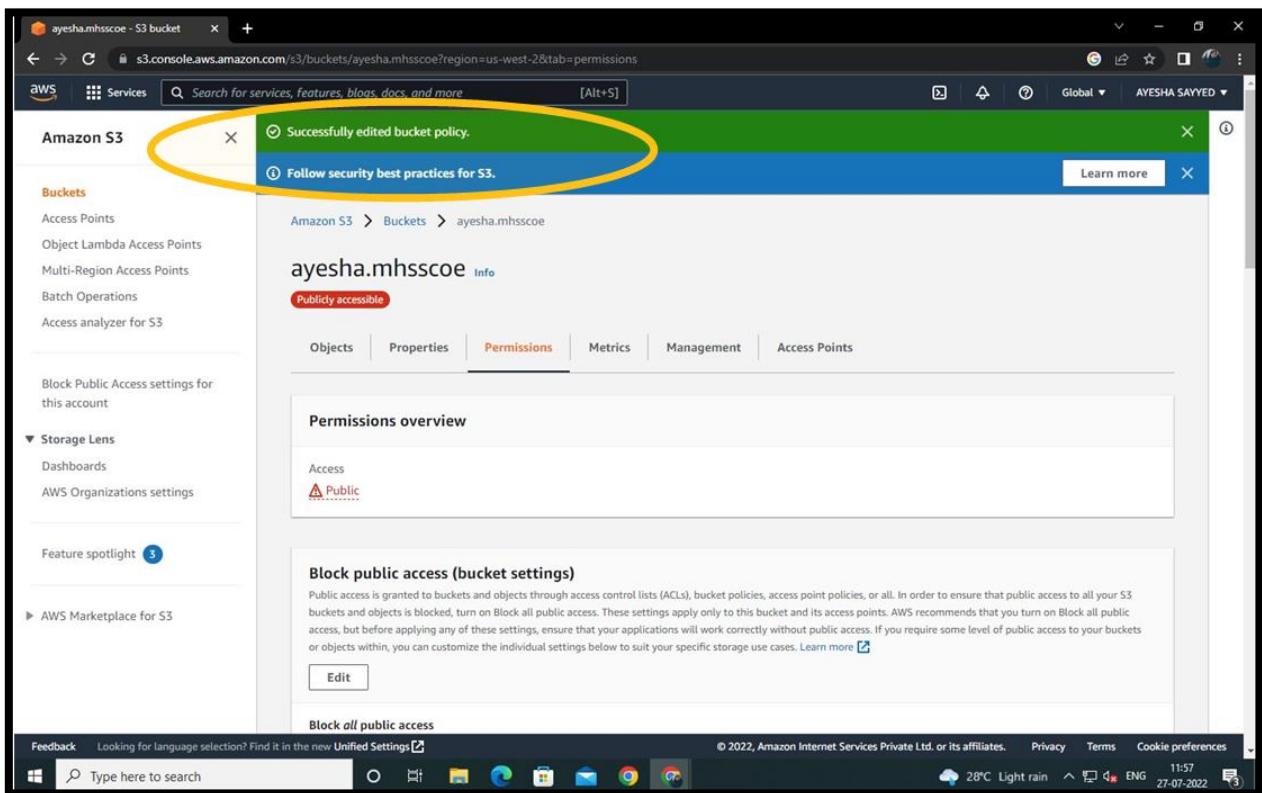
STEP 7: Permission -> Edit the Bucket policy section

The screenshot shows the AWS S3 console for the bucket 'ayesha.mhsscoe'. The left sidebar is collapsed. The main area is titled 'Bucket policy' and contains a message: 'The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts.' Below this is a text input field with the placeholder 'No policy to display.' and a 'Copy' button. At the top of the policy editor are 'Edit' and 'Delete' buttons.

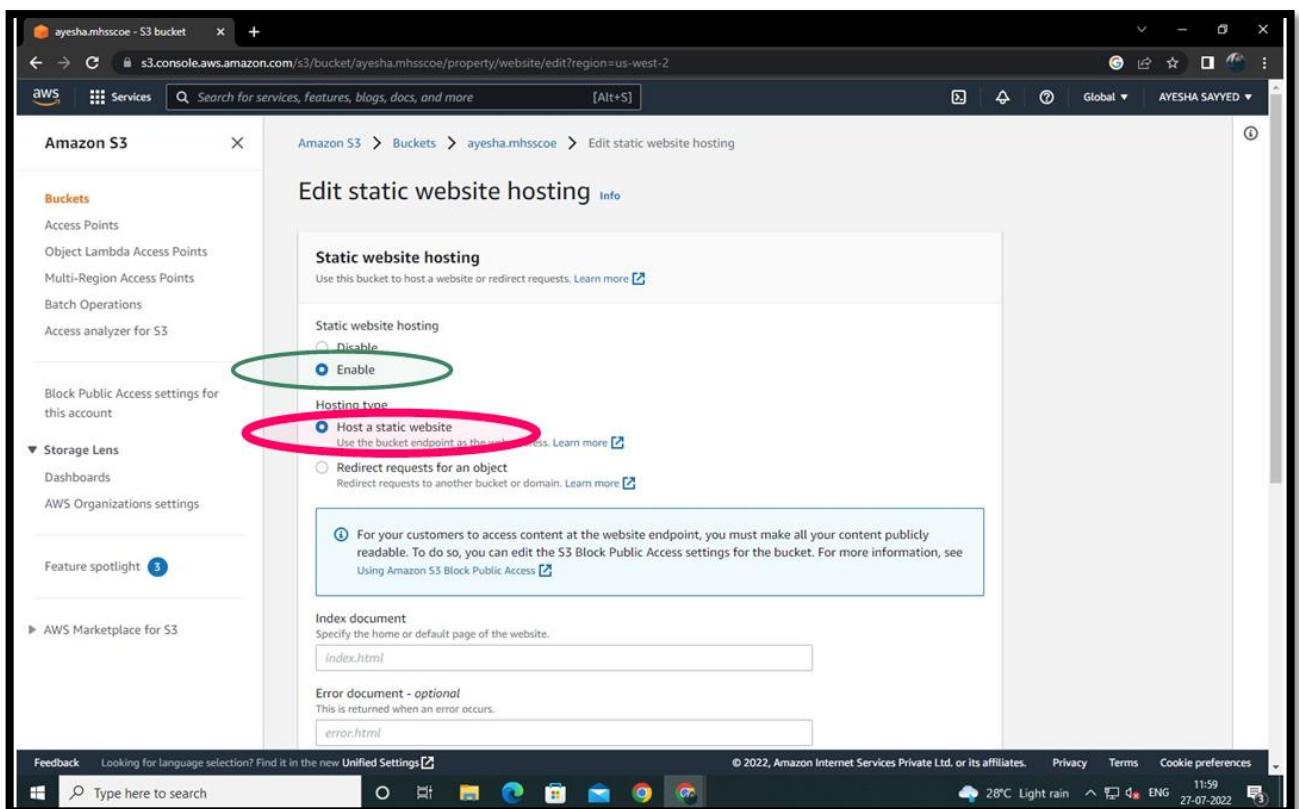
The screenshot shows the 'Edit bucket policy' page for the same bucket. The left sidebar is expanded, showing 'Buckets' and 'Storage Lens' sections. The main area has a title 'Edit bucket policy' with a 'Info' link. It includes a 'Bucket policy' section with 'Policy examples' and 'Policy generator' buttons, and a 'Bucket ARN' field containing 'arn:aws:s3:::ayesha.mhsscoe'. The 'Policy' section displays a JSON policy document:

```
1  {
2    "Version": "2008-10-17",
3    "Id": "PolicyForPublicWebsiteContent",
4    "Statement": [
5      {
6        "Sid": "PublicReadGetObject",
7        "Effect": "Allow",
8        "Principal": "*",
9        "Action": "s3:GetObject",
10       "Resource": "arn:aws:s3:::ayesha.mhsscoe/*"
11     }
12   ]
13 }
```

To the right of the policy editor is a sidebar with sections for 'Edit statement', 'Add actions', 'Included', and 'Available' services like S3, AMP, API Gateway, and API Gateway V2. The status bar at the bottom indicates it's 28°C, Light rain, 11:55, and the date is 27-07-2022.



STEP 8: Properties -> Edit “Static Website Hosting”-> Select “Enable Statics Website Hosting”.



The screenshot shows the AWS S3 console for the bucket 'ayesha.mhsscoe'. A green success message at the top right says 'Successfully edited static website hosting.' A purple box highlights this message. Below it, a blue bar says 'Follow security best practices for S3.' On the left sidebar, there's a 'Buckets' section with various options like Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and Access analyzer for S3. The main content area shows the bucket 'ayesha.mhsscoe' with its properties tab selected. It displays basic information such as AWS Region (US West (Oregon) us-west-2), Amazon Resource Name (ARN) (arn:aws:s3:::ayesha.mhsscoe), and Creation date (July 27, 2022, 11:50:55 (UTC+05:30)). A 'Bucket Versioning' section indicates it is disabled. At the bottom, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookies.

STEP 9: In the objects section attach the html code files.

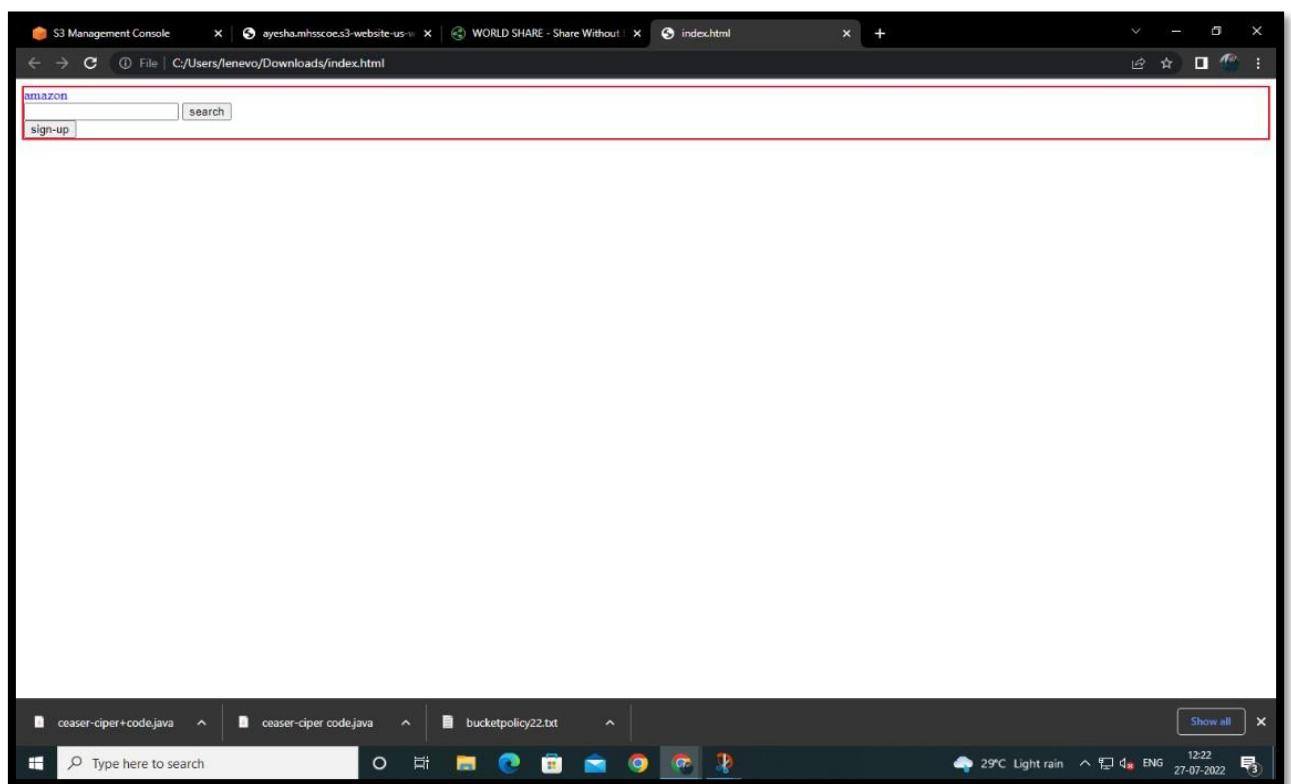
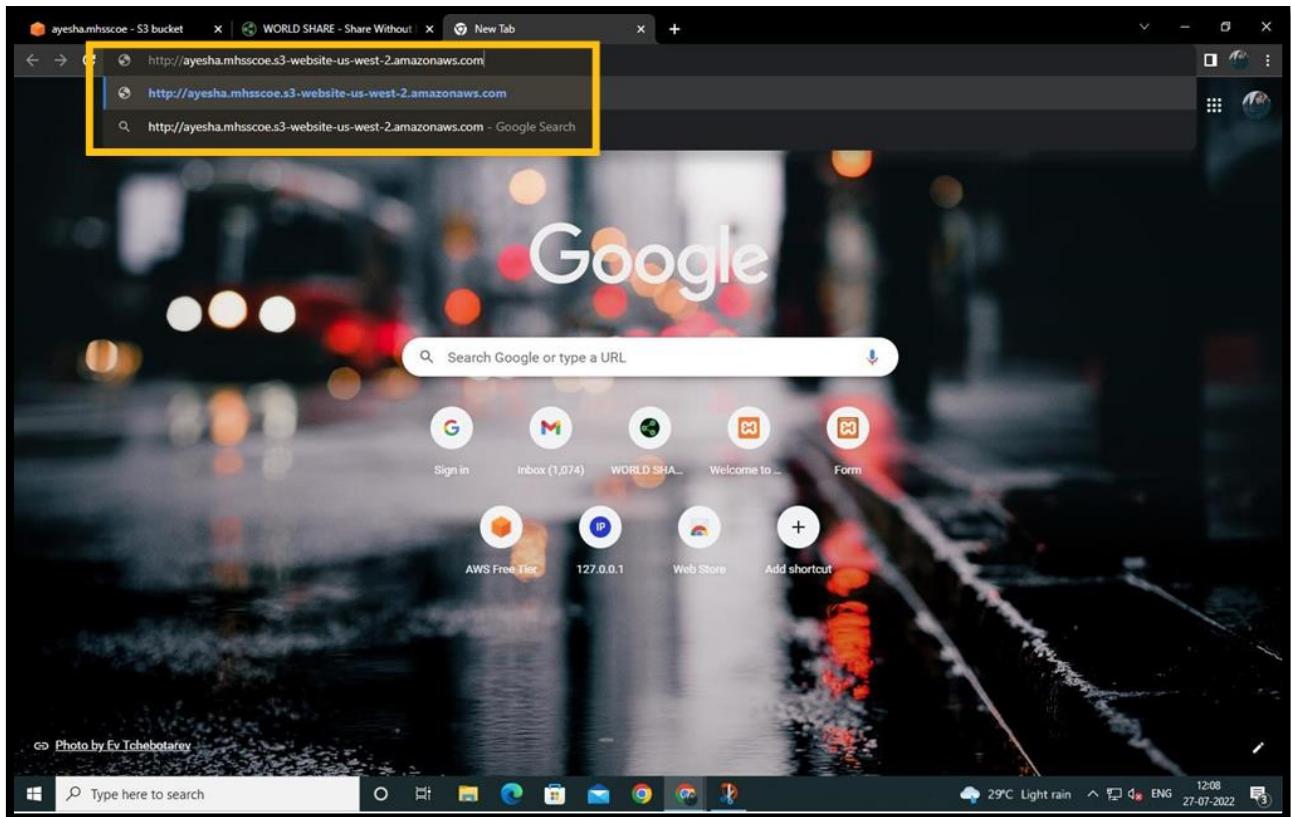
The screenshot shows the AWS S3 Management Console. A green oval highlights the 'index.html' file in the 'Files and folders' list. The 'Destination' section shows the target as 's3://ayesha.mhsscoe'. The 'Permissions' and 'Properties' sections are also visible. At the bottom right, there are 'Cancel' and 'Upload' buttons.

Now the html code file is been uploaded successfully.

The screenshot shows the AWS S3 Management Console. A green success message at the top left says "Upload succeeded". Below it, a summary table shows one file uploaded: "index.html" (text/html, 451.0 B) with a status of "Succeeded". Under "Files and folders", there is a table with one item: "index.html" (text/html, 451.0 B) with a status of "Succeeded". The status bar at the bottom right shows the date and time as 27-07-2022 12:06.

STEP 10: Now Copy the link from the Static website hosting then Paste the URL in a web browser and the web page will be displayed.

The screenshot shows the AWS S3 bucket properties page for "ayesha.mhsscoe". In the "Static website hosting" section, the "Hosting type" is set to "Bucket hosting". Below this, a note states: "When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket." To the right of this note is a purple oval highlighting the URL "http://ayesha.mhsscoe.s3-website-us-west-2.amazonaws.com".



STEP 11: After closing the web page, go to buckets and delete all object from the bucket.

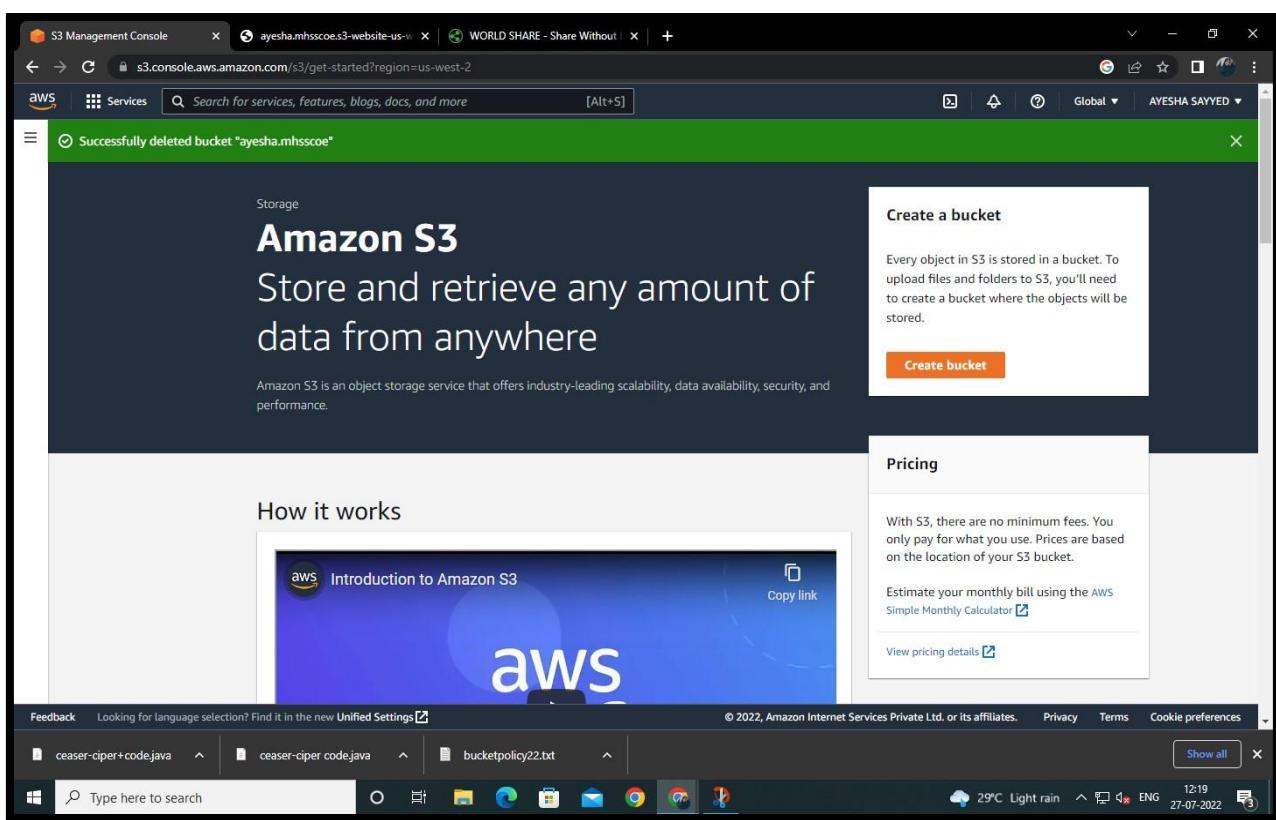
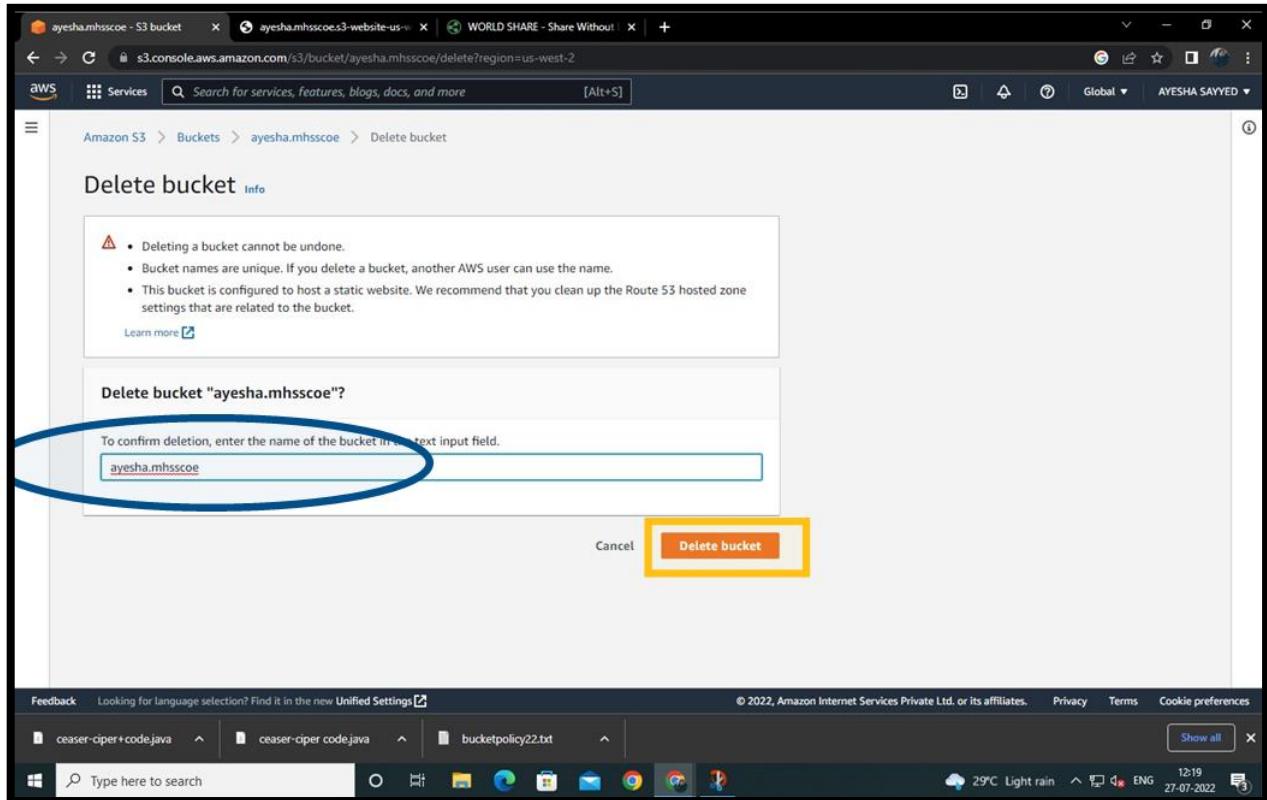
The screenshot shows the 'Delete objects' dialog box in the AWS S3 console. At the top, a warning message states: 'Deleting the specified objects can't be undone.' Below this is a table titled 'Specified objects' showing two files: 'ceaser-ciper code.java' (Type: java, Last modified: July 27, 2022, 12:10:02 (UTC+05:30), Size: 1.2 KB) and 'index.html' (Type: html, Last modified: July 27, 2022, 12:06:40 (UTC+05:30), Size: 451.0 B). A large text input field below the table contains the text 'permanently delete'. At the bottom right of the dialog are 'Cancel' and 'Delete objects' buttons. The status bar at the bottom of the browser window shows the URL 's3.console.aws.amazon.com/s3/buckets/ayesha.mhsscoe/object/delete?region=us-west-2&showversions=false'.

The screenshot shows the 'Delete objects: status' page in the AWS S3 console. A green banner at the top indicates 'Successfully deleted objects' with a link 'View details below.' Below this, a summary table shows the following data:

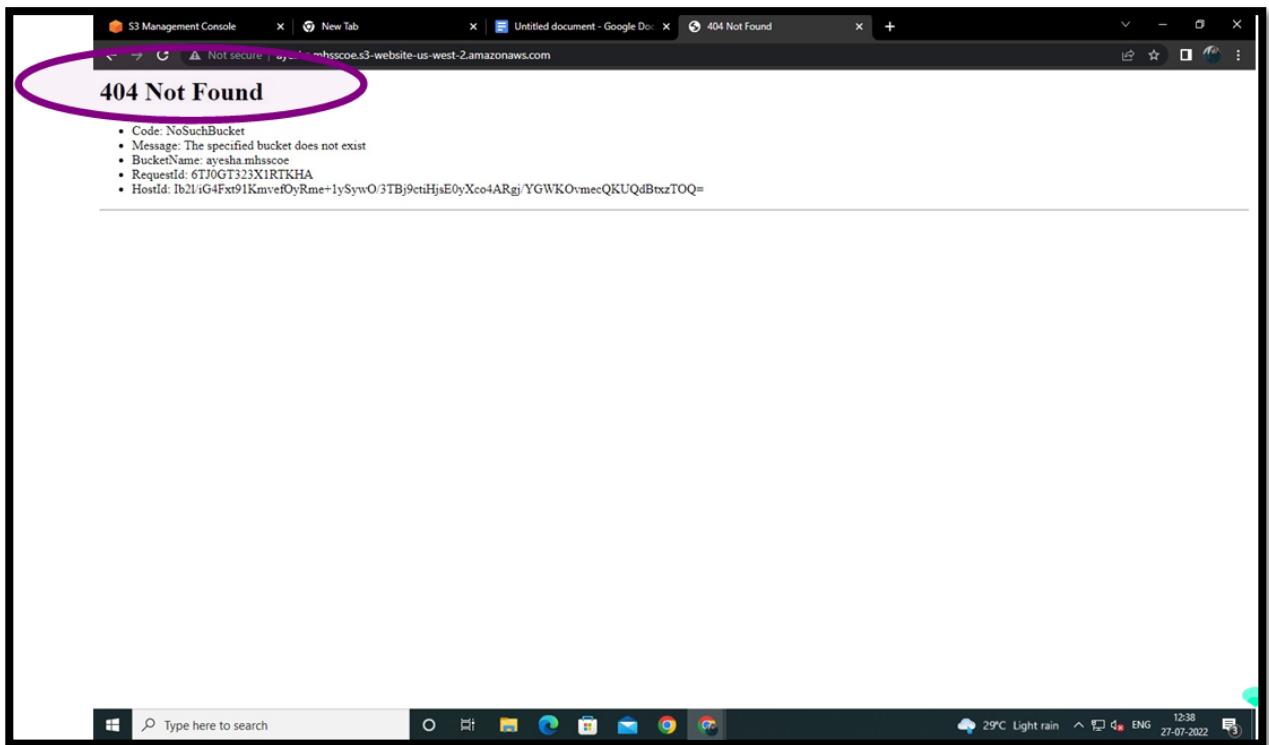
Source	Successfully deleted	Failed to delete
s3://ayesha.mhsscoe	2 objects, 1.6 KB	0 objects

A tab labeled 'Failed to delete (0)' is selected. The status bar at the bottom of the browser window shows the URL 's3.console.aws.amazon.com/s3/buckets/ayesha.mhsscoe/object/delete?region=us-west-2&showversions=false'.

STEP 12: Then select the empty BUCKET and delete it.



STEP 13: After deleting the Bucket if we try loading the web page, we will get the 404 ERROR.



EXPERIMENT – 03

Q. What is AWS Cloud9 ? Features of AWS Cloud9 ?

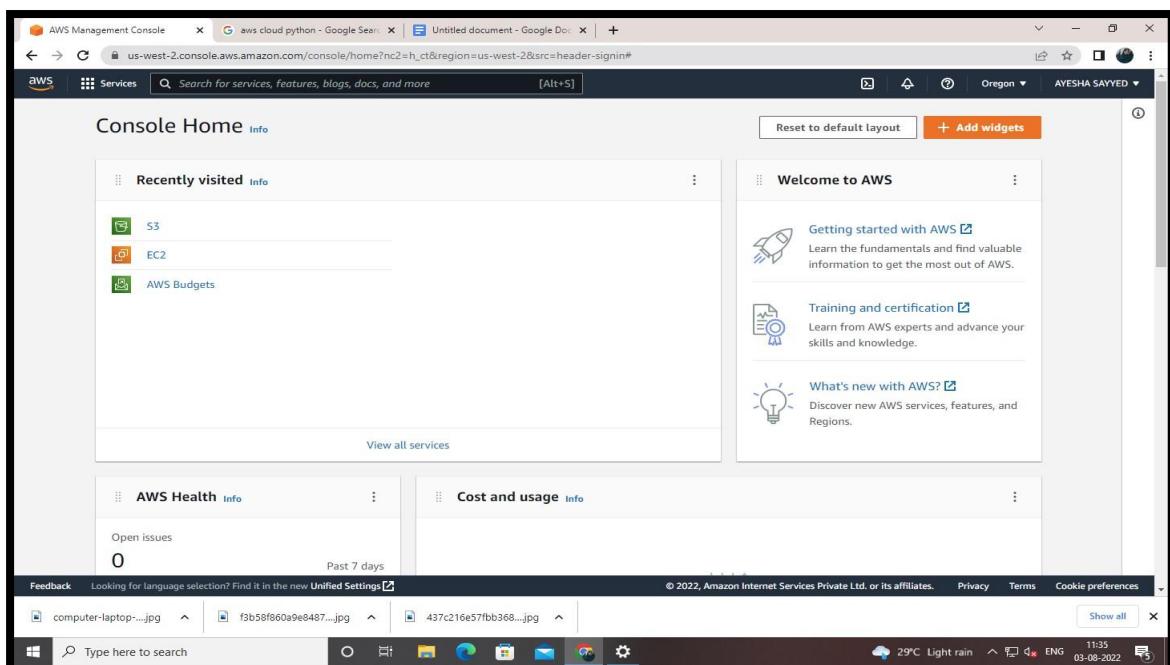
AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you to write, run, and debug code from any machine with just a browser.

The Cloud9 IDE provides the software and tooling needed for dynamic programming languages including JavaScript, Python, PHP, Ruby, Go, and C++.

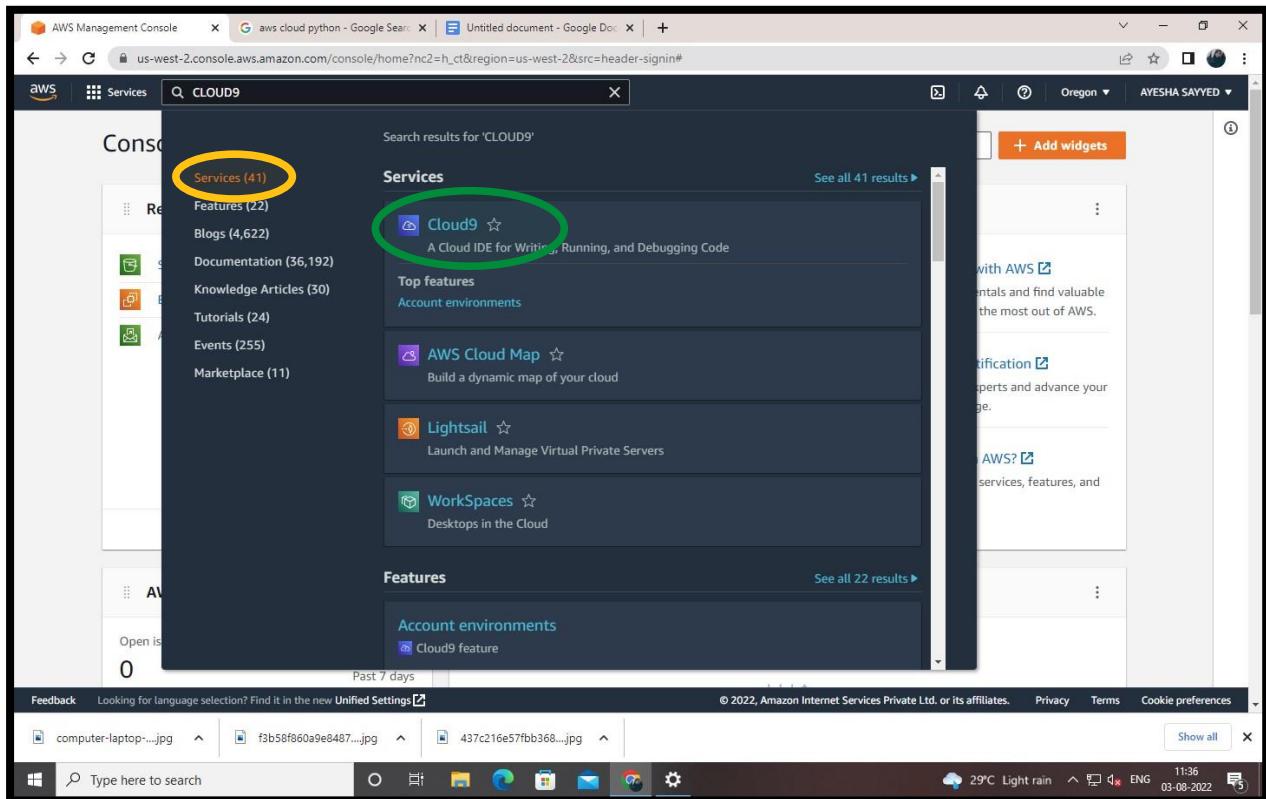
This means you no longer have to spend the time to install programs or configure your development machine. Features of AWS Cloud9 are :-

- ▲ Fully-featured Editor
- ▲ Broad Selection of Run Configurations
- ▲ Integrated Debugger
- ▲ Integrated Tools for Serverless Development
- ▲ Connectivity to Any Linux Server Platform
- ▲ Built-in Terminal
- ▲ Collaborative Editing and Chat
- ▲ Continuous Delivery Toolchain
- ▲ File Revision History
- ▲ Themes
- ▲ Keyboard Shortcuts
- ▲ Built-in Image Editor

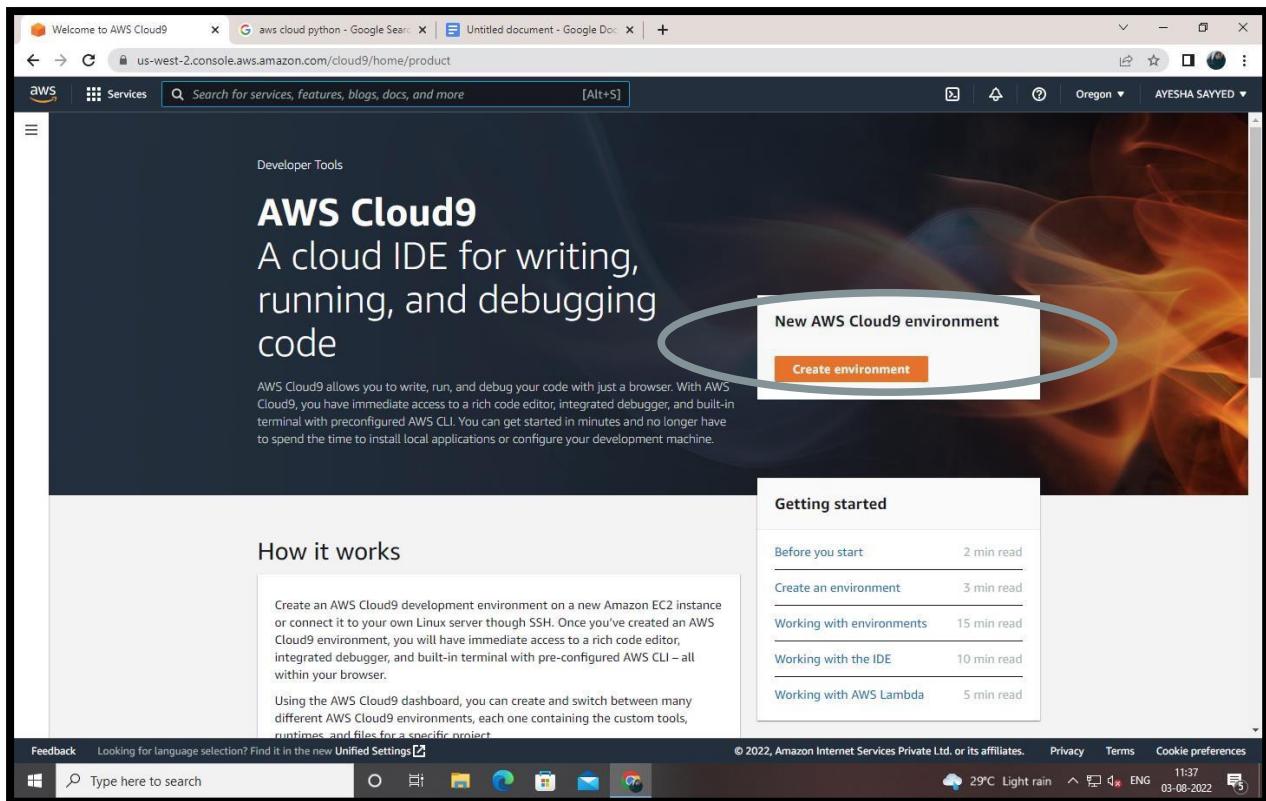
STEP 1: AWS Management Console Dashboard.



STEP 2: Search for Cloud9 and select it.



STEP 3: Click on “Create Environment”.



STEP 4: Name your environment and configure the settings.

AWS root account login detected
We do not recommend using your AWS root account to create or work with environments. Use an IAM user instead. This is an AWS security best practice. For more information, see [Setting Up to Use AWS Cloud9](#).

AWS Cloud9 > Environments > Create environment

Step 1 Name environment Step 2 Configure settings Step 3 Review

Name environment

Environment name and description

Name: AYESHA_SAYYED

Description – Optional: EXPERIMENT_03

Limit: 200 characters

Cancel **Next step**

Feedback Looking for language selection? Find it in the new Unified Settings. © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Step 1 Name environment Step 2 Configure settings Step 3 Review

Configure settings

Environment settings

Environment type Info
Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

Create a new EC2 instance for environment (direct access)
Launch a new instance in this region that your environment can access directly via SSH.

Create a new no-ingress EC2 instance for environment (access via Systems Manager)
Launch a new instance in this region that your environment can access through Systems Manager.

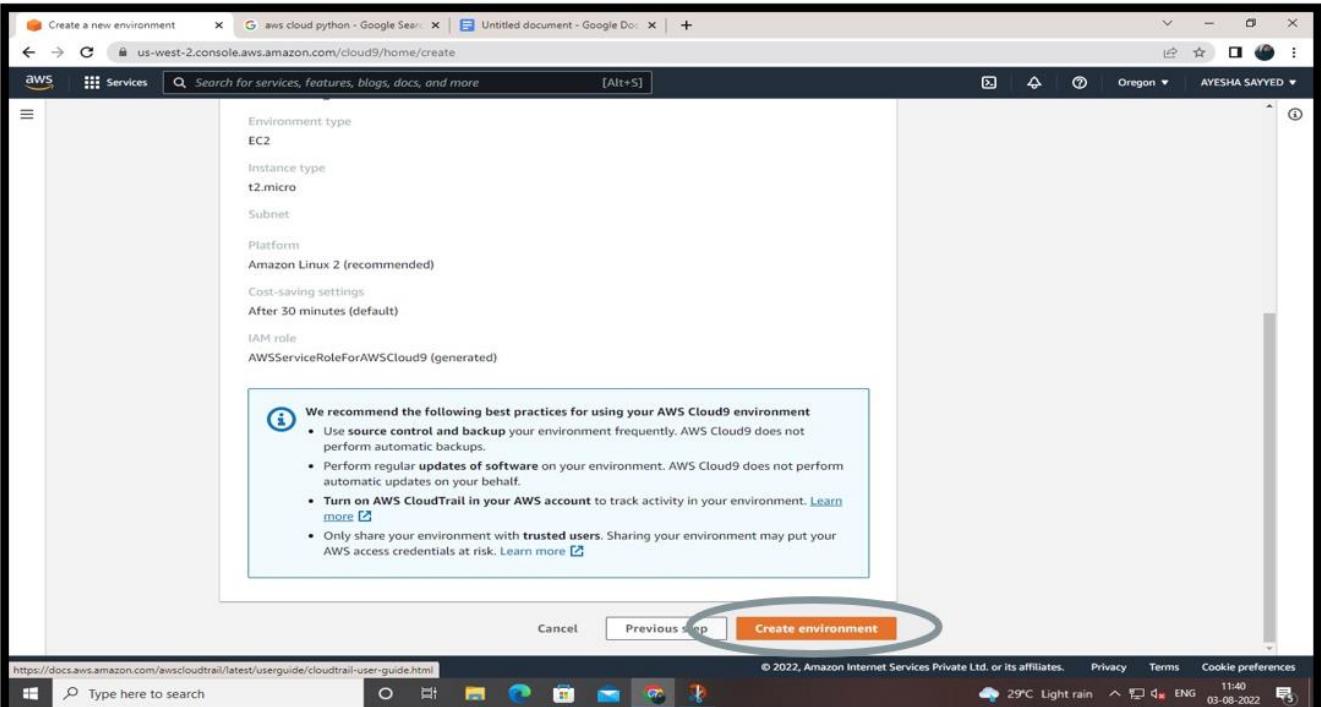
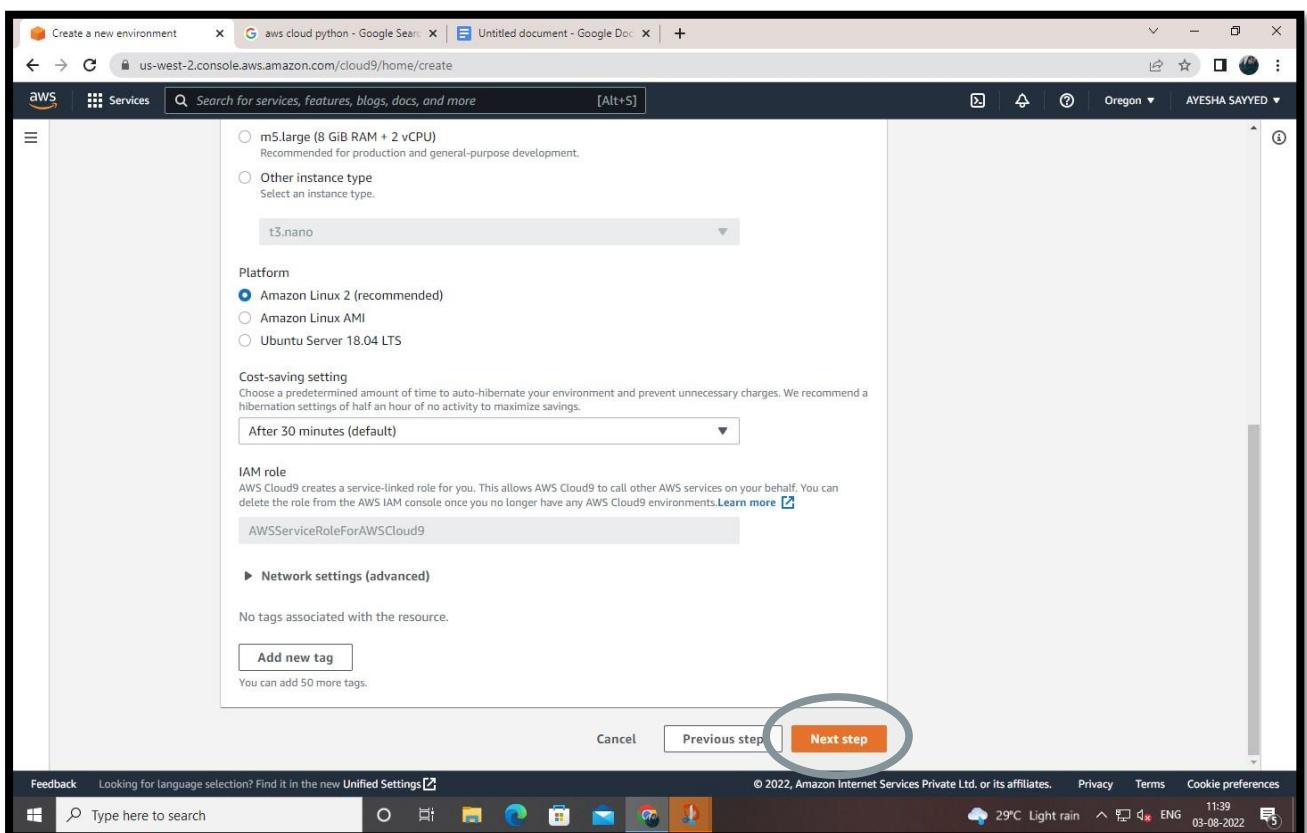
Create and run in remote server (SSH connection)
Configure the secure connection to the remote server for your environment.

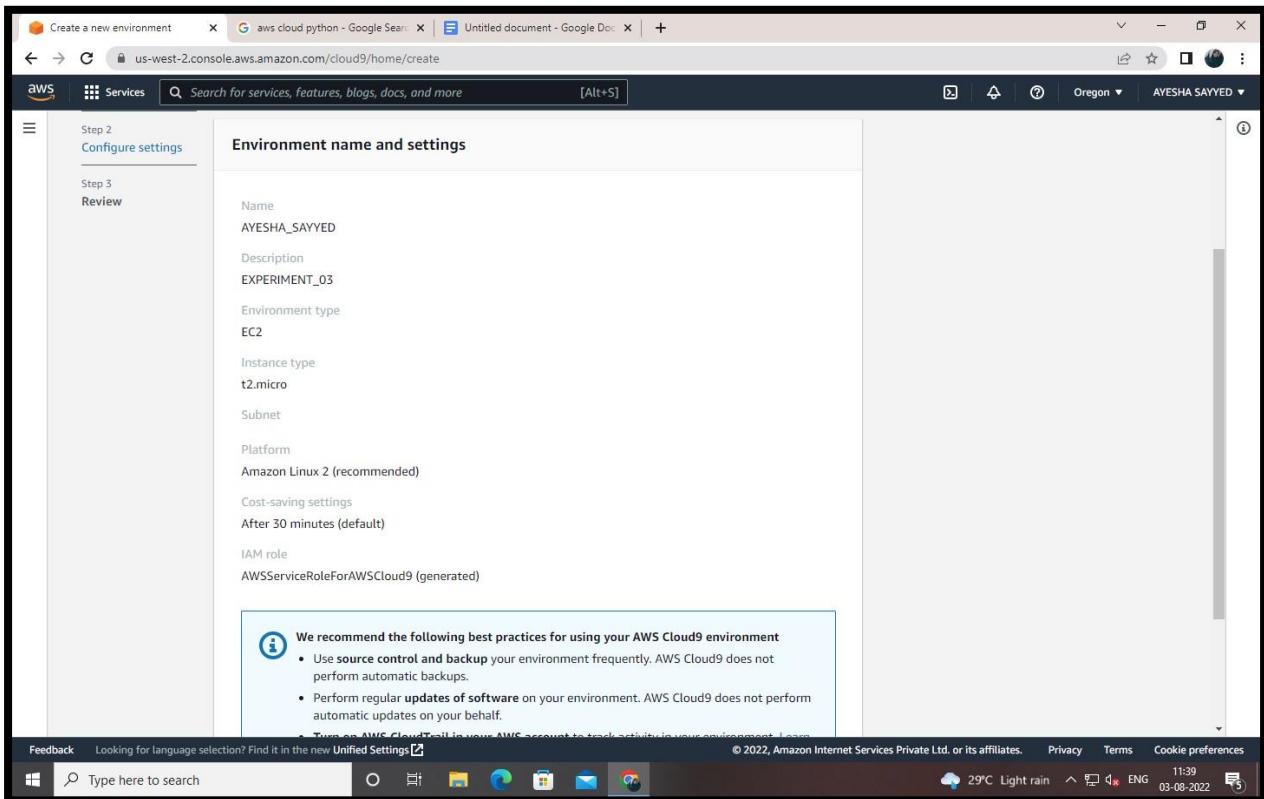
Instance type
 t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.
 t3.small (2 GiB RAM + 2 vCPU)
Recommended for small-sized web projects.
 m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and general-purpose development.
 Other instance type
Select an instance type.
t3.nano

Platform
 Amazon Linux 2 (recommended)
 Amazon Linux AMI
 Ubuntu Server 18.04 LTS

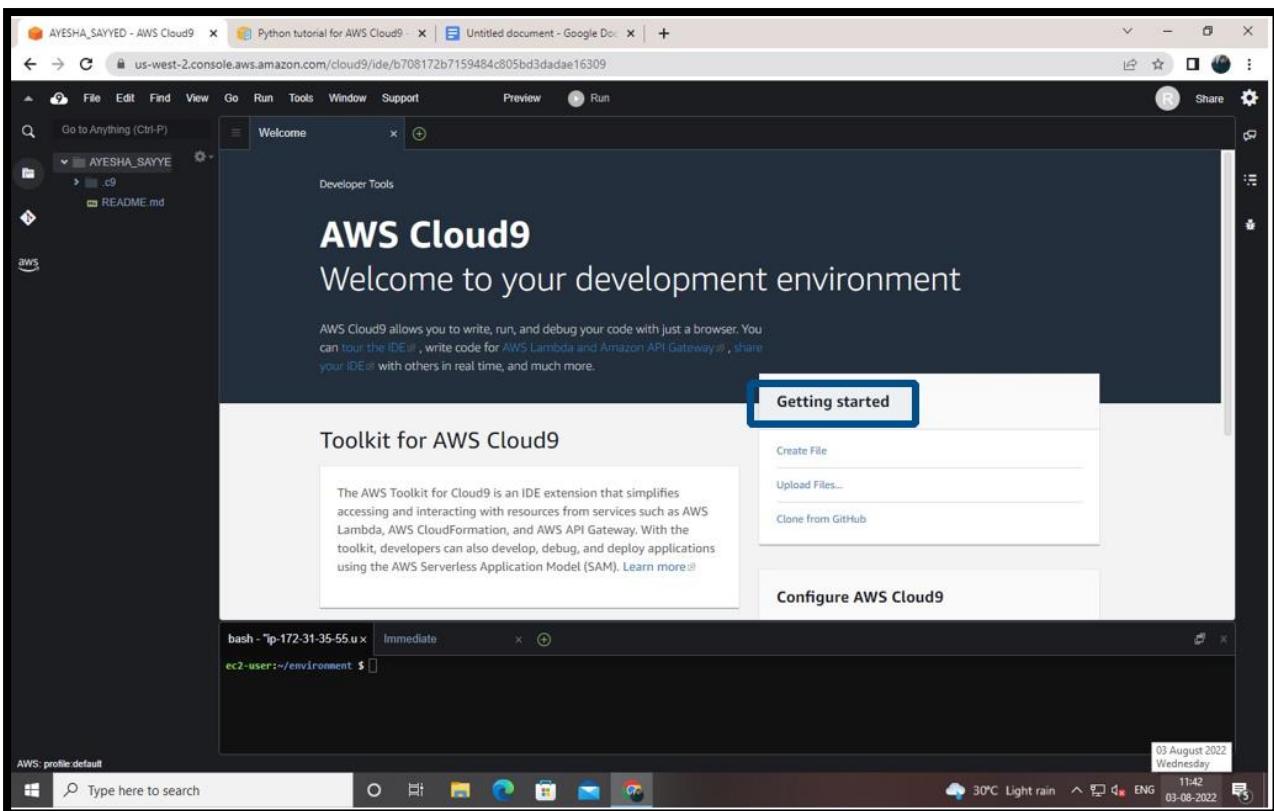
Cost-saving setting

Feedback Looking for language selection? Find it in the new Unified Settings. © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences





Now AWS Cloud9 Development Environment is being Created.



FOR PYTHON:

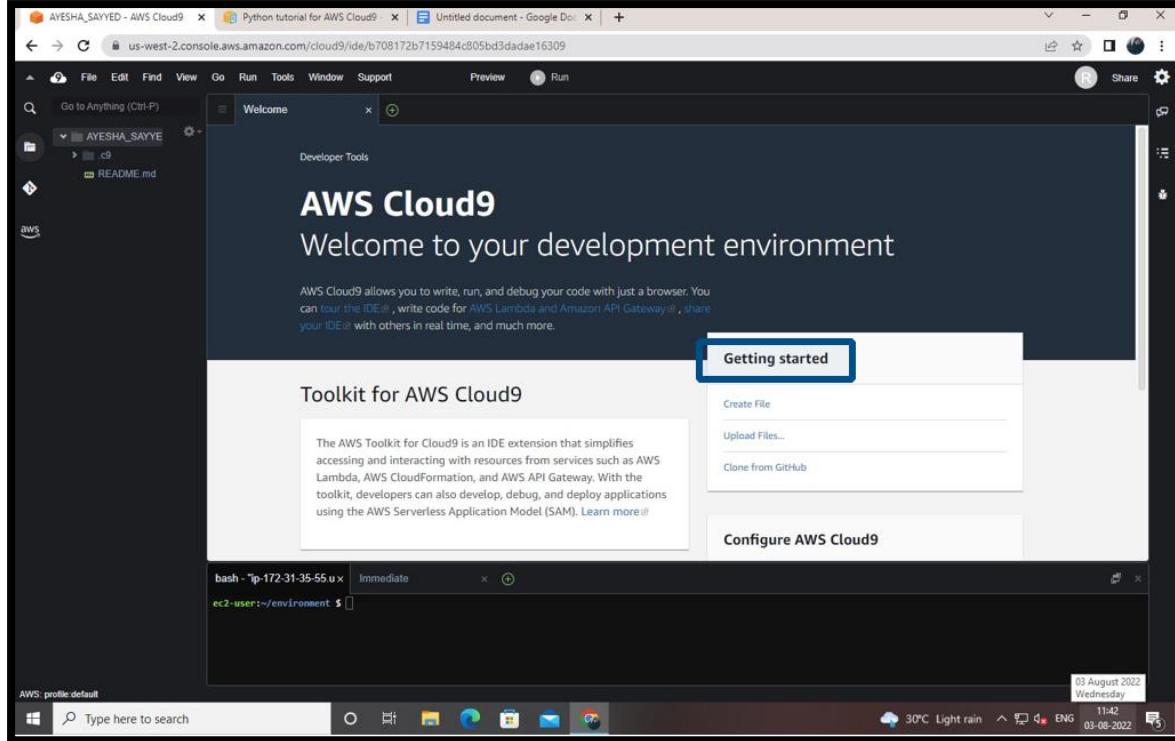
Step 1: Install Python. Run the yum update for Amazon Linux to help ensure the latest security updates and bug fixes are installed:

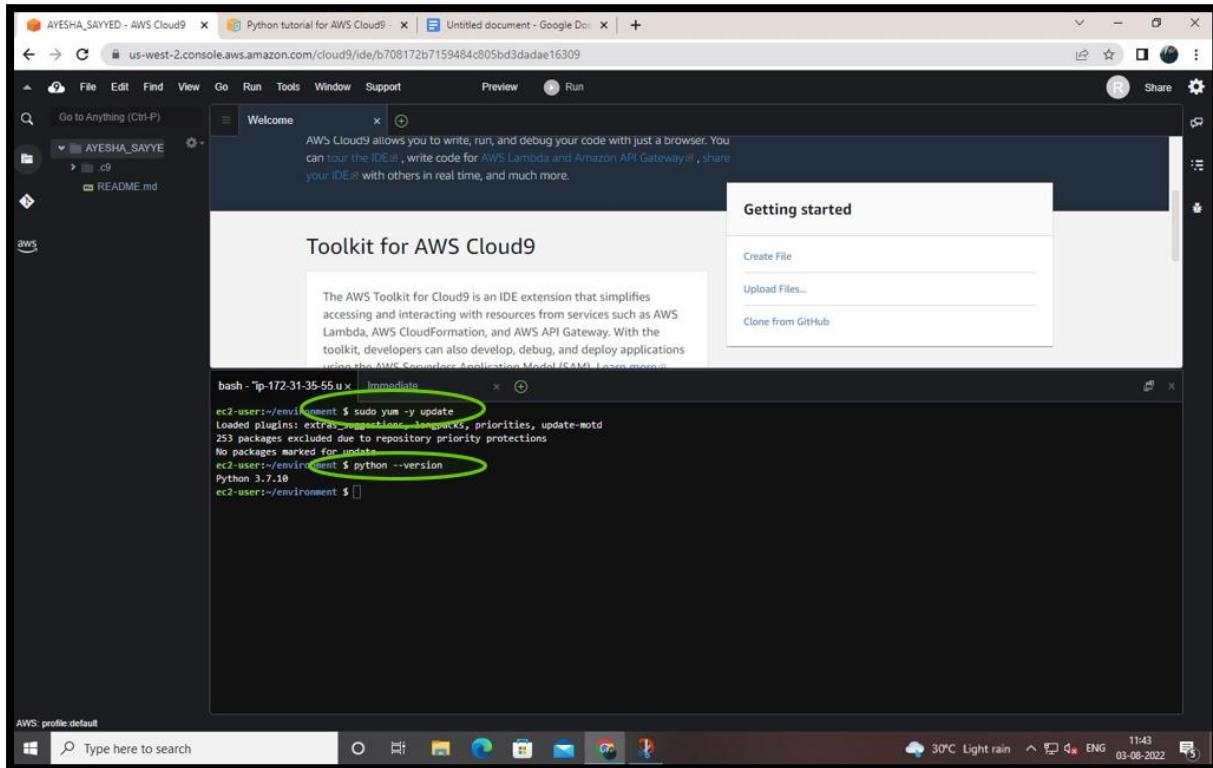
sudo yum -y update Install Python by running the install command. For Amazon Linux:

```
sudo yum -y install python3
```

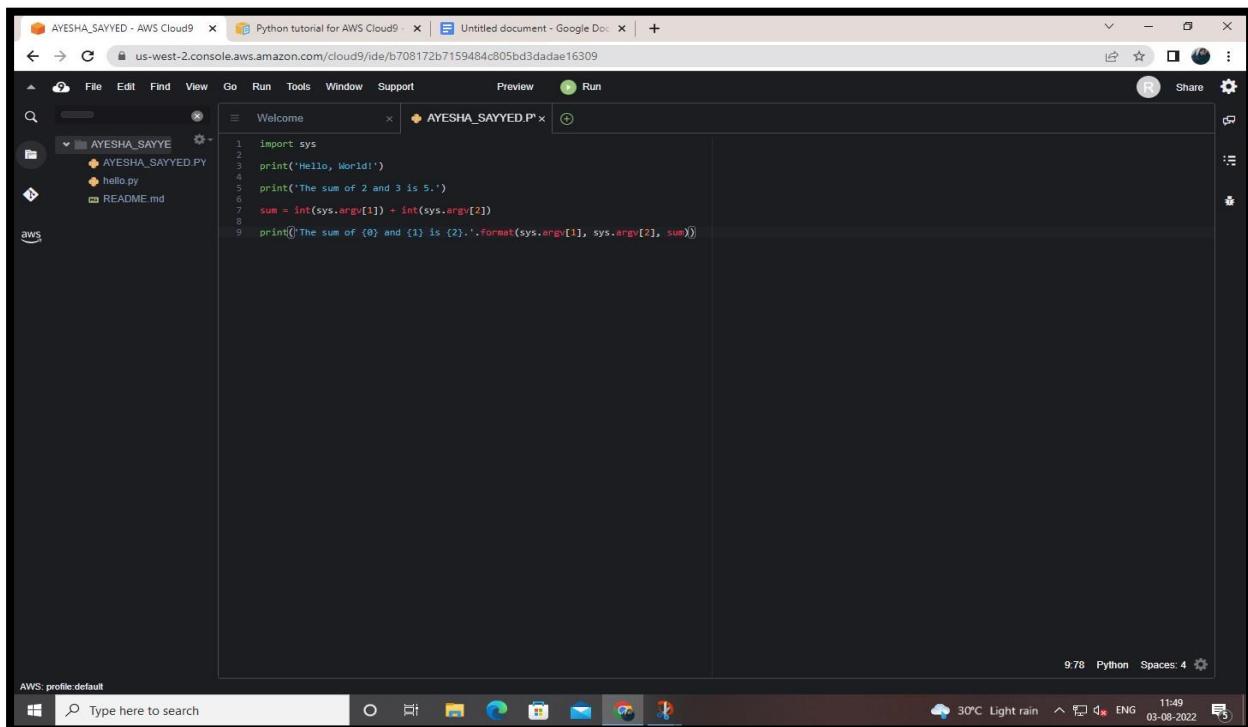
Here, python is already installed so we will check the python version by running the following command:

```
python –version
```



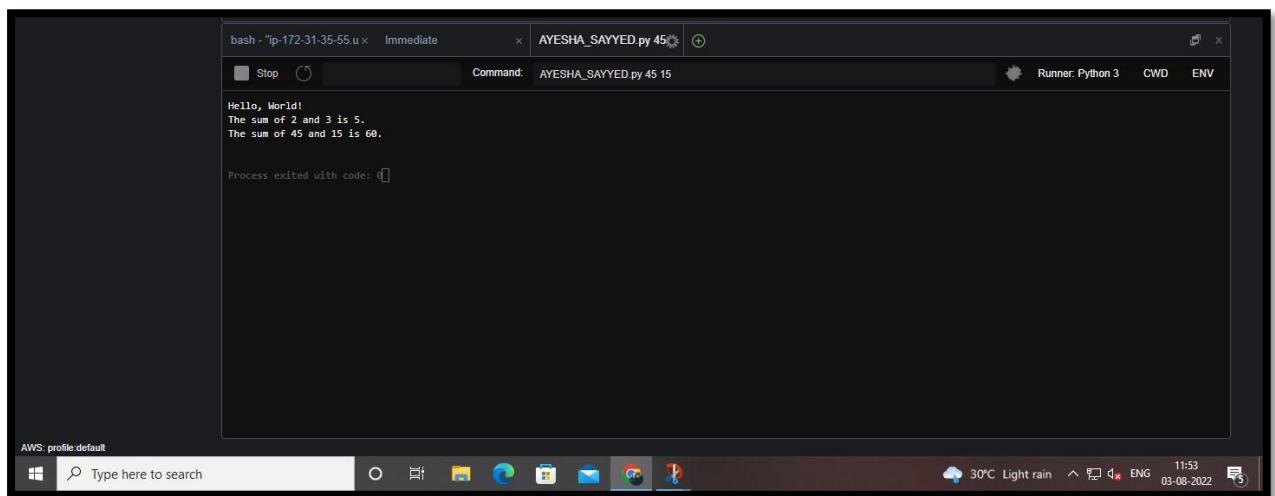
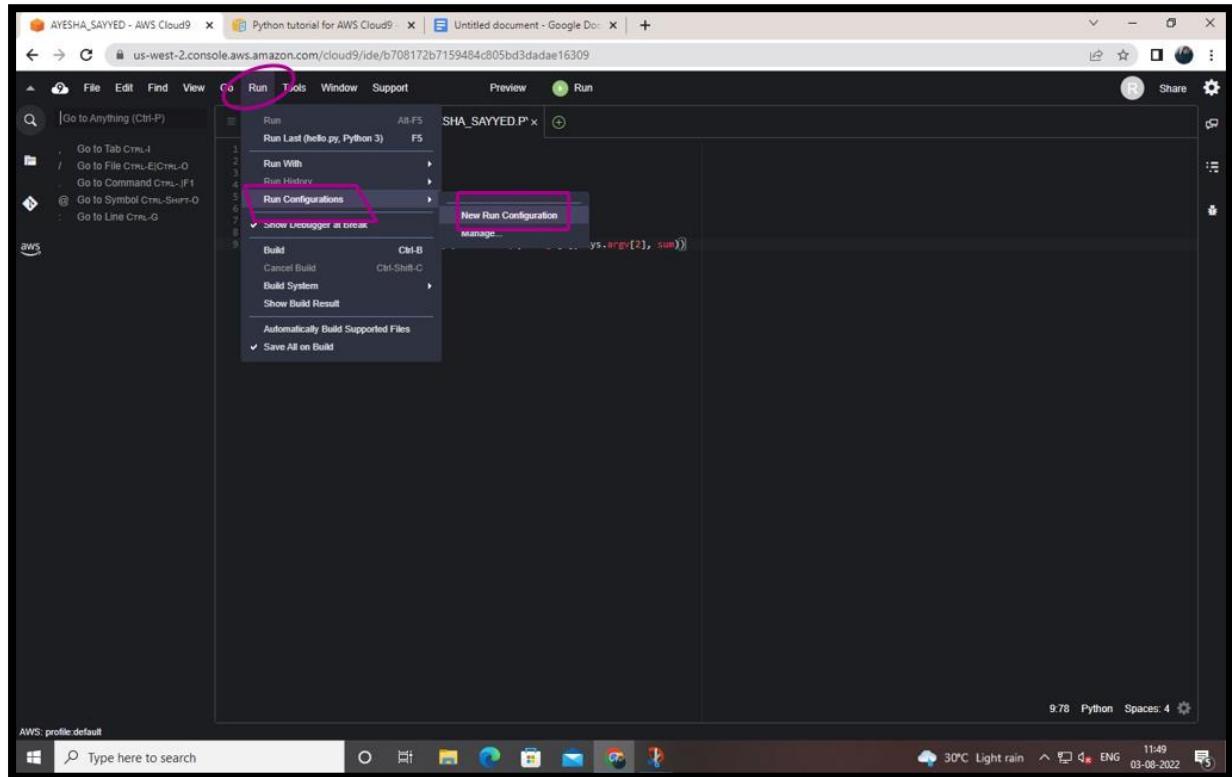


Step 2: Add code In the AWS Cloud9 IDE, create a file with the python code and save the file with some name.



