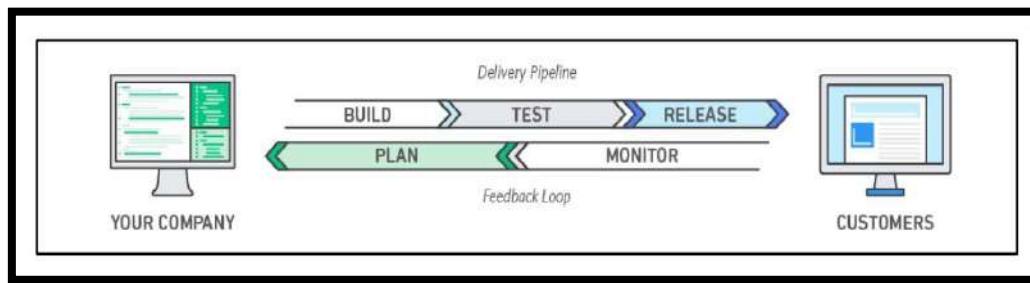


EXPERIMENT – O1

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).

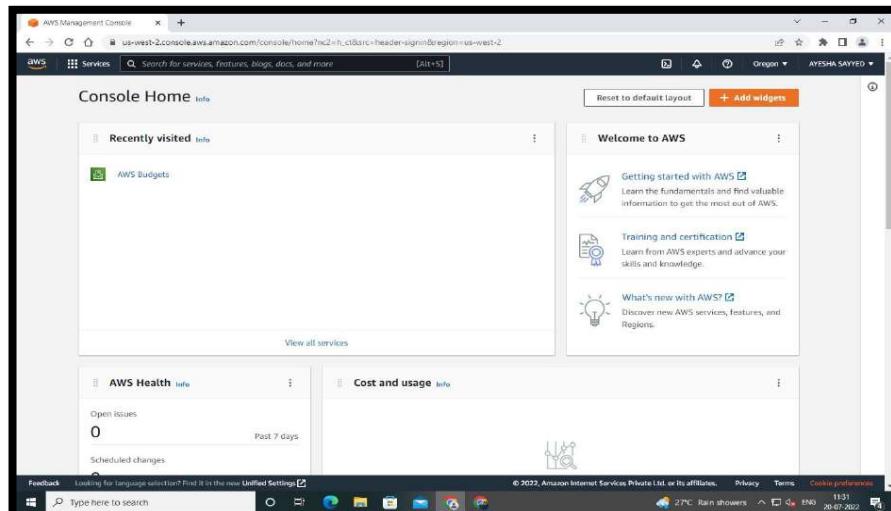
- Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as instance store volumes .
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes.
- Multiple physical locations for your resources, 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.

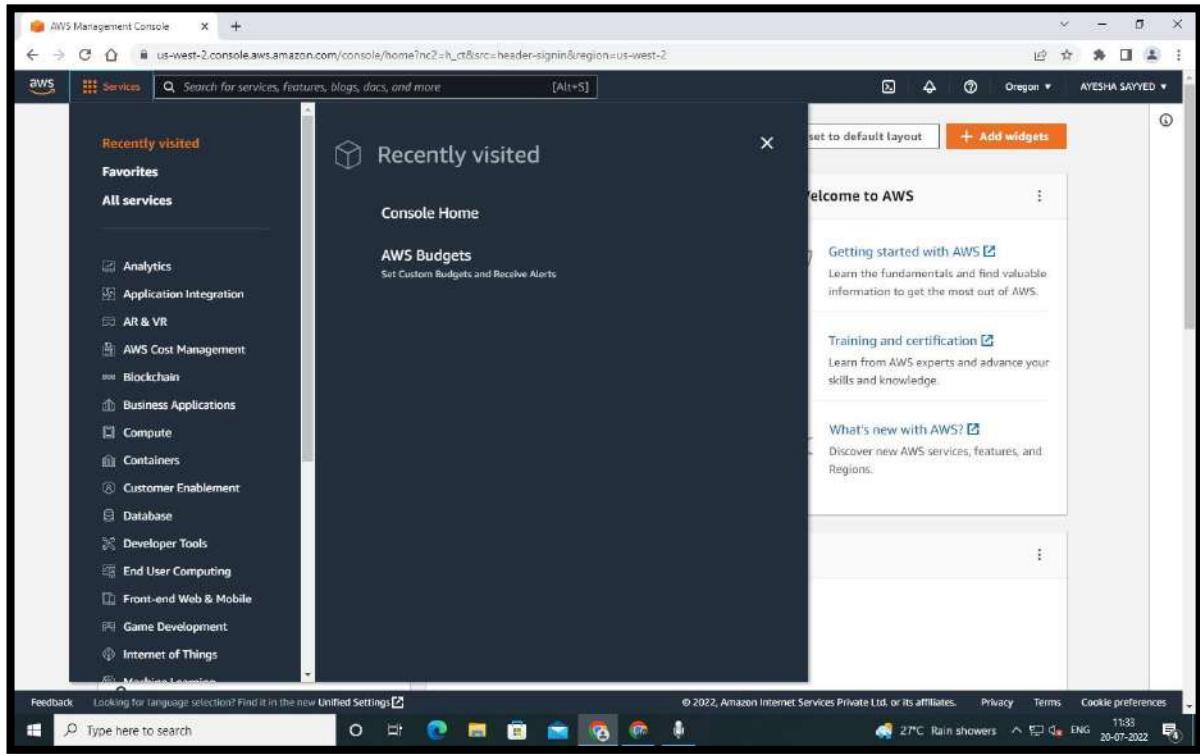
Open google.com from both the instances ,search your own name .

→[A] [WINDOWS INSTANCE]

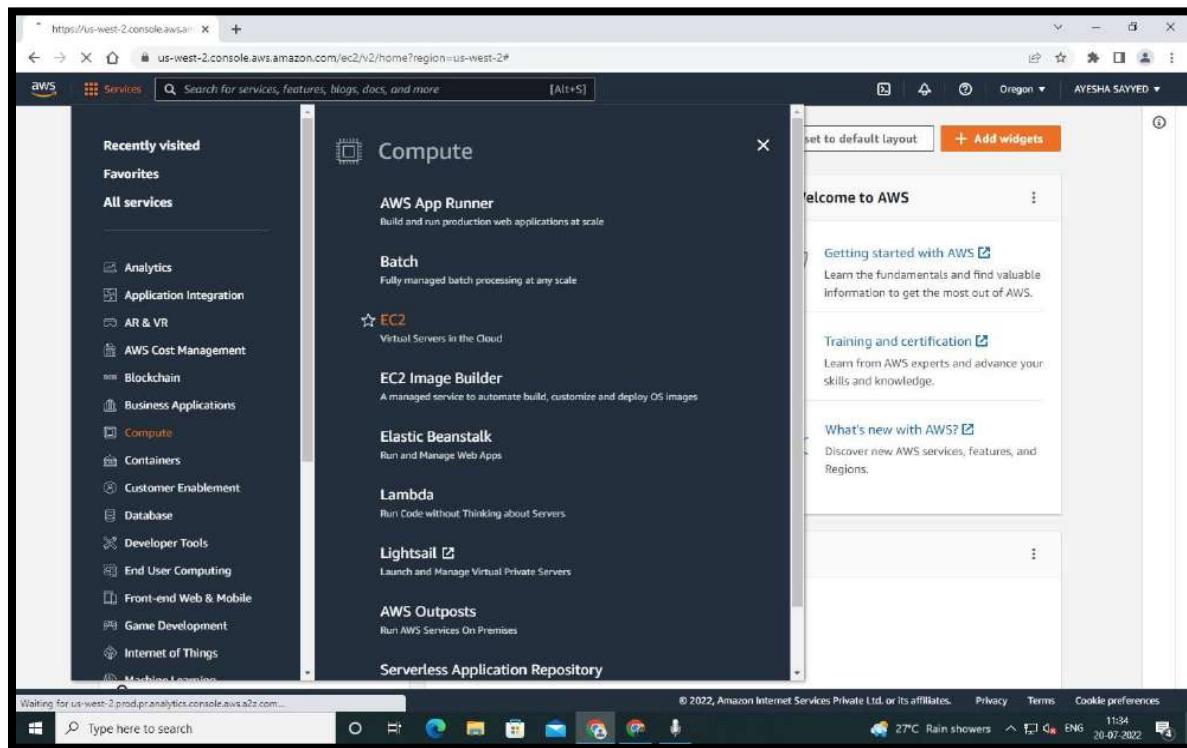
STEP 1 : AWS Management Console Dashboard



STEP 2 : Click on Services.



STEP 3 : Now CLICK on Compute -> EC2.



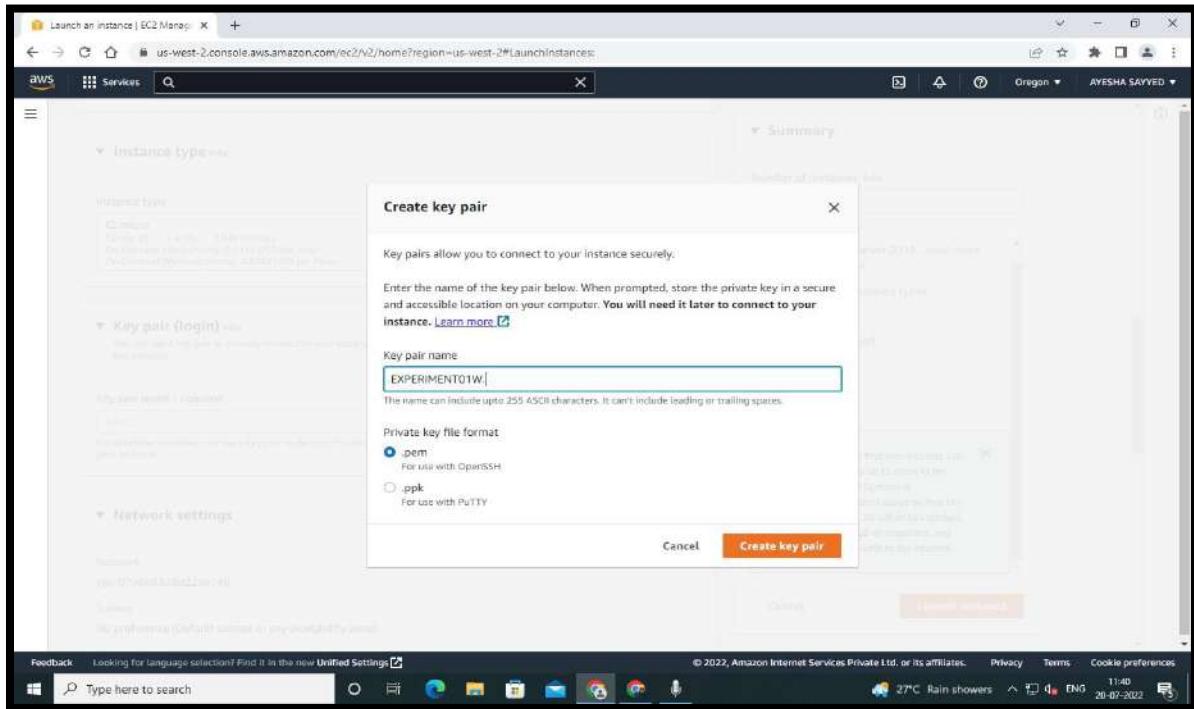
STEP 4 : Launch Instance

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#LaunchInstances>. 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 'Launch instance' button and a note: 'To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.' Below it is another button 'Launch instance from template' and a note: 'Note: Your instances will launch in the US West (Oregon) Region.'

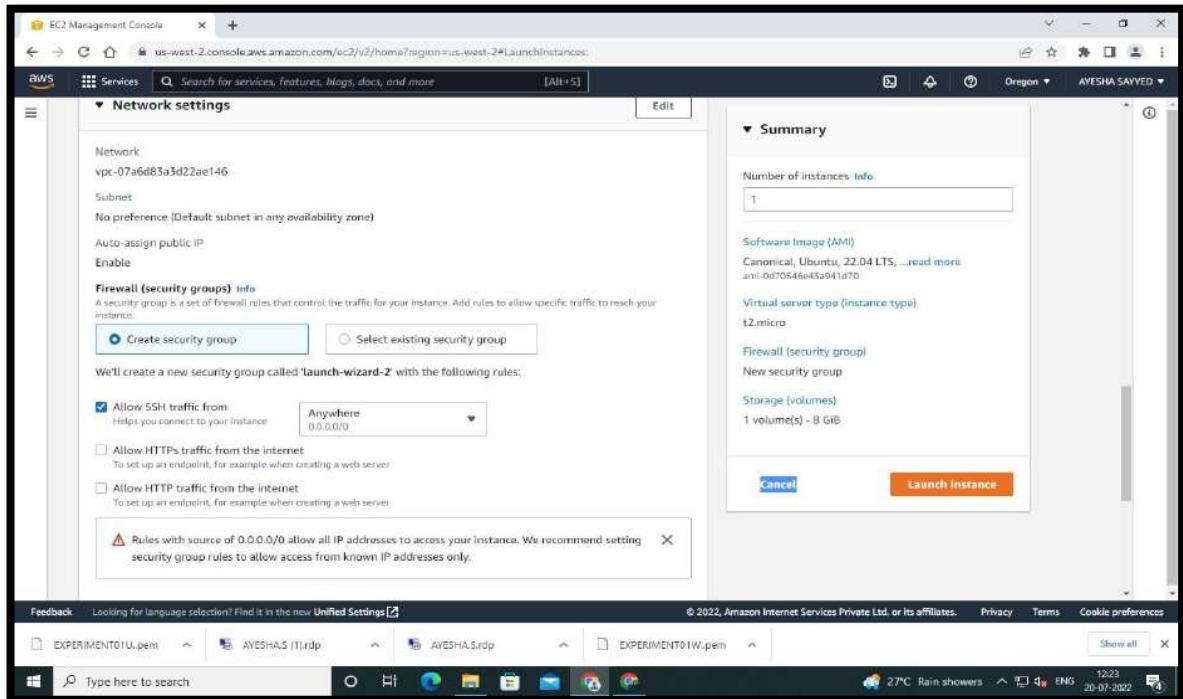
STEP 5 : Select Windows

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 'Instances > Launch an instance'. The main area has a 'Summary' section on the right with 'Number of instances: 1'. Below it are sections for 'Amazon Linux 2 Kernel 5.10 AMI...', 'Virtual server type (instance type): t2.micro', 'Firewall (security group): New security group', 'Storage (volumes): 1 volume(s) - 8 GB', and a note about the free tier. On the left, there's a 'Name and tags' section with 'Name: AYESHA.S' and a 'Search our full catalog including 1000s of application and OS images' bar. At the bottom, there's a 'Quick Start' section with icons for Amazon Linux, Ubuntu, Windows, Red Hat, and SUSE Linux, and a 'Launch Instance' button.

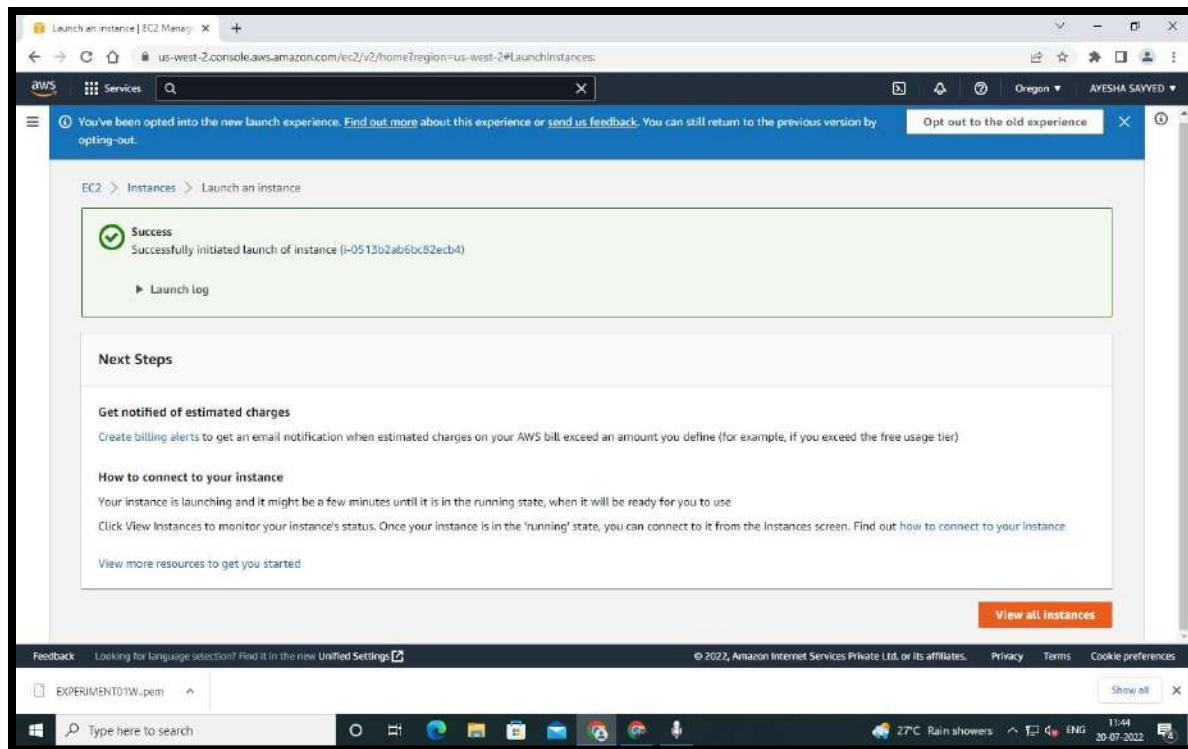
STEP 6: Create key pair. A .pem file will be downloaded which will be later used to connect to the instance.



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

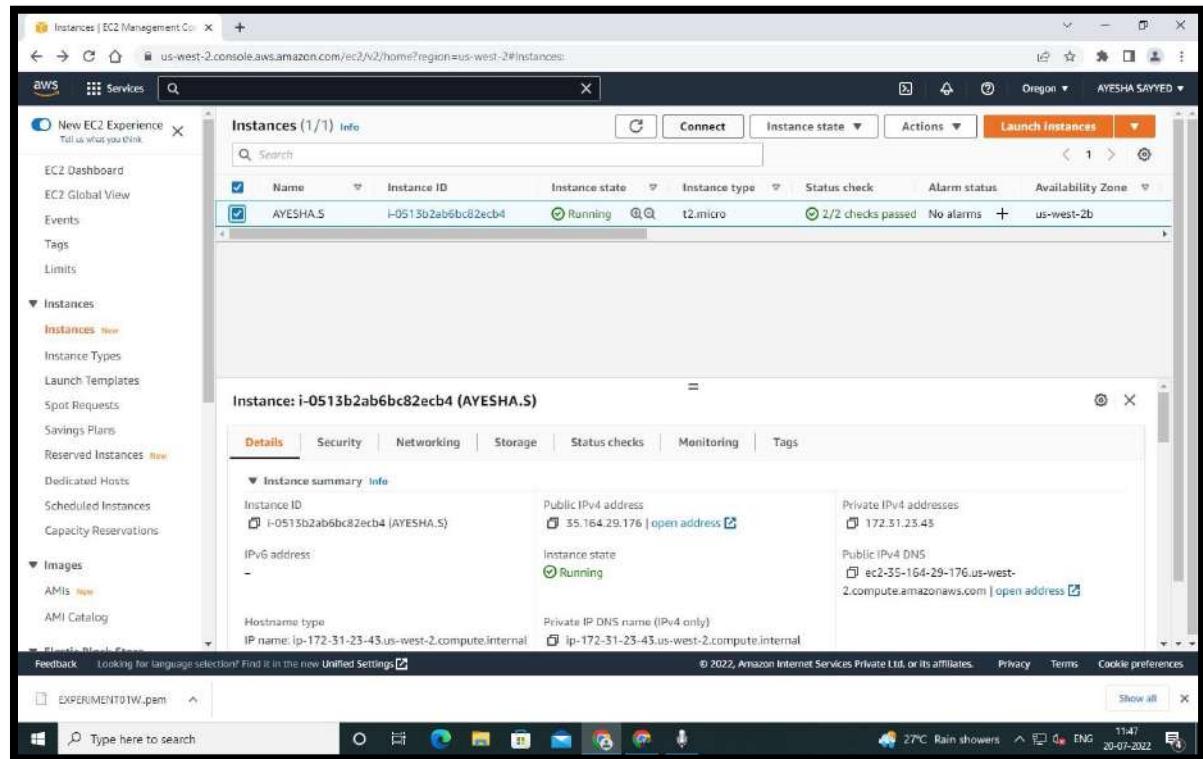


Success message will be shown after successful creation of instance



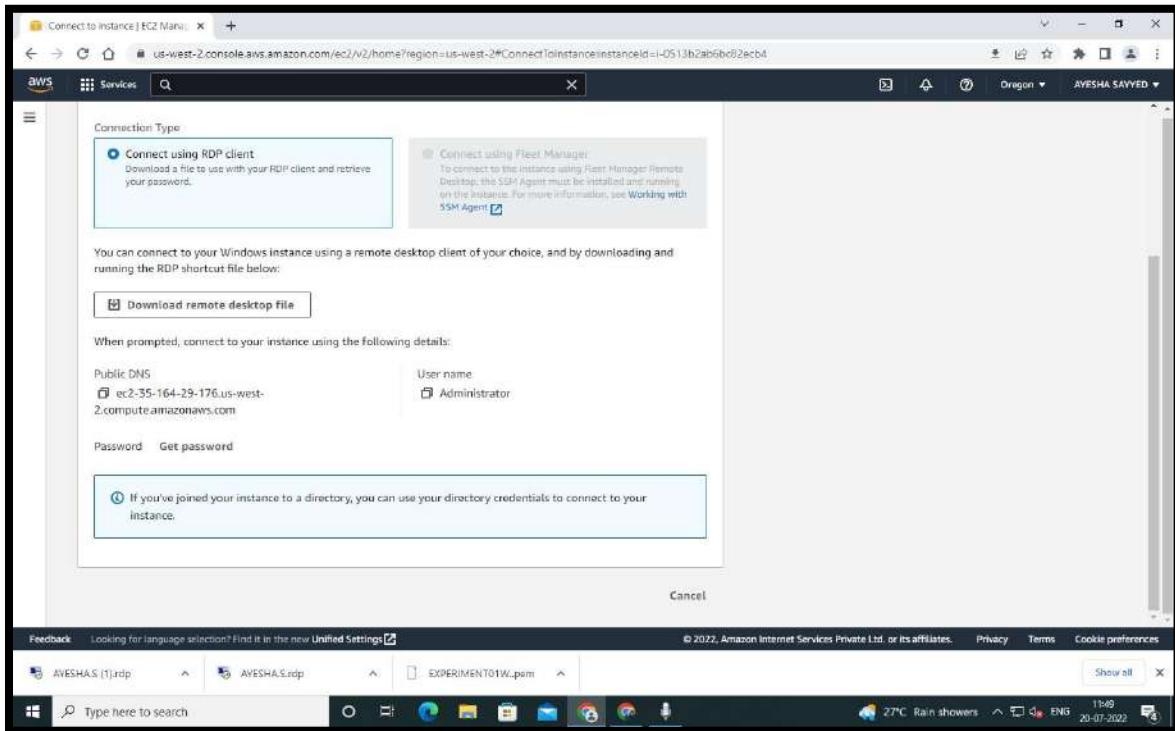
STEP 8: RDP Connection

We have to download Remote Desktop File to connect to the instance.

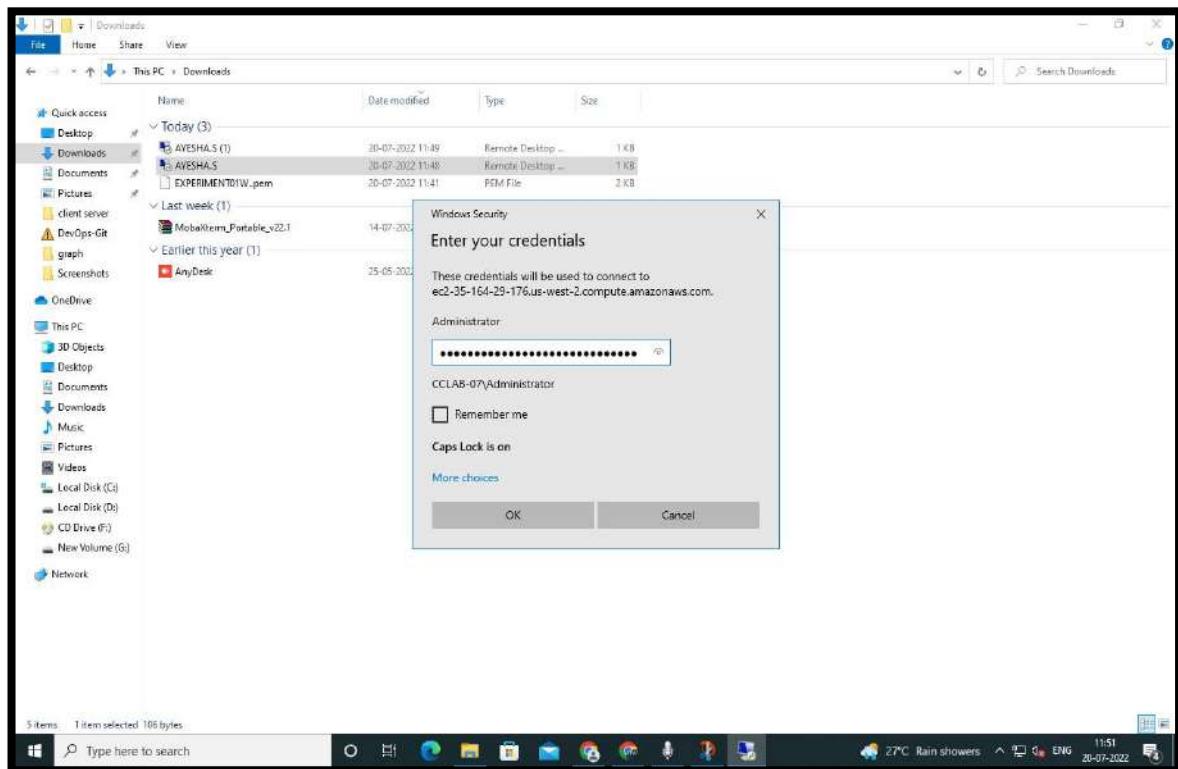


Go to RDP Client to download the Remote Desktop File

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



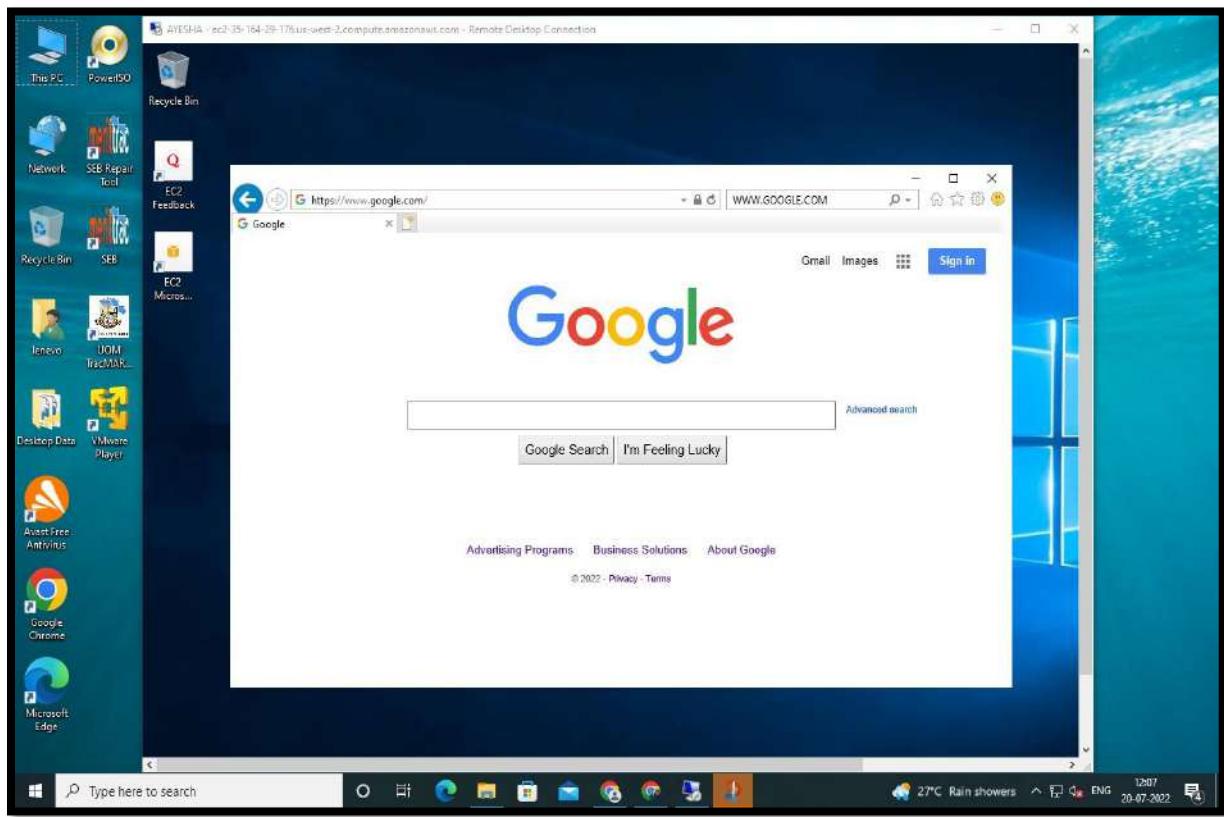
Now, run the Remote Desktop File and put the decrypted password copied earlier



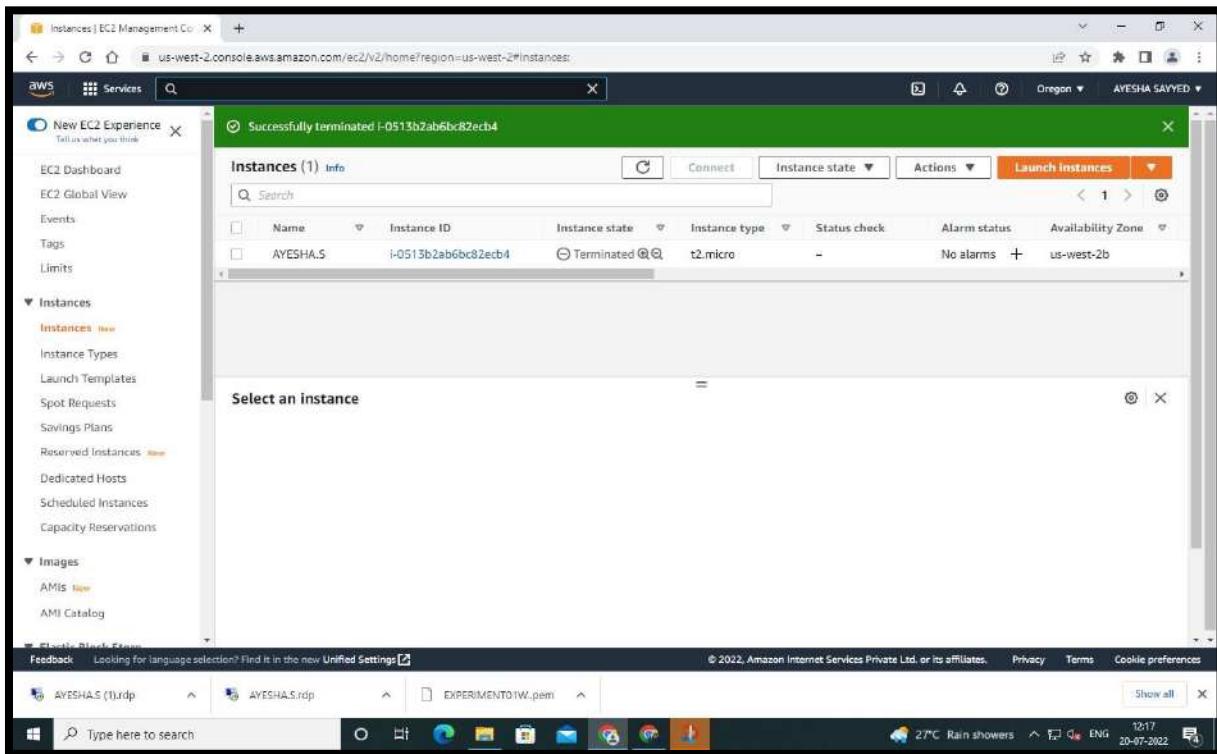
Your Windows Instance will start running.



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

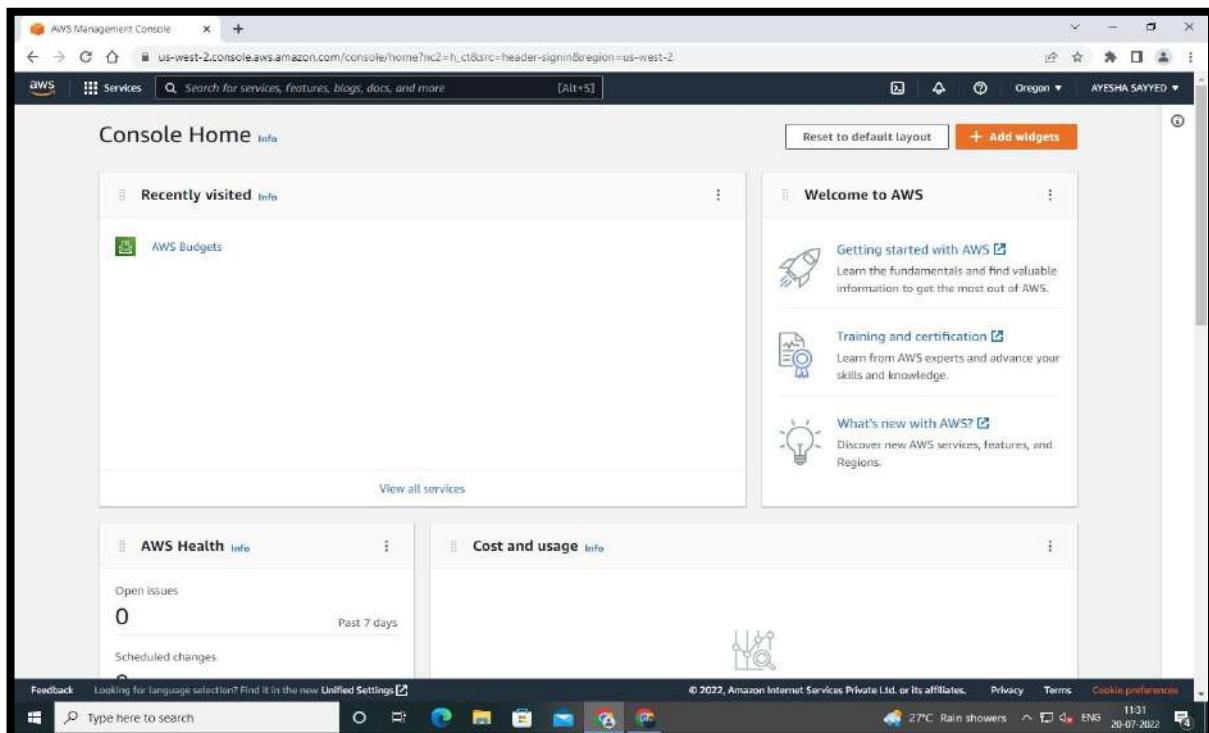


STEP 10 : Terminate the instance if you don't wish to use it again.



→[B] [UBUNTU INSTANCE]

STEP 1: AWS Management Console Dashboard



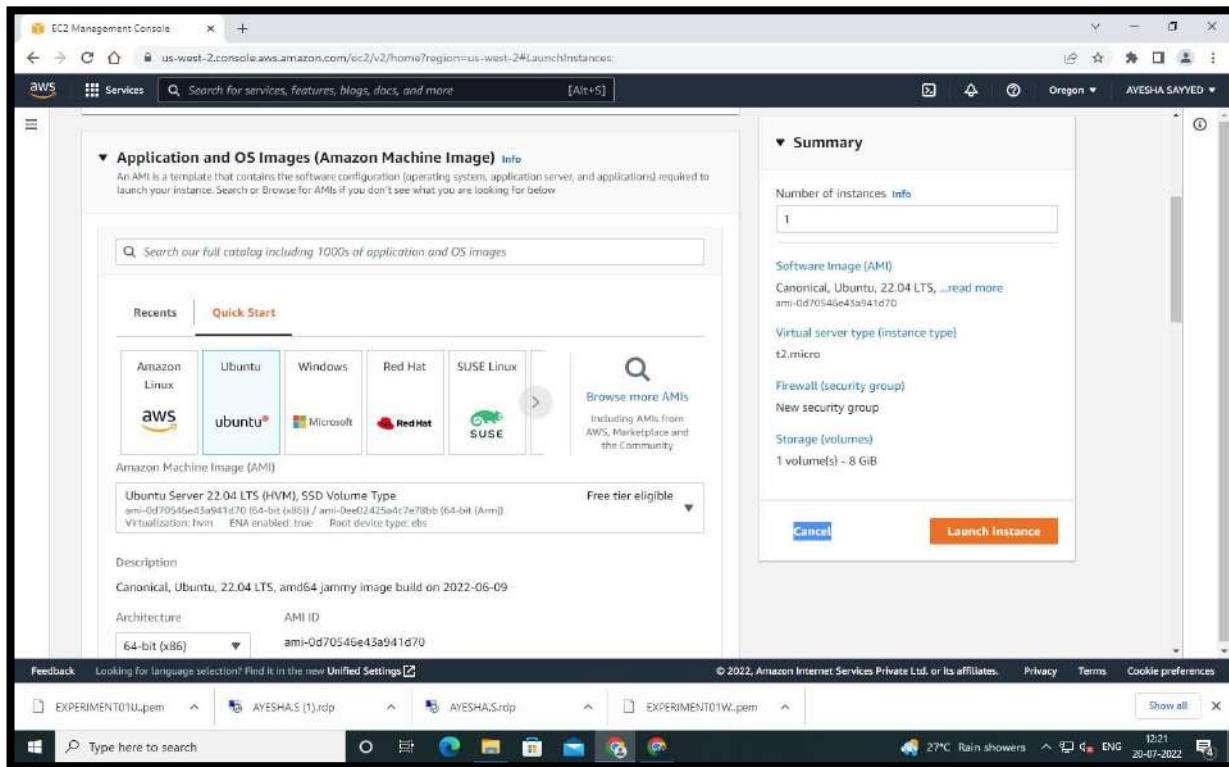
STEP 2 : CLICK on Services ->Compute ->EC2.

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. On the right, there's a 'Welcome to AWS' panel with sections for 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. The status bar at the bottom indicates 'Waiting for us-west-2.prod.analytics.console.aws.a2z.com...' and shows system information like '27°C Rain showers' and the date '20-07-2022'.

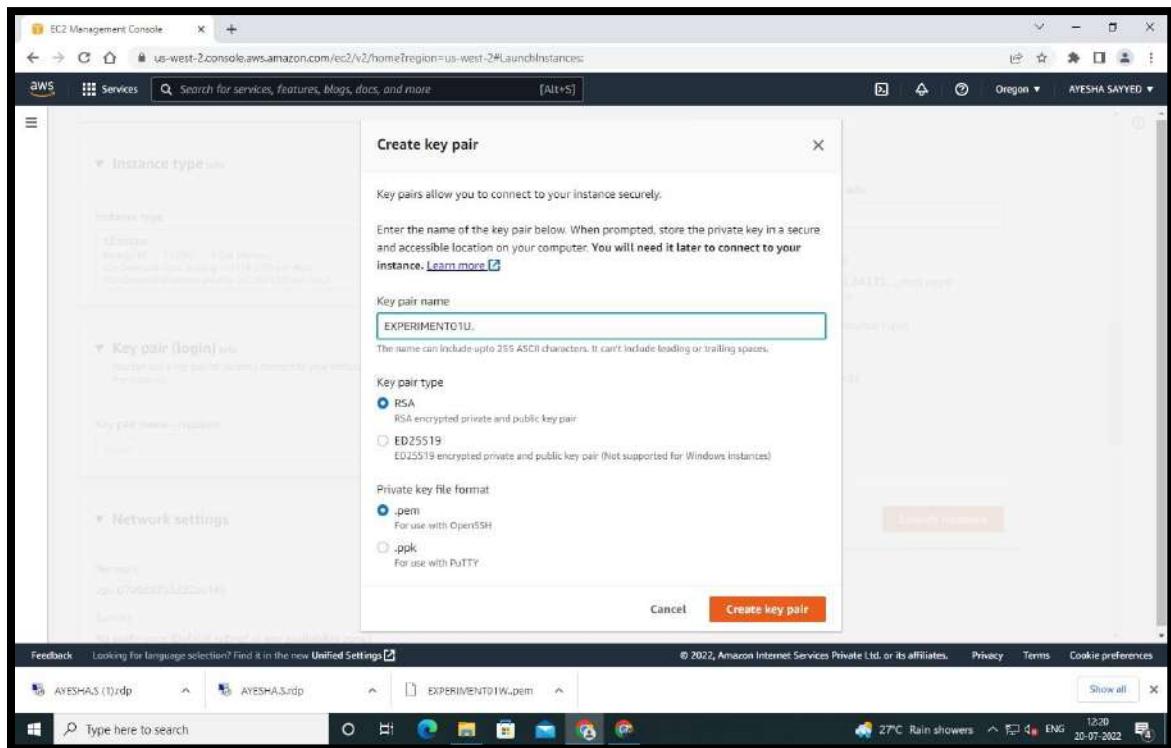
STEP 3 : Launch Instance.

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' and sections for 'EC2 Dashboard', 'Instances', 'Images', and 'Elastic Block Store'. The main content area has three main sections: 'Resources' (listing 0 instances, 0 dedicated hosts, 0 elastic IPs, etc.), 'Launch instance' (with a red 'Launch instance' button), and 'Service health' (showing 'US West (Oregon)' and 'This service is operating normally'). On the right, there's an 'Account attributes' section with details like 'Supported platforms: VPC', 'Default VPC: vpc-07a6d83a3d22ae146', and 'Explore AWS' sections for ML Inference and Price Performance. The status bar at the bottom indicates '27°C Rain showers' and the date '20-07-2022'.

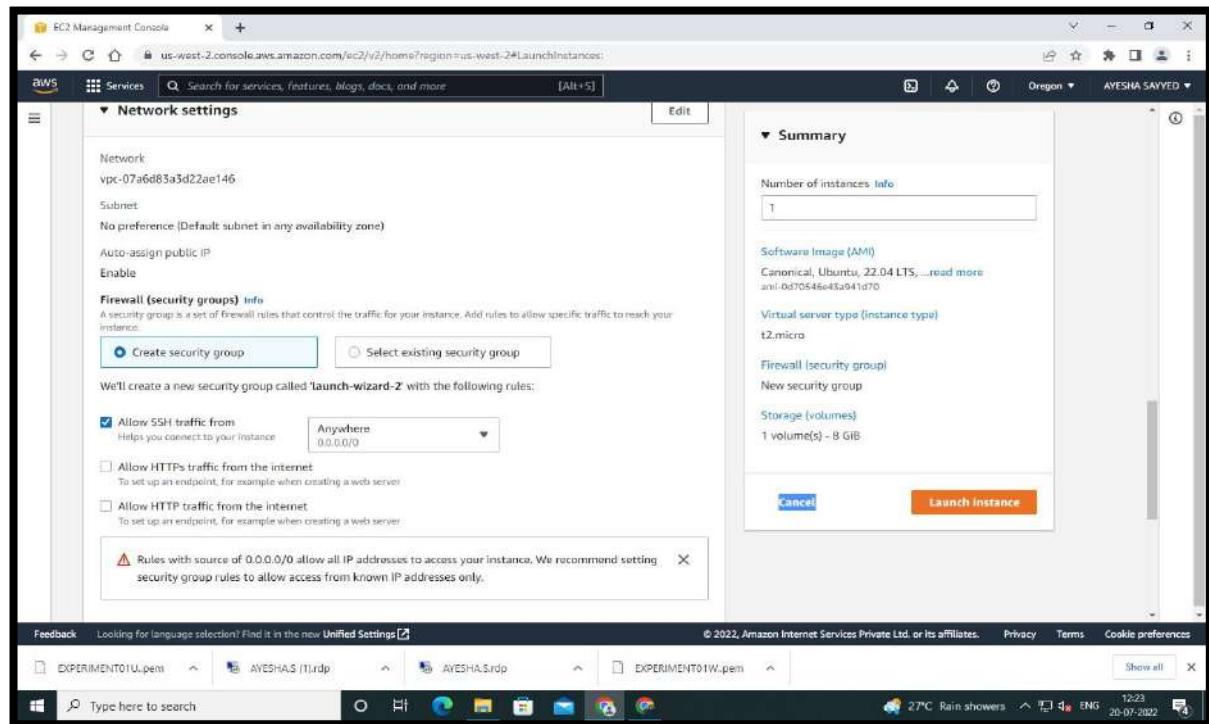
STEP 4 : Select Ubuntu



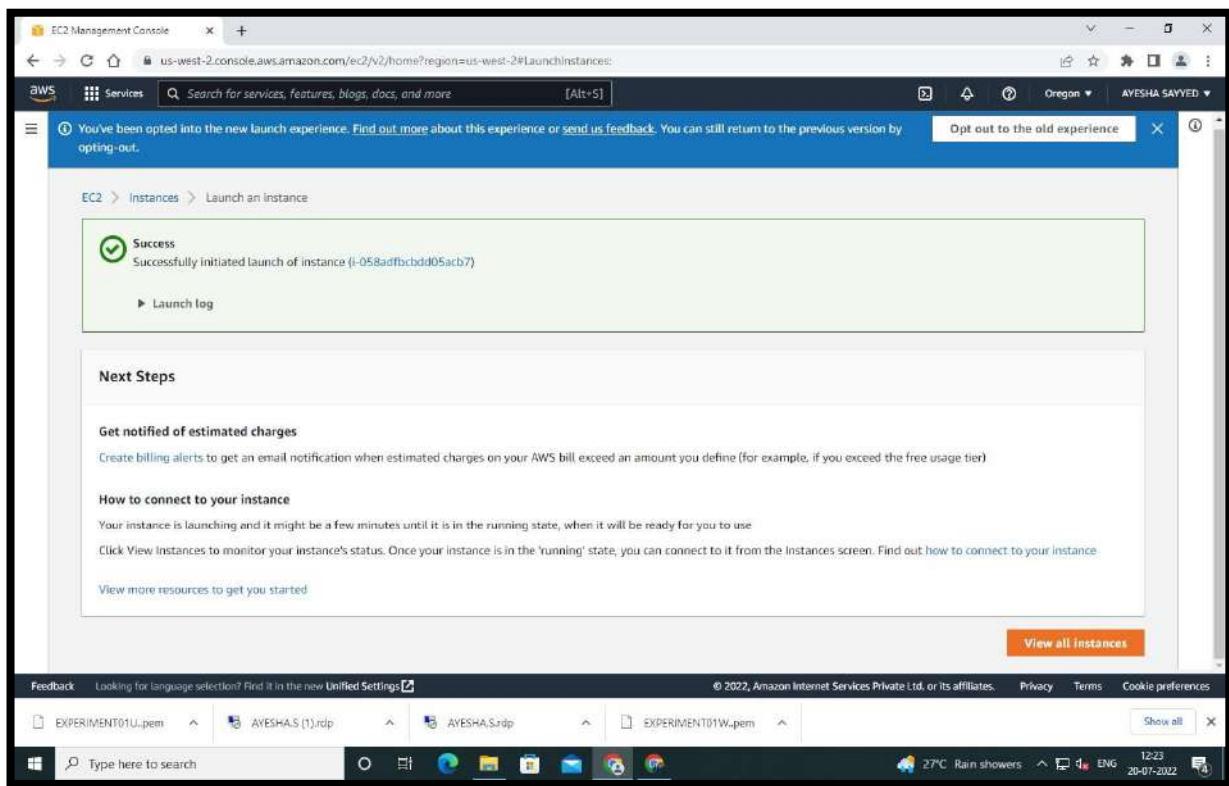
STEP 5 : Create key pair. A .pem file will be downloaded which will be later used to connect to the instance.



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.

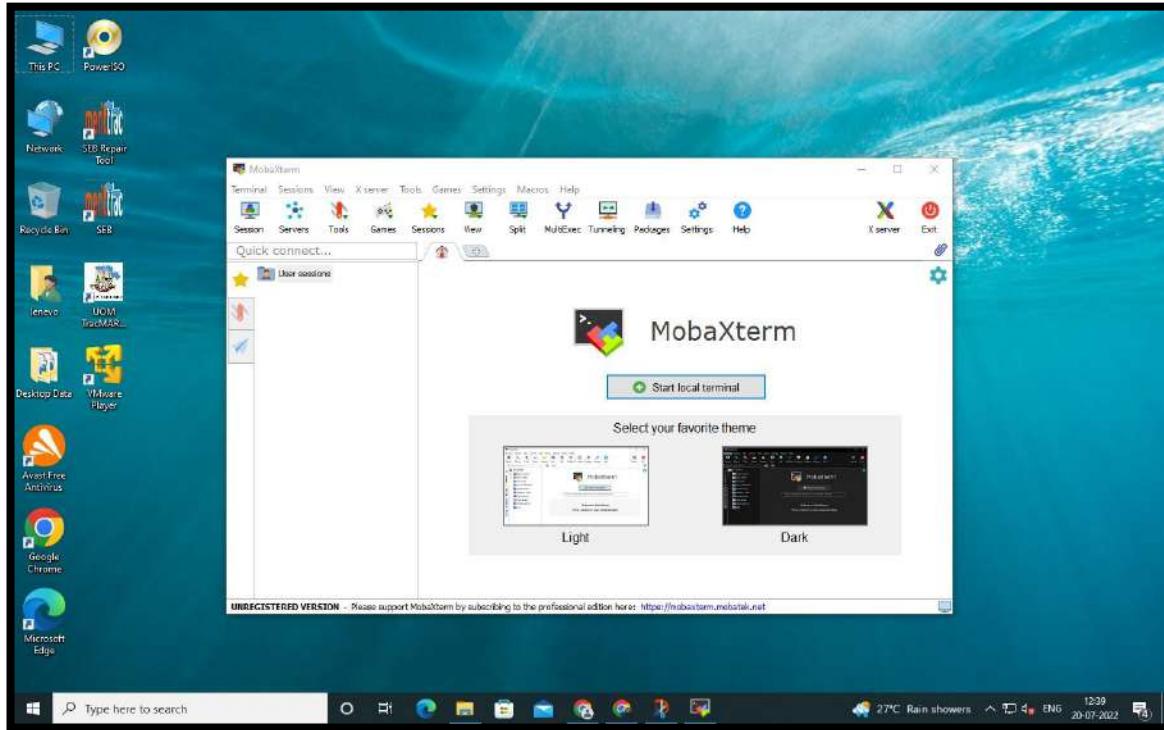
The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with 'Instances' selected. The main area shows a table of instances. One instance, 'S.AYESHA', is currently running. A modal window is open for this instance, displaying its details: Instance ID (i-058adfbcbdd05acb7), Public IPv4 address (34.221.202.91), Instance state (Running), and Private IP DNS name (ec2-34-221-202-91.us-west-2.compute.amazonaws.com). The browser taskbar at the bottom has several tabs open, including '.pem' files and RDP connections.

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

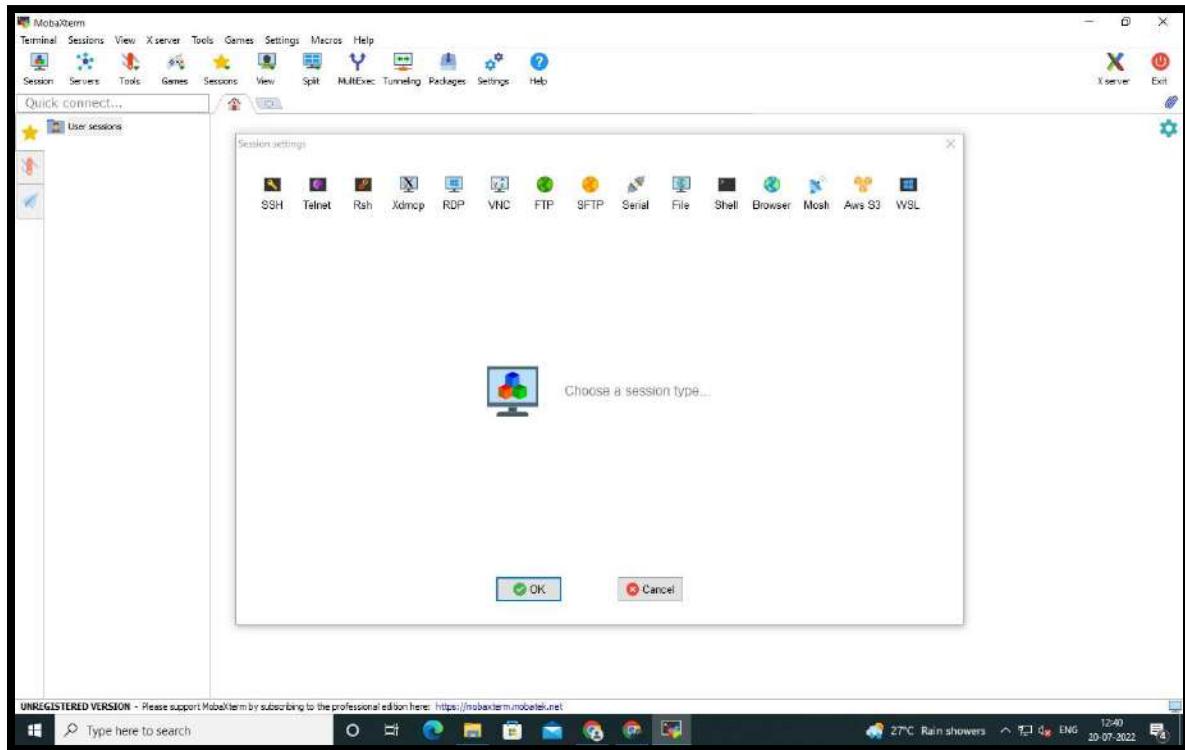
The screenshot shows the 'Connect to instance' dialog in the AWS EC2 Management Console. The 'SSH client' tab is active. It contains instructions for connecting to the instance using an SSH client, including the instance ID (i-058adfbcbdd05acb7) and the command to run (ssh -i "EXPERIMENT01U.pem" ubuntu@ec2-34-221-202-91.us-west-2.compute.amazonaws.com). A note in the dialog box says: '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 browser taskbar at the bottom has several tabs open, including '.pem' files and RDP connections.

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

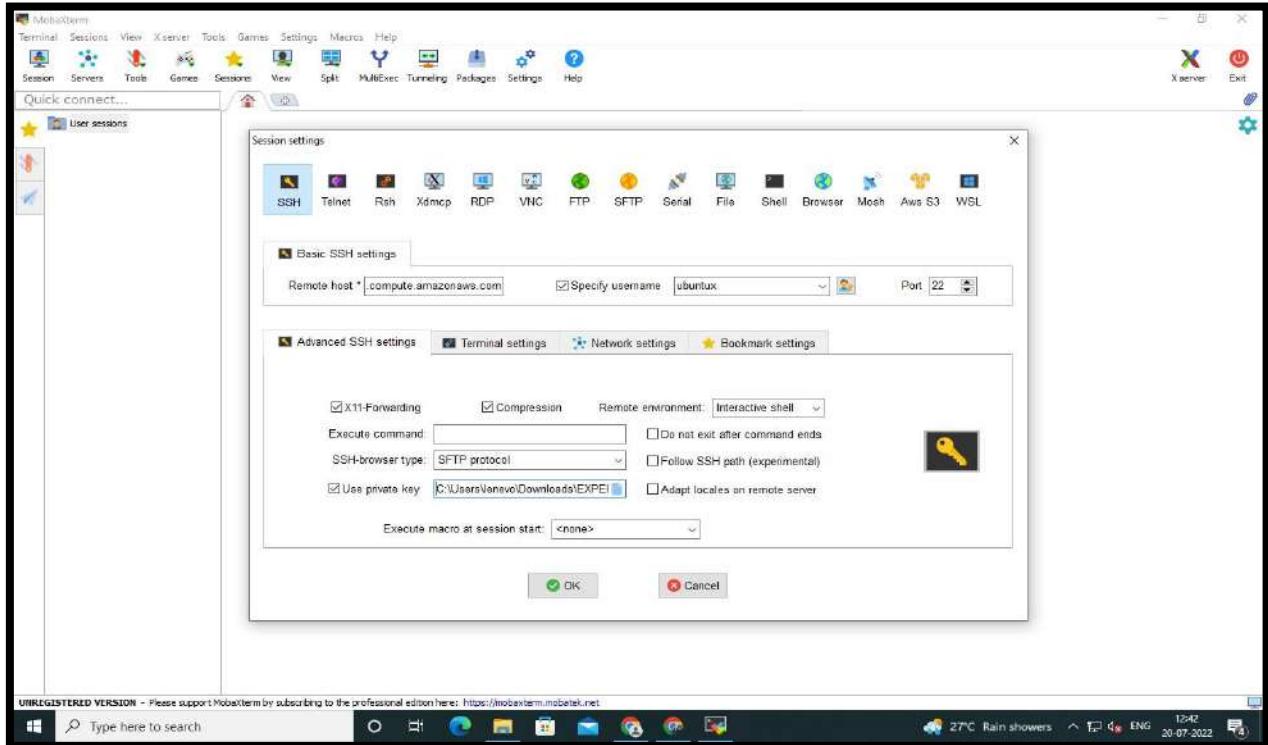
STEP 9 : Download MobaXterm to connect the instance



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 /: 10% of 7.50GB 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.

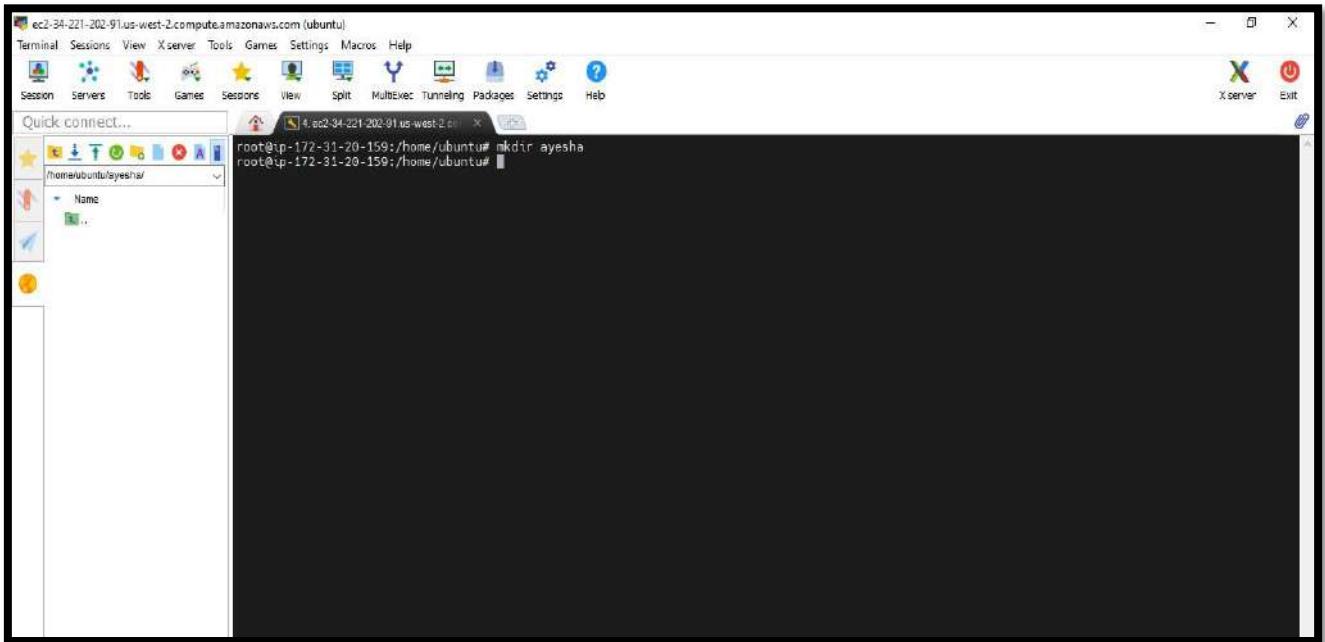
/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user 'root'), use "sudo <command>".
See "man sudo_root" for details.