Step 3 : Run the code

- In AWS Cloud9 IDE, on the menu bar choose Run -> Run Configurations -> New Run Configuration.
- On the [New] - Stopped tab, enter filename.py 15 10 for Command
- Choose Run



Step 4: Install and configure the AWS SDK for Python (Boto3)

Install pip: sudo python3.7 get-pip.py Install the AWS SDK for Python (Boto3) –

After you install pip, install the AWS SDK for Python (Boto3) by running the command:

sudo python3.6 -m pip install boto3 Check the Boto3 version by running the following command python -m pip show boto3

```

bash - ip-172-31-35-55 u x
ec2-user:~/environment $ sudo python3.7 -m pip install boto3
WARNING: Running pip install without --user is deprecated and will be a bad idea. Try `python3.7 -m pip install --user` instead.
Collecting boto3
  Downloading boto3-1.24.44-py3-none-any.whl (132 kB)
    ...
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/local/lib/python3.7/site-packages (from boto3) (1.0.1)
Collecting botocore<1.28.0,>=1.27.44
  Downloading botocore-1.27.44-py3-none-any.whl (9.0 kB)
    ...
Requirement already satisfied: s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
    ...
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/site-packages (from botocore<1.28.0,>=1.27.44->boto3) (2.8.2)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/local/lib/python3.7/site-packages (from botocore<1.28.0,>=1.27.44->boto3) (1.26.11)
Requirement already satisfied: six<1.5 in /usr/local/lib/python3.7/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.28.0,>=1.27.44->boto3) (1.16.0)
Installing collected packages: botocore, s3transfer, boto3
  Attempting uninstall: botocore
    Found existing installation: botocore 1.27.42
    Uninstalling botocore-1.27.42...
      Successfully uninstalled botocore-1.27.42
Successfully installed boto3-1.24.44 botocore-1.27.42 <s3transfer-0.6.0>
ec2-user:~/environment $ python -m pip show boto3
Name: boto3
Version: 1.24.44
Summary: The AWS SDK for Python
Home-page: https://github.com/boto/boto3
Author: Amazon Web Services
Author-email: None
License: Apache License 2.0
location: /usr/local/lib/python3.7/site-packages
Requires: s3transfer, jmespath, botocore
Required-by:
ec2-user:~/environment $ 

```

AWS: profile=default

Type here to search

30°C Light rain 12:01 03-08-2022

Step 5: Add AWS SDK code

Add code that uses Amazon S3 to create a bucket, list your available buckets, and optionally delete the bucket you just created In the AWS Cloud9 IDE, create a file with the code content and save the file with some name.

```

import sys
import boto3
from botocore.exceptions import ClientError

def list_my_buckets(s3_resource):
    print("Buckets:\n", [b.name for b in s3_resource.buckets.all()], sep="\n")

def create_and_delete_my_bucket(s3_resource, bucket_name, keep_bucket):
    list_my_buckets(s3_resource)

    try:
        print("\nCreating new bucket:", bucket_name)
        bucket = s3_resource.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                'LocationConstraint': s3_resource.meta.client.meta.region_name
            }
        )
    except ClientError as e:
        print(f"\nCouldn't create a bucket for the demo. Here's why: "
              f"{e.response['Error']['Message']}")
        raise

    bucket.wait_until_exists()
    list_my_buckets(s3_resource)

    if not keep_bucket:
        print("\nDeleting bucket:", bucket.name)
        bucket.delete()

    bucket.wait_until_not_exists()
    list_my_buckets(s3_resource)
    else:
        print("\nKeeping bucket:", bucket.name)

def main():
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('bucket_name', help='The name of the bucket to create.')
    args = parser.parse_args()

bash - ip-172-31-35-55 u x ayesha45.py mhsscoe45test-bucket us-west-2
Run Command: ayesha45.py mhsscoe45test-bucket us-west-2 Runner: Python 3 CWD ENV

```

AWS: profile=default

Type here to search

30°C Light rain 12:11 03-08-2022

The screenshot shows the AWS Cloud9 IDE interface. In the top navigation bar, the tabs are "AYESHA_SAYYED - AWS Cloud9", "Python tutorial for AWS Cloud9", and "Untitled document - Google Doc". The main editor window displays a Python script named "ayesha45.py". The script contains code for creating and deleting S3 buckets based on command-line arguments. Below the editor is a terminal window titled "bash - ip-172-31-35-55.us-west-2". The command entered is "ayesha45.py mhsscoe45test-bucket us-west-2". The terminal output shows "Process exited with code: 0". At the bottom, a Windows taskbar is visible with various icons and the system tray showing the date and time.

```

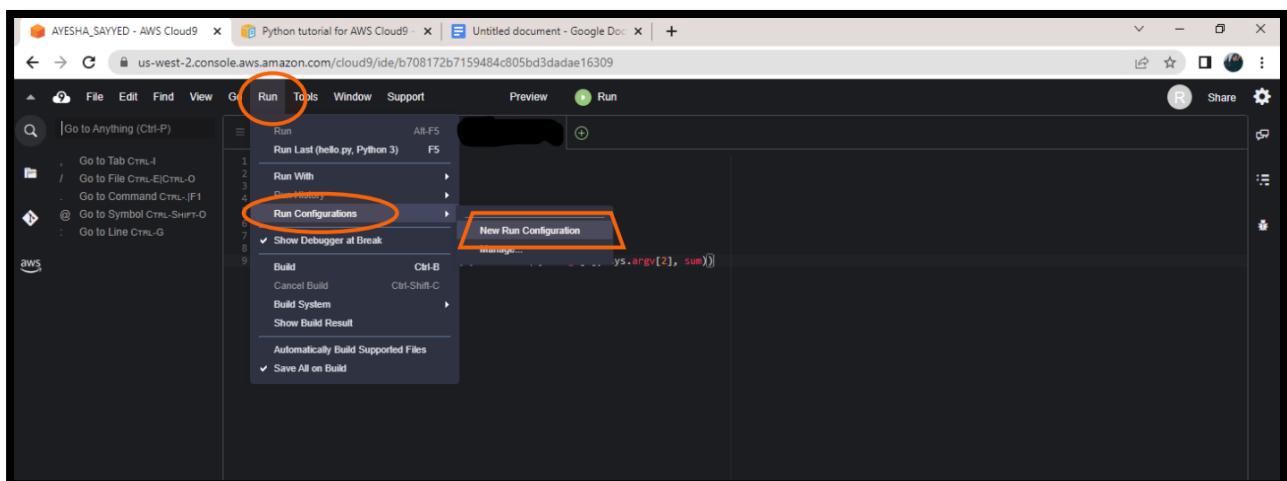
print('Deleting bucket: ', bucket.name)
bucket.delete()
bucket.wait_until_not_exists()
list_my_buckets(s3_resource)
else:
    print('\nKeeping bucket:', bucket.name)

def main():
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('bucket_name', help='The name of the bucket to create.')
    parser.add_argument('region', help='The region in which to create your bucket.')
    parser.add_argument('--keep_bucket', help='Keeps the created bucket. When not specified, the bucket is deleted at the end of the demo.', action='store_true')
    args = parser.parse_args()
    s3_resource = (
        boto3.resource('s3', region_name=args.region) if args.region
        else boto3.resource('s3'))
    try:
        create_and_delete_my_bucket(s3_resource, args.bucket_name, args.keep_bucket)
    except ClientError:
        print('Exiting the demo.')
if __name__ == '__main__':
    main()

```

Step 6: Run the AWS SDK code

- On the menu bar choose Run -> Run Configurations -> New Run Configuration
- For Command, enter filename.py 'name of bucket' us-west-2, where us-west-2 is the ID of the AWS Region where your bucket is created. By default, your bucket is deleted before the script exits
- Choose Run



A screenshot of a terminal window titled "AYESHA_SAYYED.PY". The terminal shows the command "ayesha45.py mhsscoe45t" being run. The output of the script is displayed, showing the creation and deletion of an AWS S3 bucket named "mhsscoe45test-bucket". The terminal interface includes tabs for "AYESHA_SAYYED.PY", "hello.py", and "README.md". A "Run" button is present, along with "Command:" and "Runner:" dropdowns. The status bar at the bottom indicates "AWS: profile.default".

```
Buckets:  
Creating new bucket: mhsscoe45test-bucket  
Buckets:  
mhsscoe45test-bucket  
Deleting bucket: mhsscoe45test-bucket  
Buckets:  
Process exited with code: 0
```

FOR Node.js:

Step 1: Install required tools

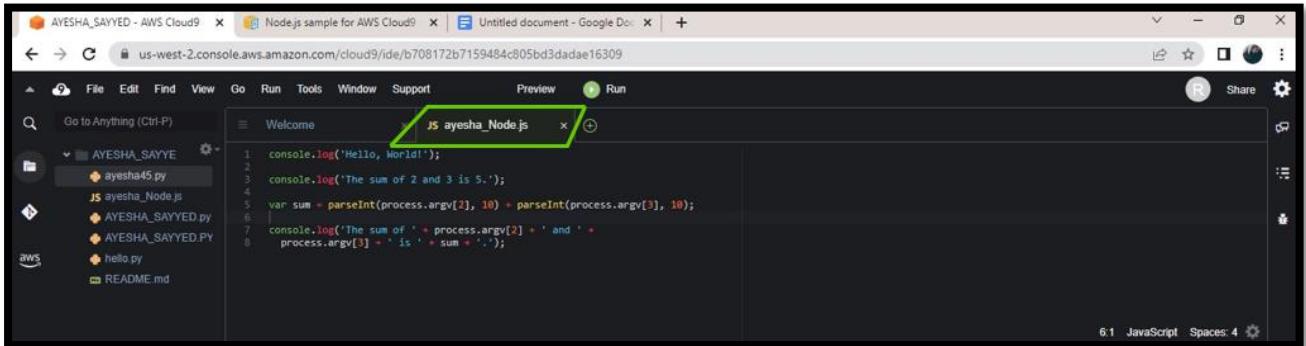
Run the yum update for Amazon Linux to help ensure the latest security updates and bug fixes are installed: sudo yum -y update

Run this command to install Node.js: nvm install v16.0.0 Here, node.js is already installed.

A screenshot of a terminal window titled "Getting started". It shows the command "sudo yum -y update" being run. The terminal output indicates that 253 packages were excluded due to repository priority protections and no packages were marked for update. The terminal interface includes tabs for "Getting started" and "AYESHA_SAYYED.PY". A "Type here to search" bar is at the bottom, along with a taskbar showing various application icons.

```
sudo -i ip-172-31-35-55 ux | +  
Installing collected packages: boto3, s3transfer, boto  
ec2-user:~/environment $ sudo yum -y update  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amazon2-core  
253 packages excluded due to repository priority protections  
No packages marked for update  
ec2-user:~/environment $ | 3.7 kB 00:00:00
```

Step 2: Add code In the AWS Cloud9 IDE, create a file with the node.js code and save the file with some name.



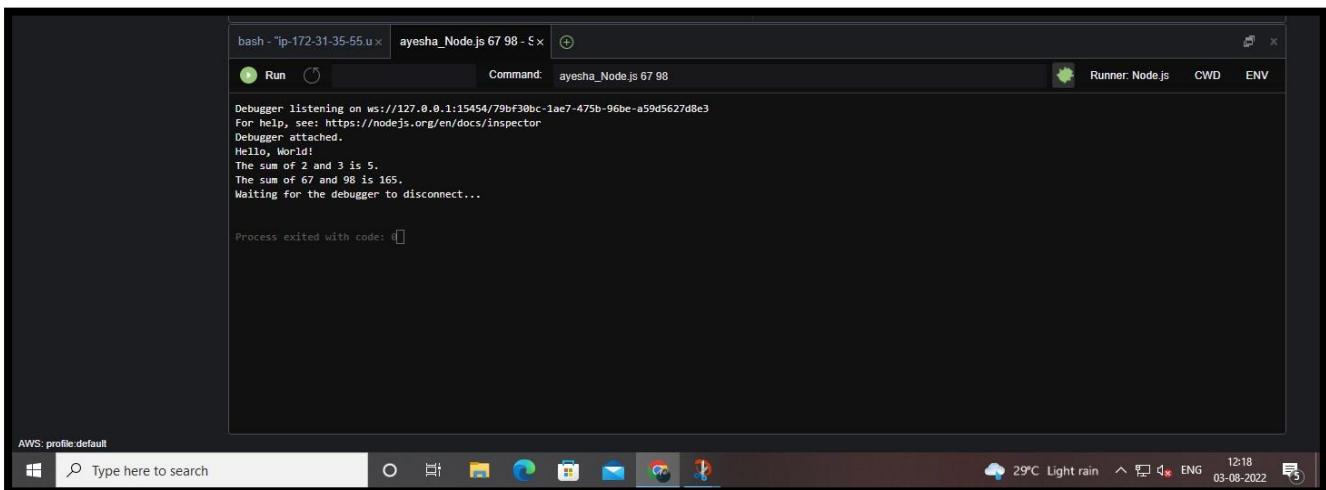
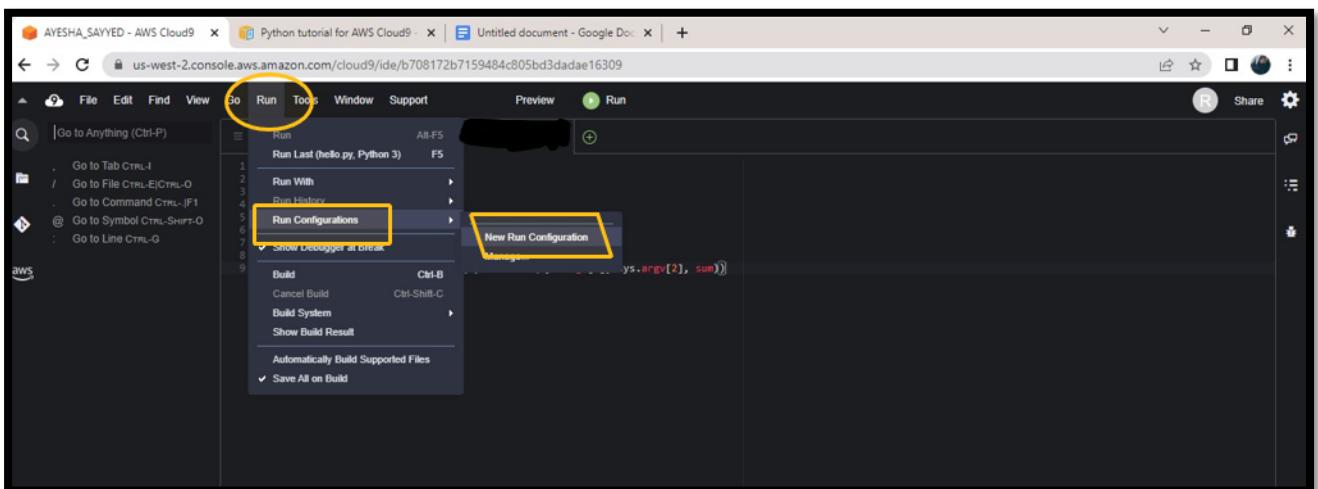
The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a sidebar with a file tree containing files like 'AYESHA_SAYYE', 'AYESHA45.py', 'JS ayesha_Node.js', 'AYESHA_SAYYE.py', 'AYESHA_SAYYE.PY', 'hello.py', and 'README.md'. The main workspace shows a terminal window with the title 'Welcome' and a code editor window titled 'JS ayesha_Node.js'. The code in the editor is:

```
1 console.log('Hello, World!');
2
3 console.log('The sum of 2 and 3 is 5.');
4
5 var sum = parseInt(process.argv[2], 10) + parseInt(process.argv[3], 10);
6
7 console.log('The sum of ' + process.argv[2] + ' and ' +
8   process.argv[3] + ' is ' + sum + '.');
```

The status bar at the bottom right indicates '6.1 JavaScript Spaces: 4'.

Step 3 : Run the code

- In AWS Cloud9 IDE, on the menu bar choose Run -> Run Configurations -> New Run Configuration.
- On the [New] – Idle tab, enter filename.js 15 10 for Command
- Choose Run



The screenshot shows the AWS Cloud9 terminal window. The command 'ayesha_Node.js 67 98' was run. The output shows the Node.js application running and printing 'Hello, World!', the sum of 2 and 3, the sum of 67 and 98, and finally waiting for the debugger to disconnect. The terminal also shows the process exiting with code 0.

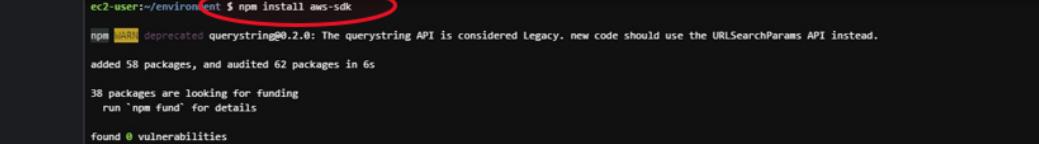
```
bash - "ip-172-31-35-55.us-east-1.compute.amazonaws.com" ayesha_Node.js 67 98 - S x
ayesha_Node.js 67 98
Run
Command: ayesha_Node.js 67 98
Runner: Node.js CWD ENV
Debugger listening on ws://127.0.0.1:15454/79bf30bc-1ae7-475b-96be-a59d5627d8e3
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Hello, World!
The sum of 2 and 3 is 5.
The sum of 67 and 98 is 165.
Waiting for the debugger to disconnect...
Process exited with code: 0
```

The status bar at the bottom shows 'AWS: profile default', a search bar, and system icons for weather, battery, and date/time.

Step 4: Install and configure the AWS SDK for JavaScript in Node.js

To install the AWS SDK for JavaScript(V2) in Node.js

Use npm to run the install command: `npm install aws-sdk`.



```
npm - "ip-172-31-35-55.us-west-2.compute.internal" ayesha Node.js 6.7.0 - S x +  
ec2-user:~/environment $ npm install aws-sdk  
npm WARN deprecated querystring@0.2.0: The querystring API is considered legacy. New code should use the URLSearchParams API instead.  
added 58 packages, and audited 62 packages in 6s  
38 packages are looking for funding  
  run 'npm fund' for details  
found 0 vulnerabilities  
ec2-user:~/environment $  
ec2-user:~/environment $
```

Step 5: Add AWS SDK code

In this step, you add some more code, this time to interact with Amazon S3 to create a bucket, list your available buckets, and then delete the bucket you just created.

In the AWS Cloud9 IDE, create a file with the code content, and save the file with some name.

The screenshot shows a browser window with the following tabs:

- AYESHA_SAYYED - AWS Cloud9
- Node.js sample for AWS Cloud9
- Untitled document - Google Docs

The main content area displays a Node.js script named `ayesha_bucket.js` in a code editor. The script uses the AWS SDK to interact with S3 buckets. It includes functions for creating a bucket and listing existing buckets. The code is highlighted in green, and the file path is shown in the address bar as `bash - "ip-172-31-35-56.us-west-2.compute.internal:8080/awscmds/nodejs/ayesha_bucket.js"`.

```
1 if (process.argv.length < 4) {
2     console.log('Usage: node s3.js <the bucket name> <the AWS Region to use>\n' +
3         'Example: node s3.js my-test-bucket us-east-2');
4     process.exit();
5 }
6
7 var AWS = require('aws-sdk'); // To set the AWS credentials and region.
8 var async = require('async'); // To call AWS operations asynchronously.
9
10 AWS.config.update({
11     region: region
12 });
13
14 var s3 = new AWS.S3({apiVersion: '2006-03-01'});
15 var bucket_name = process.argv[2];
16 var region = process.argv[3];
17
18 var create_bucket_params = {
19     Bucket: bucket_name,
20     CreateBucketConfiguration: {
21         LocationConstraint: region
22     }
23 };
24
25
26 var delete_bucket_params = {Bucket: bucket_name};
27
28 // List all of your available buckets in this AWS Region.
29 function listMyBuckets(callback) {
30     s3.listBuckets(function(err, data) {
31         if (err) {
32             console.log("Error occurred while listing buckets: " + err);
33         } else {
34             console.log("My buckets now are:\n");
35
36             for (var i = 0; i < data.Buckets.length; i++) {
37                 console.log(data.Buckets[i].Name);
38             }
39         }
40     });
41     callback(err);
42 });
43
44
45 // Create a new bucket
46 // Delete a bucket
47
48 // List all of your available buckets in this AWS Region.
```

The bottom status bar shows the AWS profile as "profile default".

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run buttons. A "Share" and "Settings" icon is also present. The left sidebar displays a file tree for the project "AYESHA_SAYYE" containing files like "ayesha45.py", "JS ayesha_bucketjs", "JS ayesha_Node.js", "JS AYESHA_SAYYED.PY", "hello.py", and "README.md". The main editor window contains a JavaScript file named "JS ayesha_bucketjs" with the following code:

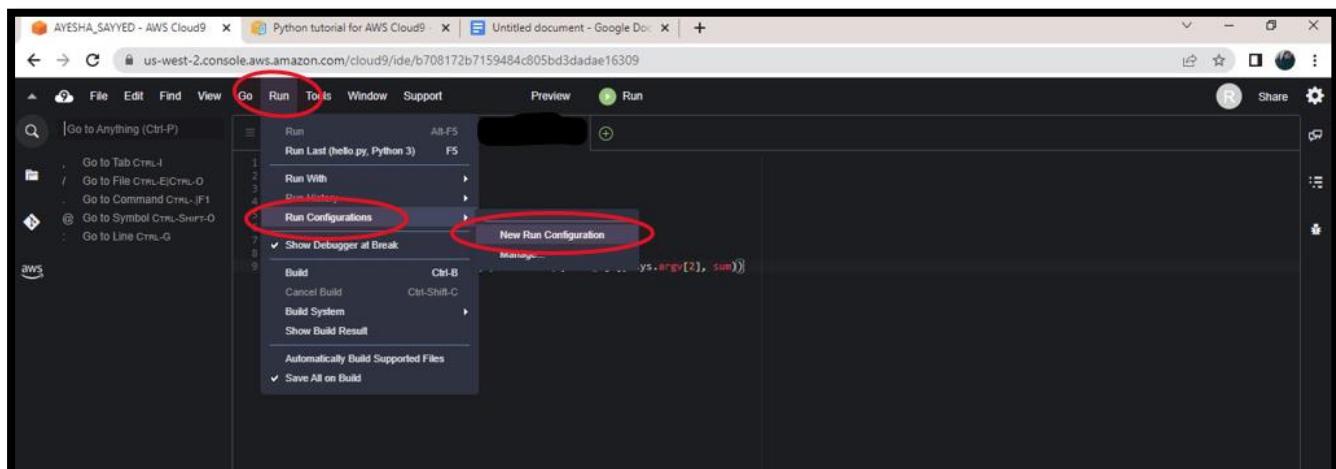
```
40      }
41      callback(err);
42    );
43  }
44
45  // Create a bucket in this AWS Region.
46  function createdMyBucket(callback) {
47    console.log(`\nCreating a bucket named ${bucket_name}\n`);
48
49    s3.createBucket(create_bucket_params, function(err, data) {
50      if (err) {
51        console.log(`Error creating bucket: ${err.code} - ${err.message}`);
52      }
53
54      callback(err);
55    );
56  }
57
58  // Delete the bucket you just created.
59  function deletedMyBucket(callback) {
60    console.log(`\nDeleting the bucket named ${bucket_name}\n`);
61
62    s3.deleteBucket(delete_bucket_params, function(err, data) {
63      if (err) {
64        console.log(`Error deleting bucket: ${err.code} - ${err.message}`);
65      }
66
67      callback(err);
68    );
69  }
70
71  // Call the AWS operations in the following order.
72  async.series([
73    listMyBuckets,
74    createdMyBucket,
75    listMyBuckets,
76    deletedMyBucket,
77    listMyBuckets
78  ]);

```

The status bar at the bottom shows "60:42 JavaScript Spaces: 2" and a terminal tab titled "bash - ip-172-31-35-55.us-east-1.compute.internal" with the command "ayesha_bucketjs mybucket" entered. The system tray indicates "29°C Light rain" and the date "03-08-2022".

Step 6: Run the AWS SDK code

- Enable the code to call Amazon S3 operations asynchronously by using npm to run the install command: npm install async
- On the menu bar choose Run -> Run Configurations -> New Run Configuration
- For Command, type filename.js 'name of bucket' us-east-2, where us-east-2 is the ID of the AWS Region you want to create the bucket in
- Choose run



```
npm - ip-172-31-35-55.us-west-2.compute.internal ayesha_bucket.js my_bucket
ec2-user:~/environment $ npm install async
added 1 package, and audited 63 packages in 694ms
38 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
ec2-user:~/environment $
```

A screenshot of a terminal window titled 'npm - ip-172-31-35-55.us-west-2.compute.internal'. It shows the command 'npm install async' being run. The output indicates that one package was added and 63 packages were audited. There were no vulnerabilities found. The user is currently in the directory 'environment'.

```
bash - ip-172-31-35-55.us-west-2.compute.internal ayesha_bucket.js my-test
Run Command: ayesha_bucket.js my-test-bucket-ayesha us-east-2
Runner: Node.js CWD ENV
Debugger listening on ws://127.0.0.1:15454/f9af3599-b940-451d-afab-b8be511cd2a0
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
My buckets now are:

Creating a bucket named my-test-bucket-ayesha...
My buckets now are:
my-test-bucket-ayesha

Deleting the bucket named my-test-bucket-ayesha...
My buckets now are:
Waiting for the debugger to disconnect...

Process exited with code: 0
```

A screenshot of a terminal window titled 'bash - ip-172-31-35-55.us-west-2.compute.internal'. It shows the command 'ayesha_bucket.js my-test' being run. The output shows the creation of a new bucket named 'my-test-bucket-ayesha'. After the bucket is created, it is immediately deleted. The process exits with a code of 0. The terminal also displays the Node.js debugger information and the AWS region setting 'us-east-2'.

Finally, close all terminals and delete the Cloud9 Environment.

AWS Cloud9

Your environments

Shared with you

Account environments

How-to guide

AWS root account login detected

We do not recommend using your AWS root account to create or work with environments. Use an IAM user instead. This is an AWS security best practice. For more information, see Setting Up to Use AWS Cloud9.

AWS Cloud9 > Your environments

Your environments (1)

AYESHA_SAYYED

Type: EC2 Permissions: Owner

Description: EXPERIMENT_03

Owner Arn: arn:aws:iam::518313021583:root

Open IDE

Delete Create environment

Feedback Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

29°C Light rain ENG 03-08-2022

Cloud9 Development Environment is now deleted.

AWS Cloud9

Your environments

Shared with you

Account environments

How-to guide

AWS root account login detected

We do not recommend using your AWS root account to create or work with environments. Use an IAM user instead. This is an AWS security best practice. For more information, see Setting Up to Use AWS Cloud9.

AWS Cloud9 > Your environments

Your environments (0)

Open IDE View details Delete Create environment

You currently do not own any AWS Cloud9 environments. Create an environment to have them show in this dashboard.

EXPERIMENT – 04

Q1. What is NO SQL, key value Databases?

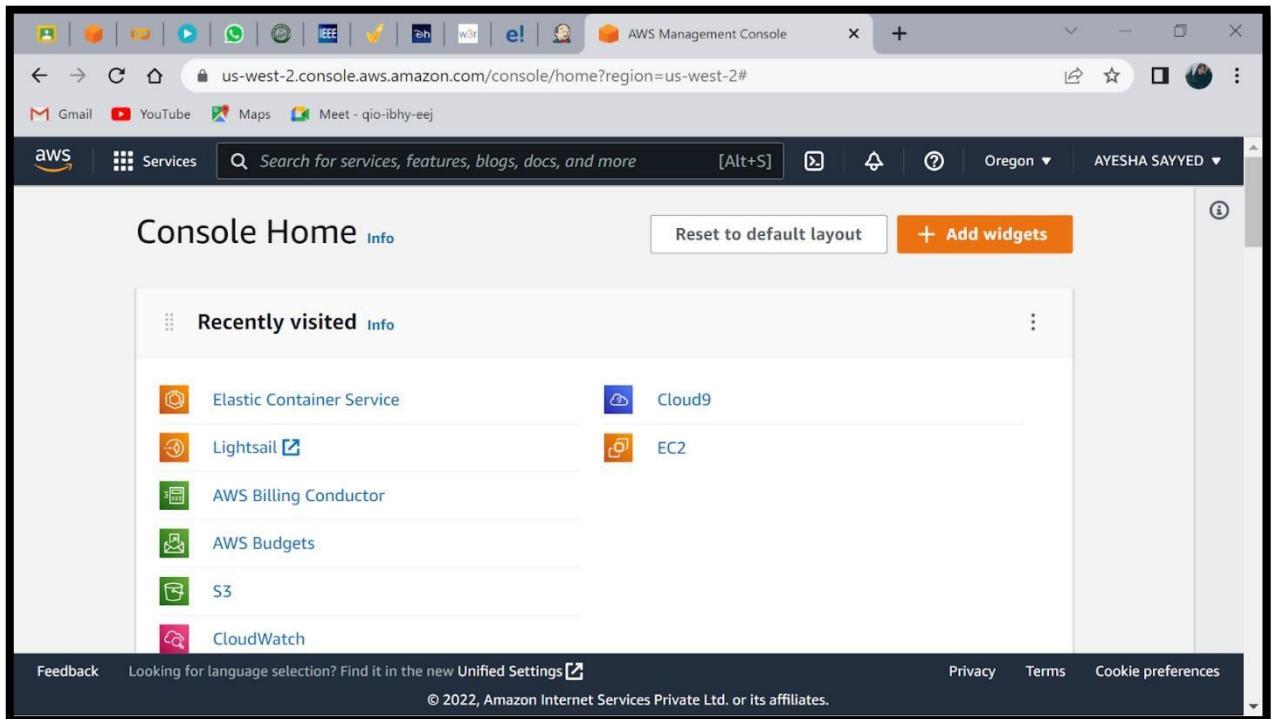
All those databases which are modeled in means other than the tabular relations used in relational databases are known as NO SQL databases. NoSQL key-value databases are the least complicated types of NoSQL databases.

They store data as a key or attribute name with its value. Each data item has a pointer and a unique key.

The key-value pairs are in the form of rows of associative arrays.

Q2. Create a Table in DynamoDB , Add items to the table (minimum 10 items) , Query the table .

Step 1: AWS Management Console Dashboard.



Step 2: Search for DynamoDB and select it.

The screenshot shows the AWS DynamoDB search results page. A search bar at the top contains the query 'dynamo'. Below the search bar, a sidebar on the left lists various AWS services and features. The main content area displays search results for 'dynamo' under the heading 'Services (3)'. The results include 'DynamoDB' (Managed NoSQL Database), 'CloudFront' (Global Content Delivery Network), and 'AWS Cloud Map' (Build a dynamic map of your cloud). An orange circle highlights the 'DynamoDB' entry. At the bottom of the page, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

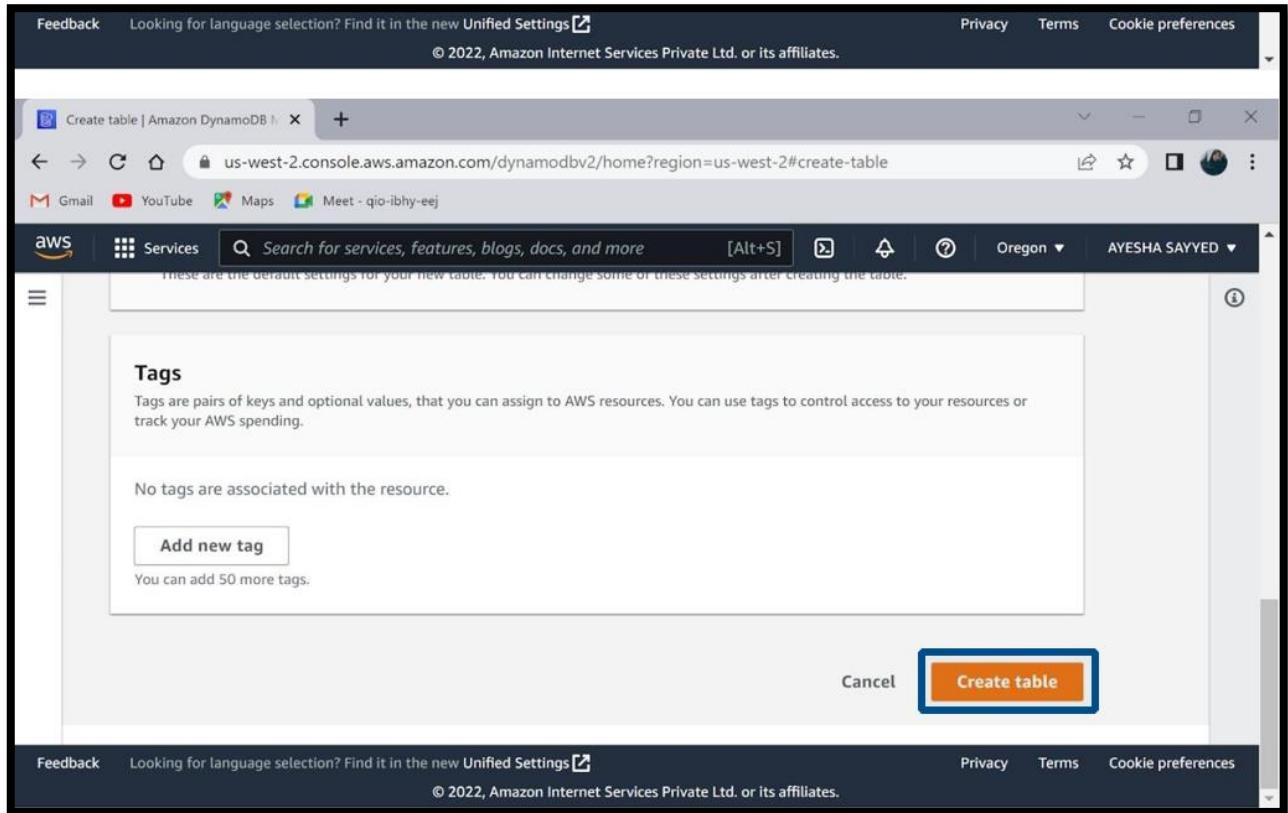
Step 3: Click on 'Create Table'.

The screenshot shows the 'Get started' page for DynamoDB. On the left, a sidebar lists service navigation options like Dashboard, Tables, and Create Table. The main content area features a 'Get started' section with a large orange 'Create table' button highlighted with a purple border. Below this, there's a 'Pricing' section with a note about charges for reading, writing, and storing data. The top navigation bar includes links for Gmail, YouTube, Maps, Meet, and a user profile for 'AYESHA SAYYED'. The bottom footer contains links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

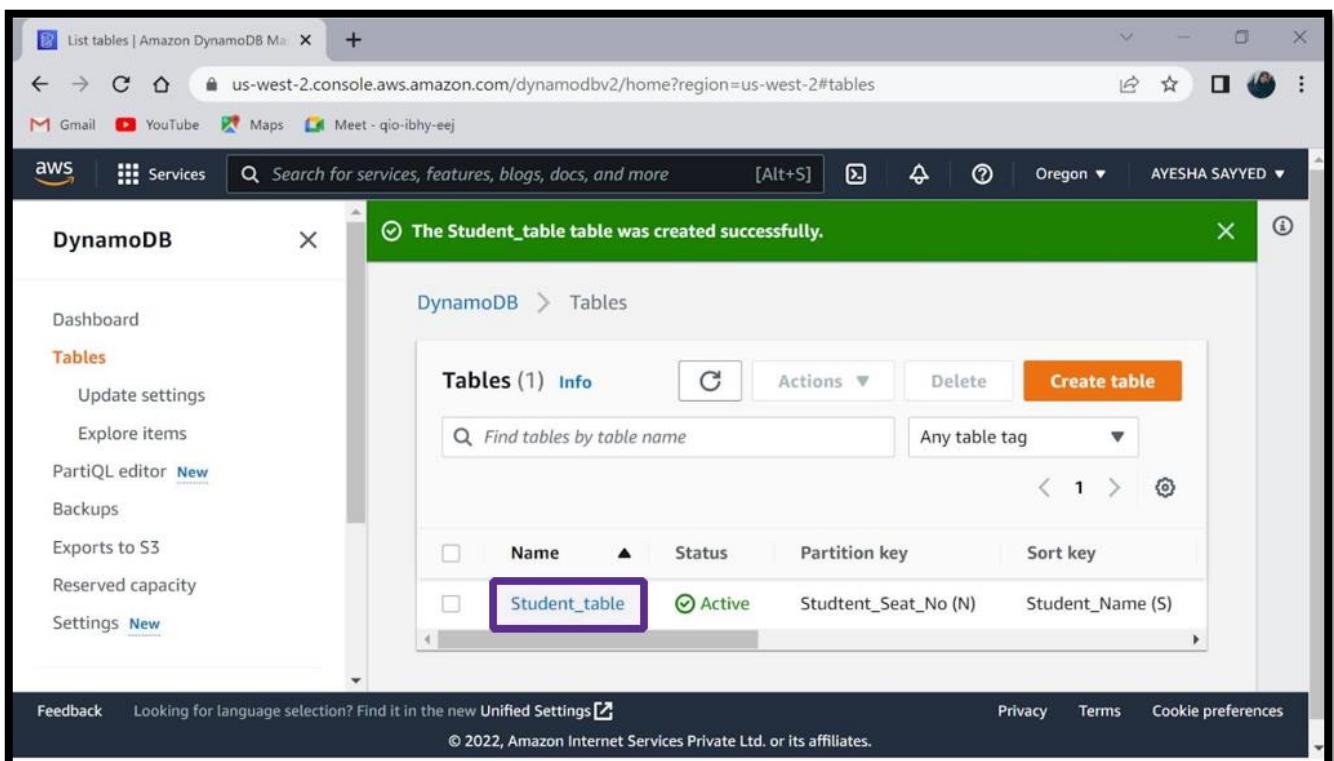
Step 4: Name your table → set ‘Partition key’ & ‘Sort key’ → select Default setting → Click on Create Table.

The screenshot shows the 'Create table' wizard in the AWS Management Console. The 'Table details' step is active. A green circle highlights the 'Table name' input field, which contains 'Student_table'. Below it, a note says 'Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)'. The 'Partition key' section is visible below, with a note about it being part of the primary key and used for retrieval and data allocation. The bottom navigation bar includes links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

The screenshot shows the 'Create table' wizard continuing through the steps. The 'Partition key' section is shown with a green circle around the 'Studtent_Seat_No' input field. A note indicates it's a hash value for retrieval and data allocation. The 'Sort key - optional' section follows, with a green circle around the 'Student_Name' input field. A note states it can be used as the second part of the primary key for sorting or searching. The 'Table settings' section is partially visible at the bottom. The bottom navigation bar includes links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.



Step 5: Table Created Successfully! From here select the blue link name as “Student_table”.



Step 6: Click on “Explore table item”→” then Create item”.

The screenshot shows the AWS DynamoDB console with the 'Student_table' selected. On the left, the navigation menu includes 'Dashboard', 'Tables', 'Update settings', 'Explore items', 'PartiQL editor', 'Backups', 'Exports to S3', 'Reserved capacity', and 'Settings'. The main area displays the 'Tables (1)' section with 'Student_table' listed. Below it, the 'Student_table' details page is shown with tabs for 'Overview', 'Indexes', and 'More'. A prominent orange button labeled 'Explore table items' is highlighted with a purple oval. At the bottom, there's a sharing message from 'meet.google.com' and standard browser controls.

The screenshot shows the 'Edit item' screen for the 'Student_table'. The top navigation bar includes 'Edit item | Amazon DynamoDB Manager' and the AWS logo. The main content area is titled 'Attributes' and contains three rows: 'Student_Seat_No - Partition key' with value '57890' and type 'Number'; 'Student_Name - Sort key' with value 'Reema' and type 'String'; and 'Student_Age' with value '21' and type 'Number'. A 'Remove' button is visible next to the 'Student_Age' row. At the bottom right, there are 'Cancel' and 'Create item' buttons, with the 'Create item' button highlighted with a red box. The footer includes a sharing message from 'meet.google.com', standard browser controls, and a copyright notice for Amazon.

As 1 item is created same way we need to create 9 more items.

The screenshot shows the 'Items' page in the Amazon DynamoDB console. A message at the top right says 'Completed' with a checkmark, indicating a successful operation. Below it, a table lists one item under 'Items returned (1)'. The item has attributes: 'Student_Seat_No' with value '57890' and 'Reem' (partially visible). The left sidebar includes links for Dashboard, Tables, Update settings, Explore items (which is highlighted in orange), PartiQL editor, Backups, Exports to S3, Reserved capacity, and Settings. The bottom of the page has standard footer links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

Step 7: Add new attributes if required→Click on “Create item”.

The screenshot shows the 'Edit item' page in the Amazon DynamoDB console. It displays a table of attributes for a student record. The 'Attribute name' column includes 'Student_Seat_No - Partition key' with value '57891' (Number type) and 'Student_Name - Sort key' with value 'Jasmin' (String type). A new attribute 'Student_Age' is being added, with its value '21' in the 'Value' field and 'Number' selected in the 'Type' dropdown. A green circle highlights the 'Student_Age' input field, and a green triangle highlights the 'Add new attribute' button. At the bottom right are 'Cancel' and 'Create item' buttons. The bottom of the page has standard footer links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

After entering each item select the “Create item” button and repeat the process until you are finished.

The screenshot shows the 'Edit item' screen in the Amazon DynamoDB Management Console. The 'Attributes' table contains the following data:

Attribute name	Value	Type
Student_Seat_No - Partition key	57892	Number
Student_Name - Sort key	James	String
Student_Age	20	Number

At the bottom right, there are 'Cancel' and 'Create item' buttons. The 'Create item' button is highlighted with a red box.

The screenshot shows the 'Items returned' list in the Amazon DynamoDB Management Console. The table displays the following data:

	Student_Seat_No	Student_Name	Student_Age
<input type="checkbox"/>	57893	Sahil	21
<input type="checkbox"/>	57892	James	20
<input type="checkbox"/>	57891	Jasmin	21
<input type="checkbox"/>	57890	Reema	21

At the top right of the list table, there is a 'Create item' button.

Items | Amazon DynamoDB Manager x New Tab us-west-2.console.aws.amazon.com/dynamodbv2/home?region=us-west-2#item-explorer?initialTagKey=&tab... Gmail YouTube Maps Meet - qio-ibhy-eej

aws Services Search for services, features, blogs, docs, and more [Alt+S] Actions Create item Oregon AYESHA SAYYED

DynamoDB

Dashboard Tables Update settings **Explore items** PartQL editor New Backups Exports to S3 Reserved capacity Settings New

Items returned (11)

	Student_Seat_No	Student_Name	Student_Age
<input type="checkbox"/>	57897	Aiman	21
<input type="checkbox"/>	57896	Hammad	20
<input type="checkbox"/>	57895	Susan	21
<input type="checkbox"/>	57894	Anjali	20

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates.

Step 8: Run query for needed items using partition key.

[1]

Items | Amazon DynamoDB Manager x New Tab us-west-2.console.aws.amazon.com/dynamodbv2/home?region=us-west-2#item-explorer?initialTagKey=&tab... Gmail YouTube Maps Meet - qio-ibhy-eej

aws Services Search for services, features, blogs, docs, and more [Alt+S] Autopreview View table details

DynamoDB

Dashboard Tables Update settings **Explore items** PartQL editor New Backups Exports to S3 Reserved capacity Settings New

Student_table

Scan/Query items

Scan/query a table or index

Scan Query Student_table

Student_Seat_No (Partition key)

57893

Student_Name (Sort key)

Equal to Enter sort key value Sort descending

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates.

The screenshot shows the AWS DynamoDB Item Explorer interface. On the left, a sidebar menu includes 'Dashboard', 'Tables' (selected), 'Update settings', 'Explore items' (highlighted in orange), 'PartiQL editor', 'Backups', 'Exports to S3', 'Reserved capacity', and 'Settings'. The main area displays a completed query result with 1 item returned. The table has columns: Student_Seat_No, Student_Name, and Student_Age. A row for '57893' is selected and highlighted with a yellow circle. The row data is Sahil and 21. Buttons for 'Run', 'Reset', 'Actions', and 'Create item' are visible at the top right of the results panel.

[2]

The screenshot shows the 'Scan/Query items' interface for the 'Student_table'. The sidebar menu is identical to the previous screenshot. The main panel shows a query configuration for 'Student_table'. The 'Scan' tab is selected. The 'Student_Seat_No' field is set as the partition key with the value '57893' circled in orange. The 'Student_Name' field is set as the sort key with the condition 'Equal to' and an input field 'Enter sort key value'. A checkbox for 'Sort descending' is present. A 'View table details' button is located in the top right corner of the main panel.

The screenshot shows the Amazon DynamoDB console. On the left, there's a sidebar with options like Dashboard, Tables, Update settings, Explore items (which is selected and highlighted in orange), PartiQL editor, Backups, Exports to S3, Reserved capacity, and Settings. The main area has a search bar at the top with 'Run' and 'Reset' buttons. Below it, a message says 'Completed' with 'Read capacity units consumed: 0.5'. A table titled 'Items returned (1)' is shown with columns: Student_Seat_No, Student_Name, and Student_Age. The first row contains the values '57895', 'Susan', and '21'. The value '57895' is circled in red.

Step 9: Finally, Select the table and click on “Delete”.

The screenshot shows the 'Delete table' dialog box. It starts with a message: 'You are about to delete a table.' followed by a list: '• Student_table'. There are two checkboxes: one checked ('Delete all CloudWatch alarms for this table.') and one unchecked ('Create a backup of this table before deleting it.'). Below is a note: 'If you do not select this check box, you will not be able to restore data being deleted.' A text input field contains the word 'delete'. At the bottom are 'Cancel' and 'Delete table' buttons, with 'Delete table' highlighted with a red box.

List tables | Amazon DynamoDB Manager New Tab

us-west-2.console.aws.amazon.com/dynamodbv2/home?region=us-west-2#tables

Gmail YouTube Maps Meet - qio-ibhy-eej

aws Services Search for services, features, blogs, docs, and more [Alt+S] Oregon AYESHA SAYYED

DynamoDB

The request to delete the "Student_table" table has been submitted successfully.

DynamoDB > Tables

Tables (1) Info Actions Delete Create table

Find tables by table name Any table tag

< 1 > ⚙

✓	Name	Status	Partition key	Sort key	Indexes	Re
✓	Student_table	⚠ Deleting	-	-	0	Pr

Feedback Looking for language selection? Find it in the new Unified Settings [↗](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Privacy Terms Cookie preferences

This screenshot shows the AWS DynamoDB console interface. A success message at the top states, "The request to delete the 'Student_table' table has been submitted successfully." Below this, the "Tables" section displays one table named "Student_table". The "Status" column for this table shows a warning icon followed by the text "Deleting". A red rectangular box highlights this status entry. The table also lists other columns: Name, Status, Partition key, Sort key, Indexes, and Re. At the bottom of the page, there is a footer with links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

EXPERIMENT – 05

Q1. What is AWS Lambda?

Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging.

With Lambda, you can run code for virtually any type of application or backend service. All you need to do is supply your code in one of the languages that Lambda supports.

Q2. What is serverless computing?

Serverless computing is a cloud architecture that allows organizations to get on-demand access to the resources they need. Customers only pay for the resources they use. Resources are not allocated to an application when it is not in use.

In a serverless computing architecture, a server's code execution is fully managed by the cloud provider. Therefore, the provider's customers do not need to develop and deploy the underlying infrastructure that would traditionally be required to run applications and programs. The primary objective of serverless computing is to make it easier for software developers to create code that is intended to run on cloud platforms and perform a clearly defined role.

Q3. What languages does AWS Lambda support?

- C#
- Go
- Java
- Node.js
- PowerShell
- Python
- Ruby

Q4. What is AWS DynamoDB Table?

- ▲ Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.
- ▲ DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.
- ▲ DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data.

Q5. Explain AWS IAM service?

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

Q6. To understand AWS Lambda, create your first Lambda functions using Python / Java / Nodejs.

Create AWS Lambda function and configure a trigger for Amazon Simple Storage Service (Amazon S3).

The trigger invokes your Lambda function every time that you add an object to your Amazon S3 bucket. Allow AWS Lambda to access Amazon DynamoDB Table.

Create IAM role that allows full access to DynamoDB Table.

Step 1: AWS Management Console Dashboard.

The screenshot shows the AWS Management Console Dashboard. On the left, there's a sidebar with 'Recently visited' services: AWS Budgets, Lambda, DynamoDB, Cloud9, Elastic Container Service, EC2, Lightsail, AWS Billing Conductor, S3, and CloudWatch. Below this is a 'View all services' link. To the right, there's a 'Welcome to AWS' section with three cards: 'Getting started with AWS' (Learn the fundamentals and find valuable information to get the most out of AWS), 'Training and certification' (Learn from AWS experts and advance your skills and knowledge), and 'What's new with AWS?' (Discover new AWS services, features, and Regions). At the bottom of the dashboard, there's a search bar, a feedback link, and a footer with copyright information and a cookie preferences link.

Step 2: Search for “IAM” and select it.

The screenshot shows the AWS Management Console search results for 'IAM'. A red box highlights the 'IAM' service result, which is described as 'Manage access to AWS resources'. Other search results include 'IAM Identity Center (successor to AWS Single Sign-On)', 'Resource Access Manager', and 'Amazon VPC IP Address Manager'. The sidebar on the left shows various AWS services like Features, Blogs, Documentation, and Events. The footer includes a search bar, feedback link, and footer links.

“IAM dashboard home page”.

The screenshot shows the AWS IAM Management Console home page. The left sidebar includes sections for Identity and Access Management (IAM), Access management, Access reports, and Related consoles. The main content area features a "Security recommendations" section with a red warning icon for "Add MFA for root user" and a green checkmark for "Root user has no active access keys". Below this is an "AWS Account" summary with fields for Account ID, Account Alias, and Sign-in URL. The "IAM resources" section displays counts for User groups (0), Users (0), Roles (5), Policies (2), and Identity providers (0). A "What's new" section lists recent changes like "Right-size permissions for more roles in your account using IAM Access Analyzer to generate 50 fine-grained IAM policies per day." On the right, there are "Quick Links" for My security credentials, Tools (Policy simulator), and other links. The bottom of the page shows the AWS navigation bar and system status.

Step 3: Go to role and click on “Create role”.

The screenshot shows the "Roles" page in the AWS IAM Management Console. The left sidebar highlights the "Roles" section under "Access management". The main content area shows a table of existing roles, each with a "Role name" and "Trusted entities" column. A blue "Create role" button is highlighted with a green box. Below the table, there is a "Roles Anywhere" section with a "Manage" button and icons for "Access AWS from your non AWS workloads" and "X.509 Standard". The bottom of the page follows the standard AWS navigation and status bar.

Step 4: Select lambda use case → Click on next.

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

EC2 Allows EC2 instances to call AWS services on your behalf.

Lambda Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

DynamoDB

Cancel Next

Step 5: Search Amazon DynamoDB in permission policies → Select policy which provides “full access to DynamoDB” → Click on next.

IAM Management Console

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles/create?commonUseCase=Lambda&step=addPermission&trustedEntityType=AWS_SERVICE

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions

Permissions policies (Selected 1/762)

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter

"DYNAMODB" X Clear filters

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon DynamoDB via the AWS Management Console.
<input type="checkbox"/> AWSLambdaDynamoDBExecutionRole	AWS managed	Provides list and read access to DynamoDB streams and write permissions to CloudWatch Metrics.
<input type="checkbox"/> AmazonDynamoDBReadOnlyAccess	AWS managed	Provides read only access to Amazon DynamoDB via the AWS Management Console.
<input type="checkbox"/> AWSLambdaInvokeRole	AWS managed	Provides read access to DynamoDB Streams.

Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous Next

Step 6: Give name to your role→ Click on ‘Create role’.

The screenshot shows the AWS IAM Management Console interface for creating a new role. The top navigation bar includes the AWS logo, services menu, search bar, and user information (AYESHA SAYYED). The main content area is titled 'Name, review, and create' and is divided into three steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create).
Step 3: Name, review, and create
Under 'Role details', the 'Role name' field is highlighted with a green circle and contains the value 'sayyed_ayesha_045'. Below it is a note: 'Maximum 24 Characters. Use alphanumeric and '+-, @-' characters.'
The 'Description' section contains the text: 'Allows Lambda functions to call AWS services on your behalf.'
Step 1: Select trusted entities
This step displays a JSON policy document:

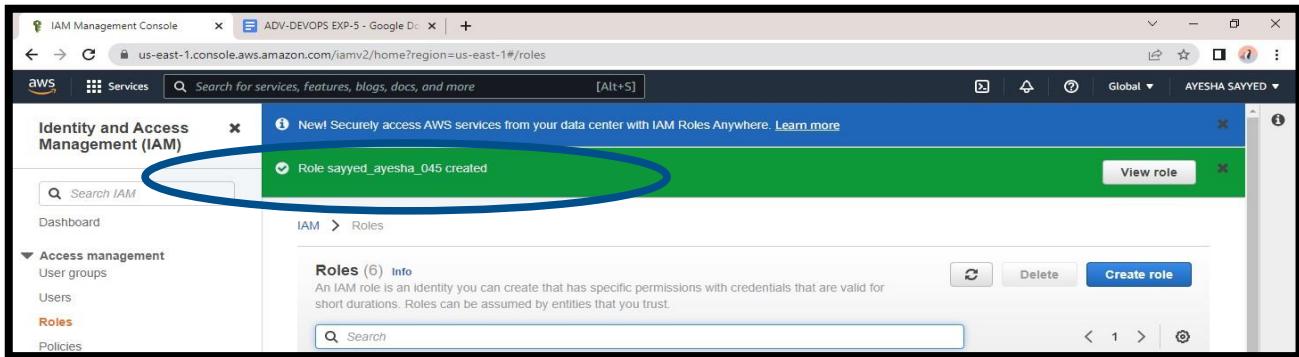
```
1 - {}
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "lambda:InvokeFunction"
8       ]
9     }
10   ]
11 }
```

Step 2: Add permissions
This step shows a 'Permissions policy summary' table:

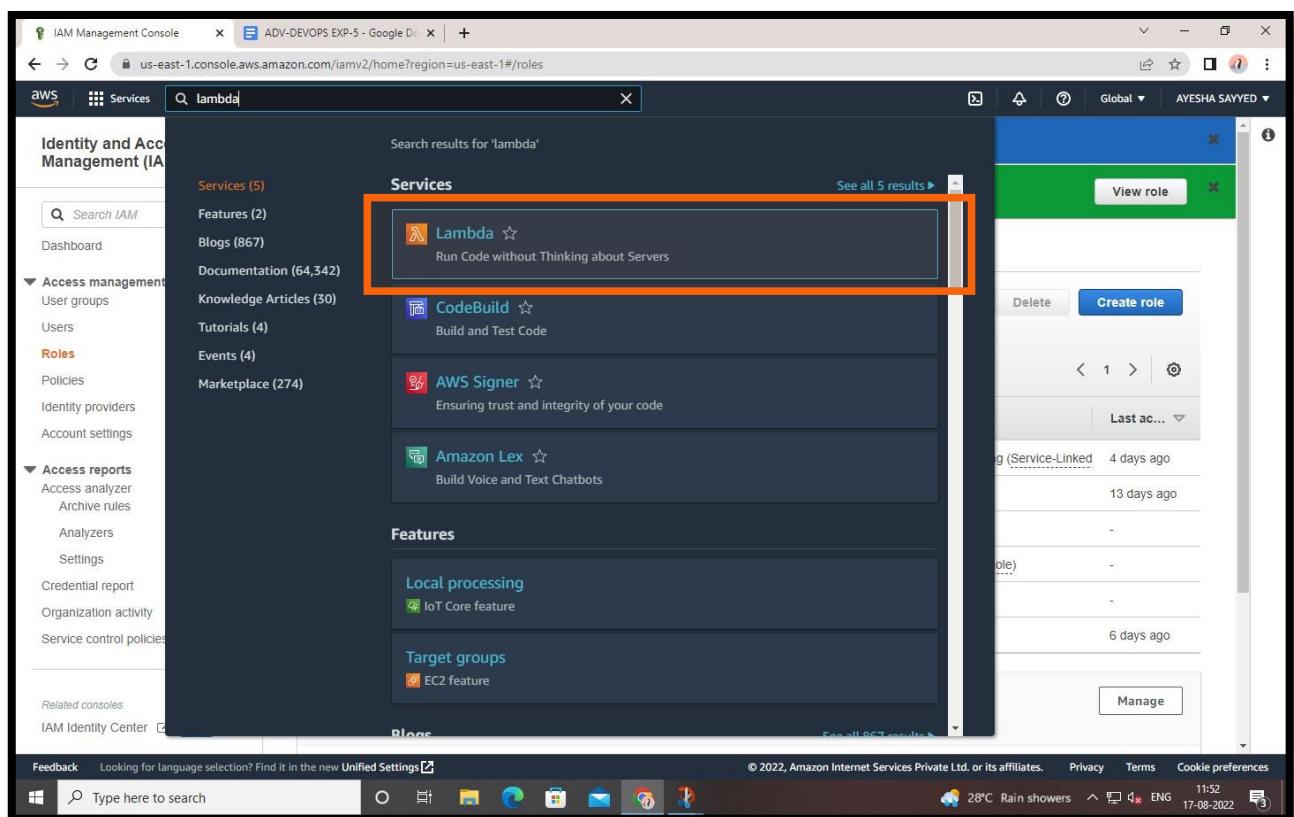
Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy

Tags
A section for adding optional tags, with a note: 'Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.'
A 'Add tag' button is present, along with a note: 'You can add up to 50 more tags.'
Action Buttons
At the bottom right of the main form are three buttons: 'Cancel', 'Previous', and 'Create role', which is highlighted with a green box.

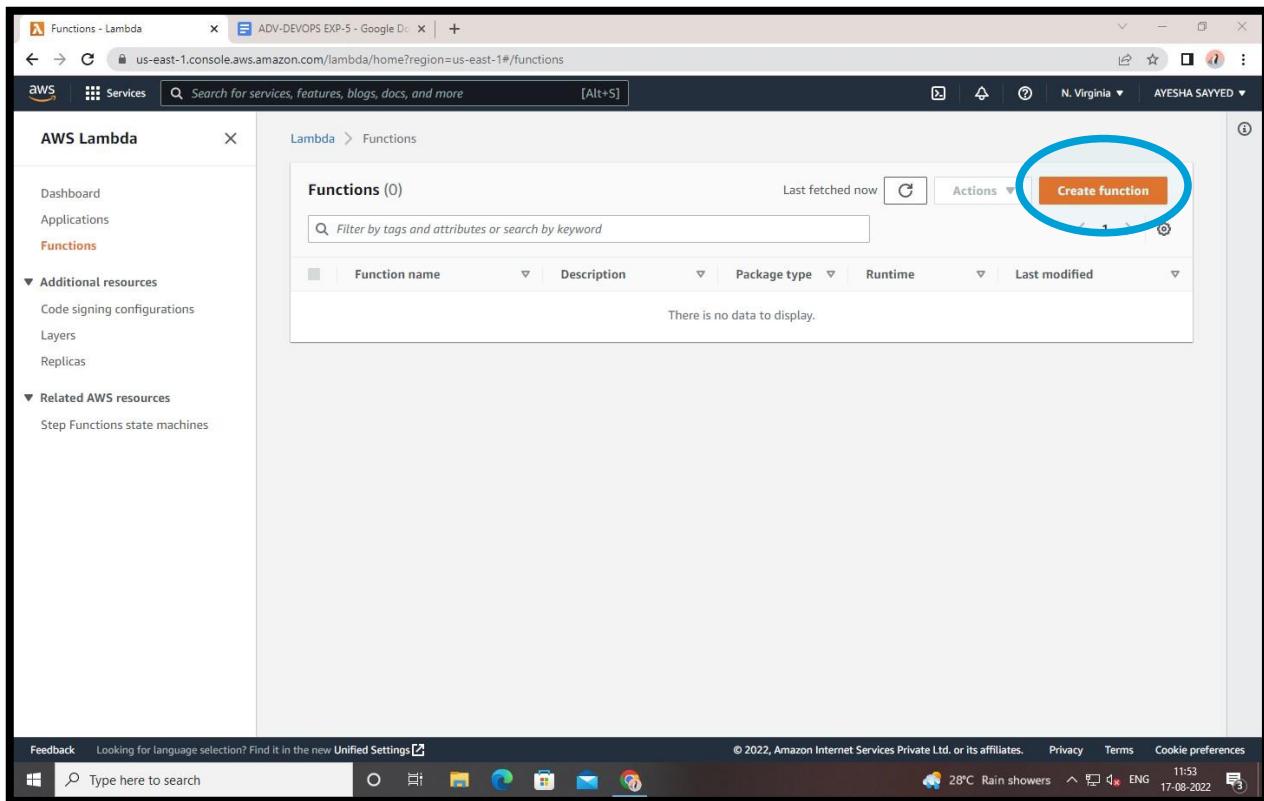
Role created successfully!



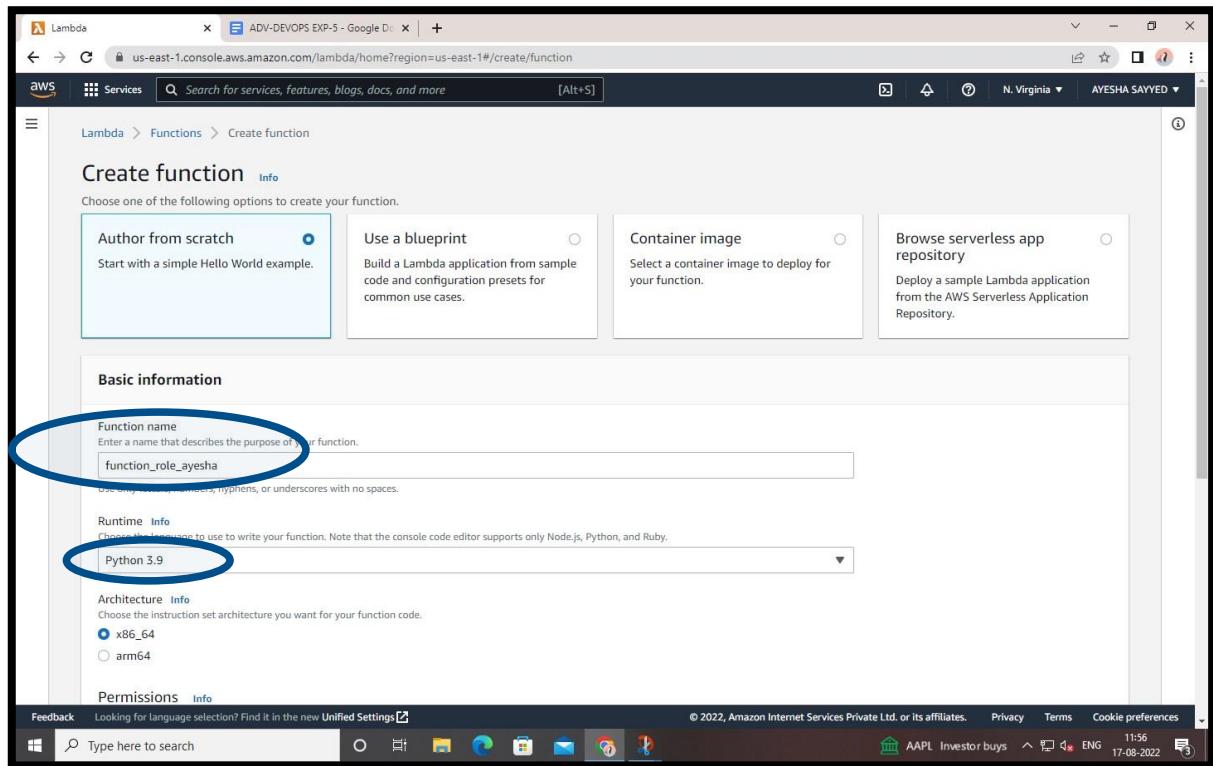
Step 7: Go to services→ click on “Lambda”.



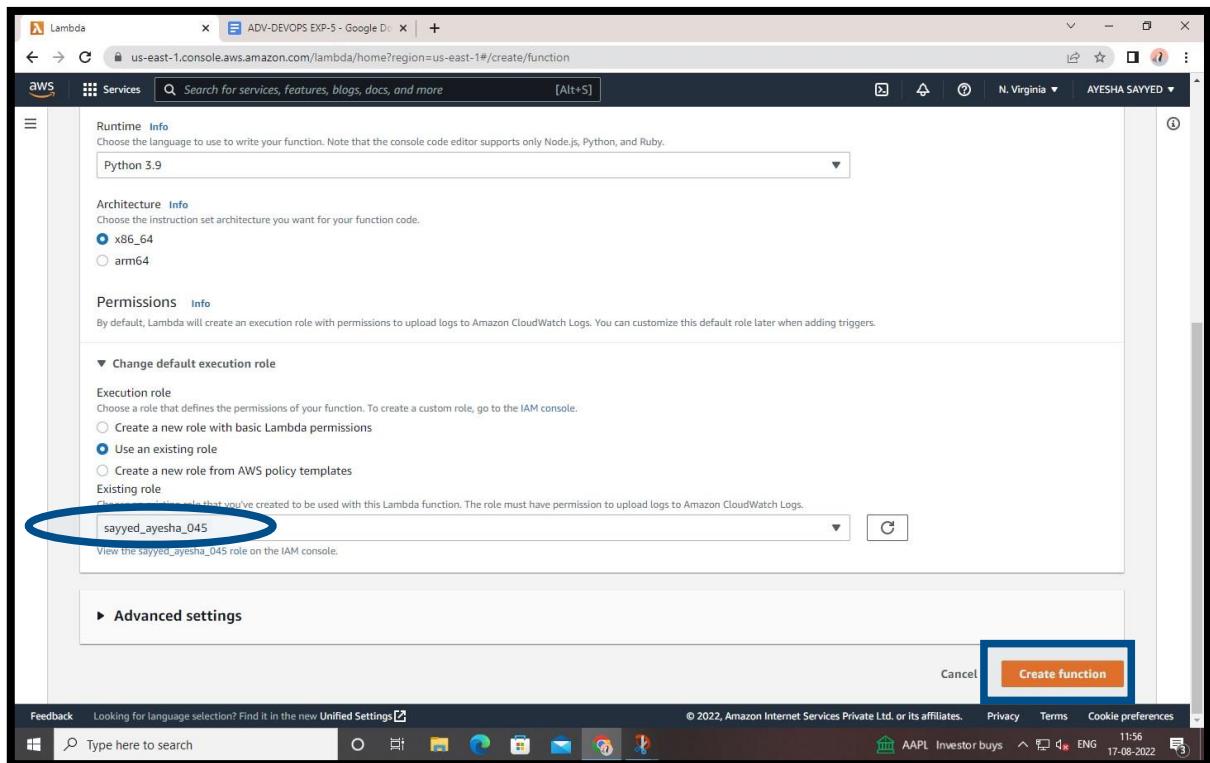
Step 8: Click on Create function.



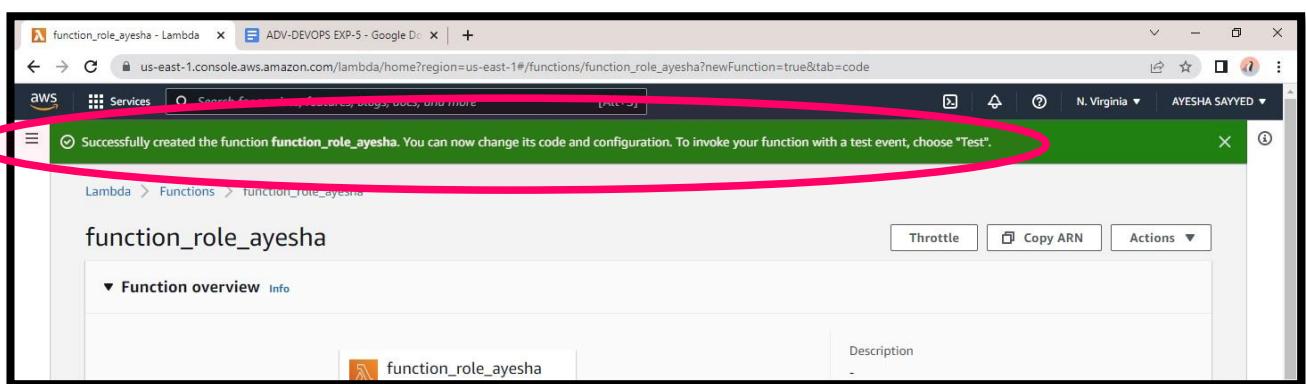
Step 9: Choose “Author from Scratch” → Give name to your function→ Choose “Python” language in runtime.



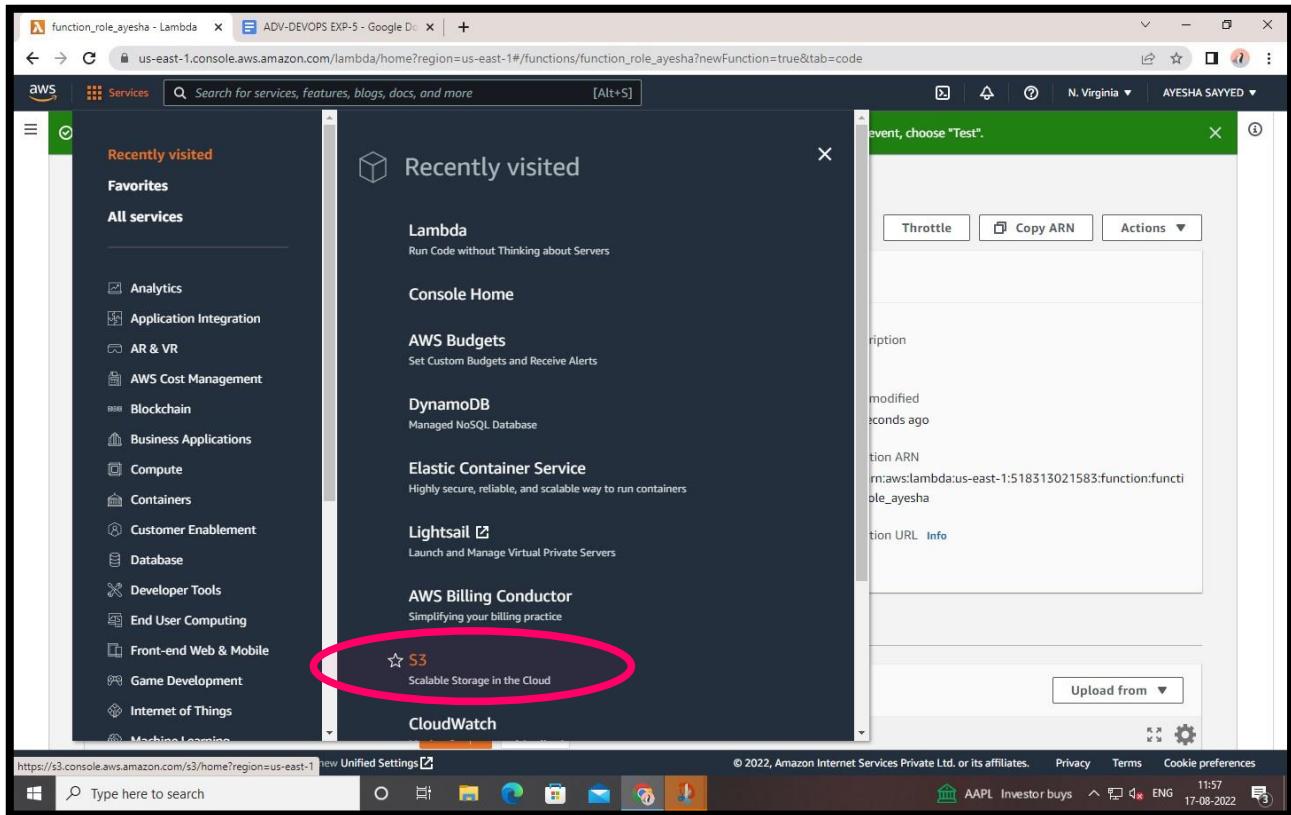
Step 10: Permissions → Change execution role → Choose “Use an existing role” → Click on Create function.



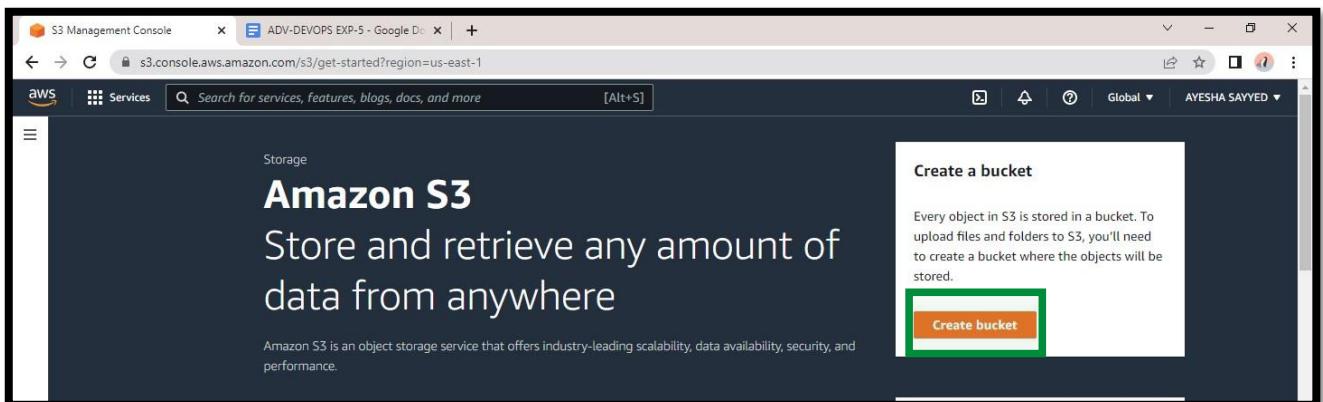
Function created successfully!



Step 11: Go to Services → Select S3.



Step 12: Click on “Create bucket”.



Step 13: Give name to your bucket.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' step is active, with the 'Bucket name' field set to 'ayeshasyeed45'. A green oval highlights this field. The 'AWS Region' is set to 'US East (N. Virginia) us-east-1'. The 'Object Ownership' section shows 'ACLs disabled (recommended)' selected. The 'Block Public Access settings for this bucket' section has 'Block all public access' checked. The 'Bucket Versioning' section has 'Disable' selected. The 'Tags (0) - optional' section shows 'No tags associated with this bucket'. The 'Default encryption' section has 'Disable' selected. At the bottom, a note says 'After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.' The 'Create bucket' button is highlighted with a green rectangle.

The screenshot shows the AWS S3 Management Console. At the top, there is a green success message: "Successfully created bucket 'ayeshasayyed45'. To upload files and folders, or to configure additional bucket settings choose View details." This message is highlighted with a red oval. Below the message, the main interface shows an "Account snapshot" section and a table of buckets. The table has one entry: "Buckets (1) Info". The bucket details are as follows:

Name	AWS Region	Access	Creation date
ayeshasayyed45	US East (N. Virginia) us-east-1	Bucket and objects not public	August 17, 2022, 12:01:18 (UTC+05:30)

Step 14: Search for Lambda and select it.

The screenshot shows the AWS IAM Management Console. The search bar at the top contains the text "lambda". The search results are displayed under the "Services" section, with the "Lambda" service highlighted and surrounded by a yellow box. The Lambda service entry includes the description "Run Code without Thinking about Servers". To the right of the search results, there is a sidebar showing a list of roles, with the first role "Last ac..." selected. The sidebar also includes buttons for "Delete" and "Create role".

Click on ‘Lambda Function Name’.

The screenshot shows the AWS Lambda Functions console. A single function named 'function_role_ayesha' is listed in the 'Functions (1)' table. The 'Function name' column contains the value 'function_role_ayesha', which is circled in yellow. The table also includes columns for Description, Package type, Runtime, and Last modified.

Upload ‘Code’ to the Lambda Function and Deploy it.

The screenshot shows the AWS Lambda Code source editor. The code editor displays a Python script named 'lambda_function.py'. The entire code block is circled in red.

```
1 import boto3
2 from uuid import uuid4
3 def lambda_handler(event, context):
4     s3 = boto3.client("s3")
5     dynamodb = boto3.resource('dynamodb')
6     for record in event['Records']:
7         bucket_name = record['s3']['bucket']['name']
8         object_key = record['s3']['object']['key']
9         size = record['s3']['object'].get('size', -1)
10        event_name = record['eventName']
11        event_time = record['eventTime']
12        dynamoTable = dynamodb.Table('Ayesha_Sayyed_45')
13        item = {
14            "unique": str(uuid4()),
15            "Bucket": bucket_name,
16            "Object": object_key,
17            "Size": size,
18            "Event": event_name,
19            "EventTime": event_time}
```

Then click on “Add Trigger” and add bucket from S3 into the trigger, so that it can generate trigger whenever something is added to the bucket.

The screenshot shows the AWS Lambda Function overview page for the function 'function_role_ayesha'. The left sidebar shows the function name and a '+ Add trigger' button. The right panel displays the function's ARN, last modified time (7 minutes ago), and a 'Function URL' section.

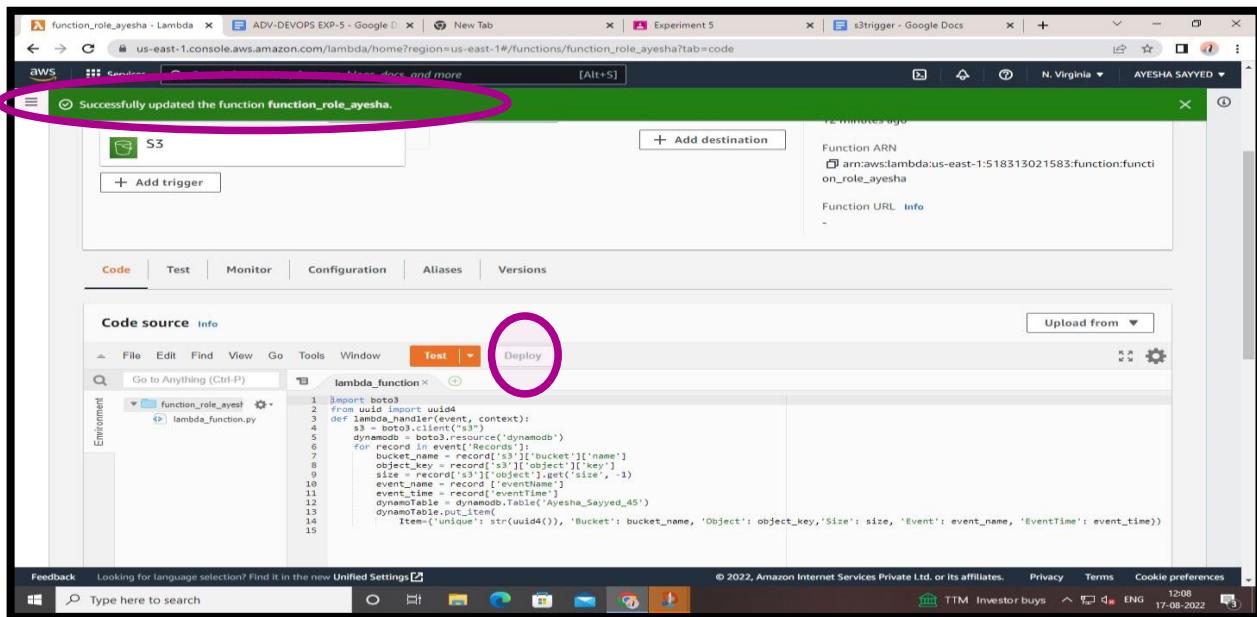
Select ‘S3’ add your ‘bucket’ click to the checkbox of “I Acknowledge” & then click to “Add”.

The screenshot shows the AWS Lambda console interface. A specific step in the 'Add trigger' configuration is highlighted:

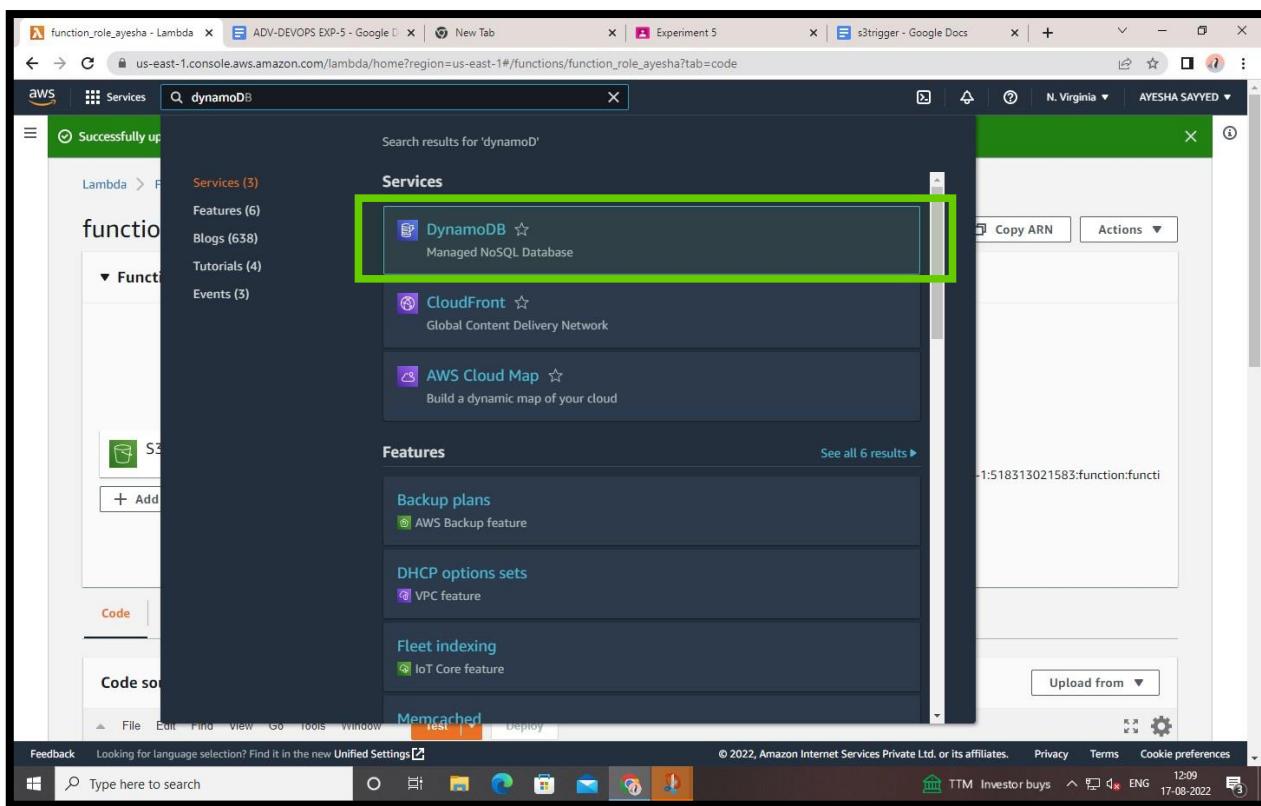
- Bucket:** The input field contains "s3/ayeshasayyed45", which is circled in green.
- Event type:** Set to "All object create events".
- Prefix - optional:** "e.g. images/"
- Suffix - optional:** "e.g. jpg"
- Recursive invocation:** A note about using different S3 buckets for input and output.
- I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.** This checkbox is checked.
- Add** button: Located at the bottom right of the configuration panel, circled in purple.

At the bottom of the browser window, there is a navigation bar with links like Feedback, Type here to search, and various system icons.

Step 15: Trigger added successfully! Also Save the code→Deploy it.



Step 16: Go to services → Select DynamoDB.



Step 17: Create the table.

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The 'Table details' section is visible, with the 'Table name' field set to 'Ayesha_Sayyed_45' and the 'Partition key' field set to 'unique'. Both fields are circled in yellow. The 'Table settings' section shows the following configuration:

Setting	Value	Editable after creation
Capacity mode	Provisioned	Yes
Read capacity	5 RCU	Yes
Write capacity	5 WCU	Yes
Auto scaling	On	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	Owned by Amazon DynamoDB	Yes
Table class	DynamoDB Standard	Yes

The 'Tags' section indicates no tags are associated with the resource. A 'Create table' button is at the bottom right.

Created Table Successfully!!

The screenshot shows the AWS DynamoDB service in the AWS Management Console. A yellow circle highlights a green success message at the top of the main content area: "The Ayesha_Sayyed_45 table was created successfully." Below this, there is a table list titled "Tables (1) Info". The table listed is "Ayesha_Sayyed_45", which is Active, has a unique partition key, and is provisioned with auto scaling for both read and write capacity.

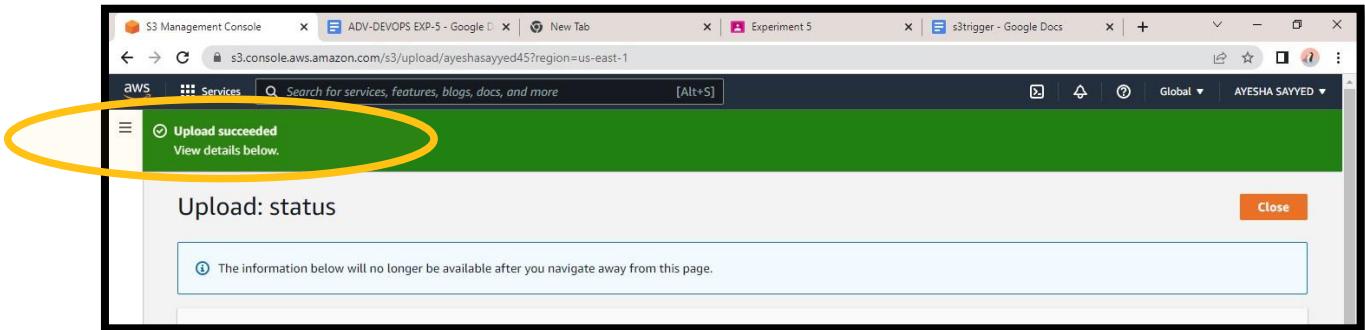
Step 18: Go to S3 → Bucket and upload an object by clicking on add files.

The screenshot shows the AWS Lambda service in the AWS Management Console. A yellow box highlights the "S3" service in the "Recently visited" list. To the right of the list, there is a detailed view of a Lambda function named "function_role_ayesha", showing its ARN, Throttle settings, and Actions menu.

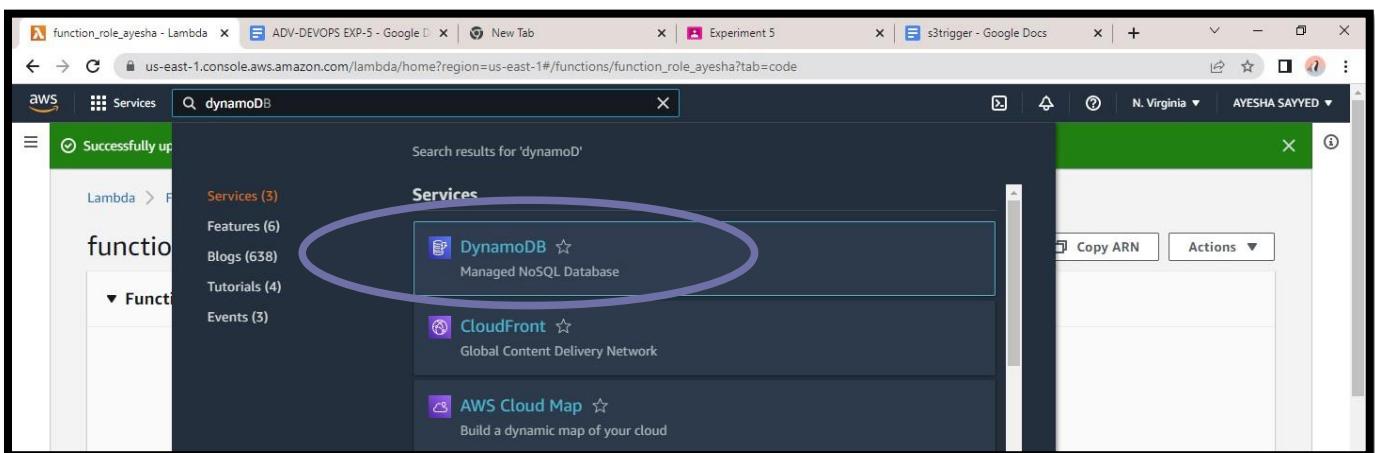
The screenshot shows the AWS S3 Management Console. The URL is s3.console.aws.amazon.com/s3/buckets/ayeshasayyed45?region=us-east-1&tab=objects. The page displays the 'Objects (0)' section, which states: "Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory [?] to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more [?].". Below this, there are several buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. The 'Upload' button is highlighted with a yellow box. A search bar labeled "Find objects by prefix" is also present. At the bottom, there is a table header with columns: Name, Type, Last modified, Size, and Storage class. The main content area below the table header says "No objects" and "You don't have any objects in this bucket." with a "Upload" button.

The screenshot shows the AWS S3 Management Console in the 'Upload' step. The URL is s3.console.aws.amazon.com/s3/upload/ayeshasayyed45?region=us-east-1. The page title is "Upload" and the breadcrumb navigation shows "Amazon S3 > Buckets > ayeshasayyed45 > Upload". The main area contains a large dashed box with the text "Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.". Below this, there is a table titled "Files and folders (1 Total, 141.9 KB)". It lists one file: "437c216e57fb36869301be60eae410d.jpg" with a size of "141.9 KB". There are "Remove", "Add files", and "Add folder" buttons above the table. The "Add files" button is highlighted with a yellow box. A "Destination" section follows, showing "Destination" set to "s3://ayeshasayyed45". A "Destination details" section is expanded, stating "Bucket settings that impact new objects stored in the specified destination." At the bottom, there is a "Feedback" link and a search bar.

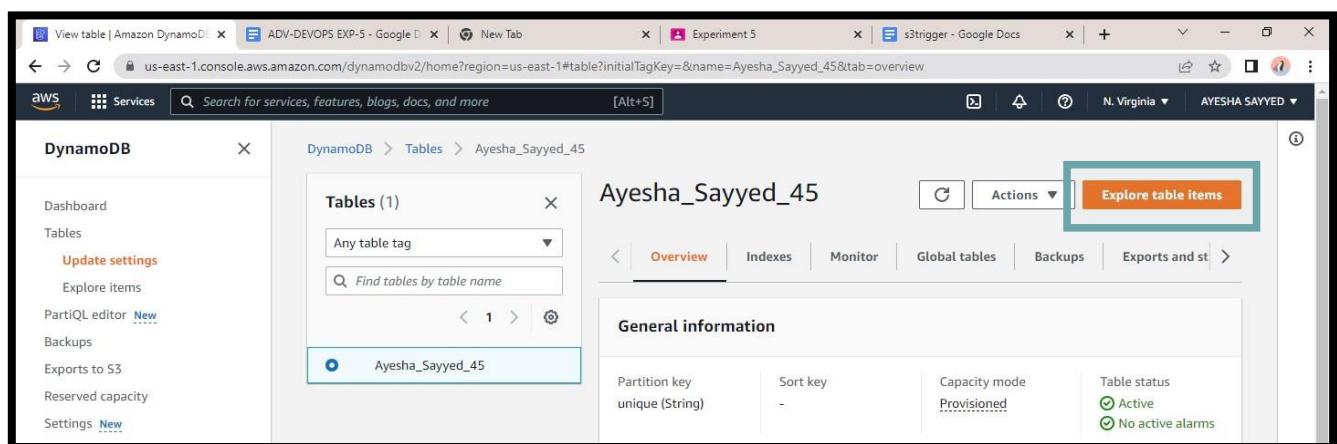
Successfully Uploaded.



Step 19: Go to DynamoDB to check whether the items are updated in the table or not.



Step 20: DynamoDB → Tables → update settings → Explore table items.



The screenshot shows the AWS DynamoDB Items page for a table named 'Ayesha_Sayyed_45'. A single item is listed in the 'Items returned' section:

	unique	Bucket	Event	EventTime	Object	Size
<input type="checkbox"/>	95b32a9f-2b50-41f3...	ayesha_sayy...	ObjectCreat...	2022-08-1...	437c216e5...	145295

Step 21: Again, go to S3 to upload 2 to 3 items in the table.

The screenshot shows the AWS Services search results page. The search term 's3' has been entered into the search bar. The results list includes the S3 service, which is highlighted with a pink rectangle.

Upload → add files → click on upload button.

The screenshot shows the AWS S3 Management Console in a browser window. The URL is `s3.console.aws.amazon.com/s3/upload/ayeshasayyed45?region=us-east-1`. The page title is "Upload".
The main area has a heading "Upload" with a link "Info". Below it is a note: "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more [?]".
A dashed box contains the instruction: "Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.". Below this is a table titled "Files and folders (2 Total, 124.7 KB)".
Table columns: Name, Folder, Type, Size.
Rows:

- computer-laptop-notebook-pen.jpg - image/jpeg 69.9 KB
- f3b58f860a9e8487e2eb7e853e1f4f8e.jpg - image/jpeg 54.8 KB

Below the table is a section titled "Destination" with a dropdown menu set to "s3://ayeshasayyed45".
A collapsed section titled "Destination details" is shown.
A collapsed section titled "Permissions" is shown.
A collapsed section titled "Properties" is shown.
At the bottom right is a large orange "Upload" button.
The browser status bar shows: Feedback, Looking for language selection? Find it in the new Unified Settings [?], © 2022, Amazon Internet Services Private Ltd. or its affiliates., Privacy, Terms, Cookie preferences, 29°C Rain showers, ENG, 12:19, 17-08-2022.

Step 22: Again, go to DynamoDB to check whether the rest of the items are updated in the table or not.

Step 23: Repeat the steps DynamoDB → Tables → update settings → Explore table items, all items added successfully.

The screenshot shows the AWS DynamoDB Items page for the table 'Ayesha_Sayyed_45'. The left sidebar includes options like Dashboard, Tables, Update settings, Explore items (which is selected), PartiQL editor, Backups, Exports to S3, Reserved capacity, and Settings. The main content area displays a table titled 'Items returned (3)'. The table has columns: unique, Bucket, Event, EventTime, Object, and Size. The data rows are:

unique	Bucket	Event	EventTime	Object	Size
5efa06ab-075f-4b1d...	ayeshasayy...	ObjectCreat...	2022-08-1...	f3b58f860a...	56068
95b32a9f-2b50-41f3-...	ayeshasayy...	ObjectCreat...	2022-08-1...	437c216e5...	145295
62540a40-7cb5-40e3...	ayeshasayy...	ObjectCreat...	2022-08-1...	computer-l...	71604

Step 24: Now delete ‘Role’ which was created in ‘IAM Dashboard’

The screenshot shows the AWS IAM Roles page. The left sidebar lists Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), Access reports (Access analyzer, Analyzers, Credential report, Organization activity, Service control policies (SCPs)), and Related consoles (IAM Identity Center). A modal dialog box is open, asking 'Delete sayyed_ayesha_045?'. It contains a warning message: 'Delete sayyed_ayesha_045 permanently? This will also delete all its inline policies and any attached instance profiles.' Below this is a 'Role name' field containing 'sayyed_ayesha_045' and a note: 'Recent activity usually appears within 4 hours. Data is stored for a maximum of 365 days, depending when your region began supporting this feature.' At the bottom of the dialog, there is a text input field with 'sayyed_ayesha_045' and a 'Delete' button highlighted with a green box. The background shows a list of other roles.

Role Deleted Successfully!!

The screenshot shows the AWS IAM Management Console. A red circle highlights a green notification bar at the top right containing the message "Role deleted sayyed_ayesha_045". The left sidebar shows "Identity and Access Management (IAM)" selected. The main pane displays a list of roles with a count of 5.

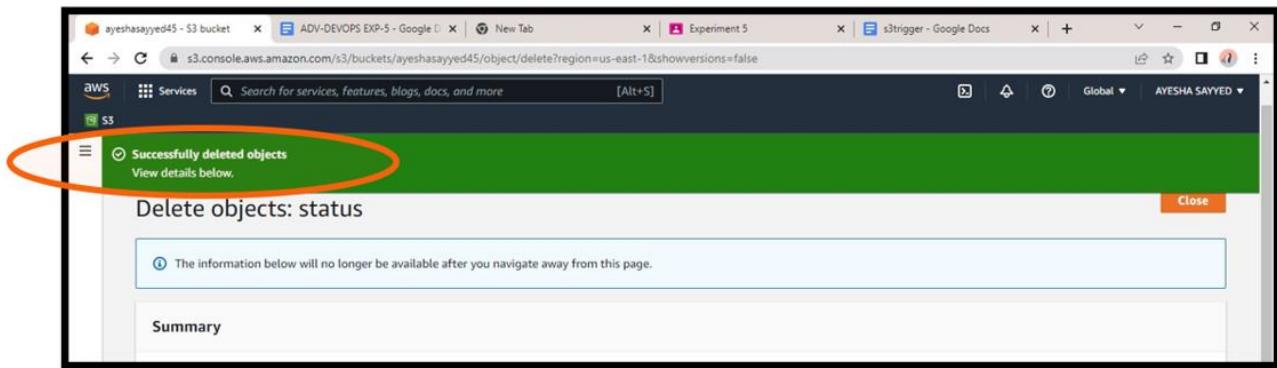
Step 25: Delete Lambda Function which was created → Also delete the table which was created in DynamoDB.

Step 26: Delete the 'objects' from the 'bucket' created in 'S3'.

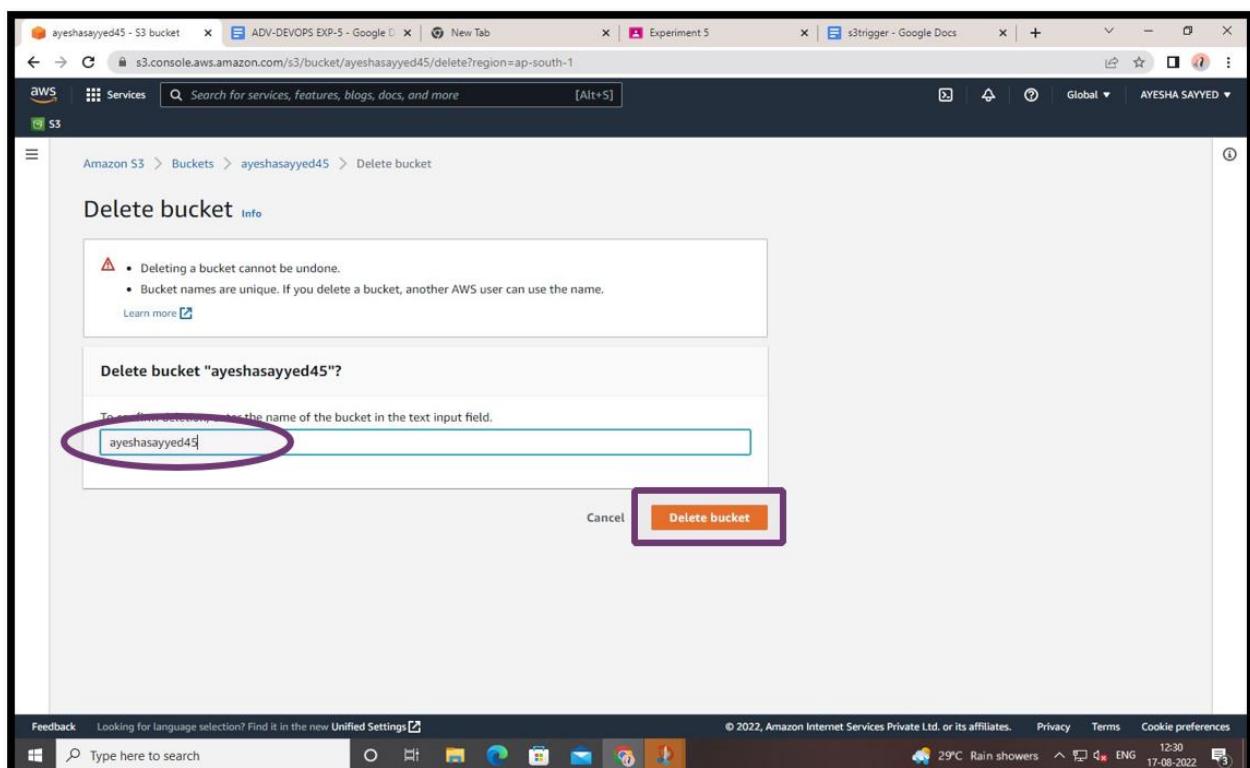
The screenshot shows the AWS S3 console. A red circle highlights the "permanently delete" button in the "Permanently delete objects?" dialog box. The dialog also contains a text input field with the placeholder "To confirm deletion, type permanently delete in the text input field." The main pane shows a list of three jpg files in the "ayeshasayyed45 - S3 bucket".

Name	Type	Last modified	Size
437c216e57fb36869301be60eae410d.jpg	jpg	August 17, 2022, 12:15:33 (UTC+05:30)	141.9 KB
computer-laptop-notebook-pen.jpg	jpg	August 17, 2022, 12:20:04 (UTC+05:30)	69.9 KB
f3b58f860a9e8487e2eb7e853e1f4f8e.jpg	jpg	August 17, 2022, 12:20:02 (UTC+05:30)	54.8 KB

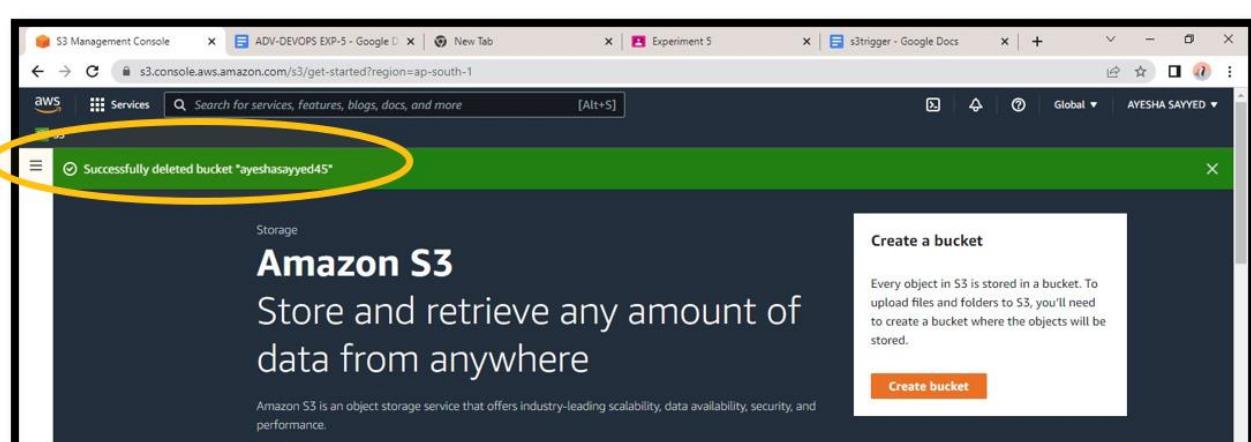
Object Deleted Successfully!!



Now delete the bucket.



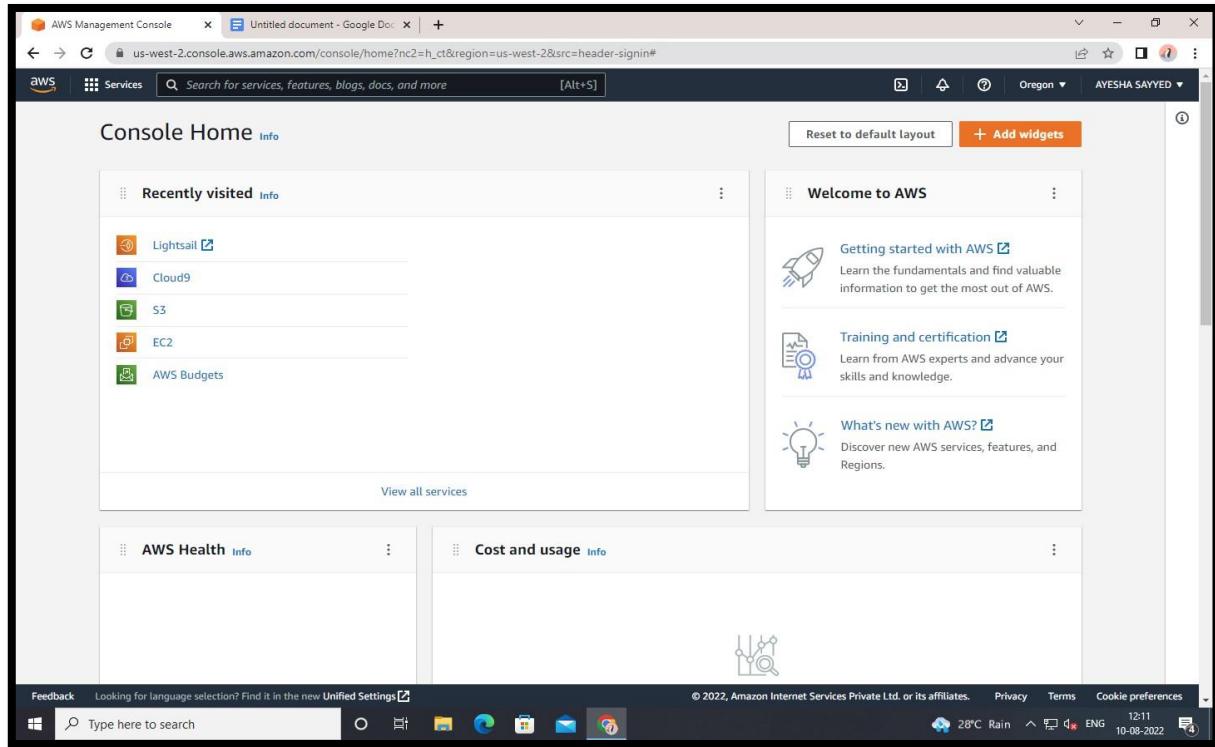
Bucket Deleted Successfully!!



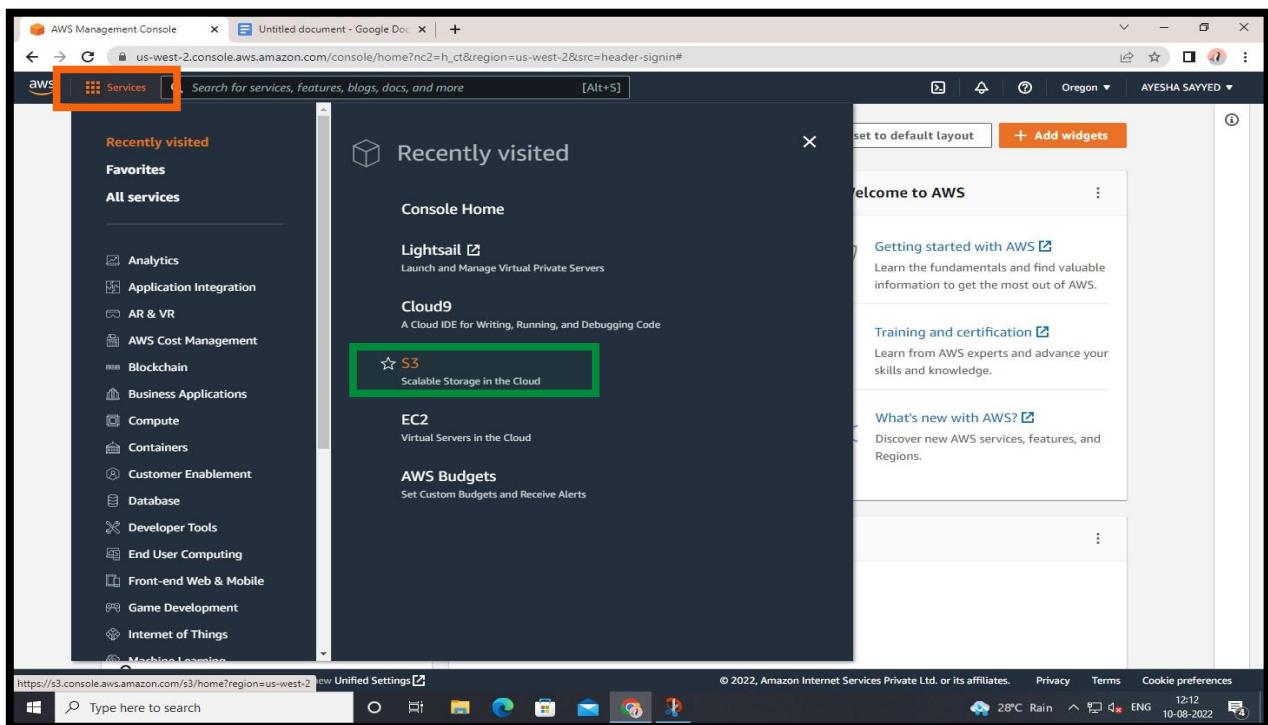
EXPERIMENT – 06

Q1. To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3. Use AWS Lambda blueprint.

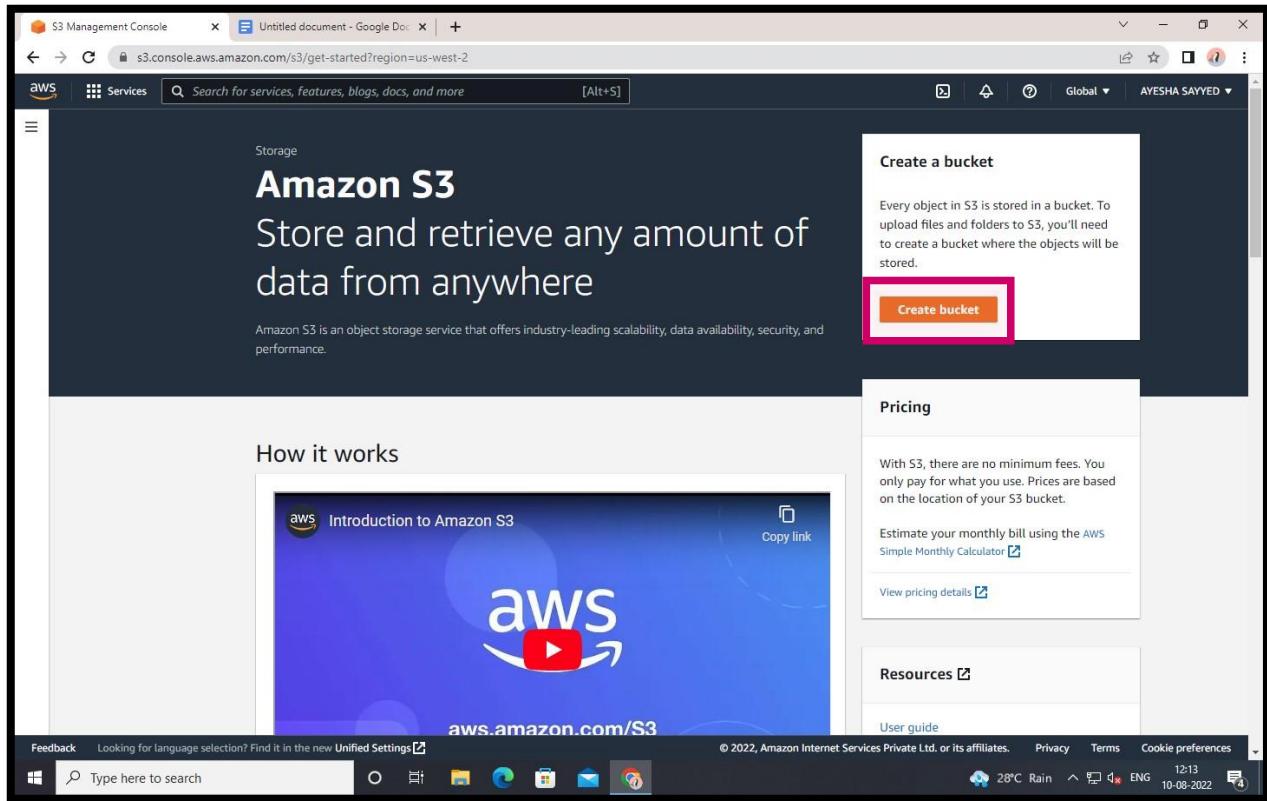
Step 1: AWS Management Console Dashboard.



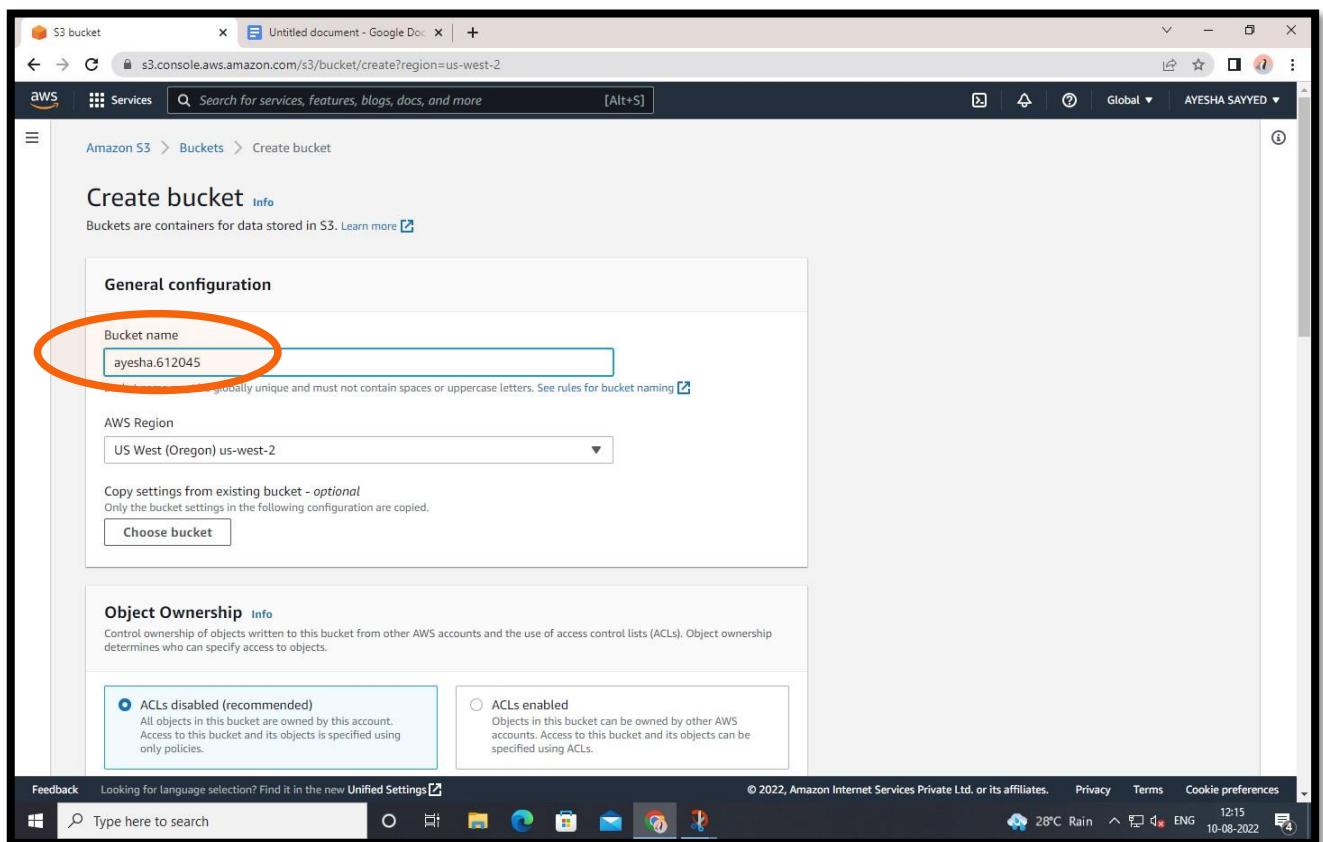
Step 2: Click on Services→Storage→S3



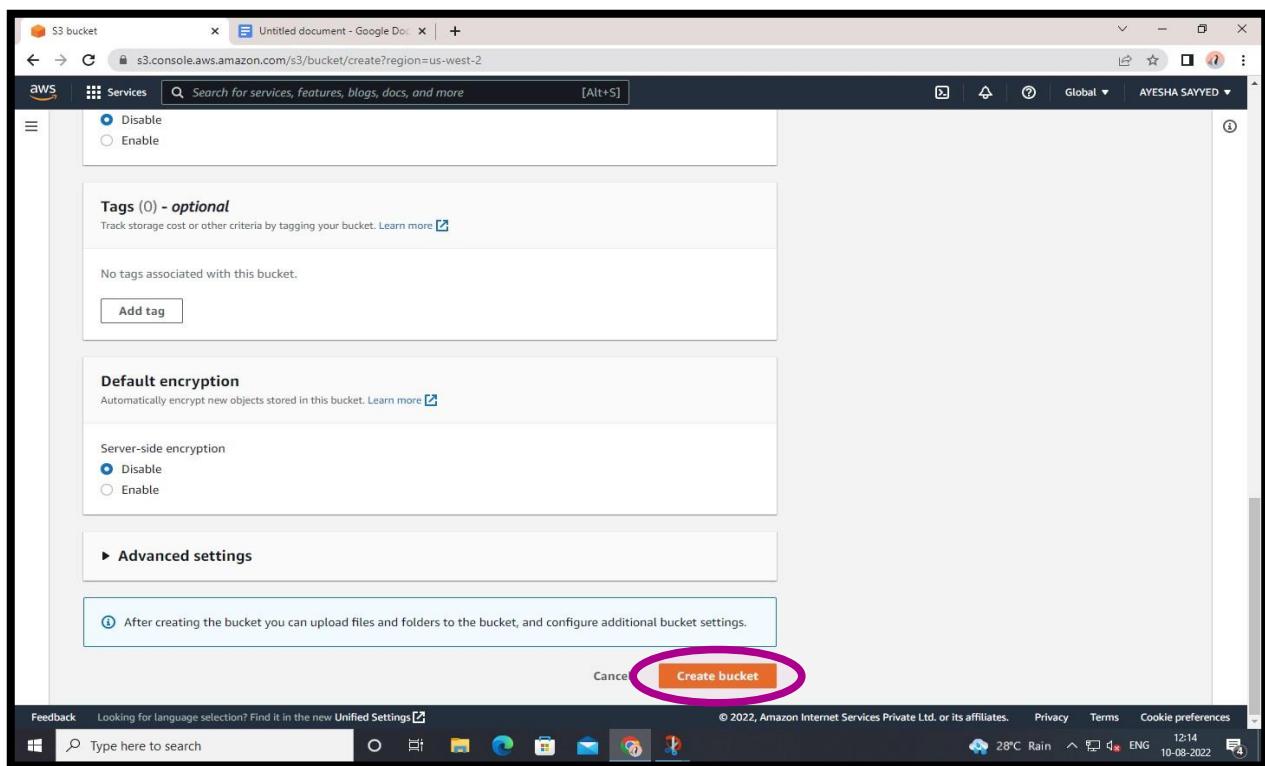
Step 3: Click on “Click Bucket”.



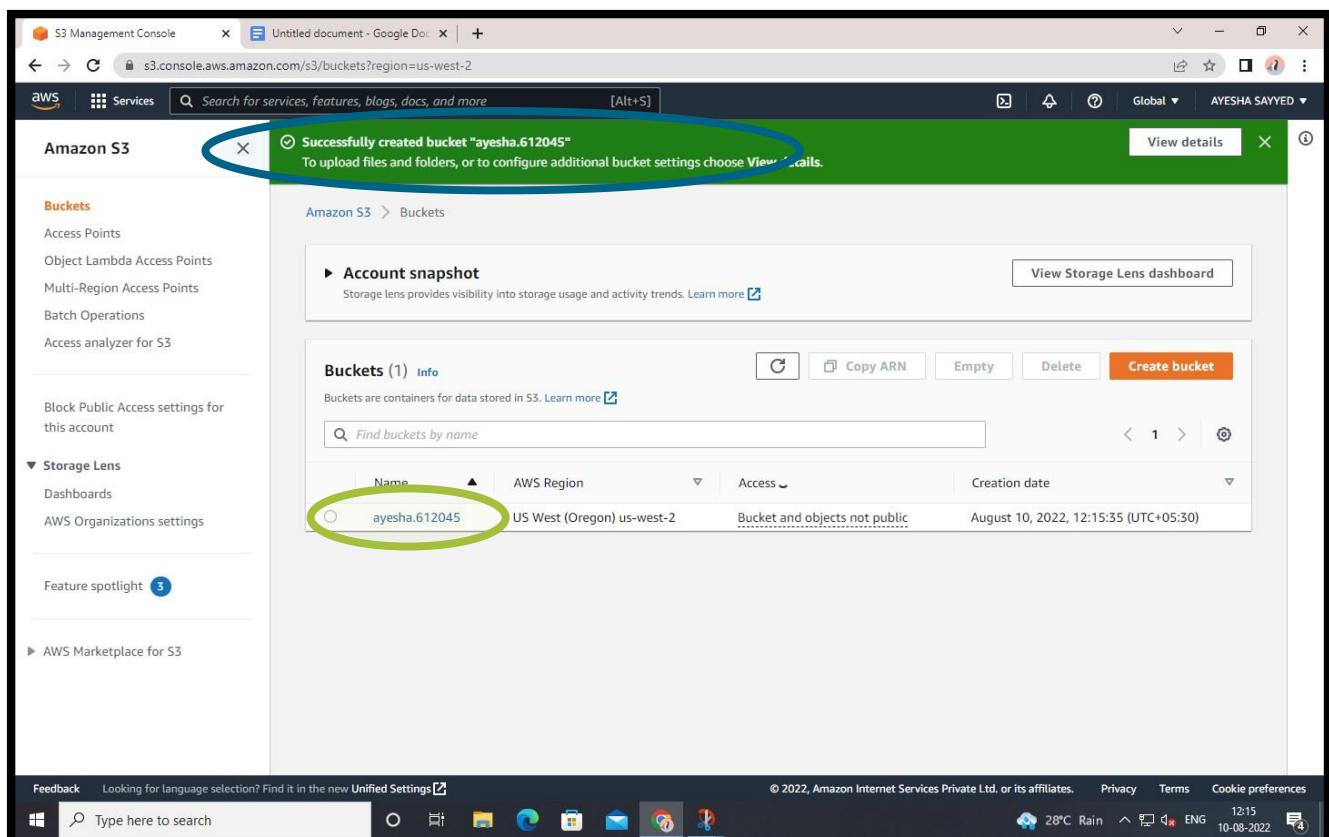
Step 4: Give your bucket a name→Click on Create bucket.