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

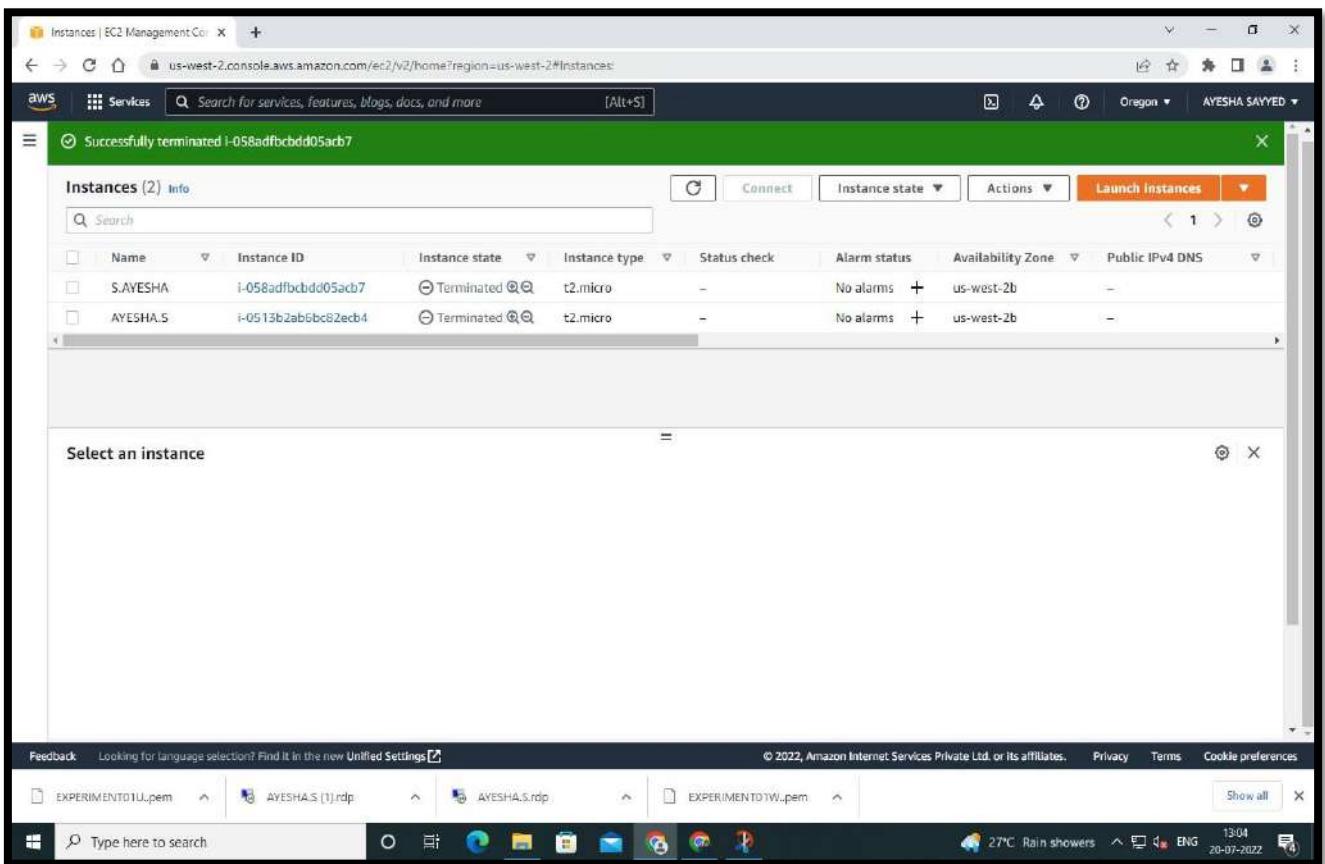
```
ec2-34-221-202-91.us-west-2.compute.amazonaws.com (ubuntu)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu
Name
.. cache ssh bash_logout bashc profile .Xauthority
Reading state information... Done
root@ip-172-31-29-159:/home/ubuntu# sudo apt update
E: Unable to locate package update
root@ip-172-31-29-159:/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 kB]
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 [217 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 B]
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 [104.8 kB]
Get:13 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [6088 B]
Get:14 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [230 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 B]
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 [69.6 kB]
Get:19 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [4160 B]
Get:20 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [2000 B]
Get:21 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [2112 B]
Get:22 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [420 B]
Get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [112 B]
Get:24 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [5412 B]
Get:26 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [8169 B]
Get:27 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [236 B]
Get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
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 B]
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 [38.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [512 B]
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 [33.8 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [2104 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4192 B]
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-172-31-29-159:/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 ... 53612 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
27°C Rain showers 12:49 ENG 20-07-2022
```

```
ec2-34-221-202-91.us-west-2.compute.amazonaws.com (ubuntu)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu
Name
.. cache ssh bash_logout bashc profile .Xauthority
Reading state information... Done
root@ip-172-31-29-159:/home/ubuntu# sudo apt update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [69.6 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [4160 B]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [2112 B]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [420 B]
Get:5 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [112 B]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [5412 B]
Get:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [8169 B]
Get:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [236 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [227 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [55.3 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [3564 B]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [203 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [38.4 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [512 B]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [93.8 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [33.8 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [2104 B]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4192 B]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [900 B]
Get:21 http://us-west-2.ec2.archive.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-172-31-29-159:/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 ... 53612 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
27°C Rain showers 12:50 ENG 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 ?

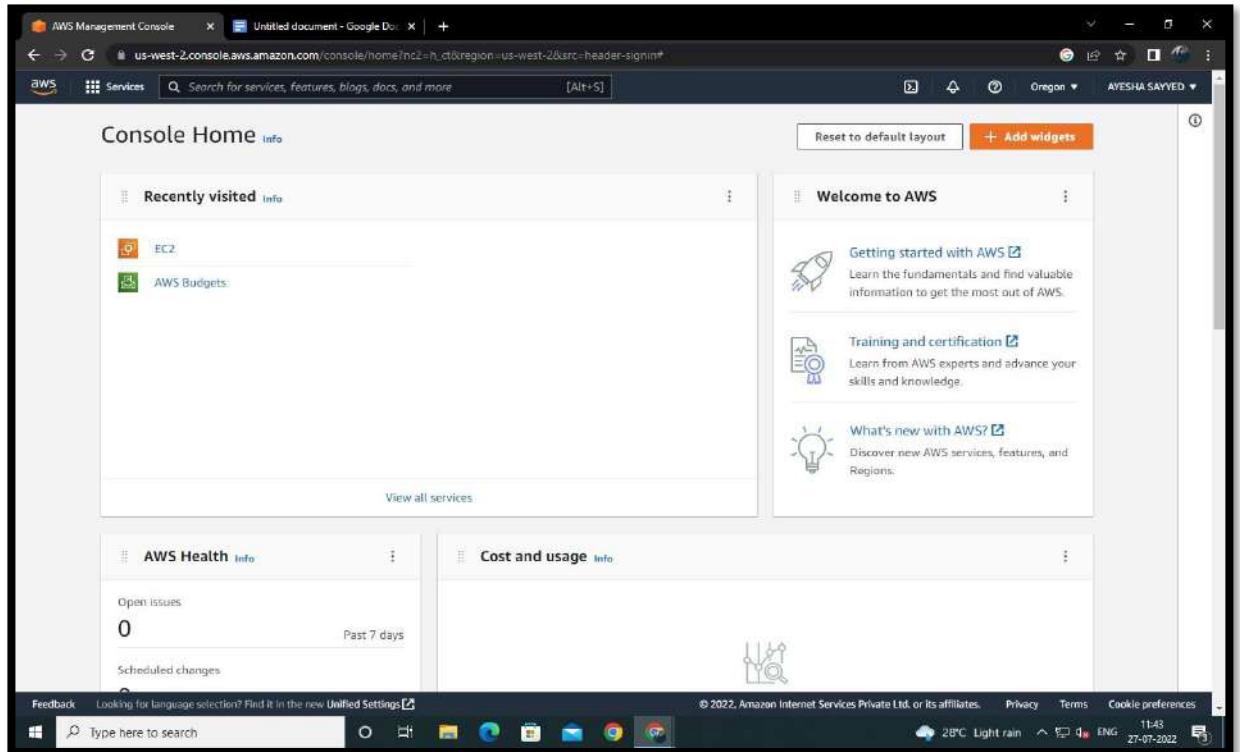
- ▲ Amazon S3 is an object storage service that stores data as objects within buckets.
- ▲ An object is a file and any metadata that describes the file. A bucket is a container for objects.
- ▲ Amazon Simple Storage Service (Amazon S3) is a scalable, high-speed, web-based cloud storage service.
- ▲ The 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: -

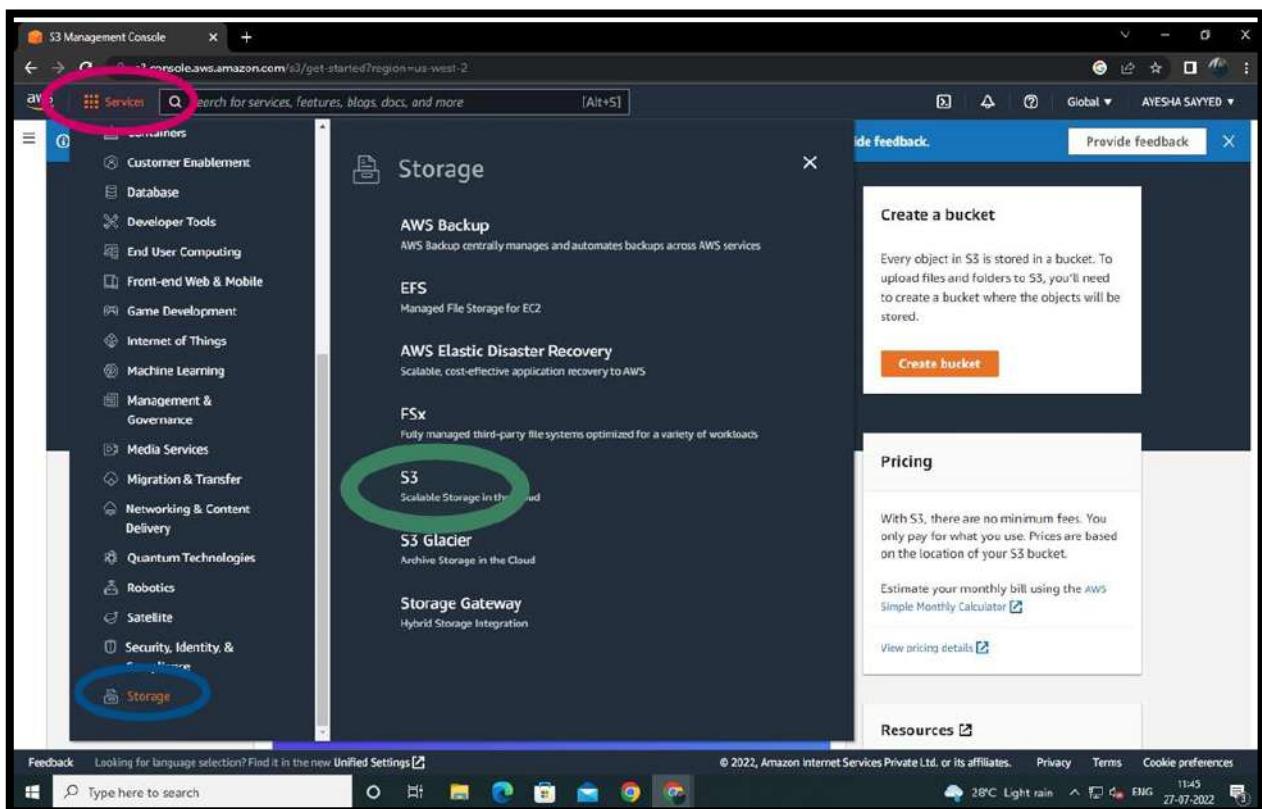
- Data storage
- Data archiving
- Application hosting for deployment, installation and management of web apps
- Software delivery
- Data backup
- Disaster recovery (DR)
- Running big data analytics tools on stored data
- Data lakes
- Mobile applications
- Internet of things (IoT) devices
- Media hosting 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].

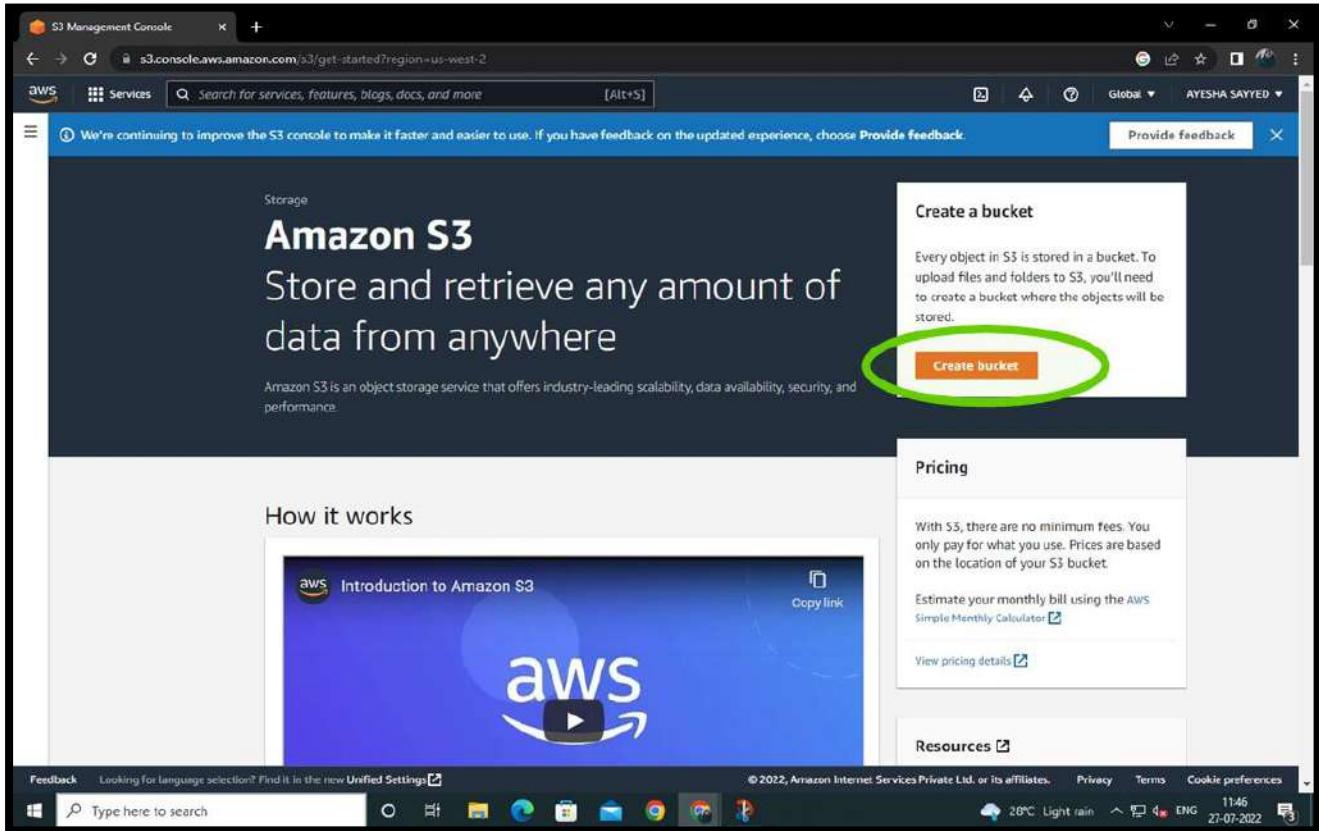
STEP 1: AWS Management Console Dashboard.



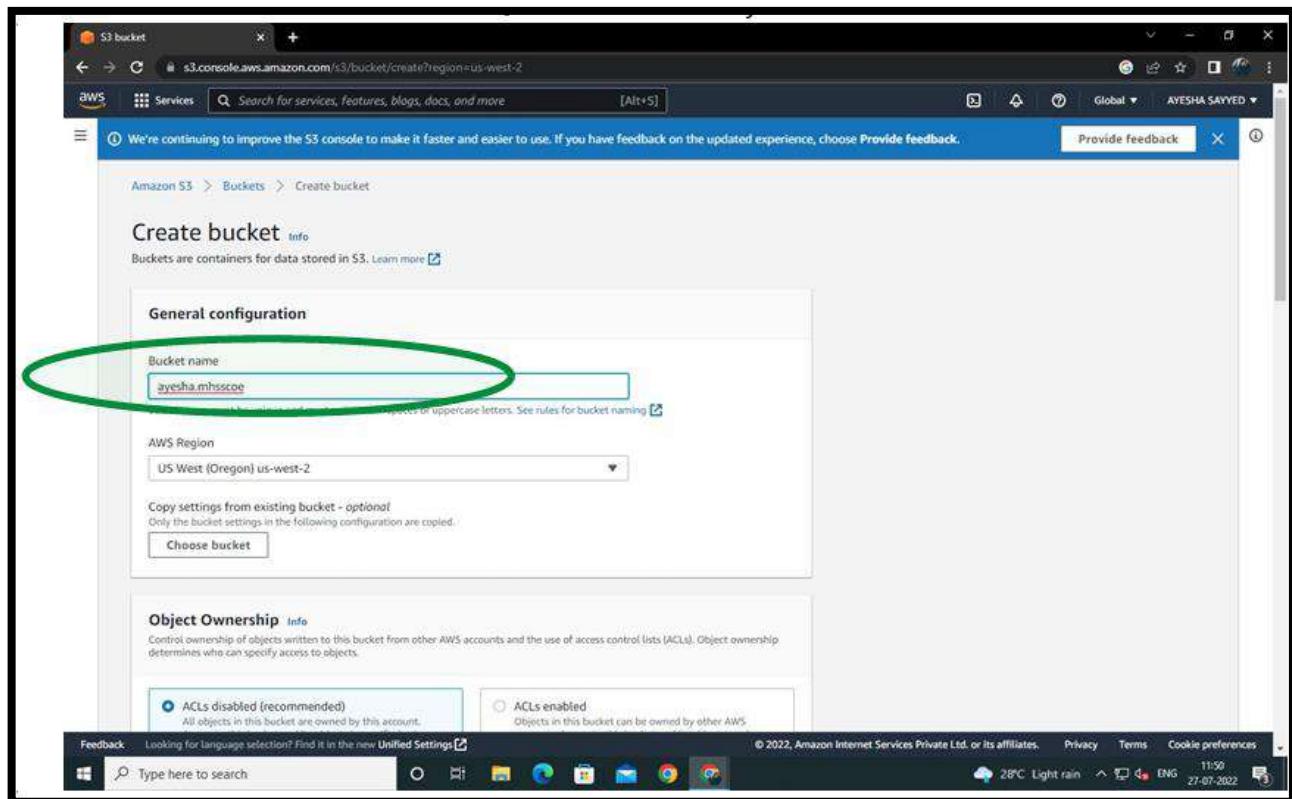
STEP 2: Click on Service -> Storage -> S3



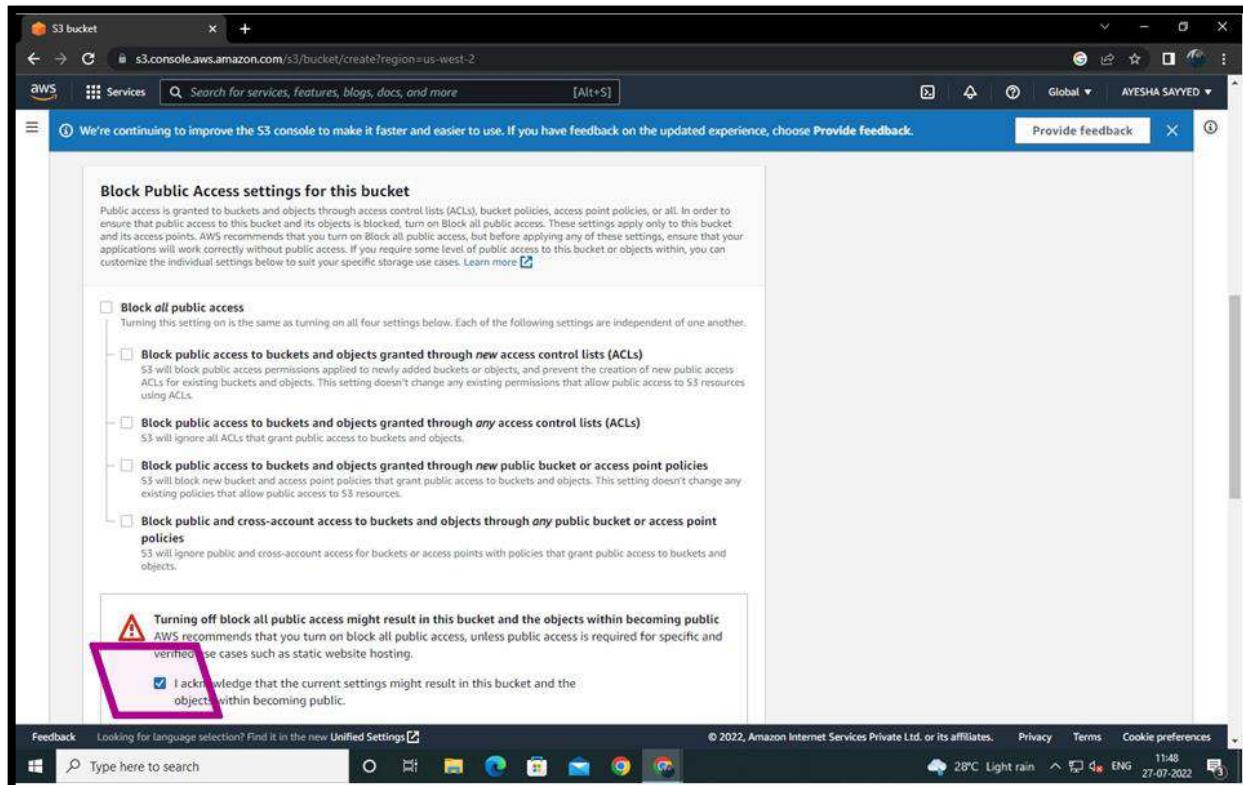
STEP 3: Click on “Create a Bucket”.



STEP 4: Give name to your Bucket.



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



STEP 6: Select the Bucket

The screenshot shows the 'Buckets' section of the AWS S3 Management Console. On the left, there's a sidebar with 'Amazon S3' and 'Storage Lens' sections. The main area displays a table of buckets. One bucket, named 'ayesha.mhsscoe', is highlighted with a blue oval. The table columns include Name, AWS Region, Access, and Creation date. The 'Access' column for this bucket shows 'Public' with a warning icon.

Name	AWS Region	Access	Creation date
ayesha.mhsscoe	US West (Oregon) us-west-2	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 interface. On the left, a sidebar menu includes 'Buckets' (selected), 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens' (with 'Dashboards' and 'AWS Organizations settings'), 'Feature spotlight', and 'AWS Marketplace for S3'. The main content area is titled 'Bucket policy' and contains the message 'No policy to display.' with a 'Copy' button. At the top of this section are 'Edit' and 'Delete' buttons. A note at the top states: '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. Learn more'.

The screenshot shows the 'Edit bucket policy' page. The left sidebar is identical to the previous screenshot. The main area has a title 'Edit bucket policy' with a 'Info' link. It features tabs for 'Policy examples' and 'Policy generator'. Below these are sections for 'Bucket ARN' (set to 'arn:aws:s3:::ayesha.mhsscoe') and 'Policy'. The 'Policy' section displays the following JSON code:

```
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 - }
14 - ]
15 - }
16 }
```

To the right of the policy text, there's a sidebar with 'Edit statement' (set to 'PublicReadGetObject'), 'Remove' (button), 'Add actions', 'Choose a service' (with a 'Filter services' search bar), and two tabs: 'Included' (set to 'S3') and 'Available' (listing 'AMP', 'API Gateway', and 'API Gateway V2').

The screenshot shows the AWS S3 console for the bucket 'ayesha.mhsscoe'. A yellow circle highlights the top banner which displays two success messages: 'Successfully edited bucket policy.' and 'Follow security best practices for S3.' Below the banner, the 'Permissions' tab is selected in the navigation bar. The 'Permissions overview' section shows 'Access' set to 'Public'. Under 'Block public access (bucket settings)', there is a note about public access being granted through ACLs, policies, or access points. A blue 'Edit' button is visible. The status bar at the bottom indicates it's 27-07-2022, 11:57, and the weather is 28°C Light rain.

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

The screenshot shows the 'Edit static website hosting' page for the 'ayesha.mhsscoe' bucket. A green circle highlights the 'Static website hosting' section where the 'Enable' option is selected. A red circle highlights the 'Hosting type' section where 'Host a static website' is selected. A callout box provides instructions for enabling public access. Below these sections are fields for 'Index document' (set to 'index.html') and 'Error document - optional' (set to 'error.html'). The status bar at the bottom indicates it's 27-07-2022, 11:59, and the weather is 28°C Light rain.

The screenshot shows the AWS S3 console for the bucket 'ayesha.mhsscoe'. A green success message at the top states 'Successfully edited static website hosting.' Below it, a blue bar says 'Follow security best practices for S3.' The main area displays 'Bucket overview' with details: 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)). Under 'Bucket Versioning', it shows 'Bucket Versioning' status as 'Disabled'. At the bottom, there's a 'Feedback' link and a search bar.

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

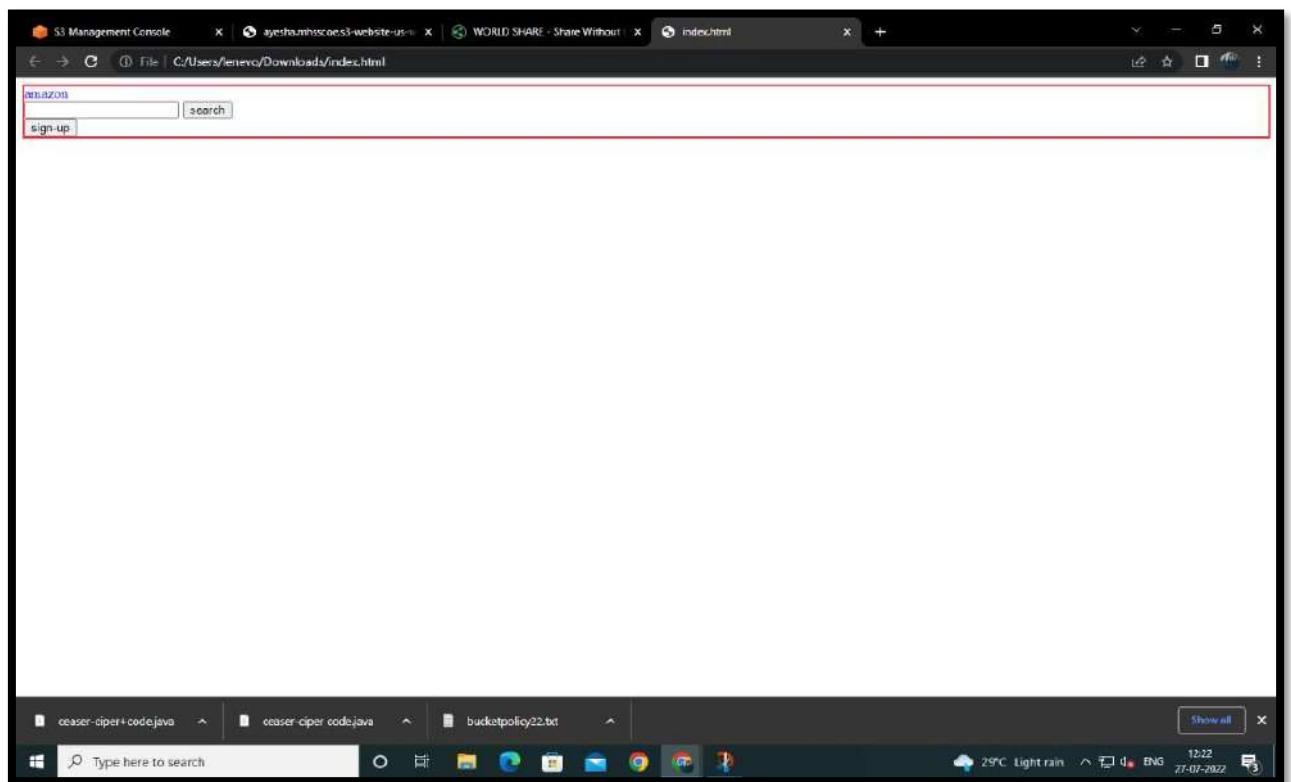
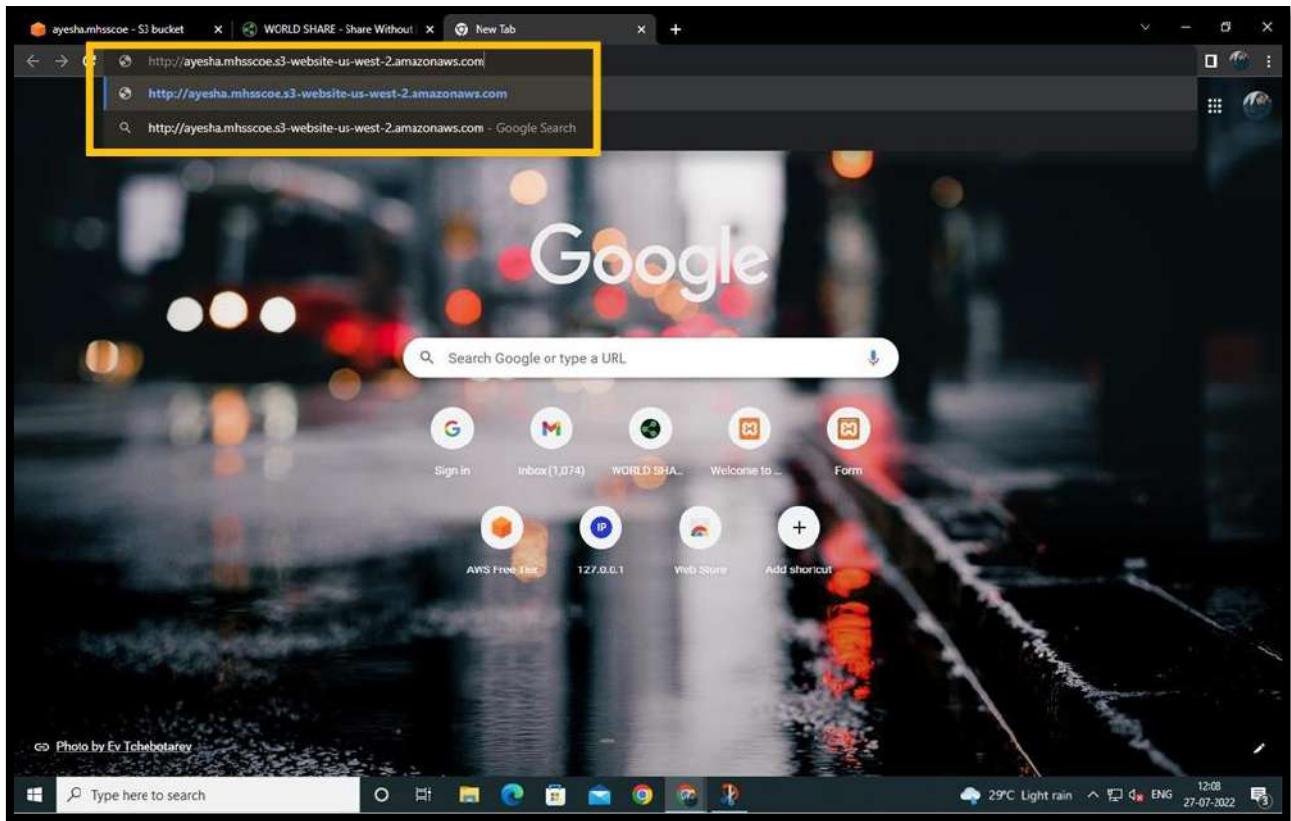
The screenshot shows the AWS S3 Management Console. In the 'Files and folders' section, a file named 'index.html' is listed with a size of 451.0 B and type text/html. This file is highlighted with a green oval. The 'Destination' section shows the destination as 's3://ayesha.mhsscoe'. The 'Permissions' 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 in a web browser. A green success message at the top left reads "Upload succeeded" with a link "View details below". Below this, a modal window titled "Upload: status" displays a summary table. The table has three columns: "Destination" (s3://ayesha.mhsscoe), "Succeeded" (1 file, 451.0 B (100.00%)), and "Failed" (0 files, 0 B (0%)). At the bottom of the modal, there are tabs for "Files and folders" (selected) and "Configuration". Under "Files and folders", a table lists "index.html" with details: Type: text/html, Size: 451.0 B, Status: Succeeded. The browser's address bar shows the URL s3.console.aws.amazon.com/s3/upload/ayesha.mhsscoe?region=us-west-2. The Windows taskbar at the bottom indicates 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 "ayesha.mhsscoe - S3 bucket" properties page in the AWS Management Console. The "Static website hosting" section is highlighted with a purple oval. It shows "Static website hosting" set to "Enabled" and "Hosting type" set to "Bucket hosting". Below this, the "Bucket website endpoint" field contains the URL "http://ayesha.mhsscoe.s3-website-us-west-2.amazonaws.com", which is also copied to the clipboard icon. The browser's address bar shows the URL s3.console.aws.amazon.com/s3/buckets/ayesha.mhsscoe?region=us-west-2&tab=properties. The Windows taskbar at the bottom indicates the date and time as 27-07-2022 12:07.

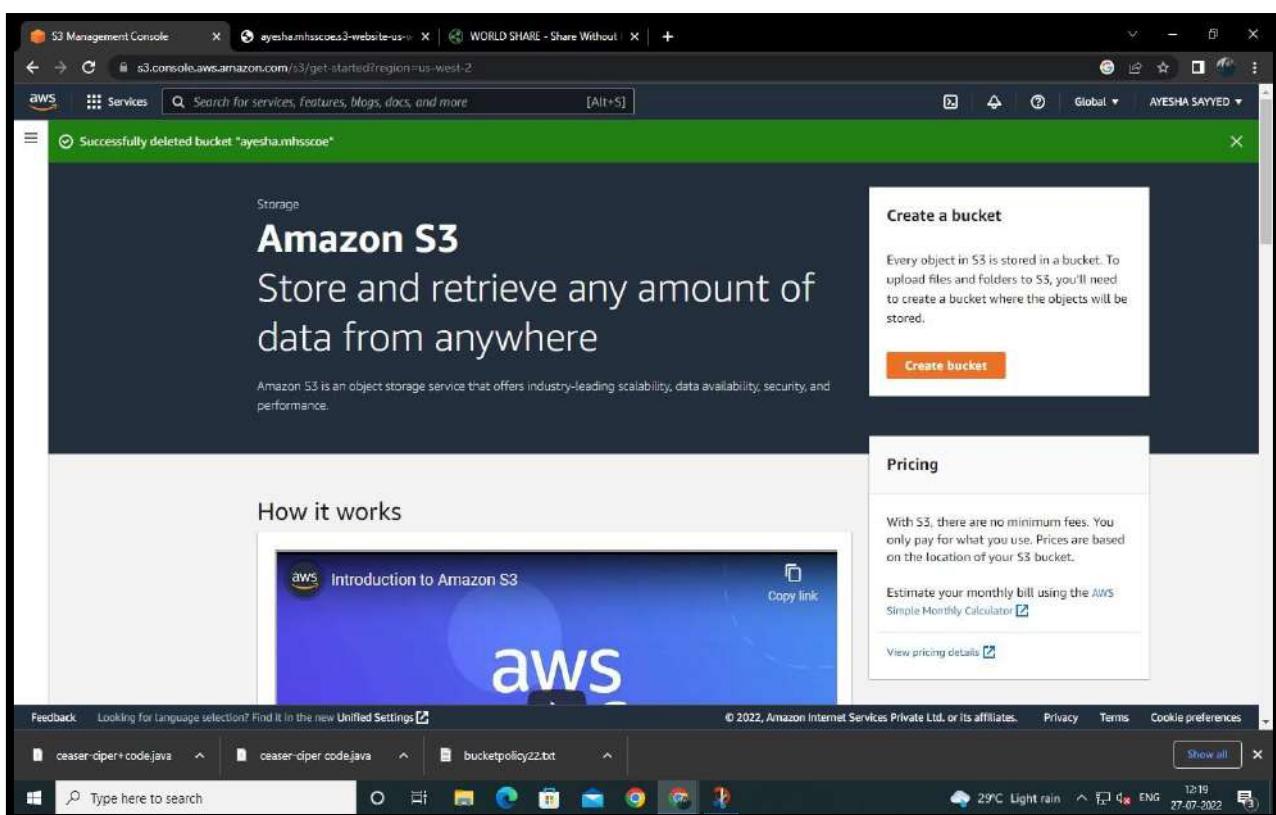
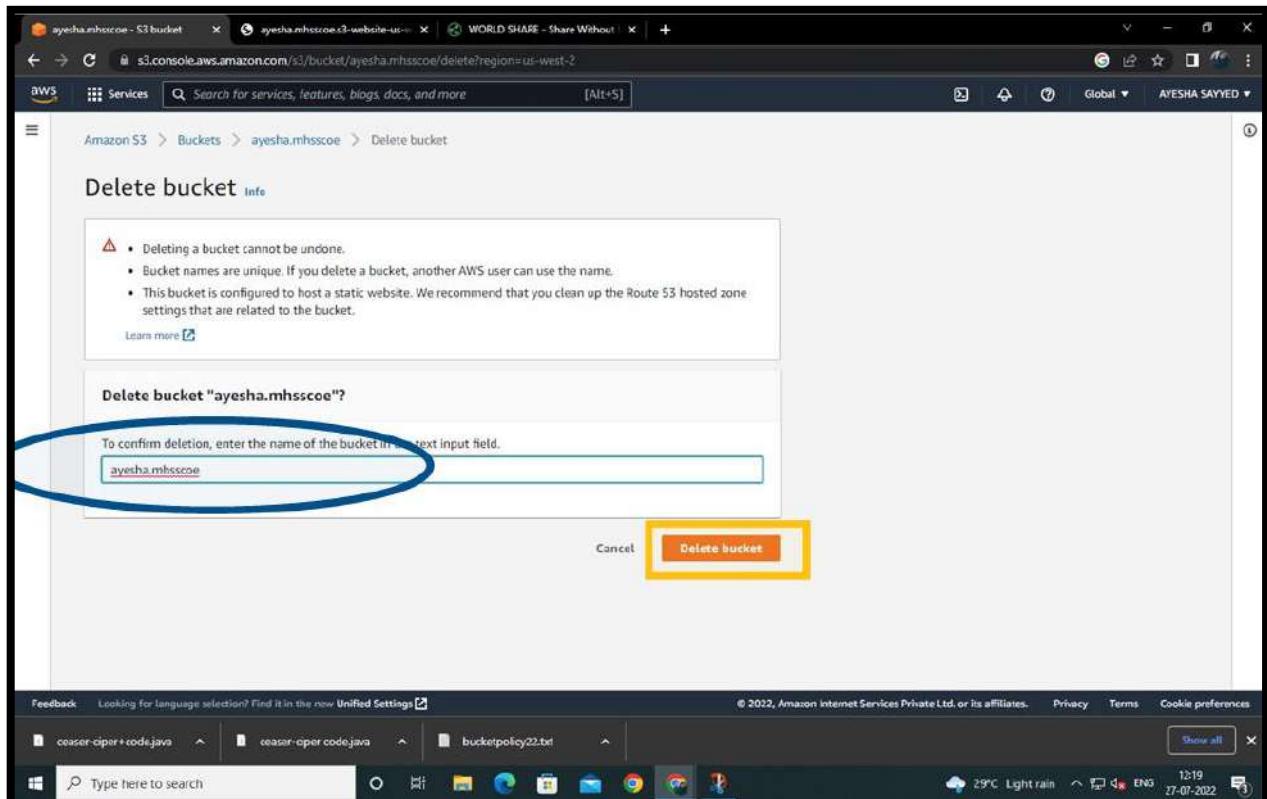


STEP 11: After closing the web page, go to buckets and delete all object from the bucket.

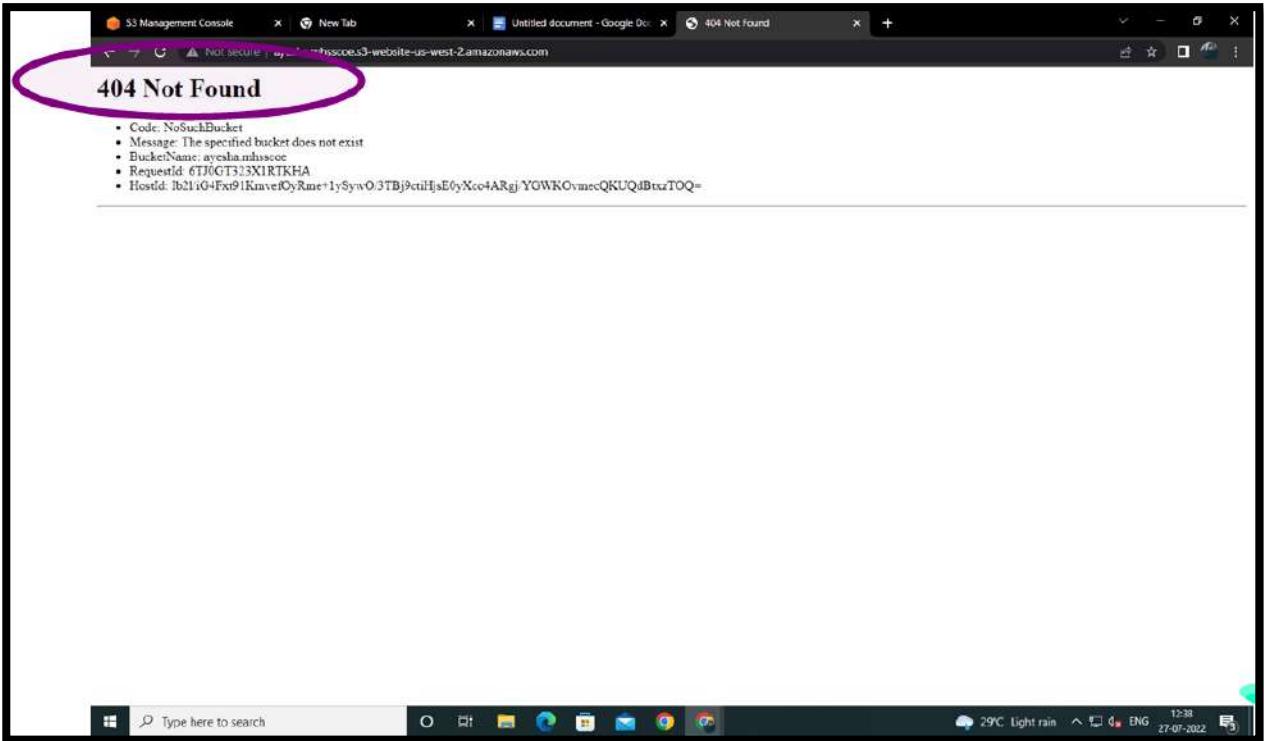
The screenshot shows the 'Delete objects' dialog box in the AWS S3 console. At the top, a warning message states: 'Deleting the specified objects can't be undone.' Below this is a table titled 'Specified objects' showing two files: 'ceaser-ciper code.java' (Type: java, Last modified: July 27, 2022, 12:10:02 (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'. The 'Delete objects' button is visible at the bottom right of the dialog. The background shows the Windows taskbar with several open browser tabs and system icons.

The screenshot shows the 'Delete objects: status' dialog box. At the top, a green banner indicates 'Successfully deleted objects' with a link 'View details below.' Below this, a summary table shows the results: 'Source s3://ayesha.mhsscoe' under 'Successfully deleted' (2 objects, 1.6 KB) and 'Failed to delete 0 objects' under 'Failed to delete'. The 'Failed to delete' tab is selected. The background shows the Windows taskbar with several open browser tabs and system icons.

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 ?

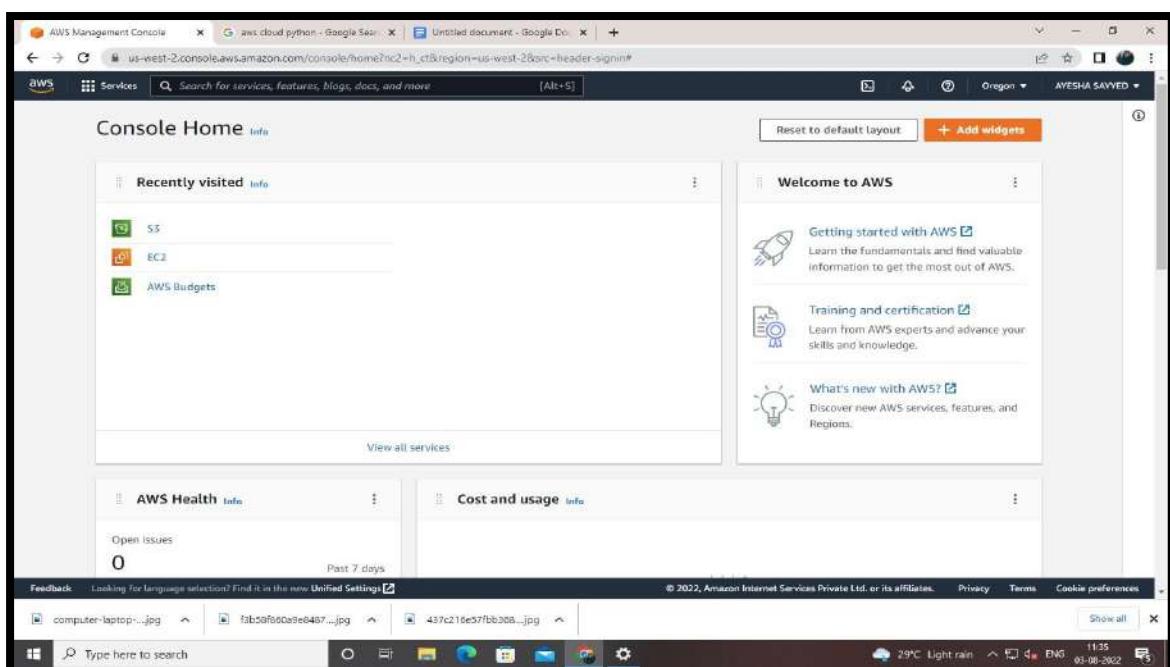
AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you to write, run, and debug code from any machine with just a browser.

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

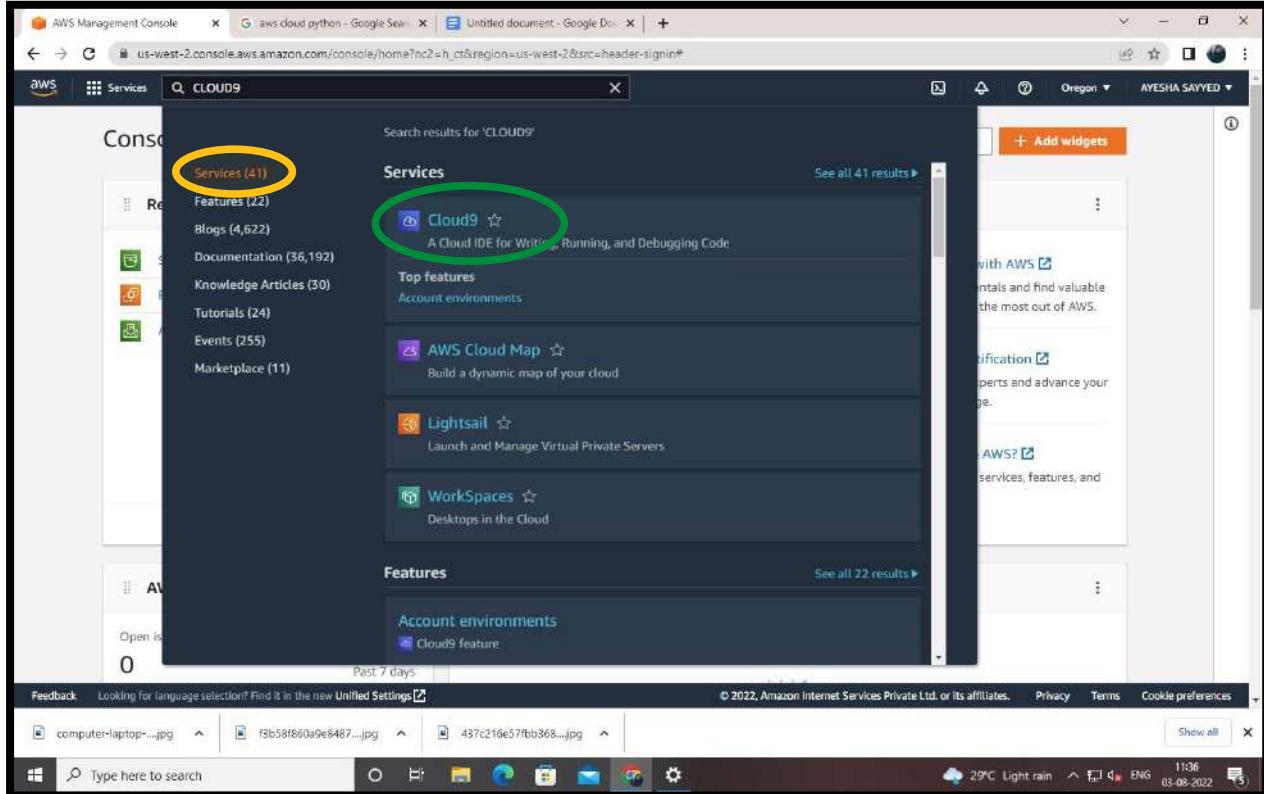
This means you no longer have to spend the time to install programs or configure your development machine. Features of AWS Cloud9 are :-

- ▲ 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.

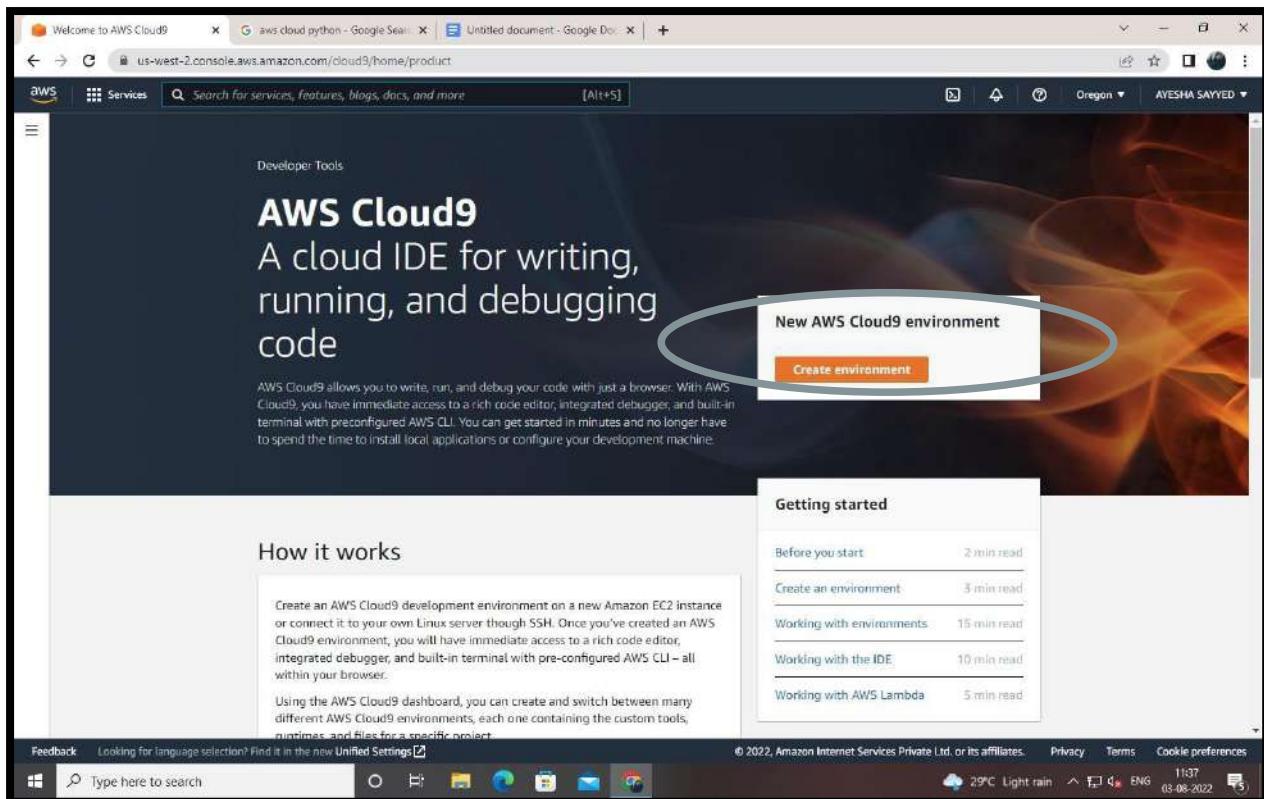


STEP 2: Search for Cloud9 and select it.



The screenshot shows the AWS Management Console search results for 'CLOUD9'. The left sidebar has categories like Services (41), Features (22), Blogs (4,622), Documentation (36,192), Knowledge Articles (30), Tutorials (24), Events (255), and Marketplace (11). The main search results are under 'Services'. The 'Cloud9' service card is highlighted with a green oval. It has a star icon, the name 'Cloud9', and the description 'A Cloud IDE for Writing, Running, and Debugging Code'. Below it are other services: AWS Cloud Map, Lightsail, and WorkSpaces. There are also sections for 'Top features' and 'Features' (Account environments).

STEP 3: Click on “Create Environment”.

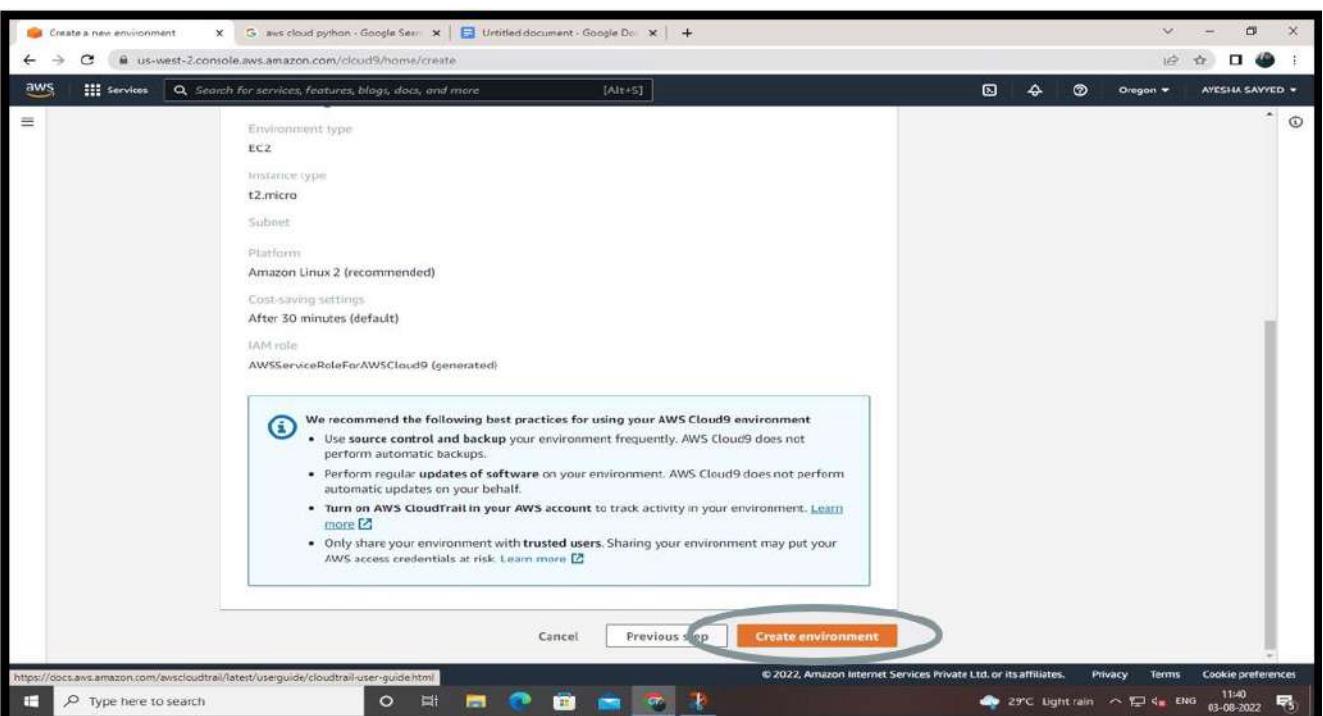
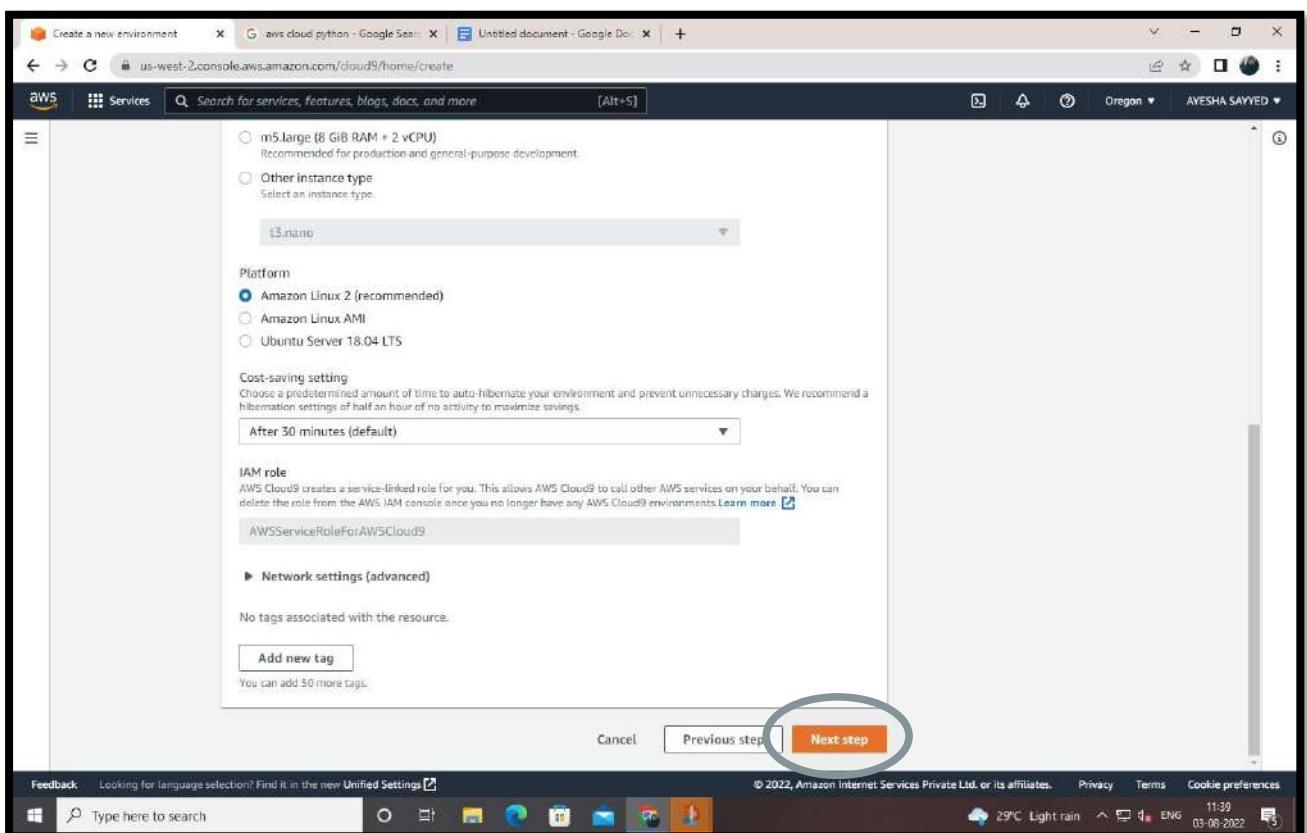


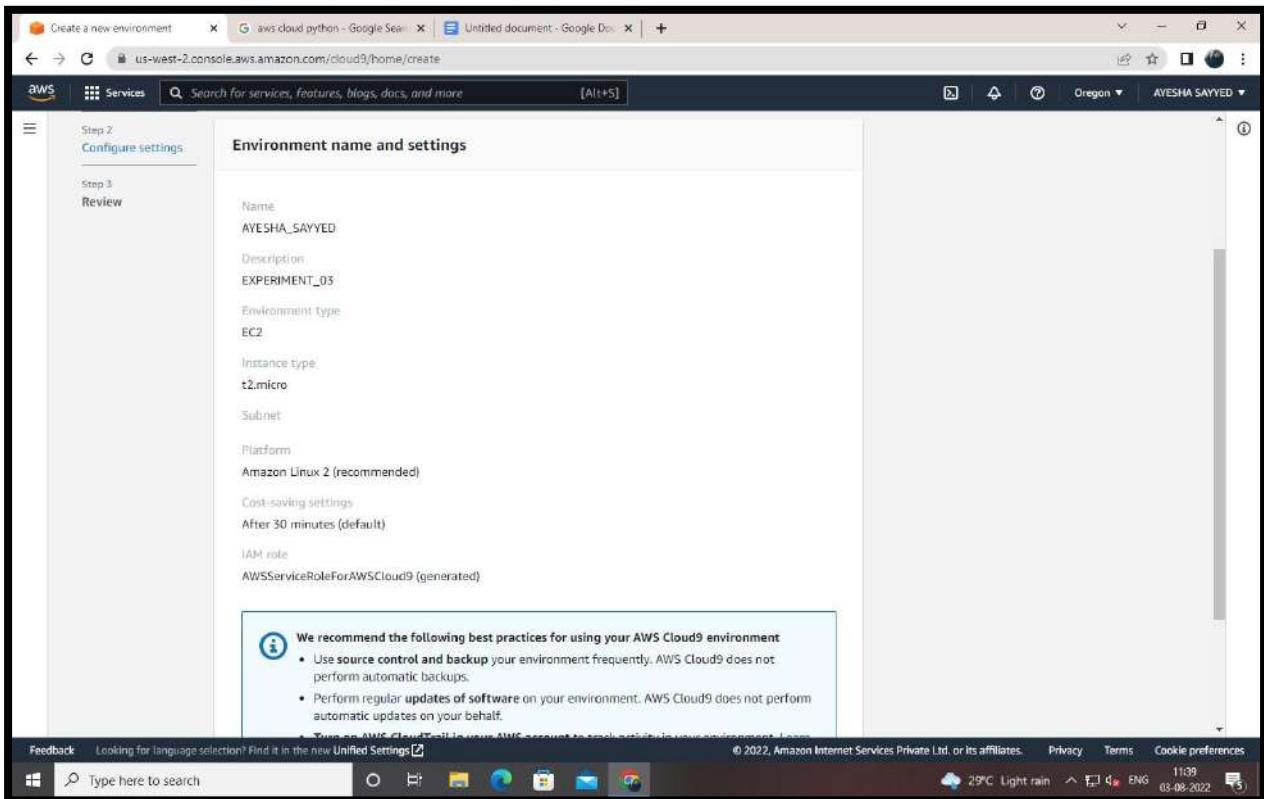
The screenshot shows the 'Welcome to AWS Cloud9' page. The main heading is 'AWS Cloud9: A cloud IDE for writing, running, and debugging code'. Below it, a paragraph explains that AWS Cloud9 allows you to write, run, and debug your code with just a browser. A large call-to-action button labeled 'Create environment' is circled with a grey oval. To the right, there's a 'Getting started' sidebar with links like 'Before you start', 'Create an environment', 'Working with environments', 'Working with the IDE', and 'Working with AWS Lambda'. At the bottom, there's a 'How it works' section with a detailed description of how to create an AWS Cloud9 development environment.

STEP 4: Name your environment and configure the settings.

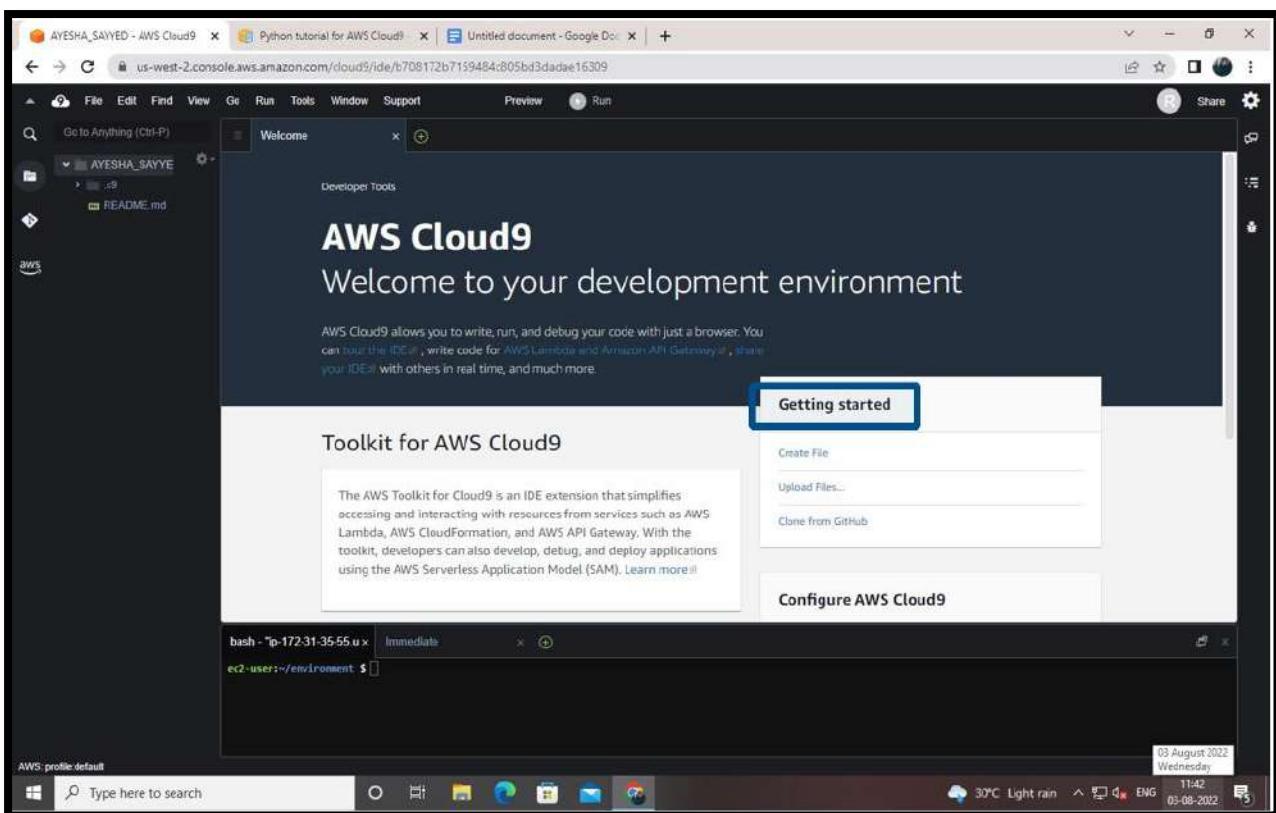
The screenshot shows the 'Name environment' step of the AWS Cloud9 creation wizard. The 'Name' field is highlighted with a red box and contains the value 'AYESHA_SAYYED'. The 'Description' field contains the value 'EXPERIMENT_03'. The 'Next step' button at the bottom right is circled in red.

The screenshot shows the 'Configure settings' step of the AWS Cloud9 creation wizard. The 'Environment type' section is set to 'Create a new EC2 instance for environment (direct access)'. The 'Instance type' dropdown is set to 't3.nano'. The 'Platform' section is set to 'Amazon Linux 2 (recommended)'. The 'Cost-saving setting' section is partially visible at the bottom.





Now AWS Cloud9 Development Environment is being Created.



FOR PYTHON:

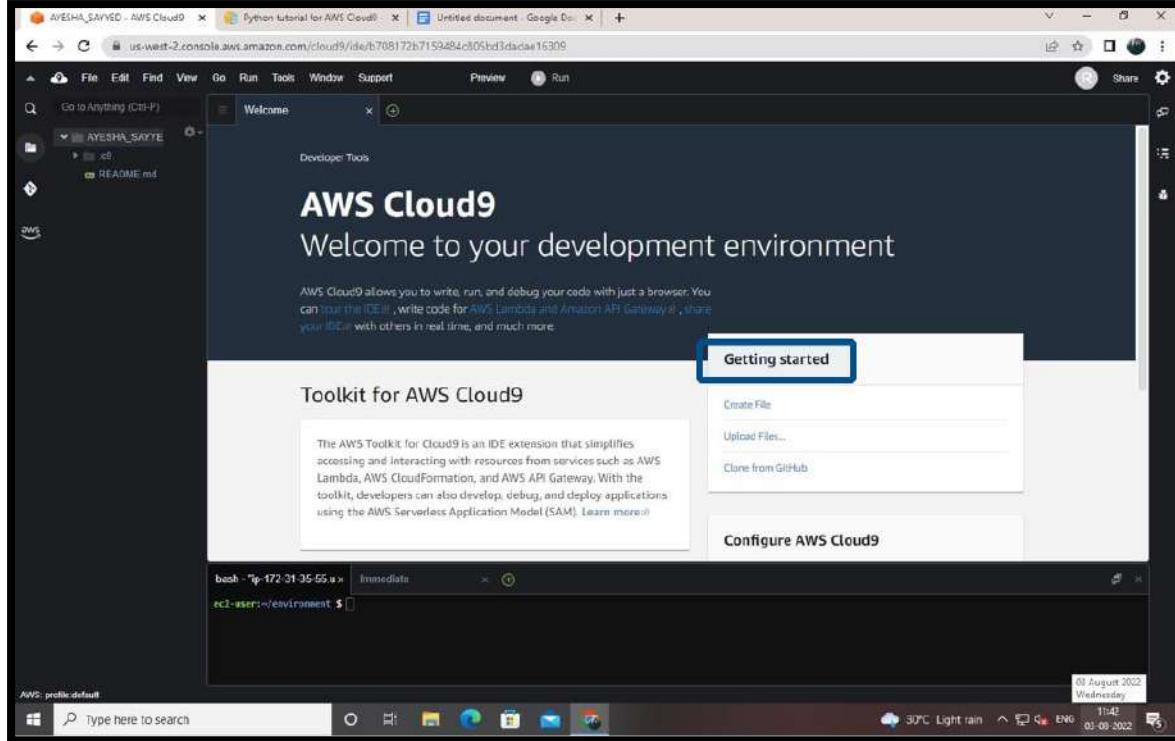
Step 1: Install Python. Run the yum update for Amazon Linux to help ensure the latest security updates and bug fixes are installed:

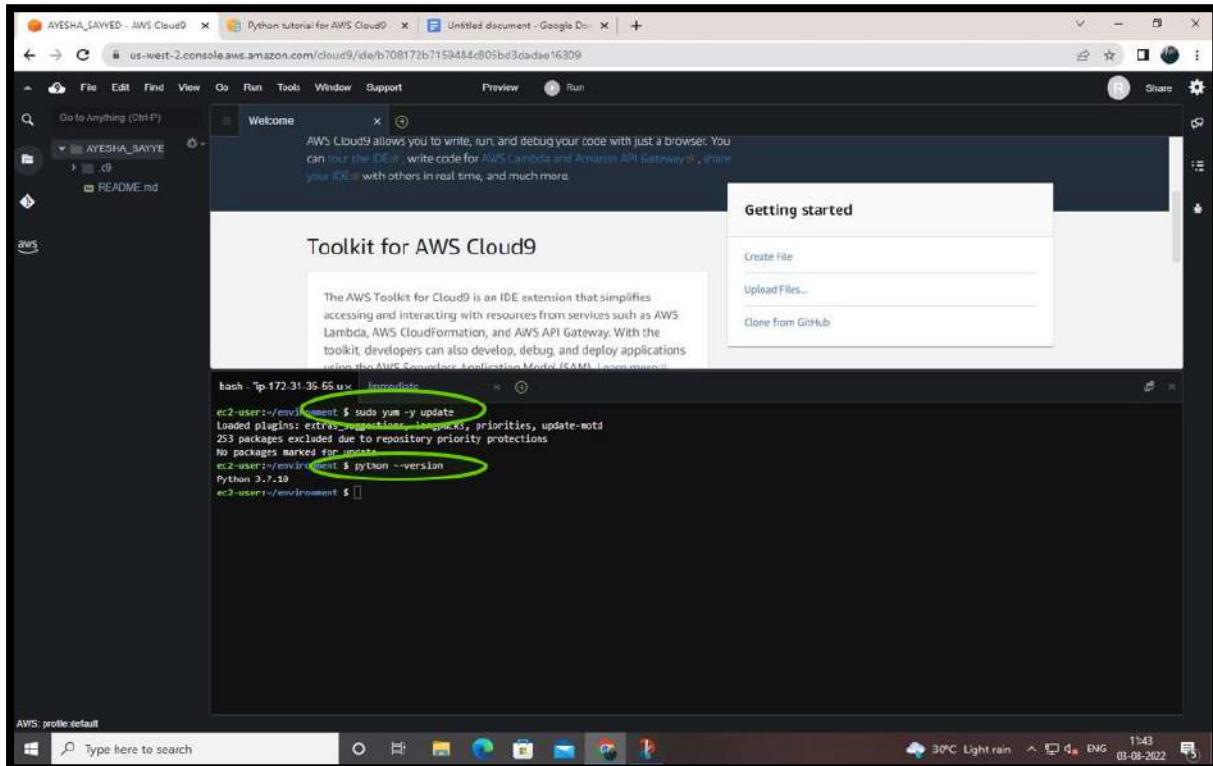
sudo yum -y update Install Python by running the install command. For Amazon Linux:

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

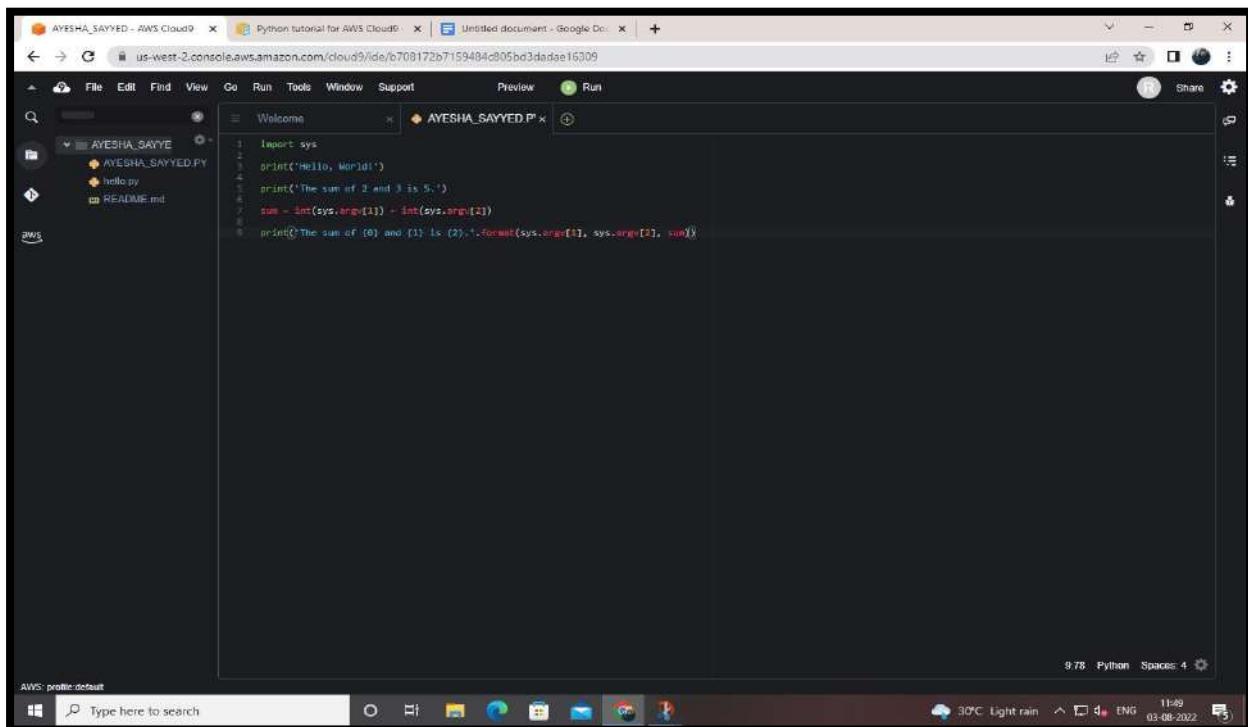
Here, python is already installed so we will check the python version by running the following command:

```
python –version
```



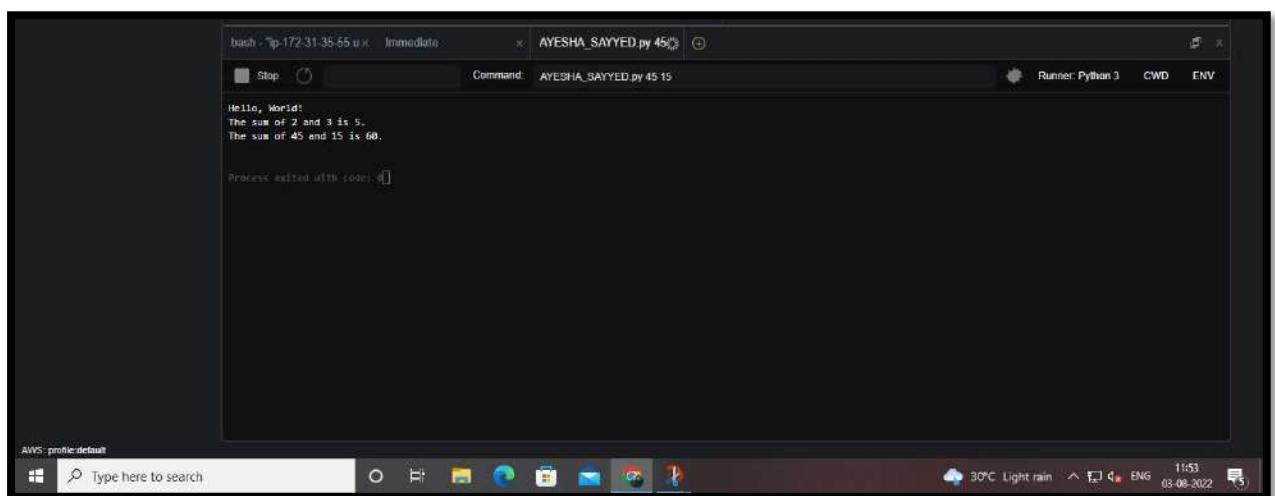
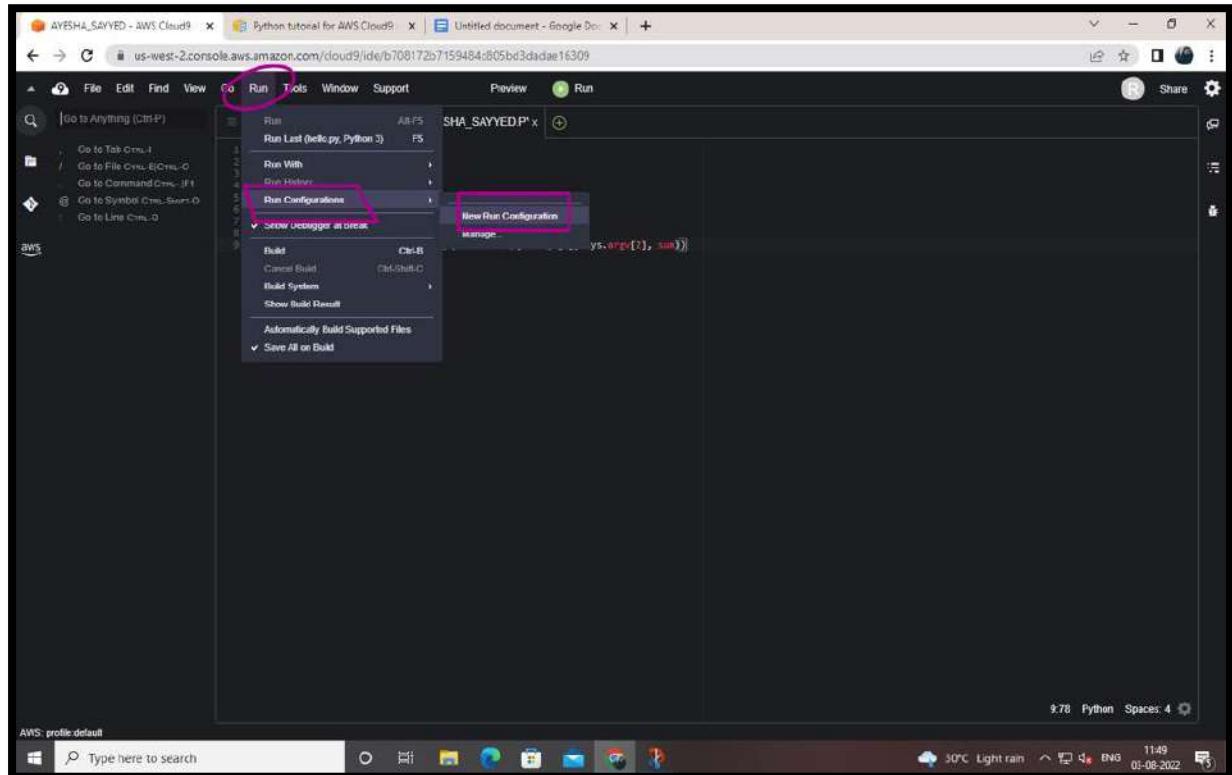


Step 2: Add code In the AWS Cloud9 IDE, create a file with the python code and save the file with some name.



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
[...]
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/local/lib/python3.7/site-packages (from boto3) (1.0.1)
Requirement already satisfied: botocore<1.28.0,>=1.27.44
Requirement already satisfied: s3transfer<0.7.0,>=0.6.8
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<1.27.0,>=1.25.4 in /usr/local/lib/python3.7/site-packages (from botocore<1.28.0,>=1.27.44->boto3) (1.26.1)
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 $ 

```

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.

```

ayesha45.py x
[...]
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)

if __name__ == "__main__":
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('bucket_name', help='The name of the bucket to create.')
    args = parser.parse_args()
    ayesha45.py mhsscoe45test-bucket us-west-2

```

The screenshot shows a browser-based code editor interface. The top navigation bar includes tabs for 'AYEHA_SAYYED - AWS Cloud9', 'Python tutorial for AWS Cloud9', and 'Untitled document - Google Doc'. The main area displays a file named 'ayesha45.py' with the following code:

```
#!/usr/bin/python3

# This demo shows how to create and delete buckets using the Boto3 library.

import boto3
from botocore.exceptions import ClientError

def create_and_delete_my_bucket(s3_resource, bucket_name, keep_bucket):
    try:
        print(f"Creating bucket {bucket_name}...")
        s3_resource.create_bucket(Bucket=bucket_name)
        if not keep_bucket:
            print(f"Deleting bucket {bucket_name}...")
            bucket = s3_resource.Bucket(bucket_name)
            bucket.delete()
            bucket.wait_until_not_exists()
        else:
            print(f"\nKeeping bucket {bucket_name}.")
    except ClientError as e:
        print(f"Exiting the demo.\n{e}")

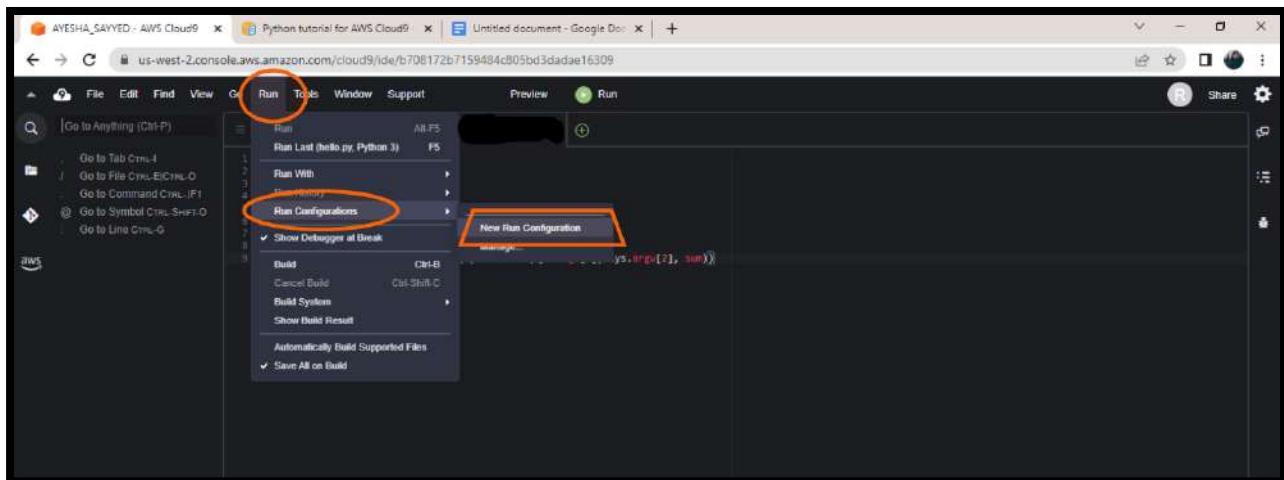
def list_my_buckets(s3_resource):
    for bucket in s3_resource.buckets.all():
        print(f"- {bucket.name}")

if __name__ == '__main__':
    main()
```

The bottom of the screen shows a terminal window titled 'bash - [ip-172-31-35-55.ux]' with the command 'ayesha45.py mhscoo45t' run. The output shows the creation and deletion of a bucket named 'mhscoo45t'. The status bar at the bottom indicates '62:1 Python Spaces:4'.

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 command "ayesha45.py mhssco45test-bucket us-west-2" is run, outputting:

```
Buckets:  
Creating new bucket: mhssco45test-bucket  
Buckets:  
mhssco45test-bucket  
Deleting bucket: mhssco45test-bucket  
Buckets:  
Process exited with code: 0
```

The terminal is part of an AWS Lambda function editor. The status bar at the bottom shows "AWS: profile.default" and a taskbar with various icons.

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

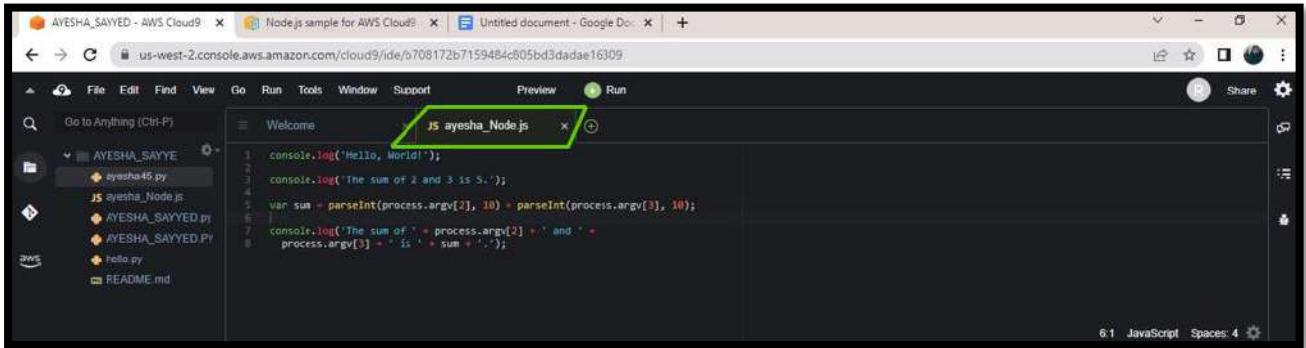
Run this command to install Node.js: nvm install v16.0.0 Here, node.js is already installed.

A screenshot of a terminal window titled "Getting started". The command "sudo yum -y update" is run, outputting:

```
sudo -i -u ec2-user /bin/bash  
Installing collected packages: boto3, s3transfer, botocore  
ec2-user:/var/www/html $ sudo yum -y update  
loaded plugins: extras_repositories, langpacks, priorities, update-motd  
amazon-core  
253 packages excluded due to repository priority protections  
No packages marked for update  
ec2-user:/var/www/html $
```

The terminal is part of an AWS Lambda function editor. The status bar at the bottom shows "AWS: profile.default" and a taskbar with various icons.

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



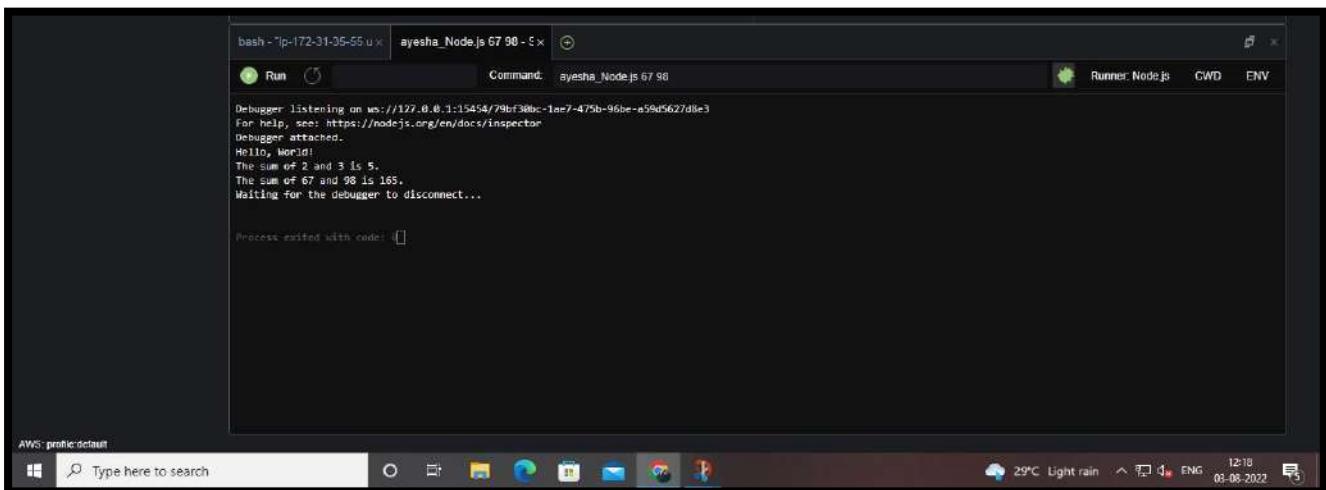
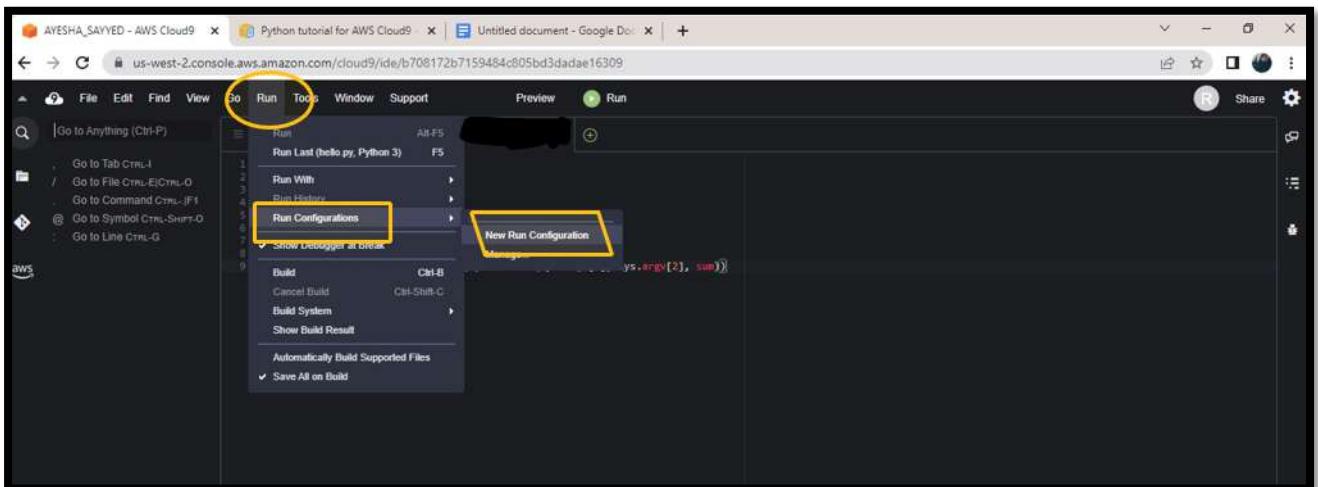
The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a file tree with a folder named 'AYESHA_SAYYE' containing files like 'ayesha4.py', 'JS ayesha_Node.js', 'AYESHA_SAYYED.PY', and 'Hello.py'. The main editor window has a tab titled 'ayesha_Node.js' with the following Node.js code:

```
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, and the output is displayed:

```
Debugger listening on ws://127.0.0.1:15454/79bf30bc-1ae7-475b-96be-a59f5627d1e3
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 right shows '29°C Light rain' and the date '03-08-2022'.

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 -i p-172-31-35-55.ux ayesha Node.js 67.98 -E x +  
ec2-user:~/environment $ npm install aws-sdk  
npm [WARN] deprecate: 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 several tabs open. The active tab is titled 'ayesha_bucket.js' and contains the following Node.js code:

```
if (process.argv.length < 4) {
    console.log('Usage: node s3.js <the bucket name> <the AWS Region to use>\n');
    'Example: node s3.js my-bucket us-east-2';
    process.exit();
}

var AWS = require('aws-sdk'); // To set the AWS credentials and region.
var s3 = new AWS.S3({apiVersion: '2006-03-01'});
var region = process.argv[2];
AWS.config.update({
    region: region
});

var create_bucket_params = {
    Bucket: bucket_name,
    CreateBucketConfiguration: {
        LocationConstraint: region
    }
};

var delete_bucket_params = {Bucket: bucket_name};

// List all of your available buckets in this AWS Region.
function listMyBuckets(callback) {
    s3.listBuckets(function(err, data) {
        if (err) {
            console.log("Error occurred: " + err);
        } else {
            console.log("My buckets now are:\n");
            for (var i = 0; i < data.Buckets.length; i++) {
                console.log(data.Buckets[i].Name);
            }
        }
        callback(err);
    });
}

```

The browser interface includes a sidebar with project files like 'AYESHA_SAYYE', 'AYESHA45.or', 'JS ayesha_bucket.js', 'JS ayesha_Note.js', 'AYESHA_BAYYED.py', 'AYESHA_BAYYED.PY', 'hello.py', and 'README.md'. The bottom status bar shows the hash 'hash = b-172.31.35.65:80' and the file 'ayesha_bucket.js'.

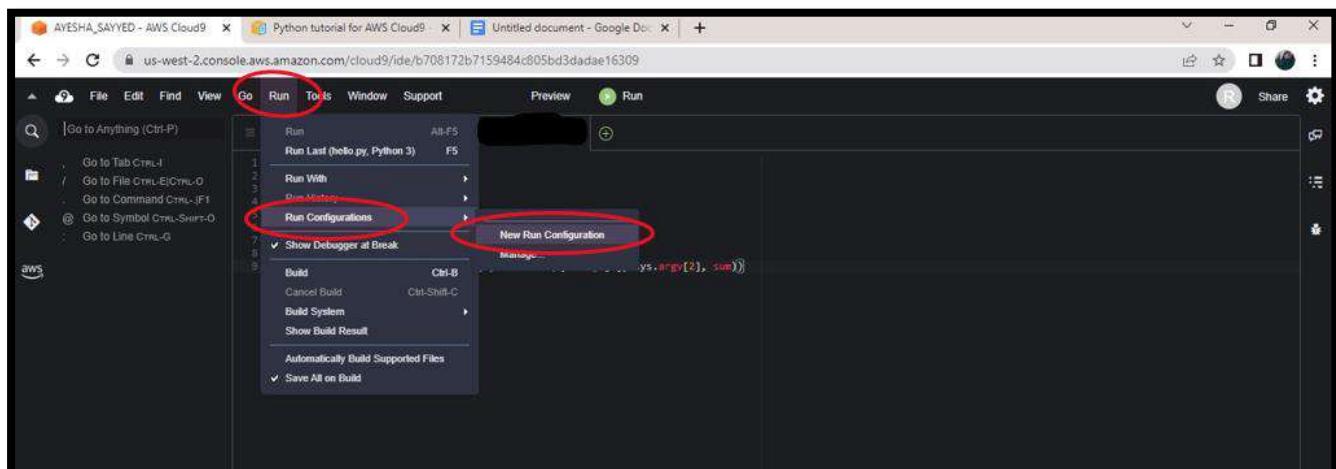
The screenshot shows the AWS Cloud9 IDE interface. The left sidebar displays a file tree with files like 'ayesha45.py', 'ayesha_bucket.js', 'ayesha_Node.js', 'AYESHA_SAYYED.py', 'AYESHA_SAYYED PY', 'hello.py', and 'README.md'. The main editor window contains the following Node.js code:

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

The status bar at the bottom indicates the file is a 'JavaScript' file with 'Spaces: 2'.

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 $
```

```
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://127.0.0.1:15454/9afaf1599-b940-451d-a7ab-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
```

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 12:37 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?

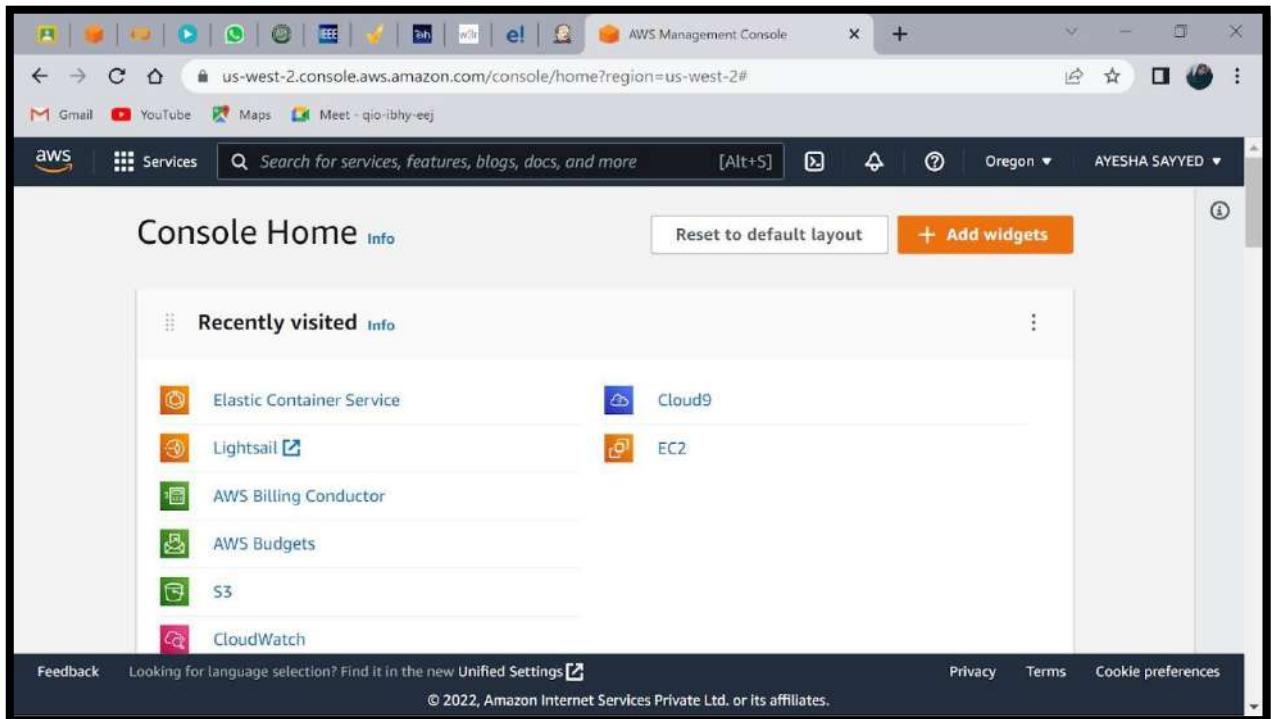
All those databases which are modeled in means other than the tabular relations used in relational databases are known as NO SQL databases. NoSQL key-value databases are the least complicated types of NoSQL databases.

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.



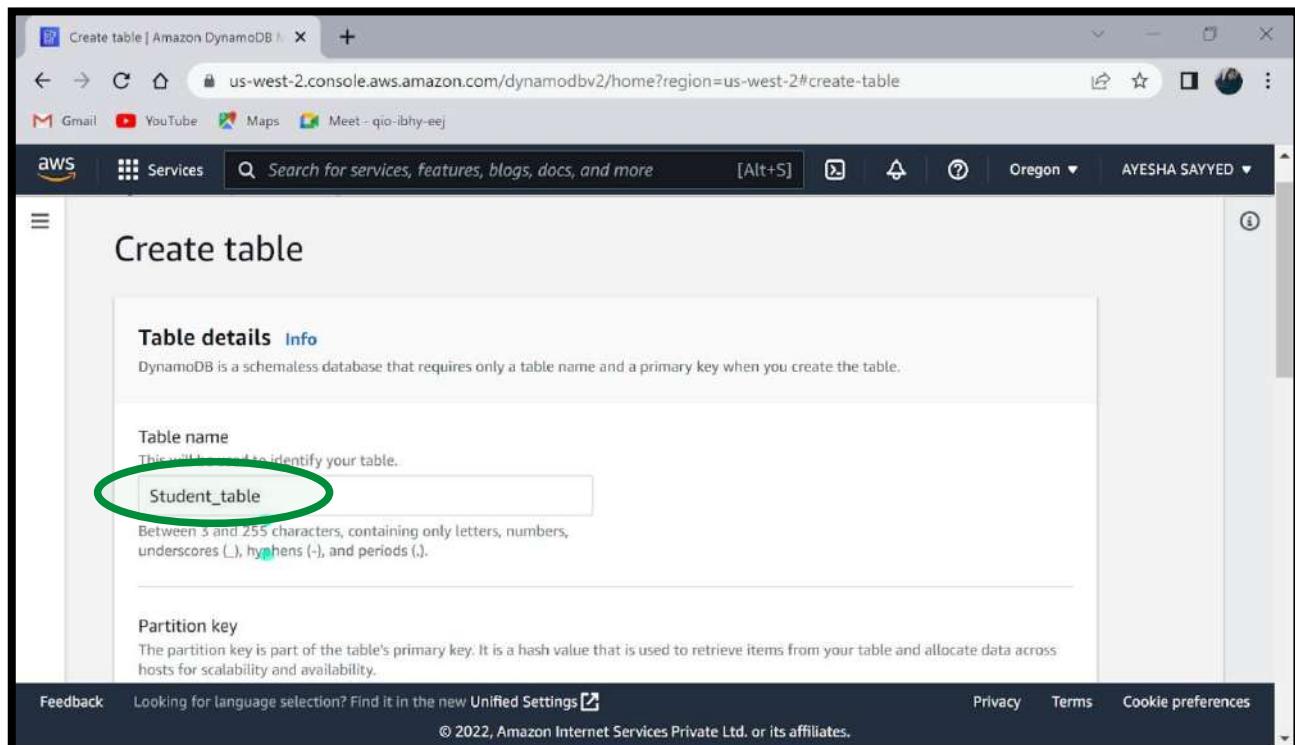
Step 2: Search for DynamoDB and select it.

The screenshot shows the AWS Lambda console interface. At the top, there is a search bar with the text 'dynamo'. Below the search bar, a sidebar on the left lists various AWS services and features. The main area displays search results for 'dynamo', including 'Services (3)', 'Features (6)', 'Blogs (640)', 'Documentation (7,255)', 'Knowledge Articles (28)', 'Tutorials (4)', 'Events (4)', and 'Marketplace (20)'. A specific result, 'DynamoDB Managed NoSQL Database', is highlighted with a large orange circle. The page also includes standard navigation links like 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

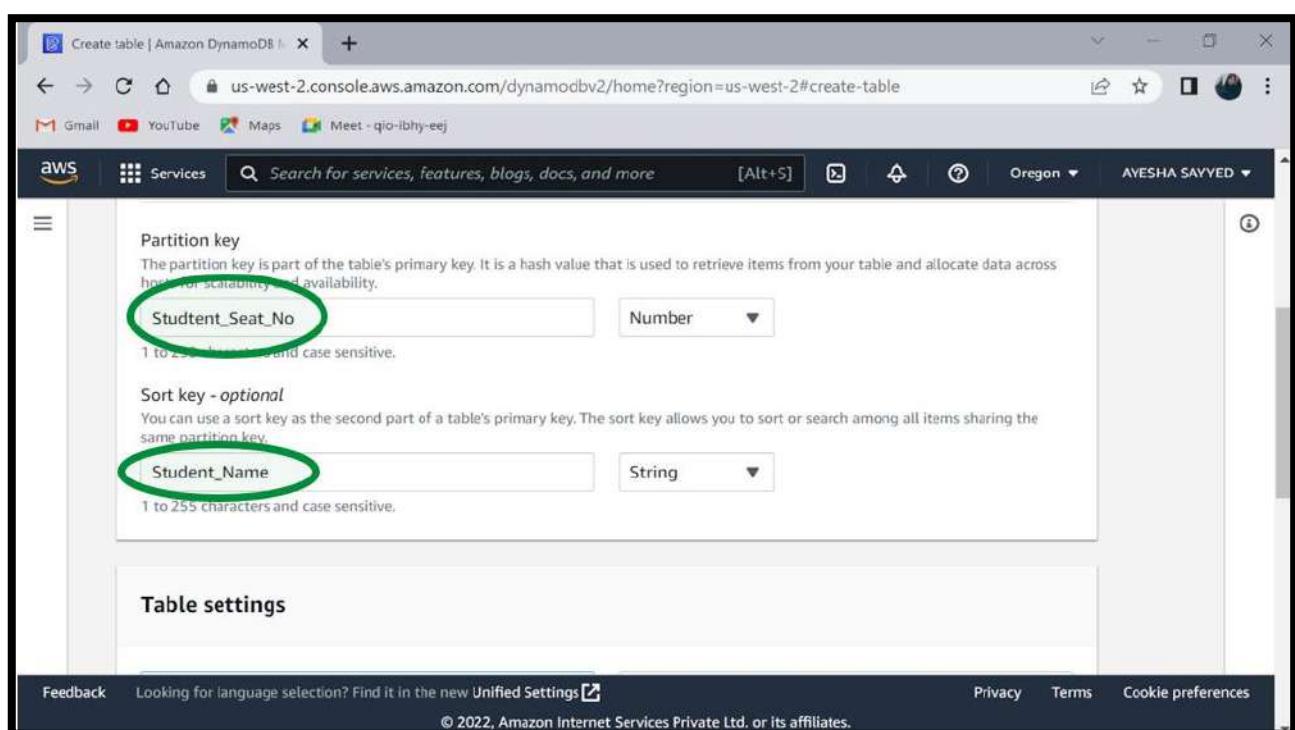
Step 3: Click on 'Create Table'.

The screenshot shows the AWS DynamoDB console. On the left, a sidebar lists options like 'Dashboard', 'Tables', 'Update settings', etc. The main content area has tabs for 'Software and internet' and 'Banking and finance'. Under the 'Get started' section, there is a call-to-action button labeled 'Create table' which is highlighted with a red box. Below this, there is a brief description of DynamoDB's benefits and a 'Pricing' section. The bottom of the screen includes standard navigation links.

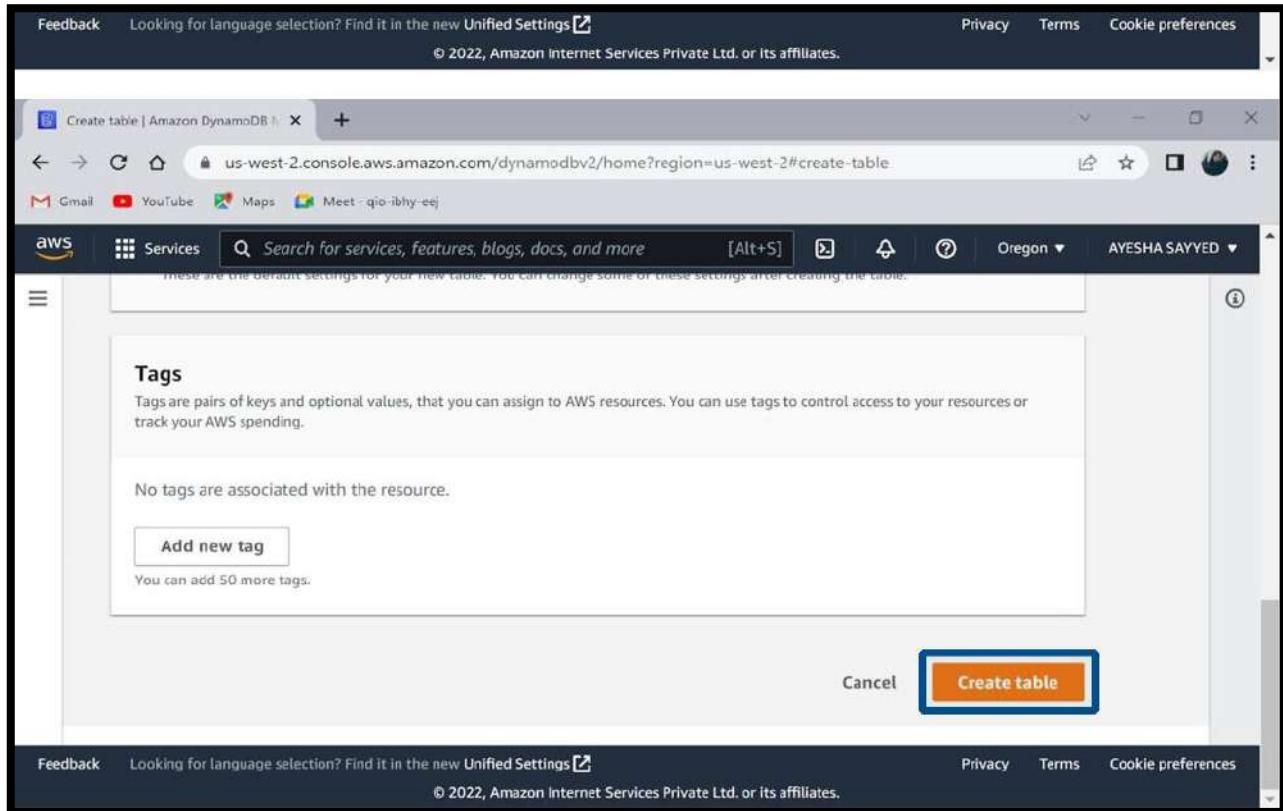
Step 4: Name your table → set ‘Partition key’ & ‘Sort key’ → select Default setting → Click on Create Table.



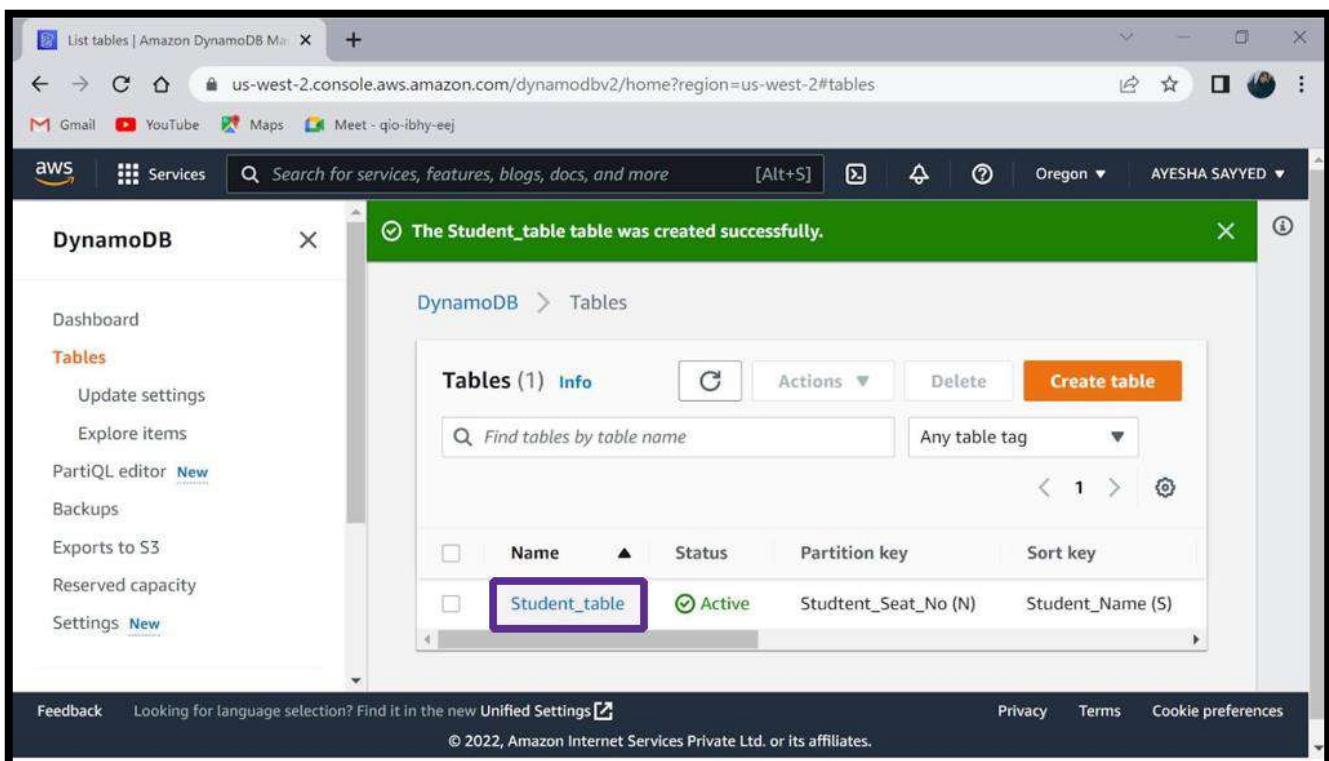
The screenshot shows the 'Create table' interface in the AWS Management Console. In the 'Table details' section, the 'Table name' field is highlighted with a green oval. It contains the value 'Student_table'. Below the table name, there is a note about character restrictions: 'Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)'.



The screenshot shows the 'Create table' interface in the AWS Management Console. In the 'Partition key' section, the primary key field contains 'Student_Seat_No', which is circled in green. In the 'Sort key - optional' section, the secondary key field contains 'Student_Name', also circled in green. Both fields have dropdown menus next to them indicating their data type: 'Number' for the partition key and 'String' for the sort key.



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, there's a sidebar with options like 'Dashboard', 'Tables', 'Update settings', etc. The main area shows a list of tables with 'Student_table' selected. At the top right of the main area, there's a 'Actions' dropdown and a prominent orange 'Explore table items' button, which is circled in purple. Below it, tabs for 'Overview', 'Indexes', and 'Metrics' are visible. A 'General information' section is also present.

The screenshot shows the 'Edit item' screen for the 'Student_table'. It displays three attributes: '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'. The 'Create item' button at the bottom right is highlighted with a red rectangle. The URL in the browser bar is https://us-west-2.console.aws.amazon.com/dynamodbv2/home?region=us-west-2#edit-item?table=Student_table®ion=us-west-2.

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 indicates 'Completed' with a green checkmark and 'Read capacity units consumed: 0.5'. Below this, a section titled 'Items returned (1)' shows a single item with attributes: 'Student_Seat_No' (Value: 57890) and 'Student_Name' (Value: Jasmin). The left sidebar lists various options like Dashboard, Tables, and Explore items. The bottom of the page includes standard 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. The 'Attributes' table has three rows: 'Student_Seat_No - Partition key' (Value: 57891, Type: Number), 'Student_Name - Sort key' (Value: Jasmin, Type: String), and a new row 'Student_Age' (Value: 21, Type: Number) which is circled in green. A green box highlights the 'Add new attribute' button at the top right of the table. At the bottom right are 'Cancel' and 'Create item' buttons. The bottom of the page includes standard 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.

Attributes

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

Add new attribute ▾

Create item

Items returned (11)

	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

Create item

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

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

[1]

Scan/Query items	
Scan	Query
Student_table	
Studtent_Seat_No (Partition key)	
<input type="text" value="57893"/>	
Student_Name (Sort key)	
Equal to	Enter sort key value
<input type="checkbox"/> Sort descending	

The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar menu includes options like Dashboard, Tables, Update settings, Explore items (which is highlighted in orange), PartiQL editor, Backups, Exports to S3, Reserved capacity, and Settings. The main area displays a table titled "Items returned (1)" with three columns: Student_Seat_No, Student_Name, and Student_Age. A single row is shown: Student_Seat_No 57893, Student_Name Sahil, and Student_Age 21. The value "57893" is circled in yellow. At the top right of the main area, there are "Run" and "Reset" buttons. Below the table, there's a note about completed status and consumed read capacity units. The bottom of the screen includes standard AWS footer links for Feedback, Privacy, Terms, and Cookie preferences.

[2]

The screenshot shows the AWS DynamoDB console interface for the "Student_table". The sidebar menu is identical to the first screenshot. The main area is titled "Student_table" and contains a "Scan/Query items" section. It has fields for "Scan/query a table or index" (set to "Query" for "Student_table"), "Studtient_Seat_No (Partition key)" (with the value "57895" circled in orange), and "Student_Name (Sort key)" (with a dropdown set to "Equal to" and an input field for "Enter sort key value"). There are also checkboxes for "Sort descending" and "Autopreview". The bottom of the screen includes the same footer links as the first screenshot.

The screenshot shows the Amazon DynamoDB console. On the left, the sidebar has options like Dashboard, Tables, Update settings, Explore items (which is selected), PartiQL editor, Backups, Exports to S3, Reserved capacity, and Settings. The main area shows a table named 'Student_table'. A message at the top says 'Completed Read capacity units consumed: 0.5'. Below it, there's a table header with columns Student_Seat_No, Student_Name, and Student_Age. One row is listed: Student_Seat_No 57895, Student_Name Susan, and Student_Age 21. The value '57895' is highlighted with a red oval. At the bottom of the table view, there are buttons for Actions and Create item.

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

The screenshot shows the 'Delete table' dialog box from the Amazon DynamoDB console. The sidebar on the left is the same as the previous screenshot. The dialog box contains the following text:
You are about to delete a table.
• Student_table
 Delete all CloudWatch alarms for this table.
 Create a backup of this table before deleting it.
If you do not select this check box, you will not be able to restore data being deleted.
To confirm the deletion of this table, type *delete* in the box.
A text input field contains the word 'delete'. At the bottom right of the dialog box is a large orange button labeled 'Delete table', which is highlighted with a red box.

List tables | Amazon DynamoDB Main | 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 X The request to delete the "Student_table" table has been submitted successfully.

Dashboard

Tables

- Update settings
- Explore items
- PartiQL editor [New](#)
- Backups
- Exports to S3
- Reserved capacity
- Settings [New](#)

DynamoDB > Tables

Tables (1) Info Actions Delete Create table

Find tables by table name Any table tag

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

The 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 shows one table named "Student_table". The "Status" column for this table is highlighted with a red box and displays the text "⚠ Deleting". The rest of the table data is visible, including columns for Name, Status, Partition key, Sort key, Indexes, and Region. The left sidebar contains navigation links for Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Reserved capacity, and Settings. The bottom of the page includes standard footer 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 weather details (28°C, Rain showers).

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

The screenshot shows the AWS Management Console search results for 'IAM'. A search bar at the top contains the text 'IAM'. The results are displayed in a grid format under the heading 'Search results for "IAM"'. The first result is 'IAM' (Manage access to AWS resources), which is highlighted with a red box. Other results include 'IAM Identity Center (successor to AWS Single Sign-On)', 'Resource Access Manager', and 'Amazon VPC IP Address Manager'. Below the search results, there are sections for 'Features' (Groups, Roles) and 'Access Analyzer'. The sidebar on the left shows other services like AWS Budgets, Lambda, and CloudWatch.

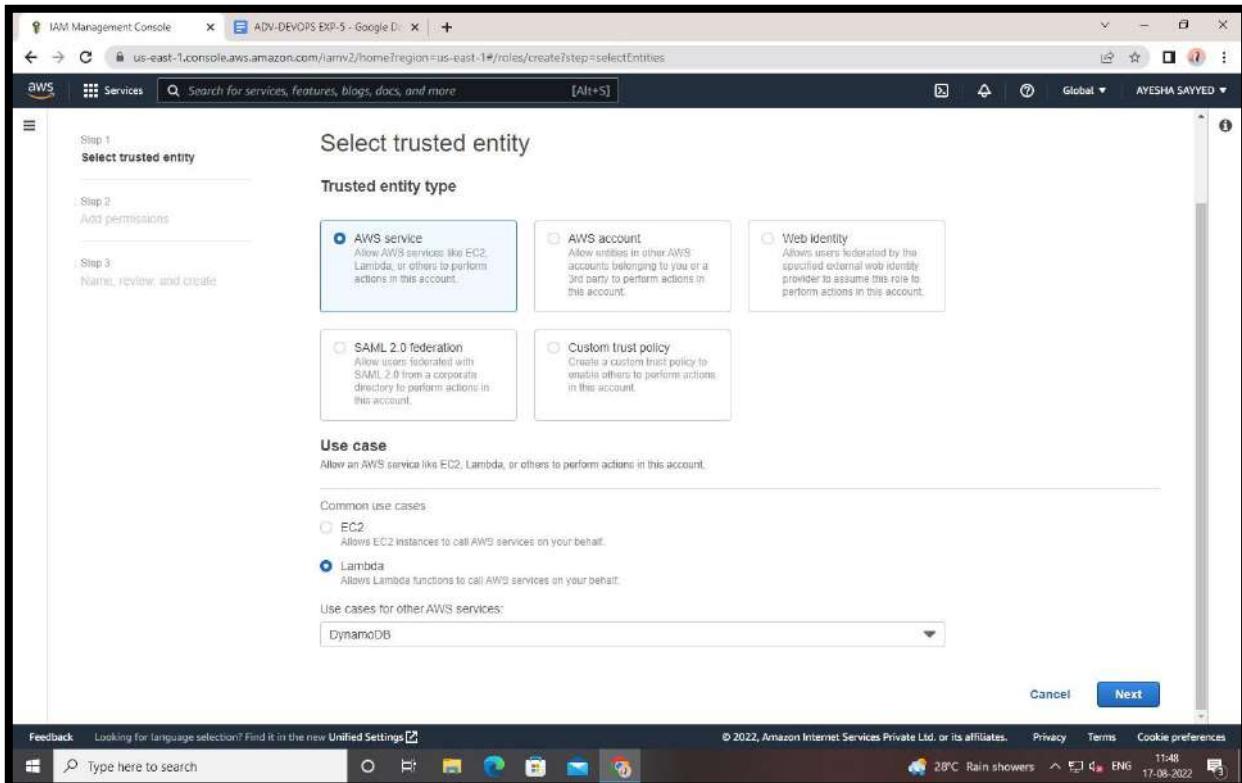
“IAM dashboard home page”.

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 "IAM resources" summary with 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 side, there are sections for "AWS Account" (Account ID: 518313021583, Account Alias: 518313021583, Sign-in URL: https://518313021583.sigin.aws.amazon.com/console) and "Quick Links" (My security credentials, Manage your access keys, multi-factor authentication (MFA) and other credentials). The bottom right corner shows the date and time (17-08-2022, 11:46) and weather information (28°C Rain showers).

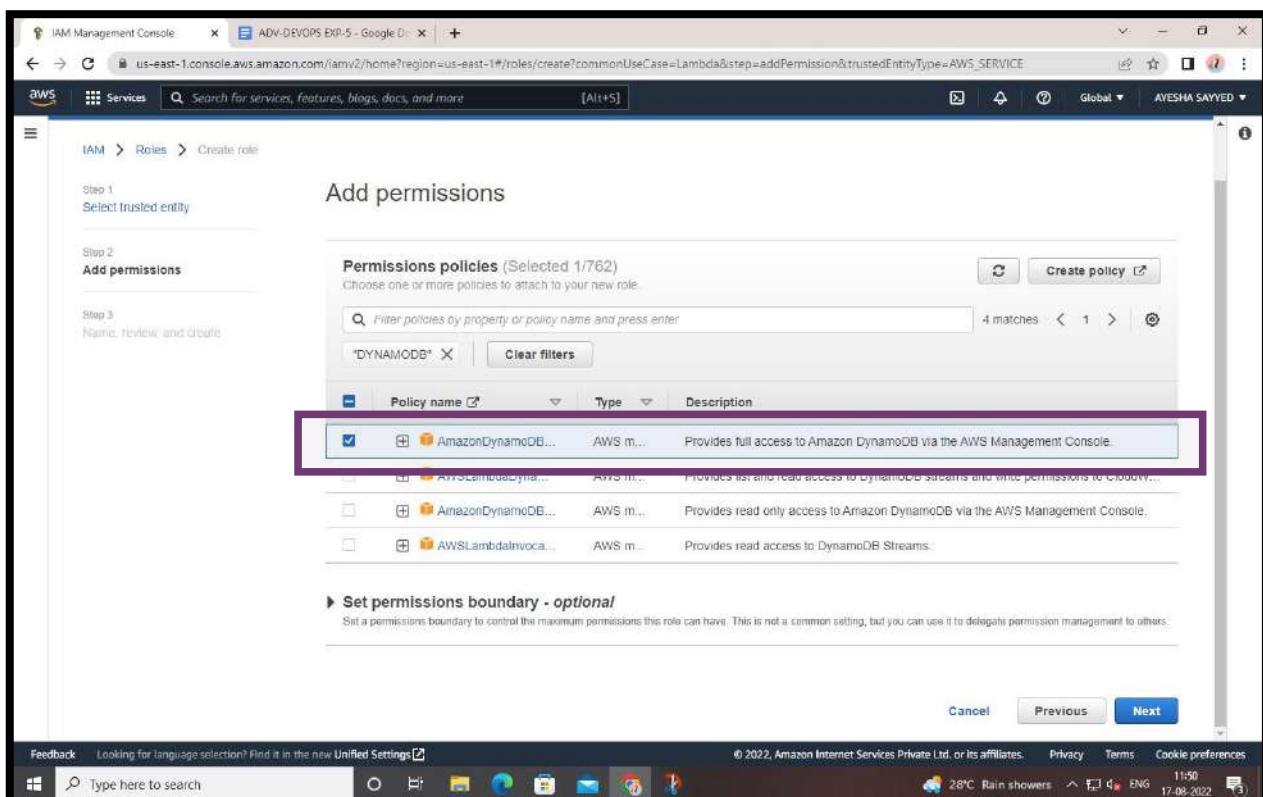
Step 3: Go to role and click on “Create role”.

The screenshot shows the "Roles" page under the IAM section of the AWS Management Console. The left sidebar is identical to the home page. The main content displays a table of existing roles, each with a "Delete" button and a "Create role" button highlighted with a green box. The table columns include Role name, Trusted entities, and Last ac... (Last activity). Below the table is a "Roles Anywhere" section with a "Manage" button and icons for X.509 Standard and Temporary credentials. The bottom right corner shows the date and time (17-08-2022, 11:46) and weather information (28°C Rain showers).

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



Step 5: Search Amazon DynamoDB in permission policies → Select policy which provides “full access to DynamoDB” → Click on 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 tabs for 'AWS Services' and 'Search for services, features, blogs, docs, and more'. The main content area is titled 'Name, review, and create'.

Role details:

- Role name:** sayeed_ayesha_045 (highlighted with a green oval)
- Description:** Allows Lambda functions 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 -                 "lambda:InvokeFunction"  
8 -             ]  
9 -         }  
10 -    ]  
11 - ]  
12 - }  
13 - }  
14 - }  
15 - ]  
16 - ]
```

Step 2: Add permissions

Permissions policy summary:

Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy

Tags:

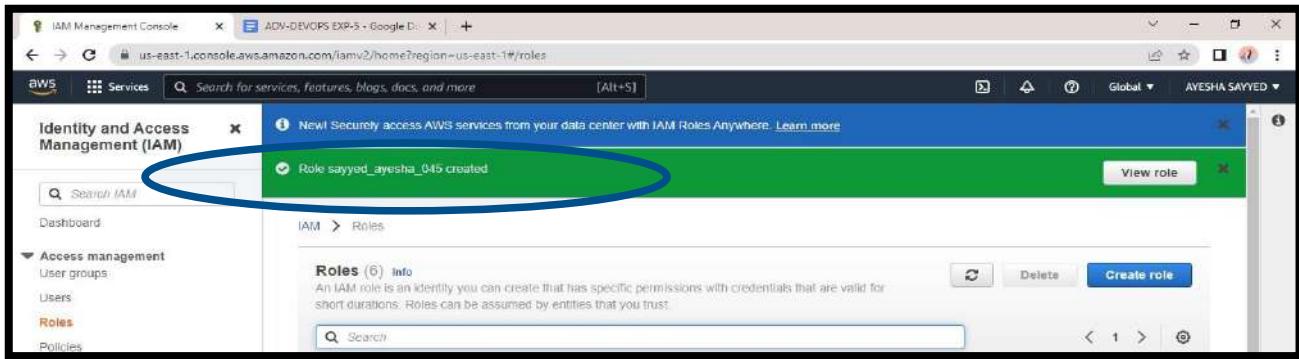
Add tags (Optional): Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

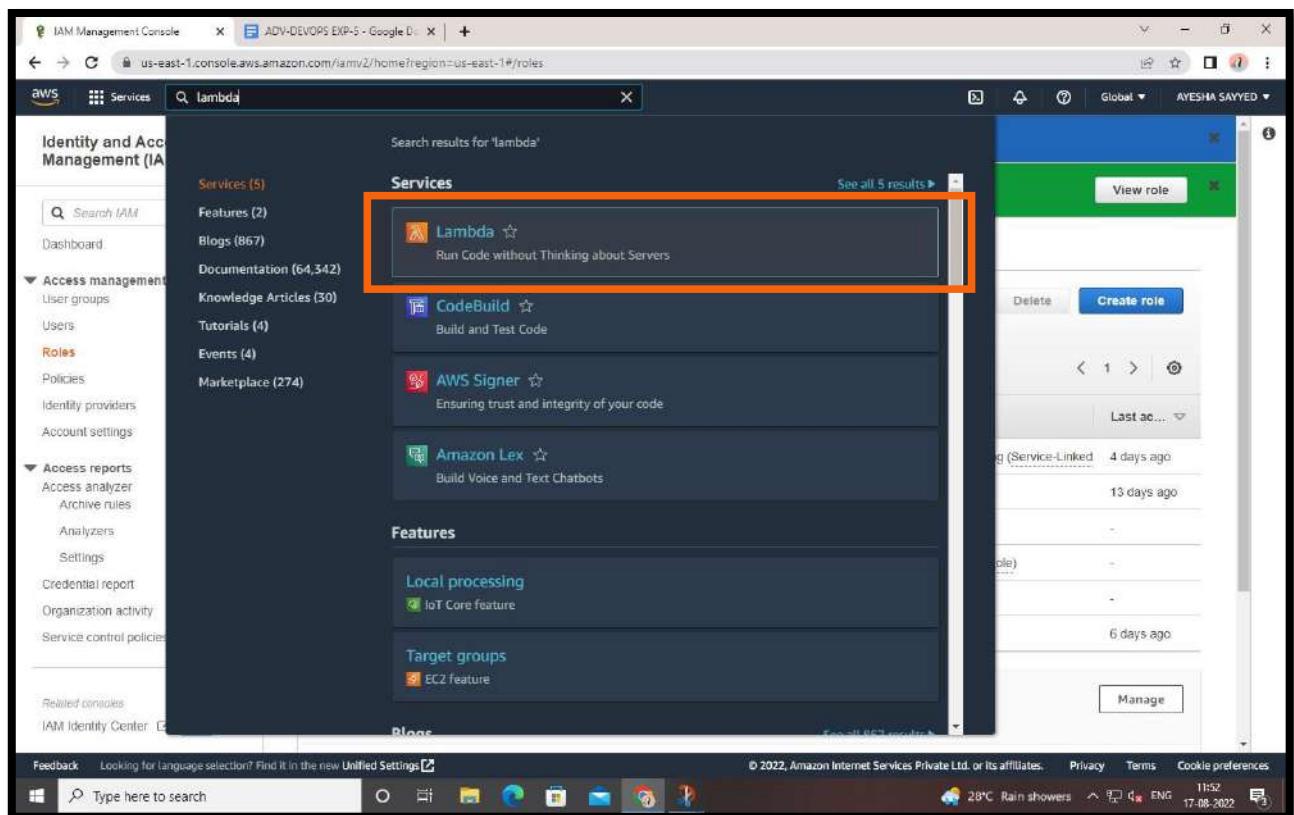
Add tag: You can add up to 50 more tags.