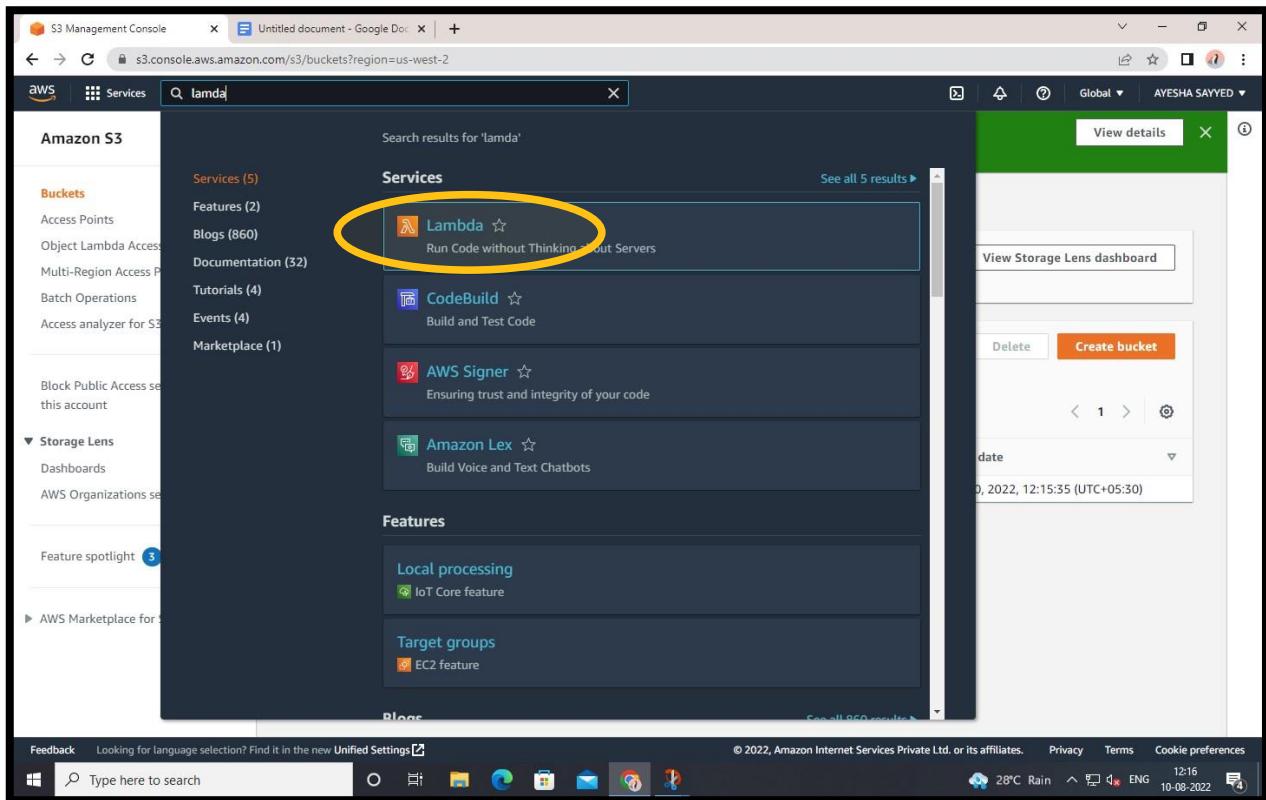
Then click on the button ‘Create Bucket’.



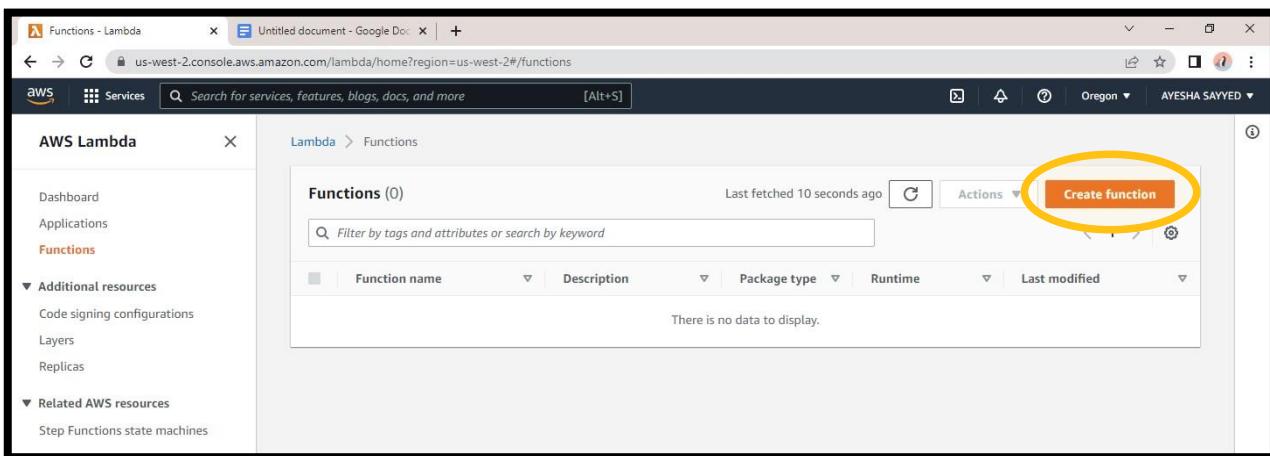
Bucket successfully created.



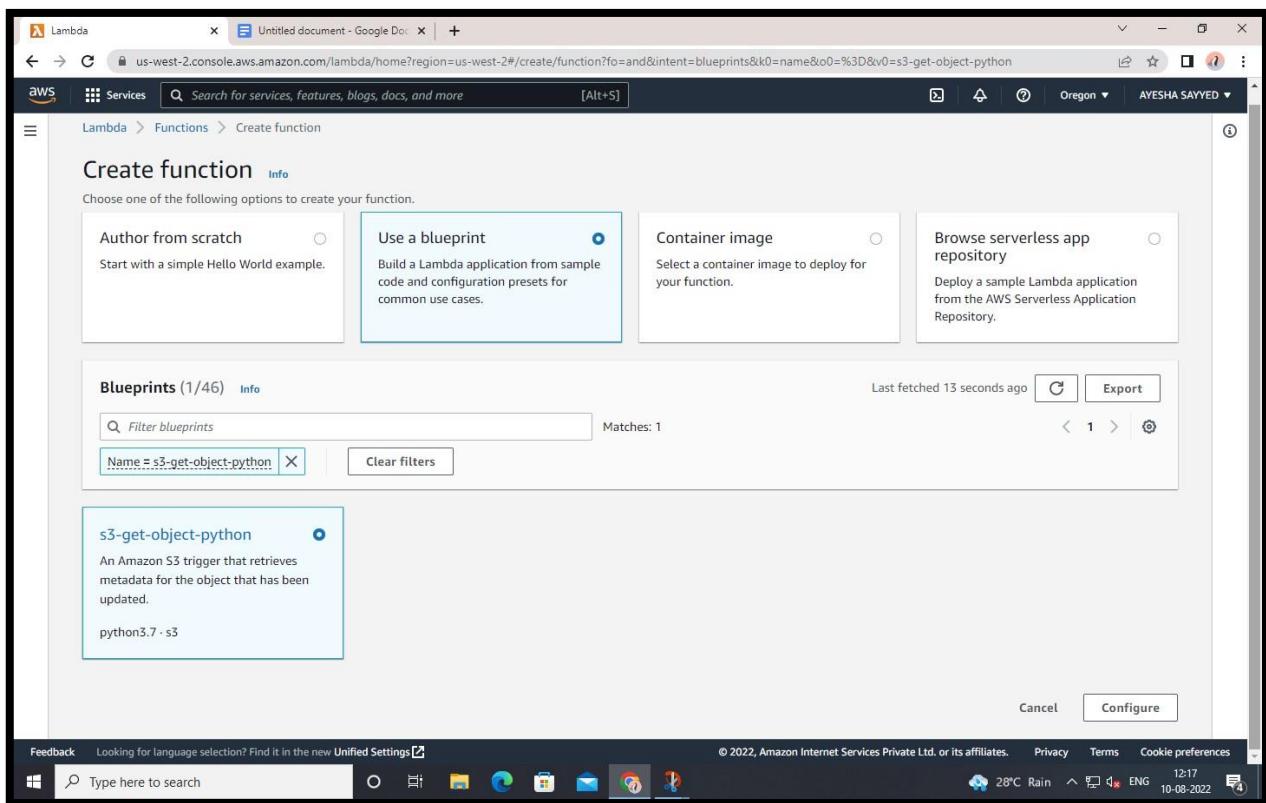
Step 5: Search for ‘Lambda’ and click it.



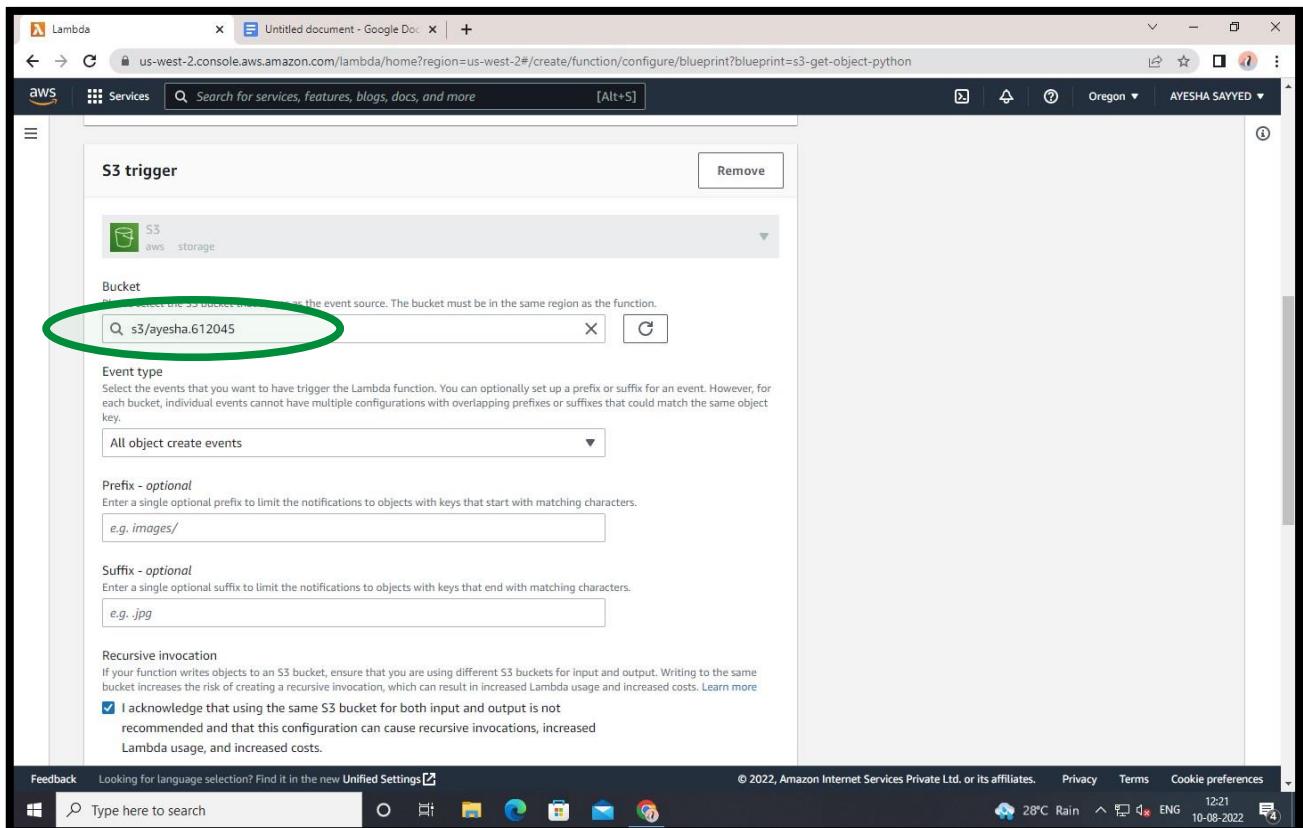
Step 6: Click on ‘Create Function’.



Step 7: Choose “Use a blueprint” → Search s3 in “Blueprints” → Select “s3-get-object-python”.



Step 8: Name the Function & Role → Select your S3 bucket as trigger → Click on Create.

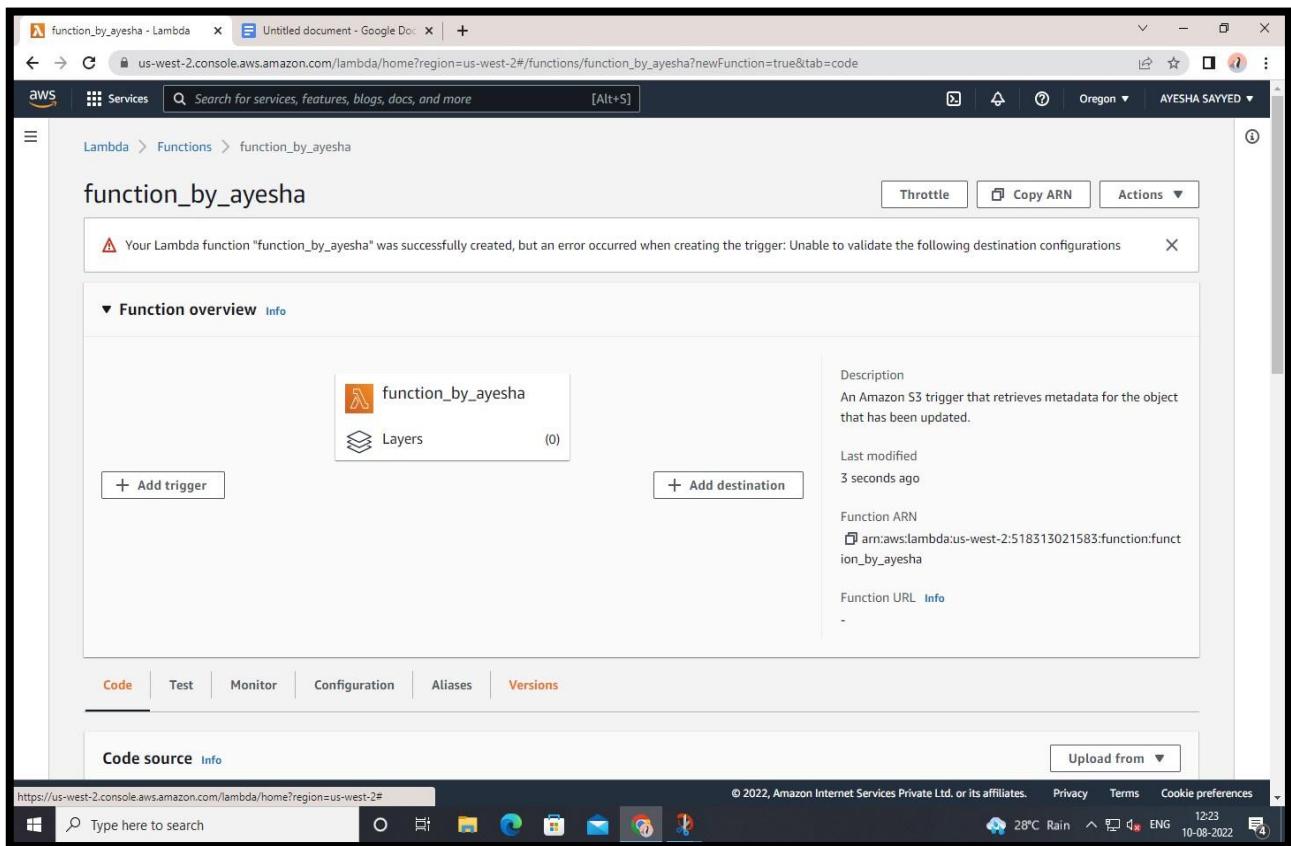


The screenshot shows the AWS Lambda code editor interface. The runtime is set to Python 3.7 and the architecture to x86_64. The code area contains a Python script for handling S3 events:

```
1 import json
2 import urllib.parse
3 import boto3
4
5 print('Loading function')
6
7 s3 = boto3.client('s3')
8
9
10 def lambda_handler(event, context):
11     #print("Received event: " + json.dumps(event, indent=2))
12
13     # Get the object from the event and show its content type
14     bucket = event['Records'][0]['s3']['bucket']['name']
15     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
16
17     response = s3.get_object(Bucket=bucket, Key=key)
18     print("CONTENT TYPE: " + response['ContentType'])
19     return response['ContentType']
20
21 except Exception as e:
22     print(e)
23     print('Error getting object {} from bucket {}. Make sure they exist and your bucket'
24         .format(key, bucket))
25     raise e
```

At the bottom right of the code editor is a red circle around the orange "Create function" button.

Step 9: Click on your function→Go to code→Add a print statement in code→Save the code and deploy it.



The screenshot shows the AWS Lambda console interface. In the top navigation bar, the URL is us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/function_by_ayesha?newFunction=true&tab=code. The main area displays the code for the function `function_by_ayesha`, specifically the file `lambda_function.py`. The code is as follows:

```
 1 #!/usr/bin/python
 2
 3 # This is a sample Python script for a Lambda function.
 4
 5 # Import required libraries
 6 import json
 7 import boto3
 8
 9
10 def lambda_handler(event, context):
11     #print("Received event: " + json.dumps(event, indent=2))
12
13     # Get the object from the event and show its content type
14     bucket = event['Records'][0]['s3']['bucket']['name']
15     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
16     try:
17         response = s3.get_object(Bucket=bucket, Key=key)
18         print("CONTENT TYPE: " + response['ContentType'])
19         print("IMAGE FILE IS BEEN UPLOADED BY AYESHA SAYYED_45")
20         return response['ContentType']
21     except Exception as e:
22         print(e)
23         print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))
24         raise e
```

A red circle highlights the line `print("IMAGE FILE IS BEEN UPLOADED BY AYESHA SAYYED_45")`. The status bar at the bottom right shows the time as 19:63 and the language as Python.

Code successfully uploaded / deployed.

The screenshot shows the AWS Lambda console interface after deployment. The top navigation bar URL is the same as the previous screenshot. The main area displays the same code for the function `function_by_ayesha`, specifically the file `lambda_function.py`. A green notification bar at the top states `Successfully updated the function function_by_ayesha.`. The rest of the interface is identical to the first screenshot, including the code editor and status bar.

Step 10: Go to your bucket → Upload an image as object in your bucket → Click on upload.

The screenshot shows the AWS S3 Management Console. On the left, there's a sidebar with 'Buckets' selected. The main area displays an 'Account snapshot' with a link to 'View Storage Lens dashboard'. Below it is a table titled 'Buckets (1) Info' with one item: 'ayesha.612045' located in 'US West (Oregon)' with 'us-west-2' as the region. This row is highlighted with a red oval. At the top of the table are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A search bar at the top says 'Find buckets by name'. The bottom of the page includes a feedback section, copyright information, and a system status bar.

The screenshot shows the AWS S3 Management Console with the path 'Amazon S3 > Buckets > ayesha.612045 > Upload'. The title is 'Upload Info'. It says 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more'.

The main area has a dashed box for dragging files and a table for 'Files and folders (1 Total, 54.8 KB)'. The table shows one item: 'f3b58f860a9e8487e2eb7e853e1f4f8e.jpg' with 'image/jpeg' type and '54.8 KB' size. Buttons for 'Remove', 'Add files', and 'Add folder' are at the top of the table. Below the table is a 'Destination' section with 'Destination' set to 's3://ayesha.612045'. Under 'Destination details', it says 'Bucket settings that impact new objects stored in the specified destination.'

At the bottom, there are sections for 'Permissions' and 'Properties'. The 'Properties' section has a note about specifying storage class, encryption settings, tags, and more. A large red box highlights the 'Upload' button at the bottom right.

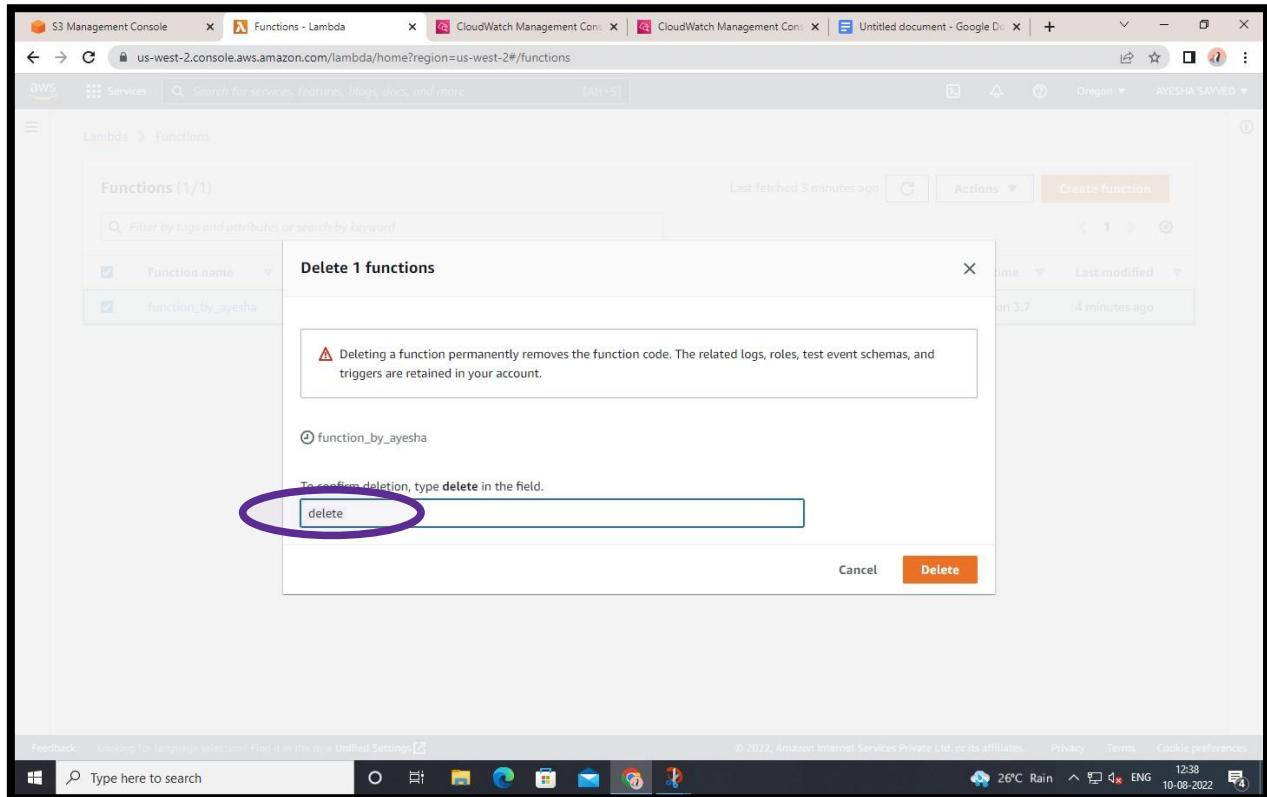
Now the ‘Image’ is successfully uploaded .

The screenshot shows the AWS S3 Management Console. A green banner at the top indicates 'Upload succeeded' with a link to 'View details below'. Below this, a summary table shows the destination 's3://ayesha.612045' with one succeeded file (54.8 KB) and zero failed files. Under 'Files and folders', a table lists one file: 'f3b58f860a9e8487e2eb7e853e1f4f8e.jpg' which is an image/jpeg file of size 54.8 KB and status 'Succeeded'. The browser's address bar shows the URL s3.console.aws.amazon.com/s3/upload/ayesha.612045?region=us-west-2.

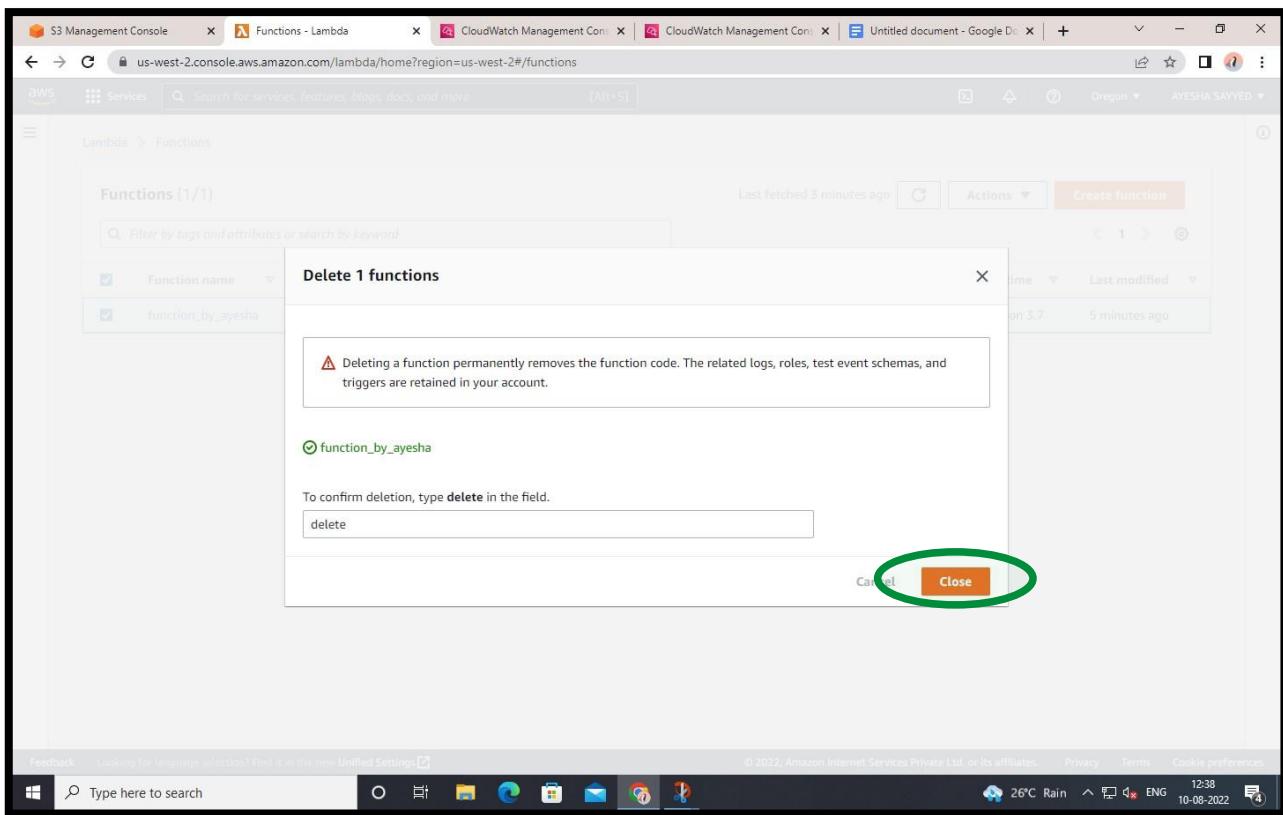
After the image is been uploaded you can check the message by Click the blue link in Log stream→ “Image file uploaded”.

Step 11: Now Delete your ‘Lambda Function’.

The screenshot shows the AWS Lambda console under the 'Functions' tab. It displays a single function named 'function_by_ayesha' with the description 'An Amazon S3 trigger that retrieves metadata for the object that has been updated.' The function is listed as 'Last fetched 2 minutes ago'. On the right, there are 'Actions' buttons for 'View details', 'Test', and 'Delete'. The 'Delete' button is highlighted. The browser's address bar shows the URL us-west-2.console.aws.amazon.com/lambda/home/?region=us-west-2#/functions.



Delete and close the function .



Step 12: Empty your Bucket→Delete the object and then delete the Bucket.

The screenshot shows the AWS S3 Management Console. In the 'Delete objects' dialog, under 'Specified objects', there are two files listed:

Name	Type	Last modified	Size
computer-laptop-notebook-pen.jpg	jpg	August 10, 2022, 12:35:58 (UTC+05:30)	69.9 KB
f3b58f860a9e8487e2eb7e853e1f4f8e.jpg	jpg	August 10, 2022, 12:30:24 (UTC+05:30)	54.8 KB

In the 'Permanently delete objects?' section, a text input field contains the text "permanently delete". A purple oval highlights this input field. A blue rectangle highlights the orange 'Delete objects' button.

The screenshot shows the 'Delete objects: status' page. At the top, a green banner displays the message "Successfully deleted objects" with a link "View details below." A red circle highlights this banner.

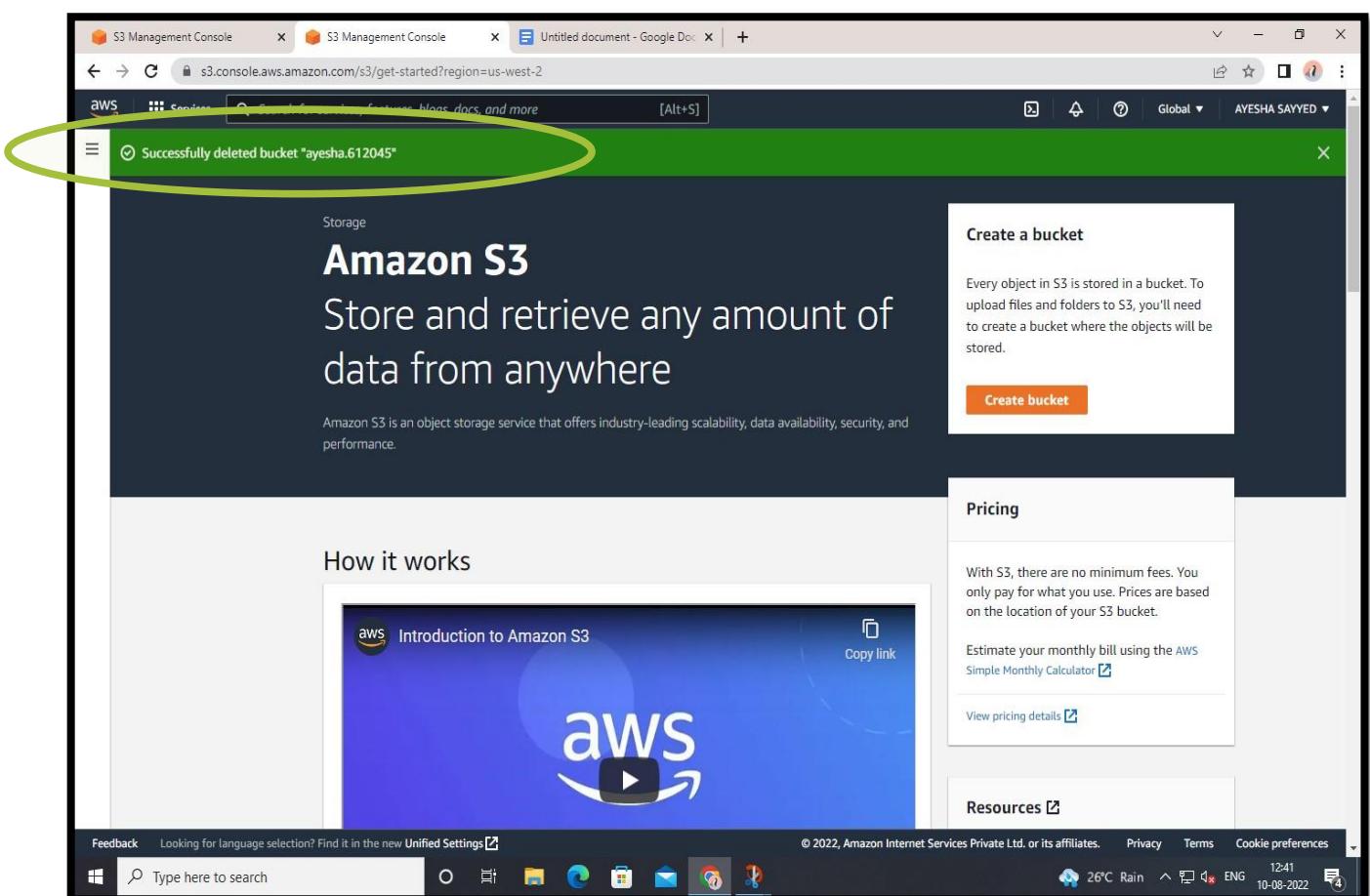
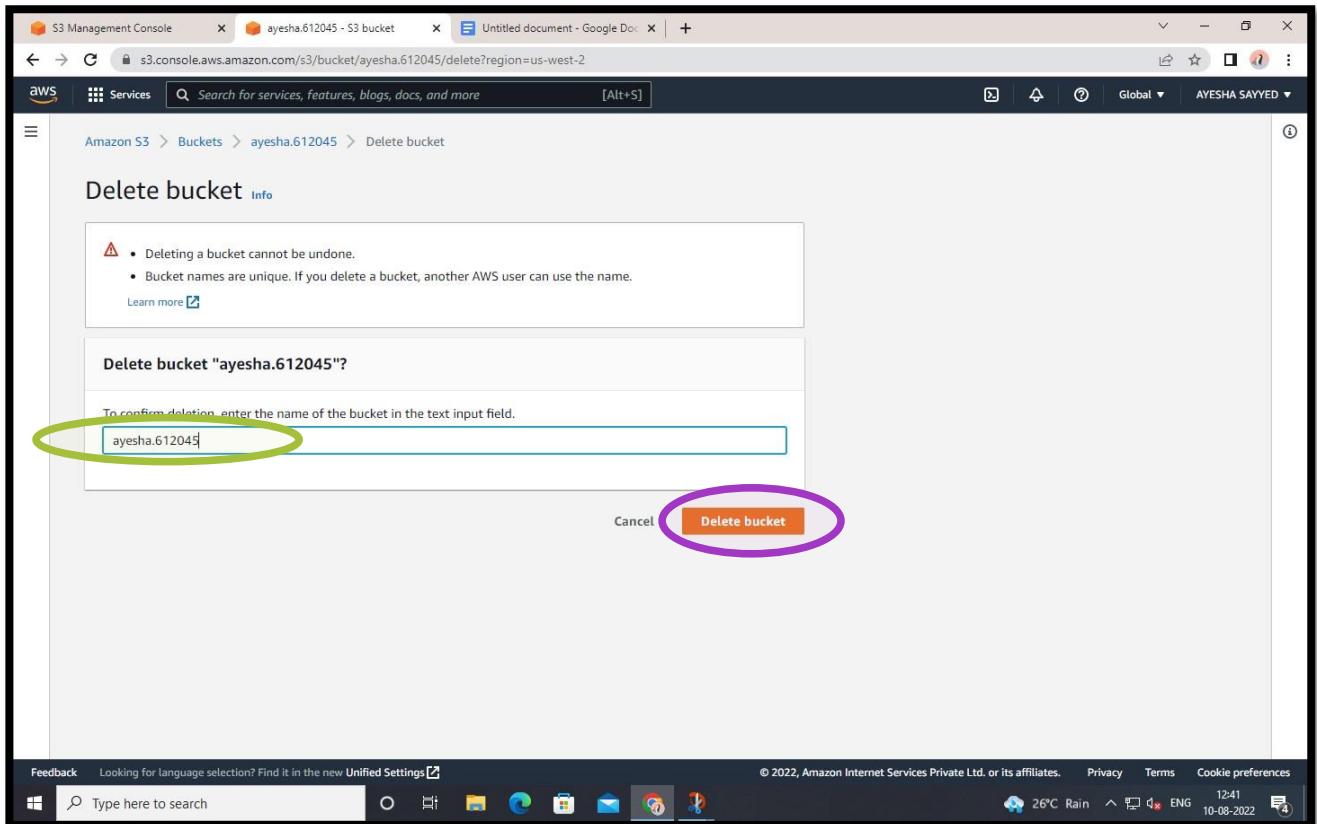
Summary

Source	Successfully deleted	Failed to delete
s3://ayesha.612045	2 objects, 124.7 KB	0 objects

Failed to delete (0)

No objects failed to delete.

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



EXPERIMENT – 07

Q1. What is Containerization / Docker? Explain Docker Architecture with the help of diagram.

Containerization is OS-based virtualization that creates multiple virtual units in the user space, known as Containers. Containers share the same host kernel but are isolated from each other through private namespaces and resource control mechanisms at the OS level.

Docker is the containerization platform that is used to package your application and all its dependencies together in the form of containers to make sure that your application works seamlessly in any environment which can be developed or tested or in production.

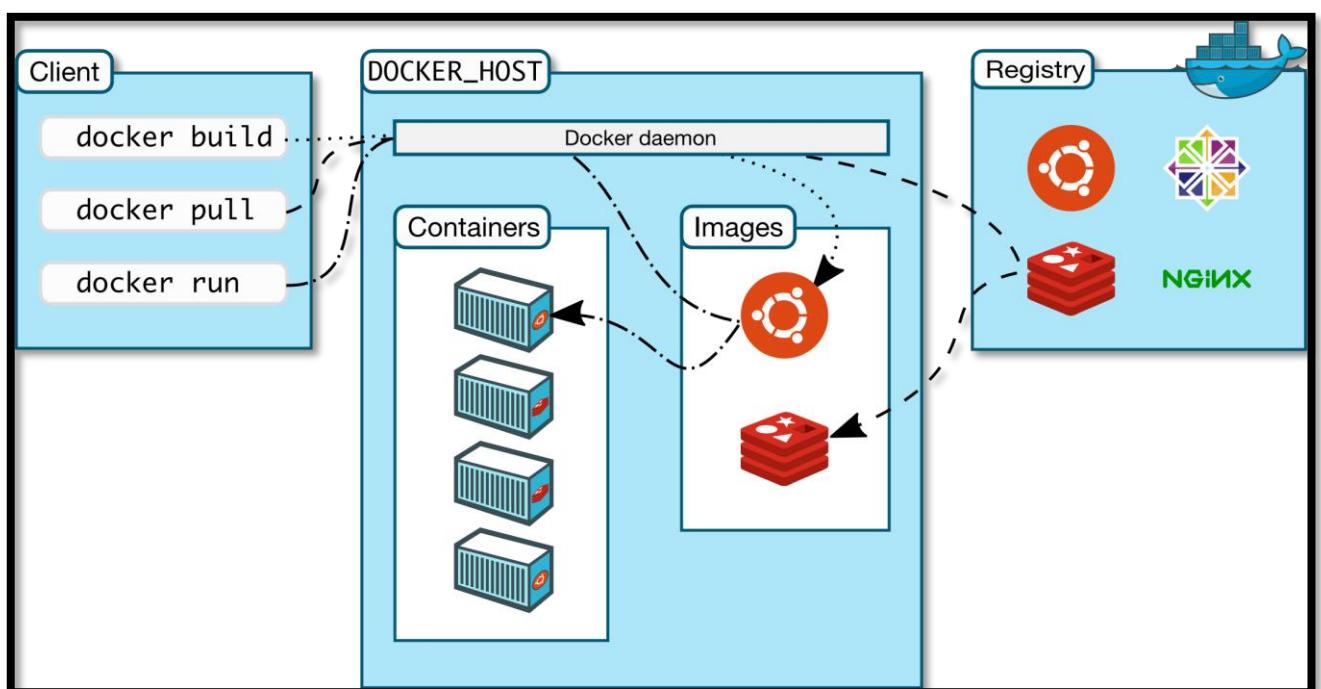
Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Docker daemon runs on the host operating system. It is responsible for running containers to manage docker services.

Docker daemon communicates with other daemons. It offers various Docker objects such as images, containers, networking, and storage.

Docker Architecture:

Docker follows Client-Server architecture, which includes the three main components that are Docker Client, Docker Host, and Docker Registry.



1. Docker Client

Docker client uses commands and REST APIs to communicate with the Docker Daemon (Server). When a client runs any docker command on the docker client terminal, the client terminal sends these docker commands to the Docker daemon. Docker daemon receives these commands from the docker client in the form of command and REST API's request.

2. Docker Host

Docker Host is used to provide an environment to execute and run applications. It contains the docker daemon, images, containers, networks, and storage.

3. Docker Registry

Docker Registry manages and stores the Docker images. There are two types of registries in the Docker –

Public Registry: Public Registry is also called as Docker hub.

Private Registry- It is used to share images within the enterprise.

Docker Objects

There are the following Docker Objects:

Docker Images

Docker images are the read-only binary templates used to create Docker Containers. It uses a private container registry to share container images within the enterprise and also uses public container registry to share container images within the whole world.

Docker Containers

Containers are the structural units of Docker, which is used to hold the entire package that is needed to run the application. The advantage of containers is that it requires very less resources.

Docker Networking

Using Docker Networking, an isolated package can be communicated. Docker contains the following network drivers Bridge, Host, None, Overlay, Macvlan.

Docker Storage

Docker Storage is used to store data on the container. Docker offers the following options for the Storage – Data Volume, Directory Mounts, Storage Plugins.

Docker Architecture:

Docker follows Client-Server architecture, which includes the three main components that are Docker Client, Docker Host, and Docker Registry.

Q2. Compare Containers vs VMs.

VMs	Containers
The hardware is virtualized to execute several Operating system instances with VMs.	Containers facilitate a way for virtualizing the operating system so that several workloads can execute on an individual operating system instance
VM is managed via hypervisor and uses VM hardware.	Containers give services of OS from an underlying host and also separate the applications utilizing virtual-memory hardware.
VM permits us for installing other software so virtually we control it as disputed to install the software on a computer directly.	The containers are software that permits distinct application's functionalities independently.
VM runs in minutes due to its large size.	Containers run in seconds.
It is highly secured.	It is less secure.
VM examples: VMware, Xen, KVM	Container examples: Containers via Docker, PhotonOS, RancherOS.

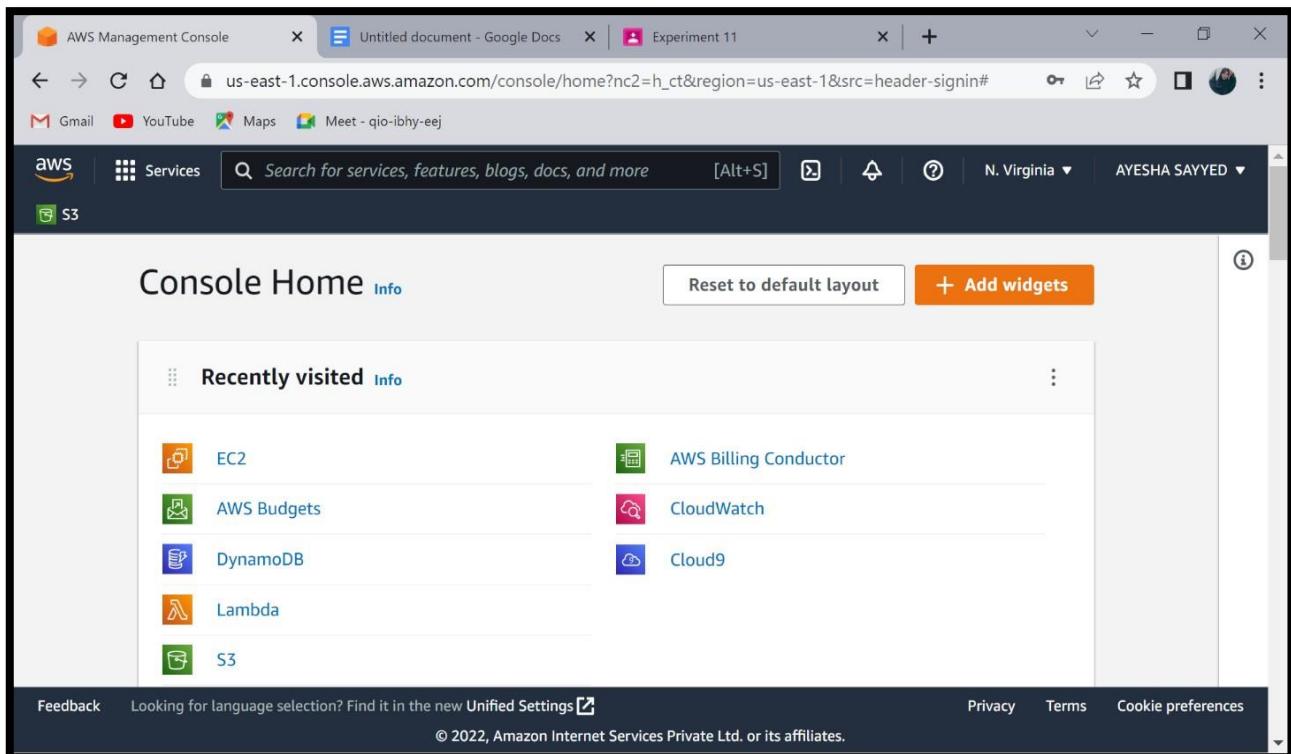
Q3. Why are Containers lightweight?

Inside a container are all the necessary executables, binary code, libraries, and configuration files. Compared to server or machine virtualization approaches, however, containers do not contain operating system images. This makes them more lightweight and portable, with significantly less overhead.

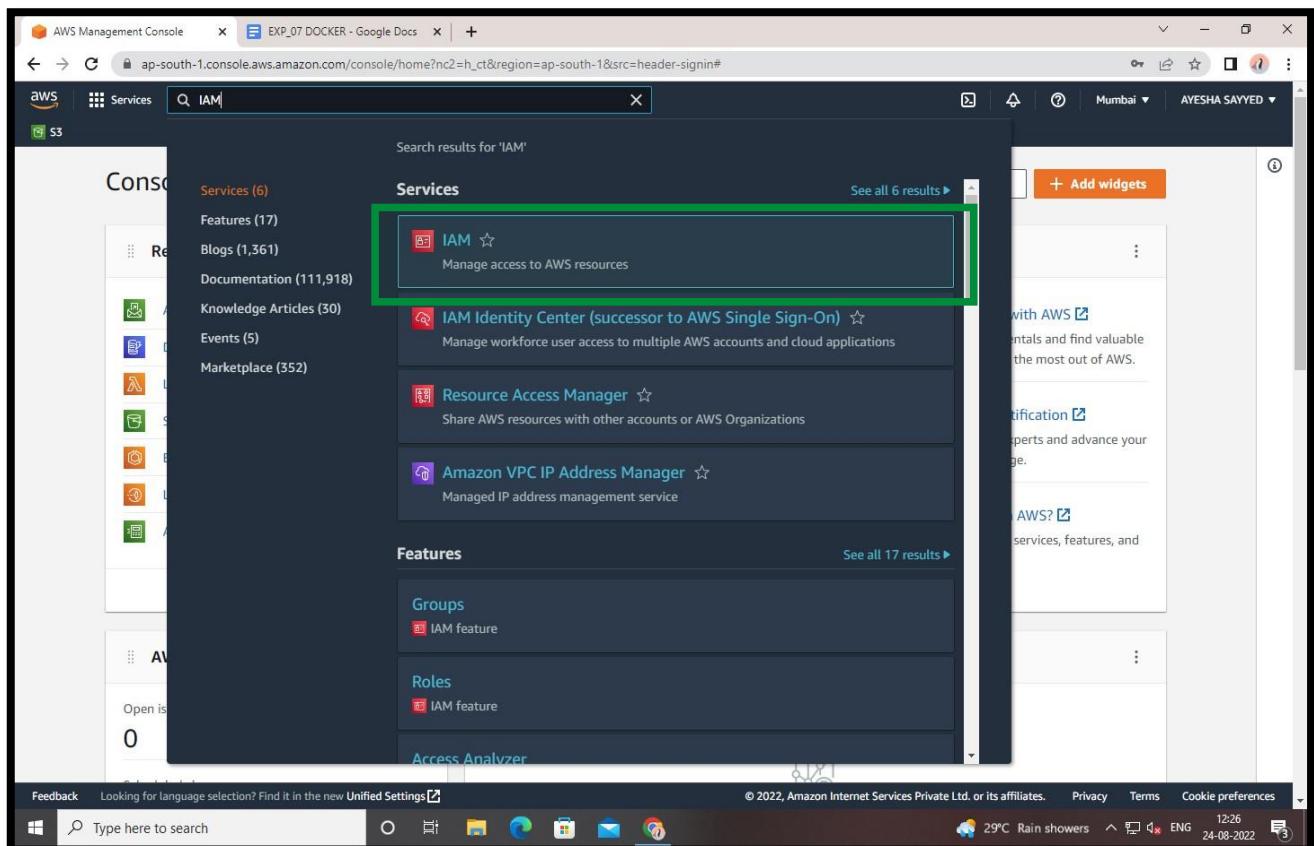
Containers are more lightweight than VMs, as their images are measured in megabytes rather than gigabytes. Containers require fewer IT resources to deploy, run, and manage. Containers spin up in milliseconds. Since their order of magnitude is smaller.

Q4. Deploy a containerized web Application on AWS EC2 Linux. [install Docker ,pull nginx image and run it].Pull python images and run the command to list all the locally stored docker images .

Step 1: AWS Management Console Dashboard.



Step 2: Search for IAM and select it.



Go to ‘Roles’ and click on ‘Create Role’.

The screenshot shows the AWS IAM Management Console. On the left, there's a navigation sidebar with 'Identity and Access Management (IAM)' selected. The main area shows a list of existing roles with columns for 'Role name', 'Trusted entities', and 'Last ac...'. At the top right of this list, there's a blue 'Create role' button with a red box around it. Below the list, there's a section titled 'Roles Anywhere' with icons for a computer monitor, a key, and a gear.

Step 4: Select Trusted Entity ‘AWS services’ use common cases as EC2 and click to the NEXT button.

This screenshot shows the 'Select trusted entity' step of the IAM Role creation wizard. On the left, there are three steps: 'Step 1 Select trusted entity', 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. The 'Step 1' section is expanded. It has a 'Trusted entity type' section with five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below that is a 'Use case' section with 'Common use cases' and two options: 'EC2' (selected) and 'Lambda'. There's also a dropdown for 'Use cases for other AWS services' with the placeholder 'Choose a service to view use case'. At the bottom right of the step, there are 'Cancel' and 'Next' buttons, with the 'Next' button having a green box around it.

Step 5: Now in Permissions Policies → select Amazon SSM Manage.

Container examples: Containers via Docker, PhotonOS, RancherOS.

Permissions policies (Selected 1/764)

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter

14 matches

Clear filters

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonEC2Rolefor...	AWS m...	This policy will soon be deprecated. Please use AmazonSSMManagedInstanceCore p...
<input type="checkbox"/> AmazonSSMAutom...	AWS m...	Provides access to view automation executions and send approval decisions to autom...
<input type="checkbox"/> AmazonSSMMag...	AWS m...	The policy for Amazon EC2 Role to enable AWS Systems Manager service core functi...
<input type="checkbox"/> AmazonSSMDirect...	AWS m...	This policy allows SSM Agent to access Directory Service on behalf of the customer fo...
<input type="checkbox"/> AmazonSSMFullAc...	AWS m...	Provides full access to Amazon SSM.
<input type="checkbox"/> AmazonSSMAutom...	AWS m...	Provides permissions for EC2 Automation service to execute activities defined within A...
<input type="checkbox"/> AmazonSSMRead...	AWS m...	Provides read only access to Amazon SSM.
<input type="checkbox"/> AmazonSSMMainte...	AWS m...	Service Role to be used for EC2 Maintenance Window
<input type="checkbox"/> AWSBudgetsAction...	AWS m...	This policy gives permissions to control AWS resources. For example, to start and sto...
<input type="checkbox"/> AWSResourceAcce...	AWS m...	Provides read only access to AWS Resource Access Manager.
<input type="checkbox"/> AmazonSSMPatch...	AWS m...	Provide access to child instances for patch association operation.
<input type="checkbox"/> AWSResourceAcce...	AWS m...	Provides full access to AWS Resource Access Manager
<input type="checkbox"/> AWSCloud9SSMInsi...	AWS m...	This policy will be used to attach a role on a InstanceProfile which will allow Cloud9 to ...
<input type="checkbox"/> AWSResourceAcce...	AWS m...	Provides access to AWS Resource Access Manager APIs needed by a resource share...

▶ Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous Next

Step 6: Now give name to your Role → then click on “Create Role”.

Role name
Enter a meaningful name to identify this role.
docker.Role45

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.

Step 1: Select trusted entities

```
1+ [ {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Action": [  
7                 "sts:AssumeRole"  
8             ],  
9         }  
10    ]  
11 }  
12 ]  
13 }  
14 ]  
15 ]  
16 }]
```

Step 2: Add permissions

Policy name	Type	Attached as
AmazonEC2RoleforSSM	AWS managed	Permissions policy

Tags

Add tags (Optional)
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag
You can add up to 50 more tags.

Create role

Role Created Successfully!!

The screenshot shows the AWS IAM Management Console. In the top navigation bar, there is a blue banner with the text "New! Securely access AWS services from your data center with IAM Roles Anywhere. Learn more". Below this, a green notification bar displays the message "Role docker.Role45 created" with a checkmark icon. The main content area shows a table titled "Roles (6) Info" with a search bar and a "Create role" button. On the left sidebar, under "Access management", the "Roles" option is selected. A red oval highlights the green notification bar.

Step 7: Now go to → Services and search →EC2.

The screenshot shows the AWS EC2 Management Console. The left sidebar lists various AWS services under "Recently visited" and "Favorites". A teal box highlights the "EC2" entry, which is described as "Virtual Servers in the Cloud". The main content area displays a "Recently visited" list with items like "Console Home", "AWS Budgets", "DynamoDB", "Lambda", "S3", "Elastic Container Service", "Lightsail", and "AWS Billing Conductor". To the right, there are two panels: "Account attributes" and "Additional information", both containing links to various AWS documentation and resources. The status bar at the bottom shows the URL "https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Home" and the date "24-08-2022".

Step 8: Click on “Launch Instances” in ‘EC2 Dashboard’.

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Instances' section, the 'Launch Templates' option is selected. In the main content area, there is a 'Launch instance' section with a large orange 'Launch instance' button. This button is highlighted with a green rectangular selection tool. Below the button, there is a note: 'Note: Your instances will launch in the US East (N. Virginia) Region'. To the right of this section is a 'Service health' panel showing 'This service is operating normally'. On the far right, there is an 'Explore AWS' panel with various promotional links. The top navigation bar shows the URL as 'us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Home:'.

Now give name to Your Instance.

The screenshot shows the 'Launch an instance' wizard. The first step, 'Name and tags', is displayed. A purple rectangular selection tool highlights the 'Name' input field, which contains the text 'Ayesha_instance_45'. To the right of the input field is a 'Add additional tags' link. Below this step is a search bar with the placeholder 'Search our full catalog including 1000s of application and OS images'. At the bottom of the page, there is a 'Quick Start' link. On the right side of the screen, there is a 'Summary' panel containing configuration details: 'Number of instances' (1), 'Software Image (AMI)' (Amazon Linux 2 Kernel 5.10 AMI...), 'Virtual server type (instance type)' (t2.micro), 'Firewall (security group)' (New security group), and 'Storage (volumes)' (1 volume(s) - 8 GiB). A blue info box at the bottom right states 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the...'. At the very bottom of the page is a 'Feedback' link and a standard Windows taskbar.

Select Amazon Linux.

The screenshot shows the AWS Lambda console with the following details:

- Region:** us-east-1
- Function Name:** EXP_07_DOCKER
- Runtime:** Node.js 14.x
- Description:** A simple Lambda function to demonstrate Docker usage.
- Code Type:** Docker container
- Docker Container:** Dockerfile
- Container Image:** 12.0.0
- Environment:** N/A
- Memory:** 128 MB
- Timeout:** 3 seconds
- Triggers:** No triggers defined.
- Logs:** View logs for EXP_07_DOCKER function.

Click to all ‘Checkbox’
Step 9: Create a Key Pair.

The screenshot shows the AWS Lambda console with the following details:

- Region:** us-east-1
- Function Name:** EXP_07_DOCKER
- Description:** A simple Lambda function to demonstrate Docker usage.
- Code Type:** Docker container
- Docker Container:** Dockerfile
- Container Image:** 12.0.0
- Environment:** N/A
- Memory:** 128 MB
- Timeout:** 3 seconds
- Triggers:** No triggers defined.
- Logs:** View logs for EXP_07_DOCKER function.

The screenshot shows the AWS EC2 Launch Instance wizard. On the left, under 'Network settings', it lists 'vpc-0cf924feb1b794701' and 'Subnet Info' (No preference). It includes sections for 'Auto-assign public IP' (Enable) and 'Firewall (security groups)' (Create security group or Select existing security group). A note says we'll create a new security group called 'launch-wizard-1'. Under 'Security group rules', three checkboxes are checked: 'Allow SSH traffic from Anywhere (0.0.0.0/0)', 'Allow HTTP traffic from the internet (To set up an endpoint, for example when creating a web server)', and 'Allow HTTPS traffic from the internet (To set up an endpoint, for example when creating a web server)'. A warning message at the bottom states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' On the right, the 'Summary' section shows 1 instance, AMI 'Amazon Linux 2 Kernel 5.10 AMI...', instance type 't2.micro', and storage '1 volume(s) - 8 GiB'. A tooltip for the 'Free tier' indicates it covers 750 hours of t2.micro usage in N. Virginia. At the bottom are 'Cancel' and 'Launch instance' buttons.

Instance Created Successfully! → Now View all Instances.

The screenshot shows the AWS EC2 Instances page. A blue banner at the top informs the user they've opted into the new launch experience, with links to find out more or opt out. Below, a success message is displayed: 'Success' with a green checkmark, followed by 'Successfully initiated launch of instance (i-0f90b0e58431086e3)'. A pink oval highlights this message. A 'Launch log' link is below it. A 'Next Steps' section follows, containing 'Get notified of estimated charges' (Create billing alerts), 'How to connect to your instance' (Your instance is launching and might be a few minutes until it's running), and a 'View more resources' link. At the bottom right is a large orange 'View all instances' button. The bottom of the screen shows the Windows taskbar with the search bar containing 'Type here to search'.

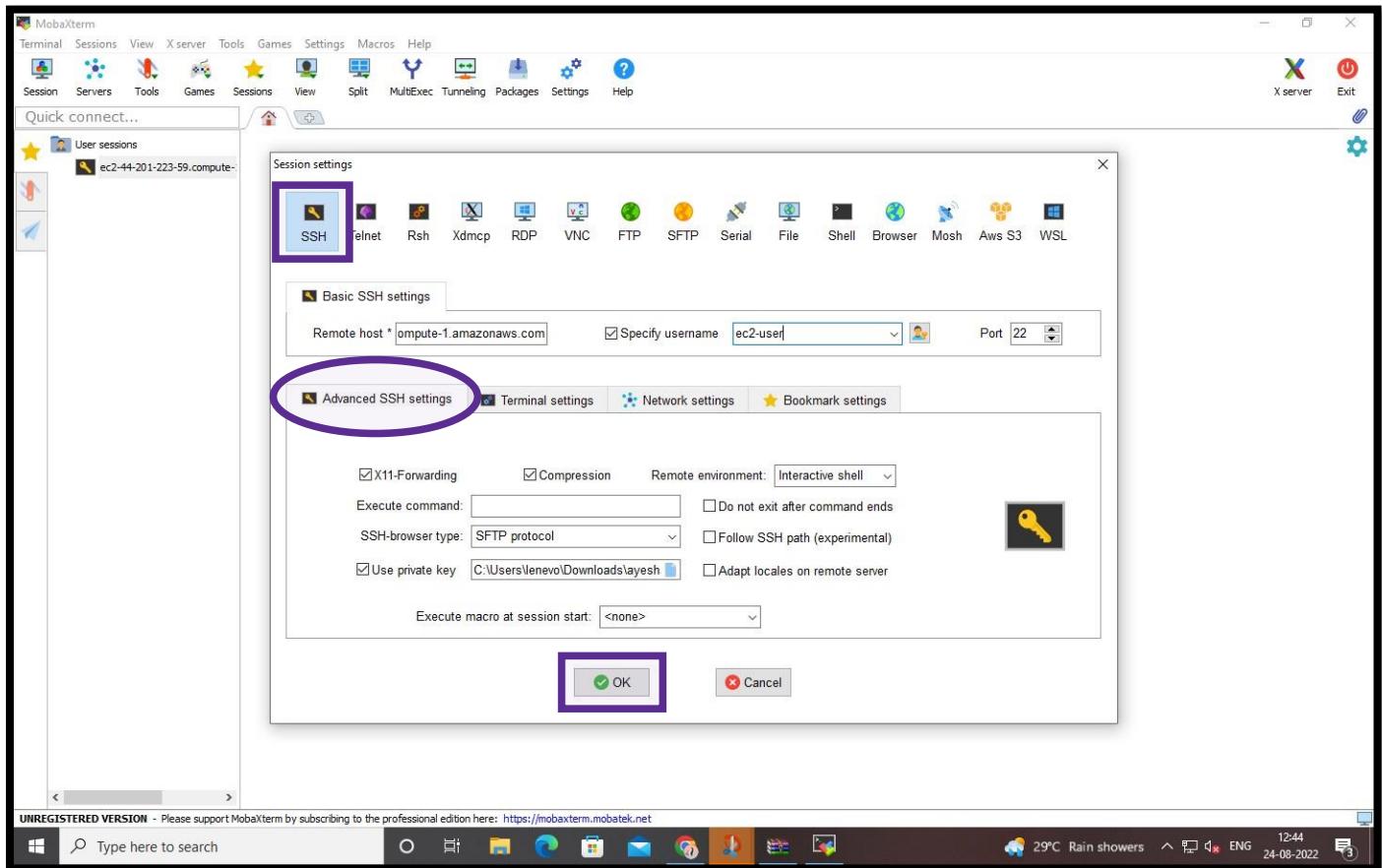
Click on Instance ID

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with 'Instances' selected. The main area displays a table titled 'Instances (1/1) Info' with one row. The row contains columns for Name (Ayesha_instance_45), Instance ID (i-0f90b0e58431086e3), Instance state (Running), Instance type (t2.micro), Status check (Initializing), Alarm status (No alarms), and Availability Zone (us-east-1c). A green box highlights the 'Instance ID' column. Below the table, a detailed view for the instance is shown with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the 'Details' tab, the Public IPv4 address is listed as 44.201.223.59. The status bar at the bottom right shows the date as 24-08-2022.

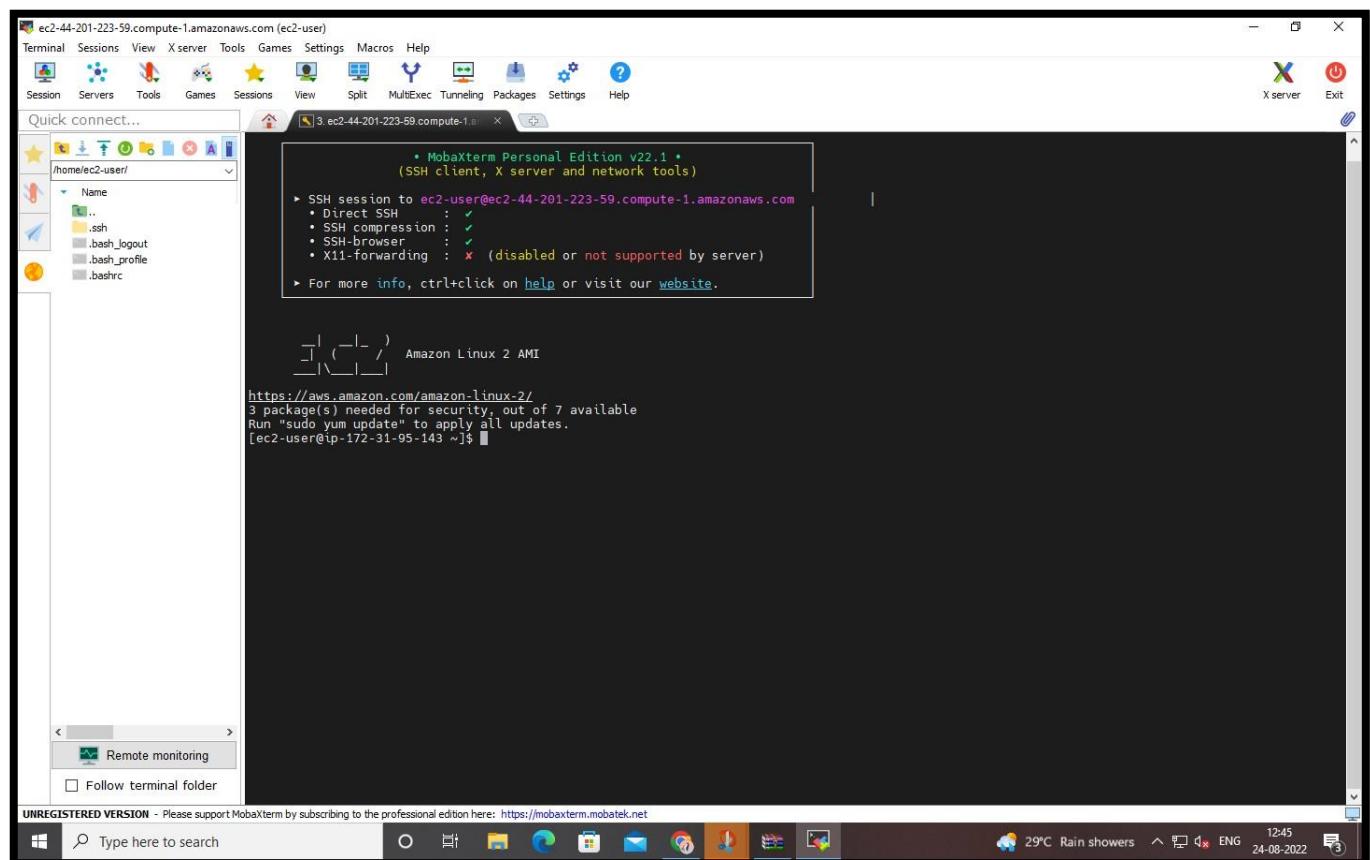
Go to Connect Instance → SSH Client → copy Public DNS

The screenshot shows the 'Connect to instance' dialog box. At the top, it says 'Connect to instance Info' and 'Connect to your instance i-0f90b0e58431086e3 (Ayesha_instance_45) using any of these options'. Below this, there are tabs for 'EC2 Instance Connect', 'Session Manager', 'SSH client' (which is selected), and 'EC2 serial console'. The 'SSH client' tab has a note: 'Public DNS copied' with a circled checkmark. It also lists the public DNS as 'ec2-44-201-223-59.compute-1.amazonaws.com'. A note at the bottom states: 'Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' The status bar at the bottom right shows the date as 24-08-2022.

**Step 10: Go to “MOBAXTERM” → Session → SSH → Advance SSH
Settings fill all details then ‘click OK’**



Connected to TERMINAL.



Step 11: Click to “sudo su” then → “yum install docker”.

The screenshot shows a MobaXterm window titled "ec2-44-201-223-59.compute-1.amazonaws.com (ec2-user)". The terminal session displays the command "sudo su" being run, followed by the output of the "yum install docker" command. The output shows the installation of Docker and its dependencies. A pink oval highlights the command "sudo su".

```
https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-143 ~]$ sudo su
[root@ip-172-31-95-143 ~]# yum install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.17-1.amzn2 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.17-1.amzn2.x86_64
--> Processing Dependency: libcgroup >= 0.40.rcl5.15 for package: docker-20.10.17-1.amzn2.x86_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.17-1.amzn2.x86_64
--> Processing Dependency: pigz for package: docker-20.10.17-1.amzn2.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.6.6-1.amzn2 will be installed
--> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.1.3-1.amzn2 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
=====
Package           Arch      Version            Repository          Size
=====
Installing:
  docker           x86_64   20.10.17-1.amzn2    amzn2extra-docker  39 M
Installing for dependencies:
  containerd       x86_64   1.6.6-1.amzn2        amzn2extra-docker  27 M
  libcgroup        x86_64   0.41-21.amzn2       amzn2-core          66 k
  pigz             x86_64   2.3.4-1.amzn2.0.1   amzn2-core          81 k
  runc             x86_64   1.1.3-1.amzn2       amzn2extra-docker  2.9 M
Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 69 M
Installed size: 260 M
Is this ok [y/d/N]:
```

Installation Completed

The screenshot shows the same MobaXterm window. The terminal session now displays the completed output of the "yum install docker" command, showing the successful installation of Docker and its dependencies. The status bar at the bottom indicates "29°C Rain showers" and the date "24-08-2022".

```
Installing:
  docker           x86_64   20.10.17-1.amzn2    amzn2extra-docker  39 M
Installing for dependencies:
  containerd       x86_64   1.6.6-1.amzn2        amzn2extra-docker  27 M
  libcgroup        x86_64   0.41-21.amzn2       amzn2-core          66 k
  pigz             x86_64   2.3.4-1.amzn2.0.1   amzn2-core          81 k
  runc             x86_64   1.1.3-1.amzn2       amzn2extra-docker  2.9 M
Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 69 M
Installed size: 260 M
Is this ok [y/d/N]: y
Downloading packages:
(1/5): libcgroup-0.41-21.amzn2.x86_64.rpm          | 66 kB  00:00:00
(2/5): pigz-2.3.4-1.amzn2.0.1.x86_64.rpm          | 81 kB  00:00:00
(3/5): containerd-1.6.6-1.amzn2.x86_64.rpm        | 27 MB  00:00:00
(4/5): docker-20.10.17-1.amzn2.x86_64.rpm         | 39 MB  00:00:00
(5/5): runc-1.1.3-1.amzn2.x86_64.rpm              | 2.9 MB  00:00:00
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : runc-1.1.3-1.amzn2.x86_64          1/5
  Installing : containerd-1.6.6-1.amzn2.x86_64     2/5
  Installing : libcgroup-0.41-21.amzn2.x86_64      3/5
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64       4/5
  Installing : docker-20.10.17-1.amzn2.x86_64      5/5
  Verifying  : runc-1.1.3-1.amzn2.x86_64          1/5
  Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64       2/5
  Verifying  : containerd-1.6.6-1.amzn2.x86_64      3/5
  Verifying  : libcgroup-0.41-21.amzn2.x86_64       4/5
  Verifying  : docker-20.10.17-1.amzn2.x86_64      5/5
Installed:
  docker.x86_64 0:20.10.17-1.amzn2
Dependency Installed:
  containerd.x86_64 0:1.6.6-1.amzn2    libcgroup.x86_64 0:0.41-21.amzn2    pigz.x86_64 0:2.3.4-1.amzn2.0.1    runc.x86_64 0:1.1.3-1.amzn2
Complete!
[root@ip-172-31-95-143 ~]#
```

Step 12: Run command → “service docker start” then → “docker pull nginx”

```
[root@ip-172-31-95-143 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-95-143 ec2-user]# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
7a6db449b51b: Pull complete
ca1981974b58: Pull complete
d4019c921e20: Pull complete
7cb804d746d4: Pull complete
e7a561828262: Pull complete
7247f6e5c182: Pull complete
Digest: sha256:b95a99feebf7797479e0c5eb5ec0bdfa5d9f504bc94da550c2f58e839ea6914f
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[root@ip-172-31-95-143 ec2-user]#
```

Command “docker images” then → “docker run -p 80:80 nginx”

```
[root@ip-172-31-95-143 ec2-user]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 2b7d6430f78d 27 hours ago 142MB
[root@ip-172-31-95-143 ec2-user]# docker run -p 80:80 nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: Info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/08/24 07:23:07 [notice] 1#1: using the 'epoll' event method
2022/08/24 07:23:07 [notice] 1#1: nginx/1.23.3
2022/08/24 07:23:07 [notice] 1#1: built by gcc 10.2.1 20220110 (Debian 10.2.1-6)
2022/08/24 07:23:07 [notice] 1#1: OS: Linux 5.10.180-118.517.amzn2.x86_64
2022/08/24 07:23:07 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 32768:65536
2022/08/24 07:23:07 [notice] 1#1: start worker processes
2022/08/24 07:23:07 [notice] 1#1: start worker process 31
[root@ip-172-31-95-143 ec2-user]#
```

Step 13: Go to EC2 again.

The screenshot shows the AWS Management Console with the 'Services' tab selected. A purple box highlights the 'Recently visited' sidebar on the left, which lists 'EC2' under 'Virtual Servers in the Cloud'. To the right, the 'Account attributes' panel is open, also with a purple box around it. The panel contains sections for 'Supported platforms', 'Default VPC', 'Settings', 'EBS encryption', 'Zones', 'EC2 Serial Console', 'Default credit specification', and 'Console experiments'. At the bottom of the page, the URL is https://console.aws.amazon.com/ec2/v2/home?region=us-east-1, and the status bar shows the date and time as 24-08-2022 12:33.

Step 14: Click on Instance ID and → copy Public IPv4 DNS.

The screenshot shows the 'Instances' page in the AWS Management Console. A purple box highlights the 'Instance ID' column in the table, specifically the entry for the running instance 'Ayesha_instan...'. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The instance is listed as 'Running' on an 't2.micro' instance type in the 'us-east-1c' availability zone. The status check shows 2/2 checks passed with no alarms. The table has a header row and several data rows, with the first data row being highlighted by the purple box.

Screenshot of the AWS EC2 Instance Details page for instance i-0f90b0e58431086e3 (Ayesha_instance_45). The Public IPv4 address is highlighted with a green box.

Instance summary for i-0f90b0e58431086e3 (Ayesha_instance_45)

Public IPv4 address: 44.201.223.59 | open address

Private IPv4 addresses: ec2-44-201-223-59.compute-1.amazonaws.com | open address

Private IP DNS name (IPv4 only): ip-172-31-95-143.ec2.internal

Instance type: t2.micro

VPC ID: vpc-0cf924feb1b794701

Subnet ID: subnet-080dd3cb634f6813d

Elastic IP addresses: -

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.

Auto Scaling Group name: -

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Step 15: Go to Google and Paste the Link

Screenshot of a web browser showing the Google homepage. The URL bar contains the Public IP address: ec2-44-201-223-59.compute-1.amazonaws.com. The search bar also shows this URL.

Google

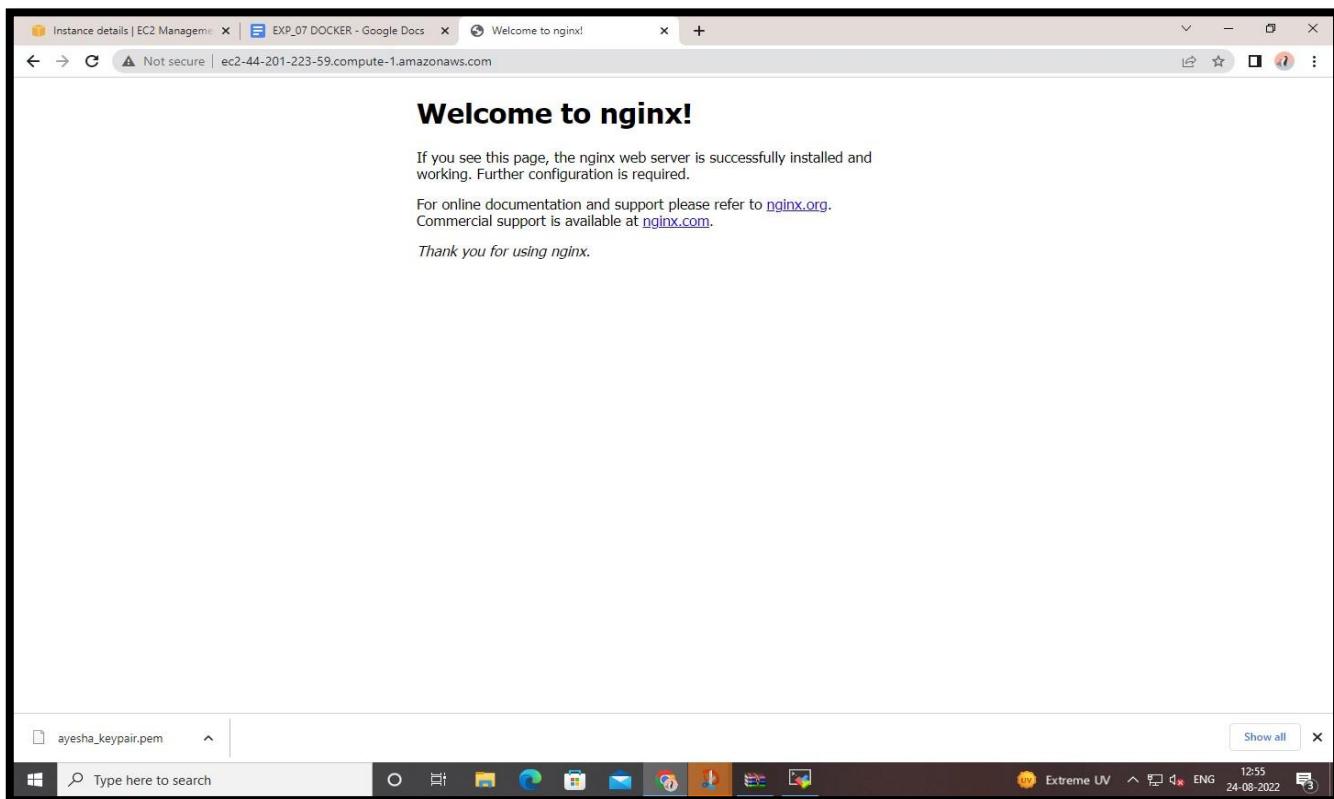
Search Google or type a URL

Sign in | YouTube | phpMyAdmin... | Welcome to... | Web Store | Add shortcut

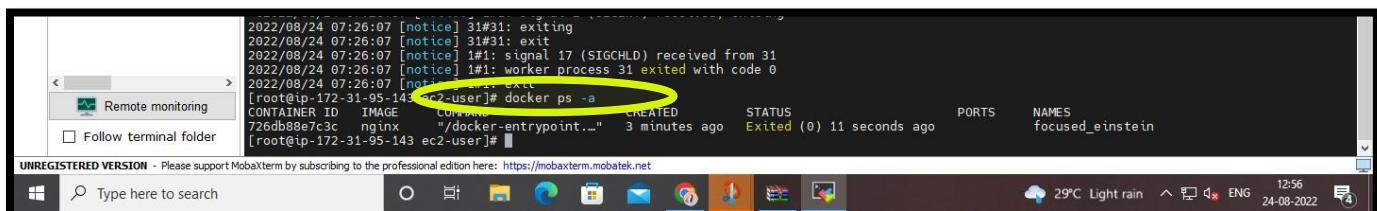
ec2-44-201-223-59.compute-1.amazonaws.com - Google Search

ayesha_keypair.pem

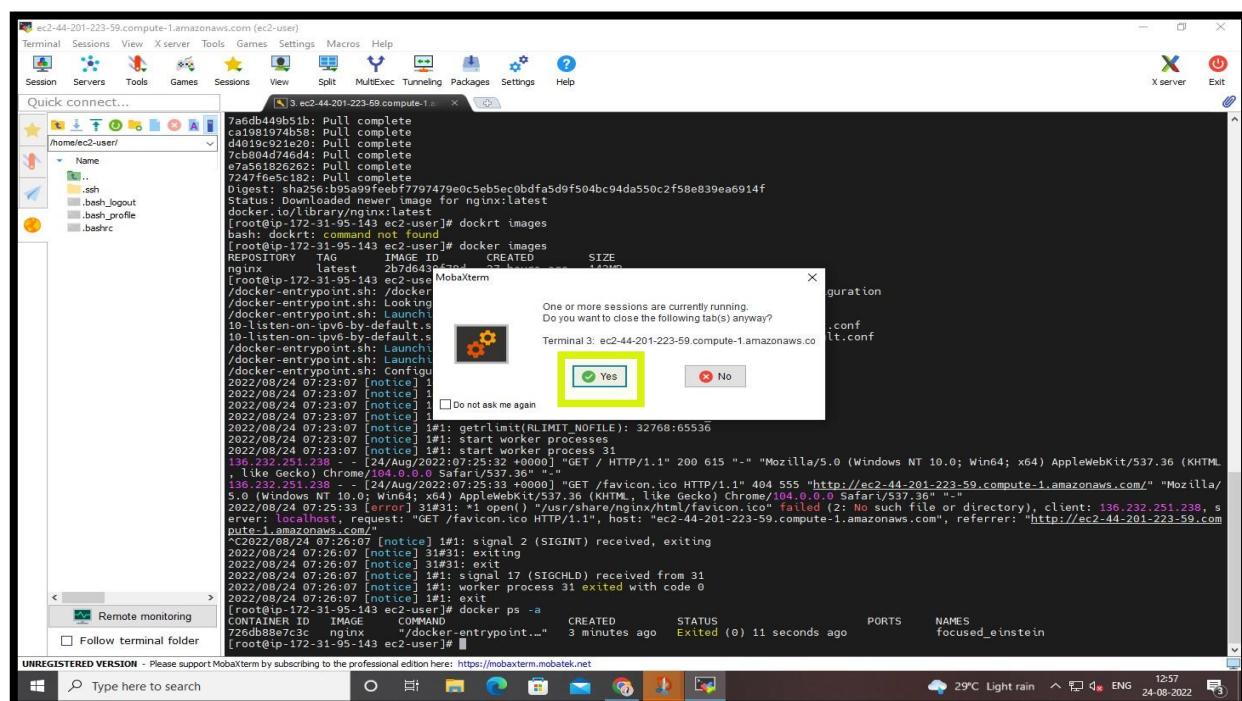
Welcome to nginx!



Run command → “docker ps -a”



Now close the tab



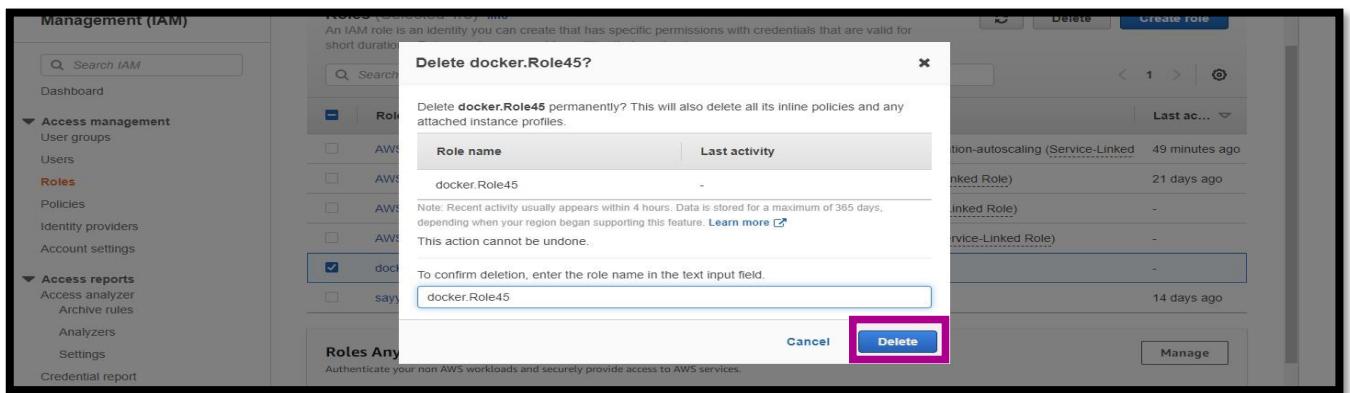
Step 16: Terminate Instances.

The screenshot shows the AWS EC2 Management Console. In the left sidebar, under the 'Instances' section, the 'Instances' link is selected. On the main page, there is a table titled 'Instances (1/1) Info' showing one instance named 'Ayesha_instance...' with the ID 'i-0f90b0e58431086e3'. The instance is currently 'Running'. In the 'Actions' column, there is a button labeled 'Terminate instance' which is circled in red. Below this, a modal dialog box titled 'Terminate instance?' is displayed. It contains a warning message: 'On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.' It also asks 'Are you sure you want to terminate these instances?' and lists the instance ID 'i-0f90b0e58431086e3 (Ayesha_instance_45)'. At the bottom of the dialog, there is a red-bordered 'Terminate' button.

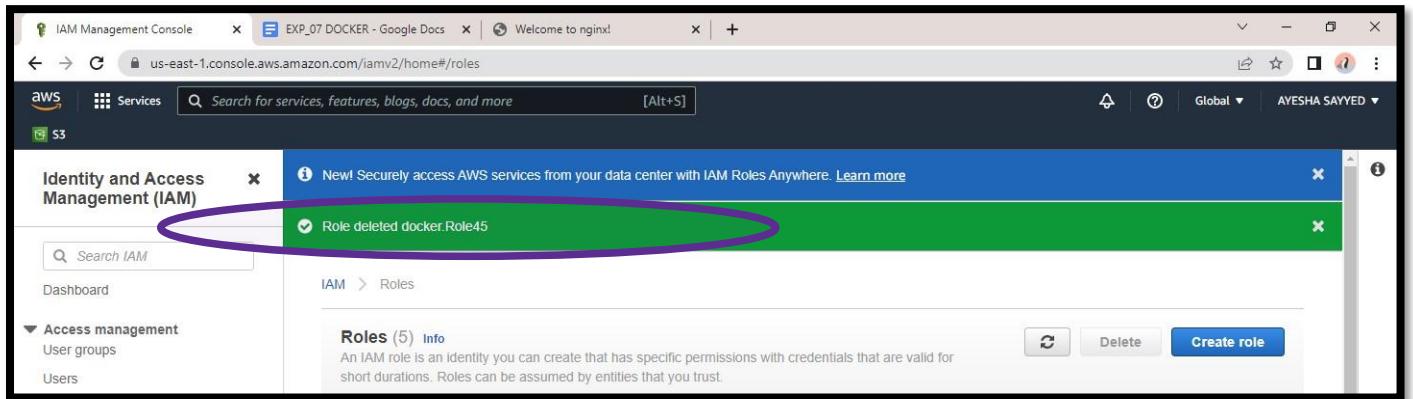
Successfully Terminated!!

The screenshot shows the AWS EC2 Management Console after the instance has been terminated. A green banner at the top of the page displays the message 'Successfully terminated i-0f90b0e58431086e3'. This message is circled in pink. The main table now shows the same instance row, but its status is listed as 'Shutting-down' with a circular icon. The rest of the table columns remain the same: Name (Ayesha_instance...), Instance ID (i-0f90b0e58431086e3), Instance type (t2.micro), Status check (2/2 checks passed), Alarm status (No alarms), and Availability Zone (us-east-1c).

Step 17: Now Delete the Role in IAM .



Role Deleted Successfully!



EXPERIMENT – 08

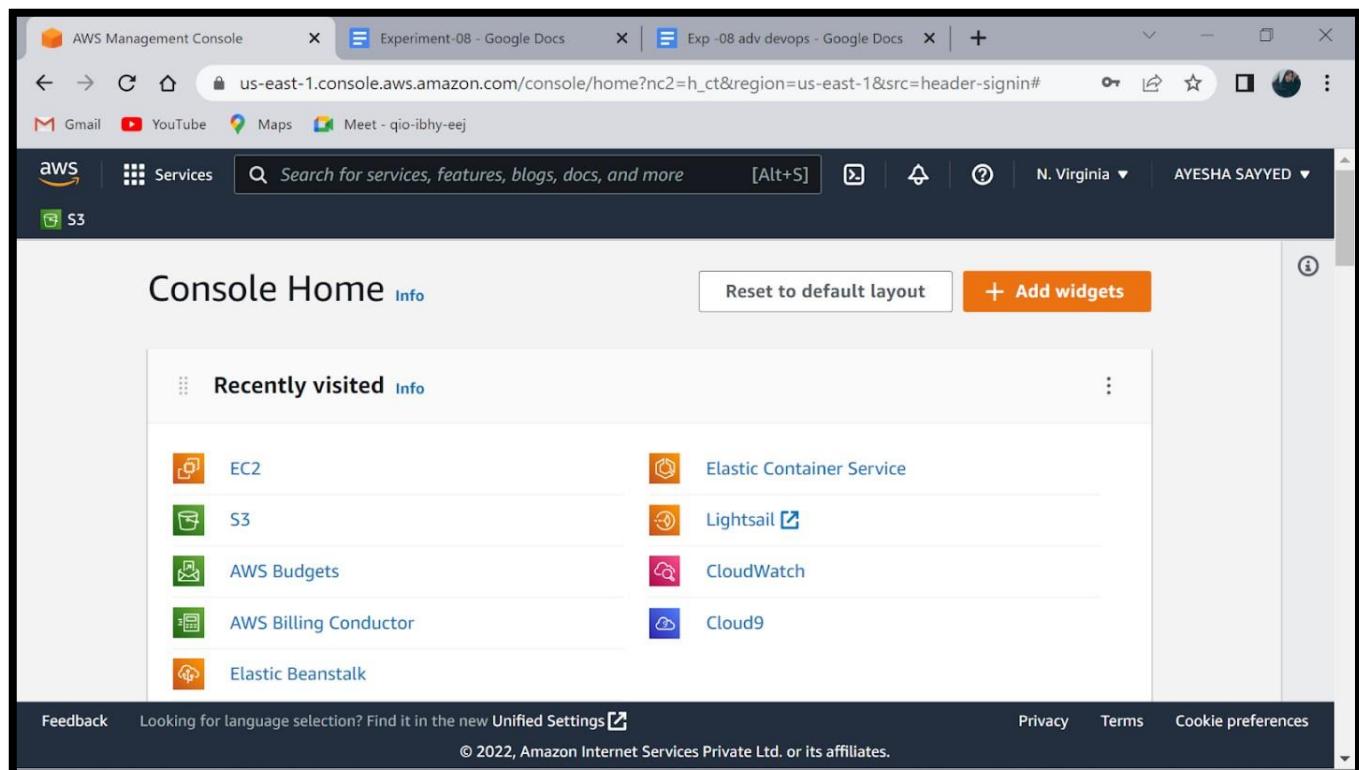
Q1. What is hub.docker.com?

- Docker Hub is a hosted repository service provided by Docker for finding and sharing container images with your team.
- Key features include: Private Repositories: Push and pull container images.
- Automated Builds: Automatically build container images from GitHub and Bitbucket and push them to Docker Hub.

Q2. What is docker hub used for?

- Docker Hub is a hosted repository service provided by Docker for finding and sharing container images with your team.
- Key features include: Private Repositories: Push and pull container images.
- Automated Builds: Automatically build container images from GitHub and Bitbucket and push them to Docker Hub.

Step 1: log in to AWS Management Console Dashboard.



Step 2: Search for EC2 and select it.

The screenshot shows the AWS Management Console search results for 'EC2'. The search bar at the top contains 'EC2'. Below the search bar, there is a sidebar titled 'Services (8)' with various links like 'Features (46)', 'Blogs (1,802)', and 'Marketplace (1,571)'. The main content area is titled 'Services' and shows three results: 'EC2' (Virtual Servers in the Cloud), 'EC2 Image Builder' (A managed service to automate build, customize and deploy OS images), and 'AWS Compute Optimizer' (Recommend optimal AWS Compute resources for your workloads). A red box highlights the 'EC2' result.

Step 3: Go to EC2 Dashboard and “Launch Instances”.

The screenshot shows the AWS EC2 Dashboard. On the left, there is a sidebar with options like 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', and 'Instances'. The 'Instances' section is expanded, showing 'Instances New'. The main content area has a heading 'Launch instance' with the sub-instruction 'To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.' Below this is a large orange 'Launch instance' button, which is highlighted with a blue box. To the right of the main content area, there is a 'Service health' section showing 'Region: US East (N. Virginia)' and 'Status: This service is operating normally'. At the bottom, there is a 'Scheduled events' section and a table for 'Zones'.

Step 4: Launch an Instance and then ‘Give name to your instances’.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' section, the 'Name' field contains 'ayesha_exp-08'. Below it, under 'Application and OS Images (Amazon Machine Image)', there is a list of operating systems including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The 'Ubuntu' option is highlighted with a blue border. At the bottom of the screen, there are links for 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

Step 5: Select UBUNTU and 20.04 LTS Sever.

The screenshot shows the AWS EC2 'Quick Start' page. It displays a grid of operating system options. The 'Ubuntu' option is selected and highlighted with a blue border. Below the grid, a detailed view of the selected 'Ubuntu Server 20.04 LTS (HVM), SSD Volume Type' AMI is shown, including its identifier (ami-08d4ac5b634553e16), supported architectures (64-bit (x86) / 64-bit (Arm)), virtualization type (hvm), ENA support (true), and root device type (ebs). A note indicates that this is a 'Free tier eligible' AMI. To the right, there is a search bar labeled 'Browse more AMIs' and a link to 'Including AMIs from AWS, Marketplace and the Community'. At the bottom of the screen, there are links for 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

Step 6: Create a Key Pair.

The screenshot shows the AWS Lambda console with a modal dialog titled "Create key pair".

Key pair name: ayesha-keypair

Key pair type: RSA (selected)

Private key file format: .pem (selected)

Create key pair button (highlighted with a purple border)

Feedback: Looking for language selection? Find it in the new Unified Settings.

Page Headers: Launch an instance, Experiment-08 - G, What is docker hub, Classes, Experiment-8 - G, us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:, Gmail, YouTube, Maps, Meet - qio-ibhy-eej, Services, Search for services, features, blogs, docs, and more, [Alt+S], N. Virginia, AYESHA SAYYED.

Step 7: Allow all the HTTPs Traffic from the Internet and click to ‘Launch Instance’.

The screenshot shows the AWS Firewall (security groups) configuration page. It displays three tabs: 'Create security group' (selected), 'Select existing security group', and 'Info'. A note states: 'A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.' Below this, it says: 'We'll create a new security group called 'launch-wizard-5' with the following rules:'

- Allow SSH traffic from Anywhere
Helps you connect to your instance
- Allow HTTPs traffic from the internet
To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

The screenshot shows the AWS Summary launch instance page. It includes sections for 'Number of instances' (set to 1), 'Software Image (AMI)' (Canonical, Ubuntu, 20.04 LTS), and 'Virtual server type (instance type)'. At the bottom right is a prominent orange 'Launch instance' button, which is highlighted with a red rectangle.

Feedback: Looking for language selection? Find it in the new Unified Settings [\[link\]](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Privacy Terms Cookie preferences

Step 8: Instances Created Successfully. Now view the instances.

The screenshot shows a browser window with multiple tabs open, including 'Launch an instance', 'Experiment-08 - G', 'What is docker hub', 'Classes', 'Experiment-8 - Go', and the current tab 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances'. The main content area is titled 'EC2 > Instances > Launch an instance'. It displays a green success message: 'Successfully initiated launch of instance (i-04f94f3196ce6c508)'. Below this, there's a link to 'Launch log'. A 'Next Steps' section follows, containing links to 'Create billing alerts', 'How to connect to your instance', and 'View more resources to get you started'. At the bottom right is an orange button labeled 'View all instances'. The footer includes links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

Success
Successfully initiated launch of instance (i-04f94f3196ce6c508)

▶ Launch log

Next Steps

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)

How to connect to your instance

Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use

Click View Instances to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out [how to connect to your instance](#)

[View more resources to get you started](#)

View all instances

Feedback Looking for language selection? Find it in the new Unified Settings [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Step 9: Connect the Instances, Go to SSH client and copy public DNS.

The screenshot shows the AWS EC2 Instances page. A single instance, 'ayesha_exp-08', is listed in the table. The 'Connect' button is highlighted with a green box. The instance details show it is running, has an instance type of t2.micro, and a status of 2/2.

The screenshot shows the 'Connect to instance' page for the instance 'ayesha_exp-08'. The 'SSH client' tab is selected. It provides instructions for connecting via SSH, including the instance ID and a command to change file permissions:

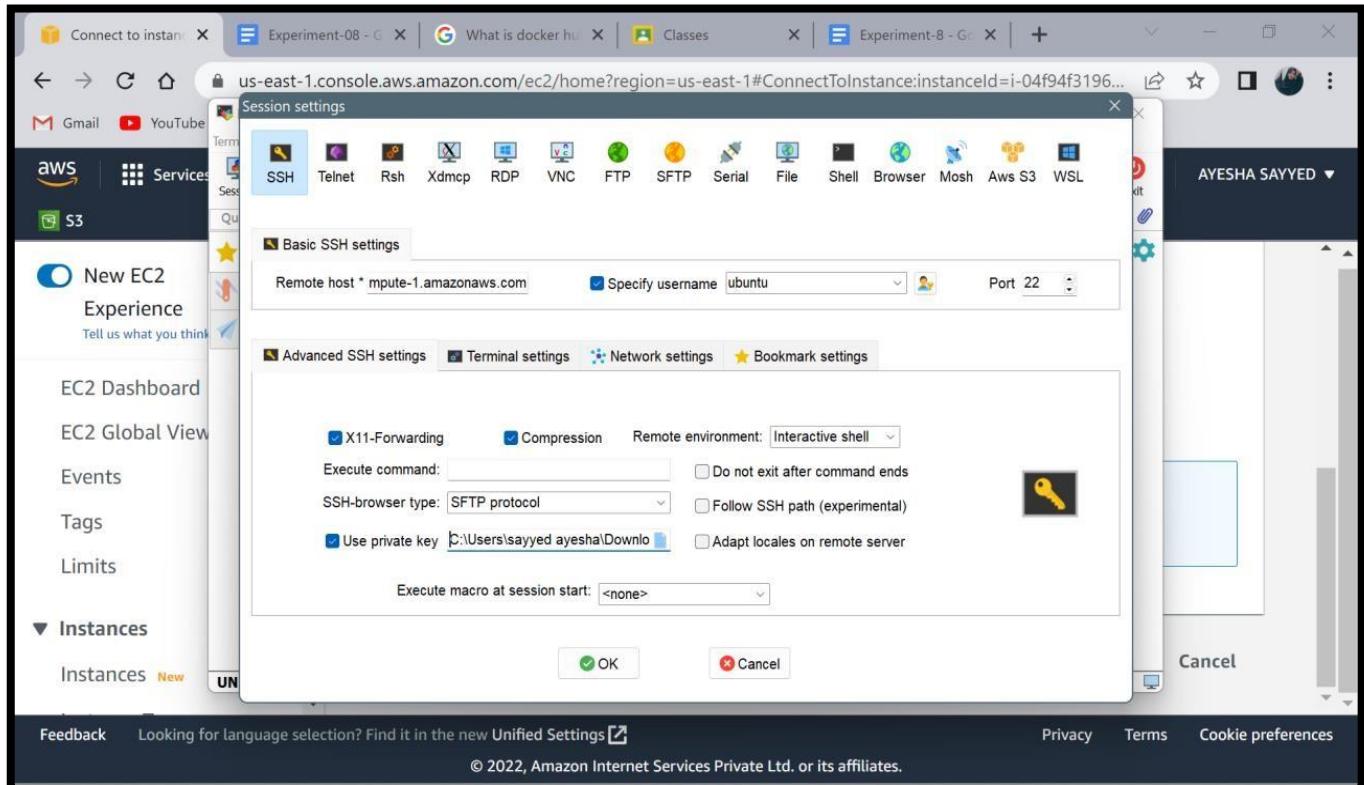
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ayesha-keypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 ayesha-keypair.pem

The screenshot shows the AWS EC2 Instances page. A message box at the top right indicates that the Public DNS has been copied. The copied text is shown in the message box:

i-04f94f3196ce6c508 (ayesha_exp-08)
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ayesha-keypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 ayesha-keypair.pem

The copied text is also pasted into the clipboard icon.

Step 10: Go to ‘MobaXterm’→Session →SSH →Adv SSH Setting then allow it.

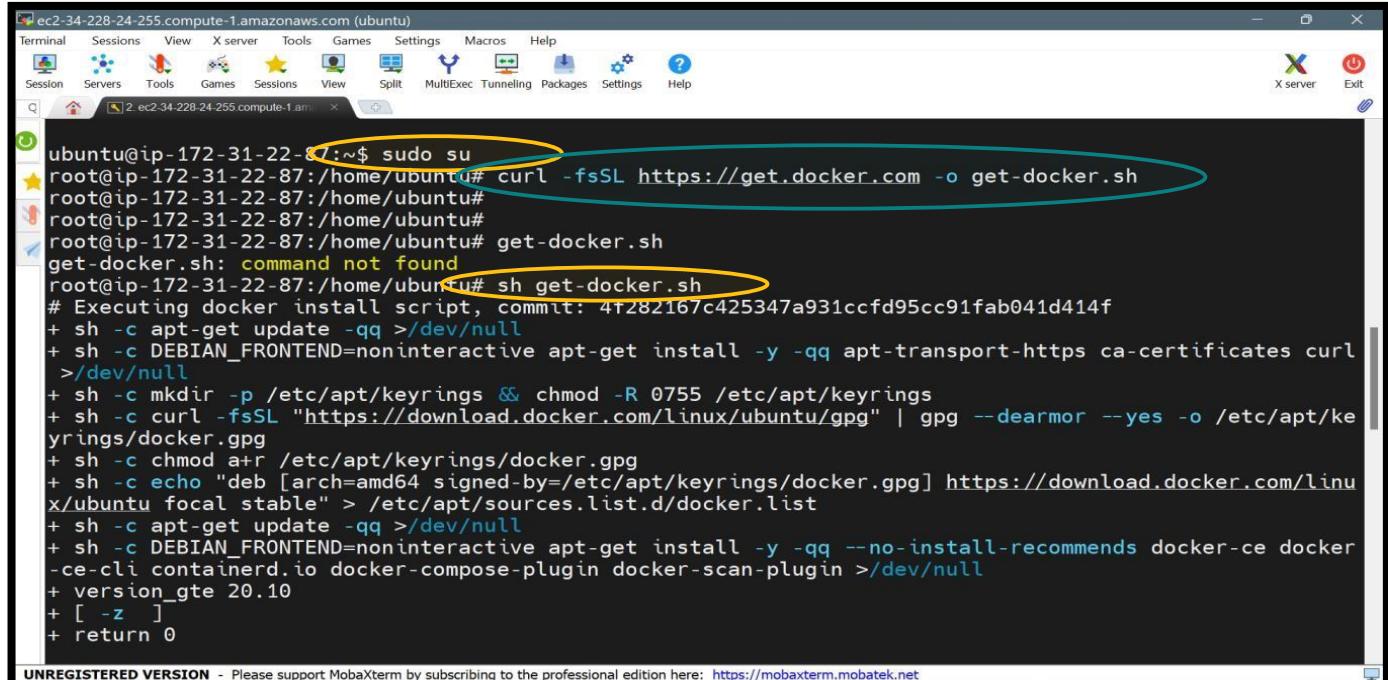


Q3. Install docker on AWS EC2 –Ubuntu by using curl

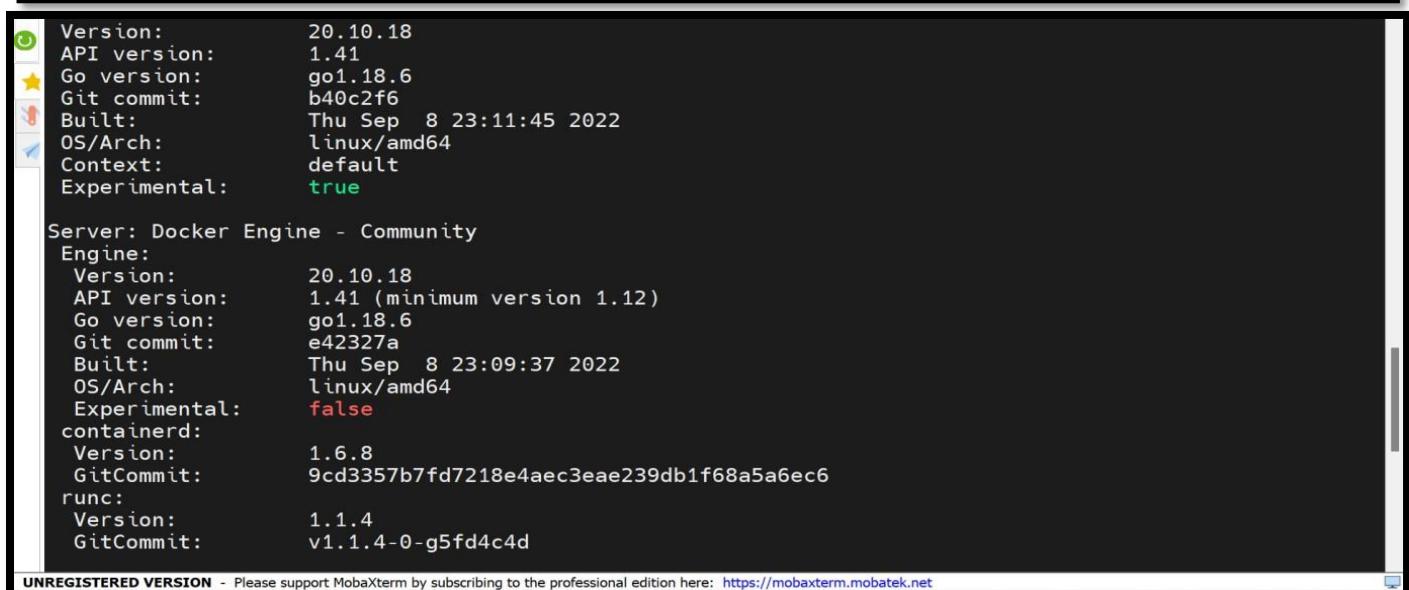
```
#curl -fsSLhttps://get.docker.com -o get-docker.sh  
#sh get-docker.sh
```

Step 11: Run the command ‘sudo su’ to gain root user access. Then enter commands:

```
[#curl -fsSLhttps://get.docker.com -o get-docker.sh #sh get-docker.sh]
```



```
ubuntu@ip-172-31-22-87:~$ sudo su
root@ip-172-31-22-87:/home/ubuntu# curl -fsSL https://get.docker.com -o get-docker.sh
root@ip-172-31-22-87:/home/ubuntu#
root@ip-172-31-22-87:/home/ubuntu# get-docker.sh
get-docker.sh: command not found
root@ip-172-31-22-87:/home/ubuntu# sh get-docker.sh
# Executing docker install script, commit: 4t282167c425347a931ccfd95cc91fab041d414f
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl
>/dev/null
+ sh -c mkdir -p /etc/apt/keyrings && chmod -R 0755 /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor --yes -o /etc/apt/ke
yrings/docker.gpg
+ sh -c chmod a+r /etc/apt/keyrings/docker.gpg
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubu
ntu focal stable" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends docker-ce docker
-ce-cli containerd.io docker-compose-plugin docker-scan-plugin >/dev/null
+ version_gte 20.10
+ [ -z ]
+ return 0
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```



```
Version:          20.10.18
API version:     1.41
Go version:      go1.18.6
Git commit:      b40c2f6
Built:           Thu Sep  8 23:11:45 2022
OS/Arch:         linux/amd64
Context:         default
Experimental:    true

Server: Docker Engine - Community
Engine:
  Version:          20.10.18
  API version:     1.41 (minimum version 1.12)
  Go version:      go1.18.6
  Git commit:      e42327a
  Built:           Thu Sep  8 23:09:37 2022
  OS/Arch:         linux/amd64
  Experimental:    false
  containerd:
    Version:        1.6.8
    GitCommit:      9cd3357b7fd7218e4aec3eae239db1f68a5a6ec6
  runc:
    Version:        1.1.4
    GitCommit:      v1.1.4-0-g5fd4c4d
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

Q4. Run hello-world from docker hub and explain the steps.

Step 12: Enter command ‘docker -- version’ to see the current docker version.

Then run command ‘docker images’ to see installed images. Also run command ‘docker run hello-world’ which will run helloworld from docker.hub.com and run it.

The screenshot shows a terminal window titled "ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)". The terminal output is as follows:

```
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@ip-172-31-22-87:/home/ubuntu# docker --version
Docker version 20.10.18, build b40c2f6
root@ip-172-31-22-87:/home/ubuntu# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:62af9efd515a25f84961b70f973a798d2eca956b1b2b026d0a4a63a3b0b6a3f2
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 13: Now run ‘docker images’ again, the repository will have an image named as ‘hello-world’.

The screenshot shows a terminal window titled "ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)". The terminal output is as follows:

```
For more examples and ideas, visit:
https://docs.docker.com/get-started/
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest feb5d9fea6a5 11 months ago 13.3kB
root@ip-172-31-22-87:/home/ubuntu# docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

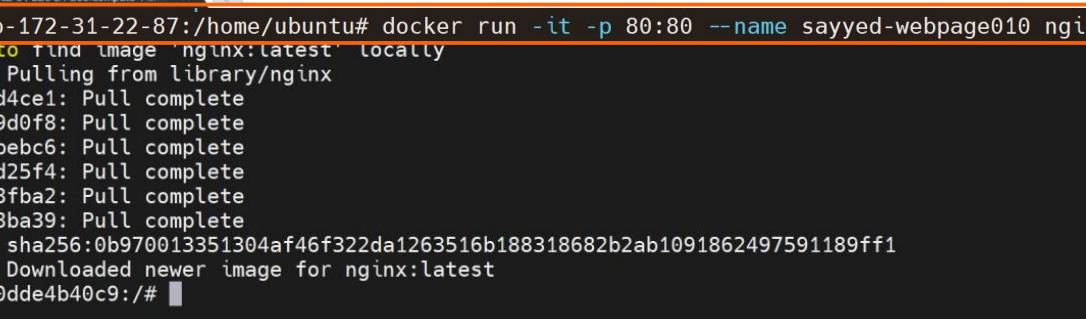
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

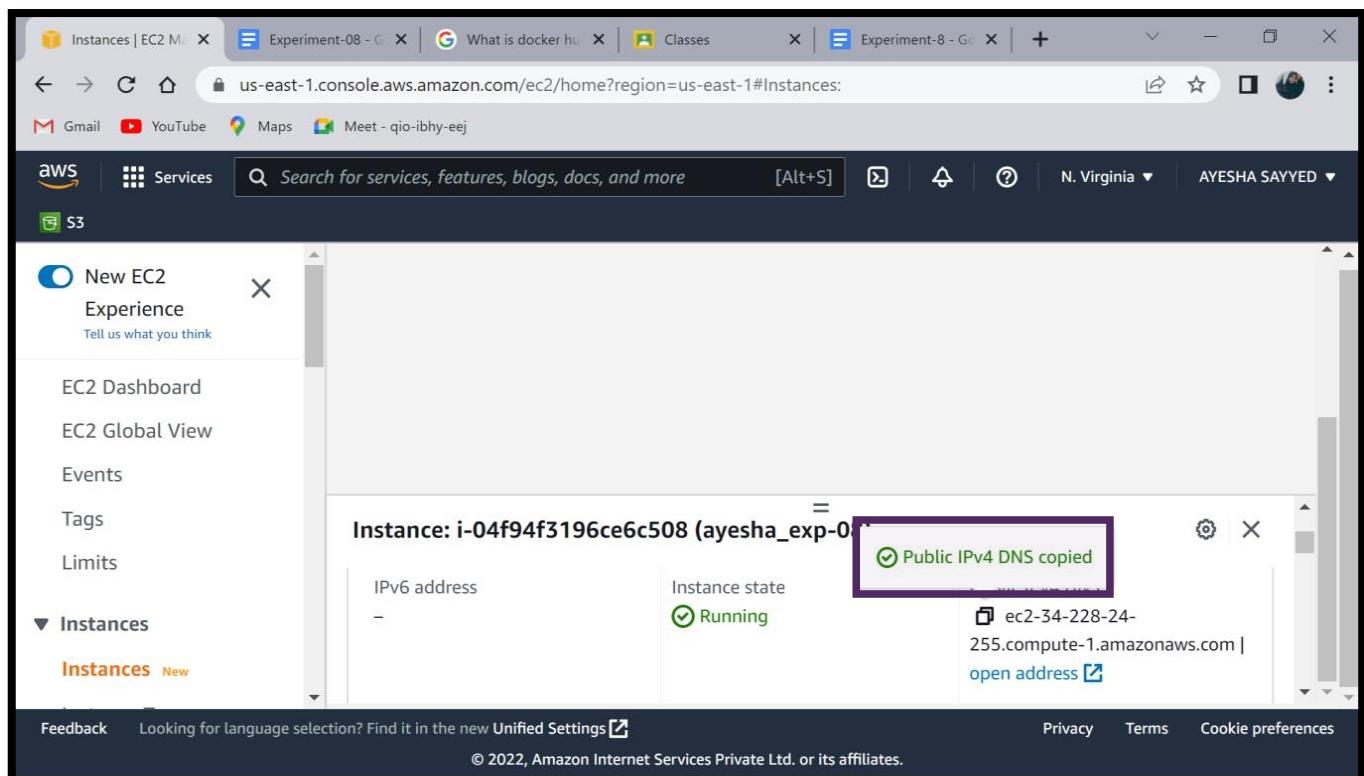
Share images, automate workflows, and more with a free Docker ID:
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

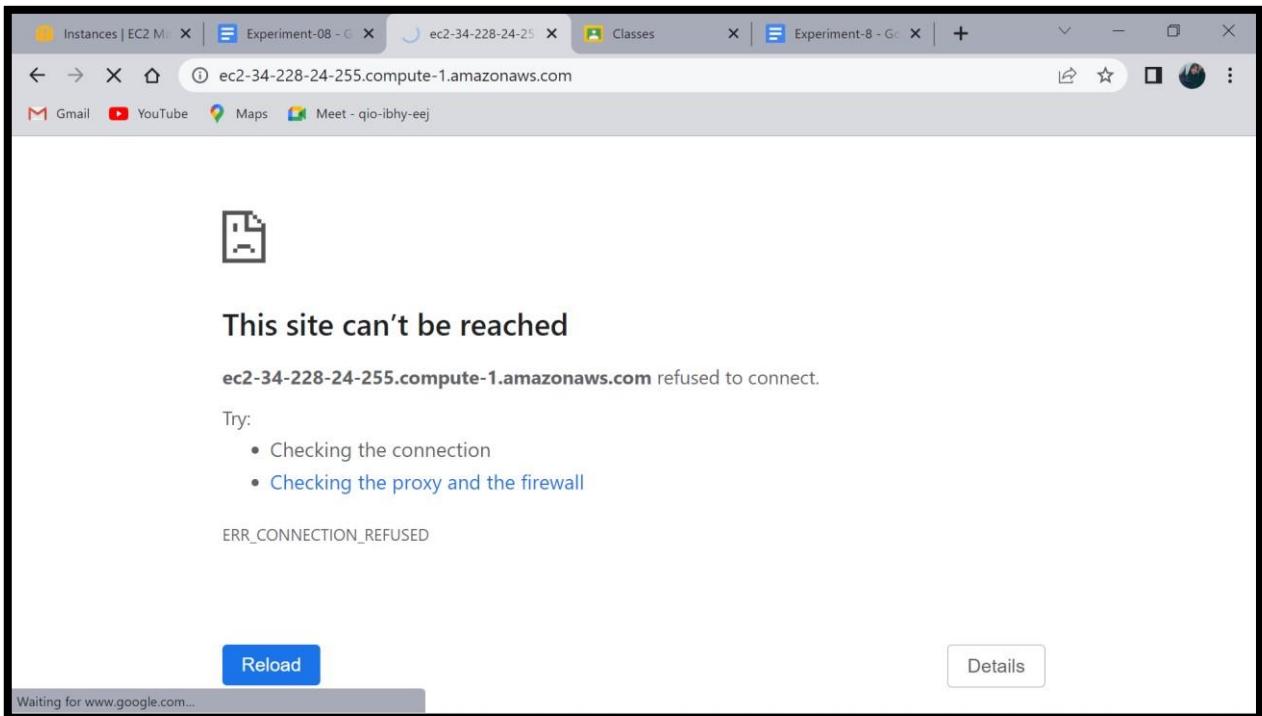
Step 14: Run command ‘`docker run -it -p 80:80 –name[name of your webpage] nginx bash`’. Now to launch the nginx web server, copy the Ipv4 address from the EC2 instance details and paste it to the address bar of Web Browser.



```
root@ip-172-31-22-87:/home/ubuntu# docker run -it -p 80:80 --name sayyed-webpage010 nginx bash
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
31b3f1ad4ce1: Pull complete
fd42b079d0f8: Pull complete
30585fbbebc6: Pull complete
18f4ffdd25f4: Pull complete
9dc932c8fba2: Pull complete
600c24b8ba39: Pull complete
Digest: sha256:0b970013351304af46f322da1263516b188318682b2ab1091862497591189ff1
Status: Downloaded newer image for nginx:latest
root@da0dde4b40c9:/#
```



Can't able to Reach the site.



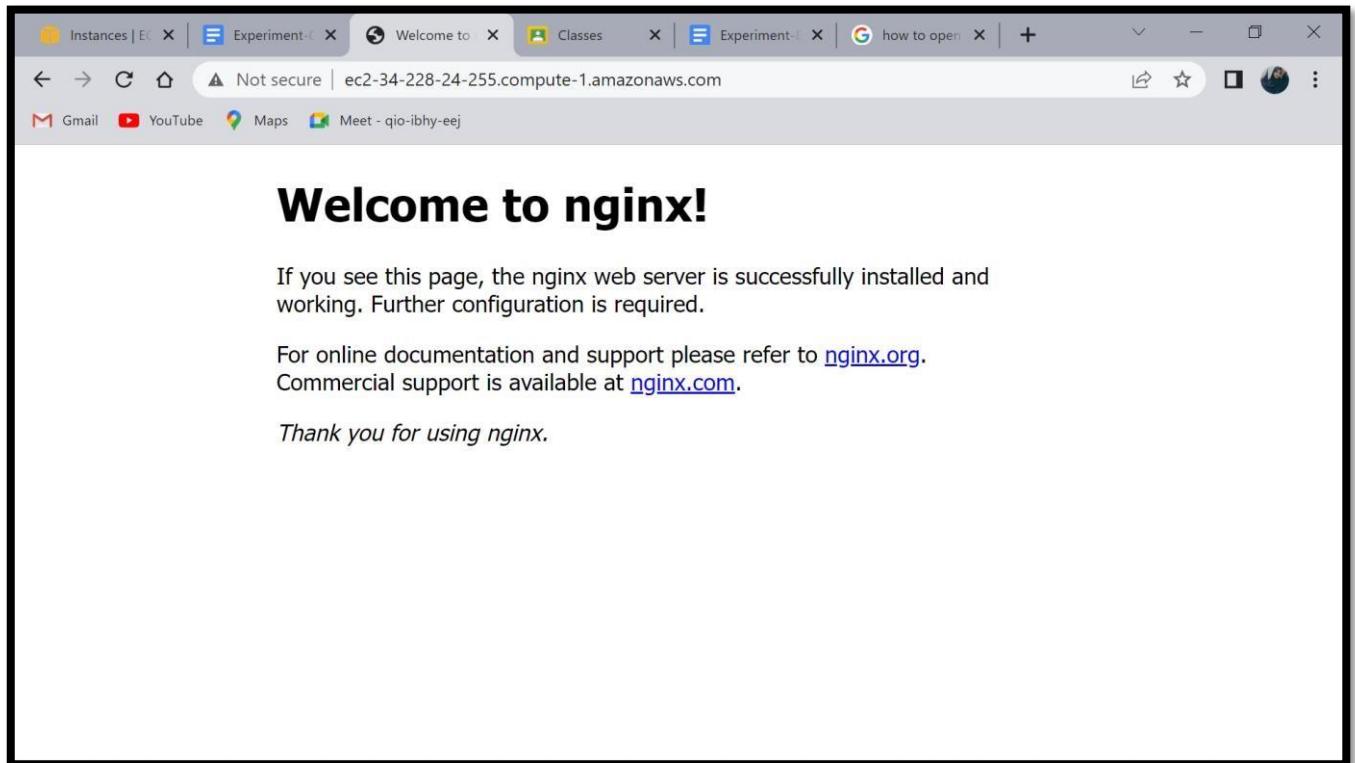
Step 15: Within the container use command ‘service nginx start’ to deploy the web server.

```
root@ip-172-31-22-87:/home/ubuntu# docker run -it -p 80:80 --name saved-webpage010 nginx bash
root@279ee355c666:/# service nginx start
2022/09/17 17:44:39 [notice] 12#12: using the "epoll" event method
2022/09/17 17:44:39 [notice] 12#12: nginx/1.23.1
2022/09/17 17:44:39 [notice] 12#12: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/09/17 17:44:39 [notice] 12#12: OS: Linux 5.13.0-1029-aws
2022/09/17 17:44:39 [notice] 12#12: getrlimit(RLIMIT_NOFILE): 1048576:1048576
root@279ee355c666:/# 2022/09/17 17:44:39 [notice] 13#13: start worker processes
2022/09/17 17:44:39 [notice] 13#13: start worker process 14

root@279ee355c666:/#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

After deploying the web server, the web page will be visible without any errors.



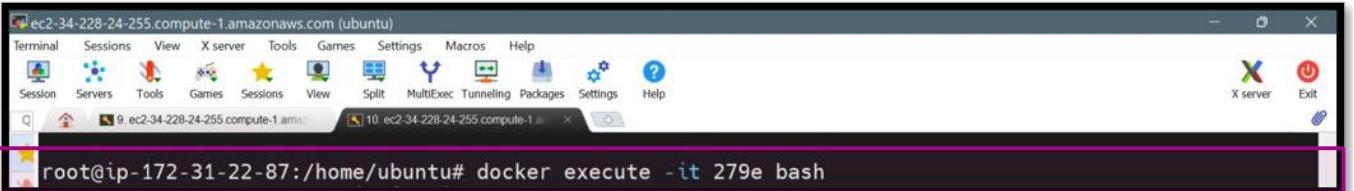
Step 16: Take a duplicate tab run command ‘sudo su’ →‘docker ps’ →‘docker ps -a’ →‘docker execute -it 279e bash’.

A screenshot of a terminal window titled 'ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)'. The terminal shows a user running 'root@ip-172-31-22-87:~/home/ubuntu# docker execute -it 279e bash'. The output includes an error message about an unknown shorthand flag, followed by usage instructions and a detailed list of Docker command-line options. A pink box highlights the error message.