Buttons at the bottom right: Cancel, Previous, **Creates role** (highlighted with a green rectangle).

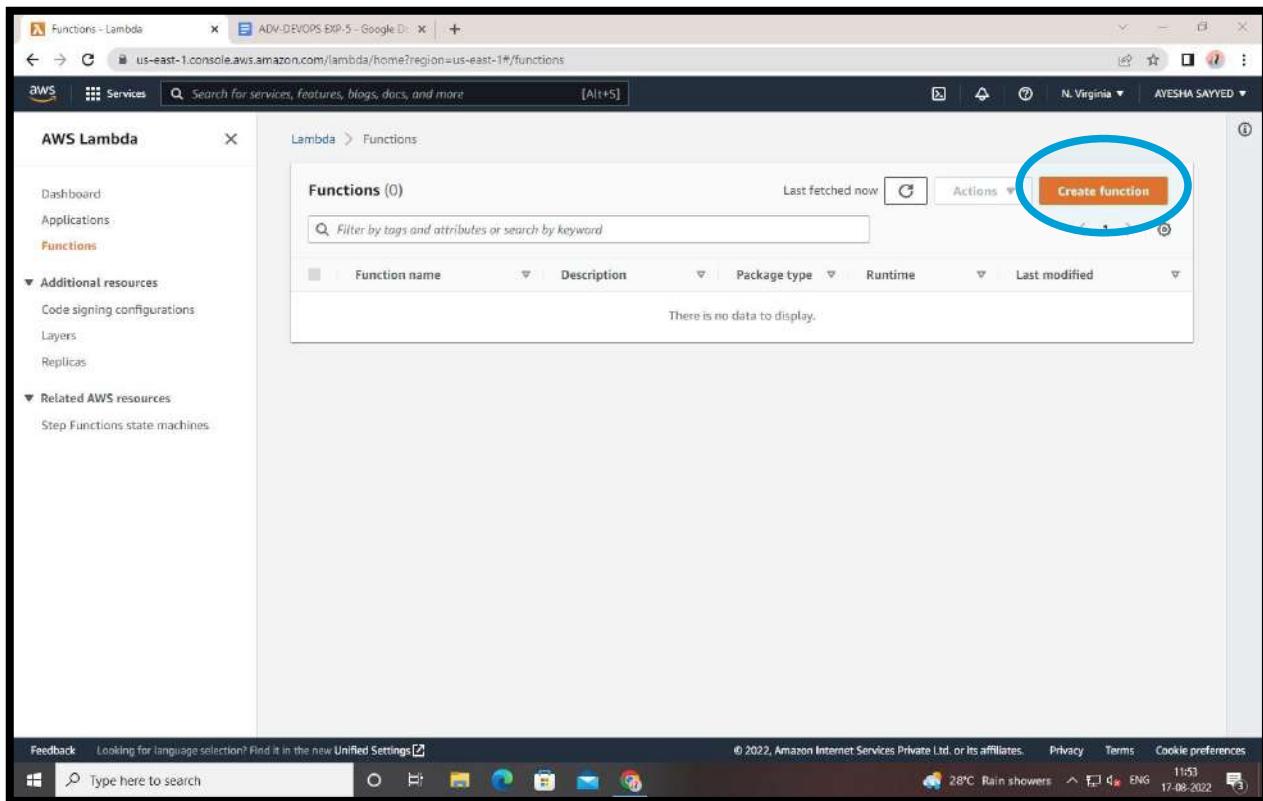
Role created successfully!



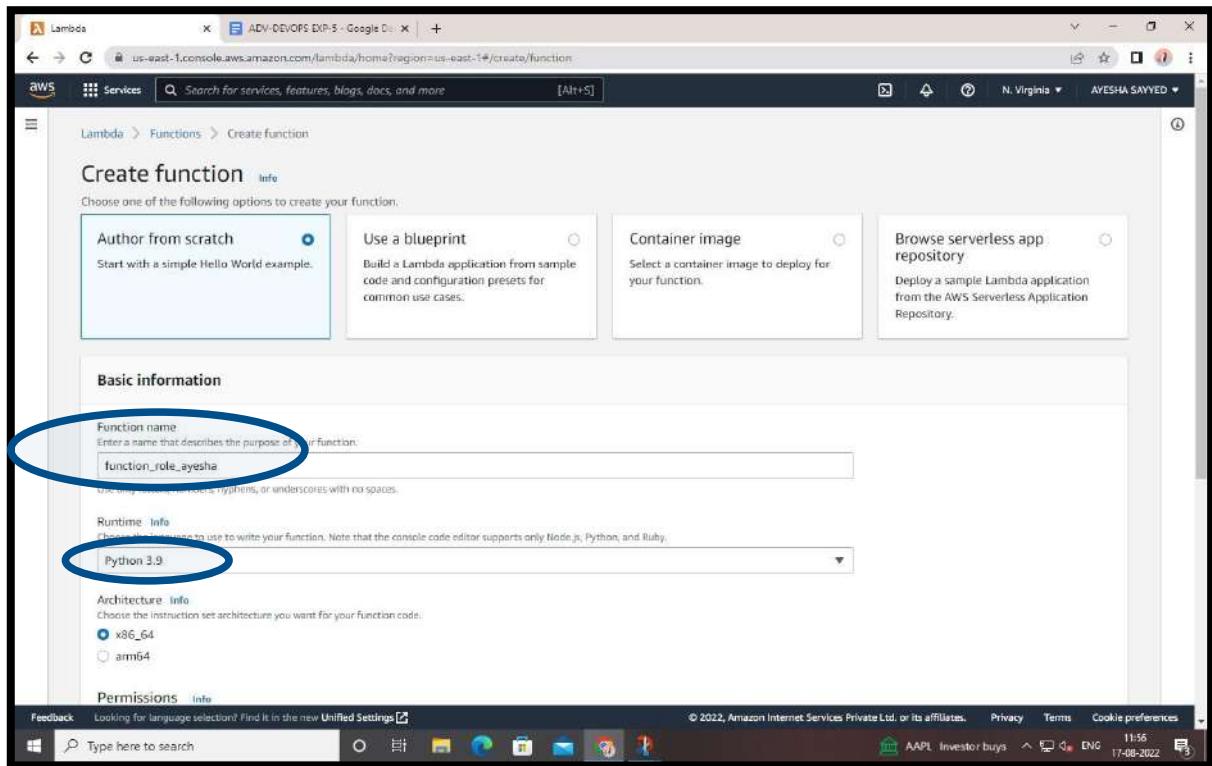
Step 7: Go to services → click on “Lambda”.



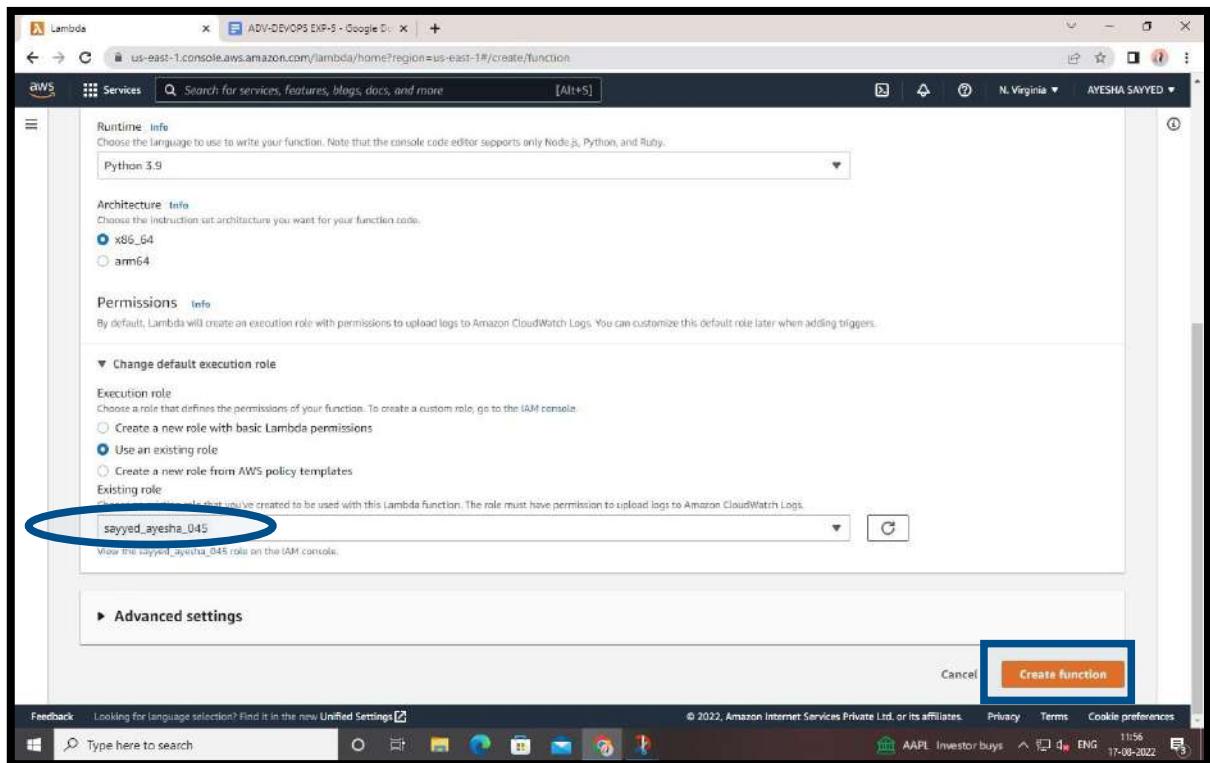
Step 8: Click on Create function.



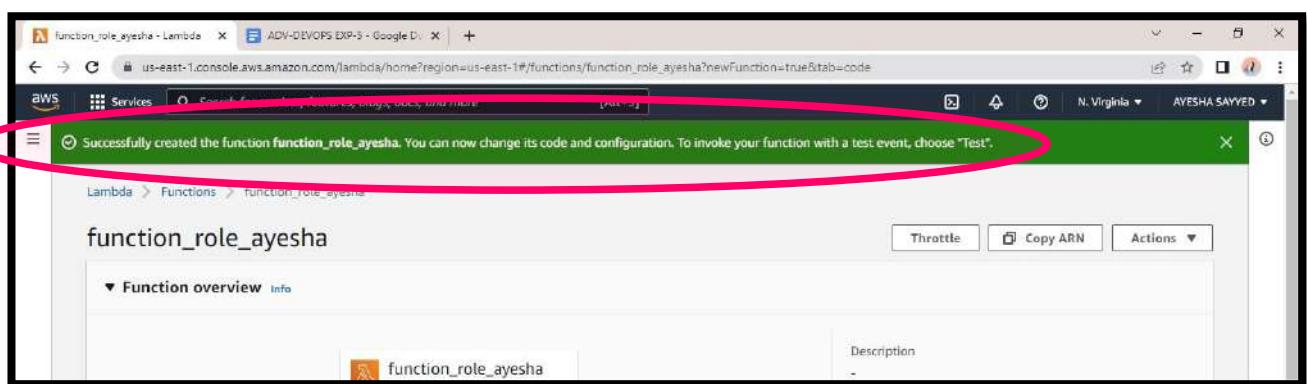
Step 9: Choose “Author from Scratch” → Give name to your function→ Choose “Python” language in runtime.



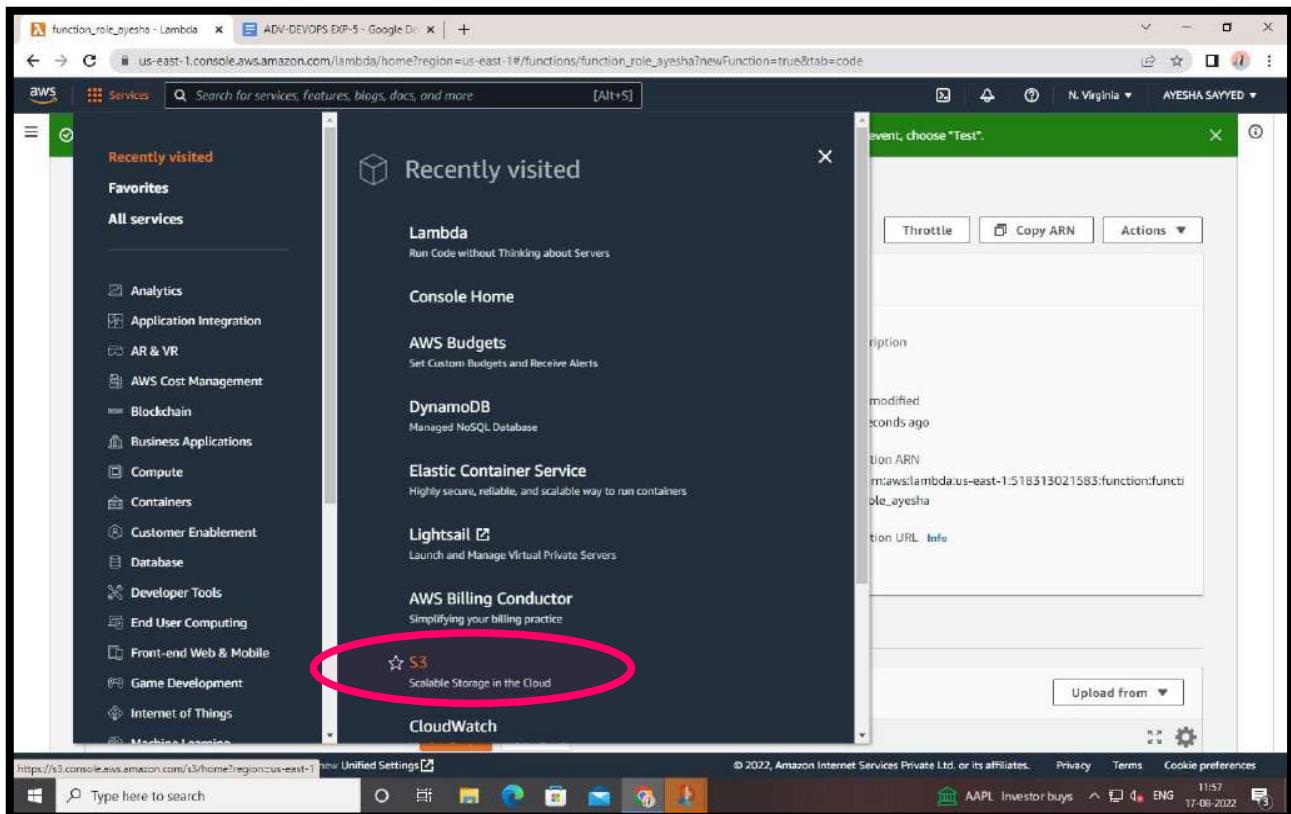
Step 10: Permissions → Change execution role → Choose “Use an existing role” → Click on Create function.



Function created successfully!



Step 11: Go to Services → Select S3.

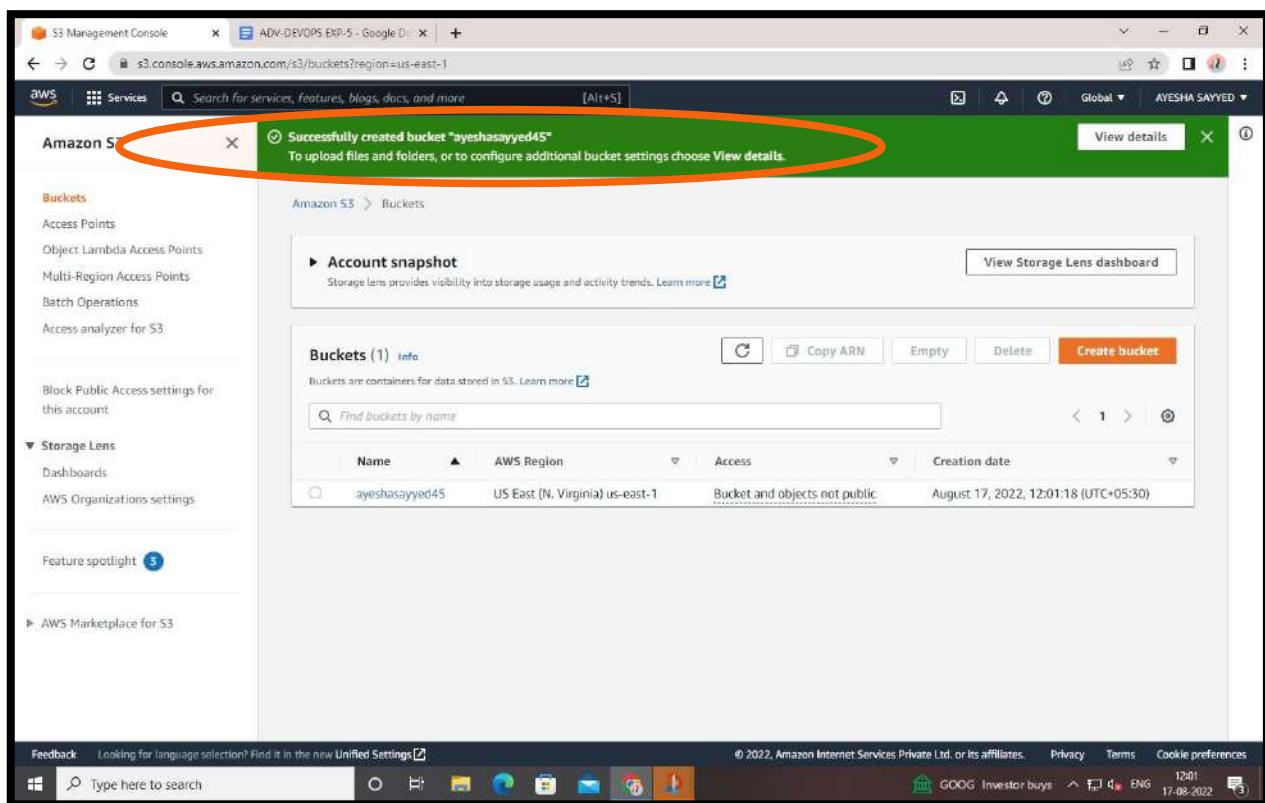


Step 12: Click on “Create bucket”.

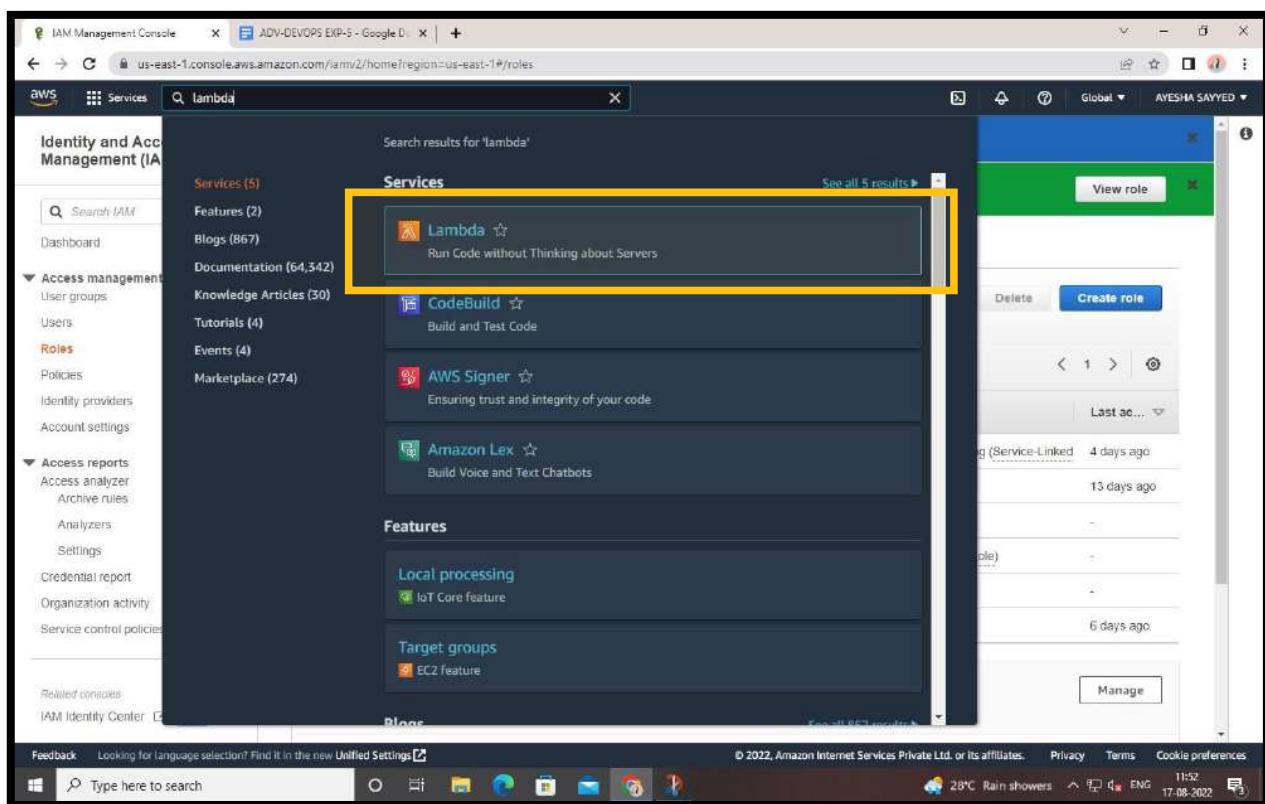


Step 13: Give name to your bucket.

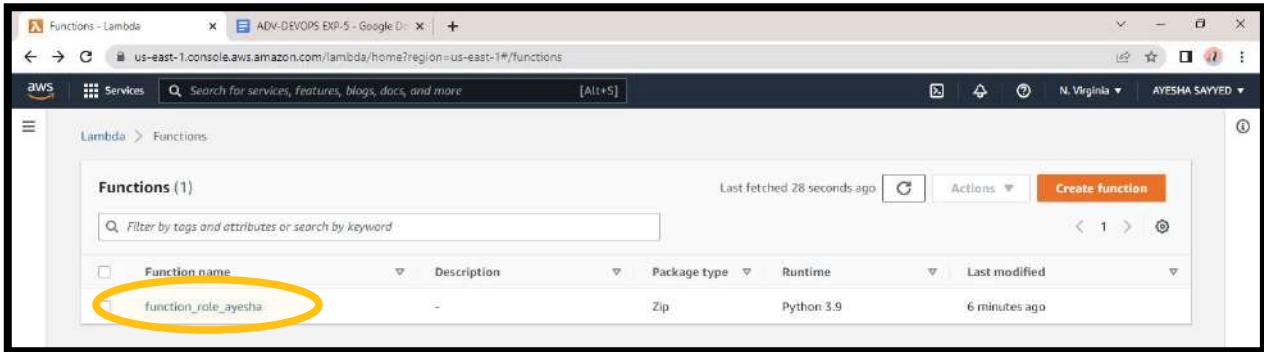
The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' step is active, with the 'Bucket name' field set to 'ayeshasayyed43'. A green oval highlights this field. The 'AWS Region' is set to 'US East (N. Virginia) us-east-1'. The 'Object Ownership' section shows '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 'Server-side encryption' set to 'Disable'. 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.



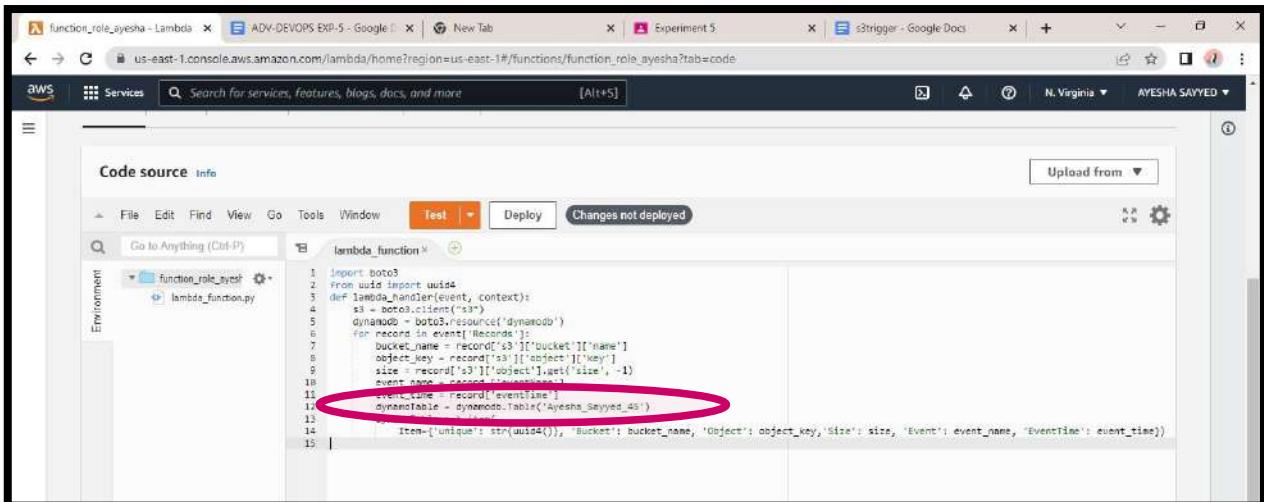
Step 14: Search for Lambda and select it.



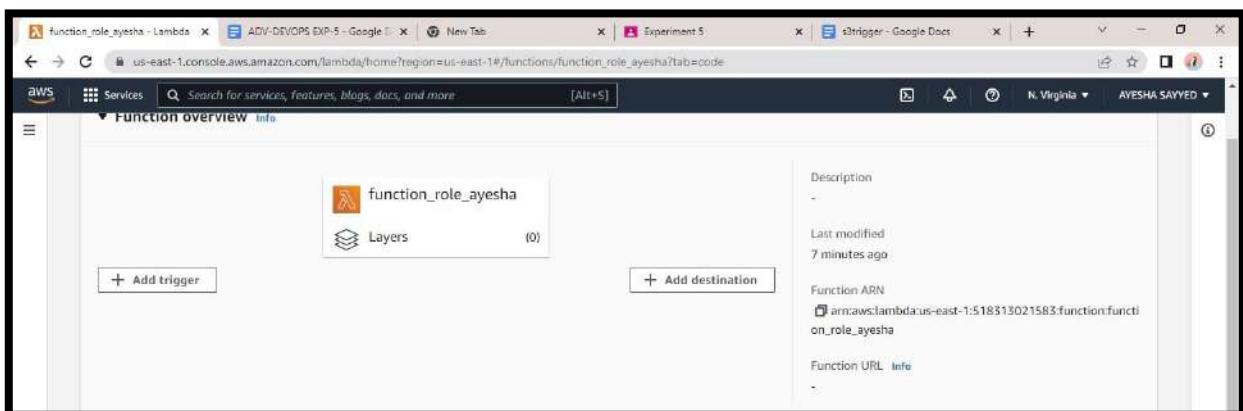
Click on ‘Lambda Function Name’.



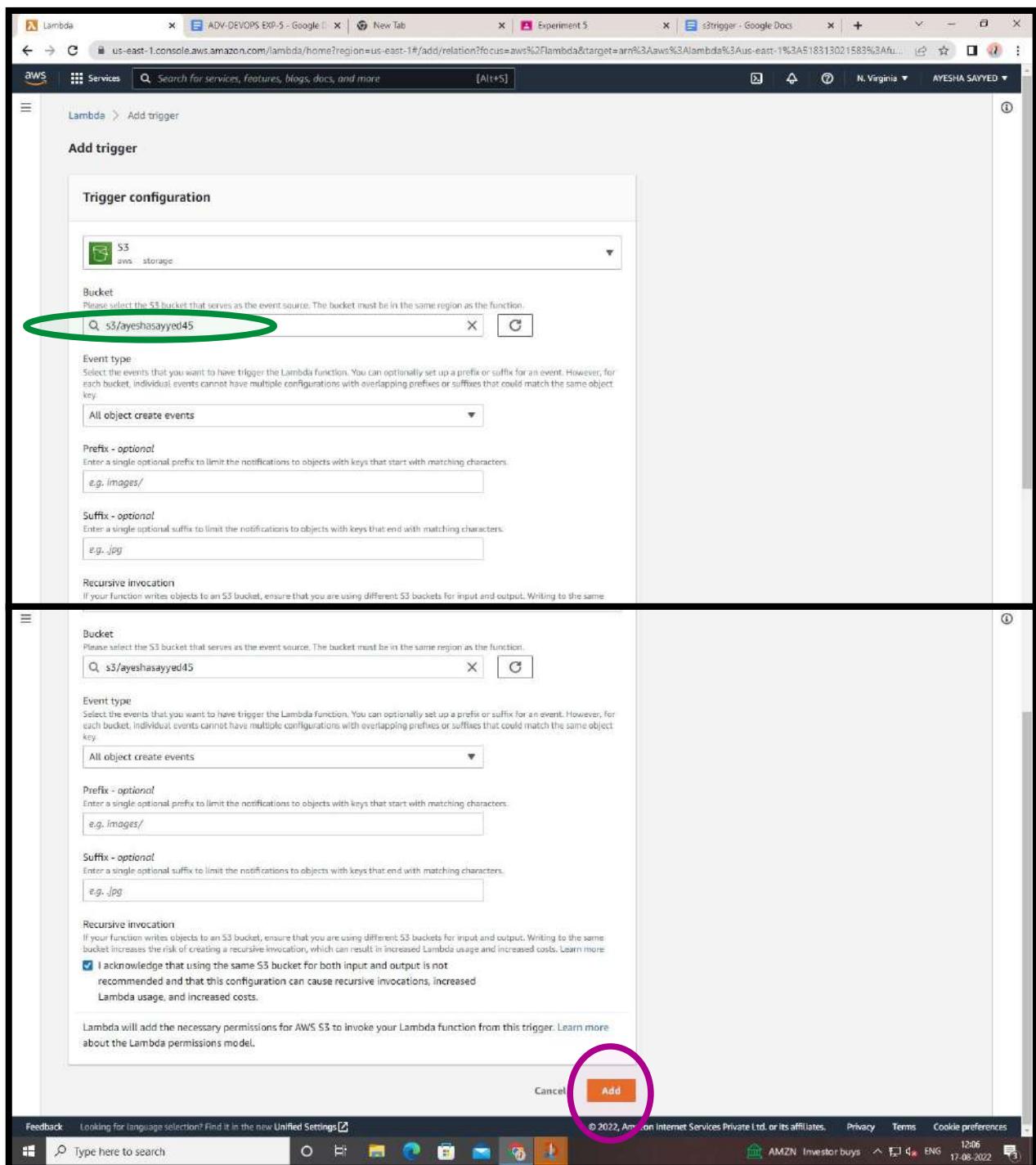
Upload ‘Code’ to the Lambda Function and Deploy it.



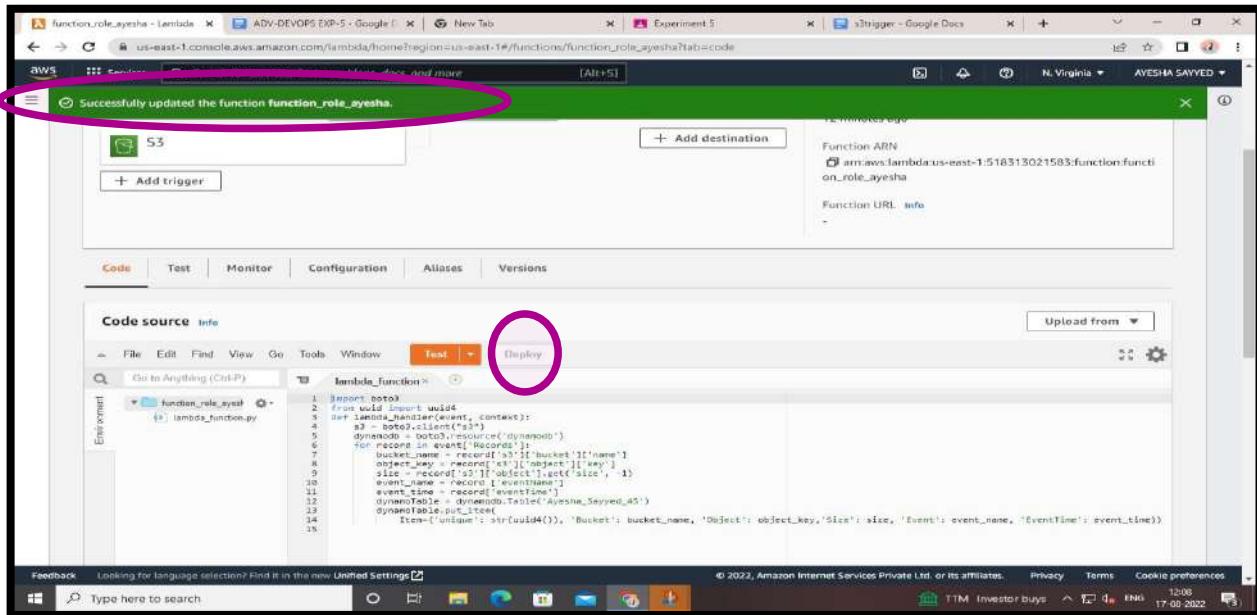
Then click on “Add Trigger” and add bucket from S3 into the trigger, so that it can generate trigger whenever something is added to the bucket.



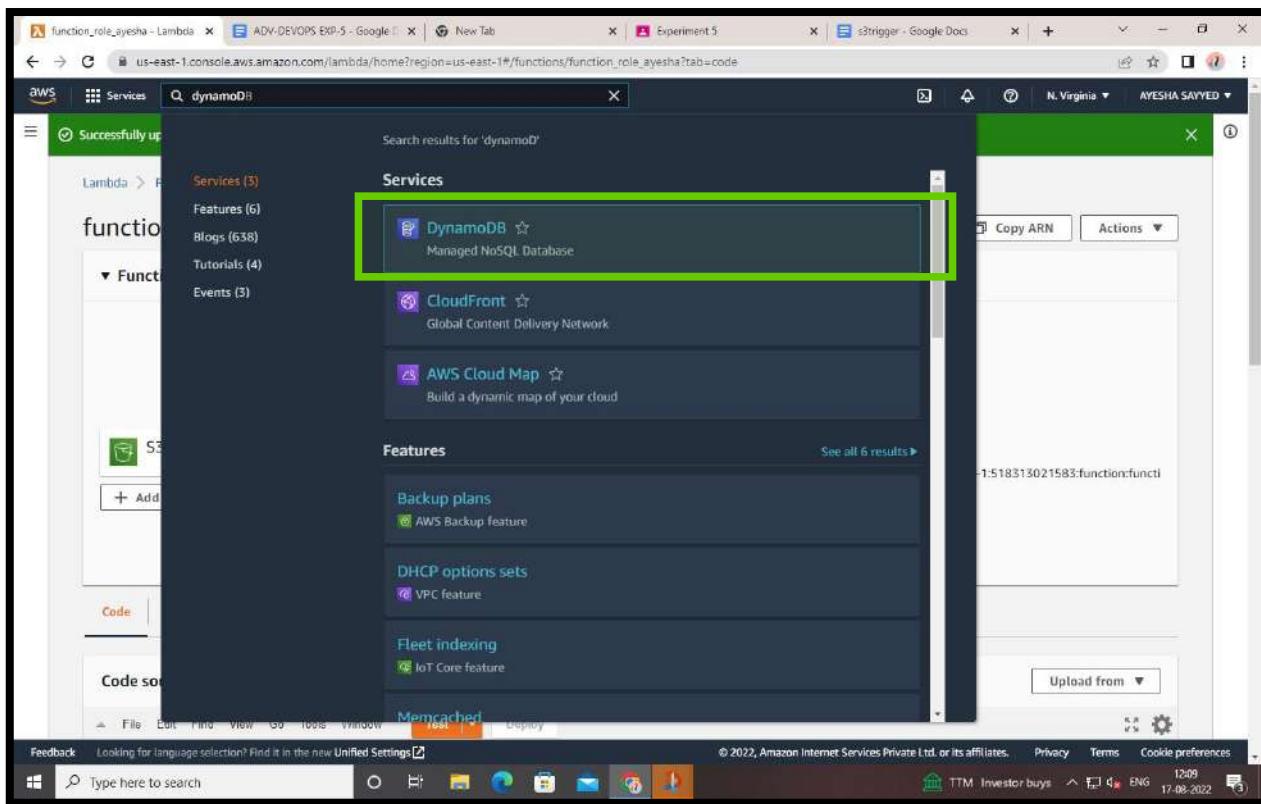
Select ‘S3’ add your ‘bucket’ click to the checkbox of “I Acknowledge” & then click to “Add”.



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	Name	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. At the bottom right, there are 'Cancel' and 'Create table' buttons.

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 the message, there is a table list titled "Tables (1) Info". The table information is as follows:

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
Ayesha_Sayyed_45	Active	unique (\$)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)

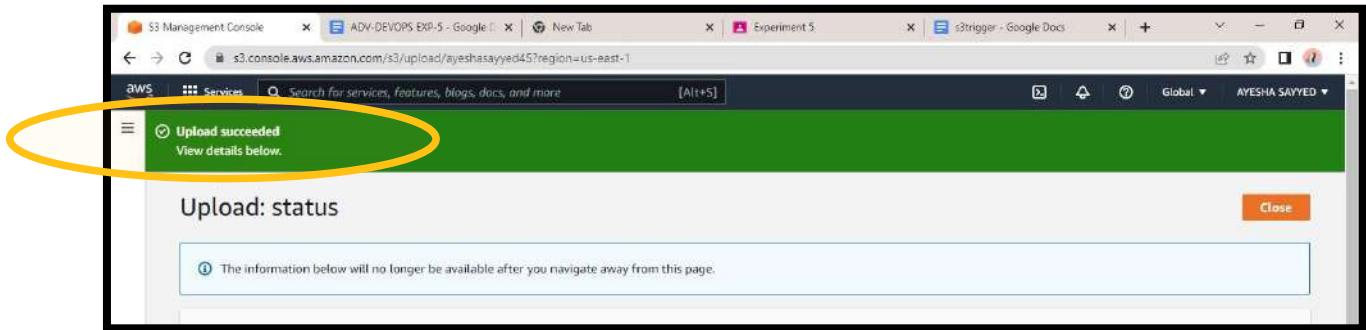
Step 18: Go to S3 → Bucket and upload an object by clicking on add files.

The screenshot shows the AWS Lambda service in the AWS Management Console. The left sidebar has a "Recently visited" section with links to Lambda, Console Home, AWS Budgets, DynamoDB, Elastic Container Service, Lightsail, AWS Billing Conductor, and CloudWatch. The "S3" link is highlighted with a yellow box. The main content area shows a Lambda function named "function_role_ayesha" with options for Throttle, Copy ARN, and Actions. At the bottom right, there is an "Upload from" button. The URL in the address bar is https://lambda.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/function_role_ayesha?newFunction=true&tab=code.

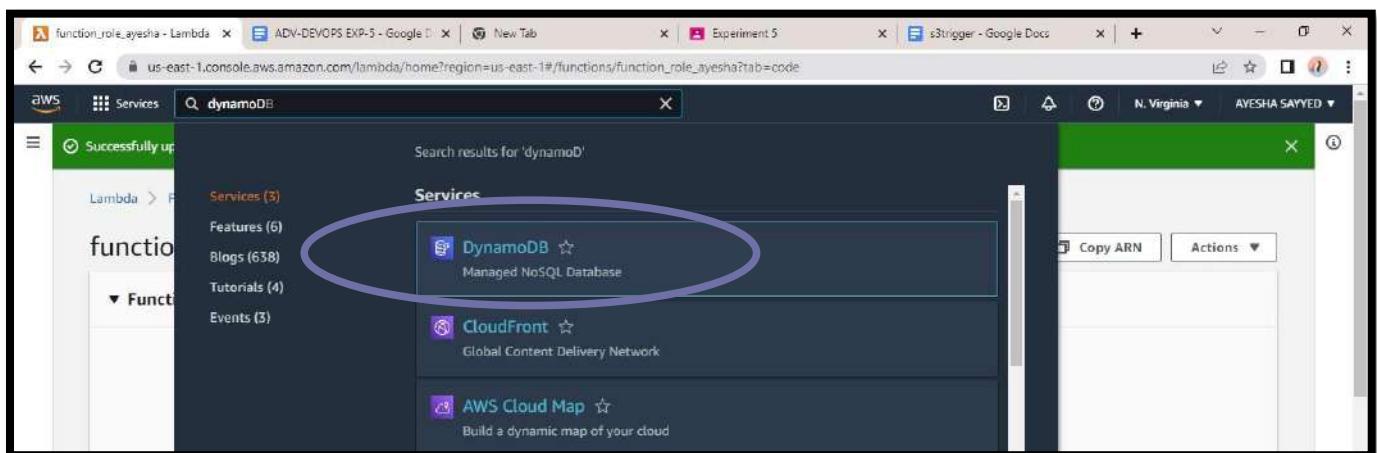
The screenshot shows the AWS S3 Management Console interface. At the top, there are several tabs: 'ayeshasayyed45 - S3 bucket', 'ADV-DEVOPS EXP-5 - Google Doc', 'New Tab', 'Experiment 5', and 's3trigger - Google Docs'. The main navigation bar includes 'AWS Services', 'Search for services, features, blogs, docs, and more', and a global search bar with the placeholder '[Alt+5]'. Below this, the breadcrumb path shows 'Amazon S3 > Buckets > ayeshasayyed45'. The main content area is titled 'ayeshasayyed45' with an 'Info' link. A horizontal menu bar below the title includes 'Objects' (which is selected), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' section is titled 'Objects (0)' and contains a message: '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 are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and a large orange 'Upload' button, which is highlighted with a yellow box. A search bar labeled 'Find objects by prefix' is also present. A table below shows columns for Name, Type, Last modified, Size, and Storage class, with a note 'No objects' and 'You don't have any objects in this bucket.' A secondary 'Upload' button is located at the bottom of the table. The bottom of the screen shows a Windows taskbar with various pinned icons and system status information.

The screenshot shows the 'Upload' step in the AWS S3 Management Console. The browser tabs are identical to the previous screenshot. The main content area shows the 'Upload' step for the bucket 'ayeshasayyed45'. The 'Upload' link is followed by an 'Info' link. A message 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'. Below this is a dashed box with the instruction 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.' An 'Add files' button is highlighted with a yellow box. A table lists 'Files and folders (1 Total, 141.9 KB)' with one item: '437c216e57fb36869301be60eee410d.jpg' (image/jpeg, 141.9 KB). The 'Destination' section shows 's3://ayeshasayyed45'. Under 'Destination details', it says 'Bucket settings that impact new objects stored in the specified destination.' The bottom of the screen shows a Windows taskbar with various pinned icons and system status information.

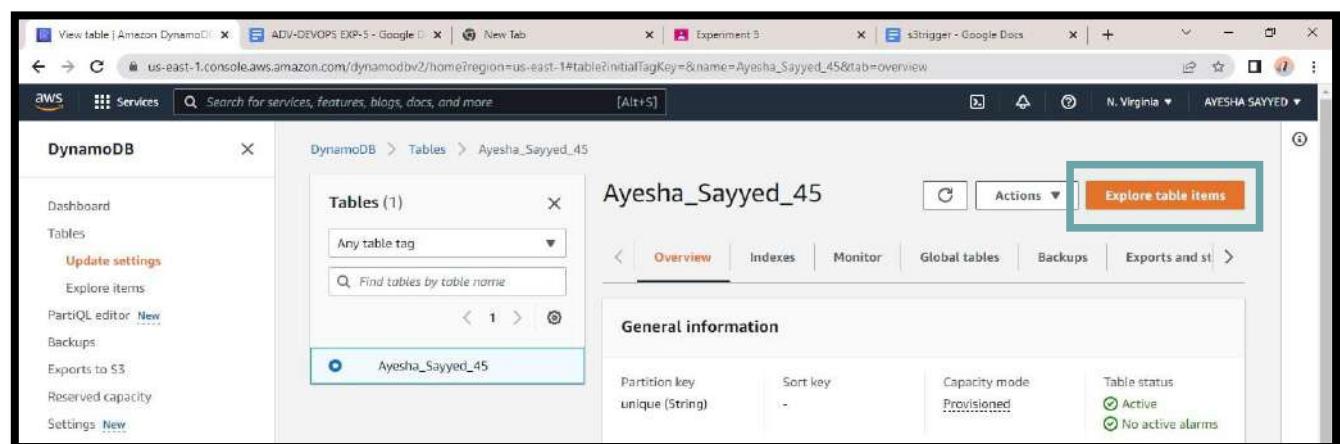
Successfully Uploaded.



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.



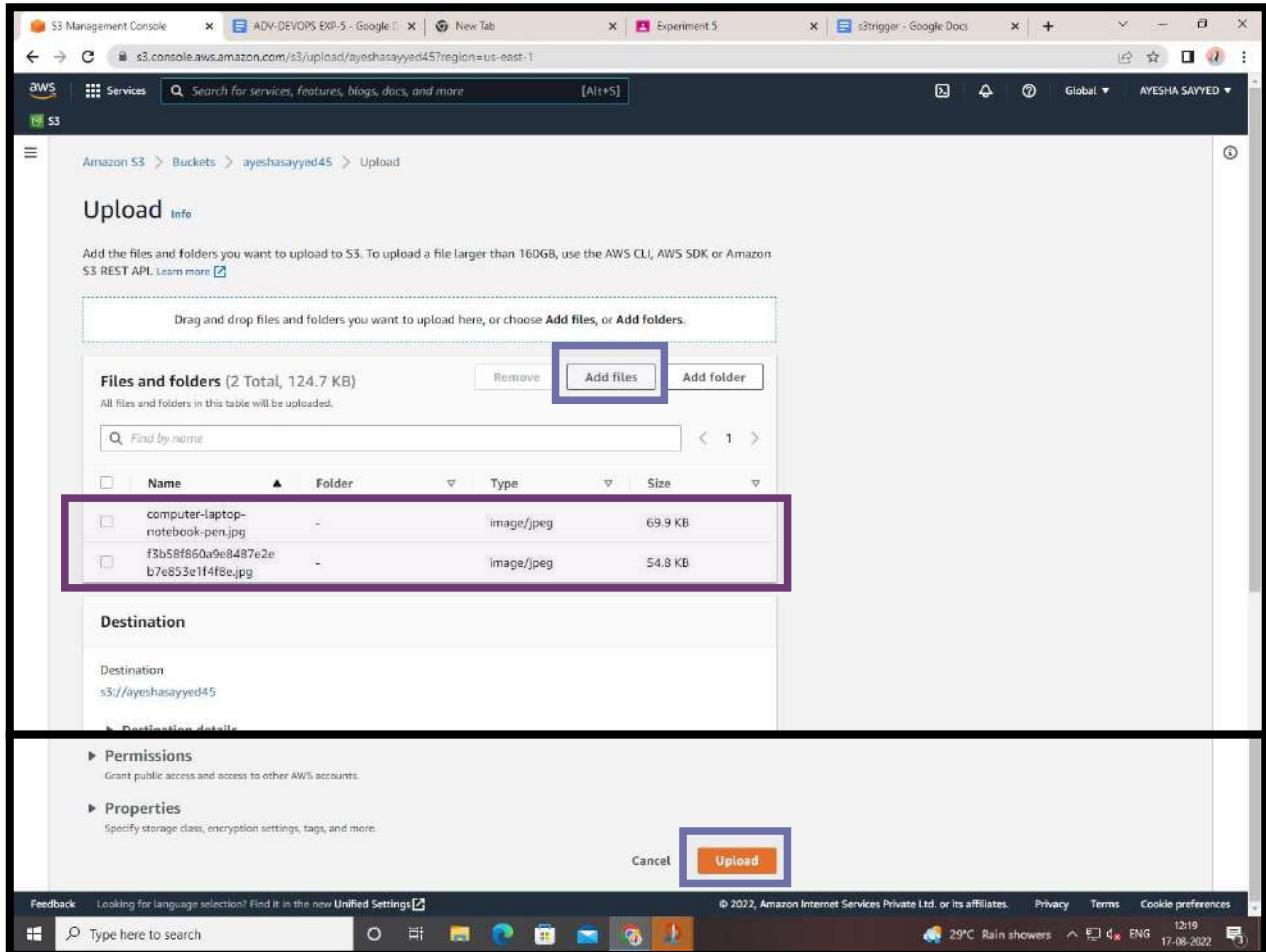
A screenshot of the AWS DynamoDB console. The left sidebar shows 'DynamoDB' with 'Tables' selected. Under 'Tables', 'Explore items' is highlighted. The main area shows a table named 'Ayesha_Sayyed_45'. A single item is listed in the 'Items returned' section:

unique	Bucket	Event	EventTime	Object	Size
95b32a9f-2b50-41f3-...	ayesha_sayy...	ObjectCreate...	2022-08-1...	437c216e5...	145295

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

A screenshot of the AWS search results page. The search term 's3' is entered in the search bar. The results show the 'Services' section with the 'S3' service card highlighted with a pink box. The 'S3' card has a star icon and the text 'Scalable Storage in the Cloud'. Other services listed include Features, Blogs, Documentation, Knowledge Articles, Tutorials, and Events.

Upload → add files → click on upload button.



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 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
5ef06ab-075f-4b1d...	ayeshasayy...	ObjectCreat...	2022-08-1...	f3b58f860a...	56068
95b52a9f-2b50-41f3...	ayeshasayy...	ObjectCreat...	2022-08-1...	437c216e5...	145295
62540a40-7ch5-40e3...	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 Identity and Access Management (IAM) services: Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), Access reports (Access analyzer, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)), and Related consoles (AWS Lambda, AWS CloudWatch Metrics). The 'Roles' section is selected. A modal dialog box titled 'Delete sayyed_ayesha_045?' is displayed. It contains a confirmation message: 'Delete sayyed_ayesha_045 permanently? This will also delete all its inline policies and any attached instance profiles.' Below this is a text input field with the value 'sayyed_ayesha_045' circled in red. At the bottom right of the modal is a blue 'Delete' button highlighted with a green box.

Role Deleted Successfully!!

The screenshot shows the AWS IAM Management Console. A red circle highlights a green success message at the top: "Role deleted sayyed_ayesha_045". The left sidebar shows "Identity and Access Management (IAM)" selected. The main pane displays "Roles (5) Info" with a "Delete" button and a "Create role" button.

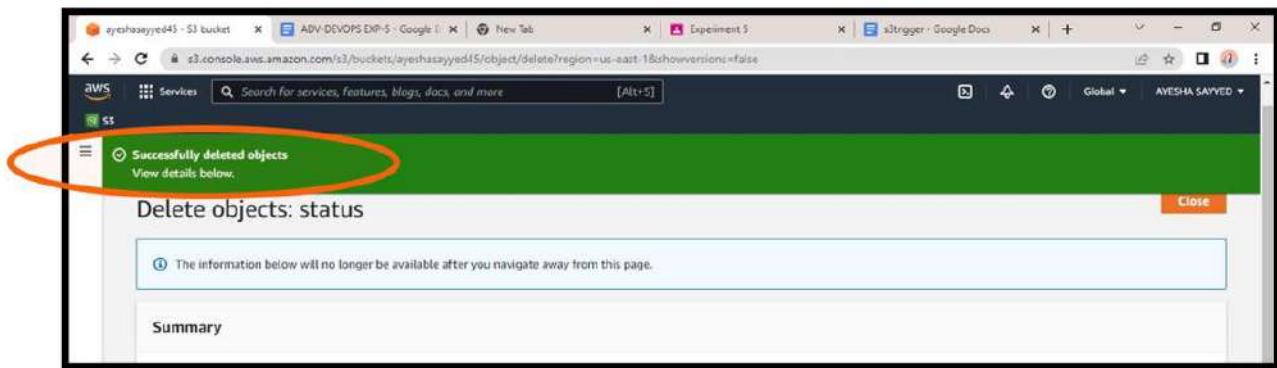
Step 25: Delete Lambda Function which was created → Also delete the table which was created in DynamoDB.

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

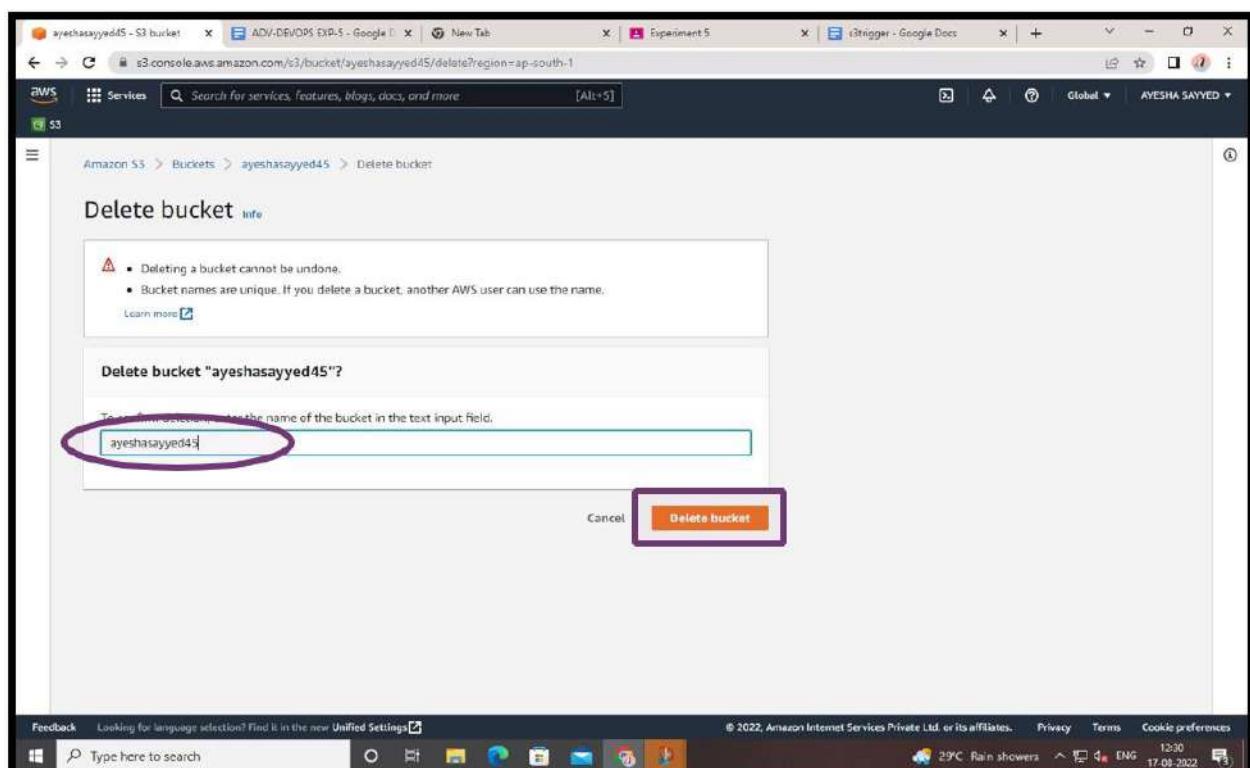
The screenshot shows the AWS S3 console. A red circle highlights the "permanently delete" button in the "Permanently delete objects?" dialog box. The dialog box also contains a text input field with the placeholder "To confirm deletion, type permanently delete in the text input field." The main pane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

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
f3b5bf860a9e8487e2eb7e853e1f4f8e.jpg	jpg	August 17, 2022, 12:20:02 (UTC+05:30)	54.8 KB

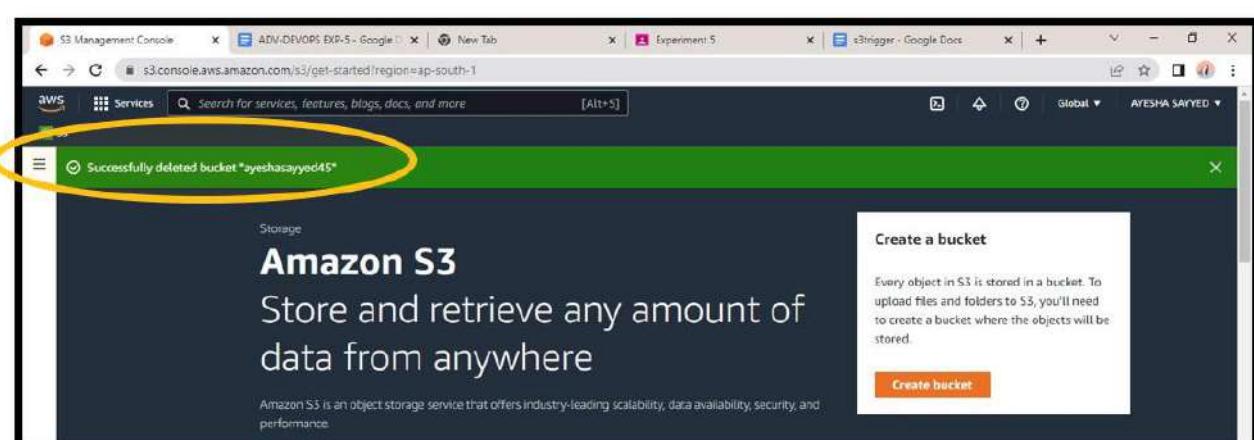
Object Deleted Successfully!!



Now delete the bucket.



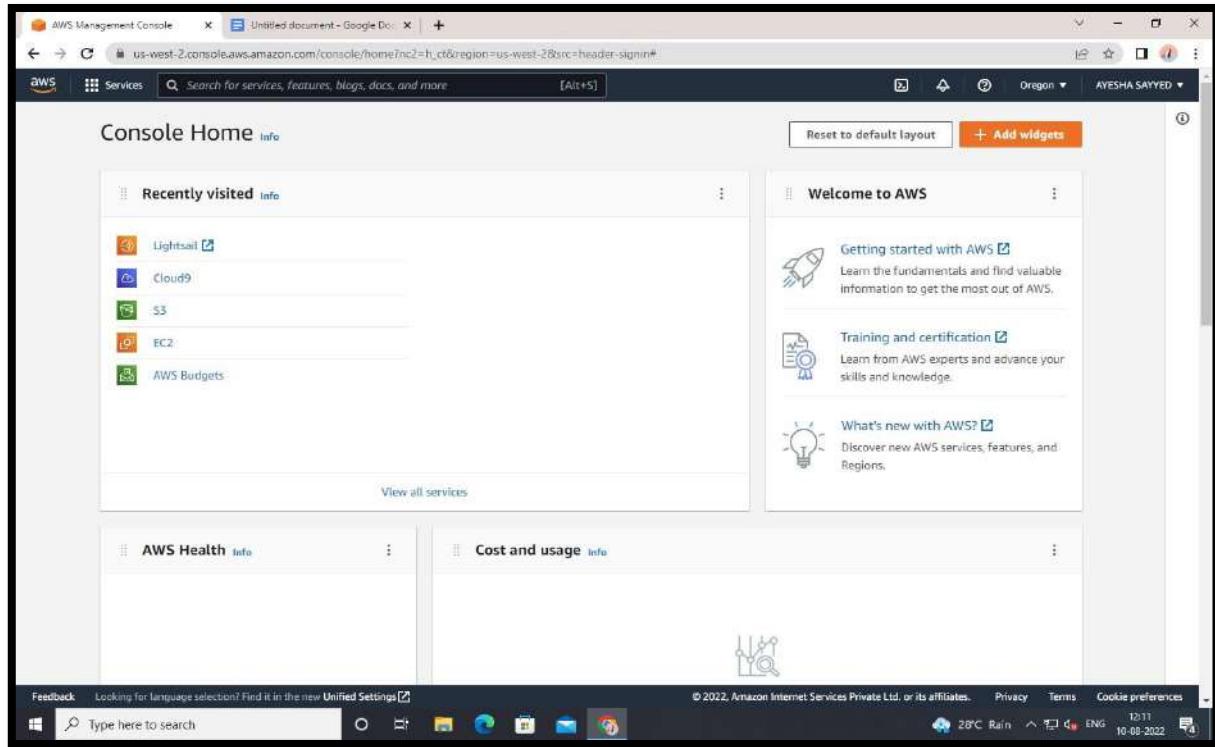
Bucket Deleted Successfully!!



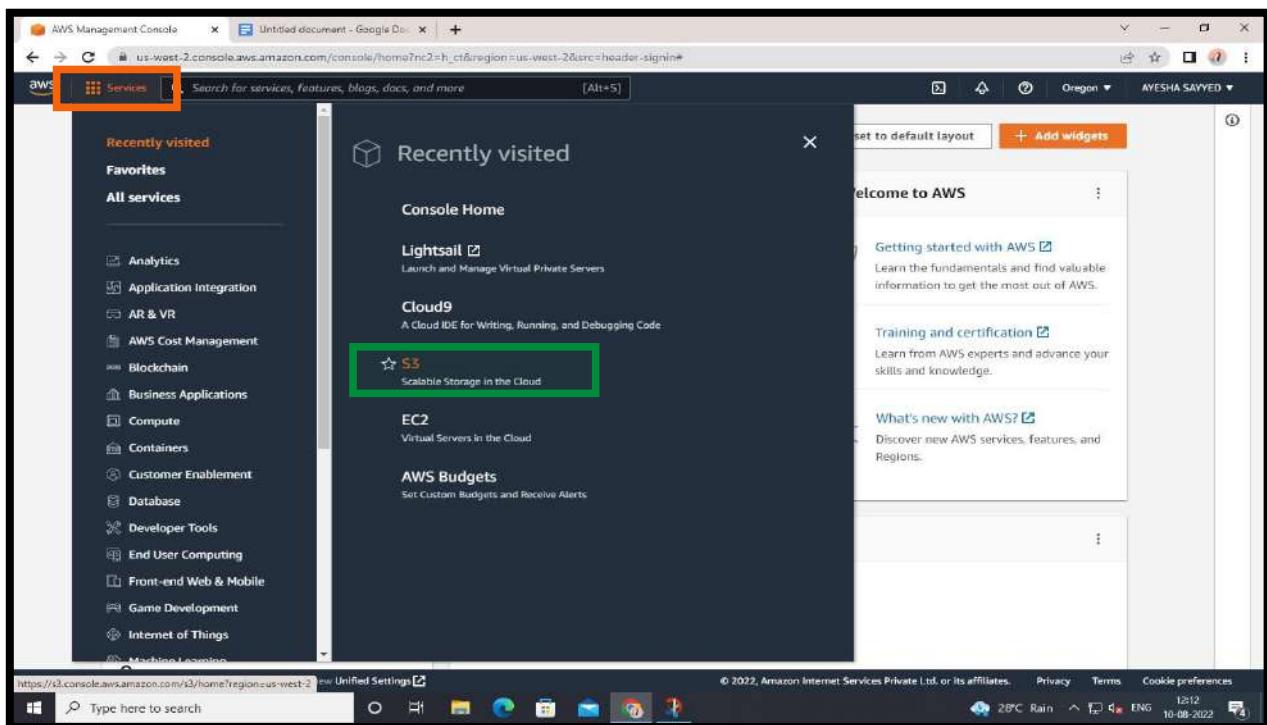
EXPERIMENT – 06

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.

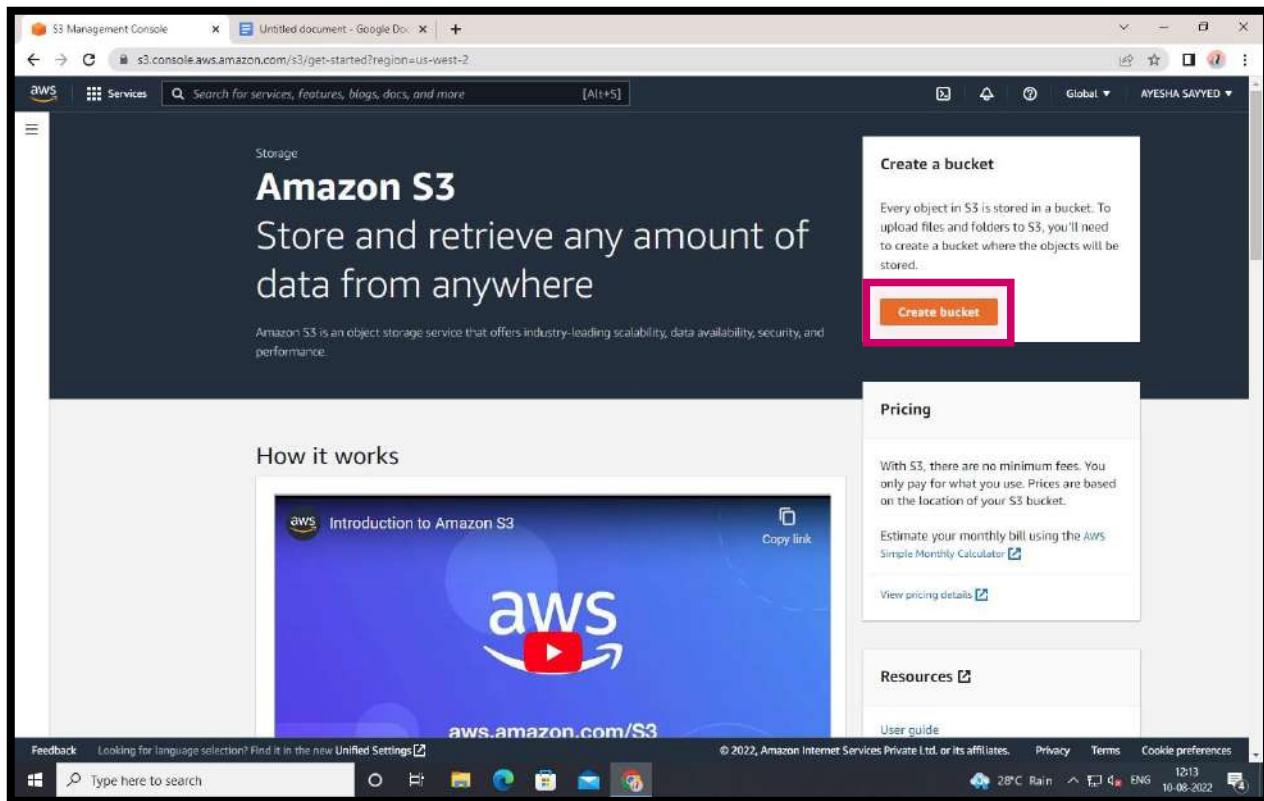
Step 1: AWS Management Console Dashboard.



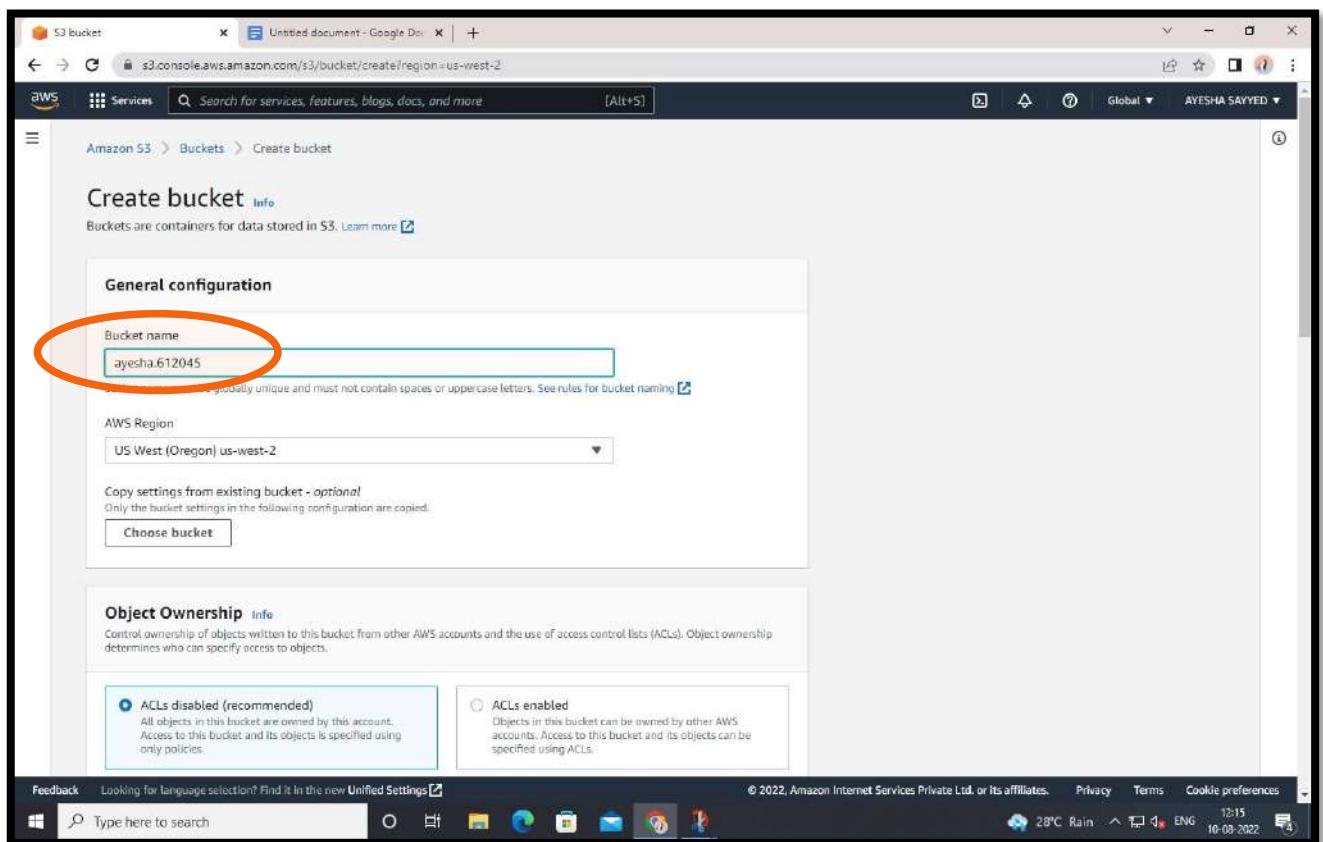
Step 2: Click on Services → Storage → S3



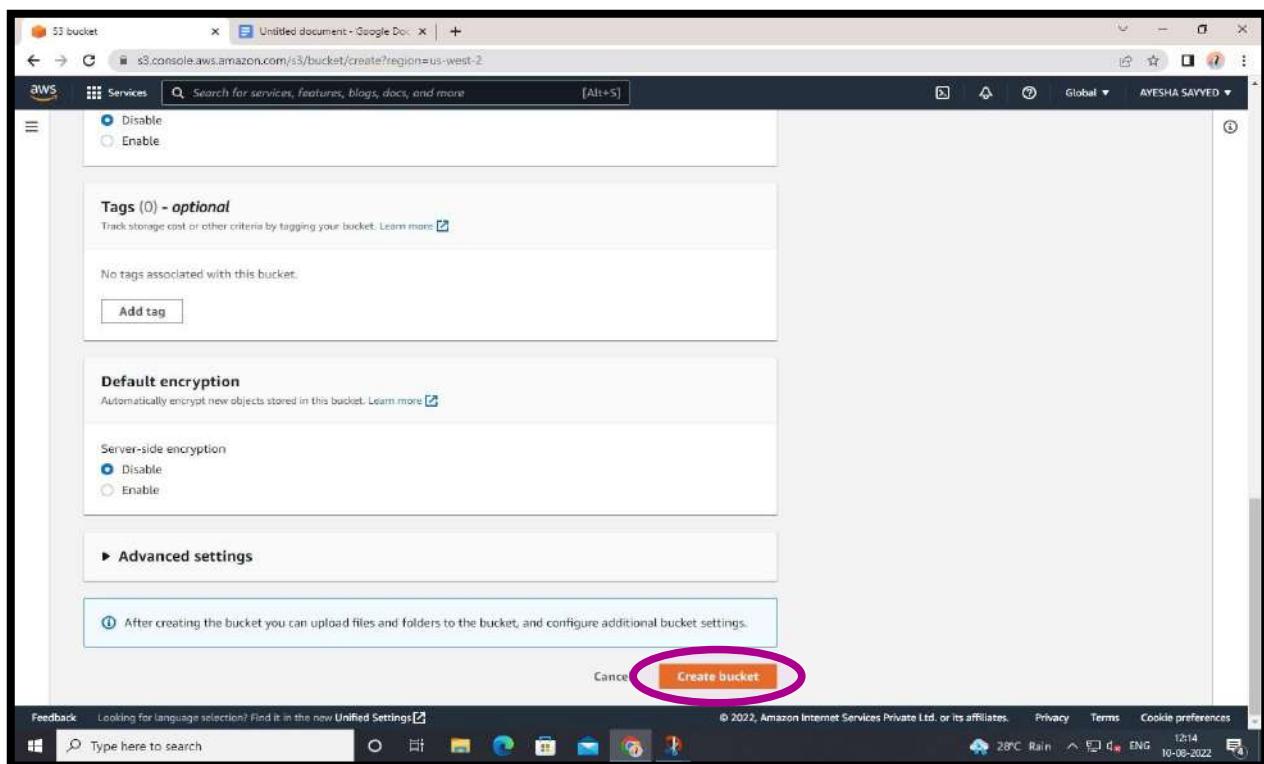
Step 3: Click on “Click Bucket”.



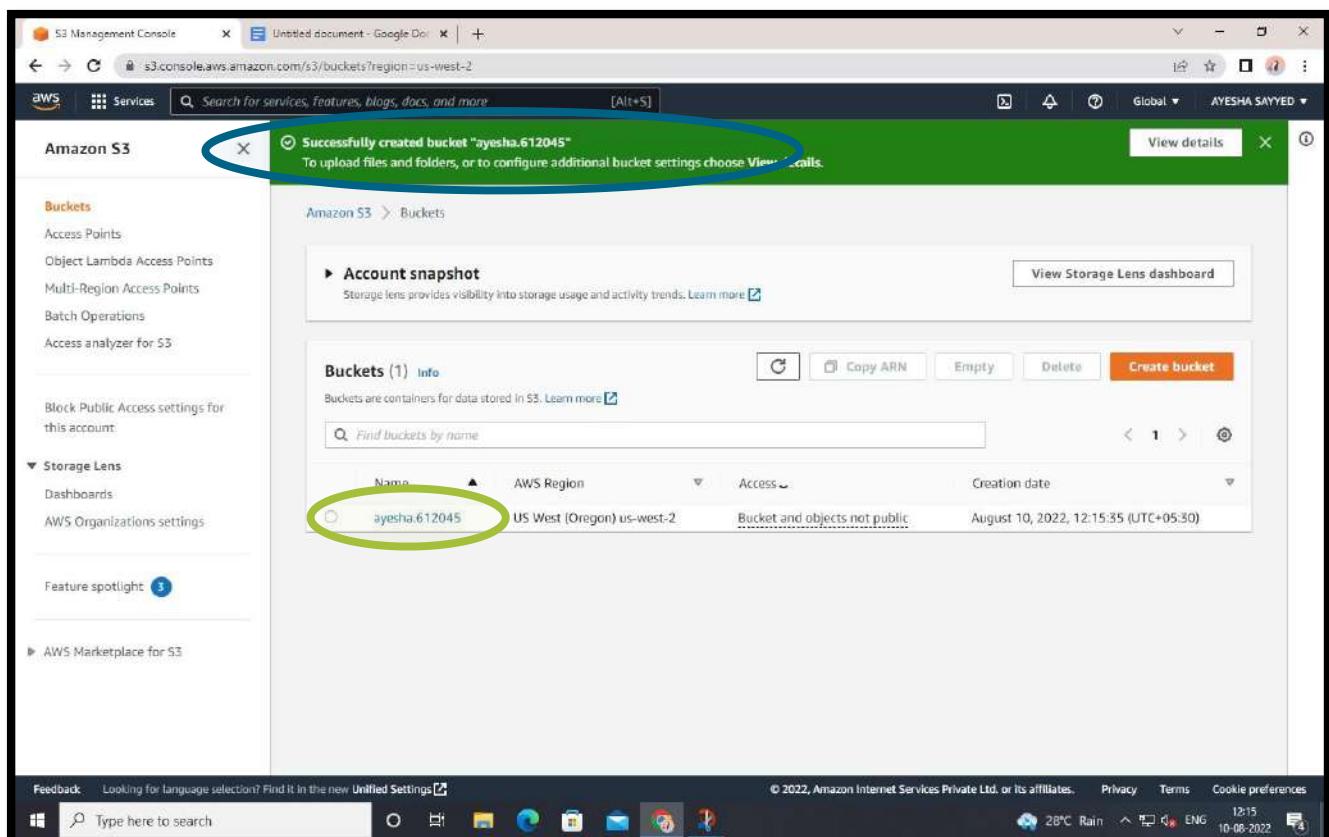
Step 4: Give your bucket a name→Click on Create bucket.



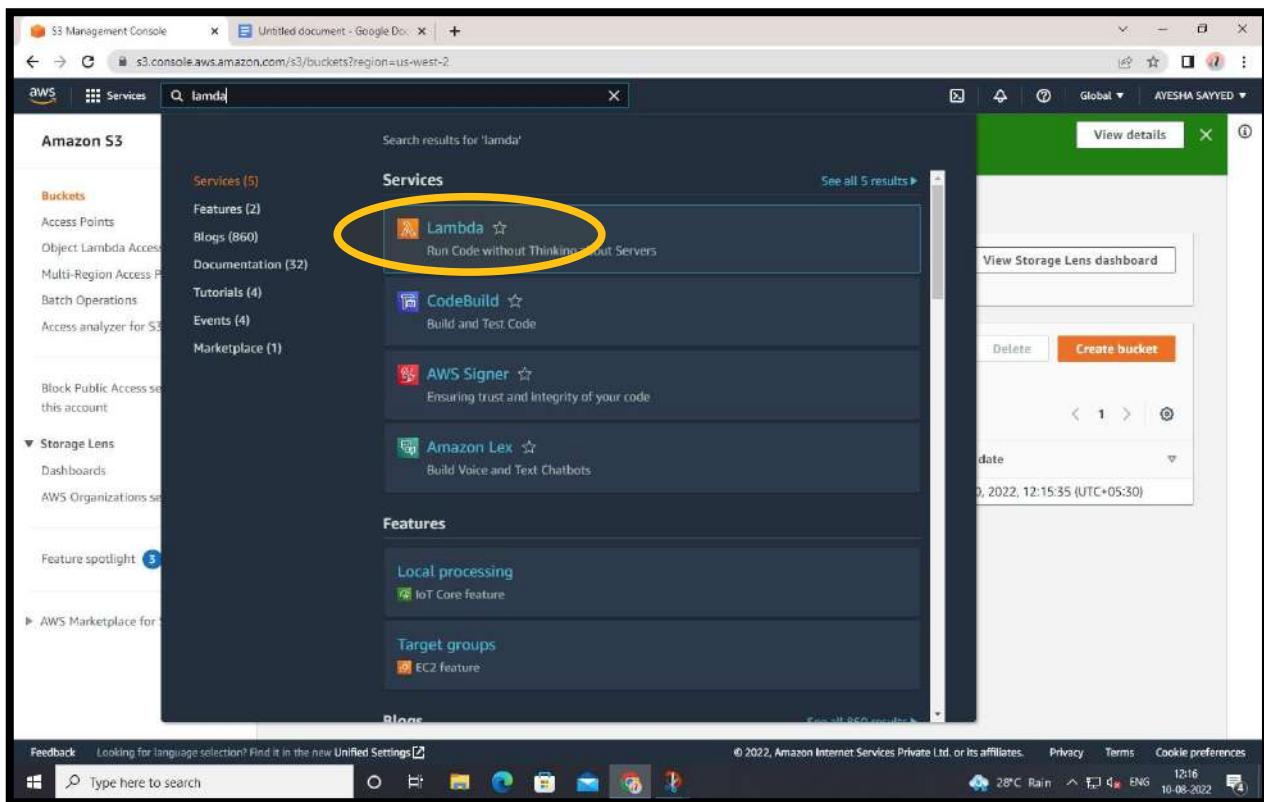
Then click on the button ‘Create Bucket’.



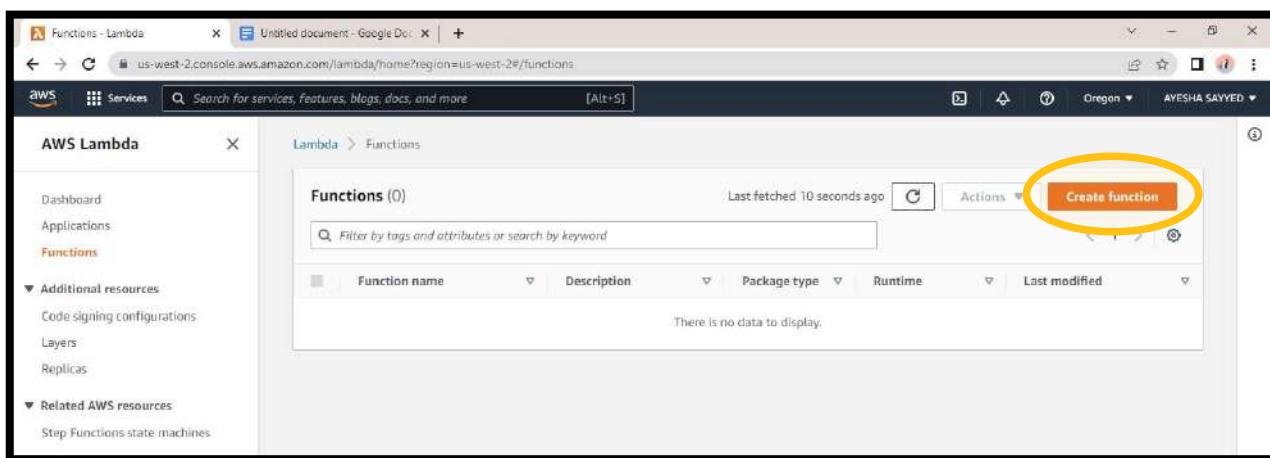
Bucket successfully created.



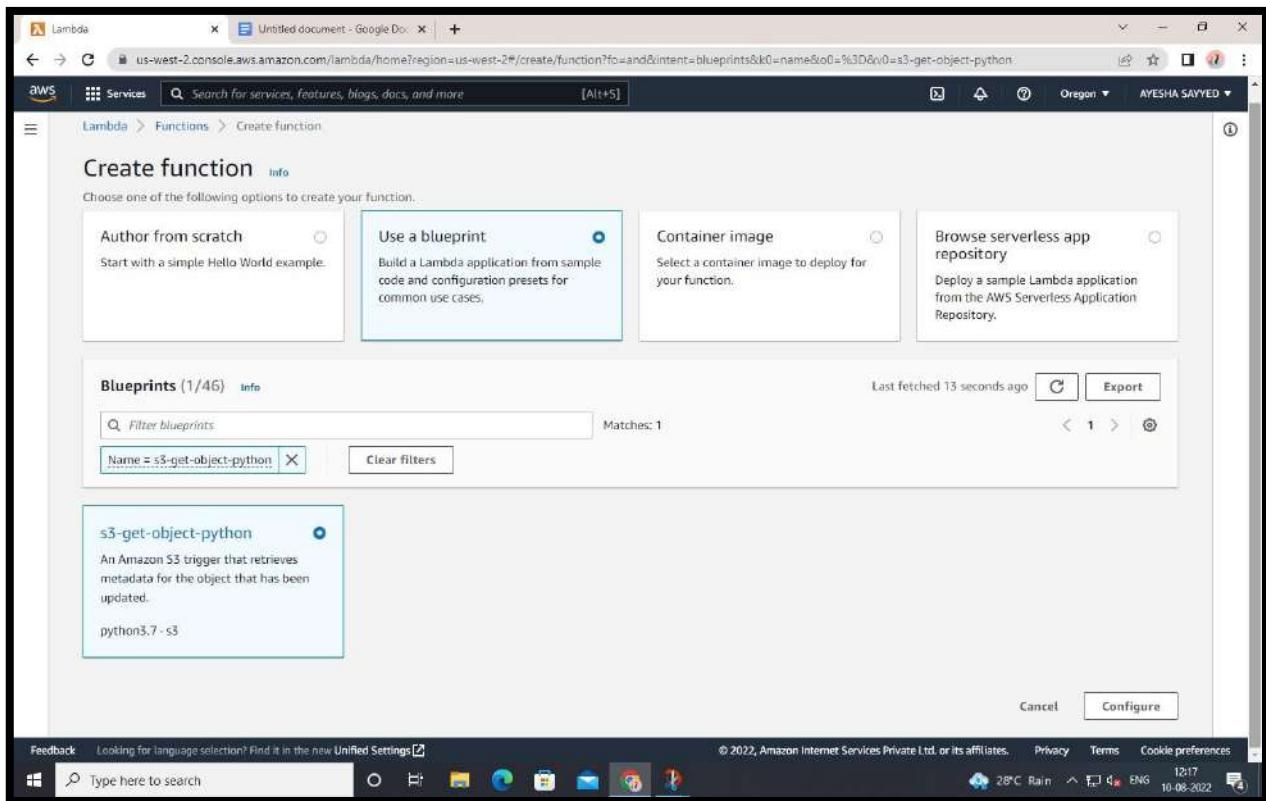
Step 5: Search for ‘Lambda’ and click it.



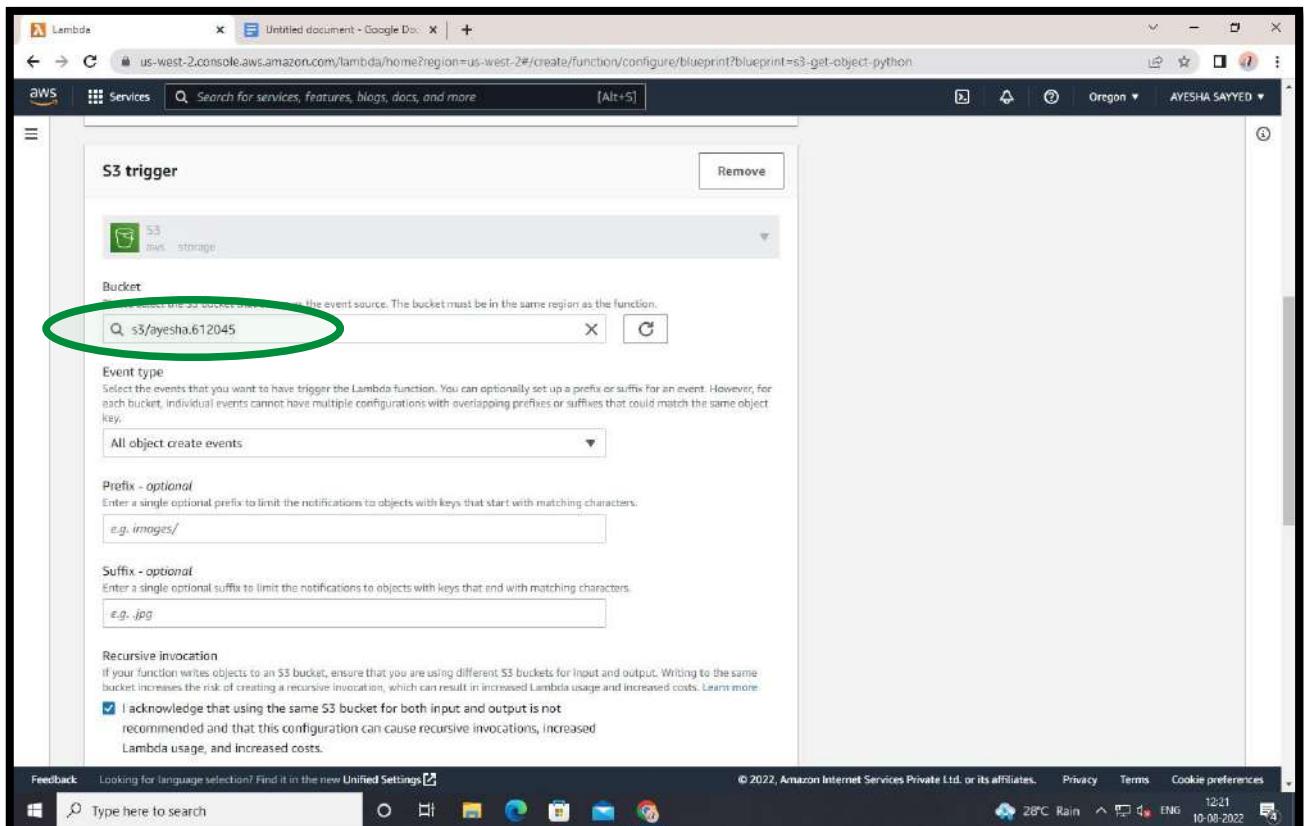
Step 6: Click on ‘Create Function’.



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.

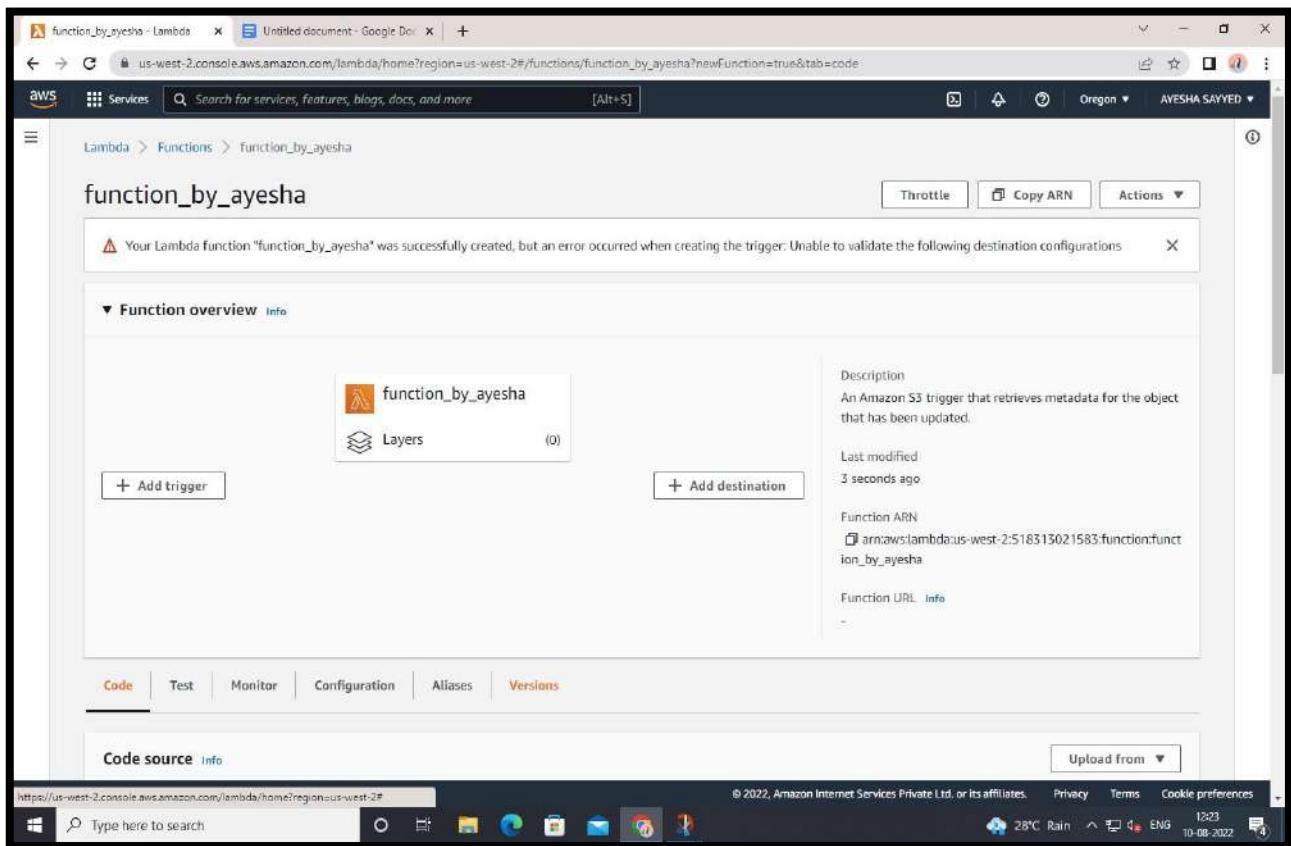


The screenshot shows the AWS Lambda console interface for creating a new function. The code editor contains the following Python script:

```
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     try:
18         response = s3.get_object(Bucket=bucket, Key=key)
19         print("CONTENT TYPE: " + response['ContentType'])
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 has the correct access permissions.'.format(key, bucket))
24         raise e
```

The 'Create function' button at the bottom right is highlighted with a blue oval.

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



```
function_by_ayesha - Lambda | Untitled document - Google Docs | +  
us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/function_by_ayesha?newFunction=true&tab=code  
AWS Services Search for services, features, blogs, docs, and more [Alt+S] Oregon AYESHA SAYYED  
Code Test Monitor Configuration Aliases Versions  
Code source Info Upload from  
File Edit Find View Go Tools Window Test Deploy Changes not deployed  
Go to Anything (Ctrl+P) lambda_function x  
function_by_ayesha lambda_function.py  
1 # This is a sample function that processes an S3 object.  
2  
3 def lambda_handler(event, context):  
4     #print("Received event: " + json.dumps(event, indent=2))  
5  
6     # Get the object from the event and show its content type  
7     bucket = event['Records'][0]['s3']['bucket']['name']  
8     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')  
9     try:  
10         response = s3.get_object(Bucket=bucket, Key=key)  
11         print("CONTENT TYPE: " + response['ContentType'])  
12         print("IMAGE FILE IS BEEN UPLOADED BY AYESHA SAYYED_4SF")  
13     except Exception as e:  
14         print(e)  
15         print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))  
16         raise e  
17  
18  
19  
20  
21  
22  
23  
24  
25
```

Code successfully uploaded / deployed.

Successfully updated the function function_by_ayesha.

```
function_by_ayesha - Lambda | Untitled document - Google Docs | +  
us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/function_by_ayesha?newFunction=true&tab=code  
AWS Services Search for services, features, blogs, docs, and more [Alt+S] Oregon AYESHA SAYYED  
Code source Info Upload from  
File Edit Find View Go Tools Window Test Deploy  
Go to Anything (Ctrl+P) lambda_function x  
function_by_ayesha lambda_function.py  
1 # This is a sample function that processes an S3 object.  
2  
3 def lambda_handler(event, context):  
4     #print("Received event: " + json.dumps(event, indent=2))  
5  
6     # Get the object from the event and show its content type  
7     bucket = event['Records'][0]['s3']['bucket']['name']  
8     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')  
9     try:  
10         response = s3.get_object(Bucket=bucket, Key=key)  
11         print("CONTENT TYPE: " + response['ContentType'])  
12         print("IMAGE FILE IS BEEN UPLOADED BY AYESHA SAYYED_4SF")  
13     except Exception as e:  
14         print(e)  
15         print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))  
16         raise e  
17  
18  
19  
20  
21  
22  
23  
24  
25
```

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

S3 Management Console | Untitled document - Google Docs | +

s3.console.aws.amazon.com/s3/buckets?region=us-west-2

Amazon S3 Services Search for services, features, blogs, docs, and more [Alt+S]

Global AYESHA SAYYED

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

Find buckets by name

Name AWS Region Access Creation date

Name	AWS Region	Access	Creation date
ayesha.612045	US West (Oregon) us-west-2	Bucket and objects not public	August 10, 2022, 12:15:35 (UTC+05:30)

View Storage Lens dashboard

C Copy ARN Empty Delete Create bucket

Block Public Access settings for this account

▼ Storage Lens

Dashboards AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

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

28°C Rain 12:28 ENG 10-08-2022

S3 Management Console | Untitled document - Google Docs | +

s3.console.aws.amazon.com/s3/upload/ayesha.612045?region=us-west-2

Amazon S3 > Buckets > ayesha.612045 > Upload

Upload Info

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

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

Files and folders (1 Total, 54.8 KB)

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
f3b58f860a9e8487e2eb7e853e1f4f8e.jpg	-	image/jpeg	54.8 KB

Remove Add files Add folder

Find by name

Destination

Destination s3://ayesha.612045

Destination details Bucket settings that impact new objects stored in the specified destination.

Permissions Grant public access and access to other AWS accounts.

Properties Specify storage class, encryption settings, tags, and more.

Cancel Upload

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

28°C Rain 12:30 ENG 10-08-2022

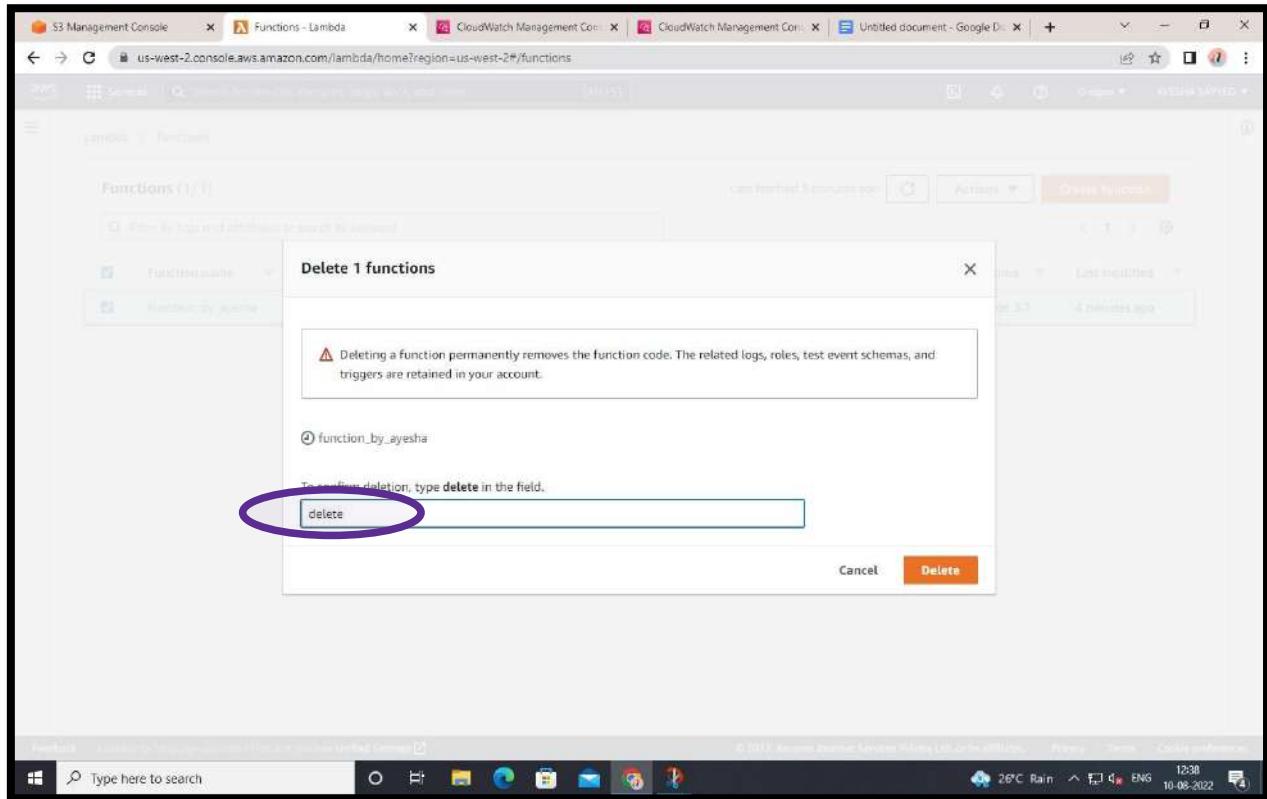
Now the ‘Image’ is successfully uploaded .

The screenshot shows the AWS S3 Management Console interface. At the top, there is a green banner with the message "Upload succeeded" and "View details below." A yellow oval highlights this message. Below the banner, the title "Upload: status" is displayed. A note says "The information below will no longer be available after you navigate away from this page." Under the "Summary" section, it shows the destination "s3://ayesha.612045" with one succeeded file (54.8 KB) and zero failed files. The "Files and folders" tab is selected, showing a table with one item: "f3b50ff860a9e0407e2eb7e853e1f4f8e.jpg" which is an image/jpeg file, 54.8 KB in size, and has a status of "Succeeded". The bottom of the screen shows the Windows taskbar with various icons and the date/time as 10-08-2022.

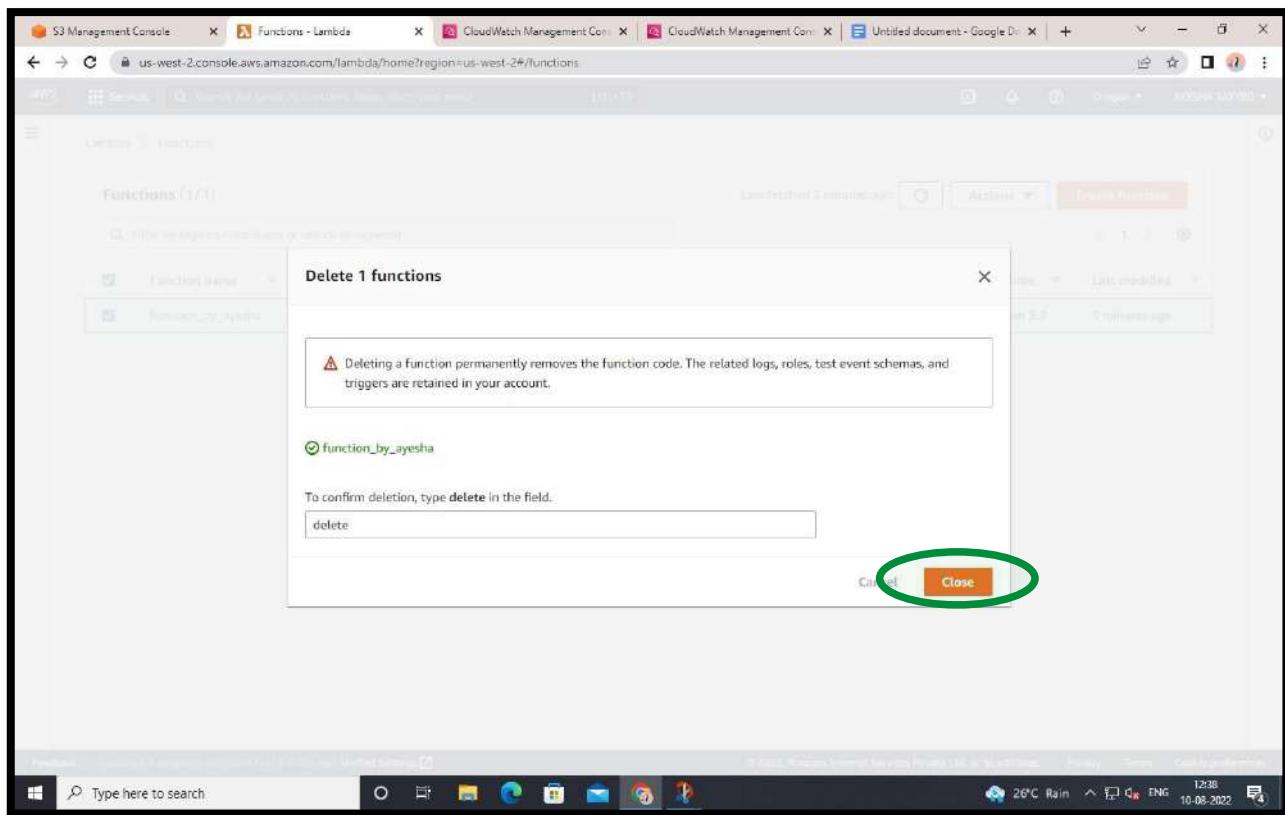
After the image is been uploaded you can check the message by Click the blue link in Log stream→ “Image file uploaded”.

Step 11: Now Delete your ‘Lambda Function’.

The screenshot shows the AWS Lambda Functions console. The title bar includes tabs for "S3 Management Console", "Functions - Lambdas", "CloudWatch Management Con...", "CloudWatch Management Con...", "Untitled document - Google Doc", and "+". The main area shows a table titled "Functions (1/1)". It lists one 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 written in "Python 3.7" and was last modified "3 minutes ago". The "Actions" column contains buttons for "View details", "Test", and "Delete". The "Create function" button is also visible at the top right of the table.



Delete and close the function .



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

The screenshot shows the 'Delete objects' page in the AWS S3 Management Console. At the top, a warning message states: 'Deleting the specified objects can't be undone.' Below this, a table lists two objects:

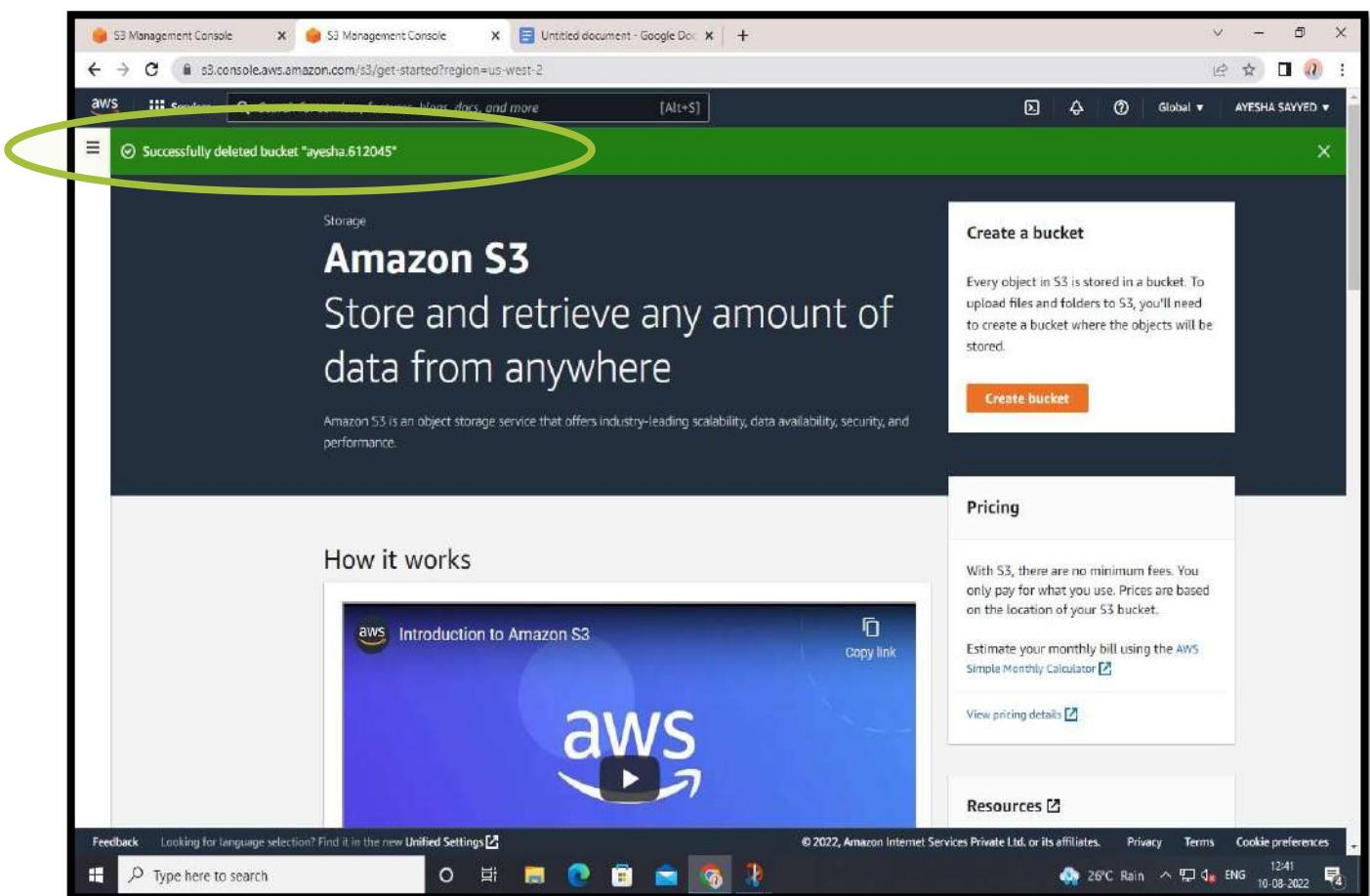
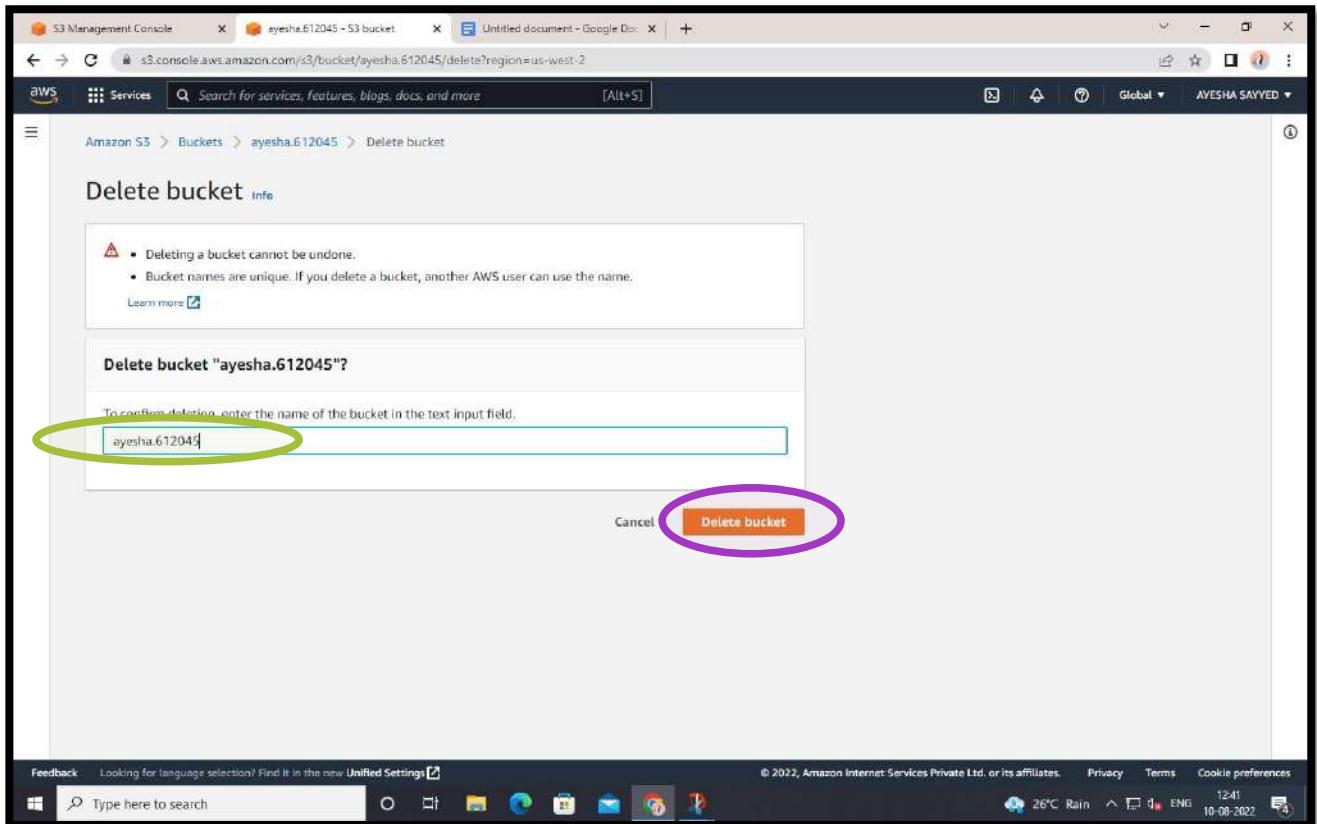
Name	Type	Last modified	Size
computer-laptop-notebook-pen.jpg	jpg	August 10, 2022, 12:35:58 (UTC+05:30)	69.9 KB
f3h58f60a9e8487e2eb7e853e1f4f8e.jpg	jpg	August 10, 2022, 12:30:24 (UTC+05:30)	54.8 KB

Below the table, a section titled 'Permanently delete objects?' contains a text input field with the placeholder 'To confirm deletion, type permanently delete in the text input field.' A purple circle highlights this input field, which contains the text 'permanently delete'. A blue box highlights the 'Delete objects' button at the bottom right of this section.

The screenshot shows the 'Delete objects: status' page in the AWS S3 Management Console. At the top, a green banner displays the message: 'Successfully deleted objects' followed by 'View details below.' A red circle highlights this message. Below the banner, a summary table provides details about the deletion:

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

Below the summary, there are tabs for 'Failed to delete' (0) and 'Configuration'. The 'Failed to delete' tab is selected, showing a table with no results: 'No objects failed to delete.' A blue box highlights the 'Delete objects' button at the bottom right of this section.



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.

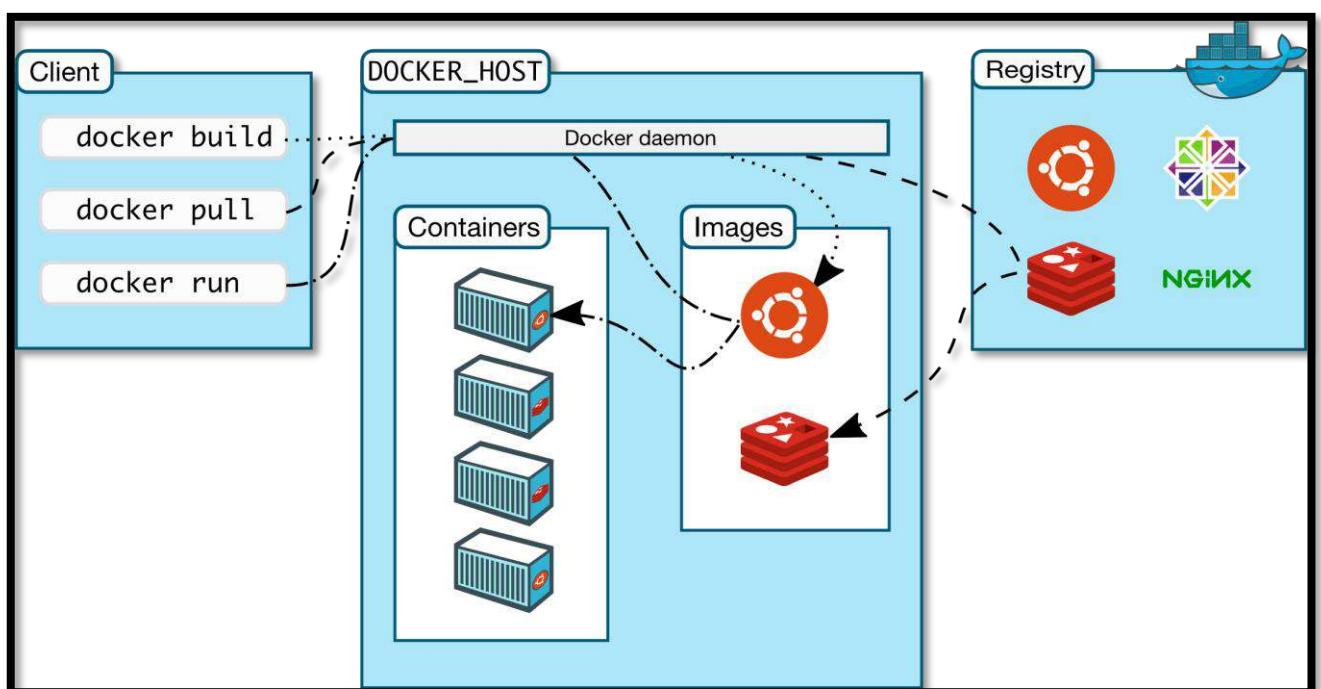
Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

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.

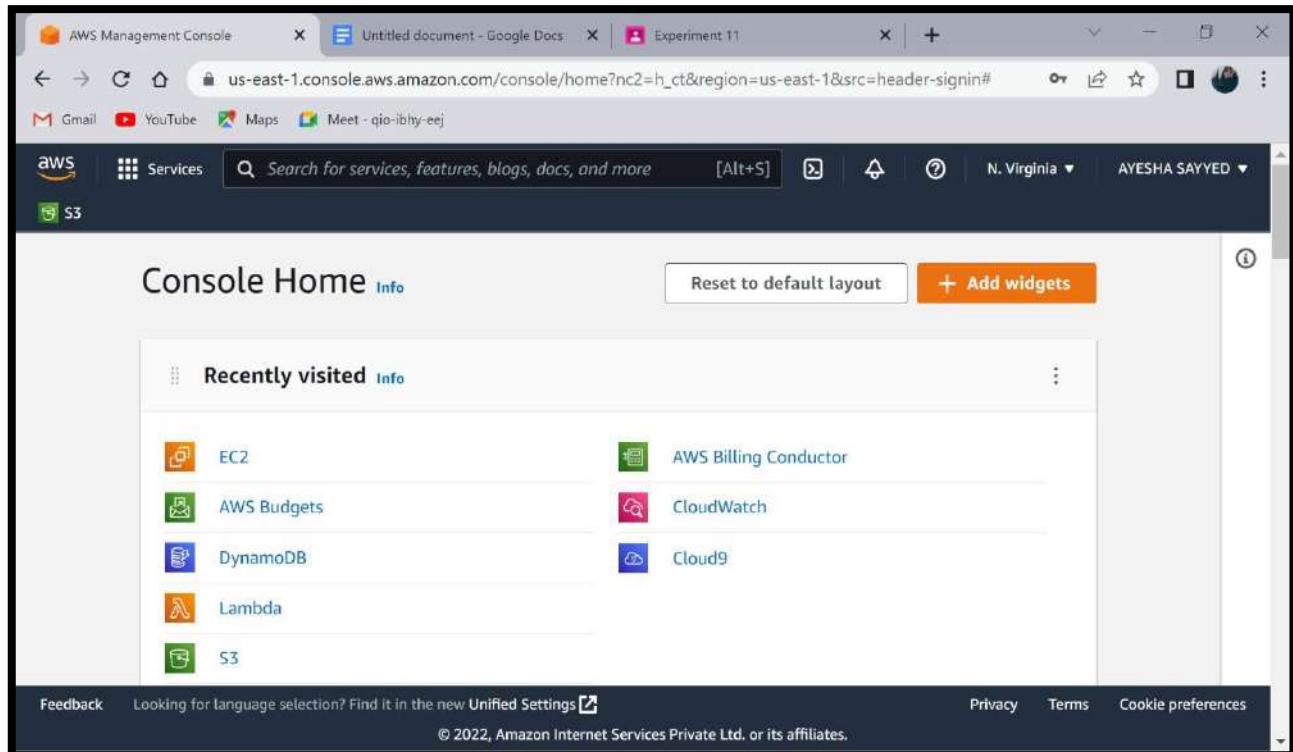
Q3. Why are Containers lightweight?

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.

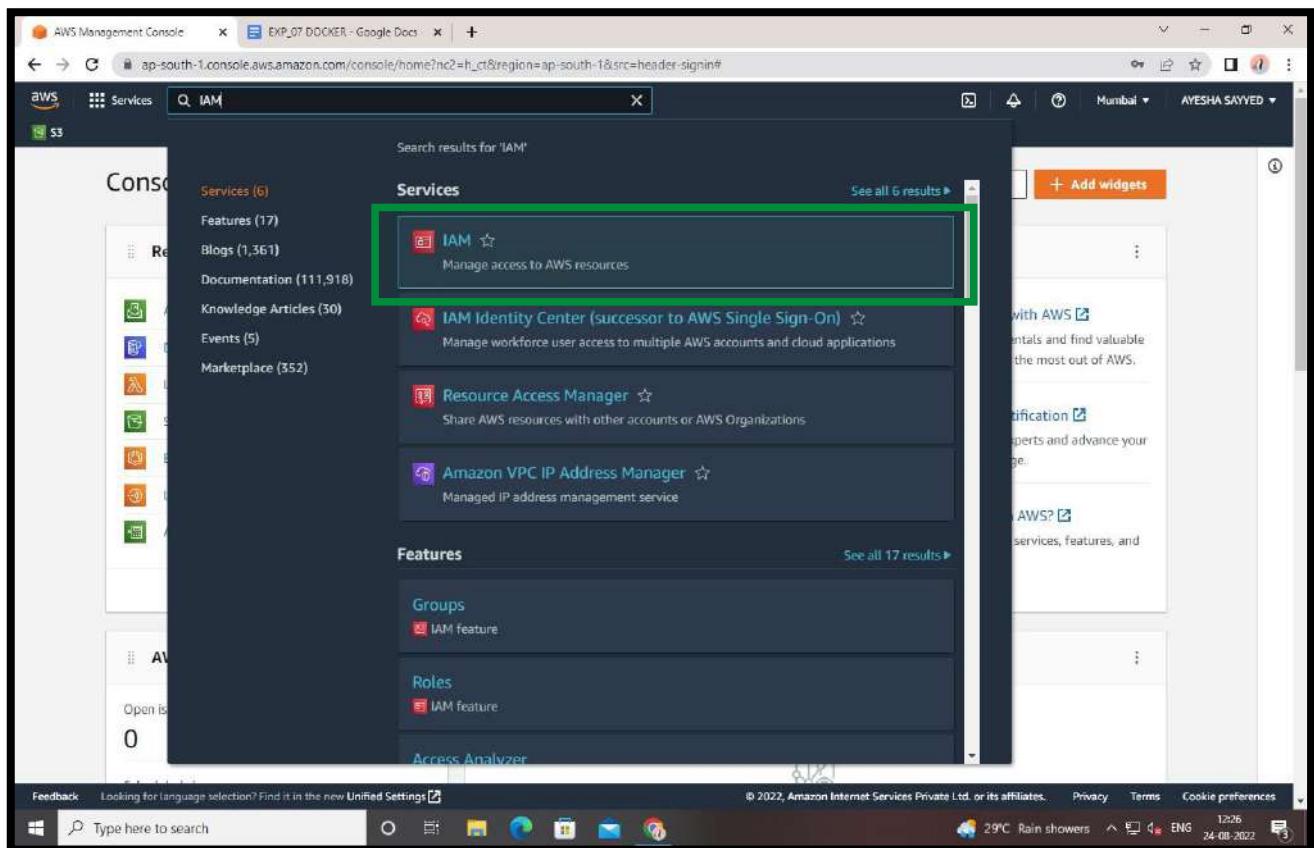
Containers are more lightweight than VMs, as their images are measured in megabytes rather than gigabytes. Containers require fewer IT resources to deploy, run, and manage. Containers spin up in milliseconds. Since their order of magnitude is smaller.

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 options like 'Identity and Access Management (IAM)', 'Access management', 'Access reports', and 'Related controls'. The main area is titled 'Roles (5) Info' and contains a table of existing roles. A red box highlights the 'Create role' button at the top right of the table. Below the table, there's a section titled 'Roles Anywhere' with icons for EC2, Lambda, and IAM.

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. It has three tabs: 'Step 1 Select trusted entity', 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. Under 'Trusted entity type', the 'AWS service' option is selected. Under 'Use case', 'EC2' is selected. At the bottom right, a green box highlights the 'Next Step' button.

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

Container examples: Containers via Docker, PhotonOS, RancherOS.

The screenshot shows the AWS IAM Management Console in a browser window. The URL is `us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles/create?commonUseCase=EC2&step=addPermission&trustedEntityType=AWS_SERVICE`. The page is titled "Permissions policies (Selected 1/764)". A search bar at the top right contains the text "SSM". Below it, a table lists various AWS managed policies. The first policy, "AmazonSSMManagedInstanceCore", has a checked checkbox and is highlighted with a red box. The table columns are "Policy name", "Type", and "Description". The "Description" column for the selected policy states: "This policy will soon be deprecated. Please use AmazonSSMManagedInstanceCore policy instead." At the bottom of the table, there is a section titled "Set permissions boundary - optional" with a note: "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." On the far right of the table, there are "Cancel", "Previous", and "Next" buttons. The "Next" button is highlighted with a red box.

Policy name	Type	Description
<input checked="" type="checkbox"/>  AmazonSSMManagedInstanc...	AWS m...	This policy will soon be deprecated. Please use AmazonSSMManagedInstanceCore policy instead.
<input type="checkbox"/>  AmazonSSMAutomat...	AWS m...	Provides access to view automation executions and send approval decisions to autom...
<input type="checkbox"/>  AmazonSSMManag...	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"/>  AWSResourceAcces...	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"/>  AWSResourceAcces...	AWS m...	Provides full access to AWS Resource Access Manager
<input type="checkbox"/>  AWSCloud9SSMMinis...	AWS m...	This policy will be used to attach a role on a InstanceProfile which will allow Cloud9 to ...
<input type="checkbox"/>  AWSResourceAcces...	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”.

The screenshot shows the AWS IAM Management Console interface for creating a new role. The top navigation bar includes tabs for 'IAM', 'Roles', and 'Create role'. The main area is titled 'Name, review, and create'.
Role details:

- Role name:** docker.Role45 (highlighted with a yellow box)
- Description:** 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.

Cancel Previous Create role (highlighted with a yellow box)

Role Created Successfully!!

The screenshot shows the AWS IAM Management Console. In the top navigation bar, there is a message: "New! Securely access AWS services from your data center with IAM Roles Anywhere. Learn more". Below this, a green notification bar says "Role docker.Role45 created" with a "View role" button. The left sidebar has a "Roles" section highlighted. The main content area shows a table with one row for "Roles (6)". The table includes columns for "Role name" and "Trusted entities".

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

The screenshot shows the AWS EC2 Management Console. On the left, a sidebar lists various services under "Recently visited" and "Favorites". A modal window titled "Recently visited" is open, showing a list of services: EC2 (Virtual Servers in the Cloud), Compute Home, AWS Budgets, DynamoDB (Managed NoSQL Database), Lambda, S3 (Scalable Storage in the Cloud), Elastic Container Service (Highly secure, reliable, and scalable way to run containers), Lightsail (Launch and Manage Virtual Private Servers), and AWS Billing Conductor. To the right of the modal, there are sections for "Account attributes" and "Additional information". The bottom status bar shows the URL "https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Home" and the copyright notice "© 2022, Amazon Internet Services Private Ltd. or its affiliates."

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

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Instances' section, there is a 'Launch instance' button highlighted with a green box. The main content area displays various EC2 resource statistics and a callout box about Microsoft SQL Server Always On availability groups. The right sidebar contains sections for 'Account attributes' and 'Explore AWS'.

Now give name to Your Instance.

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' section, the 'Name' field is filled with 'Ayesha_Instance_145'. Other configuration options like AMI, instance type, and storage are visible. A summary panel on the right shows the selected settings: 1 instance of Amazon Linux 2 Kernel 5.10 AMI, t2.micro instance type, and 1 volume(s) - 8 GB storage. A 'Launch instance' button is at the bottom right.

Select Amazon Linux.

The screenshot shows the AWS Management Console interface for launching an EC2 instance. On the left, there's a sidebar with 'Services' and a search bar. The main area is titled 'Application and OS Images (Amazon Machine Image)'. It features a 'Quick Start' section with icons for Amazon Linux, macOS, Ubuntu, Windows, and Red Hat. A search bar at the top says 'Search our full catalog including 1000s of application and OS images'. Below it, a card for 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type' is displayed, including details like AMI ID, Virtualization type, and ENA support. To the right is a 'Summary' section with fields for 'Number of instances' (set to 1), 'Software Image (AMI)' (selected as 'Amazon Linux 2 Kernel 5.10 AMI'), 'Virtual server type (instance type)' (set to 't2.micro'), and 'Launch instance' button. A note about free tier usage is shown. At the bottom, there are links for feedback, search, and system status.