```
root@ip-172-31-22-87:~/home/ubuntu# docker execute -it 279e bash
Usage: docker [OPTIONS] COMMAND
      A self-sufficient runtime for containers

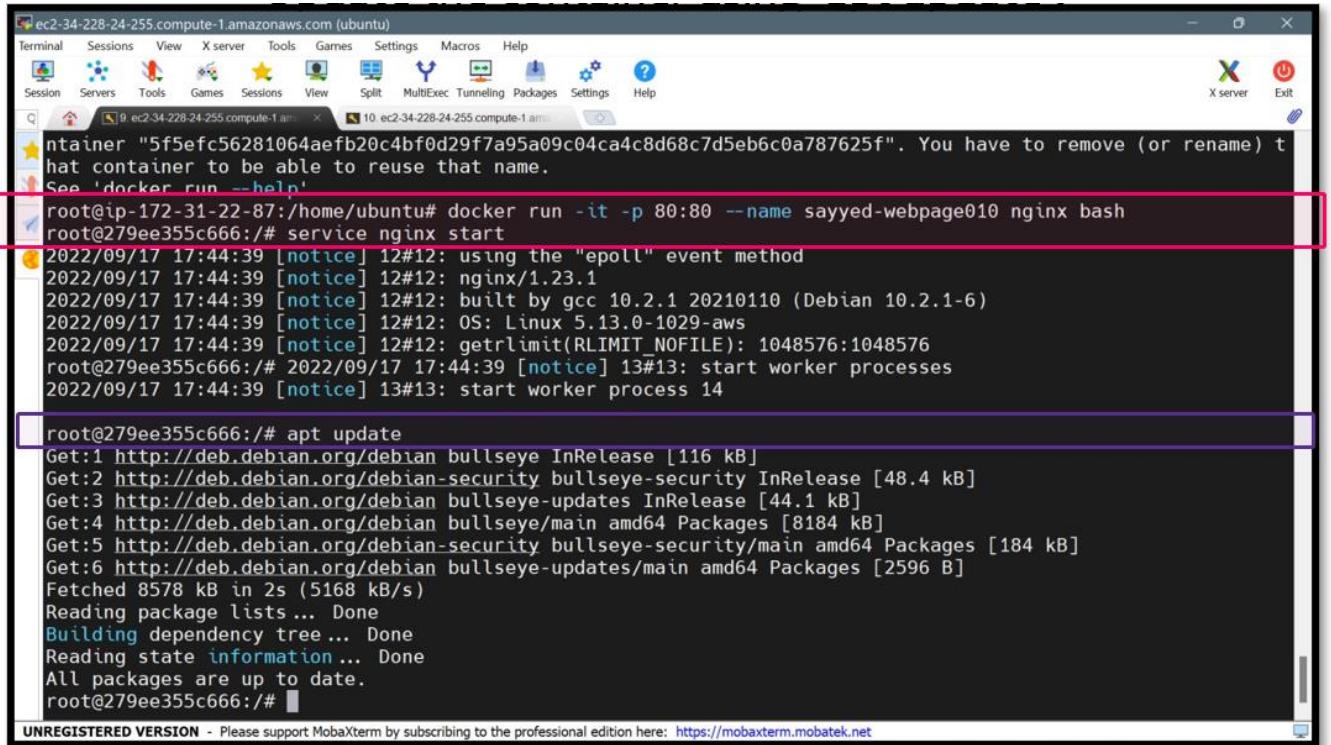
Options:
  --config string          Location of client config files (default "/root/.docker")
  -c, --context string     Name of the context to use to connect to the daemon (overrides
                           DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug               Enable debug mode
  -H, --host list           Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
  --tls
  --tlscacert string       Trust certs signed only by this CA (default "/root/.docker/ca.pem")
  --tlscert string          Path to TLS certificate file (default "/root/.docker/cert.pem")
  --tlskey string            Path to TLS key file (default "/root/.docker/key.pem")
  --tlsverify                Use TLS and verify the remote
  -v, --version              Print version information and quit
```

Step 17: To make changes to a file within the container use command ‘docker exec -it(container id bash)’.



```
root@ip-172-31-22-87:/home/ubuntu# docker execute -it 279e bash
```

Run ‘docker run -it -p 80:80 --name[name of your webpage] nginx bash’ → ‘service nginx start’. Update the container using “apt update”.

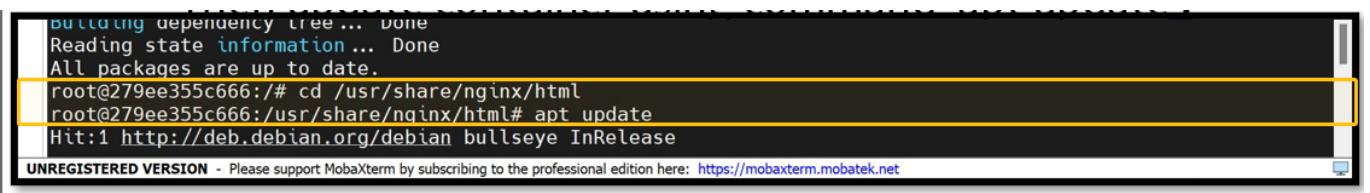


```
root@ip-172-31-22-87:/home/ubuntu# docker run -it -p 80:80 --name sayyed-webpage010 nginx bash
root@279ee355c666:/# service nginx start
2022/09/17 17:44:39 [notice] 12#12: using the "epoll" event method
2022/09/17 17:44:39 [notice] 12#12: nginx/1.23.1
2022/09/17 17:44:39 [notice] 12#12: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/09/17 17:44:39 [notice] 12#12: OS: Linux 5.13.0-1029-aws
2022/09/17 17:44:39 [notice] 12#12: getrlimit(RLIMIT_NOFILE): 1048576:1048576
root@279ee355c666:/# 2022/09/17 17:44:39 [notice] 13#13: start worker processes
2022/09/17 17:44:39 [notice] 13#13: start worker process 14

root@279ee355c666:/# apt update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8184 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [184 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [2596 B]
Fetched 8578 kB in 2s (5168 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@279ee355c666:/#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

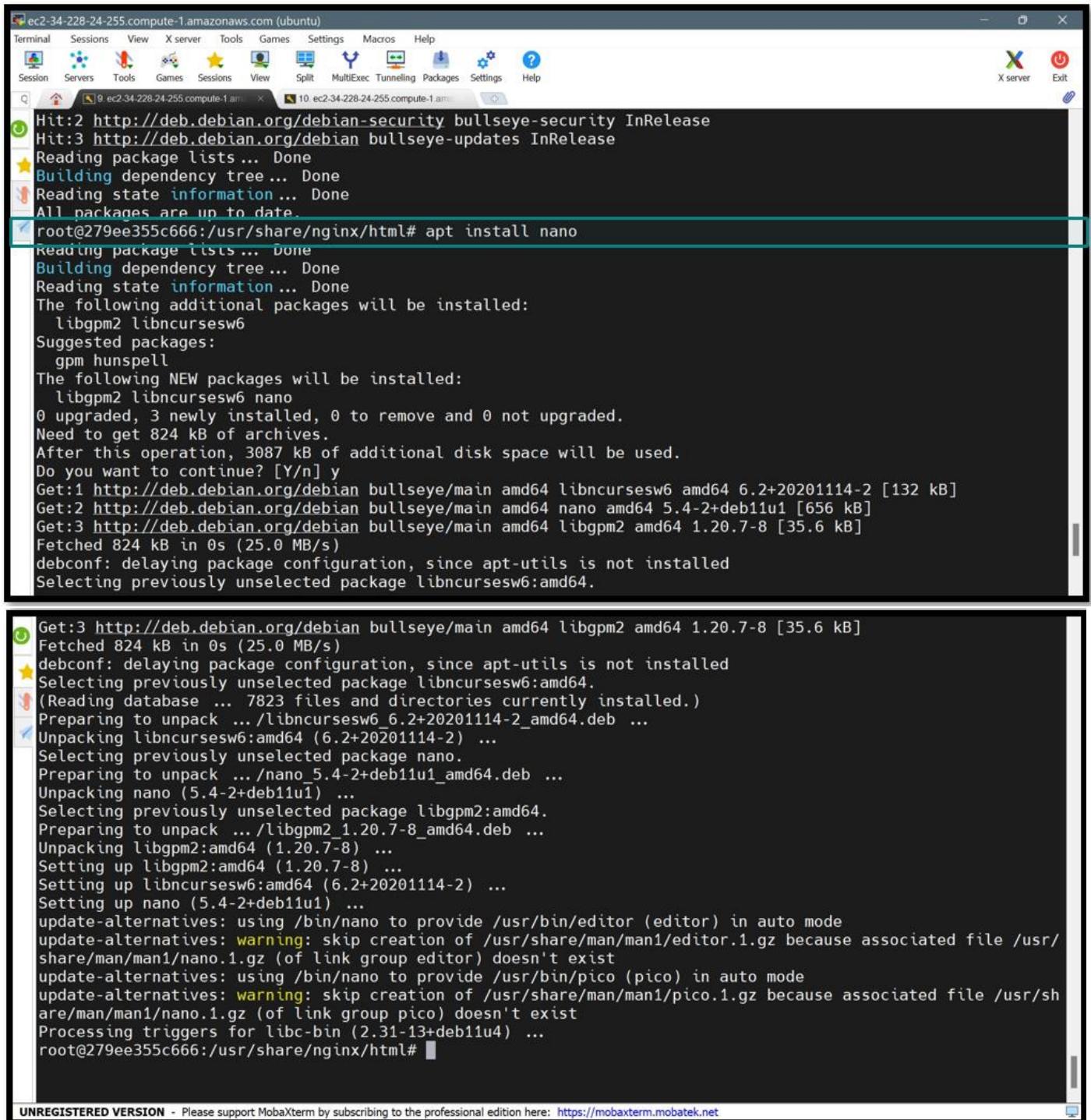
Now within the container navigate to the html directory using command: ‘cd /usr/share/nginx/html’. Then update the container using [apt update].



```
root@279ee355c666:/# cd /usr/share/nginx/html
root@279ee355c666:/# apt update
Hit:1 http://deb.debian.org/debian bullseye InRelease
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 18: Now use command ‘apt install nano’ to install nano text Editor.



```
ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunnelling Packages Settings Help
X server Exit
9 ec2-34-228-24-255.compute-1.am... 10 ec2-34-228-24-255.compute-1.am...
Hit:2 http://deb.debian.org/debian-security bullseye-security InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@279ee355c666:/usr/share/nginx/html# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libncursesw6
Suggested packages:
  gpm hunspell
The following NEW packages will be installed:
  libgpm2 libncursesw6 nano
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 824 kB of archives.
After this operation, 3087 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main amd64 libncursesw6 amd64 6.2+20201114-2 [132 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 nano amd64 5.4-2+deb11u1 [656 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 libgpm2 amd64 1.20.7-8 [35.6 kB]
Fetched 824 kB in 0s (25.0 MB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libncursesw6:amd64.

Get:3 http://deb.debian.org/debian bullseye/main amd64 libgpm2 amd64 1.20.7-8 [35.6 kB]
Fetched 824 kB in 0s (25.0 MB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libncursesw6:amd64.
(Reading database ... 7823 files and directories currently installed.)
Preparing to unpack .../libncursesw6_6.2+20201114-2_amd64.deb ...
Unpacking libncursesw6:amd64 (6.2+20201114-2) ...
Selecting previously unselected package nano.
Preparing to unpack .../nano_5.4-2+deb11u1_amd64.deb ...
Unpacking nano (5.4-2+deb11u1) ...
Selecting previously unselected package libgpm2:amd64.
Preparing to unpack .../libgpm2_1.20.7-8_amd64.deb ...
Unpacking libgpm2:amd64 (1.20.7-8) ...
Setting up libgpm2:amd64 (1.20.7-8) ...
Setting up libncursesw6:amd64 (6.2+20201114-2) ...
Setting up nano (5.4-2+deb11u1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group editor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group pico) doesn't exist
Processing triggers for libc-bin (2.31-13+deb11u4) ...
root@279ee355c666:/usr/share/nginx/html#
```

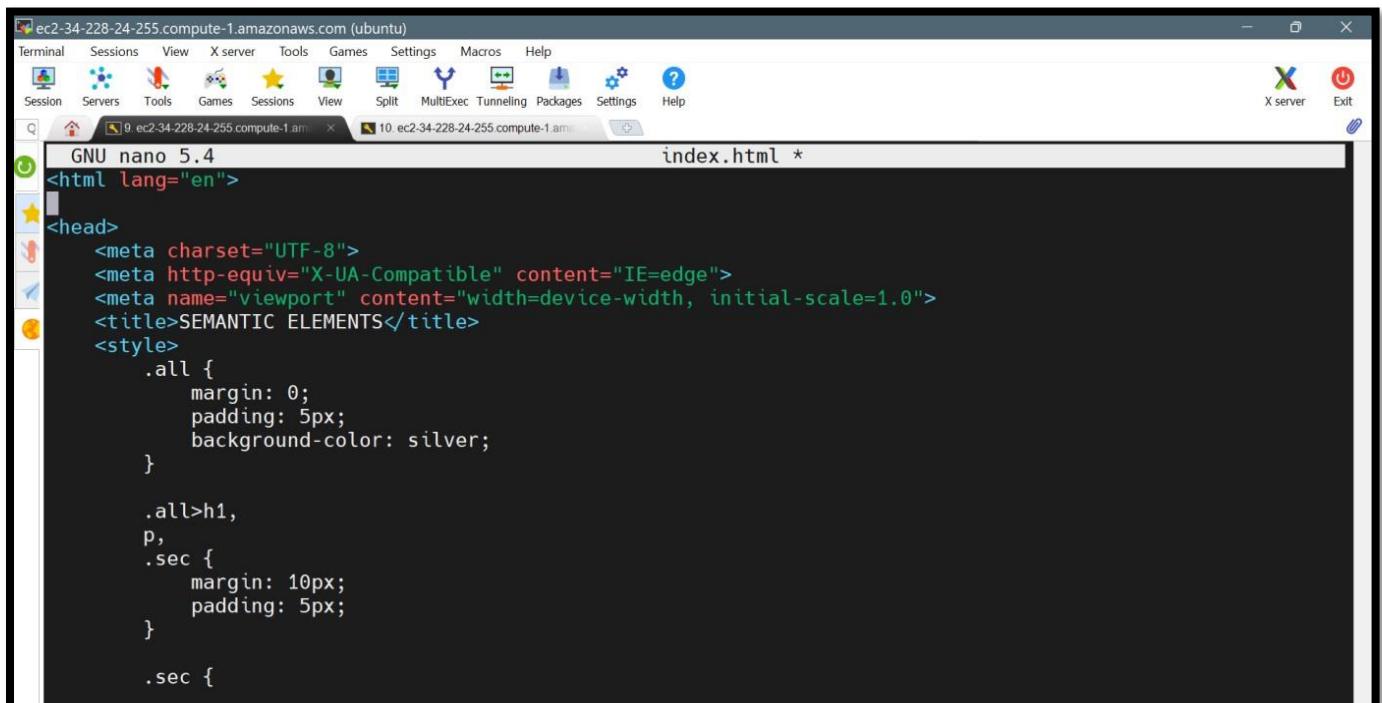
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 19: Now move the original nginx index as a backup so you can create your own html. Index file using command ‘mv index.html index.html.backup’.

Then, open nano text editor using command ‘nano index.html’.

```
root@279ee355c666:/usr/share/nginx/html# mv index.html index.html.backup
root@279ee355c666:/usr/share/nginx/html# ls
50x.html  index.html.backup
root@279ee355c666:/usr/share/nginx/html#
```

Step 20: Write the HTML code then → press ‘Ctrl+O → ENTER → ‘Ctrl+X’. This will help you to save the code and exit.

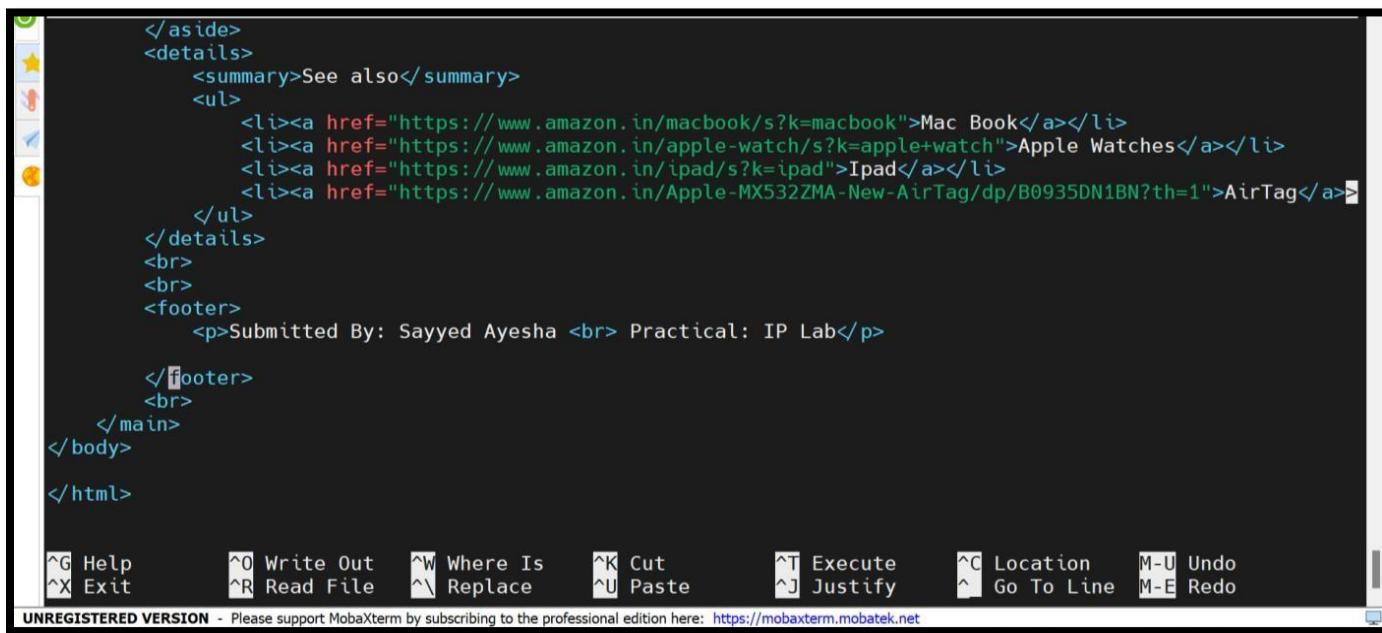


The screenshot shows a terminal window titled "index.html *". The nano text editor is open, displaying the following HTML code:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SEMANTIC ELEMENTS</title>
    <style>
        .all {
            margin: 0;
            padding: 5px;
            background-color: silver;
        }

        .all>h1,
        p,
        .sec {
            margin: 10px;
            padding: 5px;
        }

        .sec {
```



The screenshot shows the same terminal window with the completed HTML code. The code includes semantic elements like aside, details, summary, and footer, along with links to various Apple products on Amazon.

```
</aside>
<details>
    <summary>See also</summary>
    <ul>
        <li><a href="https://www.amazon.in/macbook/s?k=macbook">Mac Book</a></li>
        <li><a href="https://www.amazon.in/apple-watch/s?k=apple+watch">Apple Watches</a></li>
        <li><a href="https://www.amazon.in/ipad/s?k=ipad">Ipad</a></li>
        <li><a href="https://www.amazon.in/Apple-MX532ZMA-New-AirTag/dp/B0935DN1BN?th=1">AirTag</a>
    </ul>
</details>
<br>
<br>
<footer>
    <p>Submitted By: Sayyed Ayesha <br> Practical: IP Lab</p>
</footer>
<br>
</main>
</body>

</html>
```

At the bottom of the screen, there is a menu bar with options like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, and Redo.

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

The iPhone is a line of smartphones designed and marketed by Apple Inc. These devices use Apple's iOS mobile operating system. The first-generation iPhone was announced by then-Apple CEO Steve Jobs on January 9, 2007. Since then, Apple has annually released new iPhone models and iOS updates. As of November 1, 2018, more than 2.2 billion iPhones had been sold. The iPhone has a user interface built around a multi-touch screen. It connects to cellular networks or Wi-Fi and can make calls, browse the web, take pictures, play music and send and receive emails and text messages. Since the iPhone's launch further features have been added, including larger screen sizes, shooting video, waterproofing, the ability to install third-party mobile apps through an app store, and many accessibility features. Up to iPhone 8 and 8 Plus, iPhones used a layout with a single button on the front panel that returns the user to the home screen. Since iPhone X, iPhone models have switched to a nearly bezel-less front screen design with app switching activated by gesture recognition. The older layout today is still used for Apple's currently-produced iPhone SE series. The iPhone is one of the two largest smartphone platforms in the world alongside Android, forming a large part of the luxury market. The iPhone has generated large profits for Apple, making it one of the world's most valuable publicly traded companies. The first-generation iPhone was described as "revolutionary" and a "game-changer" for the mobile phone industry and subsequent models have also garnered praise. The iPhone has been credited with popularizing the smartphone and slate form factor, and with creating a large market for smartphone apps, or "app economy". As of January 2017, Apple's App Store contained more than 2.2 million applications for the iPhone.

The iPhone runs an operating system known as iOS (formerly iPhone OS). It is a variant of the Darwin operating system core found in macOS. Also included is the "Core Animation" software component from Mac OS X v10.5 Leopard. Together with the graphics hardware (and on the iPhone 3GS, OpenGL ES 2.0), it is responsible for the interface's motion graphics. The iPhone comes with a set of bundled applications developed by Apple and supports downloading third-party applications through the App Store. Apple provides free updates to the operating system for the

iPhone

INTERESTING FACT!

What is the meaning of "I"? Next on our list of interesting facts about iPhones is the million-dollar question. The letter 'I' appears in the name of every Apple device. But do you know what this 'I' means? Steve Jobs explained the meaning of this letter when he introduced the iMac to the world in 1998. As a result, we can deduce that the 'I' stands for 'Internet'. The internet was a new concept to the general public at the time. And the iMac was a device that allowed the general public to easily

IMPORTANT FEATURES OF iPhone

VOICE CONTROL

Voice control on the iPhone allows you to open and navigate apps, lock your screen, and more—just by speaking. To turn on the Voice Control feature, go to [Settings > Accessibility > Voice Control](#) tap [Set Up Voice Control](#) and complete the setup. To see a full list of built-in voice commands or to create your own, tap [Customize Commands](#). Since learning Voice Control can be overwhelming at first, we recommend enabling the [Show Hints](#) feature so your phone will automatically suggest commands based on your activity.

ROBOCALL SILENCER

Robocalls—those annoying automated telemarketing calls—are difficult to block. Did you know that you can help combat them with a built-in feature on your iPhone? Open your [Settings](#) menu, select [Phone](#), and then enable [Silence Unknown Callers](#). Calls from unknown numbers will go directly to voicemail without you hearing the ring. You can quickly delete them later.

NOTES SCANNER

If you need to quickly and easily scan and send a document, your iPhone can do the job. Go to your [Notes](#) app, open a new note, tap the [Camera](#) icon on the toolbar at the bottom, and select [Scan Documents](#). Place your document on a flat, well-lit surface, and take a photo. Tap [Keep Scan > Save](#) to save the document. If you want to share it, tap the [Share](#) icon at the top-right corner and select how you'd like to send the document.

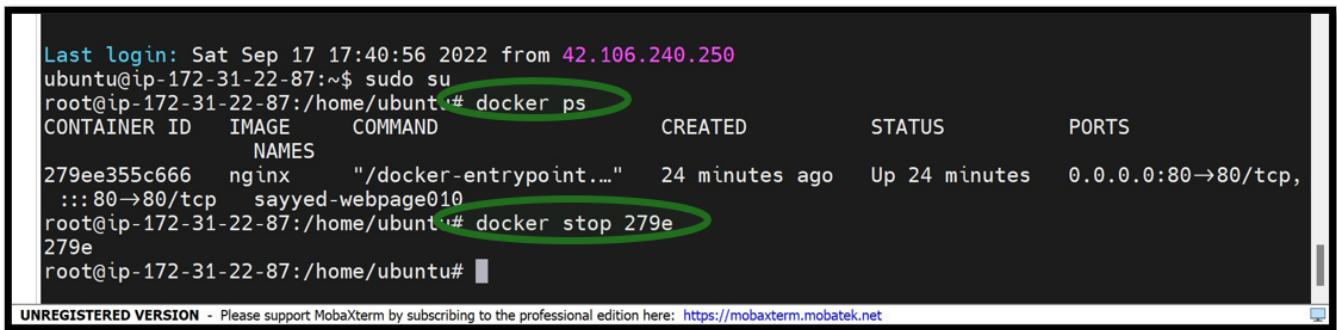
DO YOU KNOW ??

On January 9, 2007, Steve Jobs announced iPhone at the Macworld convention and received huge media attention. Finally later that year, on June 29, 2007 the first iPhone was released. But do you know that the present iPhone was earlier named as "Purple"? But this name never came into existence as the purple name was already taken by another company. After that, the iPhone name was finalized. However, Apple developers still called their development section as "Purple Drom".

See also

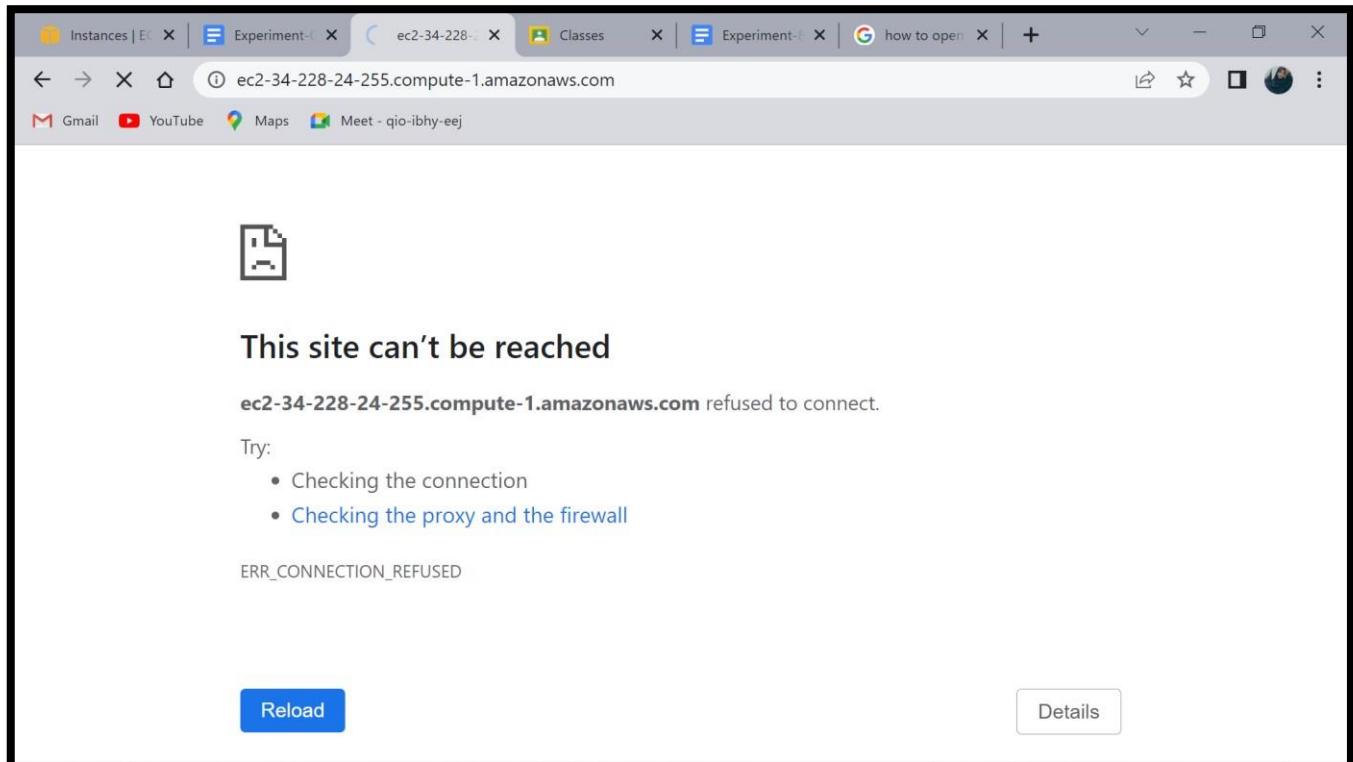
- [Mac Book](#)
- [Apple Watches](#)
- [Ipad](#)
- [AirTag](#)

Step 21: Now run command ‘docker ps’ to get the ‘container id’ and run ‘docker stop (container id)’ to stop the running container.



```
Last login: Sat Sep 17 17:40:56 2022 from 42.106.240.250
ubuntu@ip-172-31-22-87:~$ sudo su
root@ip-172-31-22-87:/home/ubuntu# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
279ee355c666        nginx              "/docker-entrypoint..."   24 minutes ago    Up 24 minutes   0.0.0.0:80→80/tcp,
                   ::80→80/tcp
root@ip-172-31-22-87:/home/ubuntu# docker stop 279e
279e
root@ip-172-31-22-87:/home/ubuntu#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



Q5. Pull 3 or 4 images, one of the python, run “Hello World ” inside container.

Step 22: Run ‘docker --version’ to get the version also run ‘docker images’ to check the images upload in the repository.

```
root@ip-172-31-22-87:/home/ubuntu# docker --version
Docker version 20.10.18, build b40c2f6
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
nginx          latest   2d389e545974  4 days ago   142MB
hello-world    latest   feb5d9fea6a5  11 months ago  13.3kB
```

Step 23: Now pull mysql from ‘docker.hub.com’ to install mysql image.
Then run command ‘docker images’ to check the images upload in the repository.

```
hello-world    latest   feb5d9fea6a5  11 months ago  13.3kB
root@ip-172-31-22-87:/home/ubuntu# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
051f419db9dd: Pull complete
7627573fa82a: Pull complete
a44b358d7796: Pull complete
95753aff4b95: Pull complete
a1fa3bee53f4: Pull complete
f5227e0d612c: Pull complete
b4b4368b1983: Pull complete
f26212810c32: Pull complete
d803d4215f95: Pull complete
d5358a7f7d07: Pull complete
435e8908cd69: Pull complete
Digest: sha256:b9532b1edea72b6cee12d9f5a78547bd3812ea5db842566e17f8b33291ed2921
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql          latest   43fcfc0776d   2 days ago   449MB
nginx          latest   2d389e545974  4 days ago   142MB
hello-world    latest   feb5d9fea6a5  11 months ago  13.3kB
root@ip-172-31-22-87:/home/ubuntu# docker pull python
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Same like mysql, now pull python from ‘docker.hub.com’ to install python image.

Then run command ‘docker images’ to check the images upload in the repository

The screenshot shows a terminal window titled 'ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)'. The terminal content is as follows:

```
root@ip-172-31-22-87:/home/ubuntu# docker pull python
Using default tag: latest
latest: Pulling from library/python
23858da423a6: Pull complete
326f452ade5c: Pull complete
a42821cd14fb: Pull complete
8471b75885ef: Pull complete
8ffa7aaef404: Pull complete
15132af73342: Pull complete
aaf3b07565c2: Pull complete
736f7bc16867: Pull complete
94da21e53a5b: Pull complete
Digest: sha256:e9c35537103a2801a30b15a77d4a56b35532c964489b125ec1ff24f3d5b53409
Status: Downloaded newer image for python:latest
docker.io/library/python:latest
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
mysql           latest        43fcfc0776d  2 days ago   449MB
python          latest        e285995a3494  4 days ago   921MB
nginx           latest        2d389e545974  4 days ago   142MB
hello-world     latest        feb5d9fea6a5  11 months ago  13.3kB
root@ip-172-31-22-87:/home/ubuntu#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 24: Now, to get access to the python container use command→ ‘docker run -it (images id) bash’ and type ‘python’ to enter the shell and write any code.

The screenshot shows a terminal window titled 'ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)'. The terminal content is as follows:

```
status: downloaded newer image for python:latest
docker.io/library/python:latest
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
mysql           latest        43fcfc0776d  2 days ago   449MB
python          latest        e285995a3494  4 days ago   921MB
nginx           latest        2d389e545974  4 days ago   142MB
hello-world     latest        feb5d9fea6a5  11 months ago  13.3kB
root@ip-172-31-22-87:/home/ubuntu# docker run -it e285 bash
root@2ac2fbc21b4e:/# python
Python 3.10.7 (main, Sep 13 2022, 14:31:33) [GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("ADV-DEVOPS EXP-08")
ADV-DEVOPS EXP-08
>>> print("Faculty-DR. ZAINAB MIRZA")
Faculty-DR. ZAINAB MIRZA
>>>
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 25: Enter command ‘docker --version’ to see the current docker version.

Then enter command ‘docker python --version’ to see the current python version.

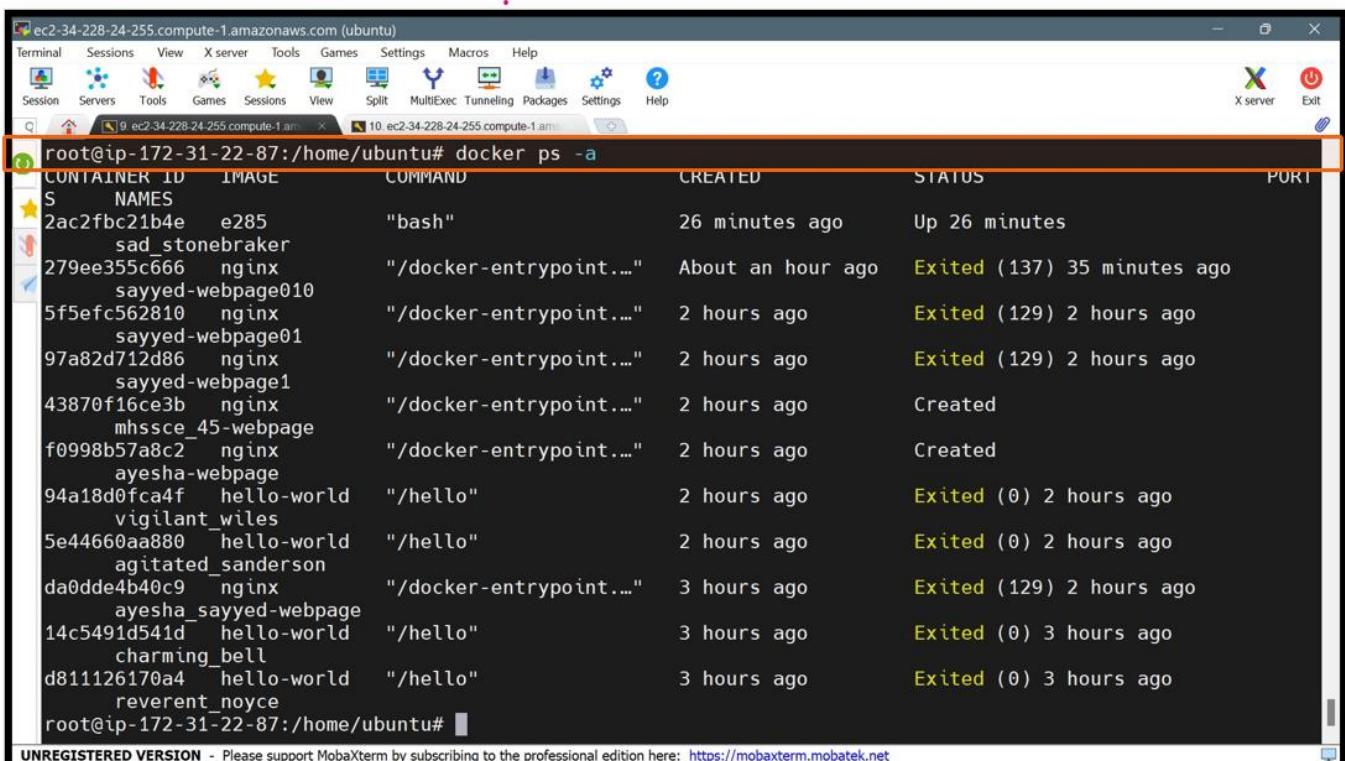
Same command ‘docker mysql --version’ to see the current mysql version.

And then enter ‘docker nginx --version’ to see the current nginx version.

```
root@ip-172-31-22-87:/home/ubuntu# docker --version
Docker version 20.10.18, build b40c2f6
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           latest    43fcfca0776d  2 days ago   449MB
python          latest    e285995a3494  4 days ago   921MB
nginx           latest    2d389e545974  4 days ago   142MB
hello-world     latest    feb5d9fea6a5  11 months ago 13.3kB
root@ip-172-31-22-87:/home/ubuntu# docker python --version
docker version 20.10.18, build b40c2f6
root@ip-172-31-22-87:/home/ubuntu# docker mysql --version
docker version 20.10.18, build b40c2f6
root@ip-172-31-22-87:/home/ubuntu# docker nginx --version
docker version 20.10.18, build b40c2f6
root@ip-172-31-22-87:/home/ubuntu#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

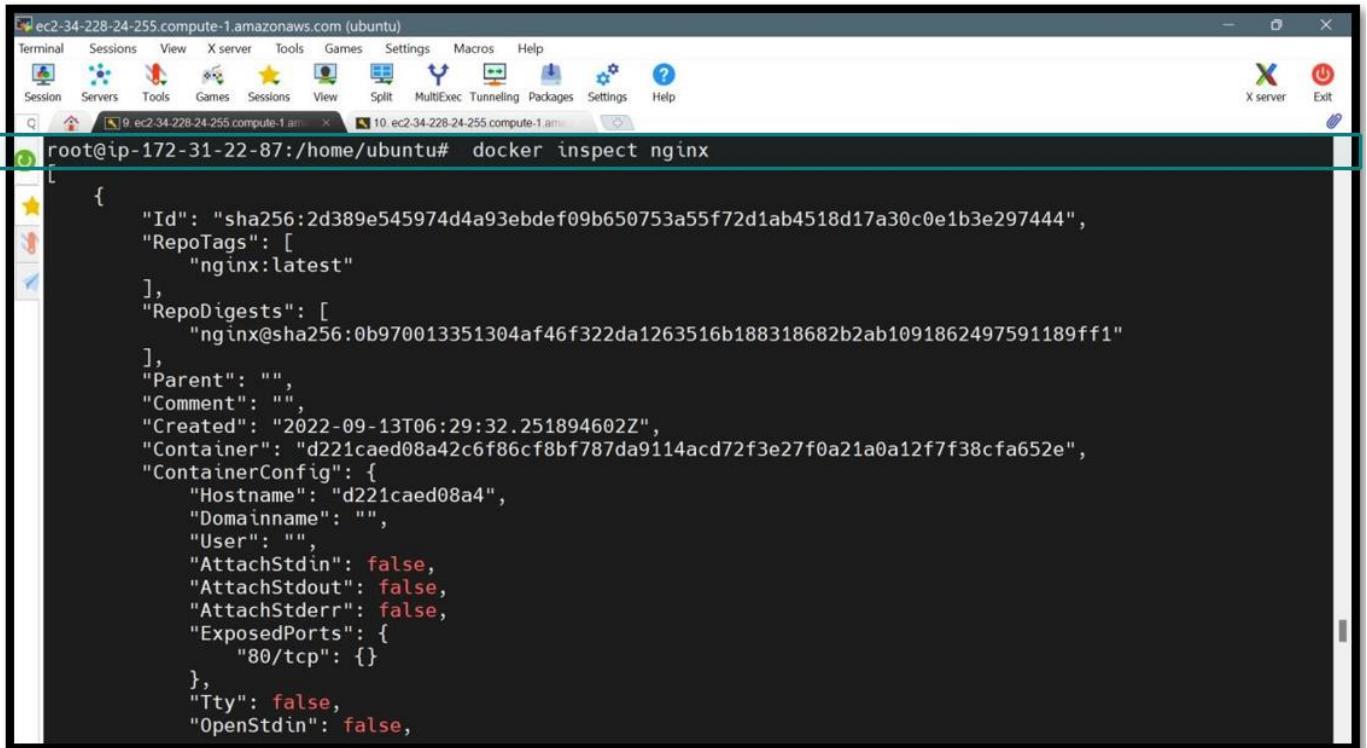
For checking the status of the container use command ‘docker ps -a’.



```
root@ip-172-31-22-87:/home/ubuntu# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
S NAMES
2ac2fb21b4e        e285                "bash"              26 minutes ago   Up 26 minutes
sad_stonebreaker
279ee355c666        nginx              "/docker-entrypoint..."  About an hour ago  Exited (137) 35 minutes ago
sayyed-webpage010
5f5efc562810       nginx              "/docker-entrypoint..."  2 hours ago      Exited (129) 2 hours ago
sayyed-webpage01
97a82d712d86       nginx              "/docker-entrypoint..."  2 hours ago      Exited (129) 2 hours ago
sayyed-webpage1
43870f16ce3b       nginx              "/docker-entrypoint..."  2 hours ago      Created
mhssce_45-webpage
f0998b57a8c2       nginx              "/docker-entrypoint..."  2 hours ago      Created
ayesha-webpage
94a18d0fcfa4f      hello-world        "/hello"            2 hours ago      Exited (0) 2 hours ago
vigilant_wiles
5e44660aa880       hello-world        "/hello"            2 hours ago      Exited (0) 2 hours ago
agitated_sanderson
da0dde4b40c9       nginx              "/docker-entrypoint..."  3 hours ago      Exited (129) 2 hours ago
ayesha_sayyed-webpage
14c5491d541d       hello-world        "/hello"            3 hours ago      Exited (0) 3 hours ago
charming_bell
d811126170a4       hello-world        "/hello"            3 hours ago      Exited (0) 3 hours ago
reverent_noxye
root@ip-172-31-22-87:/home/ubuntu#
```

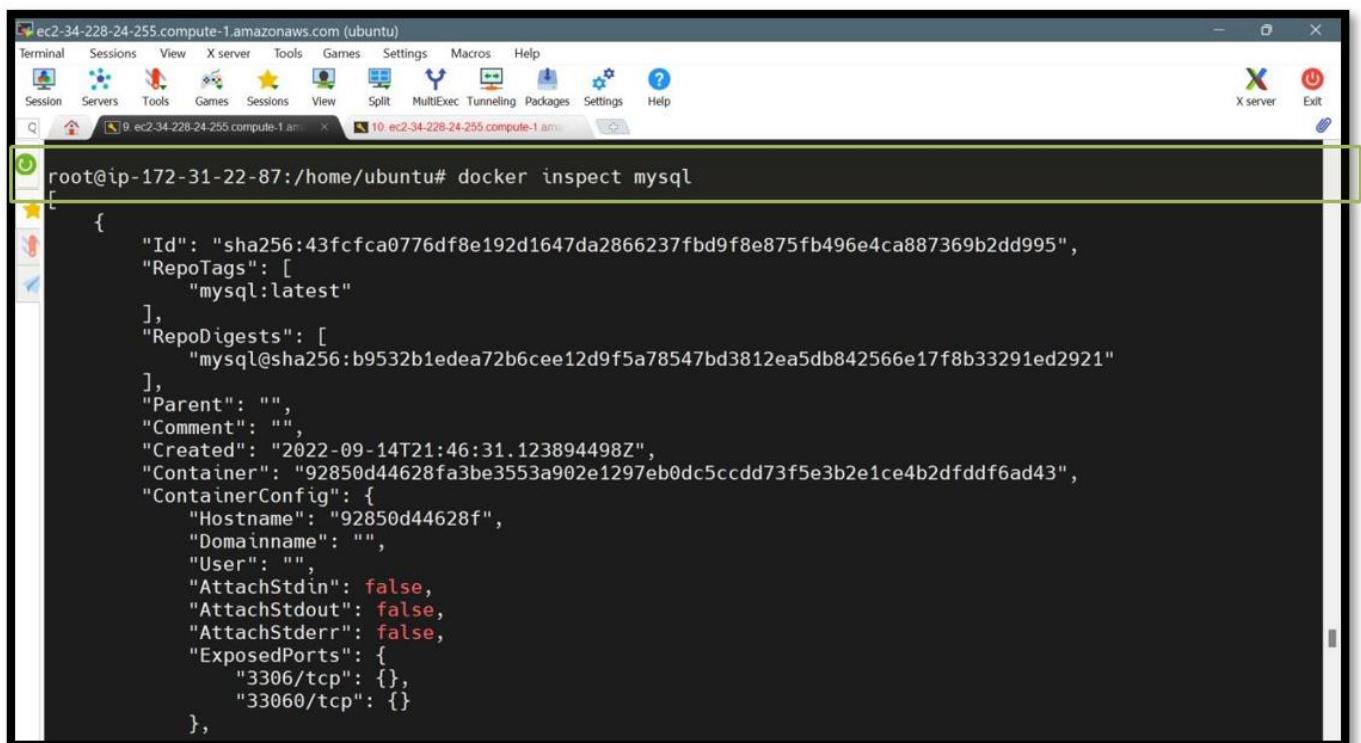
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 26: Run command ‘docker inspect nginx’ to inspect the nginx.



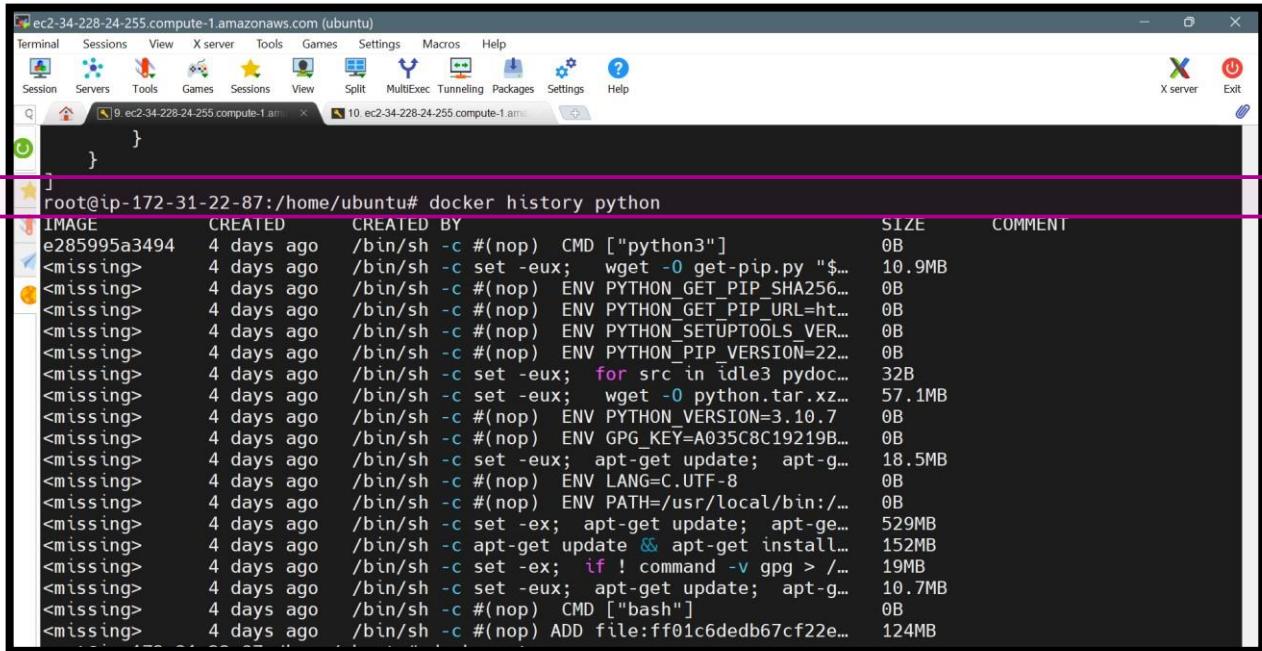
```
root@ip-172-31-22-87:/home/ubuntu# docker inspect nginx
[{"Id": "sha256:2d389e545974d4a93ebdef09b650753a55f72d1ab4518d17a30c0e1b3e297444",
 "RepoTags": [
     "nginx:latest"
 ],
 "RepoDigests": [
     "nginx@sha256:0b970013351304af46f322da1263516b188318682b2ab1091862497591189ff1"
 ],
 "Parent": "",
 "Comment": "",
 "Created": "2022-09-13T06:29:32.251894602Z",
 "Container": "d221caed08a42c6f86cf8bf787da9114acd72f3e27f0a21a0a12f7f38cfa652e",
 "ContainerConfig": {
     "Hostname": "d221caed08a4",
     "Domainname": "",
     "User": "",
     "AttachStdin": false,
     "AttachStdout": false,
     "AttachStderr": false,
     "ExposedPorts": {
         "80/tcp": {}
     },
     "Tty": false,
     "OpenStdin": false,
```

Now run ‘docker inspect mysql’ to inspect mysql.



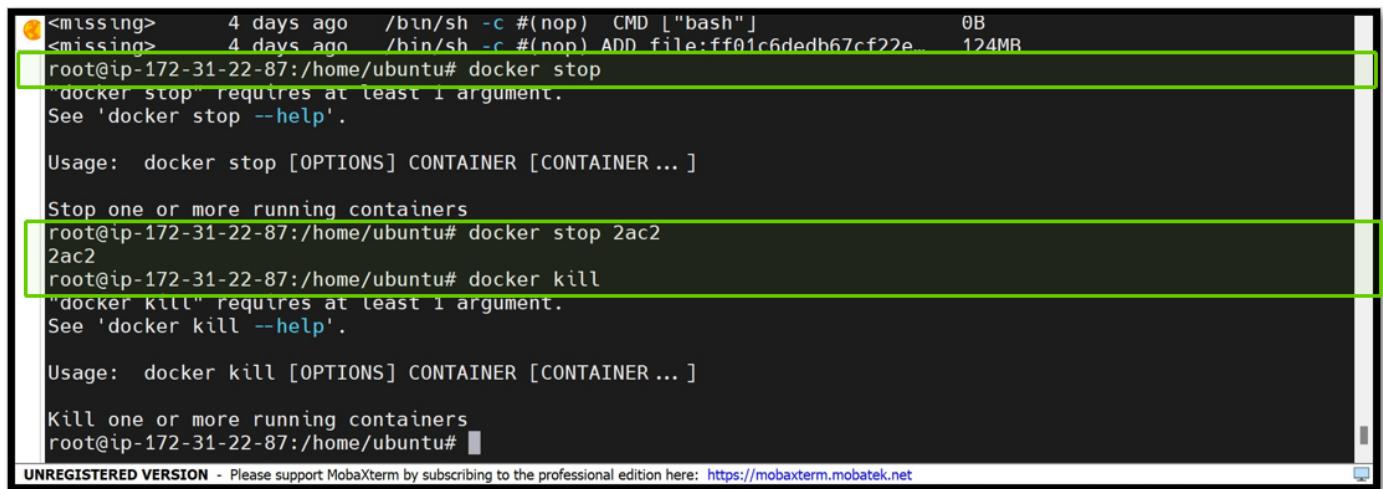
```
root@ip-172-31-22-87:/home/ubuntu# docker inspect mysql
[{"Id": "sha256:43fcfcfa0776df8e192d1647da2866237fb9f8e875fb496e4ca887369b2dd995",
 "RepoTags": [
     "mysql:latest"
 ],
 "RepoDigests": [
     "mysql@sha256:b9532b1edea72b6cee12d9f5a78547bd3812ea5db842566e17f8b33291ed2921"
 ],
 "Parent": "",
 "Comment": "",
 "Created": "2022-09-14T21:46:31.123894498Z",
 "Container": "92850d44628fa3be3553a902e1297eb0dc5ccdd73f5e3b2e1ce4b2dfddf6ad43",
 "ContainerConfig": {
     "Hostname": "92850d44628f",
     "Domainname": "",
     "User": "",
     "AttachStdin": false,
     "AttachStdout": false,
     "AttachStderr": false,
     "ExposedPorts": {
         "3306/tcp": {},
         "33060/tcp": {}
     },
```

Now run command ‘docker history python’ to get the docker python history



```
root@ip-172-31-22-87:/home/ubuntu# docker history python
IMAGE      CREATED     CREATED BY
e285995a3494  4 days ago  /bin/sh -c #(nop)  CMD ["python3"]
<missing>   4 days ago  /bin/sh -c set -eux; wget -O get-pip.py "$...
<missing>   4 days ago  /bin/sh -c #(nop) ENV PYTHON_GET_PIP_SHA256...
<missing>   4 days ago  /bin/sh -c #(nop) ENV PYTHON_GET_PIP_URL=ht...
<missing>   4 days ago  /bin/sh -c #(nop) ENV PYTHON_SETUPTOOLS_VER...
<missing>   4 days ago  /bin/sh -c #(nop) ENV PYTHON_PIP_VERSION=22...
<missing>   4 days ago  /bin/sh -c set -eux; for src in idle3 pydoc...
<missing>   4 days ago  /bin/sh -c set -eux; wget -O python.tar.xz...
<missing>   4 days ago  /bin/sh -c #(nop) ENV PYTHON_VERSION=3.10.7
<missing>   4 days ago  /bin/sh -c #(nop) ENV GPG_KEY=A035C8C19219B...
<missing>   4 days ago  /bin/sh -c set -eux; apt-get update; apt-g...
<missing>   4 days ago  /bin/sh -c #(nop) ENV LANG=C.UTF-8
<missing>   4 days ago  /bin/sh -c #(nop) ENV PATH=/usr/local/bin:/...
<missing>   4 days ago  /bin/sh -c set -ex; apt-get update; apt-ge...
<missing>   4 days ago  /bin/sh -c apt-get update & apt-get install...
<missing>   4 days ago  /bin/sh -c set -ex; if ! command -v gpg > /...
<missing>   4 days ago  /bin/sh -c set -eux; apt-get update; apt-g...
<missing>   4 days ago  /bin/sh -c #(nop) CMD ["bash"]
<missing>   4 days ago  /bin/sh -c #(nop) ADD file:ff01c6dedb67cf22e...
```

Step 27: Now stop the container by using command ‘docker stop [container id]’ → and simultaneously kill the docker by ‘docker kill’.



```
<missing>   4 days ago  /bin/sh -c #(nop)  CMD ["bash"]
<missing>   4 days ago  /bin/sh -c #(nop) ADD file:ff01c6dedb67cf22e...
root@ip-172-31-22-87:/home/ubuntu# docker stop
"docker stop" requires at least 1 argument.
See 'docker stop --help'.

Usage:  docker stop [OPTIONS] CONTAINER [CONTAINER ...]

Stop one or more running containers
root@ip-172-31-22-87:/home/ubuntu# docker stop 2ac2
2ac2
root@ip-172-31-22-87:/home/ubuntu# docker kill
"docker kill" requires at least 1 argument.
See 'docker kill --help'.

Usage:  docker kill [OPTIONS] CONTAINER [CONTAINER ...]