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

This screenshot shows the 'Create key pair' dialog box overlaid on the EC2 launch instance page. The dialog has a title 'Create key pair' and a sub-instruction: 'Key pairs allow you to connect to your instance securely.' It asks for a 'Key pair name' (with 'ayesha_keypair' entered) and a 'Key pair type' (with 'RSA' selected). Below that, 'Private key file format' options (.pem and .ppk) are listed. At the bottom are 'Cancel' and 'Create key pair' buttons, with the latter highlighted by a red box.

The screenshot shows the AWS EC2 Launch Instance wizard. On the left, under 'Network settings', it lists a VPC (vpc-0cf924feb1b794701) and a subnet (No preference). It includes options for creating or selecting a security group, and a note about allowing SSH traffic from anywhere. On the right, the 'Summary' section shows 1 instance being launched with an Amazon Linux 2 AMI, t2.micro instance type, and a new security group. A tooltip indicates a free tier of 750 hours of t2.micro usage. At the bottom right is a 'Launch instance' button.

Instance Created Successfully! → Now View all Instances.

The screenshot shows the AWS EC2 Instances page. A red circle highlights the 'Success' message: 'Successfully initiated launch of instance (i-0f90b0e584310B6e3)'. Below this, a green box highlights the 'View all Instances' button. The page also includes sections for 'Next Steps' (notified of estimated charges, how to connect to your instance), and a 'Feedback' bar at the top.

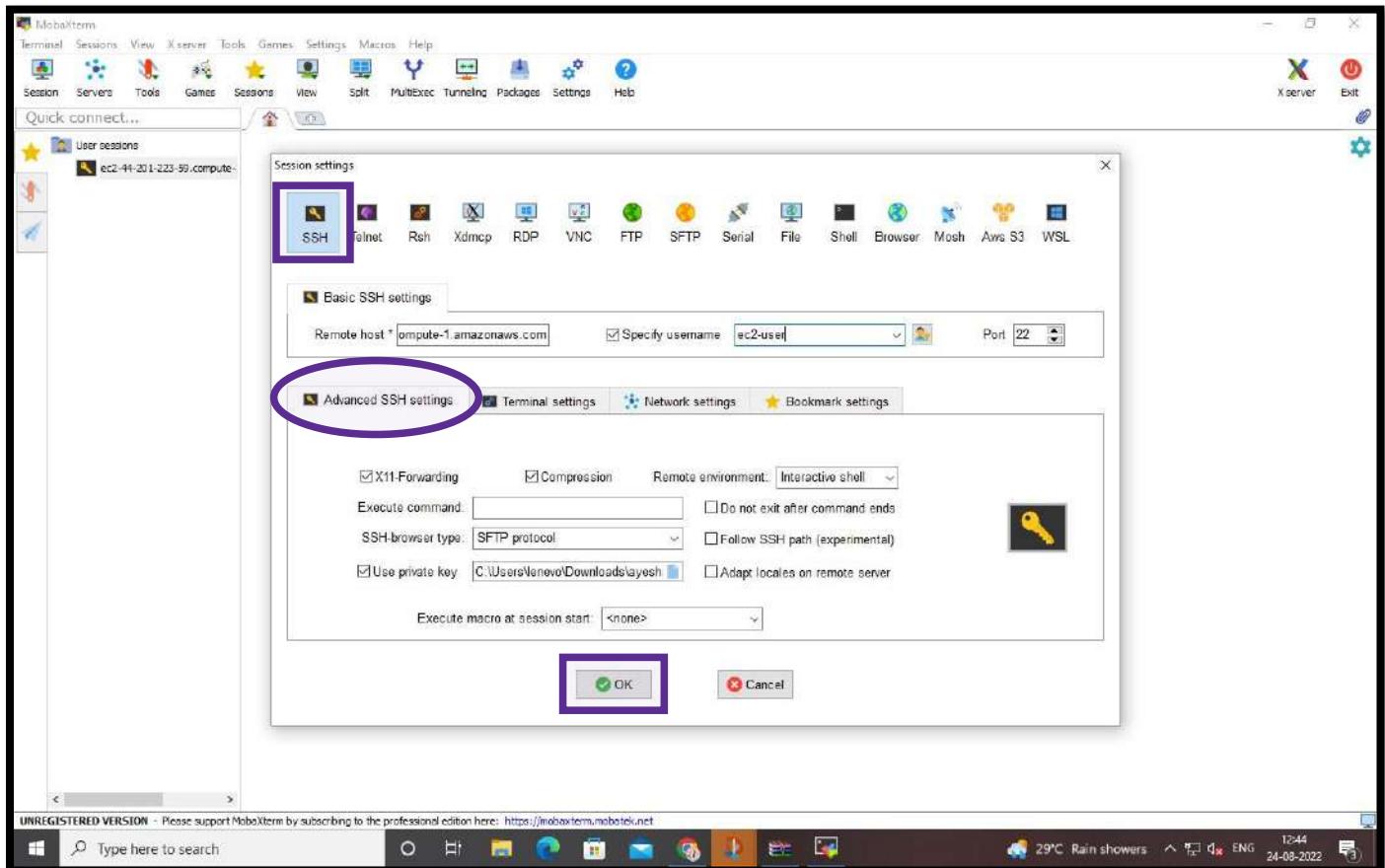
Click on Instance ID

The screenshot shows the AWS EC2 Management Console. In the left sidebar, under 'Instances', 'Instances' is selected. The main area displays a table titled 'Instances (1/1) Info' with one row. The row contains columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The 'Instance ID' column shows 'i-0f90b0e58431086e3'. This specific cell is highlighted with a green box. The 'Instance state' column shows 'Running'. The 'Instance type' column shows 't2.micro'. The 'Status check' column shows 'Initializing'. The 'Alarm status' column shows 'No alarms'. The 'Availability Zone' column shows 'us-east-1c'. At the top right of the table, there are buttons for 'Connect', 'Actions', and 'Launch instances'.

Go to Connect Instance → SSH Client → copy Public DNS

The screenshot shows the 'Connect to instance' dialog box. The 'SSH client' tab is selected. It displays the instance ID 'i-0f90b0e58431086e3 (Ayesha_instance_45)'. Below the instance ID, there are three numbered steps: 1. Open an SSH client., 2. Locate your private key file. The key used to launch this instance is 'ayesha_keypair.pem', and 3. Run this command, if necessary, to ensure your key is not publicly viewable. A note below these steps says 'Public DNS copied' with a circled checkmark. To the right of the steps, it shows the Public IPv4 address '44.201.223.59' and the Public IPv4 DNS name 'ec2-44-201-223-59.compute-1.amazonaws.com'. At the bottom of the dialog, there is an example command: 'ssh -i "ayesha_keypair.pem" ec2-user@ec2-44-201-223-59.compute-1.amazonaws.com'. A note in a box at the bottom right says: '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.'

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



Connected to TERMINAL.

```
ec2-44-201-223-59.compute-1.amazonaws.com (ec2-user)
MobaXterm Personal Edition v22.1
(SSH client, X server and network tools)

SSH session to ec2-user@ec2-44-201-223-59.compute-1.amazonaws.com
  • Direct SSH : ✓
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✘ (disabled or not supported by server)

For more info, ctrl+click on help or visit our website.

[ec2-user@ip-172-31-95-143 ~]$
```

The terminal window displays the connection details and system information. It shows the host is Amazon Linux 2 AMI. It also indicates that 3 package(s) are needed for security, and provides instructions to run 'sudo yum update' to apply all updates.

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

```

Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
3 ec2-44-201-223-59.compute-1.amazonaws.com (ec2-user)
/home/ec2-user/ Amazon Linux 2 AMI
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.r0.41-5.i5 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

Transaction Summary
Install 1 Package (+4 Dependent packages)

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

UNREGISTERED VERSION - Please support MobaTerm by subscribing to the professional edition here: https://mobaterm.mobatek.net

```

Installation Completed

```

Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
3 ec2-44-201-223-59.compute-1.amazonaws.com (ec2-user)
/home/ec2-user/ Amazon Linux 2 AMI
Installing:
  docker
Installing for dependencies:
  containerd.x86_64           1.6.6-1.amzn2
  libcgroup.x86_64             0.41-21.amzn2
  pigz.x86_64                  2.3.4-1.amzn2.0.1
  runc.x86_64                  1.1.3-1.amzn2
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”

```

[3 ec2-44-201-223-59.compute-1.amazonaws.com (ec2-user)]
Is this ok [y/N]: y
Installed size: 260 M
Is this ok [y/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                                         76 MB/s | 69 MB  00:00:00
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  : docker-20.10.17-1.amzn2.x86_64        1/5
  Verifying  : runc-1.1.3-1.amzn2.x86_64             2/5
  Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64         3/5
  Verifying  : containerd-1.6.6-1.amzn2.x86_64       4/5
  Verifying  : libcgroup-0.41-21.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 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
7cb8d4ad746dd4: Pull complete
e7a561826262: Pull complete
7247fe5c182: 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]#

```

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

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

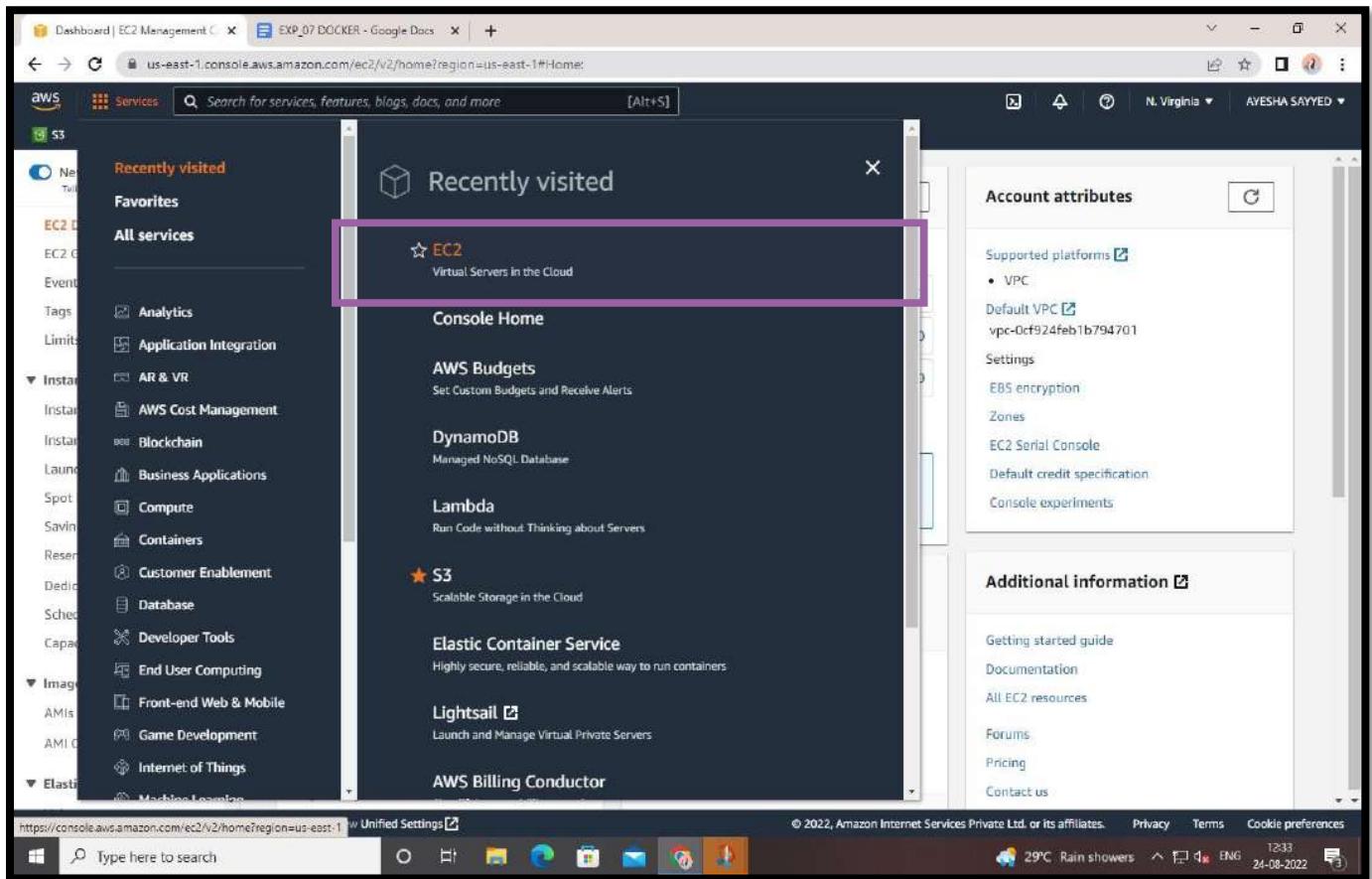
```

[3 ec2-44-201-223-59.compute-1.amazonaws.com (ec2-user)]
Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64           3/5
Verifying : containerd-1.6.6-1.amzn2.x86_64          4/5
Verifying : libcgroup-0.41-21.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 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
7cb8d4ad746dd4: Pull complete
e7a561826262: Pull complete
7247fe5c182: 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]# dockrt images
bash: dockrt: command not found
[root@ip-172-31-95-143 ec2-user]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 287d6430f78d 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
/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/1.23.1
2022/08/24 07:23:07 [notice] 1#1: built by gcc 10.2.1-102110 (Debian 10.2.1-6)
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

```

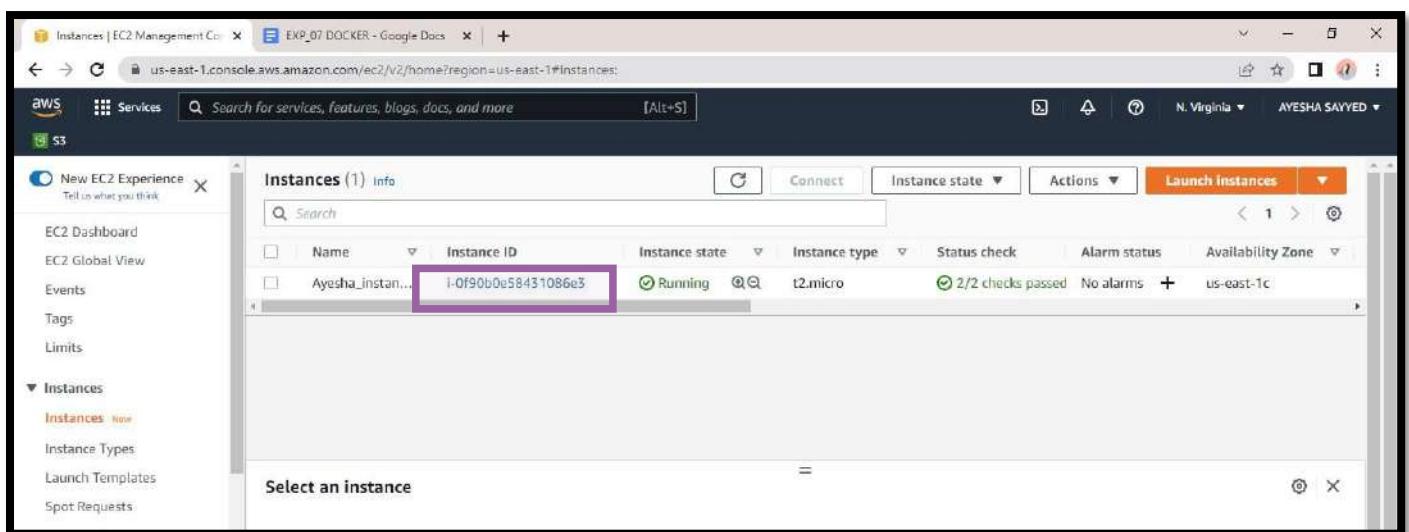
UNREGISTERED VERSION - Please support Mobaxterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 13: Go to EC2 again.



The screenshot shows the AWS Management Console interface. On the left, there's a sidebar with 'Recently visited' and 'Favorites' sections. The 'EC2' service card is highlighted with a purple box. The main content area displays various AWS services like AWS Budgets, DynamoDB, Lambda, S3, and Lightsail. On the right, there are sections for 'Account attributes' and 'Additional information'. The browser address bar shows the URL <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1>.

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



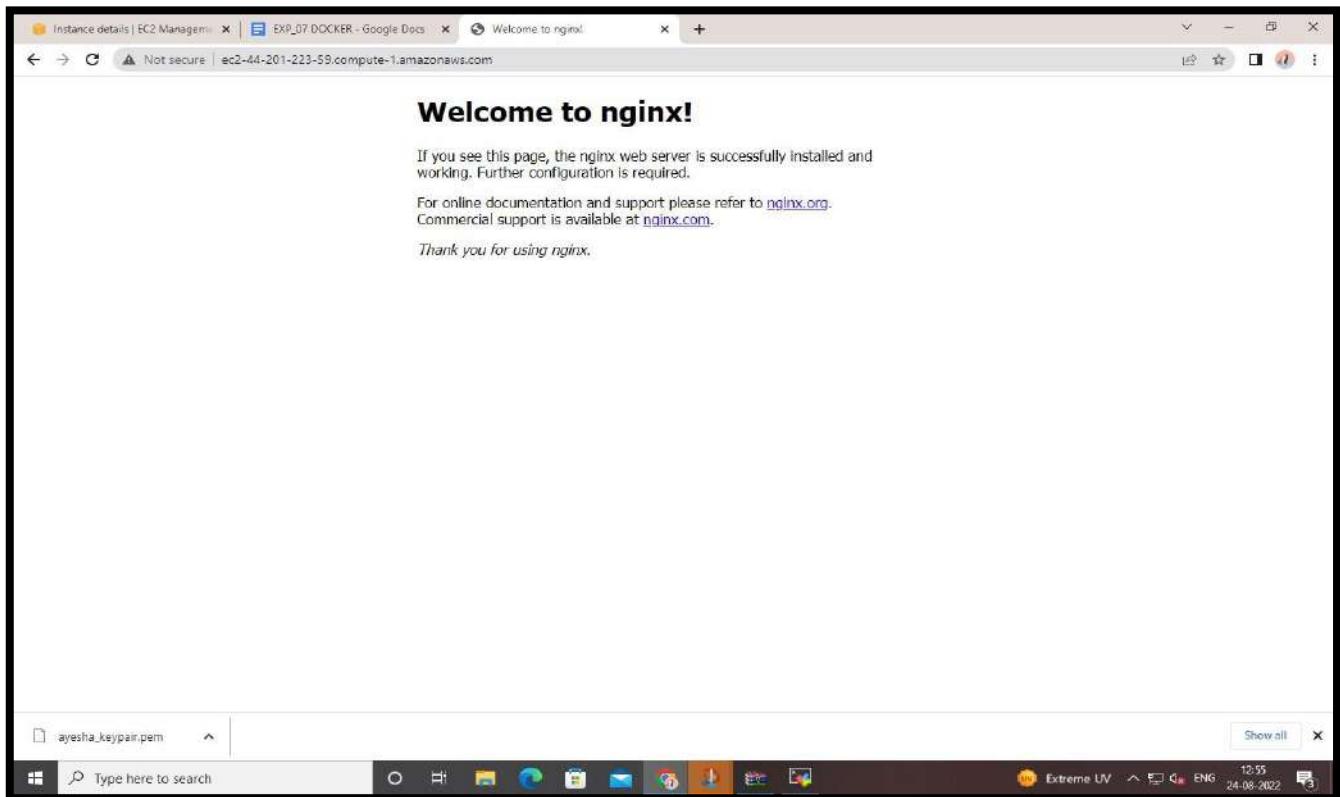
The screenshot shows the AWS EC2 Instances page. It displays a table with one instance: 'Ayesha_instan...' (Instance ID: i-0f90b0e58431086e3). The 'Instance ID' column is highlighted with a purple box. The instance is listed as 'Running' with an 't2.micro' type. The browser address bar shows the URL <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#instances>.

The screenshot shows the AWS EC2 Instance Details page for an instance named 'Ayesha_instance_45'. The 'Details' tab is selected. A green box highlights the 'Public IPv4 DNS copied' button, which is located in the 'Networking' section under the 'Private IP4 addresses' heading. The instance has a Public IP address of 44.201.223.59 and a Private IP DNS name of ip-172-31-95-143.ec2.internal.

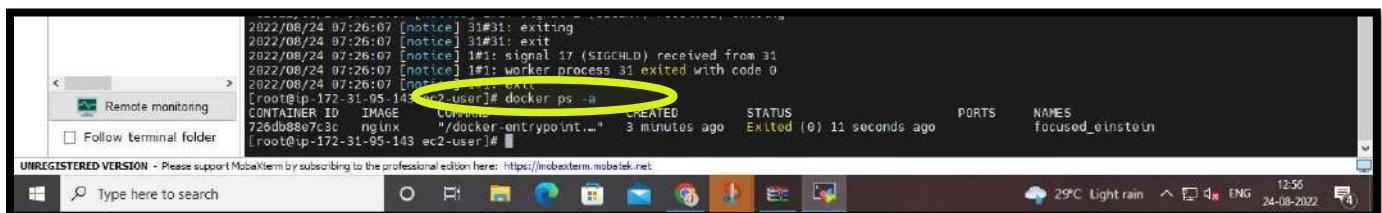
Step 15: Go to Google and Paste the Link

The screenshot shows a browser window with the URL 'ec2-44-201-223-59.compute-1.amazonaws.com' highlighted in the address bar. The browser has several tabs open, including 'Instance details | EC2 Management', 'EXP_07 DOCKER - Google Docs', and 'New Tab'. The main content area displays the Google homepage with a lion and a crow on a boat background. The search bar contains 'Search Google or type a URL'. Below the search bar are several icons for services like Google, YouTube, and phpMyAdmin. The bottom of the screen shows a taskbar with various application icons and system status indicators.

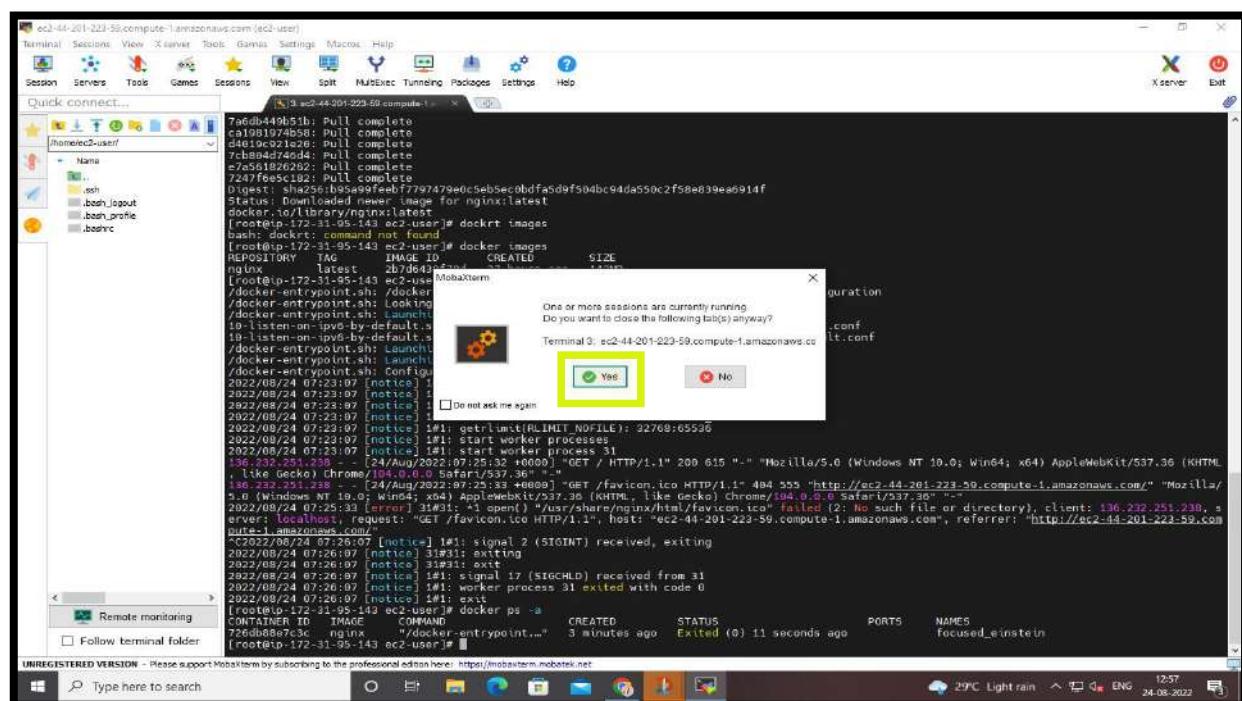
Welcome to nginx!



Run command → “docker ps -a”



Now close the tab



Step 16: Terminate Instances.

The screenshot shows the AWS EC2 Management Console. In the top navigation bar, the URL is `us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#instances:`. The main area displays a table titled "Instances (1/1) Info" with one row. The row contains the following columns: Name (Ayesha_instan...), Instance ID (i-0f90b0e58431086e3), Instance state (Running), and Instance type (t2.micro). To the right of the table is a "Actions" dropdown menu with options: Stop instance, Start instance, Reboot instance, Hibernate instance, and Terminate instance. The "Terminate instance" option is highlighted with a red oval. Below this, a modal dialog box titled "Terminate instance?" is open. 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?". A checkbox is checked next to the instance ID "i-0f90b0e58431086e3 (Ayesha_Instance_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 of the dialog are "Cancel" and "Terminate" buttons, with the "Terminate" button highlighted with a red rectangle.

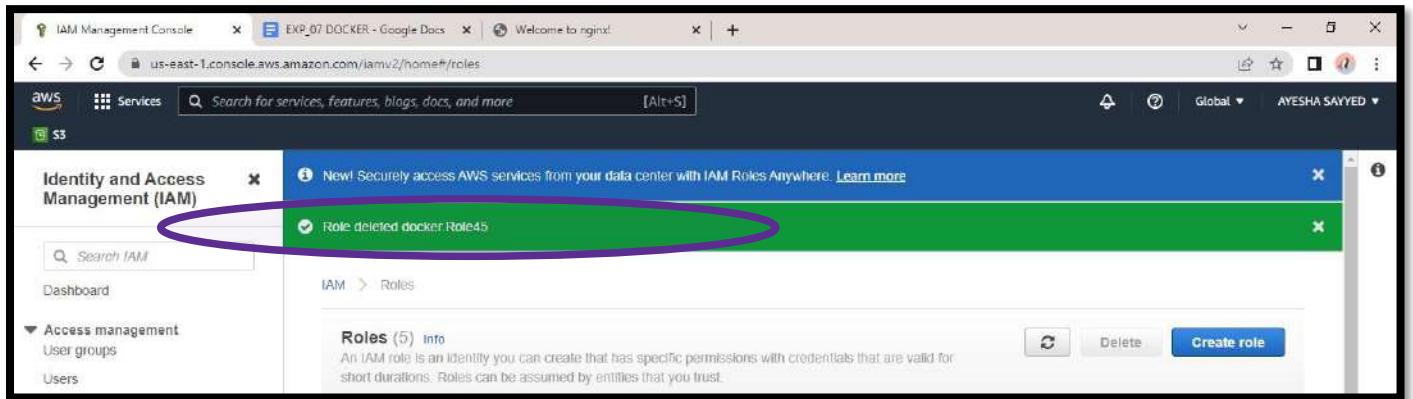
Successfully Terminated!!

The screenshot shows the AWS EC2 Management Console after the instance has been terminated. A green success message at the top of the screen reads "Successfully terminated i-0f90b0e58431086e3". The main area displays the same "Instances (1/1) Info" table as before, but the instance status is now "Shutting-down". The "Actions" dropdown menu is still visible on the right. The "Terminate instance" option from the previous step is circled with a pink oval.

Step 17: Now Delete the Role in IAM .



Role Deleted Successfully!



EXPERIMENT – 08

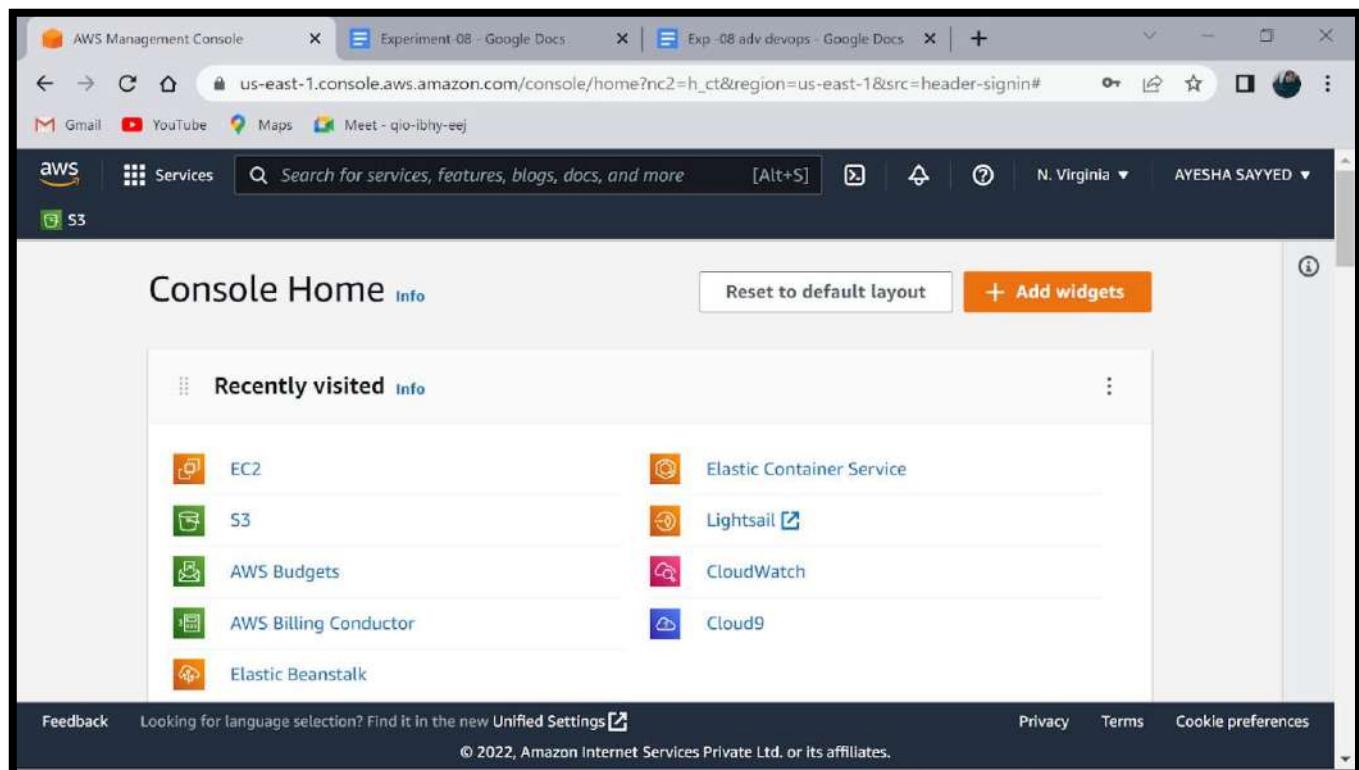
Q1. What is hub.docker.com?

- 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 interface. In the top navigation bar, there are three tabs: 'AWS Management Console', 'Experiment-08 - Google Docs', and 'Exp -08 adv devops - Google Docs'. Below the tabs, the URL is 'us-east-1.console.aws.amazon.com/console/home?nc2=h_ct®ion=us-east-1&src=header-signin#'. The search bar at the top has the word 'EC2' typed into it. To the right of the search bar are several icons: a magnifying glass, a refresh arrow, a bell, a question mark, and a user profile. The main content area is titled 'Search results for 'EC2'' and shows a list of services. A pink rectangular box highlights the first result, 'EC2', which is described as 'Virtual Servers in the Cloud'. Other results listed are 'EC2 Image Builder' and 'AWS Compute Optimizer'. On the left side, there's a sidebar with links like 'Features (46)', 'Blogs (1,802)', 'Documentation (131,392)', 'Knowledge Articles (30)', 'Tutorials (19)', 'Events (8)', and 'Marketplace (1,571)'. At the bottom of the page, there are links for 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

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

The screenshot shows the AWS EC2 Dashboard. The top navigation bar includes tabs for 'Dashboard | EC2', 'Experiment-08 - G', 'G What is docker hu', 'Classes', 'Experiment-8 - G', and '+'. The URL is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home'. The sidebar on the left has sections for 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', and 'Instances'. Under 'Instances', there is a link to 'Instances New'. The main content area features a large orange 'Launch instance' button with a dropdown arrow. Below it is a smaller 'Migrate a server' button. A note states: 'To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.' Another note says: 'Note: Your instances will launch in the US East (N. Virginia) Region'. To the right of the main content, there are sections for 'Service health' (with a 'AWS Health Dashboard' link), 'Region' (set to 'US East (N. Virginia)'), 'Status' (showing 'This service is operating normally'), and 'Zones' (a table with columns for 'Zone name' and 'Zone ID'). At the bottom, there are links for 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

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

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. 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 note about AMIs and a search bar. The bottom of the screen includes standard browser navigation and AWS footer links.

Step 5: Select UBUNTU and 20.04 LTS Sever.

The screenshot shows the 'Quick Start' section of the AWS EC2 console. It displays various operating system options: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The 'Ubuntu' option is highlighted with a blue border. Below the options, details for the selected 'Ubuntu Server 20.04 LTS (HVM), SSD Volume Type' AMI are shown, including its AMI ID, virtualization type, ENA support, and root device type. A 'Browse more AMIs' link is also present.

Step 6: Create a Key Pair.

The screenshot shows the 'Create key pair' dialog box on the AWS EC2 console. The 'Key pair name' field contains 'ayesha-keypair'. The 'Key pair type' section has 'RSA' selected. The 'Private key file format' section has '.pem' selected. The 'Create key pair' button is highlighted with a red box.

Key pair (You can create keys for instances.)

Key pair name:

Key pair type:

- RSA RSA encrypted private and public key pair
- ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format:

- .pem For use with OpenSSH
- .ppk For use with PuTTY

Cancel **Create key pair**

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 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 two options: 'Create security group' (selected) and 'Select existing security group'. Below this, it states: 'We'll create a new security group called 'launch-wizard-5' with the following rules:' followed by three checked rules: 'Allow SSH traffic from Anywhere (0.0.0.0/0)', 'Allow HTTPs traffic from the internet', and 'Allow HTTP traffic from the internet'. The 'Allow HTTPs traffic from the internet' rule includes a note: 'To set up an endpoint, for example when creating a web server'.

The screenshot shows the AWS instance launch summary 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, there are 'Cancel' and 'Launch instance' buttons, with the 'Launch instance' button highlighted with a red box.

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

The screenshot shows the AWS EC2 Instances Launch an instance page. At the top, there's a success message: "Successfully initiated launch of instance (i-04f94f3196ce6c508)" with a green checkmark icon. Below it is a "Launch log" link. The main content area is titled "Next Steps". It includes a "Get notified of estimated charges" section with a link to "Create billing alerts". Another section, "How to connect to your instance", contains text about launching and running instances, monitoring status, and connecting via SSH or RDP. A "View more resources to get you started" link is also present. At the bottom right is an orange "View all instances" button.

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

Privacy Terms Cookie preferences

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

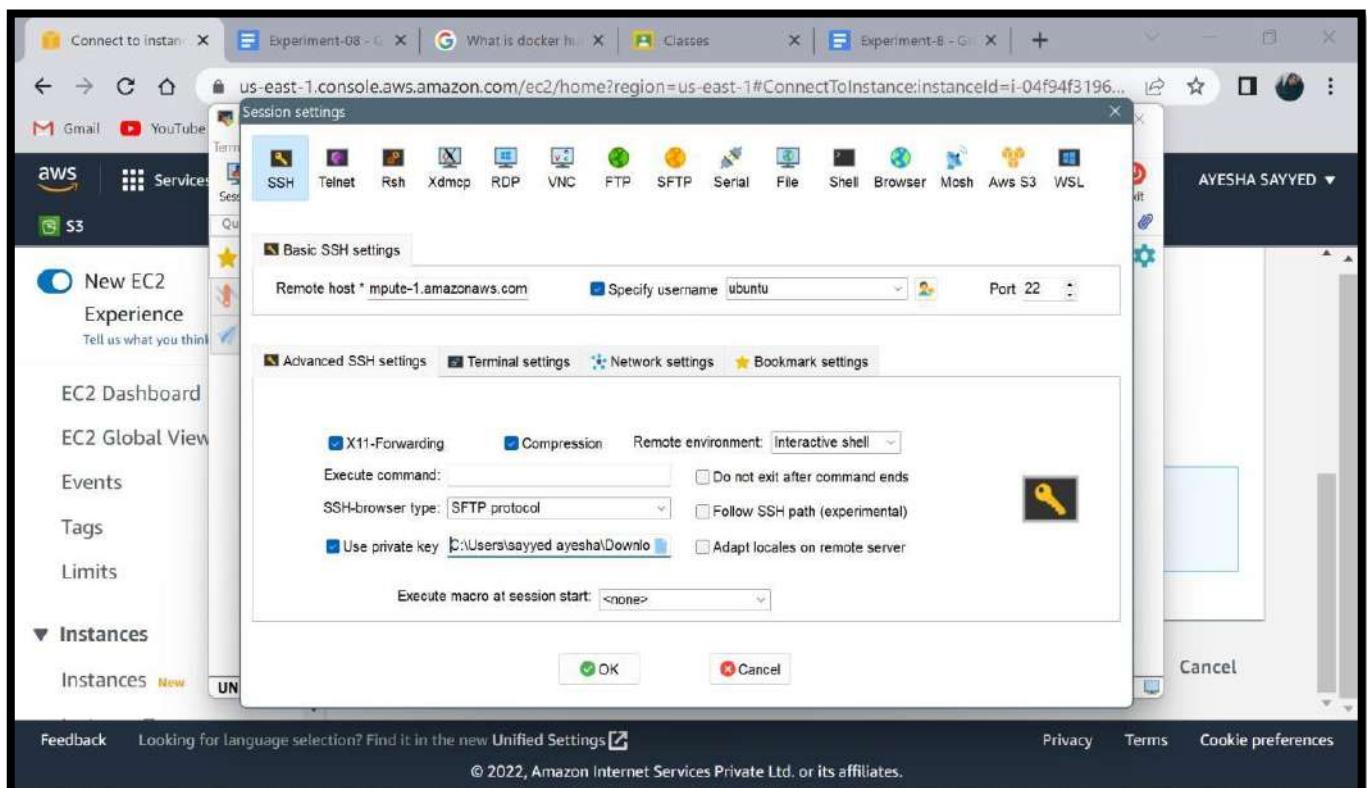
The screenshot shows the AWS EC2 Instances page. A modal window titled 'Instances (1/1) Info' is open over the main table. The table has columns: Name, Instance ID, Instance state, Instance type, and Status. One row is selected, showing 'ayesha_exp-08' as the Name, 'i-04f94f3196ce6c508' as the Instance ID, 'Running' as the Instance state, 't2.micro' as the Instance type, and '2/2' as the Status. The 'Connect' button in the modal is highlighted with a green box.

The screenshot shows the 'Connect to instance' details page for the instance 'i-04f94f3196ce6c508 (ayesha_exp-08)'. It lists three connection methods: EC2 Instance Connect, Session Manager, and SSH client (which is selected). Below these, the 'Instance ID' is listed as 'i-04f94f3196ce6c508 (ayesha_exp-08)'. A numbered list provides instructions for connecting via SSH:

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 again. A message box is displayed with the text: 'Public DNS copied' (with a green checkmark icon) and 'i-04f94f3196ce6c508 (ayesha_exp-08)'. Below this, it says '1. Open an SSH client.' and '2. Locate your private key file. The key used to launch this instance is ayesha-keypair.pem'. A note at the bottom of the message box states: '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 'Public DNS copied' message is also highlighted with a green box.

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]
```

The screenshot shows a terminal window titled "ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)". The user has run the command "#curl -fsSLhttps://get.docker.com -o get-docker.sh" and then "#sh get-docker.sh". The terminal output shows the Docker installation script running, including steps like "curl https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor --yes -o /etc/apt/keys/docker.gpg" and "apt-get update". A yellow oval highlights the command "#sh get-docker.sh", and another yellow oval highlights the line "get-docker.sh: command not found".

```
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: 4f282167c425347a931ccfd95cc91fab041d414f  
+ 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/keys/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/ubuntu 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>

The screenshot shows a terminal window titled "ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)". The user has run the command "docker info". The terminal output displays various Docker components and their versions, such as the Docker Engine (Community) version 20.10.18, Go version go1.18.6, and Git commit b40c2f6. It also shows information for containerd, runc, and docker compose.

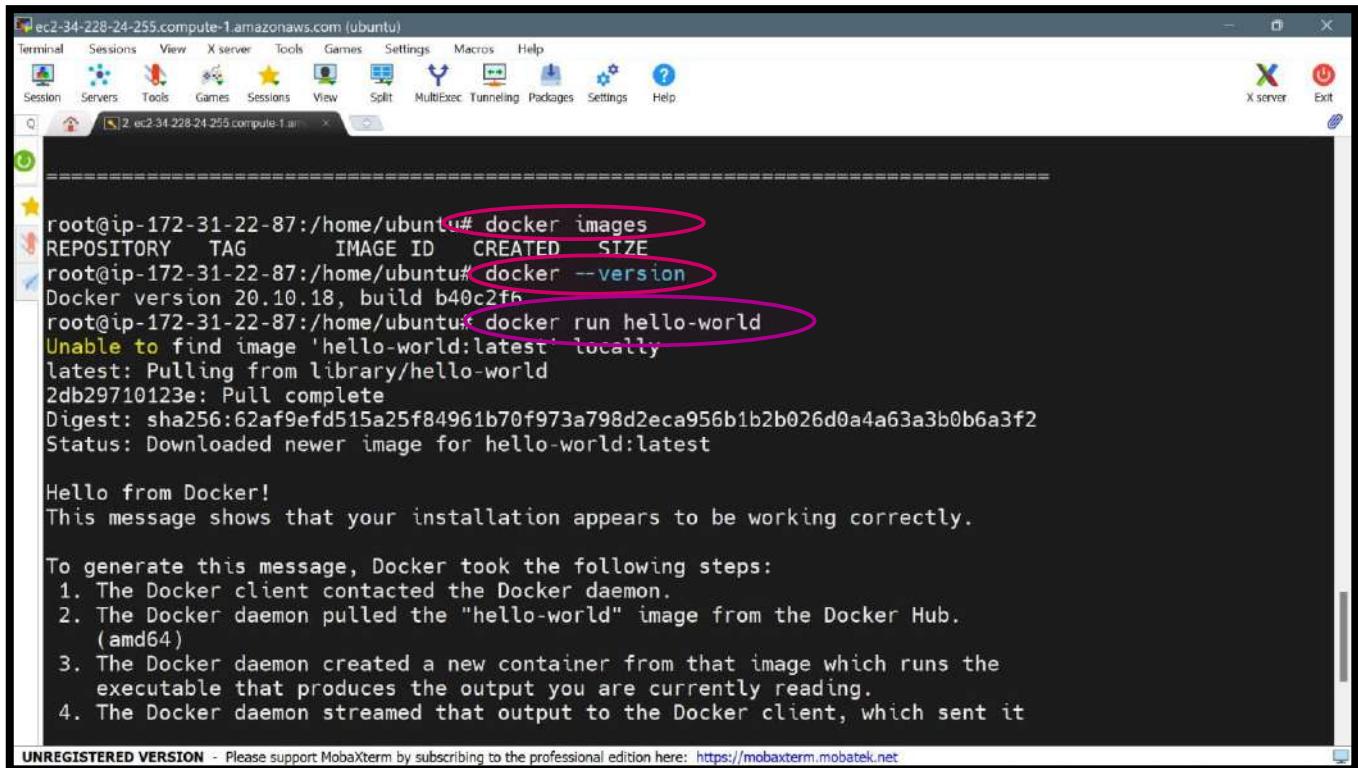
```
Version: 20.10.18  
API version: 1.41  
Go version: go1.18.6  
Git commit: b40c2f6  
Built: Thu Sep 8 23:11:45 2022  
OS/Arch: linux/amd64  
Context: default  
Experimental: true  
  
Server: Docker Engine - Community  
Engine:  
  Version: 20.10.18  
  API version: 1.41 (minimum version 1.12)  
  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 content is as follows:

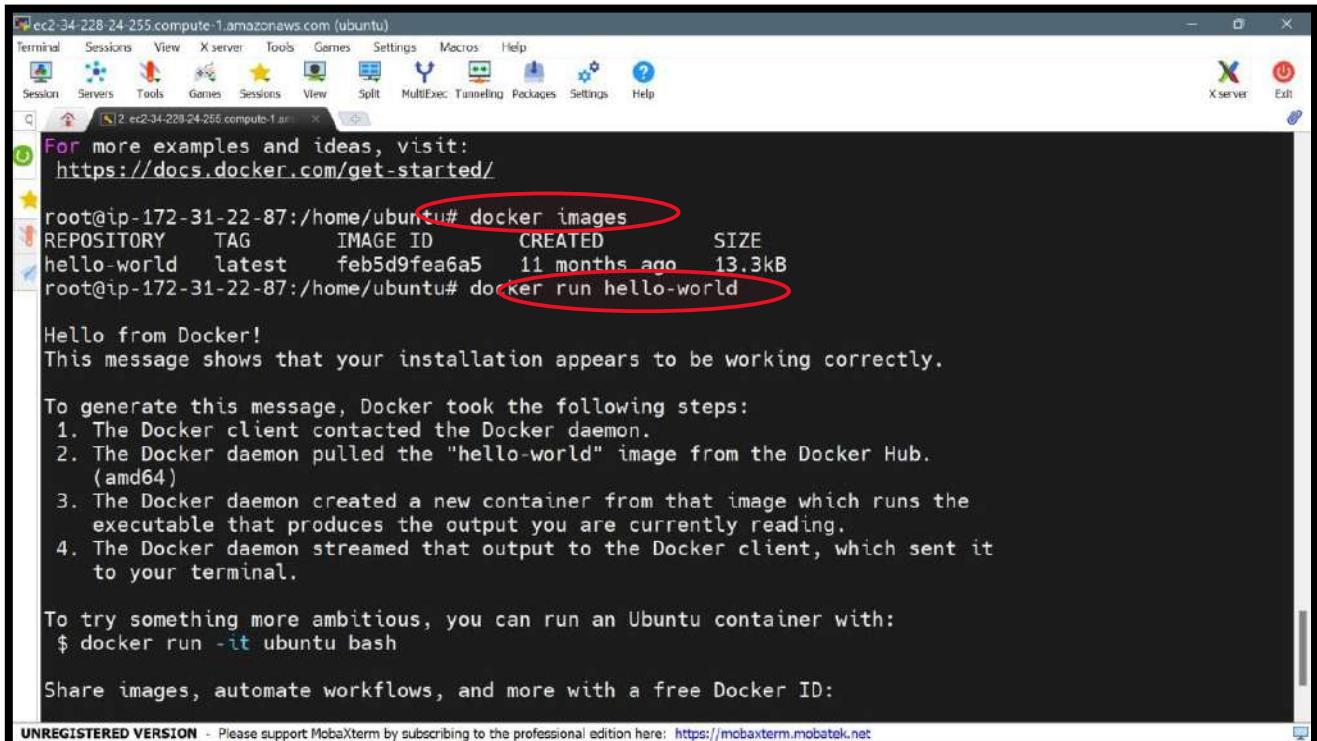
```
root@ip-172-31-22-87:/home/ubuntu# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@ip-172-31-22-87:/home/ubuntu# docker --version
Docker version 20.10.18, build b40c2f6
root@ip-172-31-22-87:/home/ubuntu# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling 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 content 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.

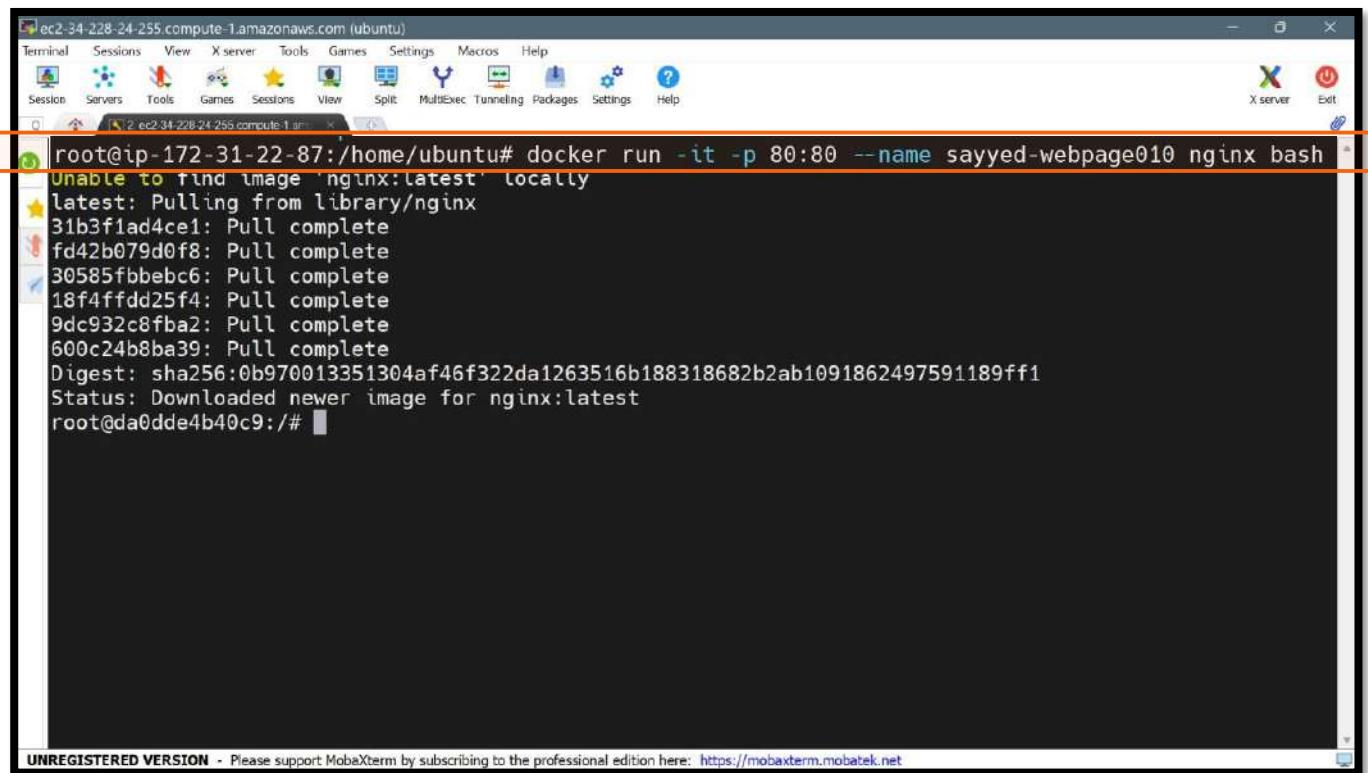
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the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

To 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>

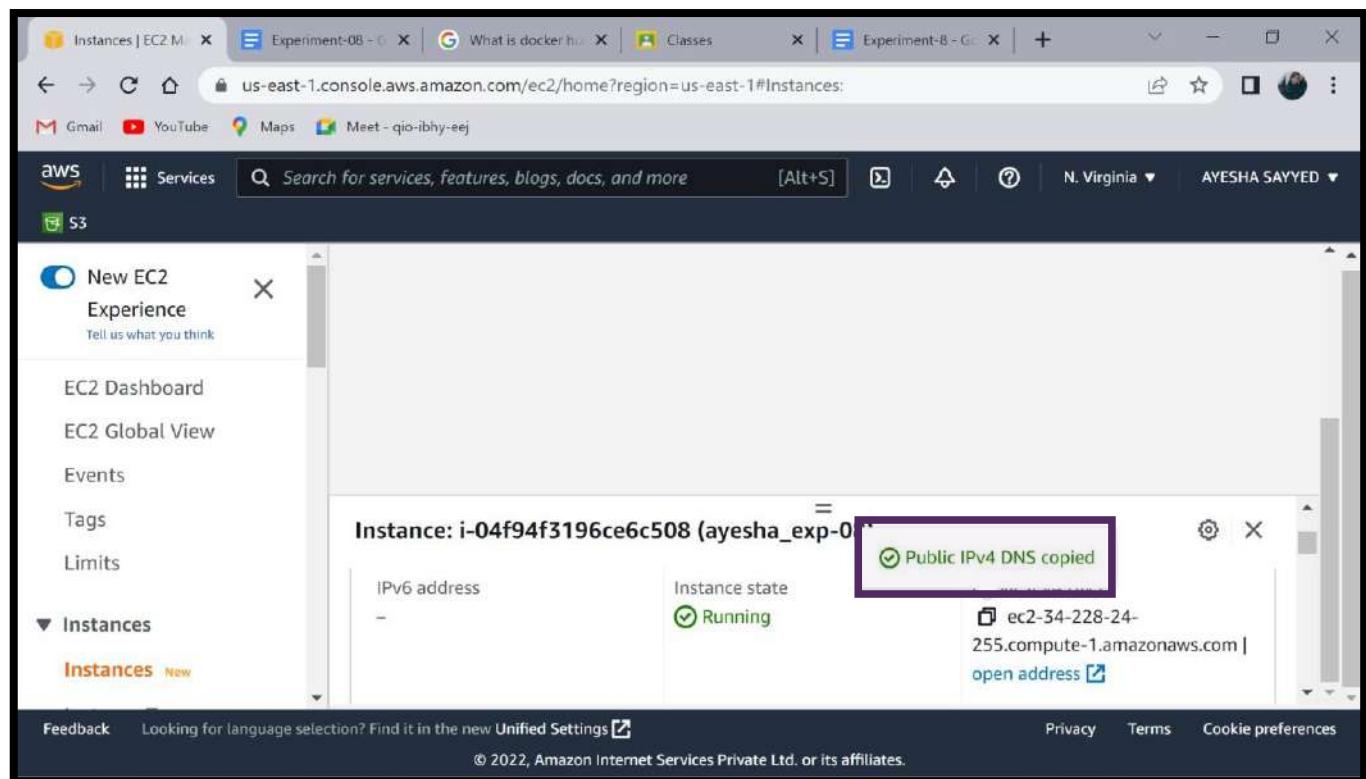
Step 14: Run command ‘docker run -it -p 80:80 –name[name of your webpage] nginx bash’. Now to launch the nginx web server, copy the Ipv4 address from the EC2 instance details and paste it to the address bar of Web Browser.



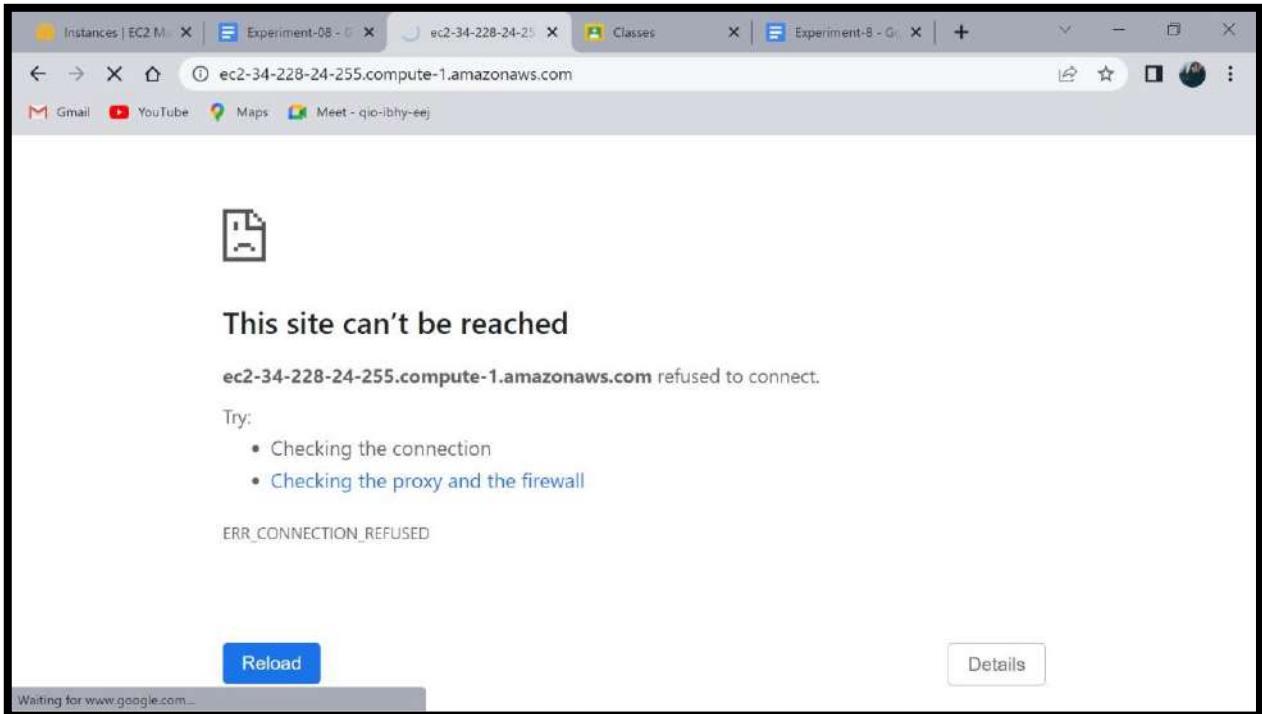
A screenshot of the MobaXterm application window titled "ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)". The terminal session shows the command "root@ip-172-31-22-87:/home/ubuntu# docker run -it -p 80:80 --name sayyed-webpage010 nginx bash" being run. The output indicates that the 'nginx:latest' image is being pulled from the library. A red box highlights the command and its output.

```
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:/#
```

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



Can't able to Reach the site.



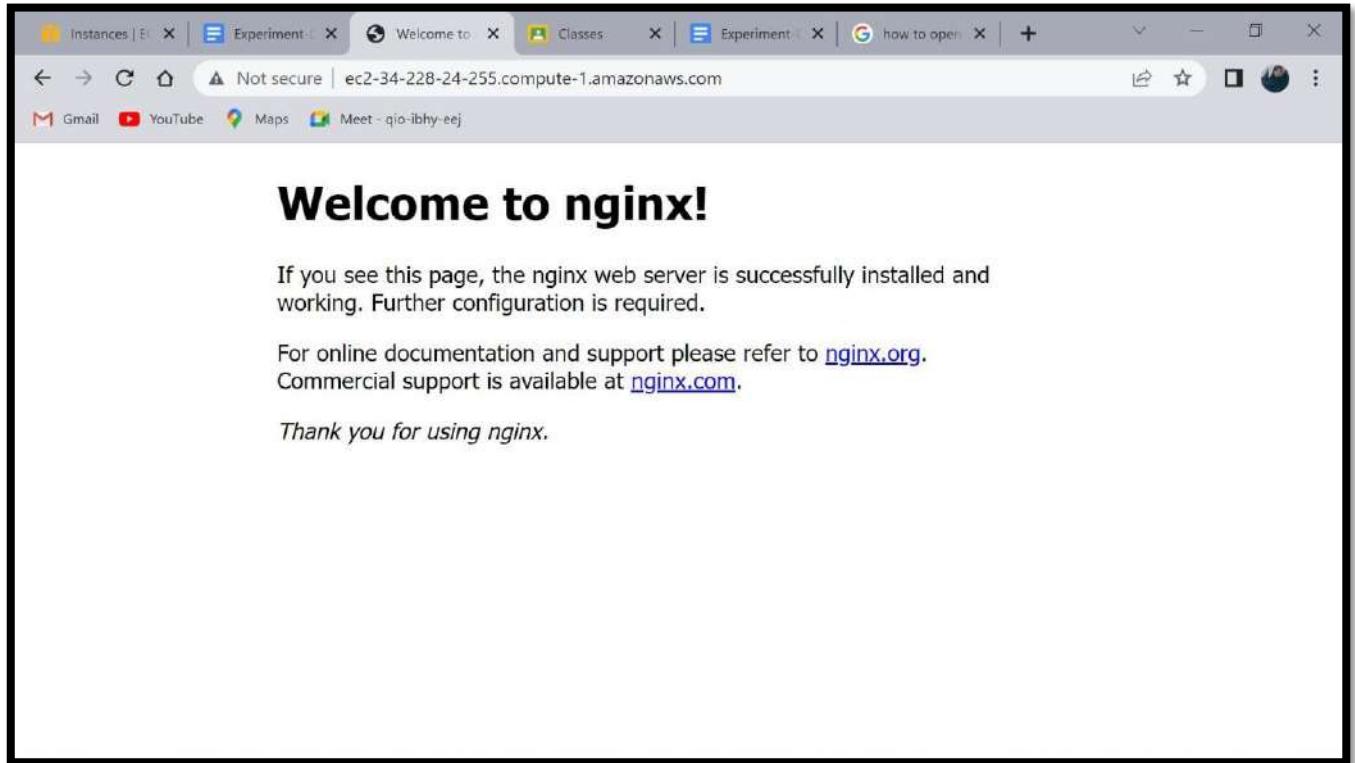
Step 15: Within the container use command ‘service nginx start’ to deploy the web server.

```
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>

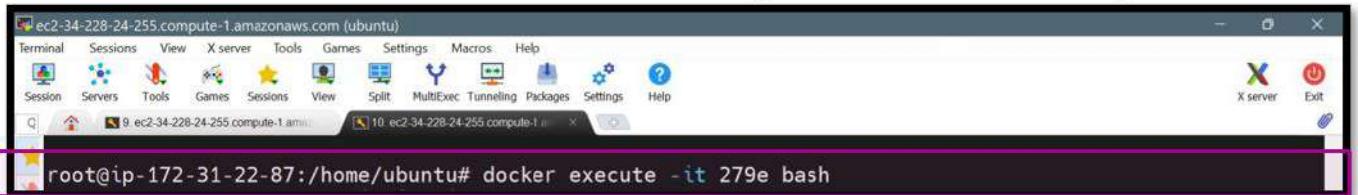
After deploying the web server, the web page will be visible without any errors.