Kill one or more running containers
root@ip-172-31-22-87:/home/ubuntu#
```

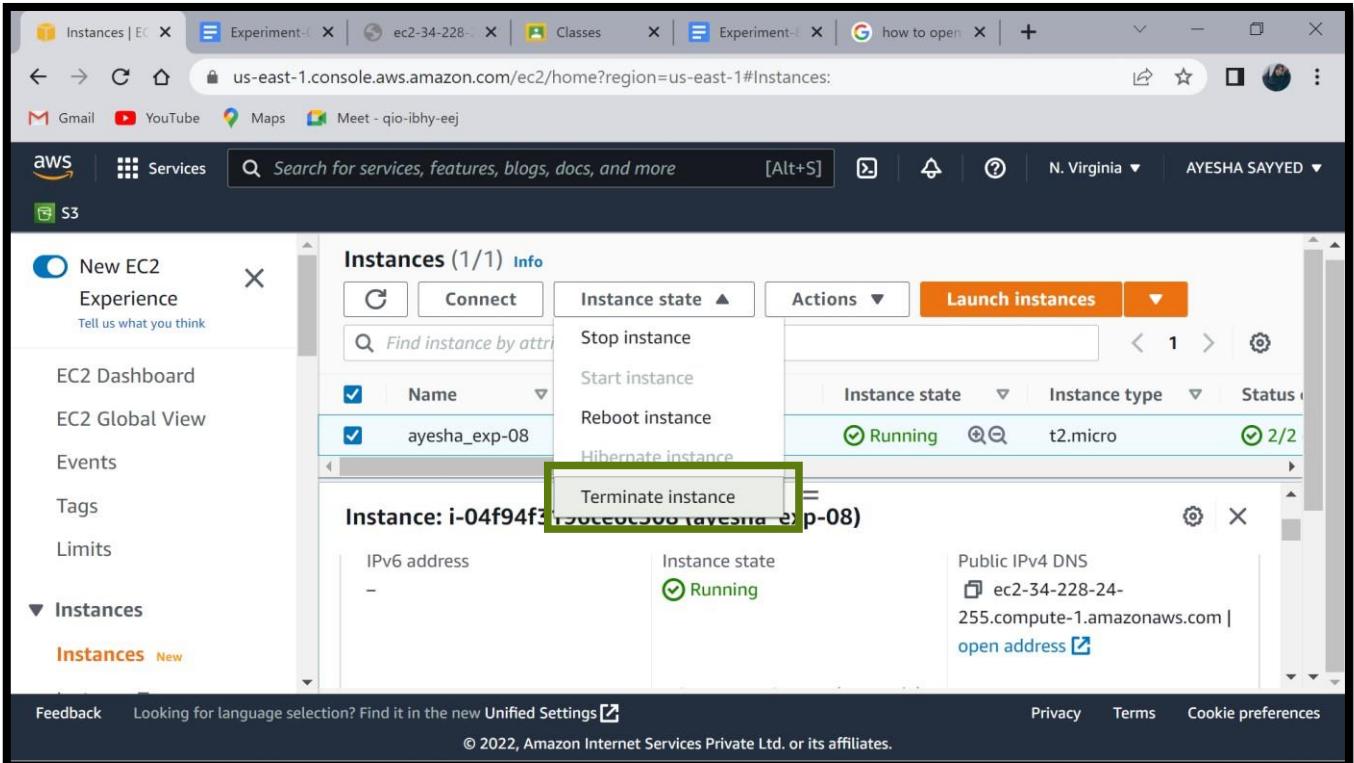
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Now exit the Session / Terminal.

Q6. Demonstrate any 15 docker command and explain its uses.

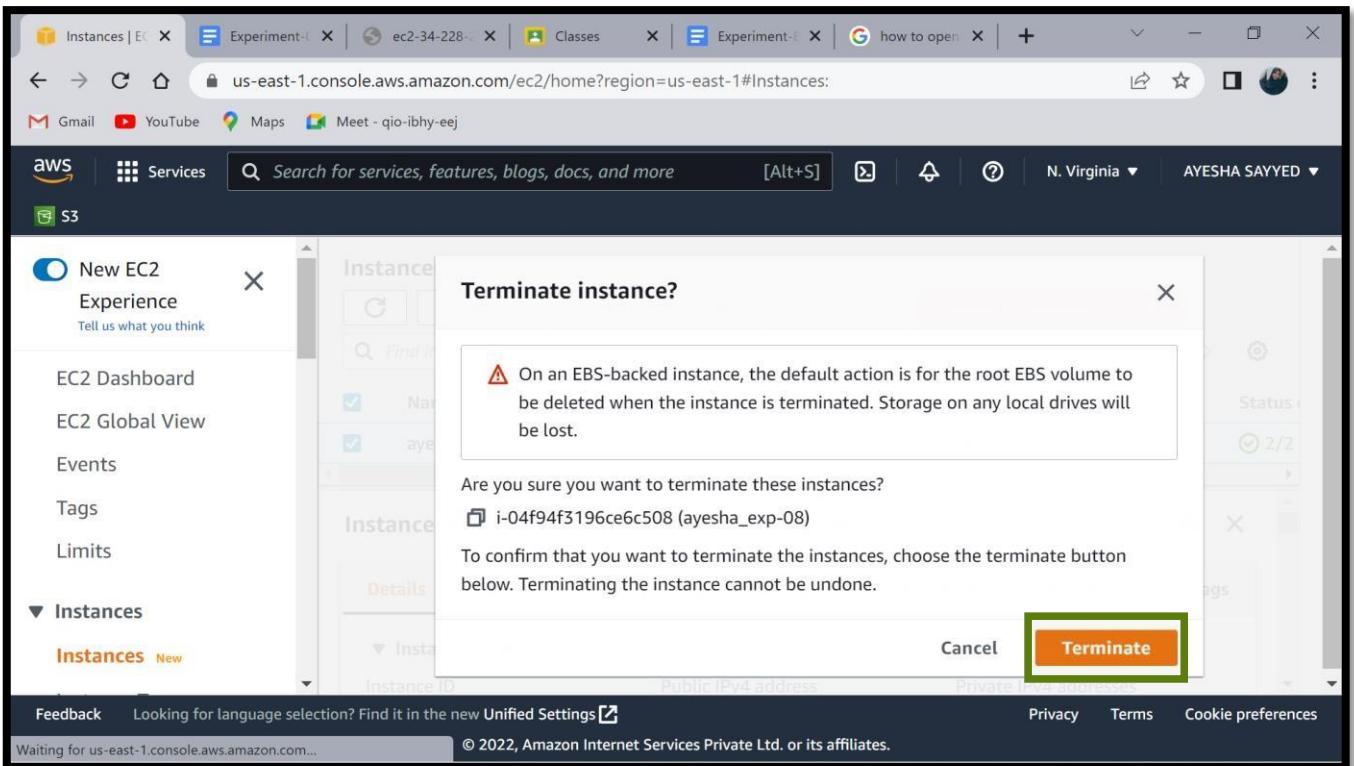
- We have demonstrated 15 docker commands and also explained it, but randomly [in steps]not one by one.

Step 27: After running all the codes, now Terminate the Instance in ES2.
Click on Terminate to ‘Terminate the Instance’.



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under Instances, 'Instances' is selected. In the main area, it says 'Instances (1/1) Info'. There's a table with one row for 'ayesha_exp-08'. The instance has a Public IPv4 DNS of 'ec2-34-228-24-255.compute-1.amazonaws.com'. The 'Terminate instance' button is highlighted with a red box.

Click on Terminate to ‘Terminate the Instance’.



The screenshot shows a 'Terminate instance?' dialog box. It contains a warning message: 'On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.' Below this, it asks 'Are you sure you want to terminate these instances?' with a checkbox next to 'i-04f94f3196ce6c508 (ayesha_exp-08)'. At the bottom, it says 'To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.' A large green 'Terminate' button is prominently displayed.

Instances Successfully Terminated!!

The screenshot shows a browser window with multiple tabs open. The active tab is 'Instances' on the AWS EC2 console at us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances. A prominent green notification bar at the top right states 'Successfully terminated i-04f94f3196ce6c508'. The main content area displays a table titled 'Instances (1/1)'. The table has columns for Name, Instance ID, Instance state, Instance type, and Status. One row is visible, labeled 'Instance: i-04f94f3196ce6c508 (ayesha_exp-08)'. Below the table, there are tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The 'Details' tab is selected. At the bottom of the page, there are links for Feedback, Privacy, Terms, and Cookie preferences, along with a copyright notice: '© 2022, Amazon Internet Services Private Ltd. or its affiliates.'

EXPERIMENT – 09

Q1. Install docker on AWS EC2 –Ubuntu by using curl.

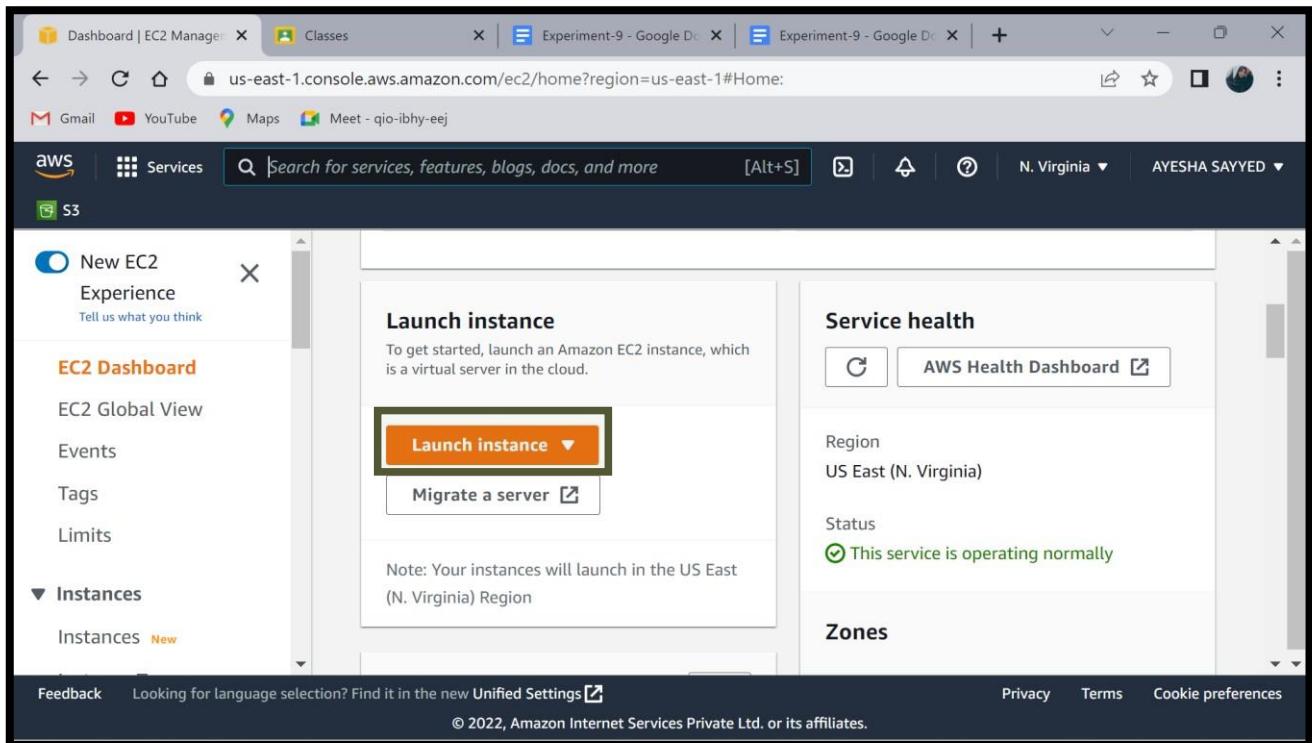
Step1: login to AWS Management Console Dashboard.

The screenshot shows the AWS Management Console Dashboard. The search bar at the top contains the text "Search for services, features, blogs, docs, and more". Below the search bar, there is a "Recently visited" section with icons and names for various AWS services: AWS Budgets, EC2, S3, AWS Billing Conductor, Elastic Beanstalk, Elastic Container Service, Lightsail, CloudWatch, and Cloud9. At the bottom of the dashboard, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

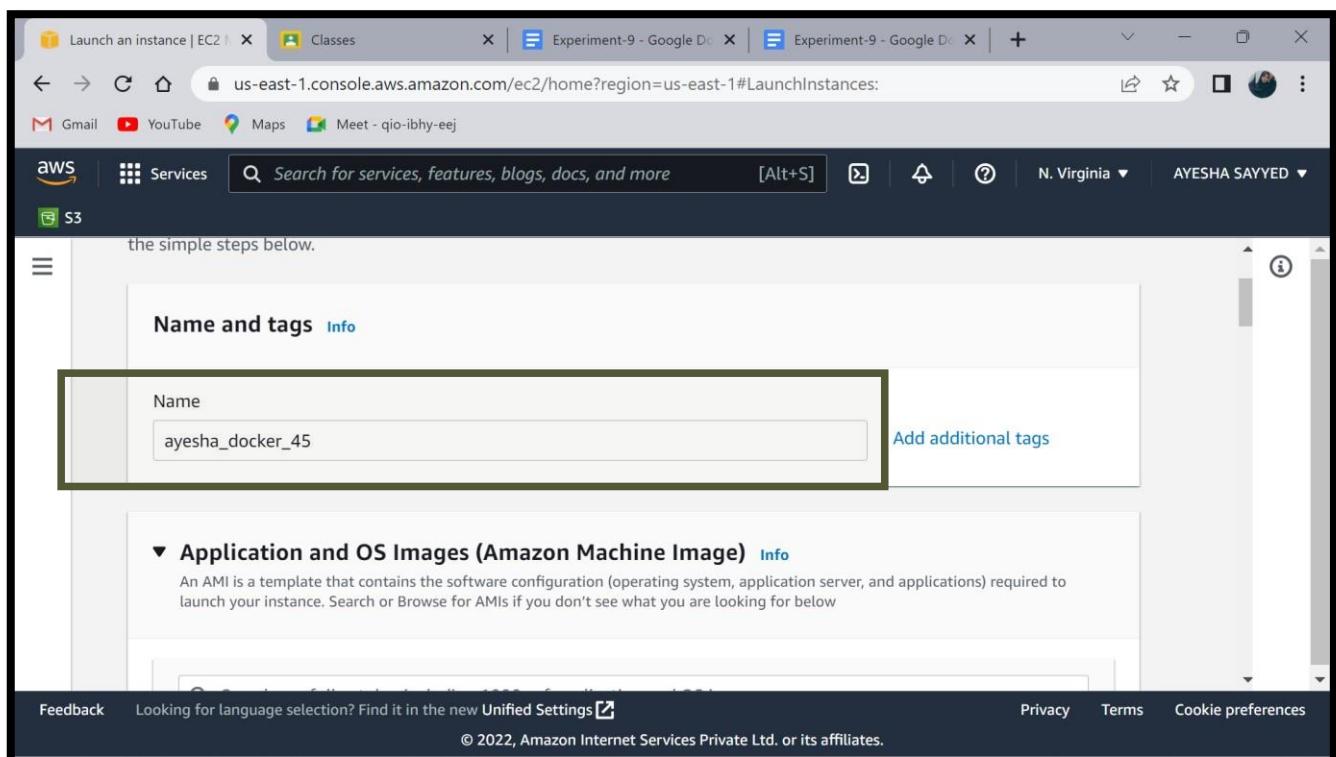
Step 2: Search for EC2 and select it.

The screenshot shows the AWS Management Console search results for "EC2". The search bar at the top has "EC2" typed into it. The results page displays a list of services under the heading "Services". The first result, "EC2", is highlighted with a red box. The description for EC2 is "Virtual Servers in the Cloud". Other results listed include "EC2 Image Builder" and "AWS Compute Optimizer". On the left side, there is a sidebar with links to "Services (8)", "Features (46)", "Blogs (1,802)", "Documentation (131,728)", "Knowledge Articles (30)", "Tutorials (19)", "Events (10)", "Marketplace (1,571)", and "Elastic Beanstalk". At the bottom of the page, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

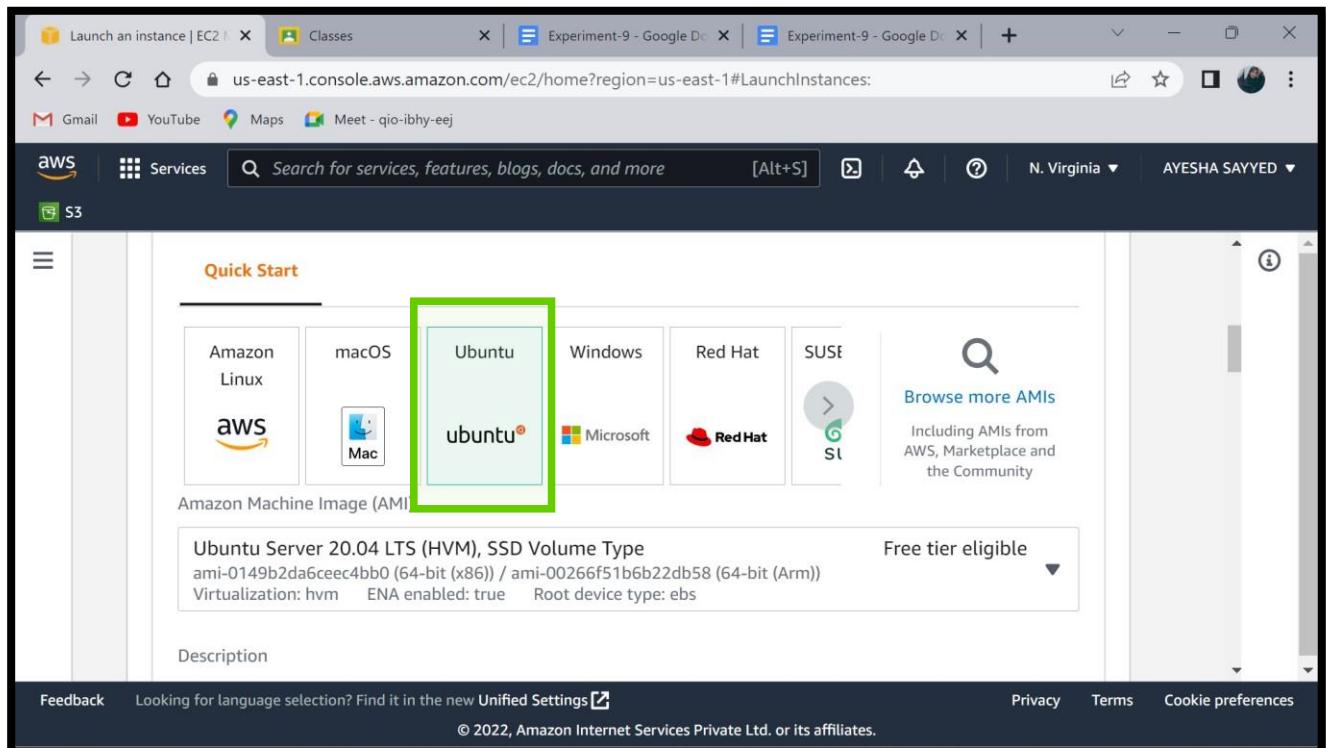
Step 3: Click on “Launch Instances”.



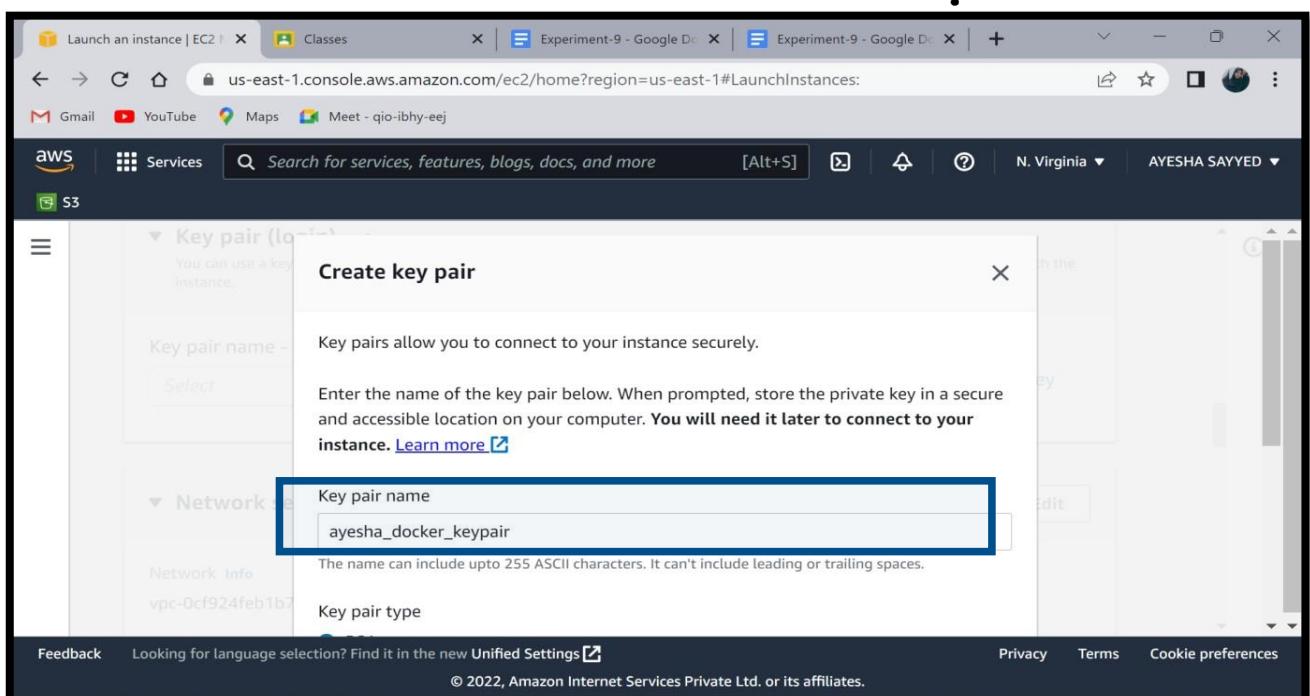
Step 4: Give any name to your “Instance”.



Step 5: Select UBUNTU and go with 20.04 LTS(HVM), SSD Volume Type.



Step 6: Create a KeyPair.



Allow all the Traffic from the Internet and click on Launch Instances. Click on View Instances.

The screenshot shows the AWS EC2 Firewall (security groups) configuration page. It displays three checkboxes for traffic rules:

- Allow SSH traffic from Anywhere (0.0.0.0/0)
- Allow HTTPS traffic from the internet
- Allow HTTP traffic from the internet

We'll create a new security group called 'launch-wizard-6' with the following rules:

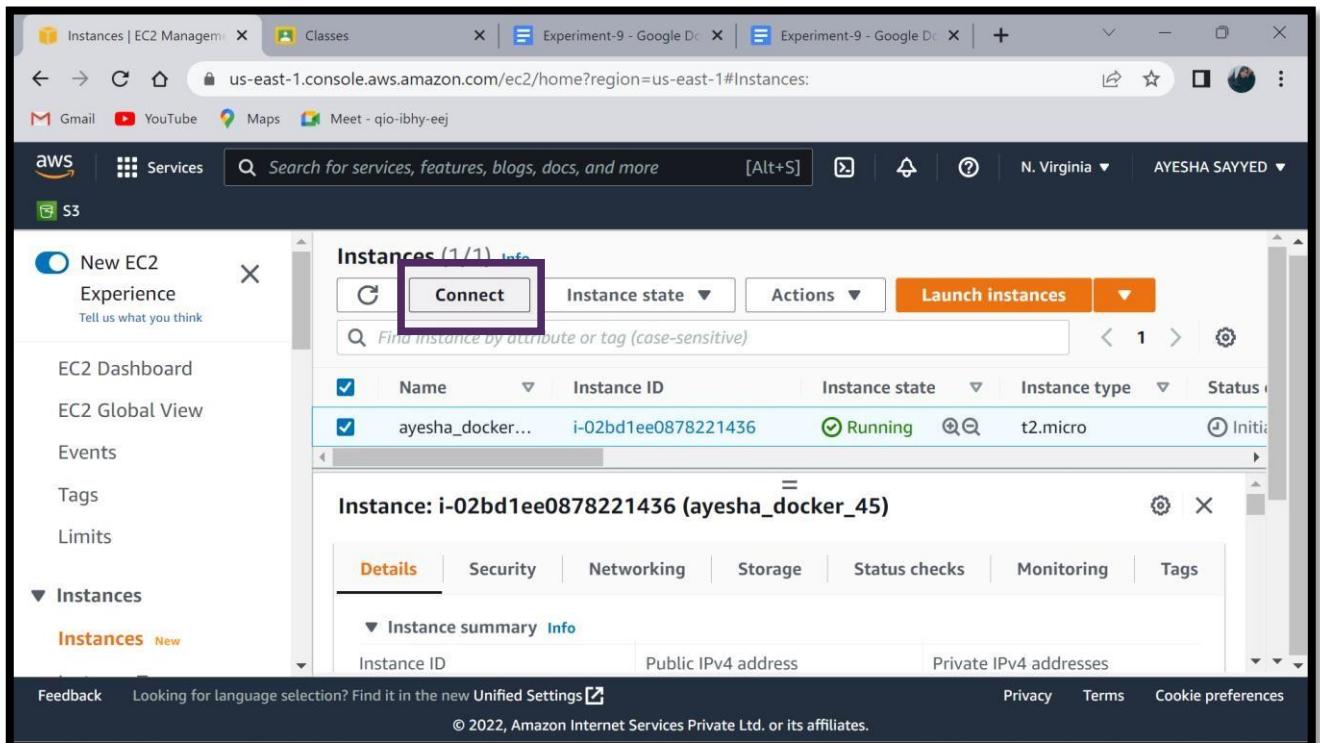
The screenshot shows the AWS EC2 Launch Instances summary page. It includes the following details:

- Number of instances: 1
- Instance Type: t2.micro
- Firewall (security group): New security group
- Action: **Launch instance**

The screenshot shows the AWS EC2 Get notified of estimated charges page. It includes the following sections:

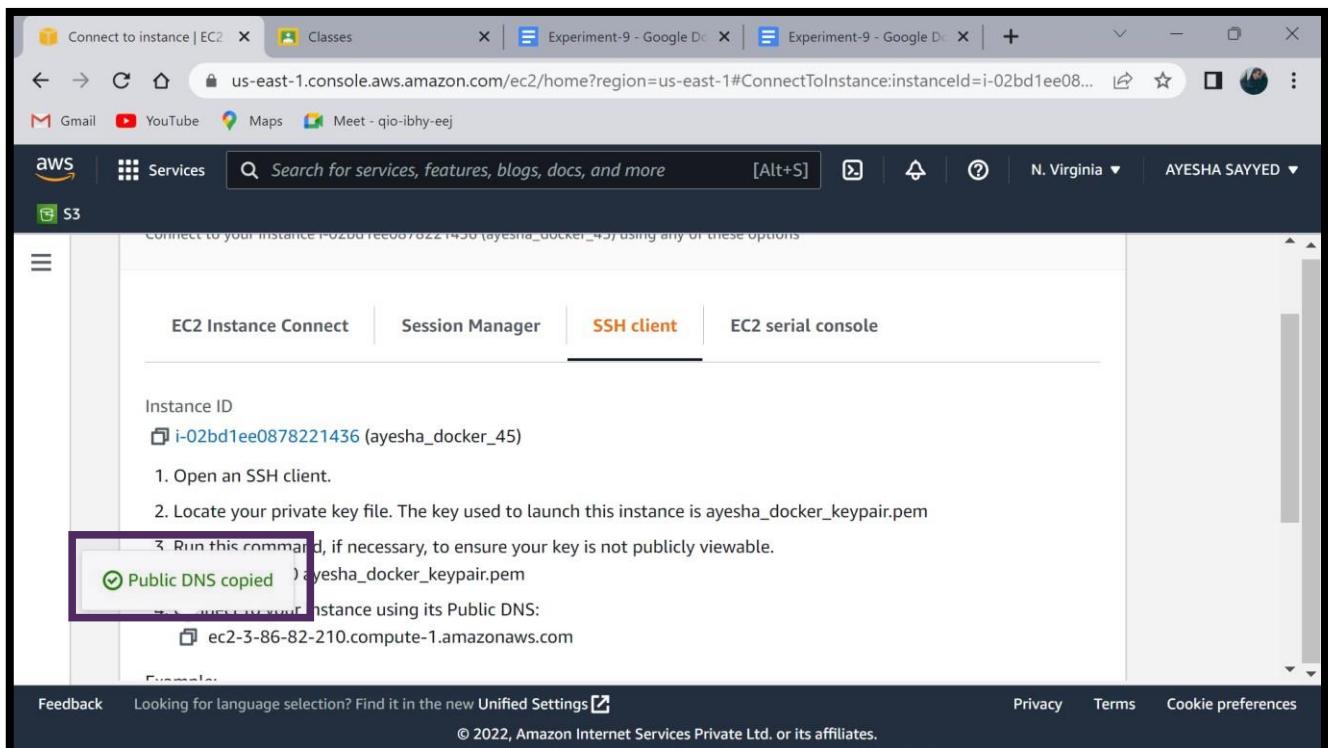
- Get notified of estimated charges**: Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).
- How to connect to your instance**: Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use. Click View Instances to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out [how to connect to your instance](#).
- [View more resources to get you started](#)
- View all instances**

Step 7: Connect to the Instance.



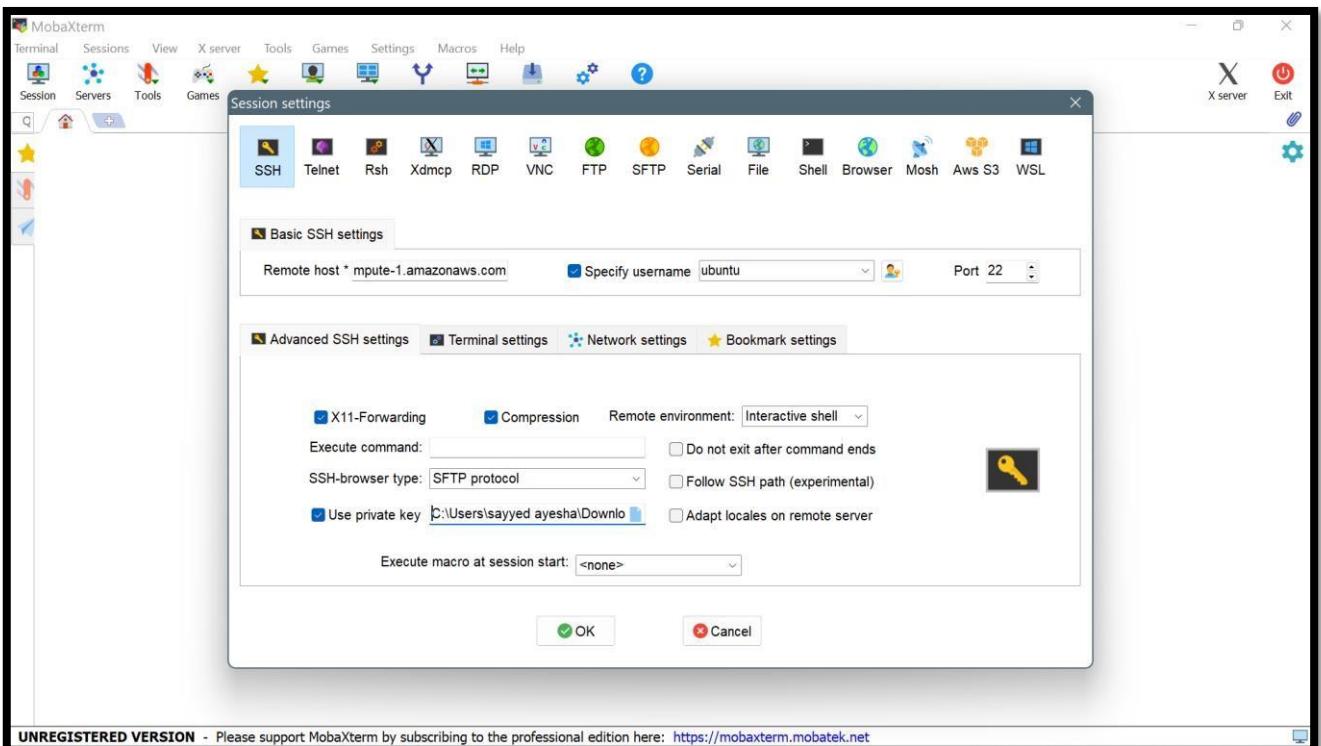
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under Instances, there's a link to 'Instances New'. The main area displays a table of instances. A single instance is selected, and its details are shown in a modal window. The instance ID is i-02bd1ee0878221436, the name is ayesha_docker_45, it's in the Running state, and the instance type is t2.micro. The 'Connect' button in the top bar of the modal is highlighted with a red box.

Step 8: Go to SSH client and copy Public DNS.



The screenshot shows the 'Connect to instance' page for the selected instance. At the top, it says 'CONNECT TO YOUR INSTANCE i-02bd1ee0878221436 (ayesha_docker_45) USING ANY OF THESE OPTIONS'. Below this, there are four tabs: 'EC2 Instance Connect', 'Session Manager', 'SSH client' (which is highlighted with a red box), and 'EC2 serial console'. The 'SSH client' tab has an 'Instance ID' field containing i-02bd1ee0878221436 (ayesha_docker_45). Below the instance ID, there are two main connection methods: 'Run this command' and 'Public DNS copied'. The 'Public DNS copied' option is highlighted with a red box. It also provides instructions to open an SSH client and locate the private key file ayesha_docker_keypair.pem. At the bottom, it shows the public DNS address ec2-3-86-82-210.compute-1.amazonaws.com.

Step 9: Now paste the Public DNS in MobaXterm for that Go to MobaXterm → Session → SSH → Adv SSH Setting.



Step 10: Your UBUNTU terminal is created, for Root access type “sudo su”.

```
Usage of /: 19.6% of 7.57GB Users logged in: 0
Memory usage: 22% IPv4 address for eth0: 172.31.84.104
Swap usage: 0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

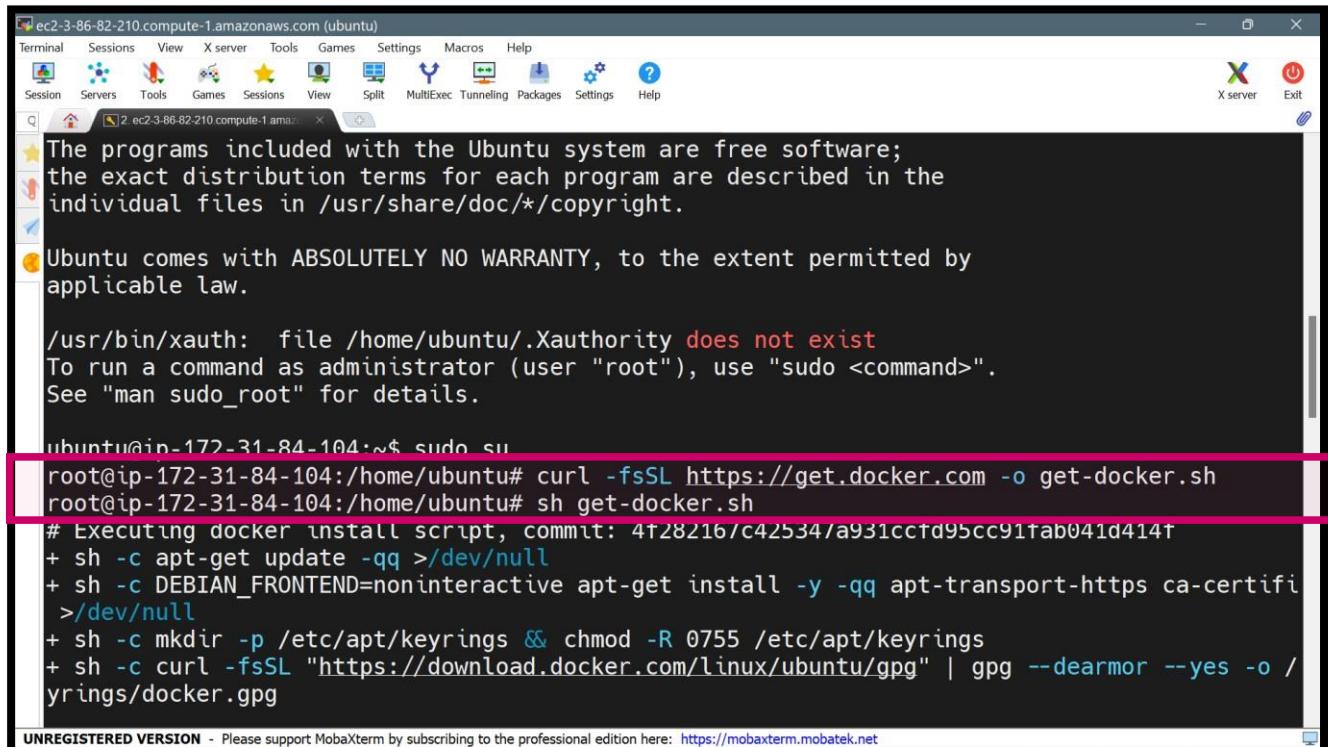
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-84-104:~$ sudo su
root@ip-172-31-84-104:/home/ubuntu#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 11: Now use the command “curl -fsSL https://get.docker.com -o get-docker.sh” & “sh get-docker.sh” to pull Docker.



```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

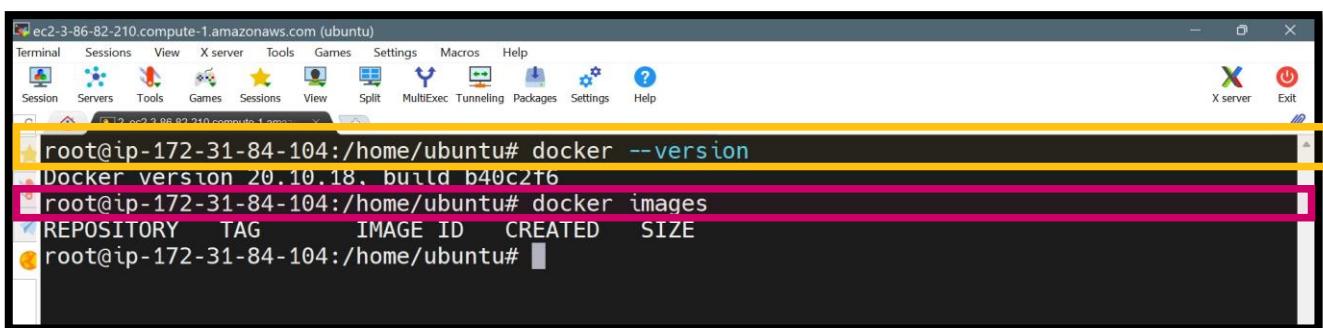
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth:  file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-84-104:~$ sudo su
root@ip-172-31-84-104:/home/ubuntu# curl -fsSL https://get.docker.com -o get-docker.sh
root@ip-172-31-84-104:/home/ubuntu# sh get-docker.sh
# Executing docker install script, commit: 4f28216/c42534/a931ccfd95cc91tab041d414t
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certifi
>/dev/null
+ sh -c mkdir -p /etc/apt/keyrings & chmod -R 0755 /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor --yes -o /
keyrings/docker.gpg
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

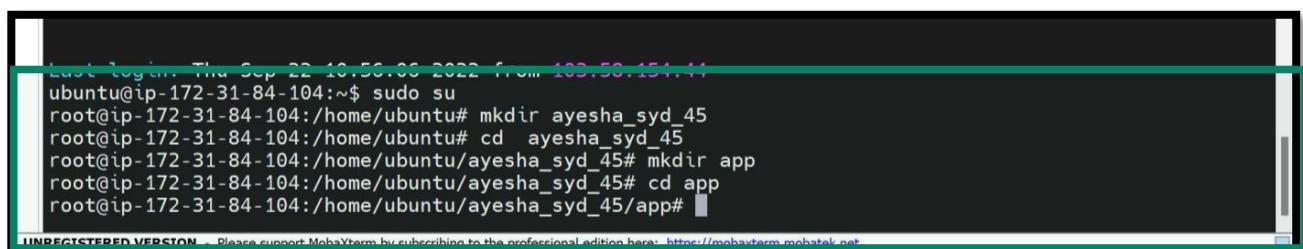
Q2. Run a Flask Application inside a Docker Container and explain the steps.

Step 12: Run command “docker --version” then “docker images”.



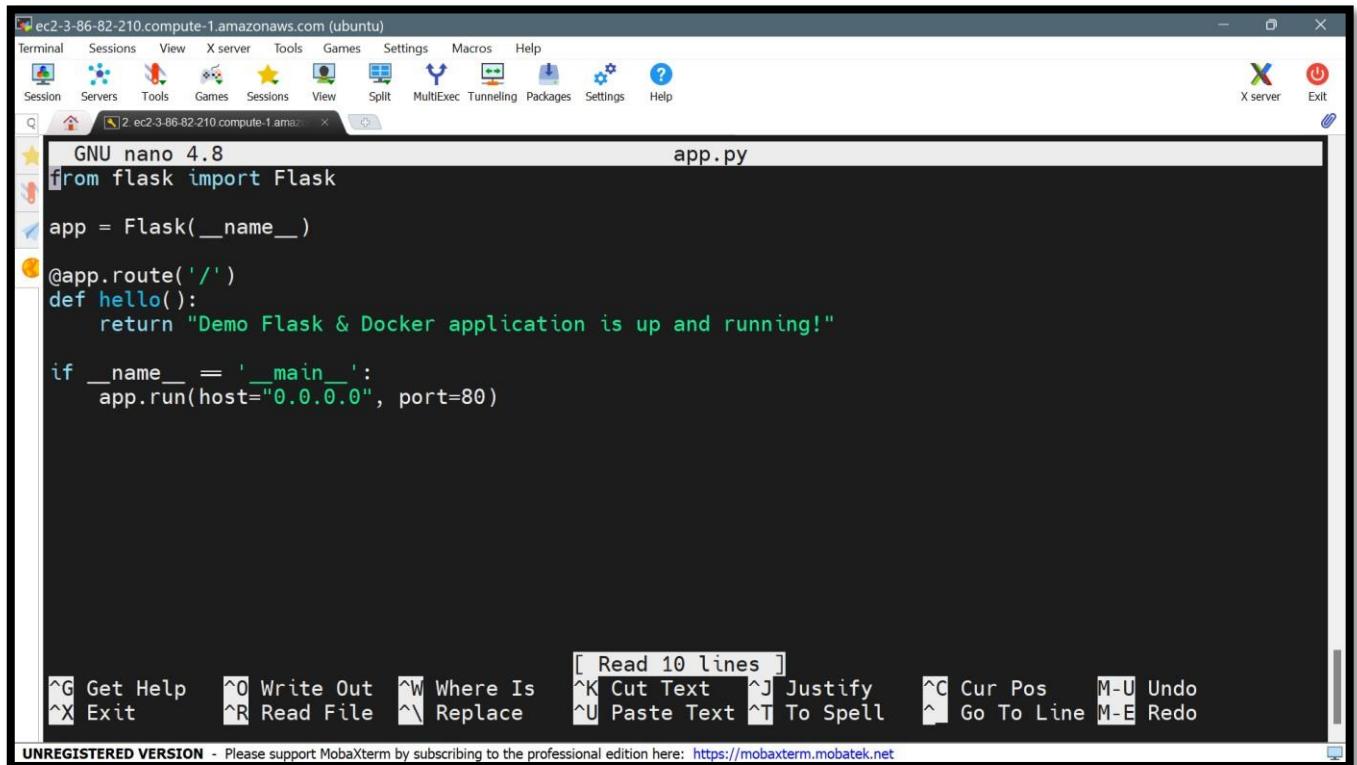
```
root@ip-172-31-84-104:/home/ubuntu# docker --version
Docker version 20.10.18, build b40c2f6
root@ip-172-31-84-104:/home/ubuntu# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@ip-172-31-84-104:/home/ubuntu#
```

Step 13: Now create a directory [your own name], go inside it and create another directory name as “APP” and go inside it too.



```
Last login: Thu Sep 22 10:50:00 2022 from 103.50.151.11
ubuntu@ip-172-31-84-104:~$ sudo su
root@ip-172-31-84-104:/home/ubuntu# mkdir ayesha_syd_45
root@ip-172-31-84-104:/home/ubuntu# cd ayesha_syd_45
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45# mkdir app
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45# cd app
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45/app#
```

Step12: Now open nano editor name by [nano app.py] and add the code of flask given in the classroom.



The screenshot shows a terminal window titled "ec2-3-86-82-210.compute-1.amazonaws.com (ubuntu)". Inside the terminal, a nano editor window is open with the file "app.py". The code in the file is:

```
GNU nano 4.8
from flask import Flask

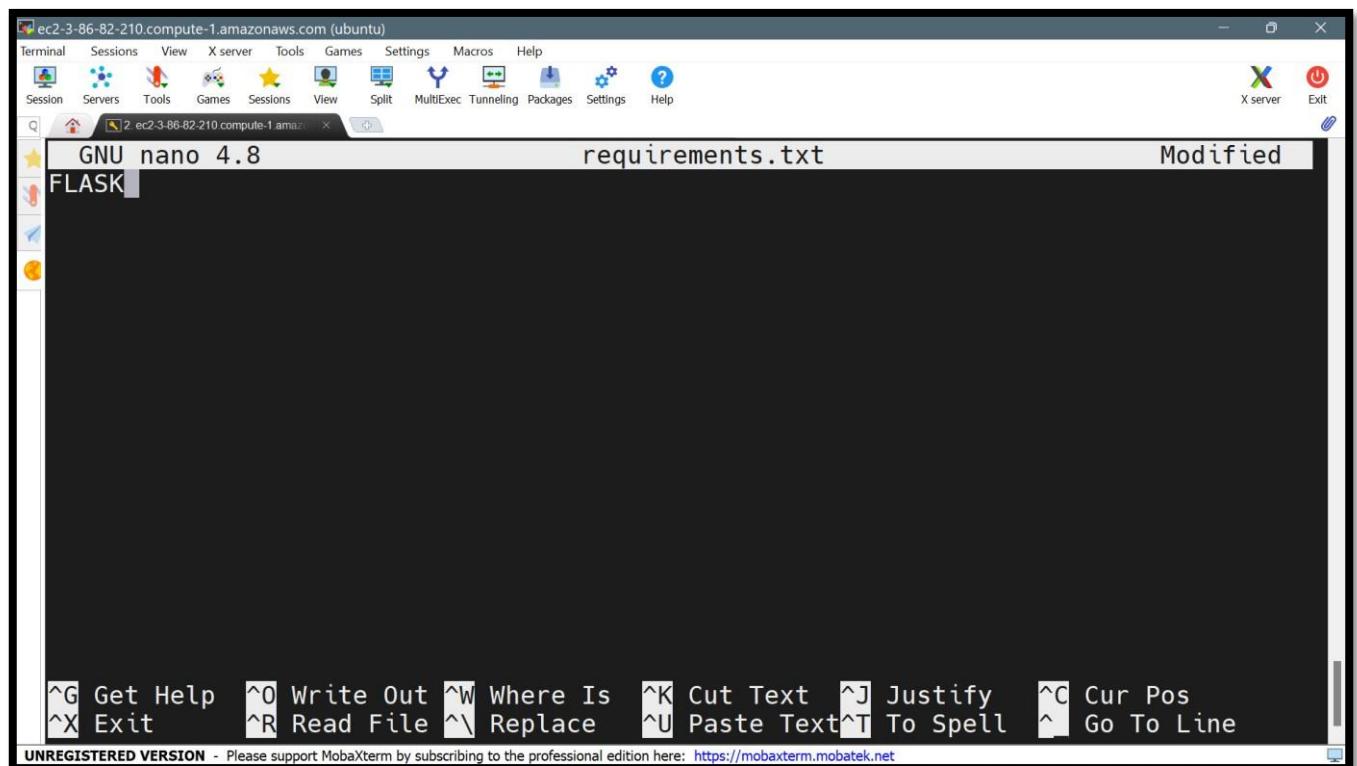
app = Flask(__name__)

@app.route('/')
def hello():
    return "Demo Flask & Docker application is up and running!"

if __name__ == '__main__':
    app.run(host="0.0.0.0", port=80)
```

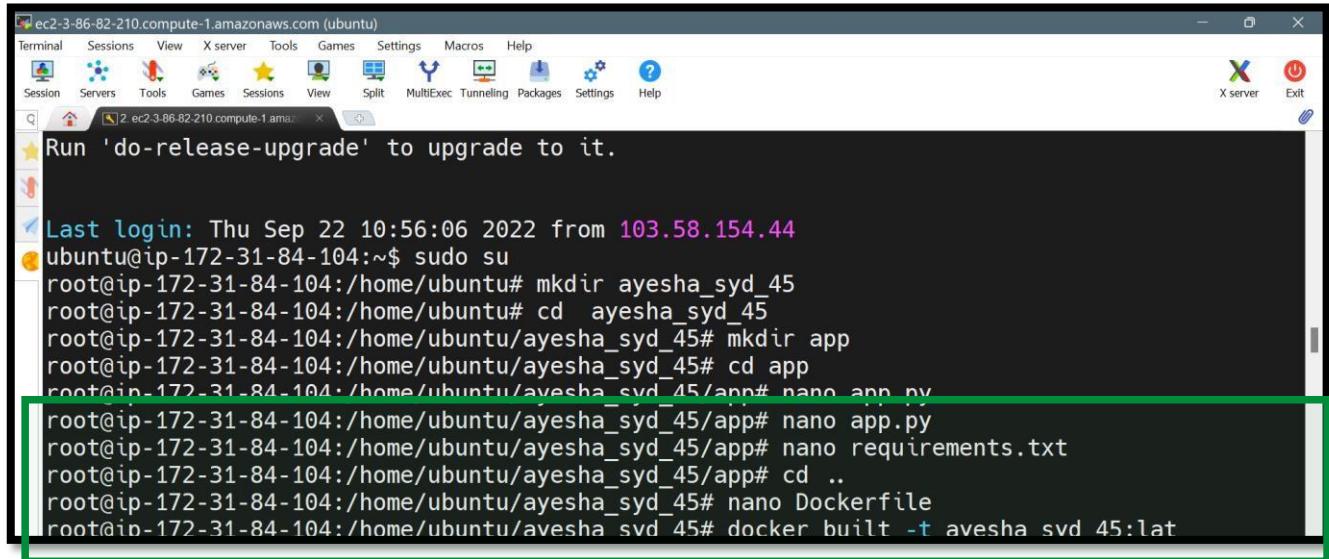
The nano editor interface includes a menu bar with "Terminal", "Sessions", "View", "X server", "Tools", "Games", "Settings", "Macros", and "Help". Below the menu is a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. On the right side of the terminal window, there are "X server" and "Exit" buttons. At the bottom of the terminal window, there is a status bar with the text "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net".

Step 13: Now open nano editor named by [requirements.txt] and write the requirements of the file to be installed on other PC.



The screenshot shows a terminal window titled "ec2-3-86-82-210.compute-1.amazonaws.com (ubuntu)". Inside the terminal, a nano editor window is open with the file "requirements.txt". The file contains the word "FLASK". The nano editor interface is identical to the previous screenshot, with a menu bar, toolbar, and status bar at the bottom.

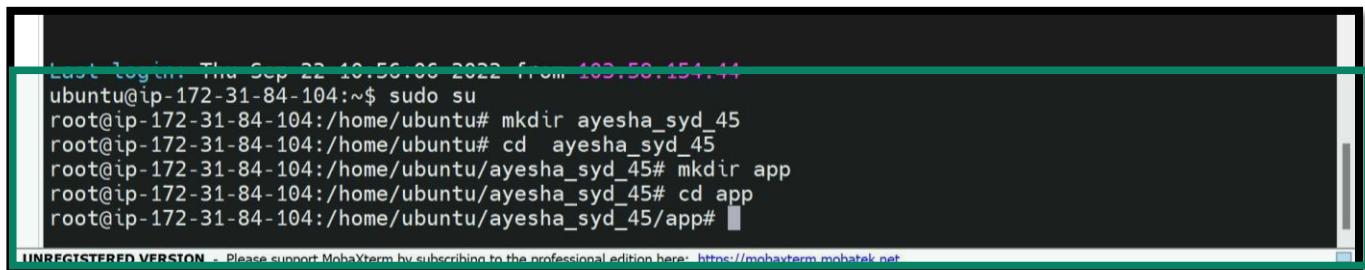
Step 14: Now go back to your Parent Directory and write [nano Dockerfile] and copy the code give in the classroom.



```
Run 'do-release-upgrade' to upgrade to it.

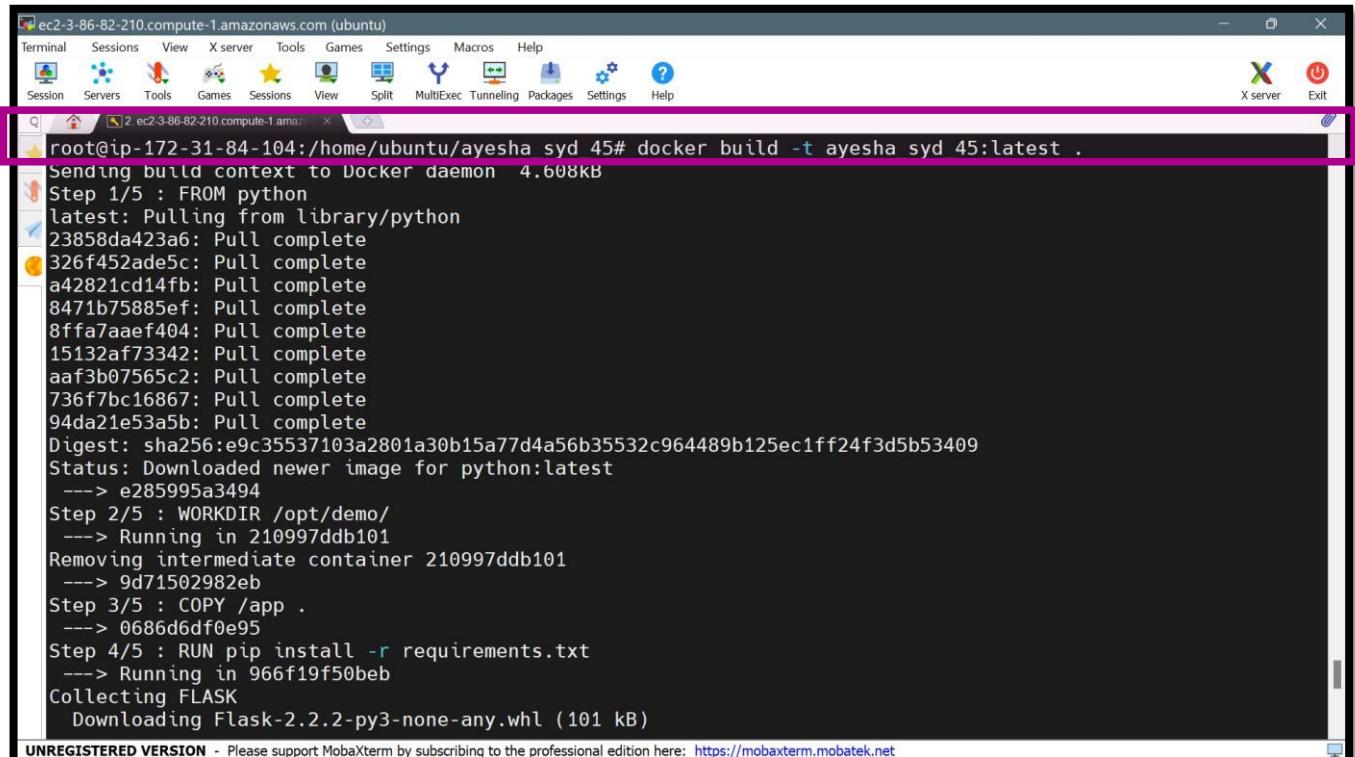
Last login: Thu Sep 22 10:56:06 2022 from 103.58.154.44
ubuntu@ip-172-31-84-104:~$ sudo su
root@ip-172-31-84-104:/home/ubuntu# mkdir ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu# cd ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# mkdir app
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# cd app
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45/app# nano app.py
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45/app# nano requirements.txt
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45/app# cd ..
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# nano Dockerfile
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# docker built -t ayesha_svd_45:lat
```

Write code in Dockerfile.

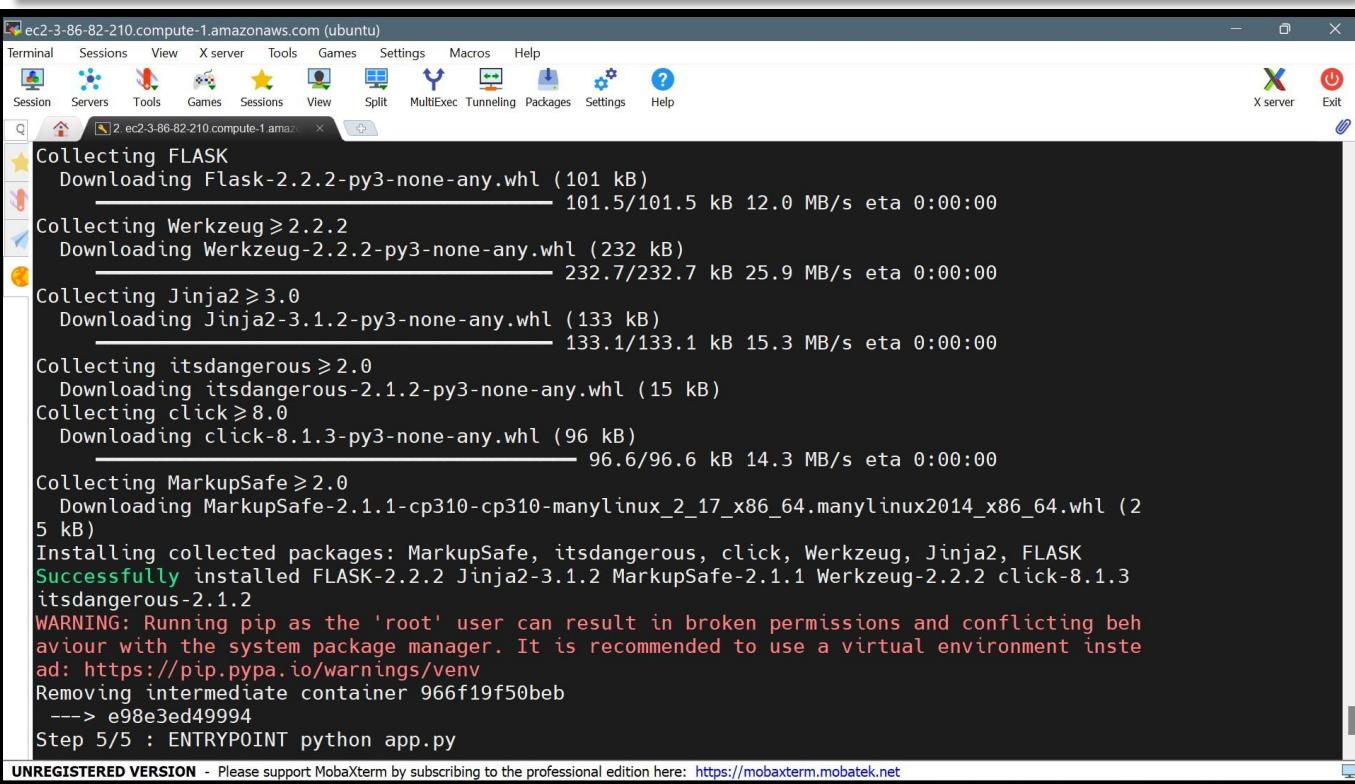


```
Last login: Thu Sep 22 10:56:06 2022 from 103.58.154.44
ubuntu@ip-172-31-84-104:~$ sudo su
root@ip-172-31-84-104:/home/ubuntu# mkdir ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu# cd ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# mkdir app
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# cd app
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45/app#
```

Step 15: Now the command “docker build -t [file name]:latest .”

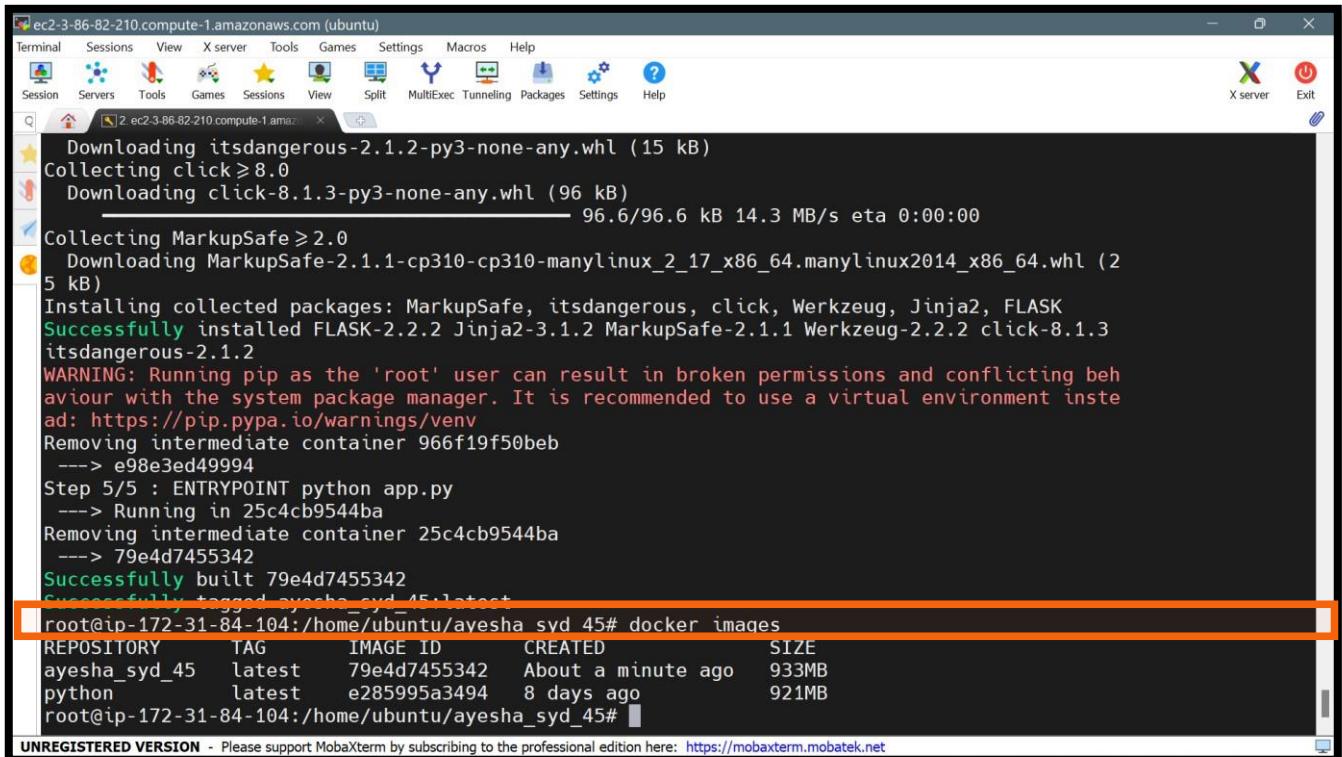


```
ec2-3-86-82-210.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
root@ip-172-31-84-104:/home/ubuntu/ayesha syd 45# docker build -t ayesha syd 45:latest .
Sending build context to Docker daemon 4.608KB
Step 1/5 : FROM python
latest: Pulling from library/python
23858da423a6: Pull complete
326f452ade5c: Pull complete
a42821cd14fb: Pull complete
8471b75885ef: Pull complete
8ffa7aaef404: Pull complete
15132af73342: Pull complete
aaf3b07565c2: Pull complete
736f7bc16867: Pull complete
94da21e53a5b: Pull complete
Digest: sha256:e9c35537103a2801a30b15a77d4a56b35532c964489b125ec1ff24f3d5b53409
Status: Downloaded newer image for python:latest
--> e285995a3494
Step 2/5 : WORKDIR /opt/demo/
--> Running in 210997ddb101
Removing intermediate container 210997ddb101
--> 9d71502982eb
Step 3/5 : COPY ./app .
--> 0686d6df0e95
Step 4/5 : RUN pip install -r requirements.txt
--> Running in 966f19f50beb
Collecting FLASK
  Downloading Flask-2.2.2-py3-none-any.whl (101 kB)
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```



```
ec2-3-86-82-210.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Collecting FLASK
  Downloading Flask-2.2.2-py3-none-any.whl (101 kB) 101.5/101.5 kB 12.0 MB/s eta 0:00:00
Collecting Werkzeug≥2.2.2
  Downloading Werkzeug-2.2.2-py3-none-any.whl (232 kB) 232.7/232.7 kB 25.9 MB/s eta 0:00:00
Collecting Jinja2≥3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB) 133.1/133.1 kB 15.3 MB/s eta 0:00:00
Collecting itsdangerous≥2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click≥8.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB) 96.6/96.6 kB 14.3 MB/s eta 0:00:00
Collecting MarkupSafe≥2.0
  Downloading MarkupSafe-2.1.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, FLASK
Successfully installed FLASK-2.2.2 Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 itsdangerous-2.1.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Removing intermediate container 966f19f50beb
--> e98e3ed49994
Step 5/5 : ENTRYPOINT python app.py
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

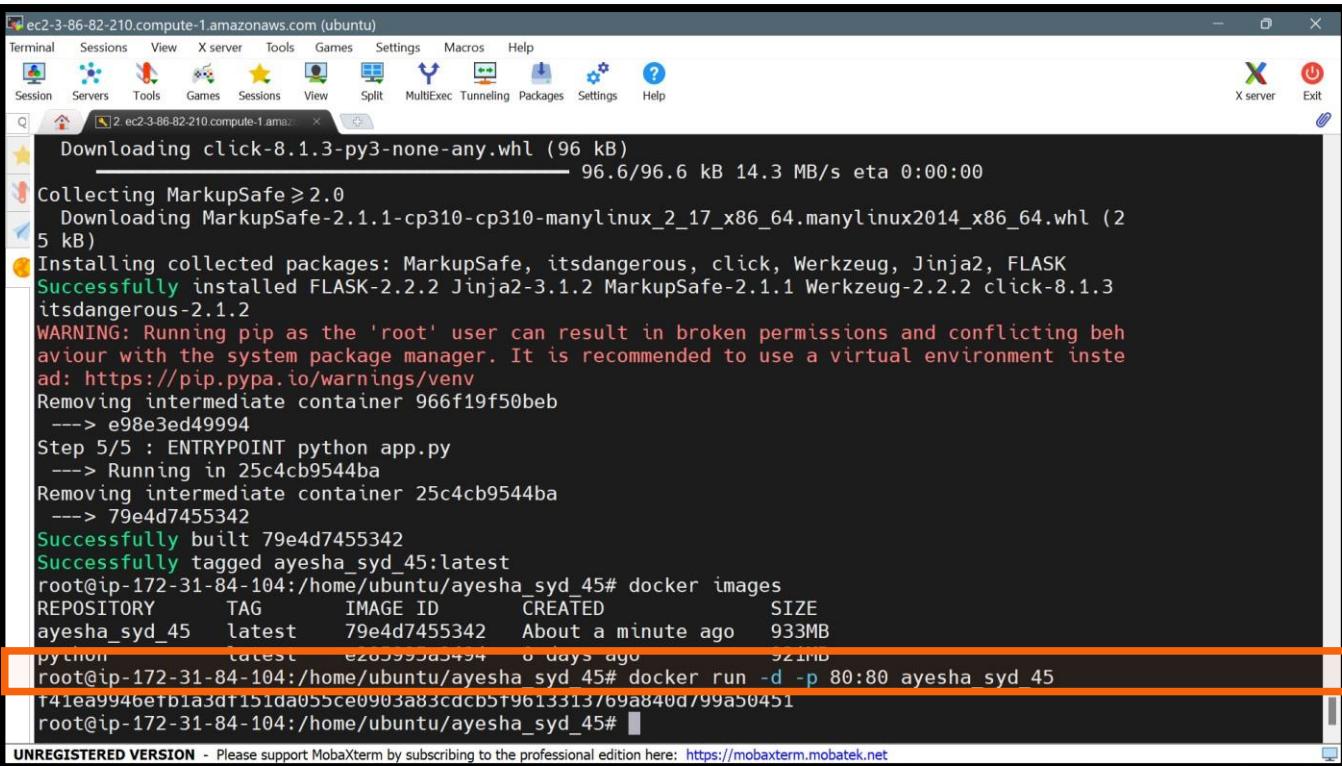
Step 16: Now for checking the images run command as “docker images”.



```
ec2-3-86-82-210.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
[2 ec2-3-86-82-210.compute-1.amazonaws.com] Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click≥8.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB) 96.6/96.6 kB 14.3 MB/s eta 0:00:00
Collecting MarkupSafe≥2.0
  Downloading MarkupSafe-2.1.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, FLASK
Successfully installed FLASK-2.2.2 Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 itsdangerous-2.1.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Removing intermediate container 966f19f50beb
--> e98e3ed49994
Step 5/5 : ENTRYPPOINT python app.py
--> Running in 25c4cb9544ba
Removing intermediate container 25c4cb9544ba
--> 79e4d7455342
Successfully built 79e4d7455342
Successfully tagged ayesha_syd_45:latest
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45# docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ayesha_syd_45  latest        79e4d7455342  About a minute ago  933MB
python          latest        e285995a3494   8 days ago    921MB
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 17: Run command “docker run -d -p 80:80 [directory name]”.



```
ec2-3-86-82-210.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
[2 ec2-3-86-82-210.compute-1.amazonaws.com] Downloading click-8.1.3-py3-none-any.whl (96 kB) 96.6/96.6 kB 14.3 MB/s eta 0:00:00
Collecting MarkupSafe≥2.0
  Downloading MarkupSafe-2.1.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, FLASK
Successfully installed FLASK-2.2.2 Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 itsdangerous-2.1.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Removing intermediate container 966f19f50beb
--> e98e3ed49994
Step 5/5 : ENTRYPPOINT python app.py
--> Running in 25c4cb9544ba
Removing intermediate container 25c4cb9544ba
--> 79e4d7455342
Successfully built 79e4d7455342
Successfully tagged ayesha_syd_45:latest
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45# docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ayesha_syd_45  latest        79e4d7455342  About a minute ago  933MB
python          latest        e285995a3494   8 days ago    921MB
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45# docker run -d -p 80:80 ayesha_syd_45
t41ea994be1b1a3df151da055ce0903a83cdcb5f9613313/69a840d/99a50451
root@ip-172-31-84-104:/home/ubuntu/ayesha_syd_45#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

To run the above command, copy the Public IPv4 DNS.

The screenshot shows the AWS EC2 Instances details page for an instance named 'ayesha_docker_45'. The Public IPv4 address listed is 3.86.82.210. A green box highlights the message 'Public IPv4 DNS copied' next to the public IP address. The browser's address bar at the top shows the URL: us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-02bd1ee0878221436.

Paste it in **the Address Bar**

The screenshot shows a web browser displaying the URL ec2-3-86-82-210.compute-1.amazonaws.com. The page content reads 'Demo Flask & Docker application is up and running!'. A yellow circle highlights this text. The browser's address bar also displays the same URL.

Step 18: After performing now terminate the instance.

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with options like New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under Instances, there's a link to Instances. The main area shows a table titled 'Instances (1/1)'. A single instance is listed: 'ayesha_docker...' with an 'Instance state' of 'Running', 'Instance type' of 't2.micro', and a status of '2/2'. Below the table, a callout box highlights the 'Terminate instance' button. At the bottom of the page, there's a footer with links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

Click on Terminate, Instance will be Terminated.

The screenshot shows the 'Terminate instance?' dialog box. It contains a warning message: 'On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.' Below the message, it asks 'Are you sure you want to terminate these instances?' with a checkbox next to the instance ID 'i-02bd1ee0878221436 (ayesha_docker_45)'. A note below says 'To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.' At the bottom right of the dialog, there are 'Cancel' and 'Terminate' buttons, with 'Terminate' being highlighted by a green box.

Successfully Instance Terminated!!

The screenshot shows the AWS EC2 Management Console interface. A green success message at the top center reads "Successfully terminated i-02bd1ee0878221436". Below this, the "Instances (1/1) Info" section displays a single row of data:

Name	Instance ID	Instance state	Instance type	Status
ayesha_docker...	i-02bd1ee0878221436	Shutting-down	t2.micro	2/2

The "Actions" dropdown menu is open next to the instance row. At the bottom of the page, a modal window titled "Instance: i-02bd1ee0878221436 (ayesha_docker_45)" is displayed.

Simultaneously, delete the Keypair too.

The screenshot shows the AWS EC2 Management Console interface. A modal window titled "ayesha_docker_keypair could be associated with one or more instances." is displayed. Inside the modal, a confirmation message says "Delete ayesha_docker_keypair" and "To confirm deletion, type Delete in the field". A text input field contains the word "delete". At the bottom right of the modal are "Cancel" and "Delete" buttons.

Q3. What is Docker file? Explain all lines of your Docker file.

Docker can build images automatically by reading the instructions from a Docker file. A Docker file is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

The all lines of your Docker file are explained as:

- FROM python // It specifies the Base Image.
- WORKDIR /opt/demo/ // It sets the Working Directory for Instruction to follow.
- COPY /flaskProject . // It copy all files from flaskProject directory to current directory.
- RUN pip install -r requirements.txt // It will Install all the dependencies from requirements.txt file.
- ENTRYPOINT python app.py // It will run the app.py in our container.

EXPERIMENT – 11

Q1. What is AWS Elastic Beanstalk?

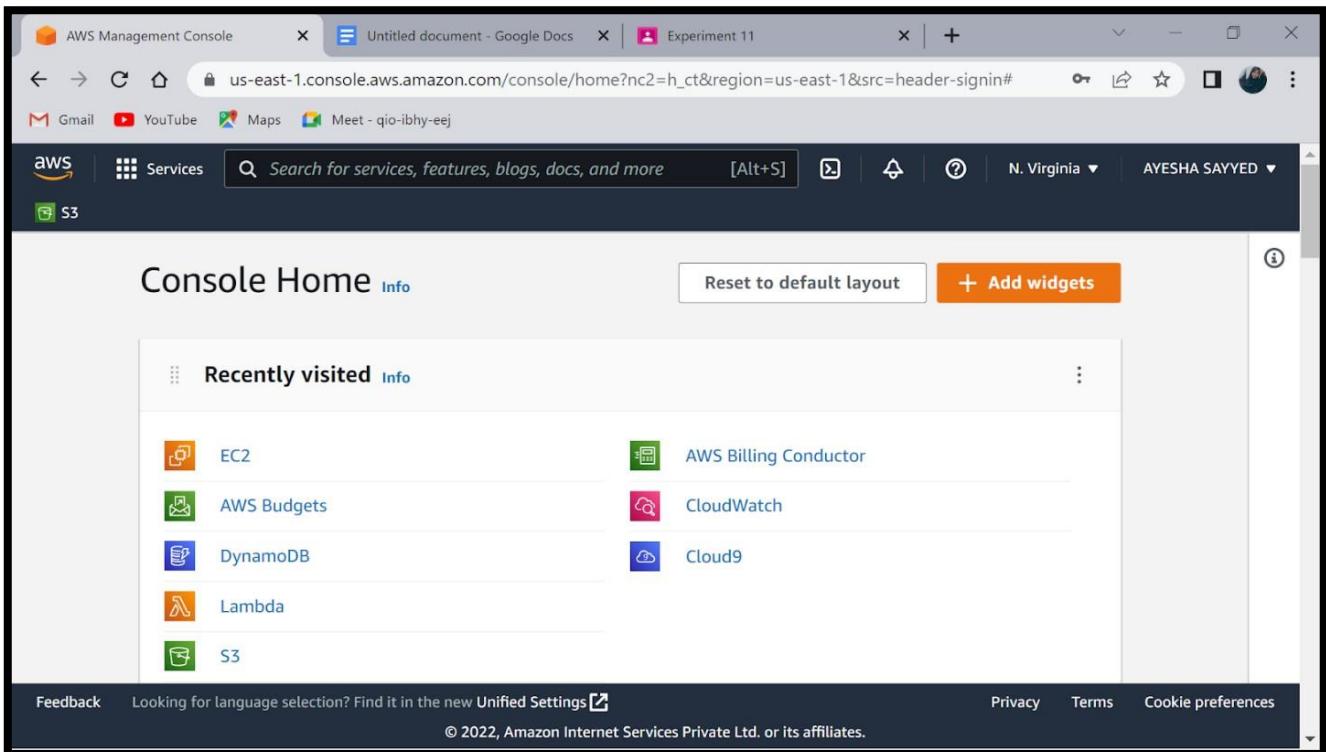
- AWS Elastic Beanstalk is cloud deployment and provisioning service that automates the process of getting applications set up on the Amazon Web Services (AWS) infrastructure.
- To use the service, developers just have to upload their applications. Provisioning, load balancing, autoscaling, and application health monitoring are all automatically handled.
- Elastic Beanstalk supports Web apps written in Java, Node.js, PHP, Python, Ruby, and .NET, among other languages and Web development stacks.
- An open architecture means that applications not written for the Web can also be deployed on the Elastic Beanstalk.
- The AWS Toolkit for Visual Studio and the AWS Toolkit for Eclipse allow developers to deploy and manage applications from within the integrated development environment (IDE).
- Developers can select elements of infrastructure management to administer, if desired.
- There is no separate charge for AWS Elastic Beanstalk – customers pay only for the resources used to store and run their applications.

Q2. Who should use AWS Elastic Beanstalk?

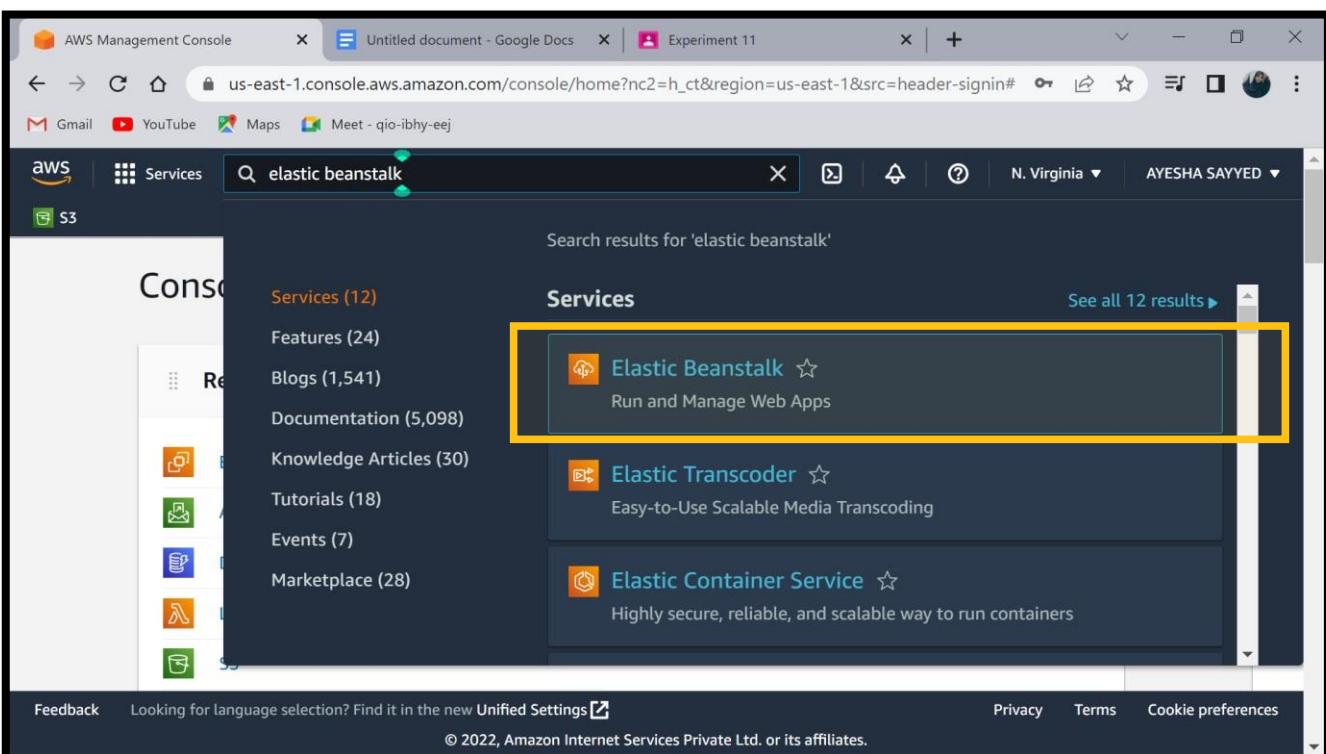
- ▲ Those who want to deploy and manage their applications within minutes in the AWS Cloud.
- ▲ You don't need experience with cloud computing to get started.
- ▲ AWS Elastic Beanstalk supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker web applications.

Q3. Deploy a web Application [any language] using AWs Elastic beanstalk.

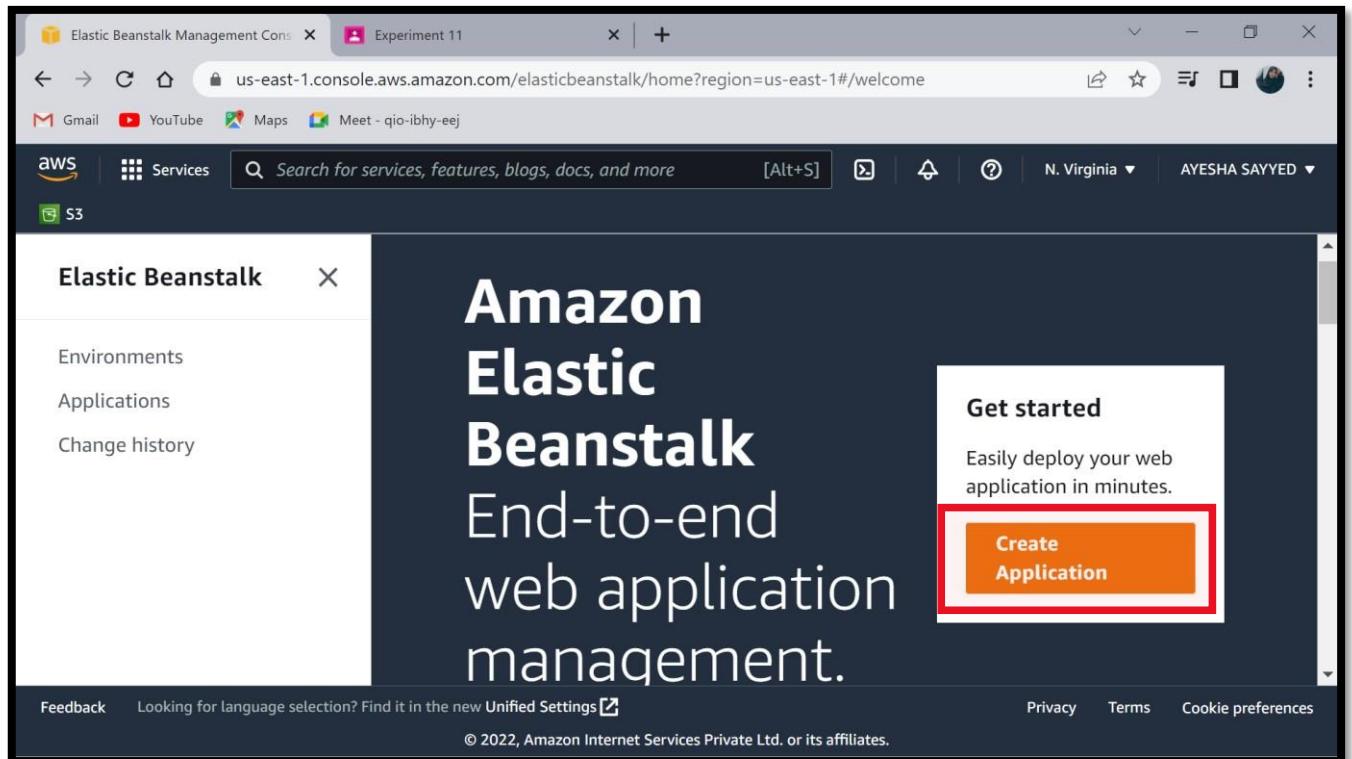
Step 1: Login to AWS Management Console Dashboard.



Step 2: Search for Elastic Beanstalk and select it.



Step 3: Click on Create Application.



Step 4: Give name to your ‘Application’ → ‘Create Application’.

The image consists of three vertically stacked screenshots of the AWS Elastic Beanstalk Management Console. Each screenshot shows a different step in the 'Create a web app' wizard.

Screenshot 1: Application Information

This screenshot shows the 'Create a web app' wizard with the title 'Create a web app'. It includes a brief description: 'Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf.' A link to 'Learn more' is provided. The 'Application name' field contains 'application_by_ayesha45', which is circled in red. Below the field is the note: 'Up to 100 Unicode characters, not including forward slash (/)'.

Screenshot 2: Platform Configuration

This screenshot shows the 'Platform' configuration section. It includes dropdown menus for 'Platform' (set to 'Java'), 'Platform branch' (set to 'Corretto 17 running on 64bit Amazon Linux 2'), and 'Platform version' (set to '3.3.1 (Recommended)').

Screenshot 3: Application Code Selection

This screenshot shows the 'Application code' selection section. It offers two options: 'Sample application' (selected) and 'Upload your code'. The 'Sample application' option is described as 'Get started right away with sample code.' The 'Upload your code' option is described as 'Upload a source bundle from your computer or copy one from Amazon S3.' At the bottom of this screen are three buttons: 'Cancel', 'Configure more options', and a large orange 'Create application' button, which is also circled in red.

Step 5: Now the application is creating within 5 to 10 mins.

The screenshot shows a browser window for the AWS Elastic Beanstalk console. The URL is us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/launchEnvironment?applicationName=Applicationbyayesha45-env&environmentName=Applicationbyayesha45-env. The page title is "Create environment". A purple circle highlights the message "Creating Applicationbyayesha45-env This will take a few minutes." Below this, a log window displays the following text:

```
9:17pm Using elasticbeanstalk-us-east-1-518313021583 as Amazon S3 storage bucket for environment data.  
9:17pm createEnvironment is starting.
```

At the bottom, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

This screenshot shows the same AWS Elastic Beanstalk console page after some time has passed. The log window now includes additional entries:

```
9:17pm Created security group named:  
awseb-e-hbh7em9knj-stack-AWSEBSecurityGroup-11DXFTYZ9EJL7  
9:17pm Created security group named:  
sg-01ffd550a2824d105  
9:17pm Created target group named:  
arn:aws:elasticloadbalancing:us-east-1:518313021583:targetgroup/awseb-AWSEB-1V8IXY2ERMJDJ/80e4894fb68e0922  
9:17pm Environment health has transitioned to Pending. Initialization in progress (running for 8 seconds). There are no instances.  
9:17pm Using elasticbeanstalk-us-east-1-518313021583 as Amazon S3 storage bucket for environment data.  
9:17pm createEnvironment is starting.
```

The rest of the interface is identical to the first screenshot, with the "Creating Applicationbyayesha45-env" message still circled in purple.

Step 6: The Application has been created in “Java”.

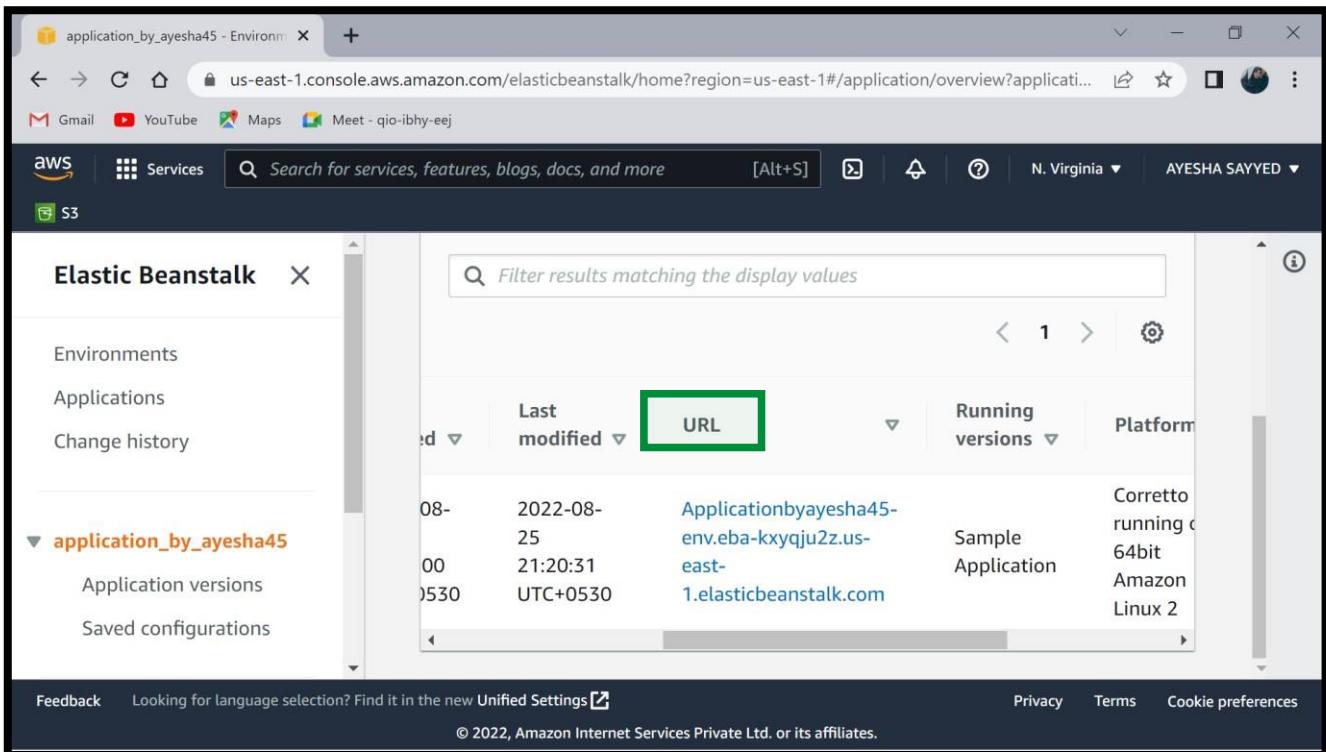
The screenshot shows the AWS Elastic Beanstalk application dashboard for the environment 'Applicationbyayesha45-env'. The main panel displays the application's status as 'Ok' with a green checkmark icon. It also shows the 'Running version' as 'Sample Application' and the 'Platform' as 'Corretto 17 running on 64bit Amazon Linux 2/3.1'. On the left sidebar, under the 'application_by_ayesha45' section, there are links for 'Application versions' and 'Saved configurations'. The bottom of the screen includes standard AWS navigation links like Gmail, YouTube, Maps, Meet, and a search bar.

The screenshot shows the AWS Elastic Beanstalk application dashboard focusing on the 'Recent events' log for the environment 'Applicationbyayesha45-env'. The log table lists several events:

Time	Type	Details
2022-08-25 21:21:21 UTC+0530	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 55 seconds ago and took 3 minutes.
2022-08-25 21:20:31 UTC+0530	INFO	Successfully launched environment: Applicationbyayesha45-env
2022-08-25		Application available at
2022-08-25 21:20:31 UTC+0530		seconds ago and took 3 minutes.
2022-08-25 21:20:31 UTC+0530	INFO	Successfully launched environment: Applicationbyayesha45-env
2022-08-25 21:20:30 UTC+0530	INFO	Application available at Applicationbyayesha45-env.eba-kxyqju2z.us-east-1.elasticbeanstalk.com.
2022-08-25 21:20:22 UTC+0530	INFO	Added instance [i-07d263faaf4aea485] to your environment.
2022-08-25 21:20:14 UTC+0530	INFO	Instance deployment completed successfully.

The left sidebar of the dashboard shows the same navigation options as the first screenshot, including 'Environments', 'Applications', and 'Change history'.

Step 7: Click on the URL given.

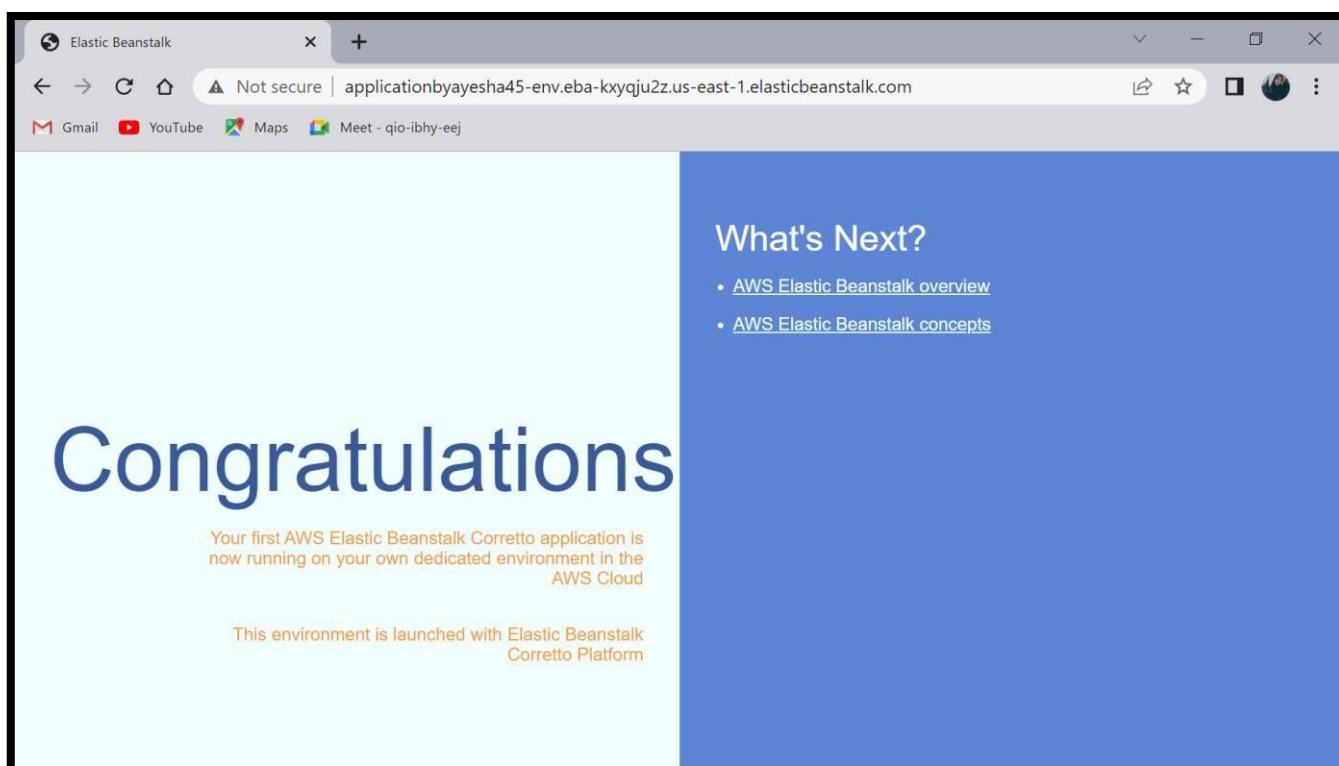


The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with links for Environments, Applications, and Change history. Under Applications, the 'application_by_ayesha45' application is selected. A green box highlights the 'URL' column header in the main table. The table lists one environment entry:

Created	Last modified	URL	Running versions	Platform
08-25 00:05:30	2022-08-25 21:20:31 UTC+0530	Applicationbyayesha45-env.eba-kxyqju2z.us-east-1.elasticbeanstalk.com	Sample Application	Corretto running on 64bit Amazon Linux 2

At the bottom of the page, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

Congratulations your Java Application has been created.



The screenshot shows a browser window with the title 'Elastic Beanstalk'. The address bar displays a 'Not secure' warning and the URL 'applicationbyayesha45-env.eba-kxyqju2z.us-east-1.elasticbeanstalk.com'. The page content is as follows:

Congratulations

Your first AWS Elastic Beanstalk Corretto application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Corretto Platform

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)

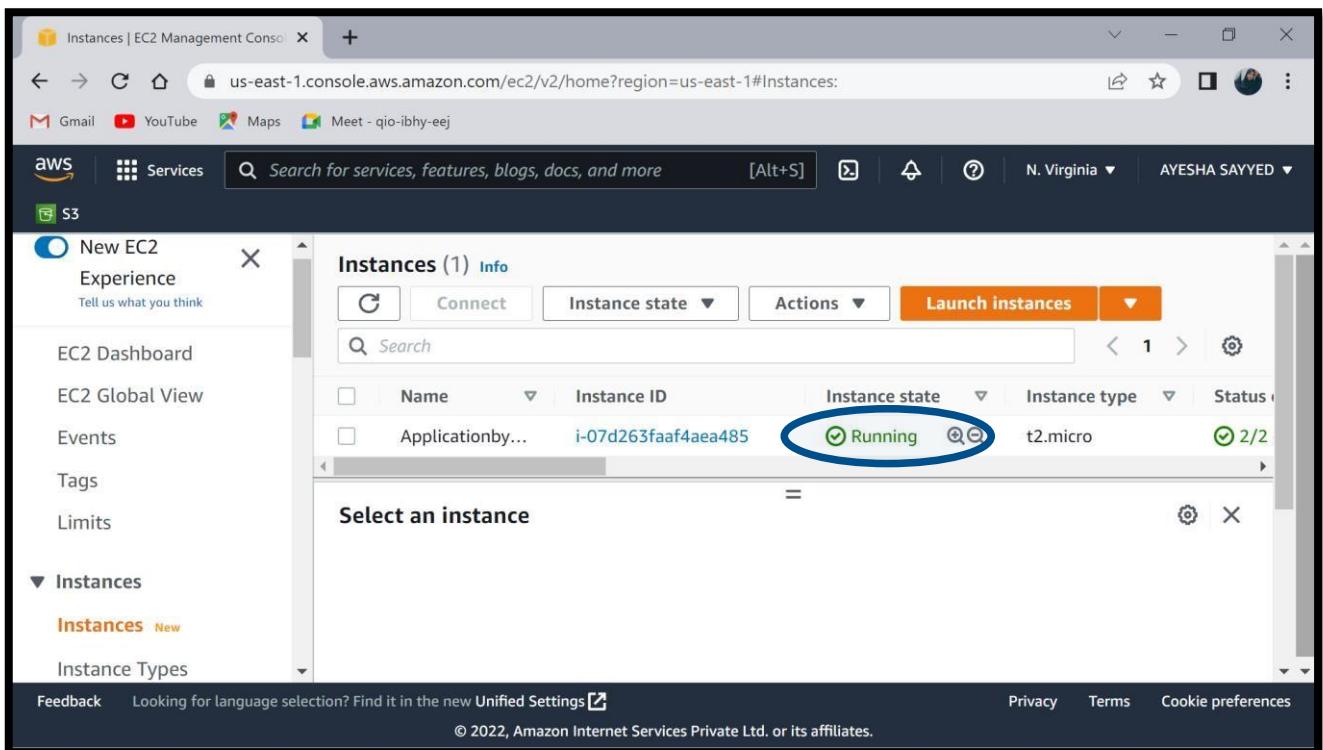
Step 8: Now search for EC2 and select it.

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with links like Environments, Applications, and Change history. Below that, under 'application_by_ayesha45', are Application version and Saved configurations. The main area has a search bar at the top with 'EC2' typed in. A search result for 'EC2' is shown, with the first item, 'EC2 Virtual Servers in the Cloud', highlighted with an orange box. Other results include 'EC2 Image Builder' and 'AWS Compute Optimizer'. At the bottom, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

Step 9: Click on Instance.

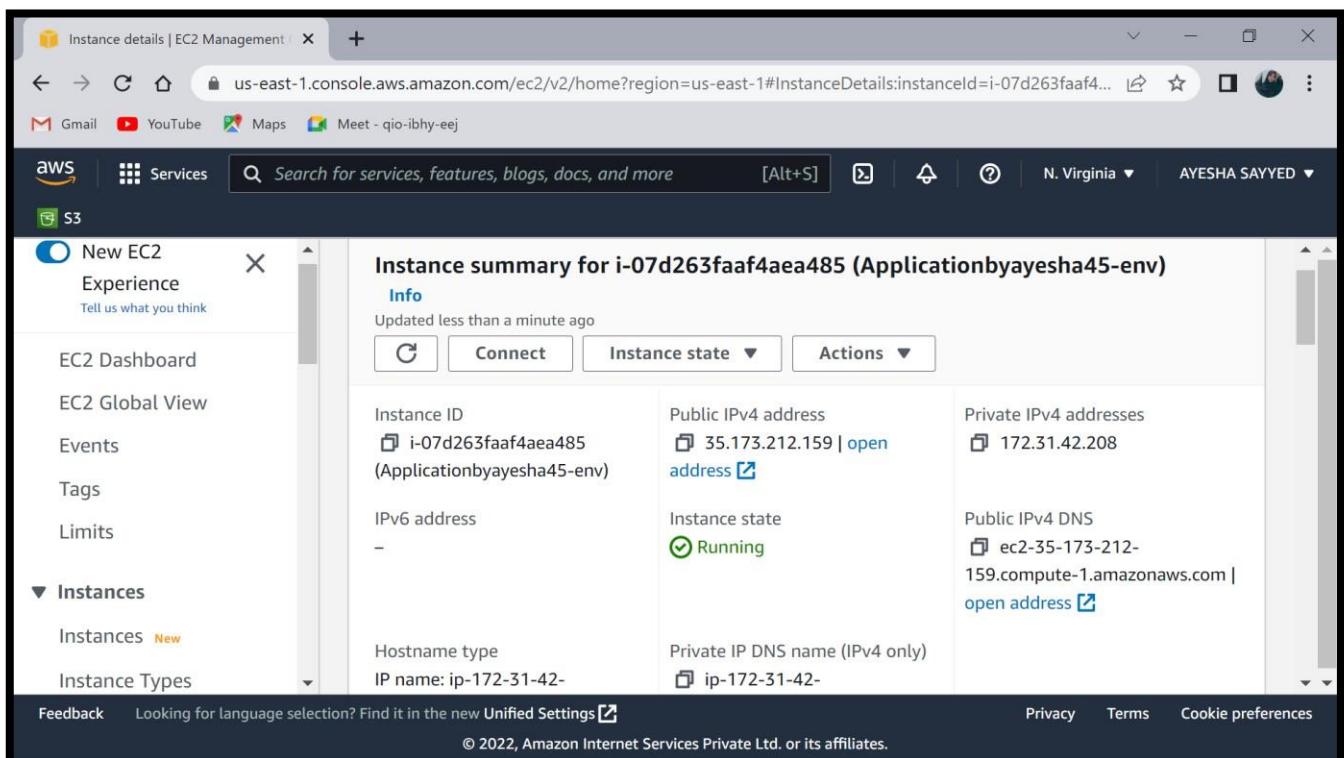
The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with 'New EC2 Experience' (circled with a blue oval), 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', and 'Instances' (also circled with a blue oval). Under 'Instances', there are 'Instances' and 'Instance Types' options. The 'Instances' option is highlighted with a red box. The main area has sections for 'Placement groups' (0), 'Security groups' (4), 'Snapshots' (0), and 'Volumes' (1). A callout box says 'Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server.' Below this are 'Launch instance' and 'Service health' sections. At the bottom, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

This is the influence of Elastic Beanstalk on EC2 and now the instance is running.



The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', and 'Instances'. Under 'Instances', 'Instances' is selected. The main area displays a table titled 'Instances (1) Info' with one row. The row contains columns for 'Name' (Applicationby...), 'Instance ID' (i-07d263faaf4aea485), 'Instance state' (Running, highlighted with a blue oval), 'Instance type' (t2.micro), and 'Status' (2/2). Below the table, a modal window titled 'Select an instance' is open. At the bottom of the page, there are links for 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences', along with a copyright notice for 2022, Amazon Internet Services Private Ltd. or its affiliates.

We can see all the details of the instance.



The screenshot shows the AWS EC2 Management Console interface, specifically the 'Instance details' view for the instance i-07d263faaf4aea485. The left sidebar shows 'Instances' is selected. The main area displays a table titled 'Instance summary for i-07d263faaf4aea485 (Applicationbyayesha45-env)'. The table includes columns for 'Instance ID' (i-07d263faaf4aea485, Applicationbyayesha45-env), 'Public IPv4 address' (35.173.212.159), 'Private IPv4 addresses' (172.31.42.208), 'IPv6 address' (empty), 'Instance state' (Running), 'Public IPv4 DNS' (ec2-35-173-212-159.compute-1.amazonaws.com), and 'Private IP DNS name (IPv4 only)' (ip-172-31-42-). There are also 'Actions' buttons for each row. At the bottom, there are links for 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences', along with a copyright notice for 2022, Amazon Internet Services Private Ltd. or its affiliates.

New EC2 Experience	208.ec2.internal	208.ec2.internal	Elastic IP addresses
Tell us what you think	Answer private resource DNS name	Instance type	-
EC2 Dashboard	-	t2.micro	
EC2 Global View	Auto-assigned IP address	VPC ID	AWS Compute Optimizer finding
Events	35.173.212.159 [Public IP]	vpc-0cf924feb1b794701	Opt-in to AWS Compute Optimizer for recommendations.
Tags			Learn more
Limits			
Instances	IAM Role	Subnet ID	Auto Scaling Group name
Instances New	aws-elasticbeanstalk-ec2-role	subnet-046d11ccc72afe728	awseb-e-hbh7em9knj-stack-AWSEBAutoScalingGroup-1J8GTQLDVIGQ3
Instance Types			

	Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance details <small>Info</small>							

New EC2 Experience	Platform	AMI ID	eb_corretto17_amazon_linux_2-hvm-2022-08-02T13-13
Tell us what you think	Linux/UNIX (Inferred)	ami-016fd50b7ccadc069	
EC2 Dashboard	Platform details	AMI name	Monitoring disabled
EC2 Global View	Linux/UNIX	aws-elasticbeanstalk-amzn-2.0.20220719.64bit-eb_corretto17_amazon_linux_2-hvm-2022-08-02T13-13	Termination protection Disabled
Events			
Tags			
Limits			
Instances	Stop protection	Launch time	AMI location
Instances New	Disabled	Thu Aug 25 2022 21:18:09 GMT+0530 (India Standard Time) (17 minutes)	amazon/aws-elasticbeanstalk-amzn-2.0.20220719.64bit-
Instance Types			

New EC2 Experience	Host ID	Affinity	Placement group
Tell us what you think	-	-	-
EC2 Dashboard	Host resource group name	Tenancy	
EC2 Global View	-	default	
Events	Virtualization type	Reservation	Partition number
Tags	hvm	r-0e64f217ed4b4ec99	-
Limits	Number of vCPUs		
Instances	1		
Instances New			
Instance Types			

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

[Privacy](#) [Terms](#) [Cookie preferences](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Step 10: Now terminate the Instances.

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances (which is currently selected). Under Instances, there are links for Instances (New) and Instance Types. The main area is titled 'Instances (1/1)' and shows one instance: 'i-07d263faaf4aea485 (Applicationbyayesha45-env)'. The instance is listed as 'Running' with an 't2.micro' instance type. Below the instance table, it says 'Instance: i-07d263faaf4aea485 (Applicationbyayesha45-env)'. At the top right, there are buttons for Stop instance, Start instance, Reboot instance, Hibernate instance, and Terminate instance. The 'Terminate instance' button is highlighted with a purple oval. The status bar at the bottom indicates the instance is 'Running' with '2/2' instances.

This screenshot shows a confirmation dialog box titled 'Terminate instance?'. It contains a warning message: 'On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.' Below the message, it asks 'Are you sure you want to terminate these instances?' followed by a list of instances: 'i-07d263faaf4aea485 (Applicationbyayesha45-env)'. At the bottom, it says 'To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.' There are 'Cancel' and 'Terminate' buttons, with 'Terminate' being highlighted with a purple box. The background shows the same EC2 Management Console interface as the previous screenshot.

Terminated Successfully

The screenshot shows the AWS EC2 Management Console interface. A green notification bar at the top right displays the message "Successfully terminated i-07d263faaf4aea485". This message is highlighted with a yellow oval. Below the notification, the main content area shows a table titled "Instances (1/1) Info" with one row. The row contains the following data:

Name	Instance ID	Instance state	Instance type	Status
Applicationby...	i-07d263faaf4aea485	Shutting-down	t2.micro	2/2

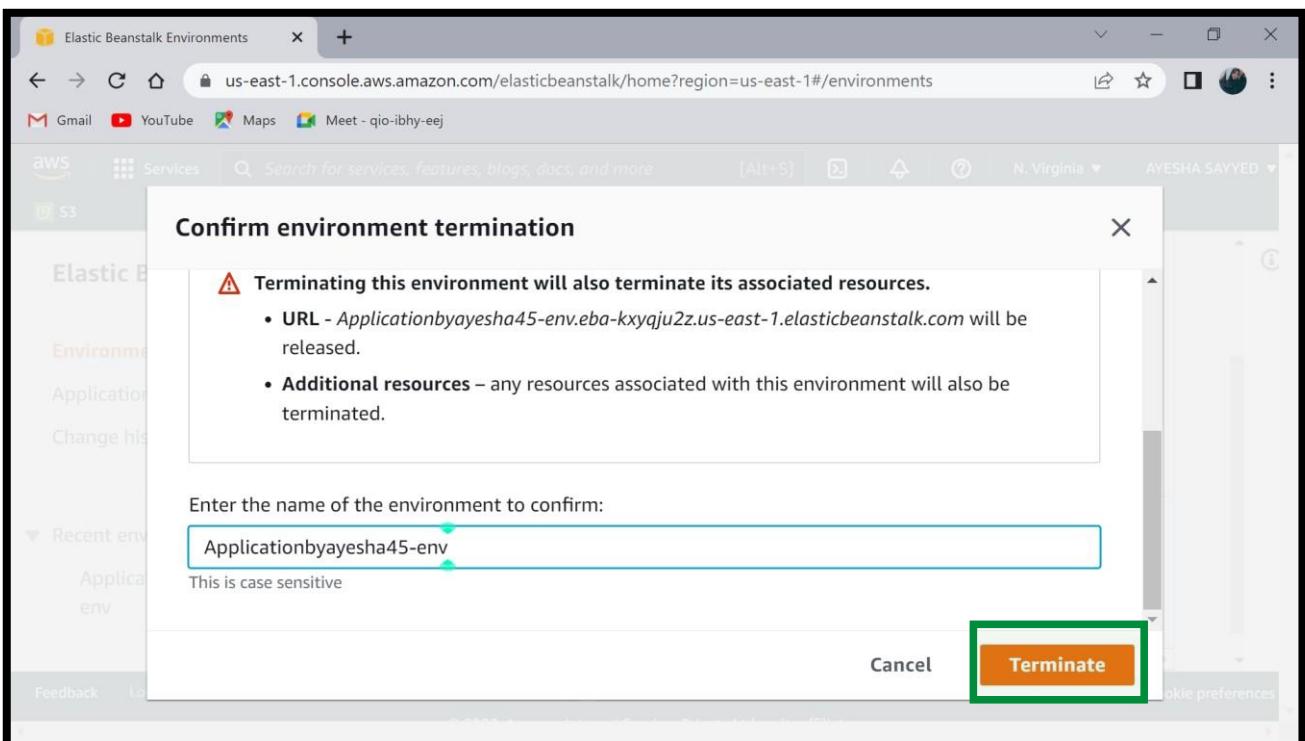
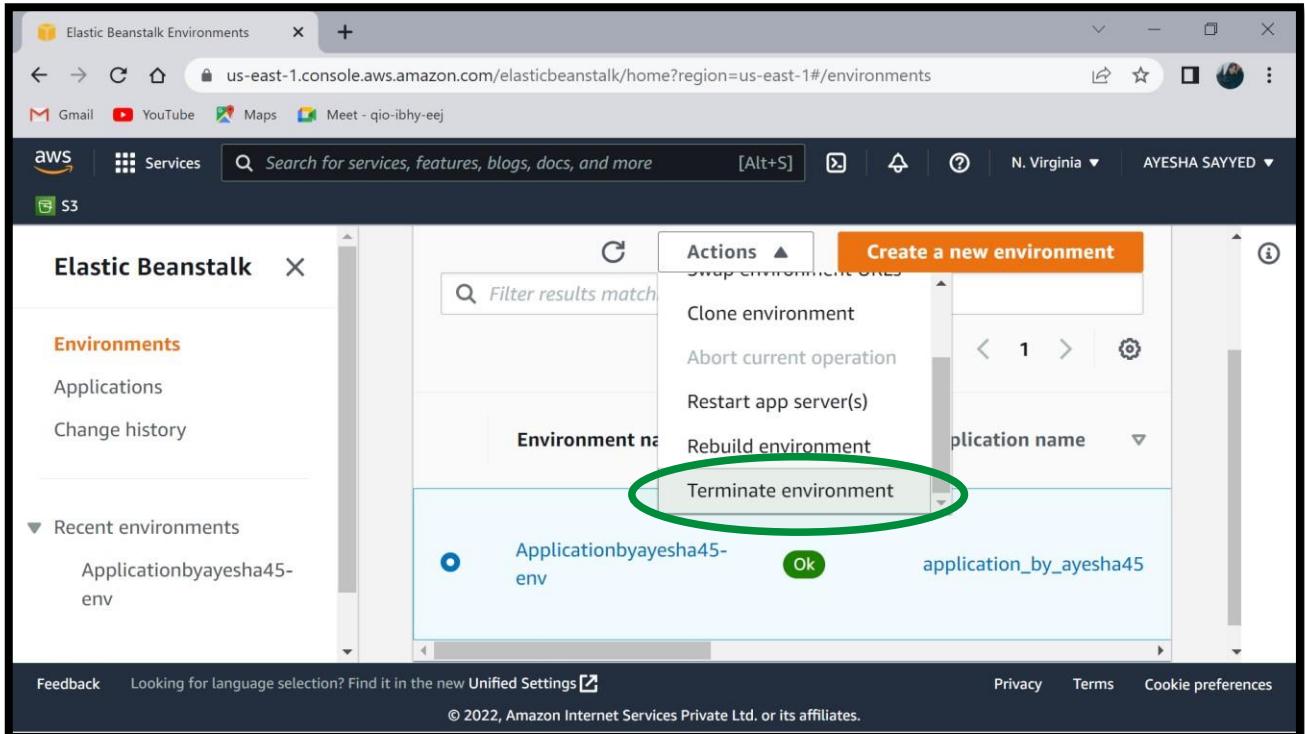
Below the table, a section titled "Instance: i-07d263faaf4aea485 (Applicationbyayesha45-env)" is displayed. The entire screenshot is framed by a thick black border.

The screenshot shows the AWS EC2 Management Console interface. The main content area displays a table titled "Instances (1) Info" with one row. The row contains the following data:

Name	Instance ID	Instance state	Instance type	Status
Applicationby...	i-07d263faaf4aea485	Terminated	t2.micro	-

The "Instance state" column for the single instance is highlighted with a yellow rectangle. Below the table, a section titled "Select an instance" is visible. The entire screenshot is framed by a thick black border.

Step 11: Now go to → ‘ELASTIC BEANSTALK’ → Terminated the Environment.



Now delete the Application.

The screenshot shows the AWS Elastic Beanstalk Applications page. On the left, there's a sidebar with links for Environments, Applications (which is highlighted in orange), and Change history. Below that is a section for Recent environments. The main area displays a table of applications. One row is selected, showing the application name "application_by_ayesha45" and its environment "Applicationbyayesha45-env". To the right of the application name is a "Date created" timestamp: "2022-08-25 21:16:53 UTC+0530". Above the application table is an "Actions" dropdown menu with options: Create environment, Delete application, View application versions, View saved configurations, and Restore terminated environment. The "Restore terminated environment" option is circled in blue. The top navigation bar includes links for Gmail, YouTube, Maps, Meet, and a search bar. The top right corner shows the user's name "AYESHA SAYYED".

The screenshot shows a "Confirm Application Deletion" dialog box. It contains a message stating: "If you proceed with this action, the following environments will be terminated:" followed by a bulleted list: "• Applicationbyayesha45-env". Below this is a field labeled "Enter the name of the application to confirm:" with the text "application_by_ayesha45" entered into it. At the bottom of the dialog are two buttons: "Cancel" and a large orange "Delete" button, which is also circled in blue. The background of the dialog is semi-transparent, showing the Elastic Beanstalk Applications page from the previous screenshot.

CASE-STUDY

[KUBERNETES]

Q1. What is Kubernetes?

Kubernetes automates operational tasks of container management and includes built-in commands for deploying applications, rolling out changes to your applications, scaling your applications up and down to fit changing needs, monitoring your applications, and more making it easier to manage applications.

Kubernetes, or K8s for short, is an open-source container-orchestration tool designed by Google. It's used for bundling and managing clusters of containerized applications a process known as 'orchestration' in the computing world. The name Kubernetes originates from Greek, meaning helmsman or pilot.

Kubernetes is a portable, extensible, opensource platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Q2. How is Kubernetes related to Docker?

Kubernetes is open-source orchestration software that provides an API to control how and where those containers will run. It allows you to run your Docker containers and workloads and helps you to tackle some of the operating complexities when moving to scale multiple containers, deployed across multiple servers.

You can decide to use Kubernetes without Docker, or even Docker without Kubernetes for that matter (but we advise you to use it for different purposes than running containers). Still, even though Kubernetes is a rather extensive tool, you will have to find a good container runtime for it – one that has implemented CRI.

Kubernetes is most commonly used with Docker managed containers, although it doesn't strictly depend on it.

There are still problems with this architecture though, for example:
Stopping and starting instances is slow.

Communicating between nodes in this architecture can be quite complex. If you're using a monolithic architecture over microservices, you may not need to deal with communication between nodes and so won't experience these issues. You will still need to worry about instance startup time whenever introducing rolling upgrades or performing a failover. This can incur extra costs since the risk of losing requests

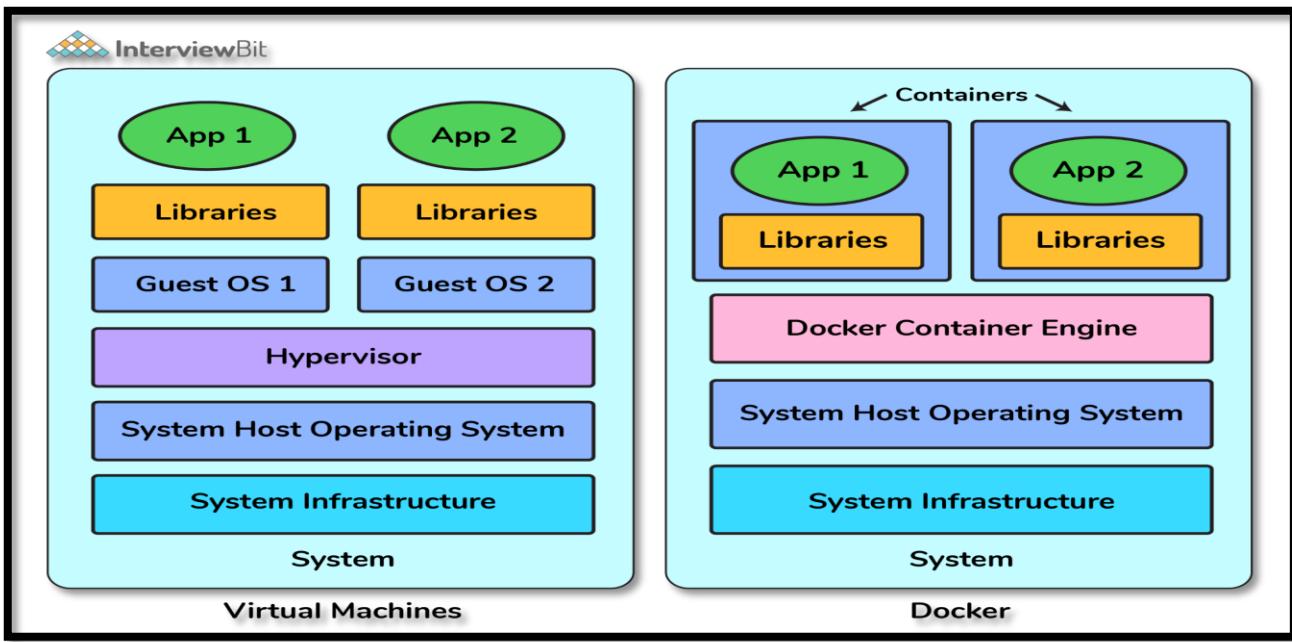
becomes higher the longer a restart takes, so you're likely to want to have another instance to minimize that risk. This is where Kubernetes comes in. Kubernetes is a platform for managing containerised services. This means that it's a tool made to abstract away details such as separation of nodes, while automating things like rolling upgrades, failover, and scaling of services. The idea is that you should be able to deploy in a very similar way whether running locally or in the cloud.

This blog will walk through some of the basics of Kubernetes, and will setup a simple microservice application locally to demonstrate its use. This blog will assume you have the following tools installed and configured already:

- ▲ Maven (v3+)
- ▲ JDK 8

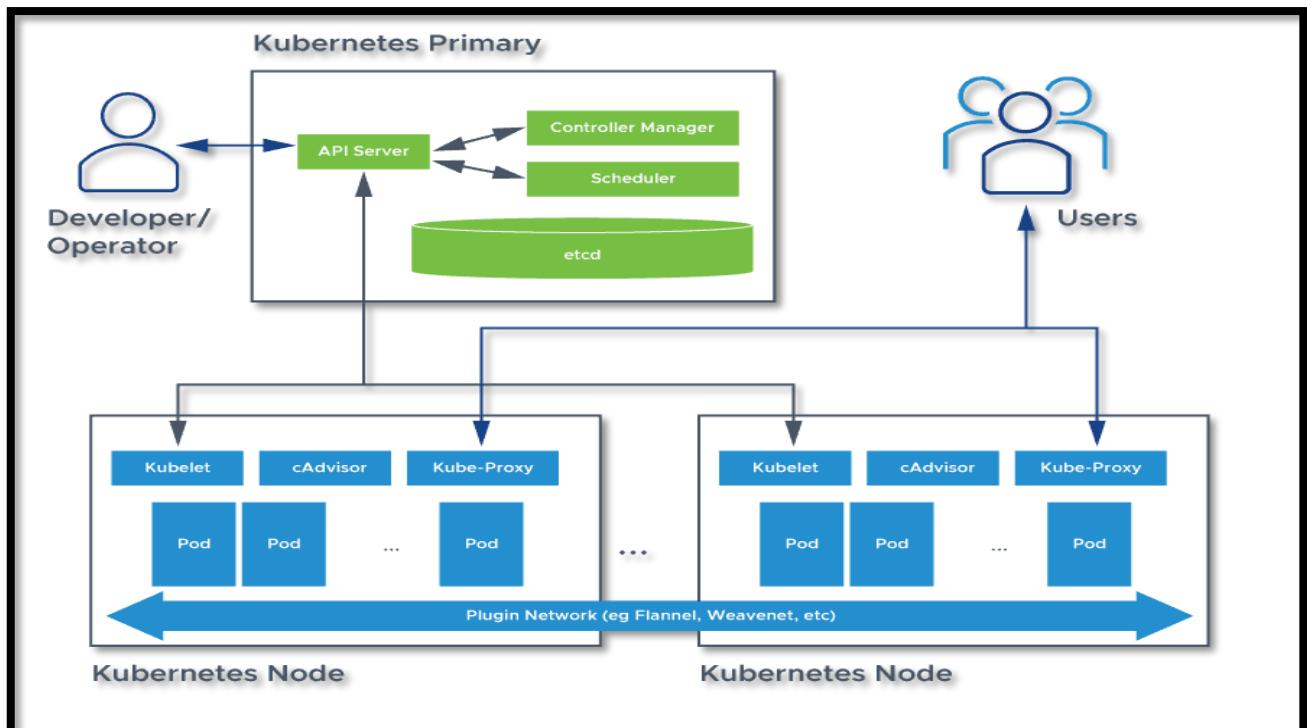
Q3. What is the difference between deploying applications on hosts and containers?

Parameter	Virtual Machines	Containers
Guest OS	Each VM runs on virtual hardware and Kernel is loaded into its own memory region	All the guests share same OS and Kernel. Kernel image is loaded into the physical memory
Communication	Will be through Ethernet Devices	Standard IPC mechanisms like Signals, pipes, sockets etc.
Security	Depends on the implementation of Hypervisor	Mandatory access control can be leveraged
Performance	Virtual Machines suffer from a small overhead as the Machine instructions are translated from Guest to Host OS.	Containers provide near native performance as compared to the underlying Host OS.
Isolation	Sharing libraries, files etc between guests and between hosts not possible.	Subdirectories can be transparently mounted and can be shared.
Startup time	VMs take a few mins to boot up	Containers can be booted up in a few secs as compared to VMs.
Storage	VMs take much more storage as the whole OS kernel and its associated programs have to be installed and run	Containers take lower amount of storage as the base OS is shared



Q6. Explain Kubernetes architecture with neat diagram and explain all the component of Kubernetes architecture.

Basic Kubernetes architecture exists in two parts: the control plane and the nodes or compute machines. Each node could be either a physical or virtual machine and is its own Linux environment. Every node also runs pods, which are composed of containers.



Control Plane Components:

The control plane's components make global decisions about the cluster (for example, scheduling), as well as detecting and responding to cluster events (for example, starting up a new pod when a deployment's replicas field is unsatisfied). Control plane components can be run on any machine in the cluster. However, for simplicity, set up scripts typically start all control plane components on the same machine, and do not run user containers on this machine. See [Creating Highly Available clusters with kubeadm](#) for an example control plane setup that runs across multiple machines.

kube-apiserver:

The API server is a component of the Kubernetes control plane that exposes the Kubernetes API. The API server is the front end for the Kubernetes control plane. The main implementation of a Kubernetes API server is kube-apiserver. kube-apiserver is designed to scale horizontally—that is, it scales by deploying more instances. You can run several instances of kube-apiserver and balance traffic between those instances.

Etcd:

Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data. If your Kubernetes cluster uses etcd as its backing store, make sure you have a backup plan for that data. You can find in-depth information about etcd in the official documentation.

kube-scheduler:

Control plane component that watches for newly created Pods with no assigned node, and selects a node. Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.

kube-controller-manager:

Control plane component that runs controller processes. Logically, each controller is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process.

Some types of these controllers are:

- ▲ **Node controller:** Responsible for noticing and responding when nodes go down.
- Job controller: Watches for Job objects that represent one-off tasks, then creates Pods to run those tasks to completion.

- **EndpointSlice controller:** Populates EndpointSlice objects (to provide a link between Services and Pods).
- **ServiceAccount controller:** Create default ServiceAccounts for new namespaces.

cloud-controller-manager:

A Kubernetes control plane component that embeds cloud-specific control logic. The cloud controller manager lets you link your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that only interact with your cluster.

The cloud-controller-manager only runs controllers that are specific to your cloud provider. If you are running Kubernetes on your own premises, or in a learning environment inside your own PC, the cluster does not have a cloud controller manager.

As with the kube-controller-manager, the cloud-controller-manager combines several logically independent control loops into a single binary that you run as a single process. You can scale horizontally (run more than one copy) to improve performance or to help tolerate failures.

The following controllers can have cloud provider dependencies:

- **Node controller:** For checking the cloud provider to determine if a node has been deleted in the cloud after it stops responding
- **Route controller:** For setting up routes in the underlying cloud infrastructure
- **Service controller:** For creating, updating and deleting cloud provider load balancers

Node Components:

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.

Kubelet:

An agent that runs on each node in the cluster. It makes sure that containers are running in a Pod. The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.

kube-proxy:

kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept. kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster. kube-proxy uses the operating system packet filtering layer if there is one and it's available. Otherwise, kube-proxy forwards the traffic itself.

Container runtime:

The container runtime is the software that is responsible for running containers. Kubernetes supports container runtimes such as containerd, CRI-O, and any other implementation of the Kubernetes CRI (Container Runtime Interface).

Addons:

Addons use Kubernetes resources (DaemonSet, Deployment, etc) to implement cluster features. Because these are providing cluster-level features, namespaced resources for addons belong within the kube-system namespace. Selected addons are described below; for an extended list of available addons, please see Addons.

DNS :

While the other addons are not strictly required, all Kubernetes clusters should have cluster DNS, as many examples rely on it. Cluster DNS is a DNS server, in addition to the other DNS server(s) in your environment, which serves DNS records for Kubernetes services. Containers started by Kubernetes automatically include this DNS server in their DNS searches.

Web UI (Dashboard):

Dashboard is a general purpose, web-based UI for Kubernetes clusters. It allows users to manage and troubleshoot applications running in the cluster, as well as the cluster itself.

Container Resource Monitoring:

Container Resource Monitoring records generic time-series metrics about containers in a central database, and provides a GUI for browsing that data. Cluster-level Logging A cluster-level logging mechanism is responsible for saving container logs to a central log store with search/browsing interface.

CASE-STUDY [SONARQUBE]

Q1. What is SonarQube? Why use SonarQube?

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.

SonarQube (formerly Sonar) is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells on 29 programming languages.

SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be measured continually over time.

Q2. What is software quality measurement ?

In Software Engineering, Software Measurement is done based on some Software Metrics where these software metrics are referred to as the measure of various characteristics of a Software. In Software engineering Software Quality Assurance (SAQ) assures the quality of the software.

Q3. What is Static and Dynamic Code Analysis?

Dynamic analysis is the testing and evaluation of an application during runtime. Static analysis is the testing and evaluation of an application by examining the code without executing the application. Many software defects that cause memory and threading errors can be detected both dynamically and statically.

For example, static code analysis is a form of white-box testing that can help identify security issues in source code. On the other hand, dynamic code analysis is a form of black-box vulnerability scanning that allows software teams to scan running applications and identify vulnerabilities.

Q4. What are the benefits of using SonarQube?

- Sustainability** - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
- Increase productivity** - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- Quality code** - Code quality control is an inseparable part of the process of software development.
- Detect Errors** - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
- Increase consistency** - Determines where the code criteria are breached and enhances the quality.
- Business scaling** - No restriction on the number of projects to be evaluated.
- Enhance developer skills** - Regular feedback on quality problems helps developers to improve their coding skills.

EXPERIMENT – 15

STEPS TO CLOSE THE AWS ACCOUNT

Close your account

To close your AWS account, do the following:

1. Sign in to the AWS Management Console as the root user of the account.
2. From the navigation bar, choose your account name, and then choose Account.
3. Scroll to the Close Account section.
4. Read and understand the terms of closing your account.
5. Select all checkboxes, and then choose Close Account.
6. In the confirmation box, choose Close Account.

Within a few minutes, you receive email confirmation that your account is closed successfully.

You can choose to sign in to your account three days after closing the account to check if all the resources are terminated. Open the AWS Billing and Cost Management console to monitor whether you continue to incur charges. You can contact AWS Support if you continue to incur charges after terminating all resources.

Before closing your AWS account, do the following:

1. Review the following account considerations for closing your account.
2. Review the following billing considerations for closing your account.
3. Be sure to terminate all resources before closing your account.
4. Be sure to pay your outstanding bills.

After completing these steps, you can close your account.

Before closing your account

- You must have completed the account sign up process.

Sign in as the AWS account root user. If you sign in to an account with an AWS Identity and Access Management (IAM) user or role, you can't close the account.

- Back up any resources or data that you want to keep. For instructions about how to back up a particular resource, see the AWS documentation for that service.

- For AWS Organization accounts:

By default, member accounts don't have a root password. Before you can sign in as the root user, you must reset the root user password for these accounts. If your account is the management account of an organization, you must make

sure that all member accounts are closed or removed from your organization. For more information, see [Removing a member account from your organization](#). To close the payer account in an organization, first delete the organization.

After closing your account

- ▲ You can still sign in and file an AWS Support case or contact Support for 90 days.
- ▲ Your EC2 instances may be stopped after your account has been closed. If you reopen your account, you may need to restart your EC2 instances.
- ▲ After 90 days, any content remaining in your account will be permanently deleted, and AWS services that aren't already terminated will be terminated. However, service attributes might be retained as long as necessary for billing and administration purposes. AWS retains your account information as described in the Privacy Notice. You can't permanently delete your account before 90 days. You can't reopen the account after 90 days.

Note: The account resources in AWS China (Beijing) and AWS China (Ningxia) Regions are subject to the policies of operating partners (Sinnet in the Beijing Region and NWCD in the Ningxia Region). Account closure procedures in China might take longer than in other AWS Regions.

- ▲ You can't create new AWS accounts using the email address that was associated with your account at the time of its closure.

Terminate all your resources before closing your account

Closing your account might not automatically terminate all your active resources. You might continue to incur charges for some of your active resources even after you close your account. You're charged for any usage fees incurred before closure.

Before closing your account, do the following:

1. Find all your active resources. For more information, see [How do I check for active resources that I no longer need on my AWS account?](#)
2. Terminate all your resources. For more information, see [How do I terminate active resources that I no longer need on my AWS account?](#)



Account Closure Is A Permanent Action

Please note account closure is a permanent action and once your account is closed it will no longer be available to you and cannot be restored. If you decide later that you want to start ordering from us again, or if you would like to use products and services that require an account, you will need to create a new account.

Please

Yes, I want to per

For closing your Amazon account (Optional)

My Amazon Account and delete my data.

Close My Account