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

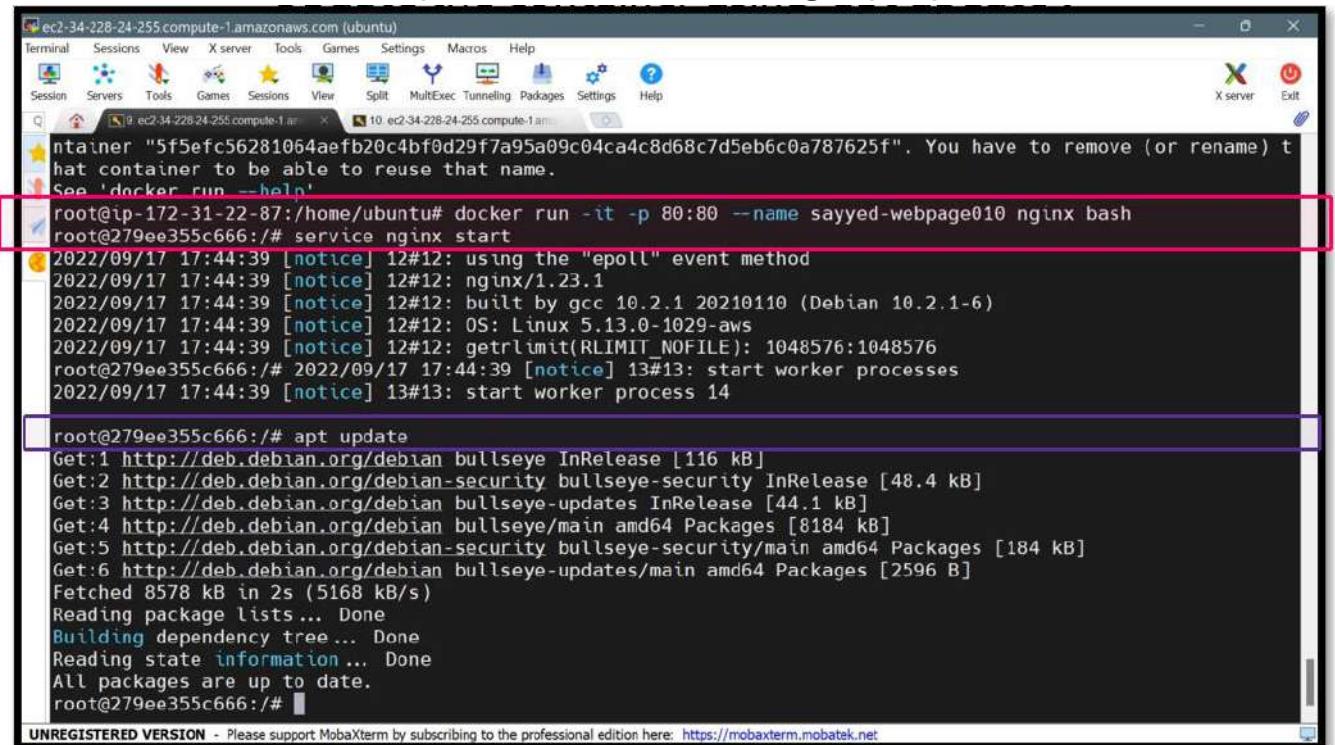
A screenshot of a terminal window titled 'ec2-34-228-24-255.compute-1.amazonaws.com (ubuntu)'. The terminal is running as root. A pink box highlights the command 'docker execute -it 279e bash'. The output shows an error: 'unknown shorthand flag: 'i' in -it'. It then displays the usage information for the docker command, which includes options like --config, --context, -D, -H, -l, --log-level, --tls, --tlscacert, --tlscert, --tlskey, --tlsverify, and -v, --version. The terminal interface includes a menu bar with 'Terminal', 'Sessions', 'View', 'X server', 'Tools', 'Games', 'Settings', 'Macros', 'Help', and icons for 'Session', 'Servers', 'Tools', 'Games', 'Sessions', 'View', 'Split', 'MultiExec', 'Tunnelling', 'Packages', 'Settings', and 'Help'. There are also 'X server' and 'Exit' buttons in the top right.

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 exec -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:#
```

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:/usr/share/nginx/html# 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.

```
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#
```

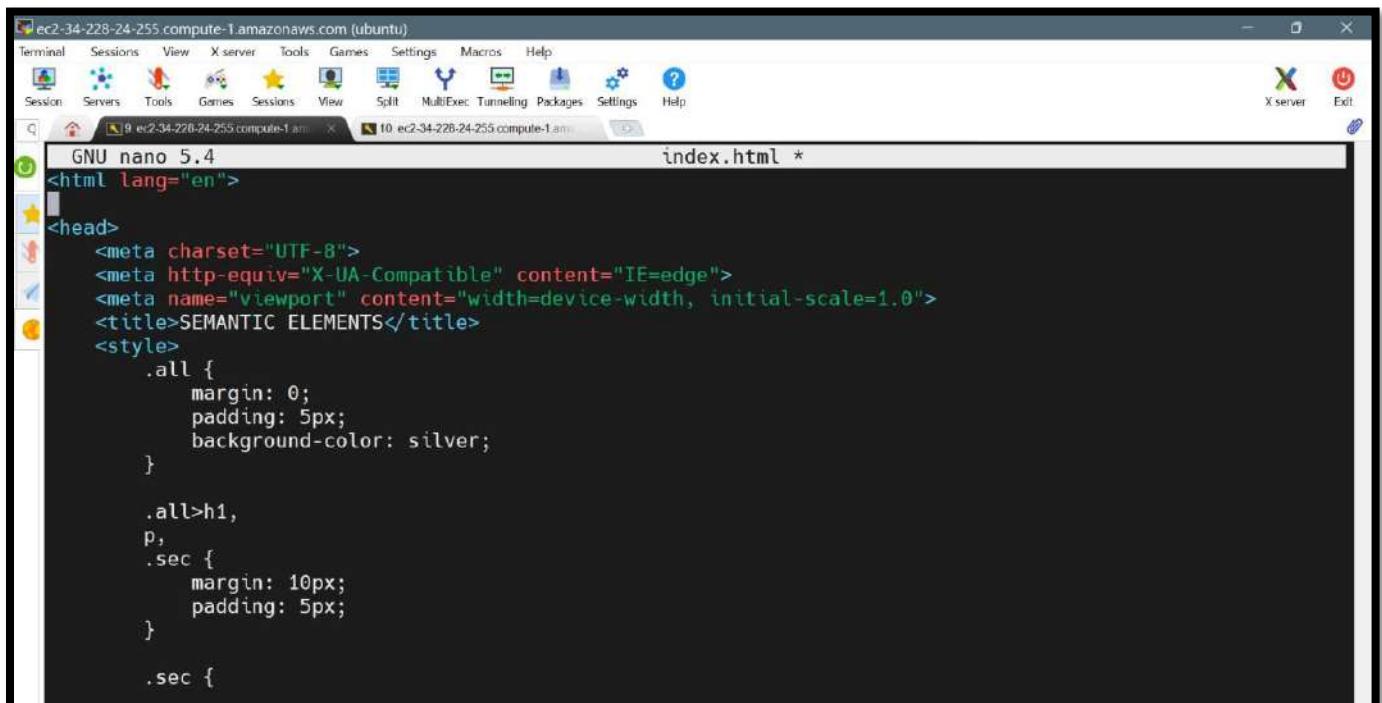
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

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#
```

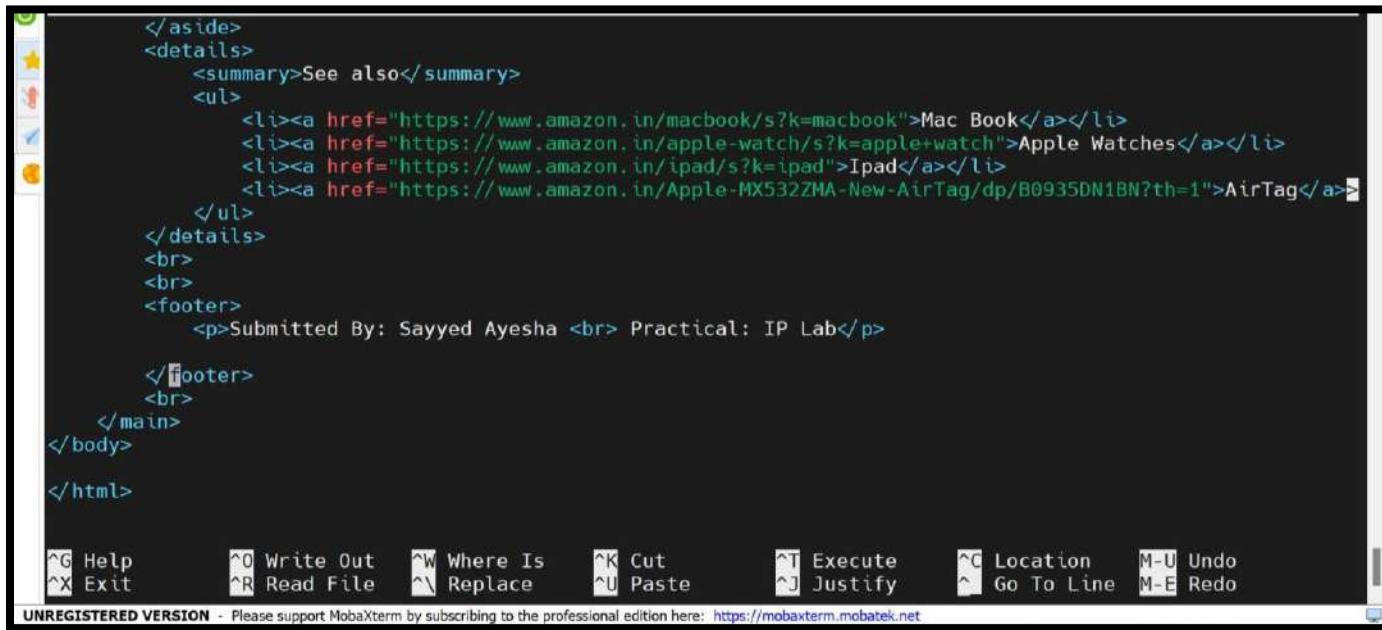
Step 20: Write the HTML code then → press ‘Ctrl+O → ENTER → ‘Ctrl+X’. This will help you to save the code and exit.



```
<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 {
```



```
</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>
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^I Go To Line M-E 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 ??

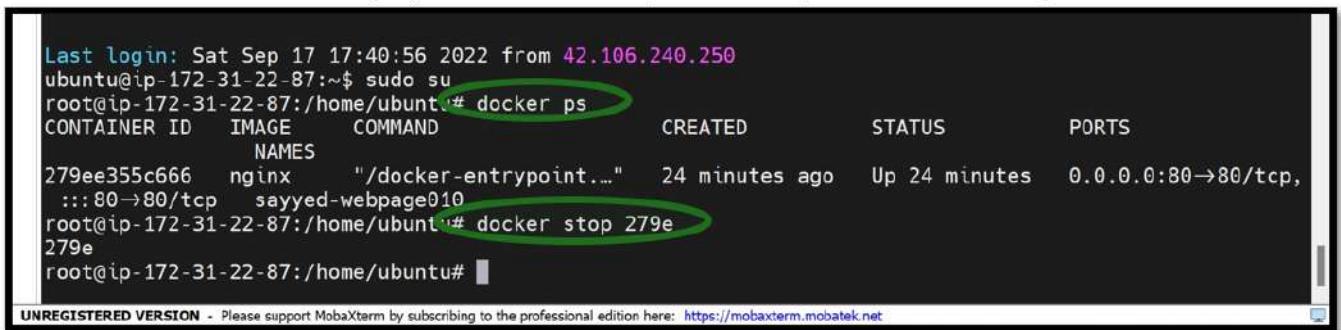
On January 9, 2007, Steve Jobs announced iPhone at the Macworld convention and received huge media attention. Finally later that year, on June 29, 2007 the first iPhone was released. But do you know that the present iPhone was earlier named as "Purple"? But this name never came into existence as the 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 Drift".

See also

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

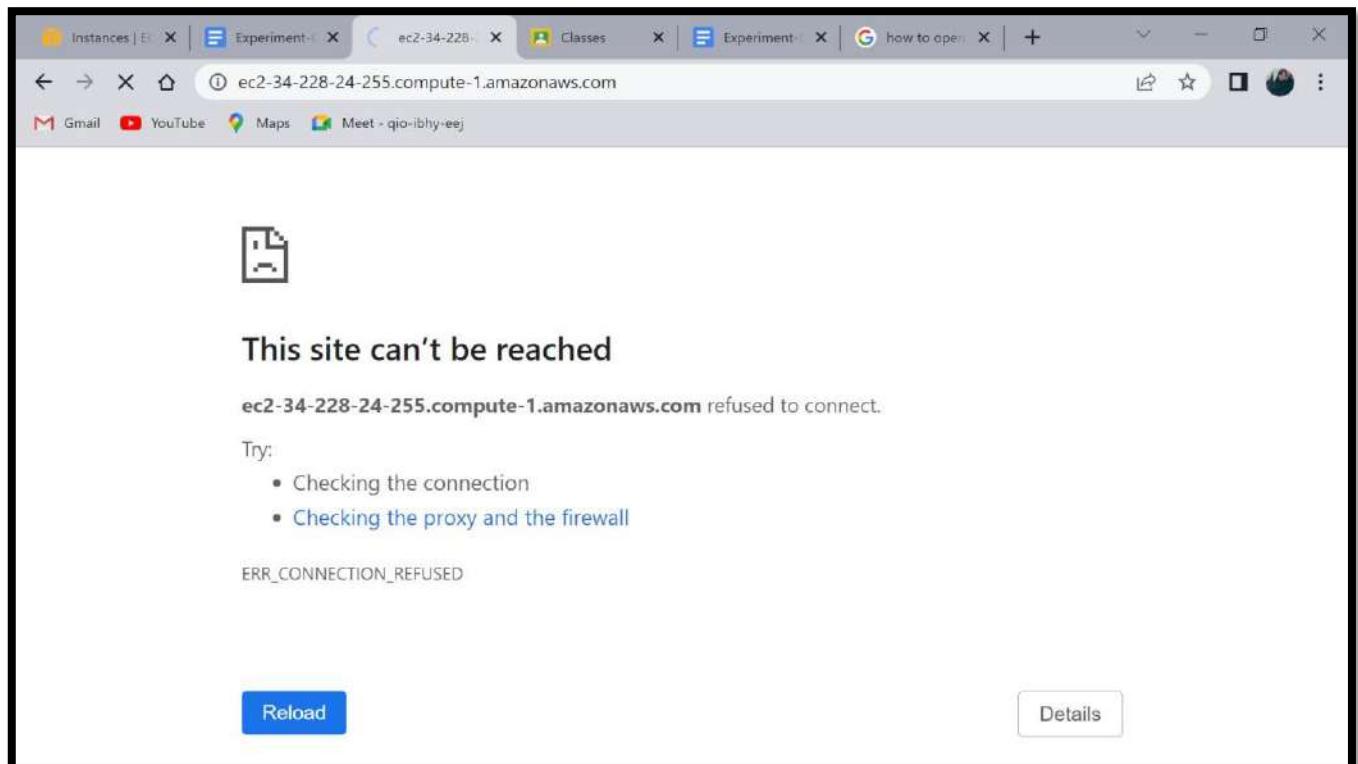
Submitted By: Sayyed Ayesha
Practical: IP Lab

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



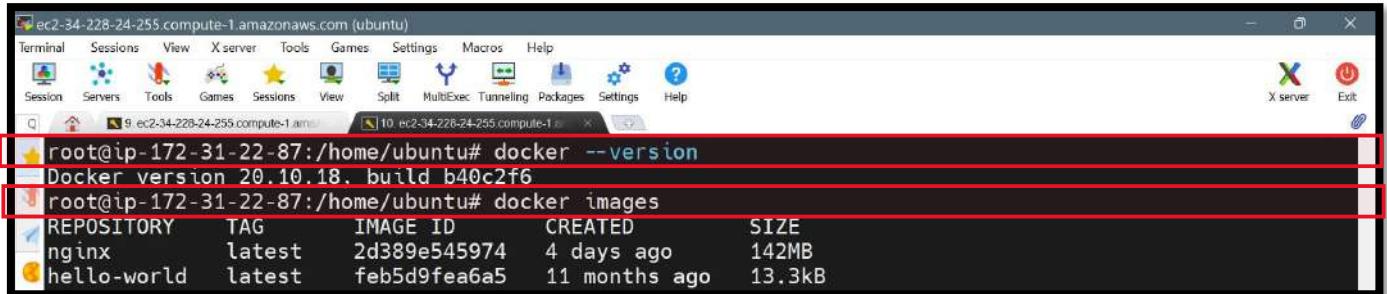
```
Last login: Sat Sep 17 17:40:56 2022 from 42.106.240.250
ubuntu@ip-172-31-22-87:~$ sudo su
root@ip-172-31-22-87:/home/ubuntu# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
279ee355c666        nginx              "/docker-entrypoint..."   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>



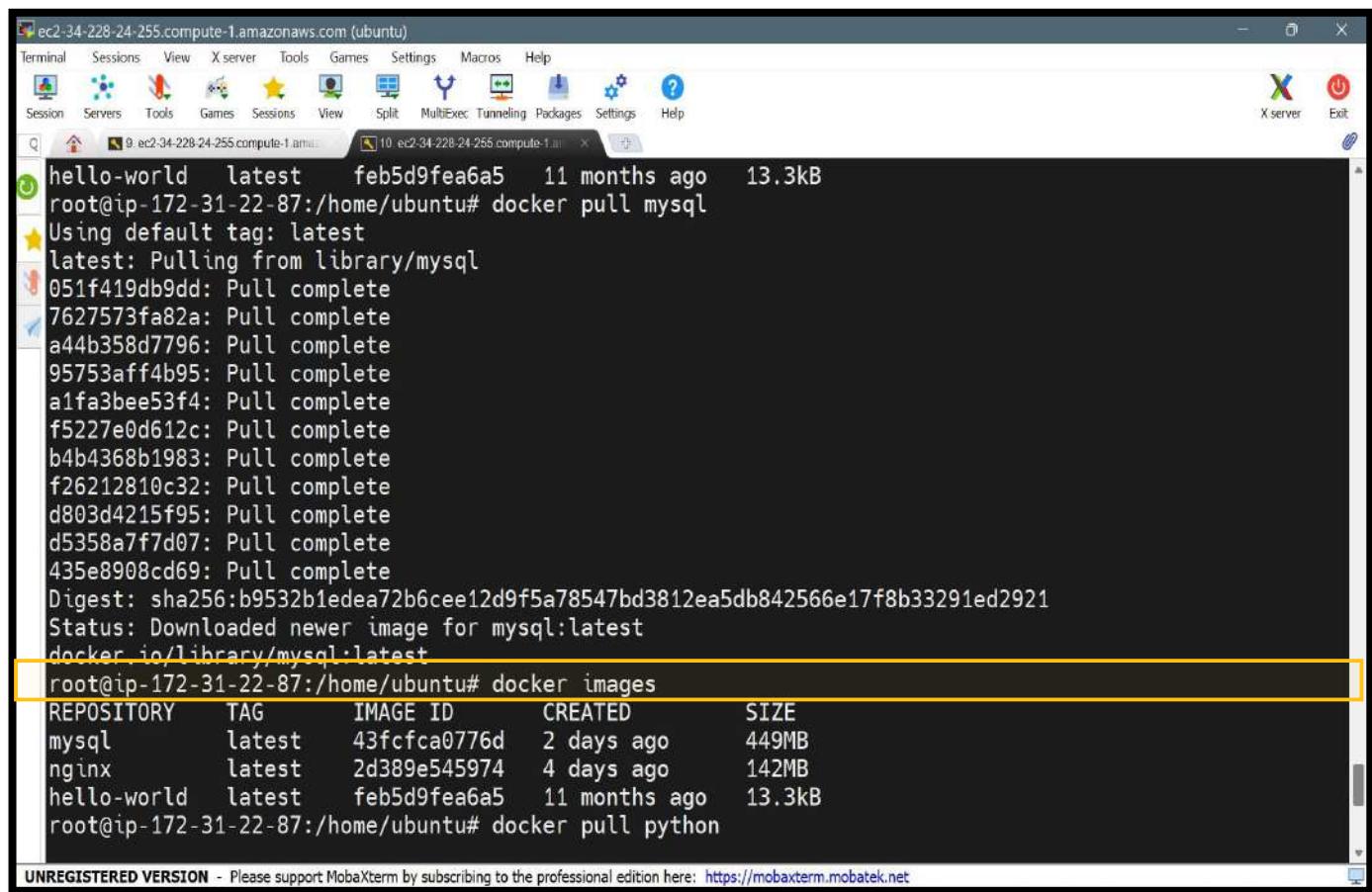
Q5. Pull 3 or 4 images, one of the python, run “Hello World ” inside container.

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.



```
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    43fcfcfa0776d   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 in MobaXterm. The user has pulled the Python image from Docker Hub:

```
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
```

Then, the user lists all Docker images:

```
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
```

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 in MobaXterm. The user runs the Docker container:

```
root@ip-172-31-22-87:/home/ubuntu# docker run -it e285 bash
```

They then enter the Python shell:

```
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
>>>
```

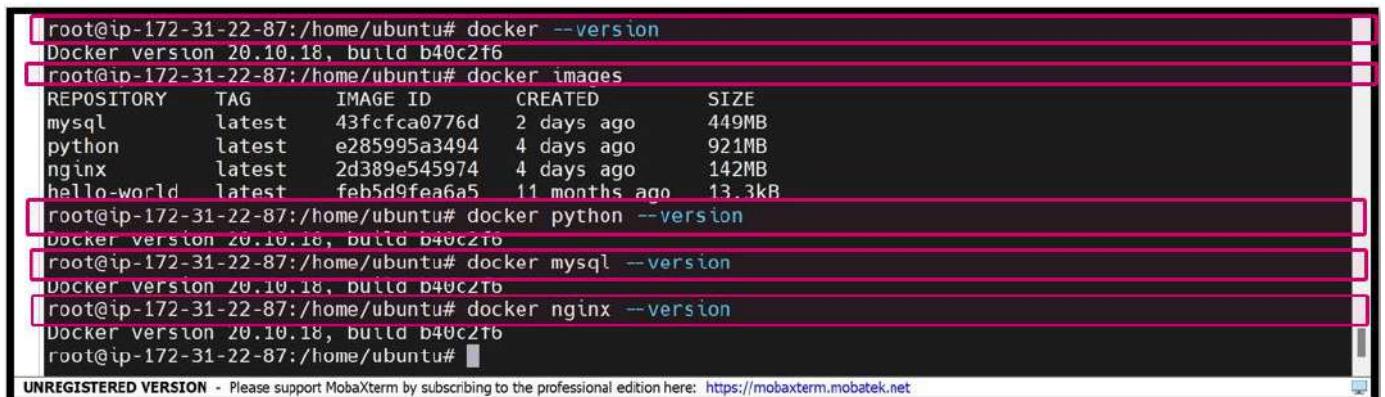
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

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.

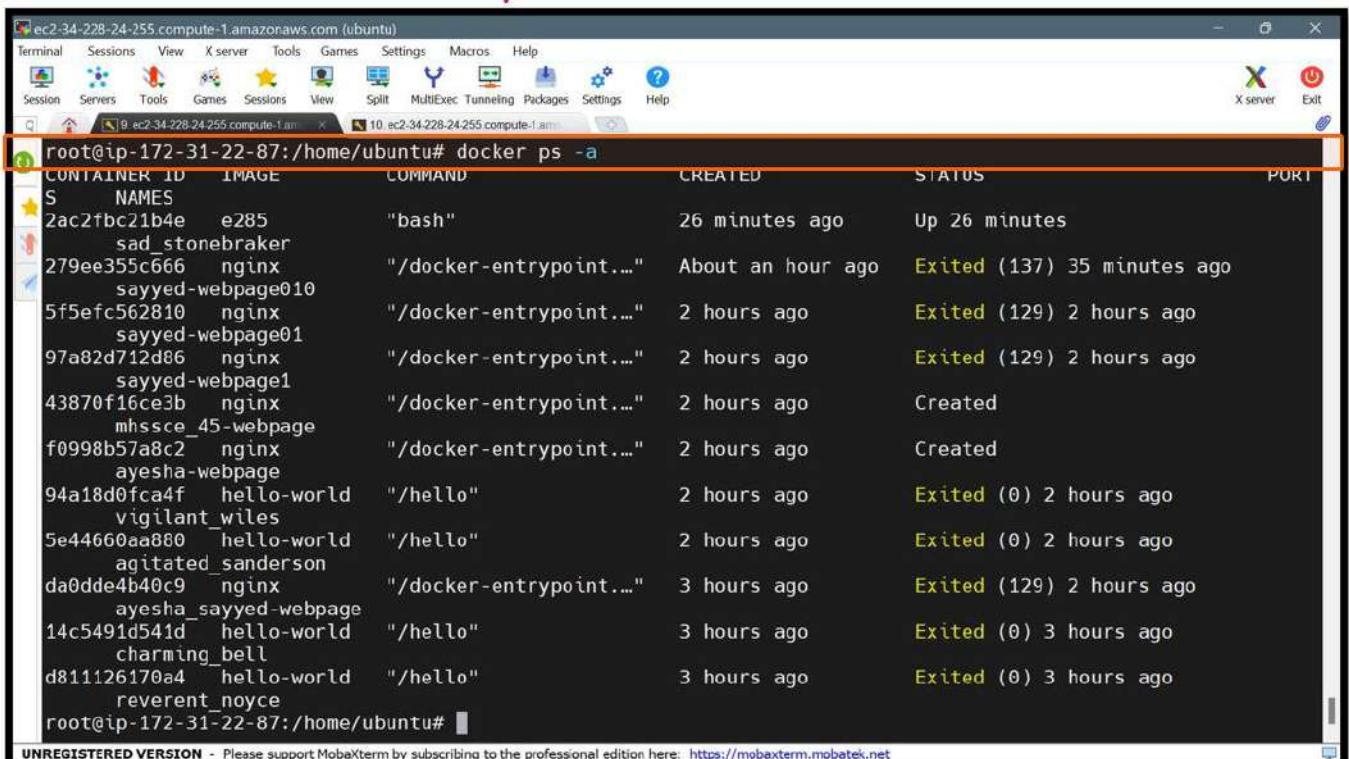
And then enter ‘docker nginx --version’ to see the current nginx version.



```
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    43fcfcfa0776d   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>

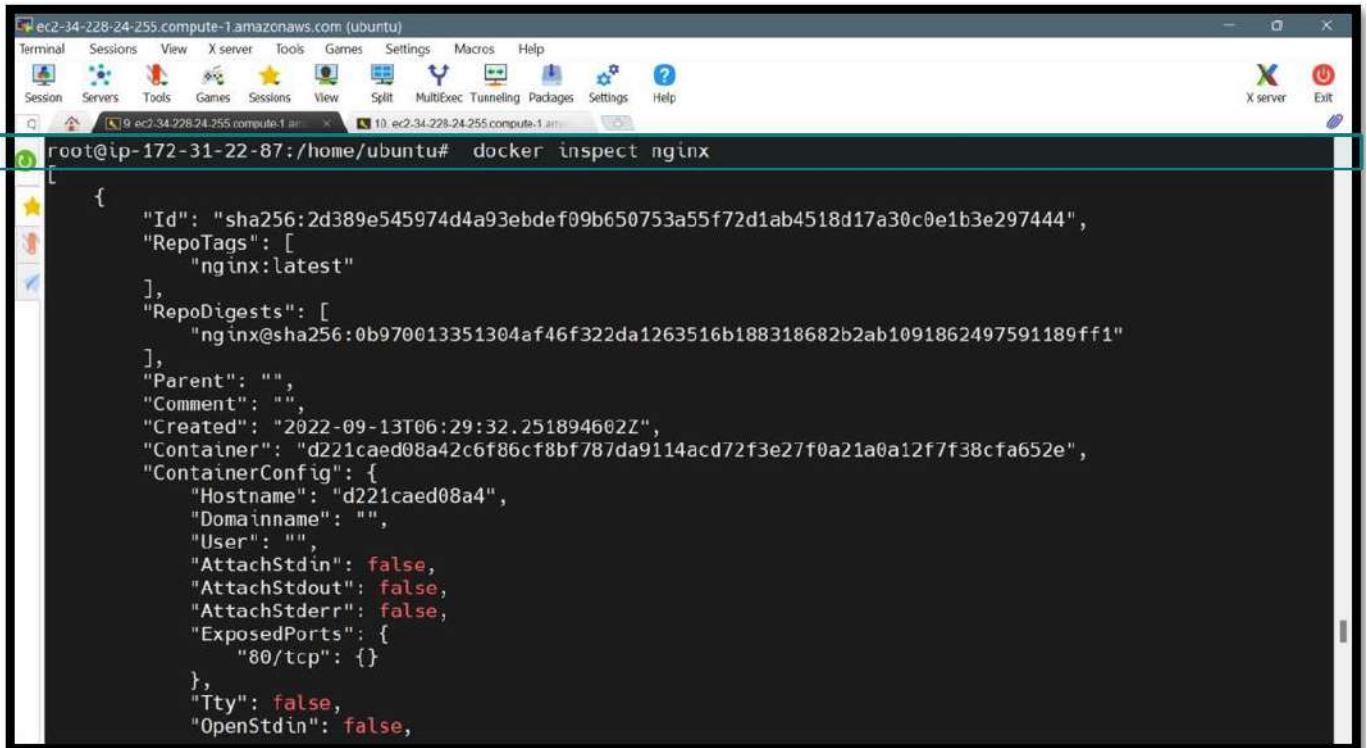
For checking the status of the container use command ‘docker ps -a’.



```
root@ip-172-31-22-87:/home/ubuntu# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
S NAMES
2ac2fbc21b4e      e285                "bash"              26 minutes ago   Up 26 minutes
sad_stonebraker    sad_stonebraker
279ee355c666      nginx              "/docker-entrypoint..."  About an hour ago  Exited (137) 35 minutes ago
sayyed-webpage010  nginx              "/docker-entrypoint..."  2 hours ago       Exited (129) 2 hours ago
5f5efc562810      nginx              "/docker-entrypoint..."  2 hours ago       Exited (129) 2 hours ago
97a82d712d86      nginx              "/docker-entrypoint..."  2 hours ago       Exited (129) 2 hours ago
sayyed-webpage01   nginx              "/docker-entrypoint..."  2 hours ago       Created
43870f16ce3b      nginx              "/docker-entrypoint..."  2 hours ago       Created
mhssce_45-webpage nginx              "/docker-entrypoint..."  2 hours ago       Created
f0998b57a8c2      nginx              "/docker-entrypoint..."  2 hours ago       Created
ayesha-webpage    ayesha-webpage
94a18d0fcfa4f     hello-world        "/hello"            2 hours ago       Exited (0) 2 hours ago
vigilant_wiles    hello-world        "/hello"            2 hours ago       Exited (0) 2 hours ago
5e44660aa880      hello-world        "/hello"            2 hours ago       Exited (0) 2 hours ago
agitated_sanderson nginx              "/docker-entrypoint..."  3 hours ago       Exited (129) 2 hours ago
da0dde4b40c9      nginx              "/docker-entrypoint..."  3 hours ago       Exited (129) 2 hours ago
ayesha_sayyed-webpage  nginx              "/docker-entrypoint..."  3 hours ago       Exited (0) 3 hours ago
14c5491d541d      hello-world        "/hello"            3 hours ago       Exited (0) 3 hours ago
charming_bell     hello-world        "/hello"            3 hours ago       Exited (0) 3 hours ago
d811126170a4      hello-world        "/hello"            3 hours ago       Exited (0) 3 hours ago
reverent_noxye    reverent_noxye
root@ip-172-31-22-87:/home/ubuntu#
```

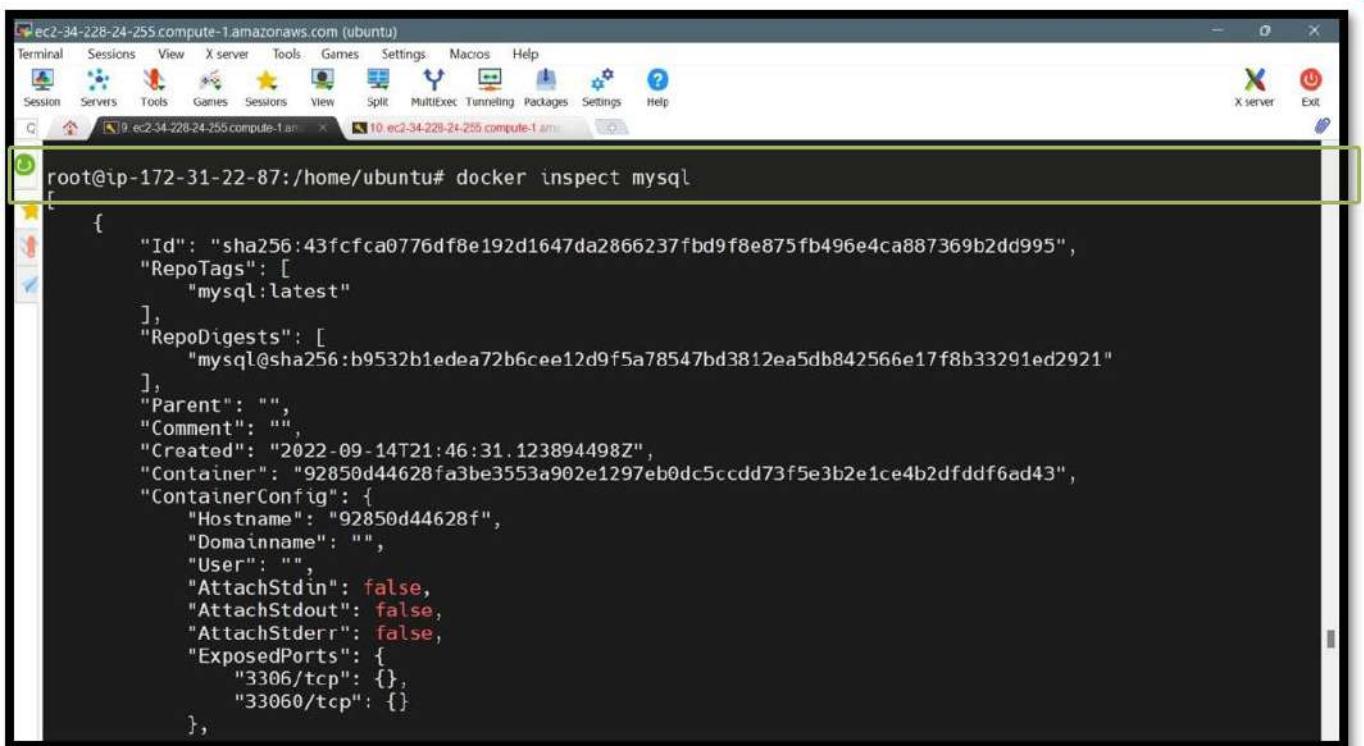
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

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



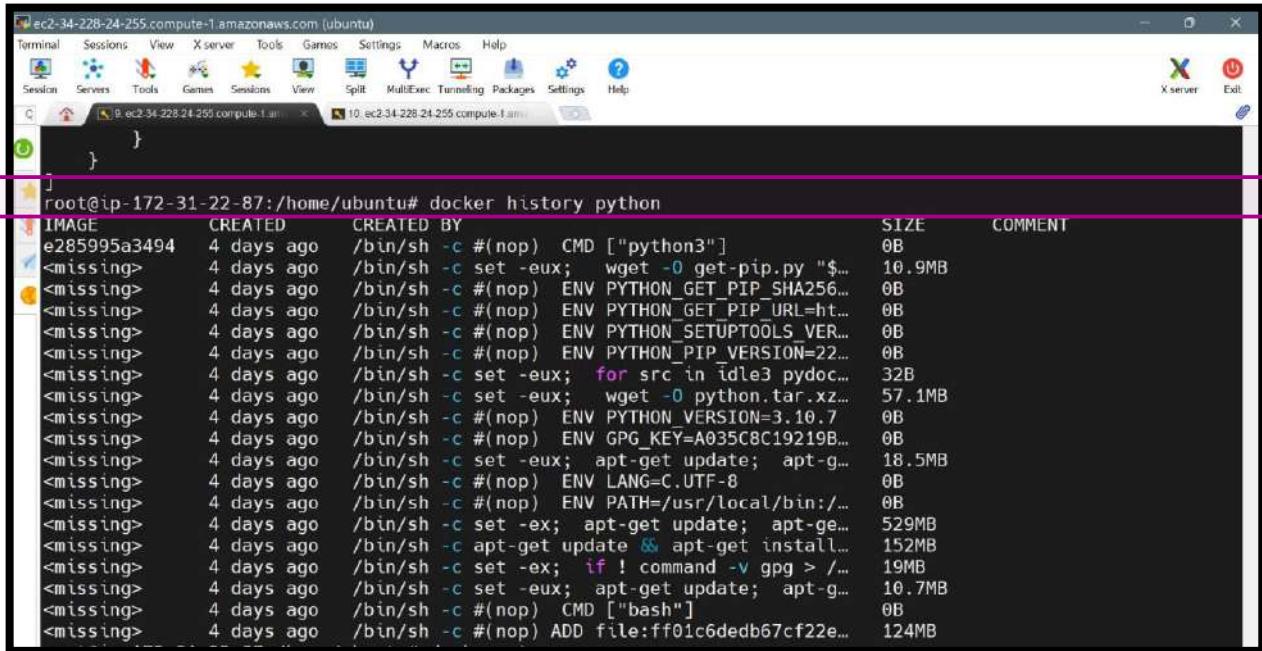
```
root@ip-172-31-22-87:/home/ubuntu# docker inspect nginx
[{"Id": "sha256:2d389e545974d4a93ebdef09b650753a55f72d1ab4518d17a30c0e1b3e297444",
 "RepoTags": [
     "nginx:latest"
 ],
 "RepoDigests": [
     "nginx@sha256:0b970013351304af46f322da1263516b188318682b2ab1091862497591189ff1"
 ],
 "Parent": "",
 "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.



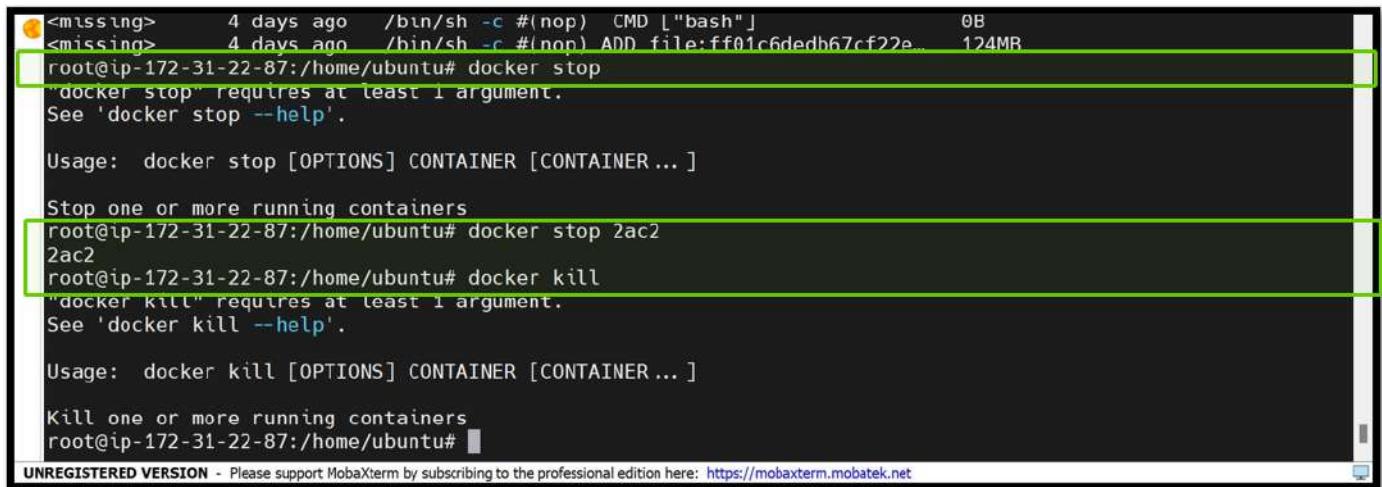
```
root@ip-172-31-22-87:/home/ubuntu# docker inspect mysql
[{"Id": "sha256:43fcfcfa0776df8e192d1647da2866237fb9f8e875fb496e4ca887369b2dd995",
 "RepoTags": [
     "mysql:latest"
 ],
 "RepoDigests": [
     "mysql@sha256:b9532b1ede472b6ce12d9f5a78547bd3812ea5db842566e17f8b33291ed2921"
 ],
 "Parent": "",
 "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...  124MB
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#
```

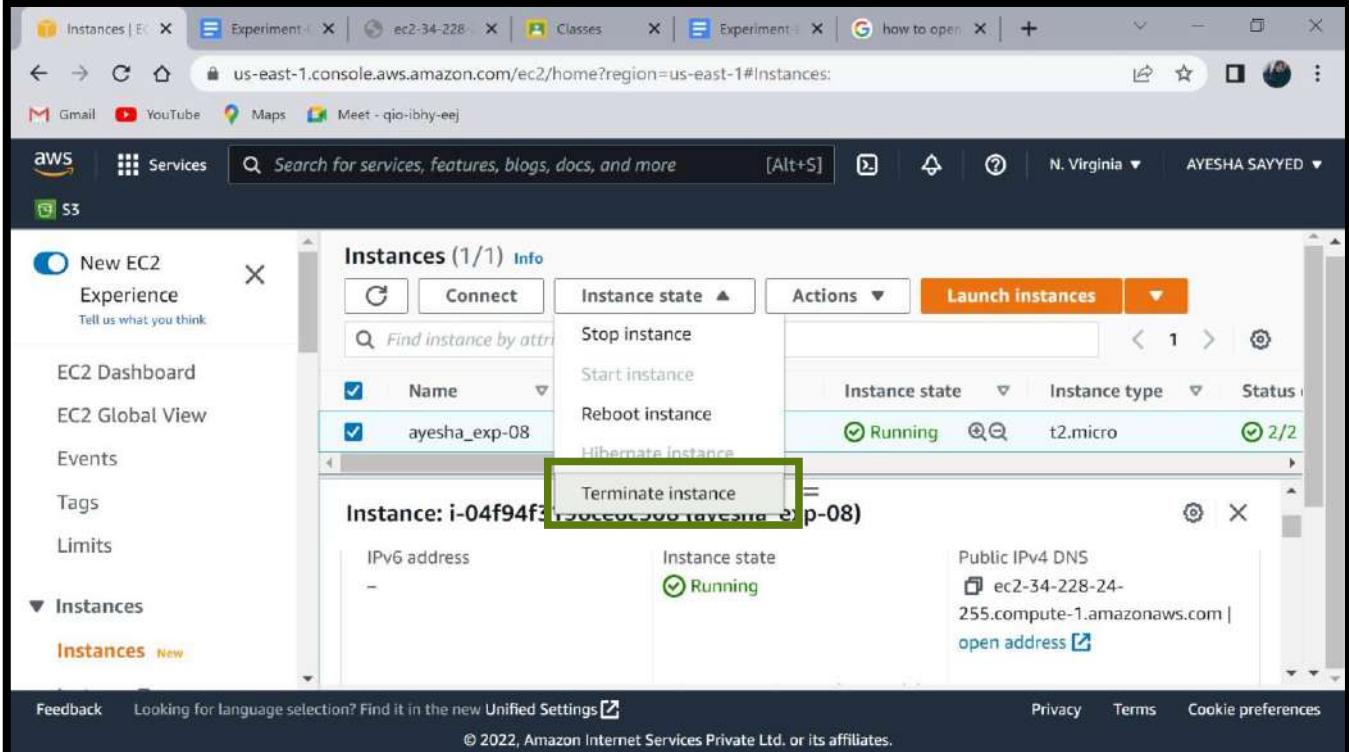
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Now exit the Session / Terminal.

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

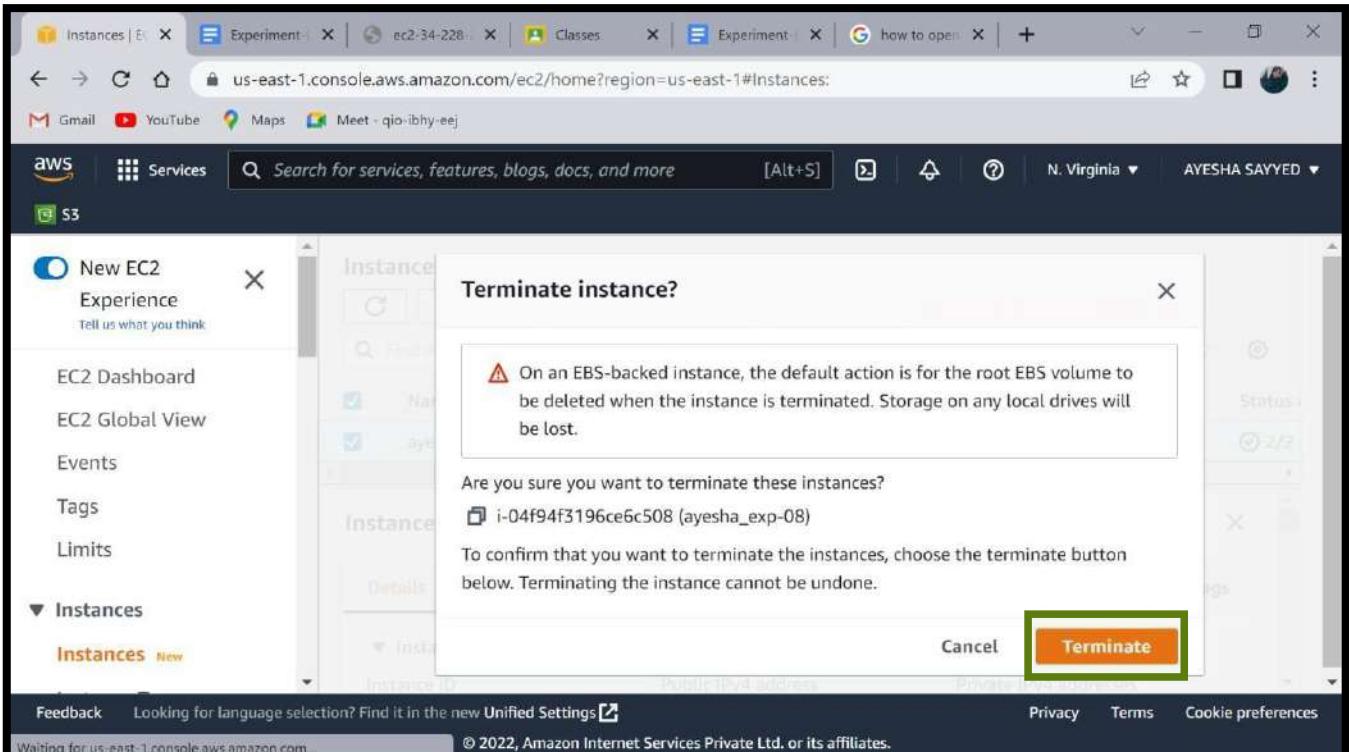
- We have demonstrated 15 docker commands and also explained it, but randomly [in steps]not one by one.

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, there's a link to Instances. The main area displays 'Instances (1/1) Info'. It lists one instance: 'ayesha_exp-08'. The instance details show it's 'Running' with a Public IPv4 DNS of 'ec2-34-228-24-255.compute-1.amazonaws.com'. A context menu is open over the instance, with the 'Terminate instance' option highlighted. The status bar at the bottom indicates 'Waiting for us-east-1.console.aws.amazon.com...'.

Click on Terminate to ‘Terminate the Instance’.



The screenshot shows a 'Terminate instance?' confirmation 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 list of instances: '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.' There are 'Cancel' and 'Terminate' buttons, with 'Terminate' being highlighted. The background shows the same EC2 Instances page as the previous screenshot.

Instances Successfully Terminated!!

The screenshot shows a browser window with multiple tabs open, including 'Instances', 'Experiment', 'ec2-34-228', 'Classes', 'Experiment', and 'Google how to open'. The main content is the AWS EC2 Instances page. A green notification bar at the top right says 'Successfully terminated i-04f94f3196ce6c508'. The main table shows one instance: 'Instance: i-04f94f3196ce6c508 (ayesha_exp-08)'. The 'Details' tab is selected. At the bottom, there's a footer with 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.

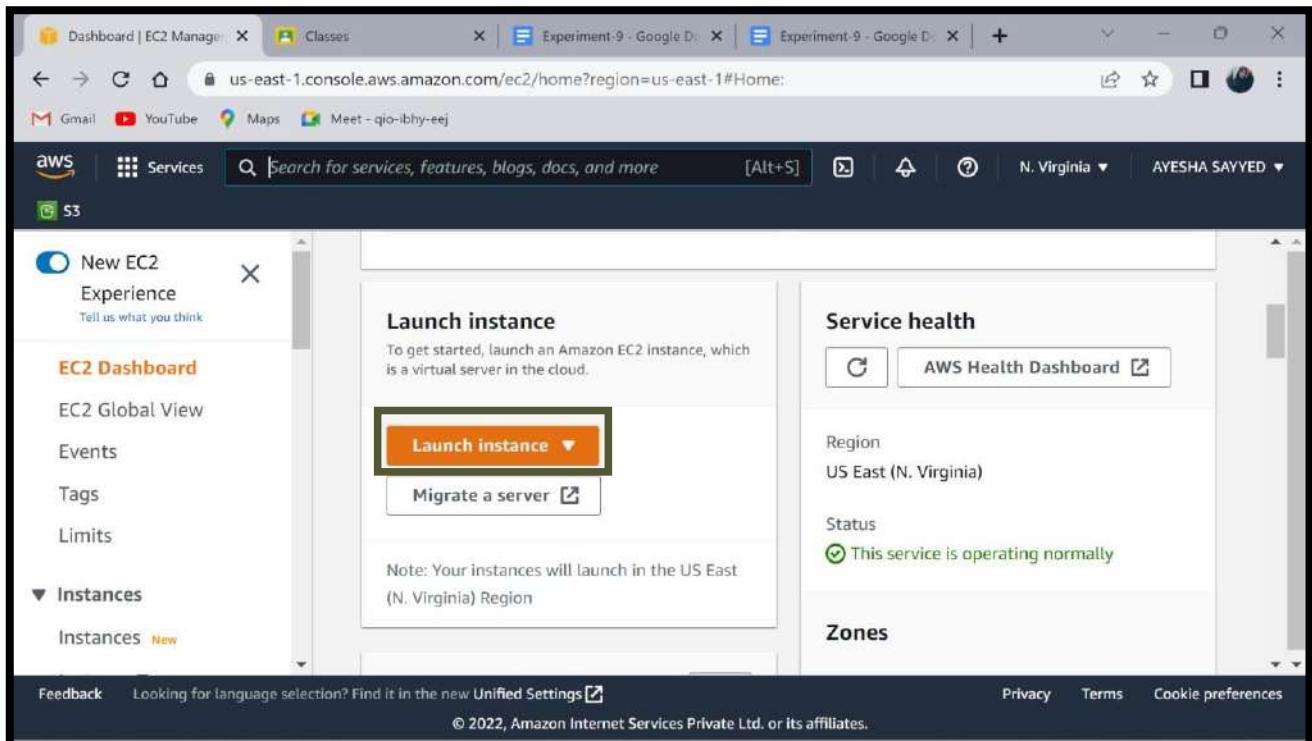
Step1: login to AWS Management Console Dashboard.

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

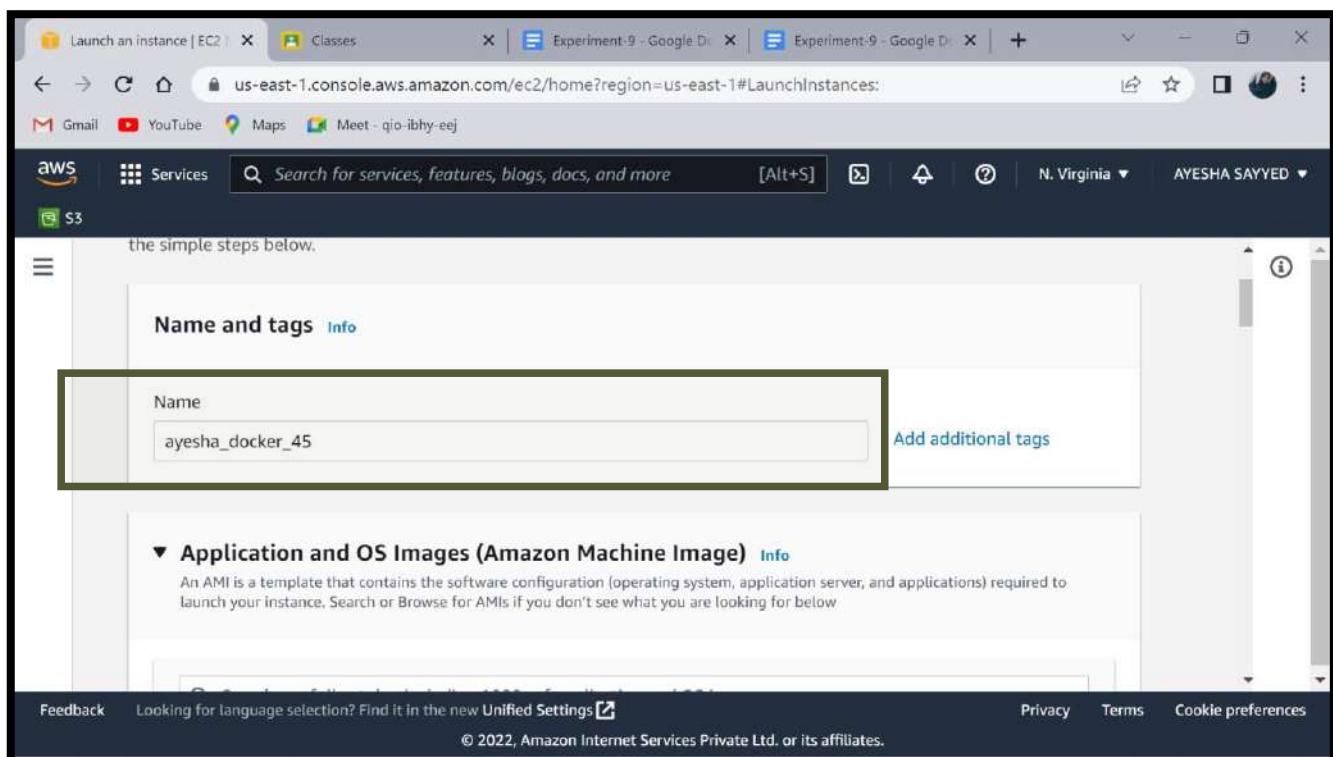
Step 2: Search for EC2 and select it.

The screenshot shows the AWS Management Console search results for "EC2". The search bar at the top contains the query "EC2". In the main search results area, the "EC2" service is highlighted with a red box. The description "Virtual Servers in the Cloud" is visible below the service name. Other search results include "EC2 Image Builder" and "AWS Compute Optimizer". On the left side, there is a sidebar with categories like "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, there are links for "Feedback", "Unified Settings", "Privacy", "Terms", and "Cookie preferences".

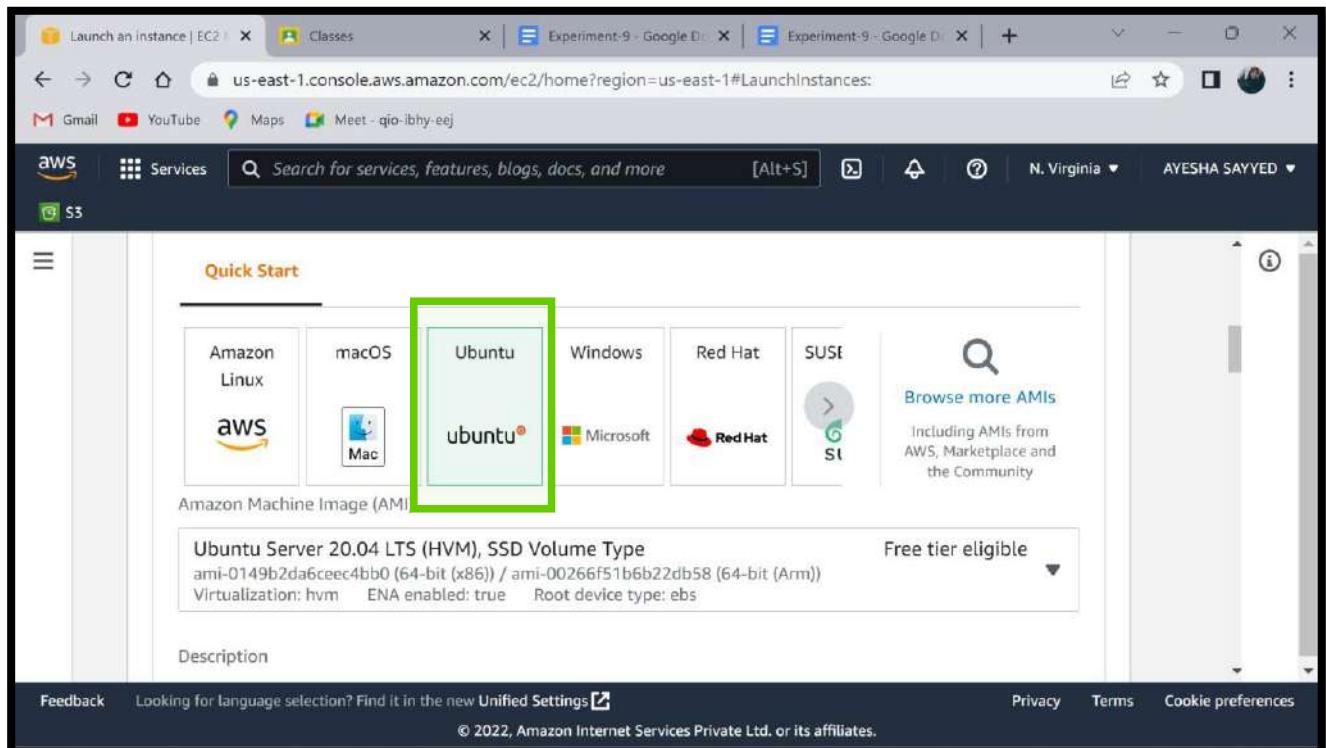
Step 3: Click on “Launch Instances”.



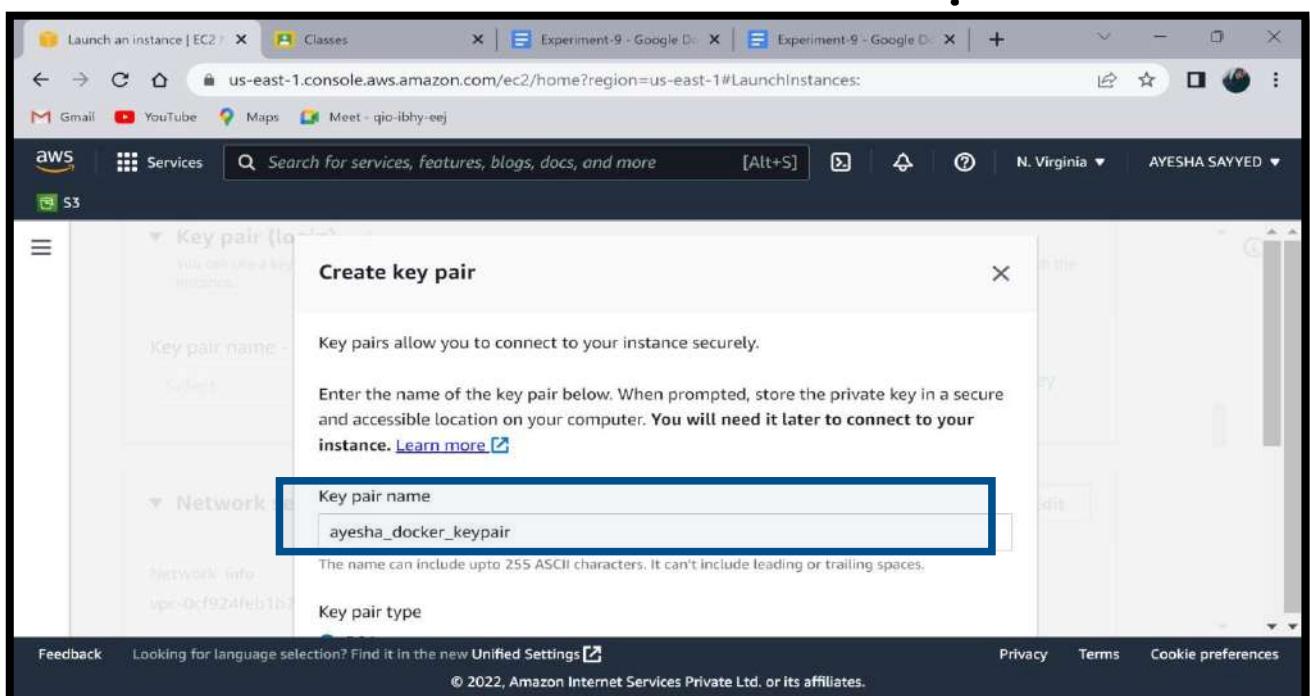
Step 4: Give any name to your “Instance”.



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.

Firewall (security groups) [Info](#)
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.

Create security group Select existing security group

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

Allow SSH traffic from
 Helps you connect to your instance Anywhere
0.0.0.0/0

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

▼ Summary

Number of instances [Info](#)
1

t2.micro

Firewall (security group)
New security group

[Cancel](#) Launch instance

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

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

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 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. The main area shows a table of instances with one entry:

Name	Instance ID	Instance state	Instance type	Status
ayesha_docker...	i-02bd1ee0878221436	Running	t2.micro	Initializ...

A purple box highlights the "Connect" button in the top navigation bar. Below the table, the instance details are shown:

Instance: i-02bd1ee0878221436 (ayesha_docker_45)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID: i-02bd1ee0878221436 Public IPv4 address: ec2-3-86-82-210.compute-1.amazonaws.com Private IPv4 addresses: 172.31.10.10

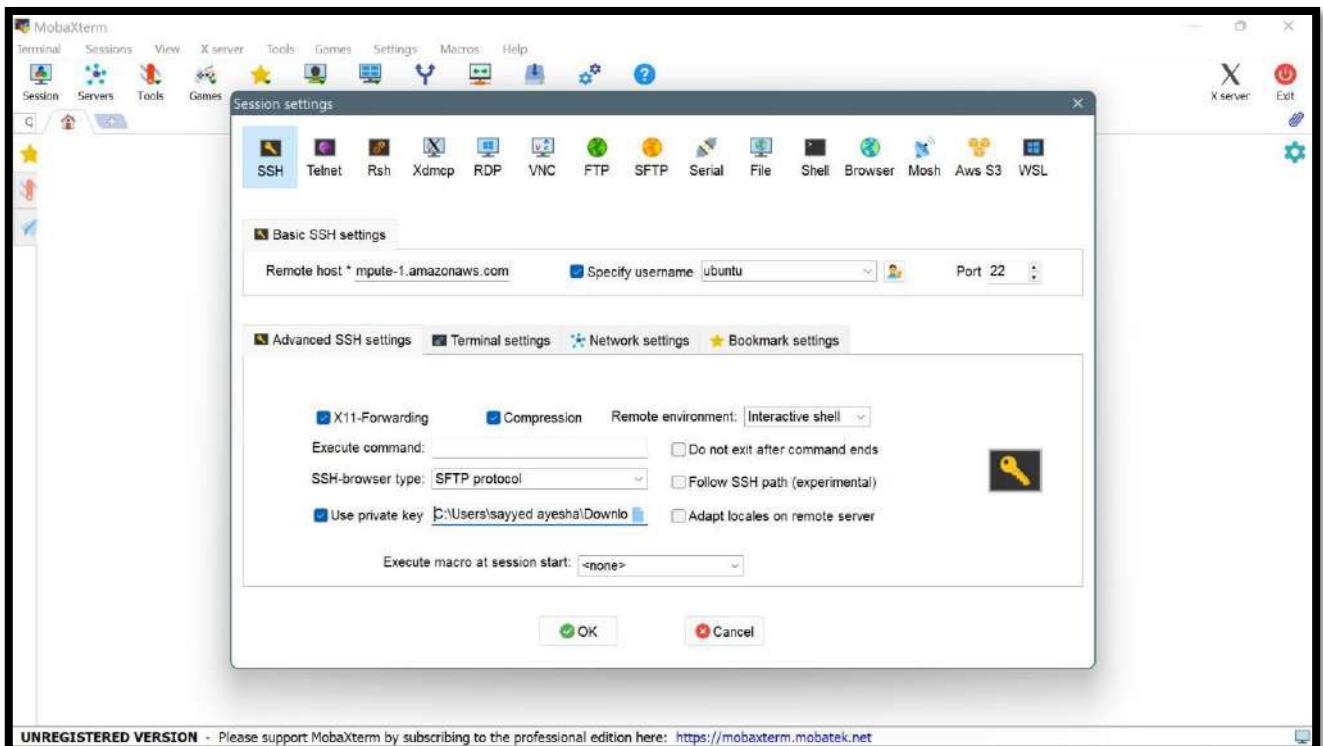
Step 8: Go to SSH client and copy Public DNS.

The screenshot shows the "Connect to instance" page for the same instance. The top navigation bar has tabs for EC2 Instance Connect, Session Manager, SSH client, and EC2 serial console. The SSH client tab is active, indicated by a purple box around its label. Below it, the instance ID is listed as i-02bd1ee0878221436 (ayesha_docker_45). A numbered list provides instructions:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ayesha_docker_keypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
Public DNS copied

Below the list, there's a note: "To connect to your instance using its Public DNS:" followed by the URL ec2-3-86-82-210.compute-1.amazonaws.com. A purple box highlights the "Public DNS copied" message.

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”.

```
ec2-3-86-82-210.compute-1.amazonaws (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
[ 2. ec2-3-86-82-210.compute-1.amaz... ] X server Exit

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.

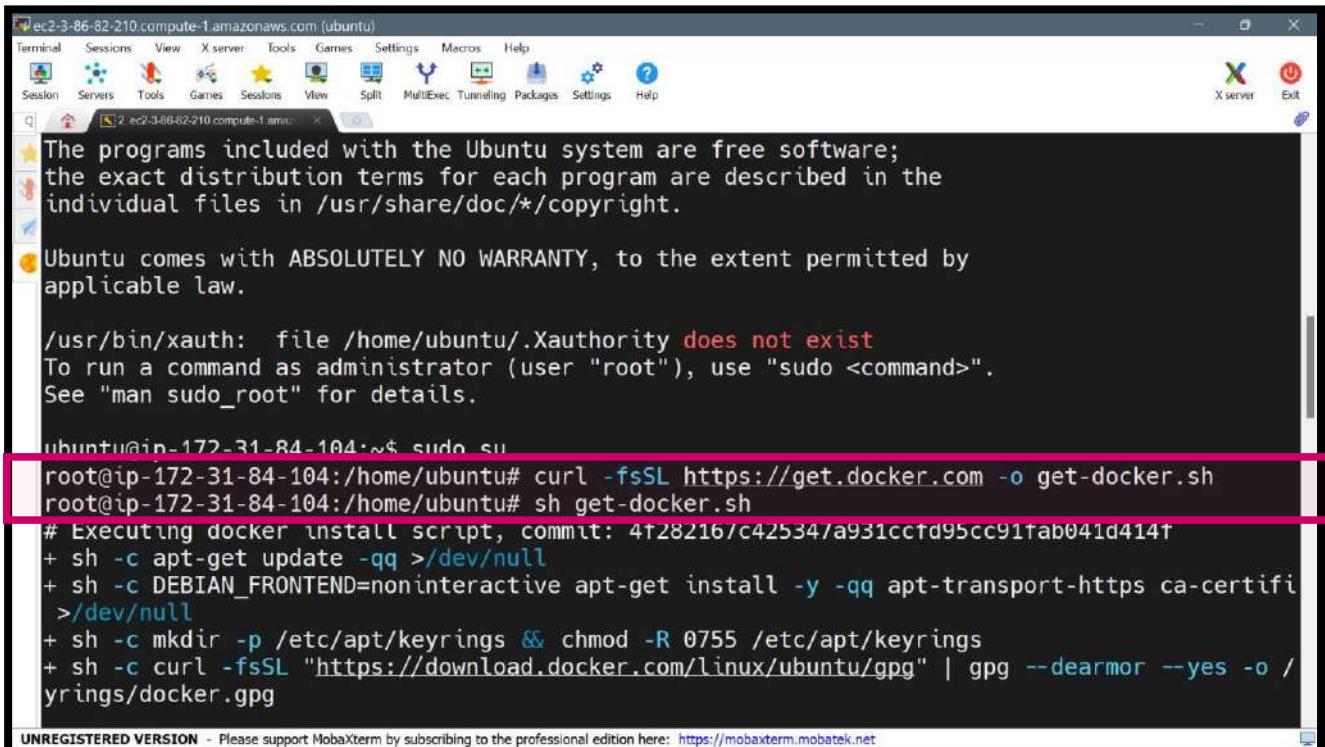
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/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.



```
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 Tunnelling Packages Settings Help
X server Exit

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.

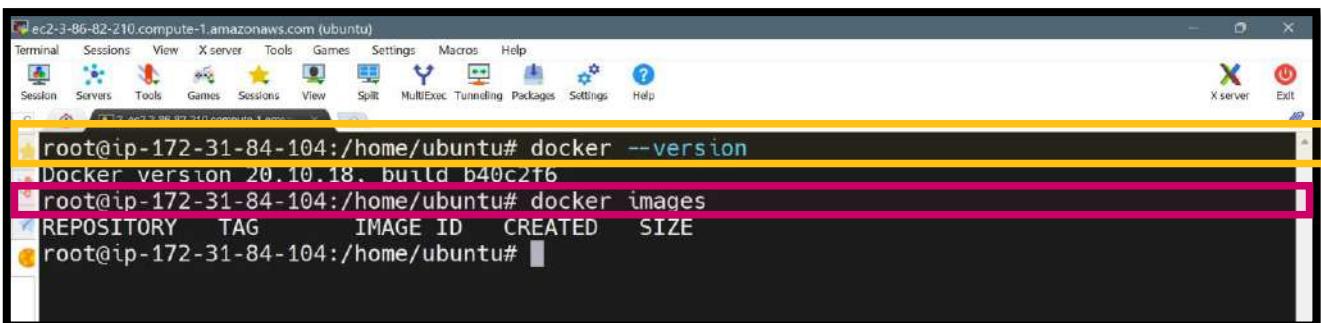
/usr/bin/xauth:  file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

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.

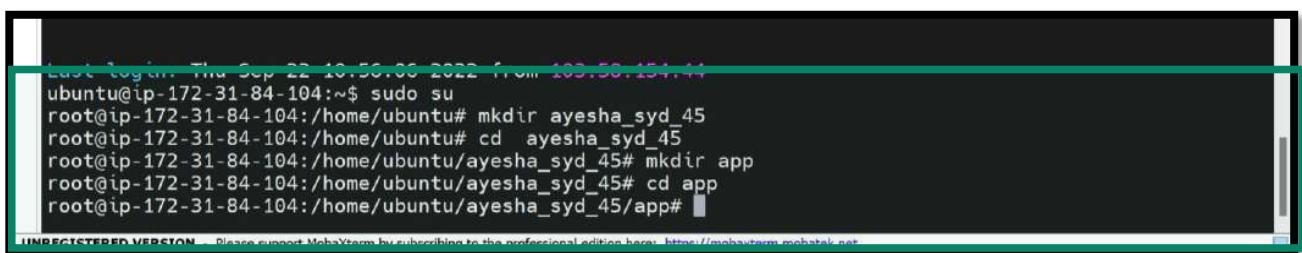
Step 12: Run command “docker --version” then “docker images”.



```
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 Tunnelling Packages Settings Help
X server Exit

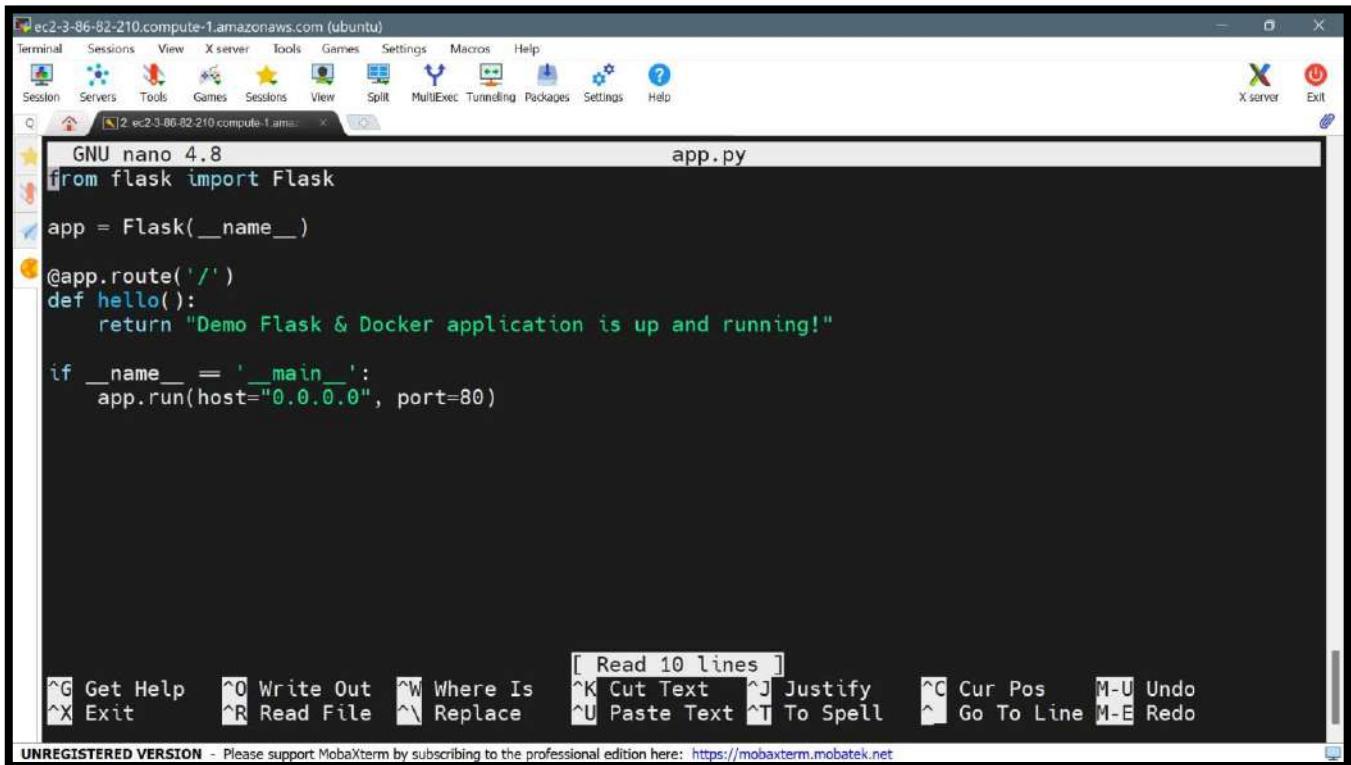
root@ip-172-31-84-104:/home/ubuntu# docker --version
Docker version 20.10.18, build b40c2f6
root@ip-172-31-84-104:/home/ubuntu# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@ip-172-31-84-104:/home/ubuntu#
```

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 103.69.151.11
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 app.py
from flask import Flask

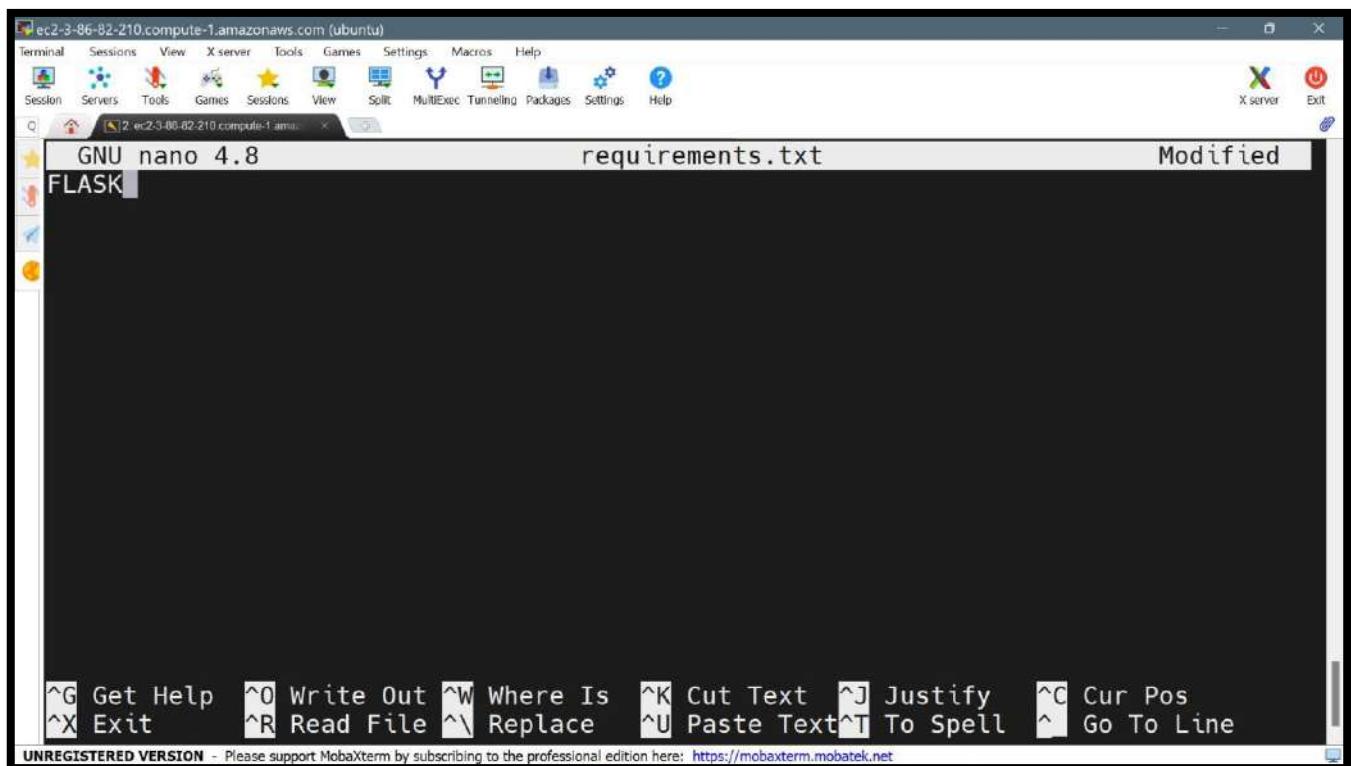
app = Flask(__name__)

@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, Tunnelling, Packages, Settings, and Help. The bottom of the screen shows a status bar with keyboard shortcuts for various functions like Get Help (^G), Write Out (^O), Where Is (^W), Cut Text (^K), Justify (^J), Cur Pos (^C), Undo (^U), Exit (^X), Read File (^R), Replace (^R), Paste Text (^U), To Spell (^T), Go To Line (^L), and Redo (^E). A note at the bottom says "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 the same menu bar, toolbar, and status bar. The status bar at the bottom again notes the "UNREGISTERED VERSION" of MobaXterm.

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

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

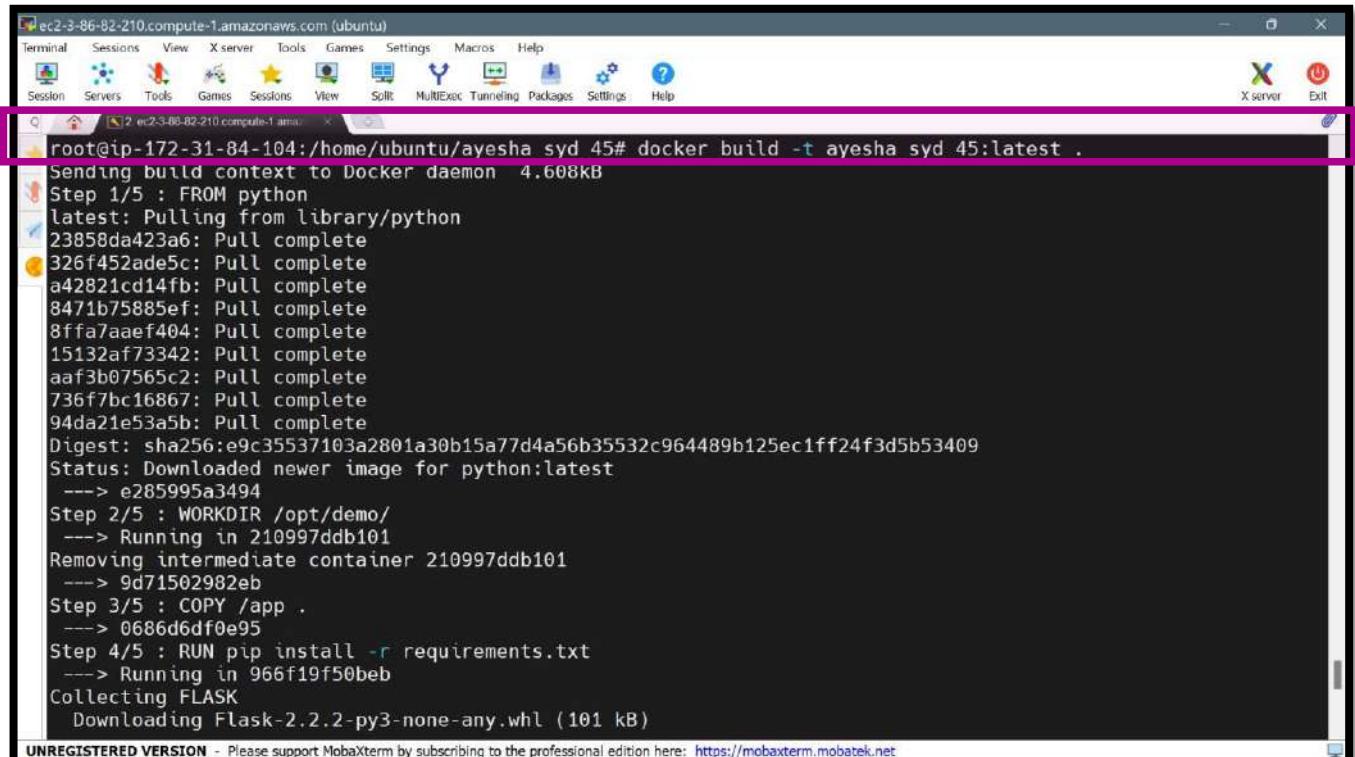
Last login: Thu Sep 22 10:56:06 2022 from 103.58.154.44
ubuntu@ip-172-31-84-104:~$ sudo su
root@ip-172-31-84-104:/home/ubuntu# mkdir ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu# cd ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# mkdir app
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45#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

Write code in Dockerfile.

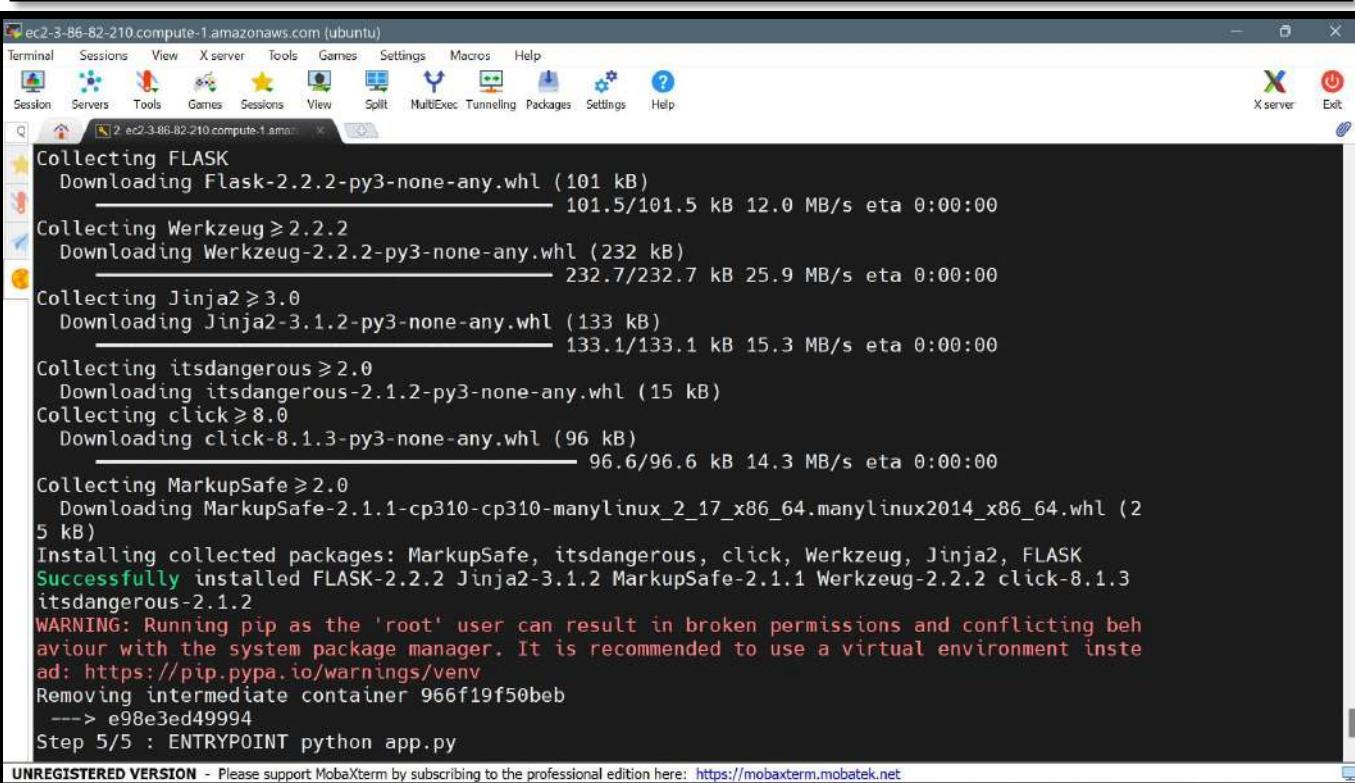
```
Last login: Thu Sep 22 10:56:06 2022 from 103.58.154.44
ubuntu@ip-172-31-84-104:~$ sudo su
root@ip-172-31-84-104:/home/ubuntu# mkdir ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu# cd ayesha_svd_45
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45# mkdir app
root@ip-172-31-84-104:/home/ubuntu/ayesha_svd_45#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

UNREGISTERED VERSION - Please support MobayTerm by subscribing to the professional edition here: https://mobayterm.mobatek.net
```

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

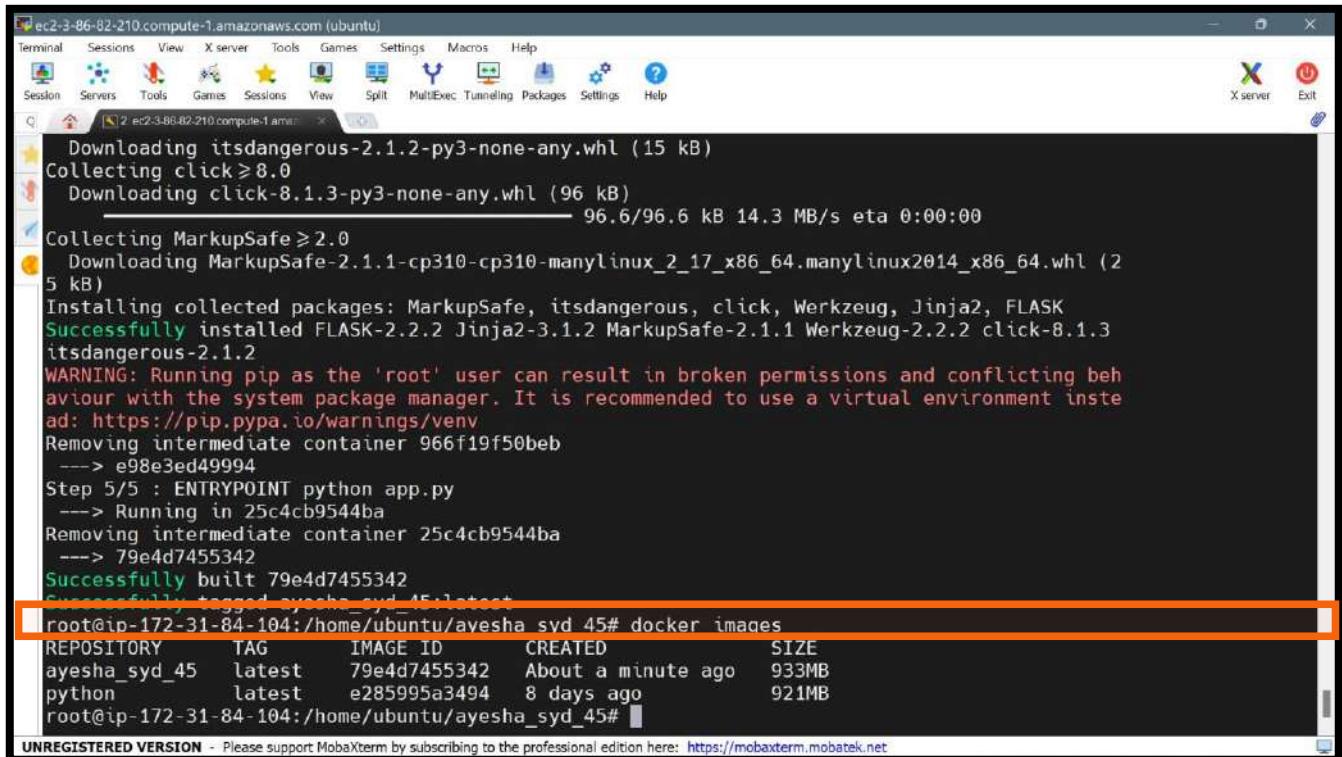


```
ec2-3-86-82-210.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
root@ip-172-31-84-104:/home/ubuntu/ayesha syd 45# docker build -t ayesha syd 45:latest .
Sending 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
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
```

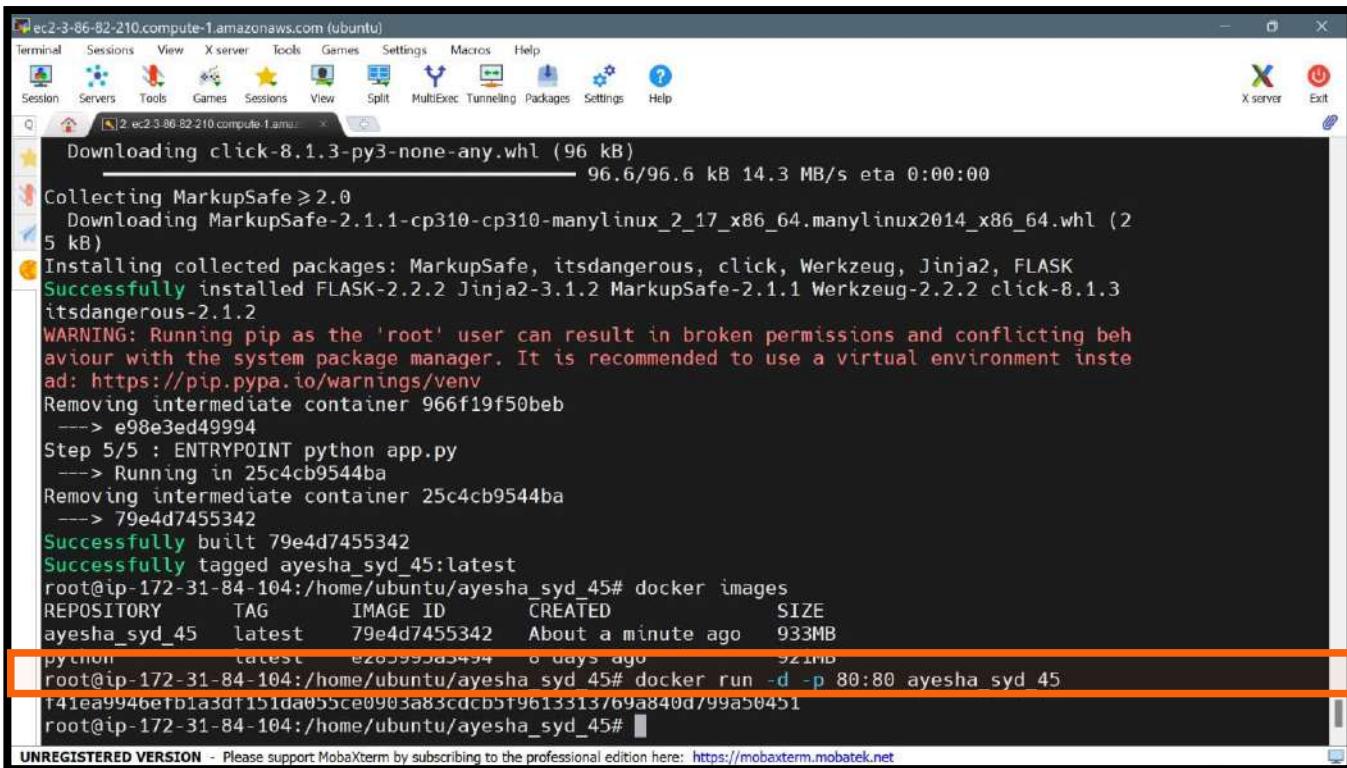
Step 16: Now for checking the images run command as “docker images”.



```
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
t41ea9946efb1a3df151da055ce0903a83cdcb5f9613313769a840d/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 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, there's a link to Instances. The main area displays the Instance summary for an instance with ID i-02bd1ee0878221436, which is named ayesha_docker_45. The summary table includes columns for Instance ID, Public IPv4 address, Private IPv4 addresses, IPv6 address, Instance state, and Private IP DNS name (IPv4 only). The Public IPv4 address is 3.86.82.210. The Instance state is Running. A yellow box highlights the message "Public IPv4 DNS copied" next to the Public IPv4 address row.

Paste it in **the Address Bar**

The screenshot shows a web browser window with the URL ec2-3-86-82-210.compute-1.amazonaws.com. The page content displays the message "Demo Flask & Docker application is up and running!". A yellow circle highlights this message. The browser interface includes a back button, forward button, refresh button, and a search bar at the top. Below the search bar, there are links for Gmail, YouTube, Maps, and Meet.

Step 18: After performing now terminate the instance.

The screenshot shows the AWS EC2 Management Console. In the left sidebar, under 'Instances', the 'Instances' tab is selected. In the main content area, there is one instance listed: 'ayesha_docker...' (Instance ID: i-02bd1ee0878221436). Below the instance list, there is a 'Actions' dropdown menu with several options: 'Stop instance', 'Start instance', 'Reboot instance', 'Hibernate instance', and 'Terminate instance'. The 'Terminate instance' option is highlighted with a green rectangular box. At the bottom of the page, there is a confirmation message: 'Instance: i-02bd1ee0878221436 (ayesha_docker_45)'. The status bar at the bottom right indicates '2/2'.

Click on Terminate, Instance will be Terminated.

The screenshot shows the 'Terminate instance?' confirmation dialog box. The dialog has 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, there is a question: 'Are you sure you want to terminate these instances?'. A checkbox is present next to the instance ID: 'i-02bd1ee0878221436 (ayesha_docker_45)'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Terminate', with 'Terminate' being highlighted with a green rectangular box. The background of the dialog is semi-transparent, showing the EC2 Instances page from the previous screenshot.

Successfully Instance Terminated!!

The screenshot shows the AWS EC2 Management Console. A green notification bar at the top right says "Successfully terminated i-02bd1ee0878221436". Below it, the "Instances (1/1) Info" section displays a table with one row. The row shows an instance named "ayesha_docker..." with Instance ID "i-02bd1ee0878221436", Instance state "Shutting-down", Instance type "t2.micro", and Status "2/2". The left sidebar has a "Instances" section expanded, showing "Instances New". The bottom of the page includes standard links like Feedback, Privacy, Terms, and Cookie preferences.

Simultaneously, delete the Keypair too.

The screenshot shows the AWS EC2 Management Console. A modal dialog box is open, prompting the user to "Delete ayesha_docker_keypair". It contains the instruction "To confirm deletion, type Delete in the field" and a text input field where the word "delete" is typed. There are "Cancel" and "Delete" buttons at the bottom. The background shows the "Key pairs (1/1) Info" section with a note: "ayesha_docker_keypair could be associated with one or more instances." The left sidebar has a "Instances" section expanded, showing "Instances New". The bottom of the page includes standard links like Feedback, Privacy, Terms, and Cookie preferences.

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?

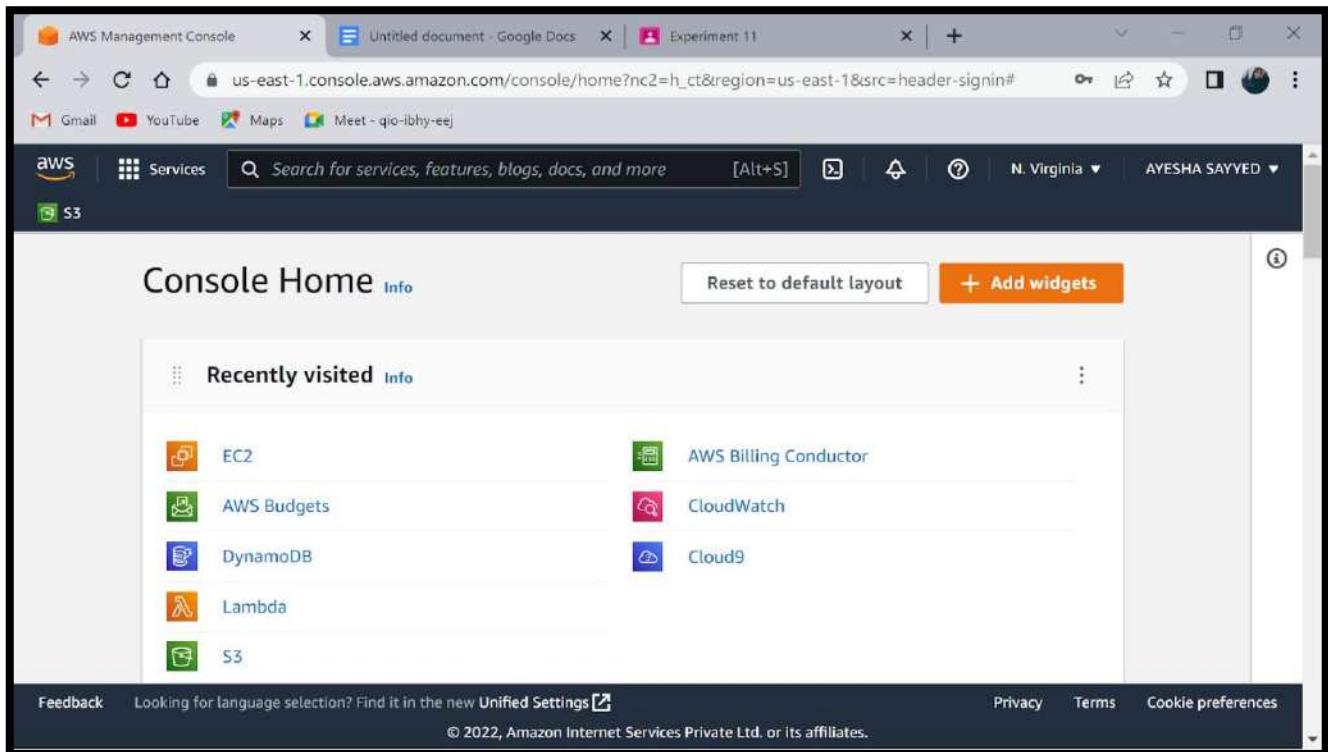
- AWS Elastic Beanstalk is cloud deployment and provisioning service that automates the process of getting applications set up on the Amazon Web Services (AWS) infrastructure.
- To use the service, developers just have to upload their applications. Provisioning, load balancing, autoscaling, and application 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?

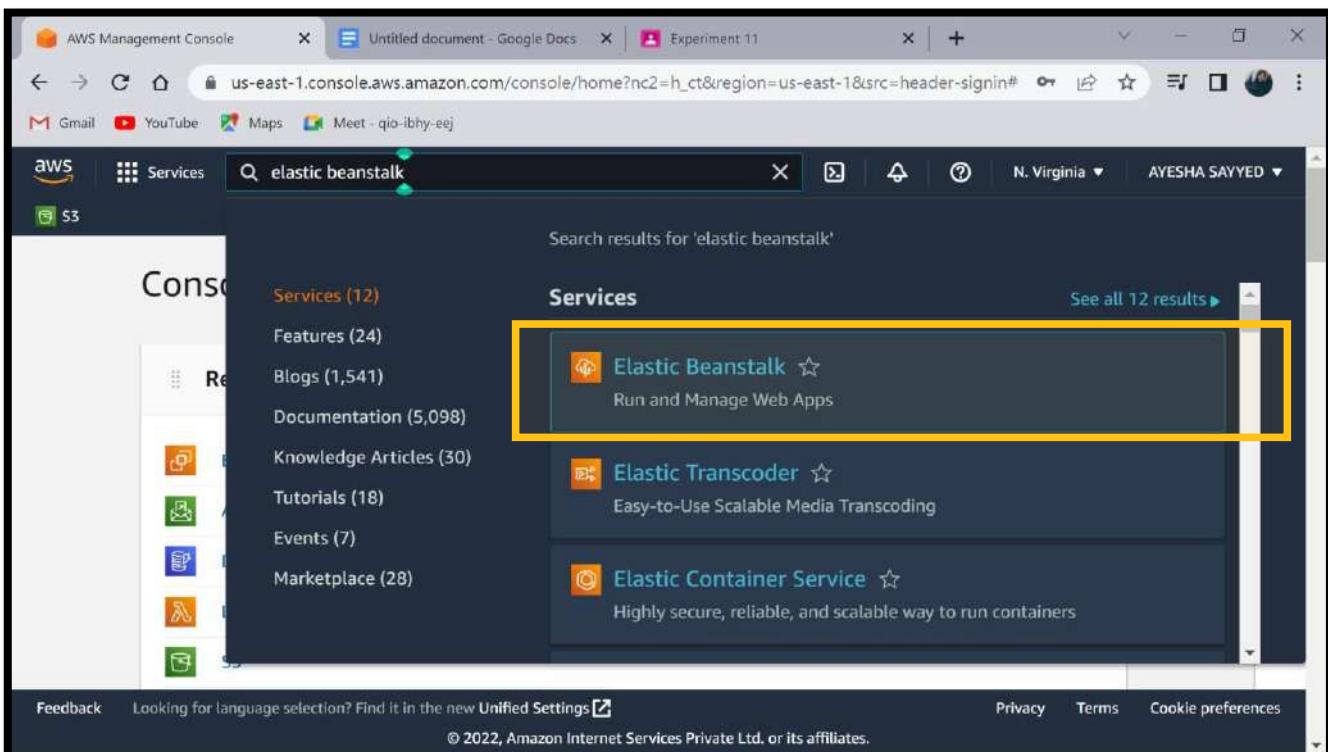
- ▲ Those who want to deploy and manage their applications within minutes in the AWS Cloud.
- ▲ You don't need experience with cloud computing to get started.
- ▲ AWS Elastic Beanstalk supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker web applications.

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

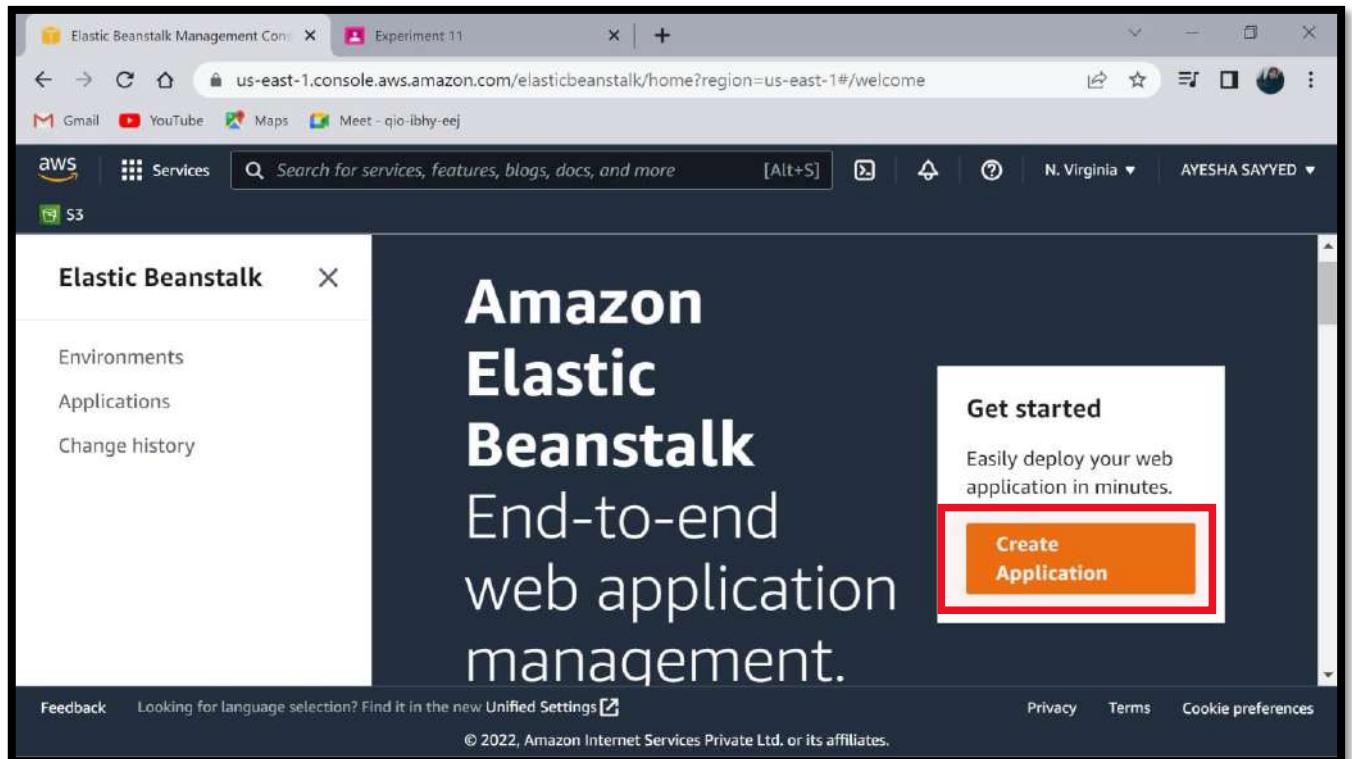
Step 1: Login to AWS Management Console Dashboard.



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 screen shows the 'Application information' section. A purple oval highlights the 'Application name' input field, which contains the value 'application_by_ayesha45'. Below the input field is a note: 'Up to 100 Unicode characters, not including forward slash (/).'

Screenshot 2: Platform Configuration

This screen 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 screen shows the 'Application code' selection section. It offers two options: 'Sample application' (selected) and 'Upload your code'. At the bottom right is a large orange 'Create application' button, which is also highlighted with a purple rectangle.

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

The screenshot shows the AWS Elastic Beanstalk console with a purple circle highlighting the message "Creating Applicationbyayesha45-env This will take a few minutes." Below this, the terminal output shows the following logs:

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

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 Elastic Beanstalk console with a purple circle highlighting the message "Creating Applicationbyayesha45-env This will take a few minutes..". Below this, the terminal output shows the following logs:

```
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.
```

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

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

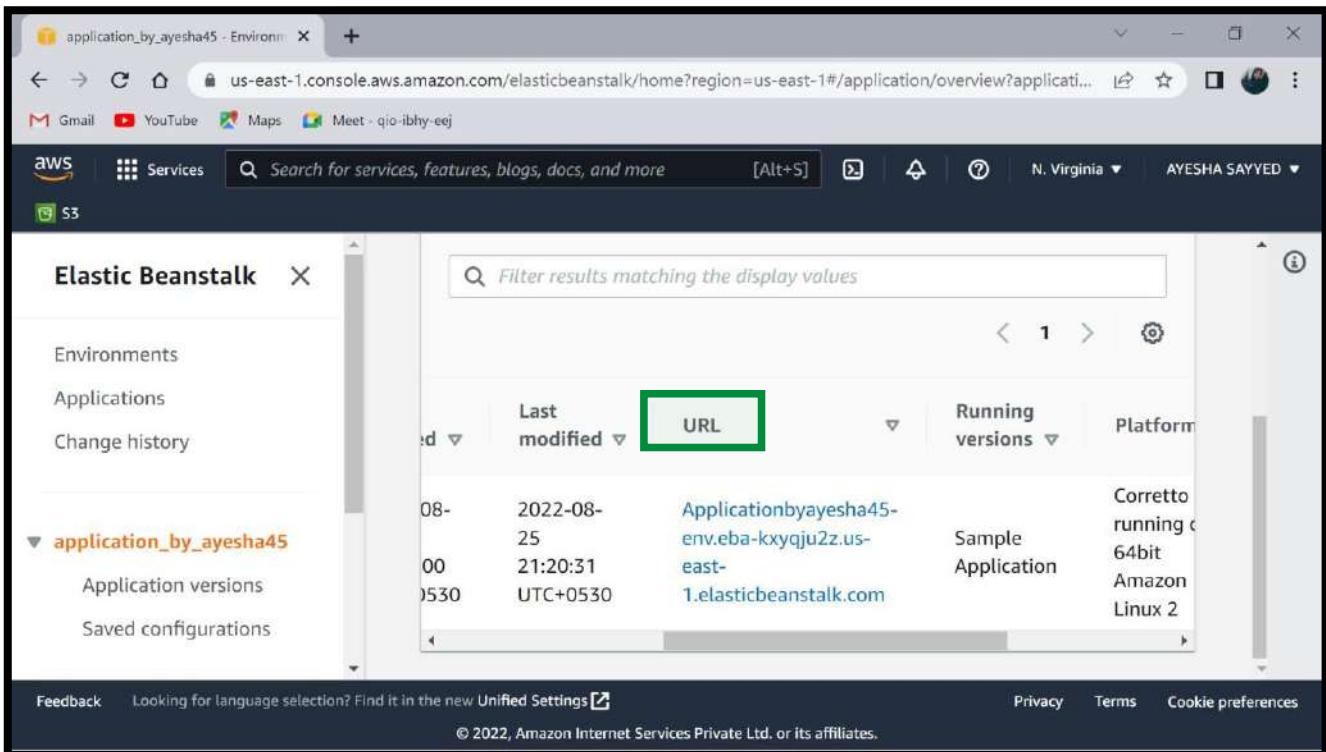
The screenshot shows the AWS Elastic Beanstalk application dashboard for the environment 'Applicationbyayesha45-env'. The main panel displays the application's status: 'Health' is 'Ok' (green checkmark), 'Running version' is 'Sample Application' (with a 'Upload and deploy' button), and the 'Platform' is 'Corretto 17 running on 64bit Amazon Linux 2/3.1'. The left sidebar shows the application's configuration, including 'Application versions' and 'Saved configurations'. The bottom of the screen includes standard AWS navigation links and copyright information.

The screenshot shows the AWS Elastic Beanstalk application dashboard focusing on the 'Recent events' log for the environment 'Applicationbyayesha45-env'. The log details the following 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 shows the application's configuration, including 'Application versions' and 'Saved configurations'. The bottom of the screen includes standard AWS navigation links and copyright information.

Step 7: Click on the URL given.

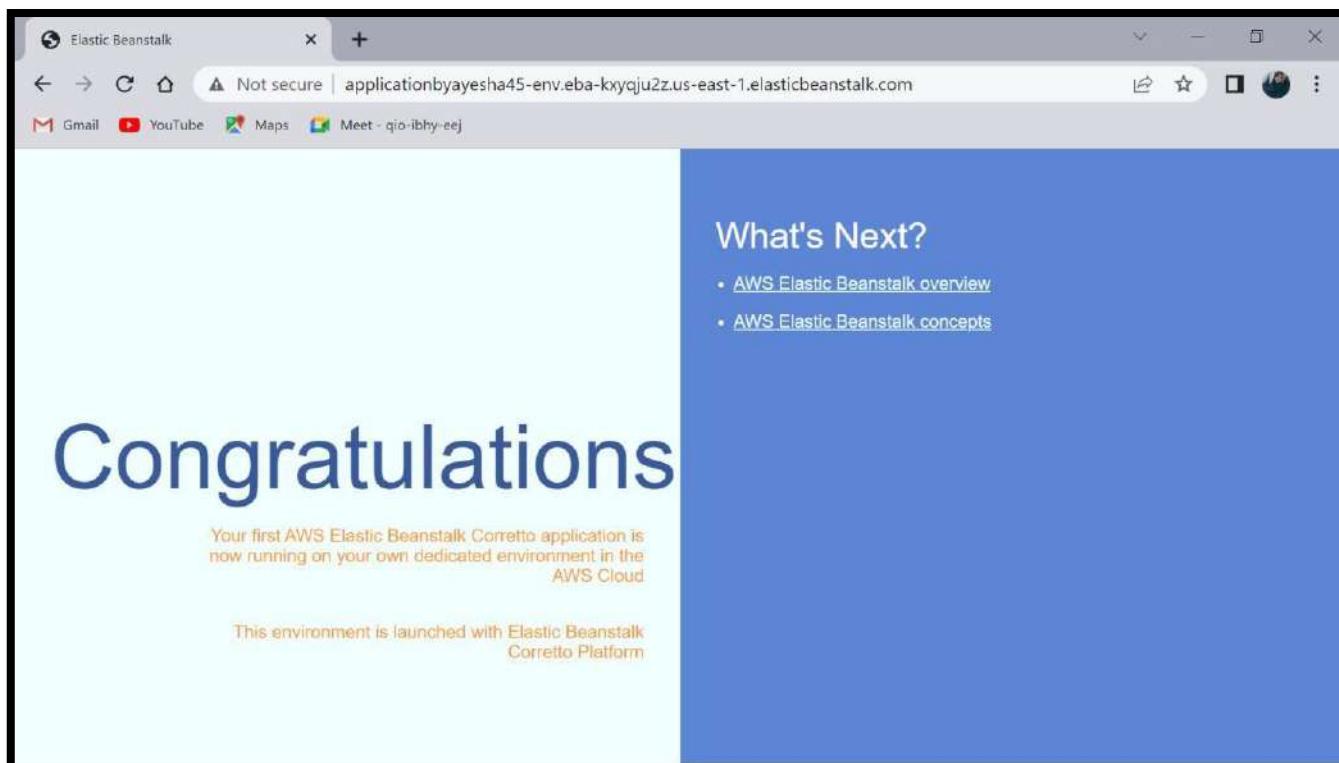


The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with links for Environments, Applications, and Change history. Under Applications, the 'application_by_ayesha45' application is selected. A green box highlights the 'URL' column header in the main table. The table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

Environment	Last modified	URL	Running versions	Platform
application_by_ayesha45	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 the URL: 'Not secure | applicationbyayesha45-env.eba-kxyqju2z.us-east-1.elasticbeanstalk.com'. The page content is divided into two sections. The left section, with a white background, contains a large blue 'Congratulations' heading, followed by the text: 'Your first AWS Elastic Beanstalk Corretto application is now running on your own dedicated environment in the AWS Cloud'. Below this, it says: 'This environment is launched with Elastic Beanstalk Corretto Platform'. The right section, with a blue background, is titled 'What's Next?' and lists two items: 'AWS Elastic Beanstalk overview' and 'AWS Elastic Beanstalk concepts'.

Step 8: Now search for EC2 and select it.

The screenshot shows the AWS Elastic Beanstalk search results. The search bar at the top contains 'EC2'. Below the search bar, there is a sidebar with links to 'Environments', 'Applications', 'Change history', and a section for the selected application 'application_by_ayesha45' which includes 'Application versions' and 'Saved configurations'. The main content area shows a list of services under 'Services (8)'. The 'EC2' service card is highlighted with an orange box. Other services listed include 'Features (46)', 'Blogs (1,794)', 'Documentation (131,021)', 'Knowledge Articles (30)', 'Tutorials (19)', 'Events (8)', and 'Marketplace (1,540)'. To the right of the list, there is a link 'See all 8 results ▾'. At the bottom of the page, there is a feedback link, privacy terms, and cookie preferences information.

Step 9: Click on Instance.

The screenshot shows the AWS EC2 Management Console dashboard. The left sidebar has a 'New EC2 Experience' section and a 'EC2 Dashboard' section with links to 'EC2 Global View', 'Events', 'Tags', 'Limits', and 'Instances'. The 'Instances' section is expanded, showing 'Instances New' (which is highlighted with a blue oval) and 'Instance Types'. The main content area features a 'Placement groups' section with 0 items, a 'Security groups' section with 4 items, and a 'Snapshots' section with 0 items. There is also a 'Volumes' section with 1 item. A callout box provides information about creating Microsoft SQL Server Always On availability groups. Below this, there is a 'Launch instance' section with a large orange 'Launch instance' button and a 'Service health' section with a 'AWS Health Dashboard' button. At the bottom, there is a feedback link, privacy terms, and cookie preferences information.

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' page for the instance i-07d263faaf4aea485. The sidebar on the left is identical to the previous screenshot. The main content area is titled 'Instance summary for i-07d263faaf4aea485 (Applicationbyayesha45-env)'. It includes sections for 'Info' (Updated less than a minute ago), 'Actions' (Connect, Instance state: Running), and various details like Instance ID, Public IPv4 address, Private IPv4 addresses, Instance state, Public IPv4 DNS, Hostname type, and IP name. The 'Actions' section also includes links for 'Edit' and 'Delete'. 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	① 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	Monitoring
Tell us what you think	Linux/UNIX (Inferred)	vpc-ami-016fd50b7ccadc069	disabled
EC2 Dashboard	Platform details	AMI name	Termination protection
EC2 Global View	Linux/UNIX	aws-elasticbeanstalk-amzn-2.0.20220719.64bit-eb_corretto17_amazon_linux_2-hvm-2022-08-02T13-13	Disabled
Events	Stop protection	Launch time	AMI location
Tags	Disabled	Thu Aug 25 2022 21:18:09 GMT+0530 (India Standard Time)	amazon/aws-elasticbeanstalk-amzn-2.0.20220719.64bit-
Limits		(17 minutes)	
Instances			
Instances New			
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		
1			
Instances			
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 options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under Instances, 'Instances' is selected. The main area shows a table titled 'Instances (1/1)'. A single instance is listed: 'Instance: i-07d263faaf4aea485 (Applicationbyayesha45-env)'. The instance is in a 'Running' state, t2.micro type, with 2/2 available. Below the table, there's a 'Actions' dropdown menu with several options: Stop instance, Start instance, Reboot instance, Hibernate instance, and Terminate instance. The 'Terminate instance' option is circled in pink. At the bottom of the screen, there's a footer with links for Feedback, Privacy, Terms, and Cookie preferences, along with a copyright notice for 2022, Amazon Internet Services Private Ltd. or its affiliates.

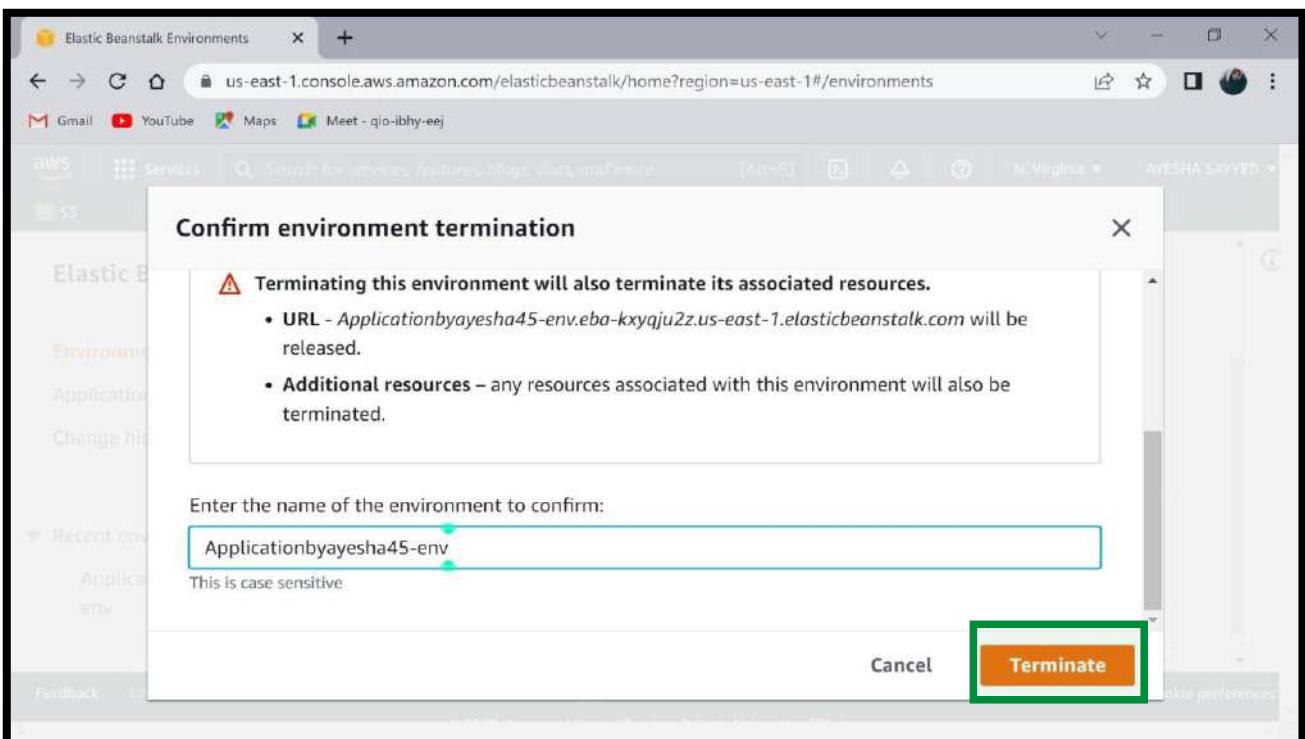
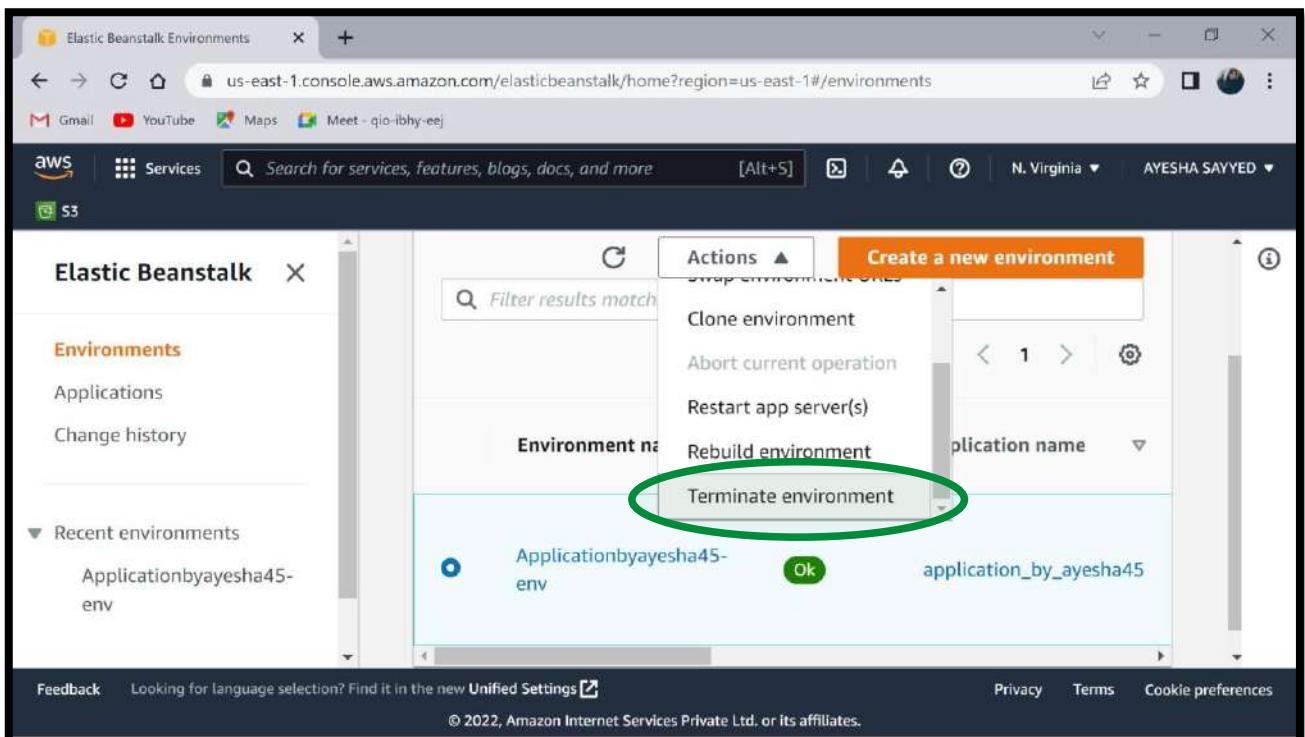
This screenshot shows a confirmation dialog box titled 'Terminate instance?'. Inside the dialog, there's a warning message: 'On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.' Below the message, it asks 'Are you sure you want to terminate these instances?' followed by a list: '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 at the bottom right. The background of the dialog has a purple border.

Terminated Successfully

The screenshot shows the AWS EC2 Management Console interface. A green notification bar at the top center 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". The table has columns for Name, Instance ID, Instance state, Instance type, and Status. One row is visible, showing "Applicationby..." as the Name, "i-07d263faaf4aea485" as the Instance ID, "Shutting-down" as the Instance state, "t2.micro" as the Instance type, and "2/2" as the Status. At the bottom of the page, there is a section titled "Instance: i-07d263faaf4aea485 (Applicationbyayesha45-env)".

The screenshot shows the AWS EC2 Management Console interface. The main content area displays a table titled "Instances (1) Info". The table has columns for Name, Instance ID, Instance state, Instance type, and Status. One row is visible, showing "Applicationby..." as the Name, "i-07d263faaf4aea485" as the Instance ID, "Terminated" as the Instance state, "t2.micro" as the Instance type, and "-" as the Status. The "Instance state" column for this row is highlighted with a yellow rectangle.

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, showing 'Applicationbyayesha45-env'. The main area has a table of applications. The first row in the table is highlighted with a blue oval. The table columns include Application name, Environment name, and Date created. The Actions menu at the top right of the table has several options: Create environment, Delete application, View application versions, View saved configurations, and Restore terminated environment. The 'Restore terminated environment' option is also circled in blue. The table shows one entry: 'application_by_ayesha45' with environment 'Applicationbyayesha45-env' and creation date '2022-08-25 21:16:53 UTC+0530'.

This 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, there's a text input field with the placeholder 'Enter the name of the application to confirm:' and a pre-filled value 'application_by_ayesha45'. At the bottom of the dialog are two buttons: 'Cancel' and a large orange 'Delete' button, which is also outlined in blue.

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

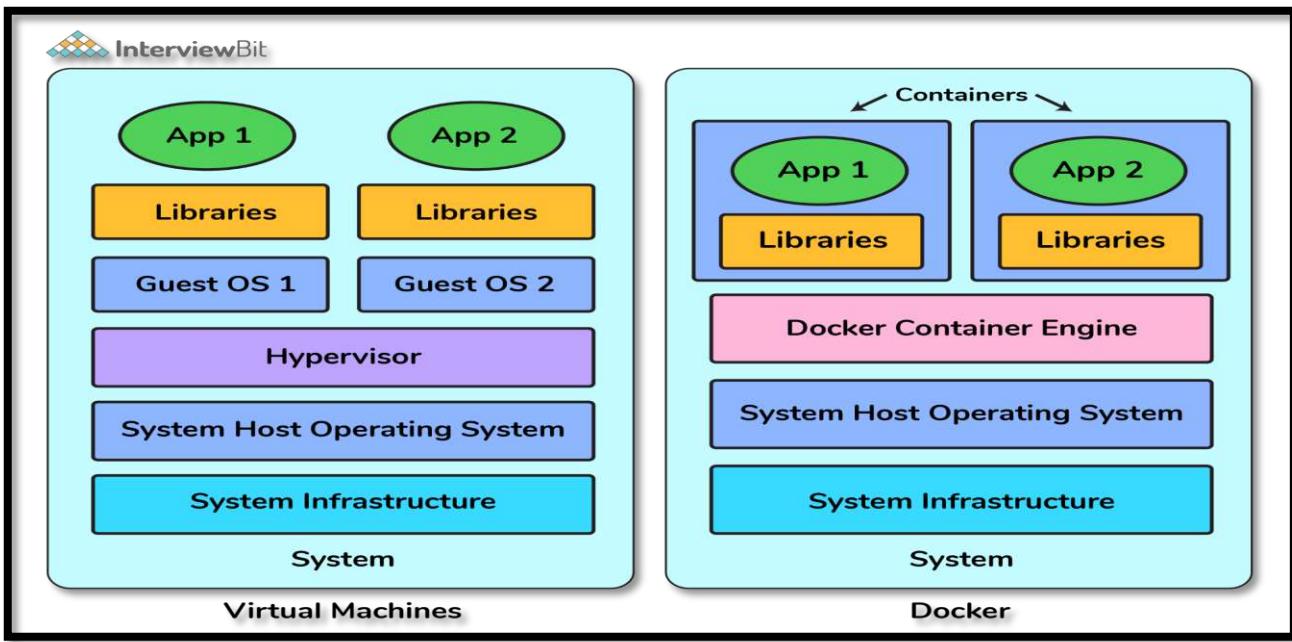
becomes higher the longer a restart takes, so you're likely to want to have another instance to minimize that risk. This is where Kubernetes comes in. Kubernetes is a platform for managing containerised services. This means that it's a tool made to abstract away details such as separation of nodes, while 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

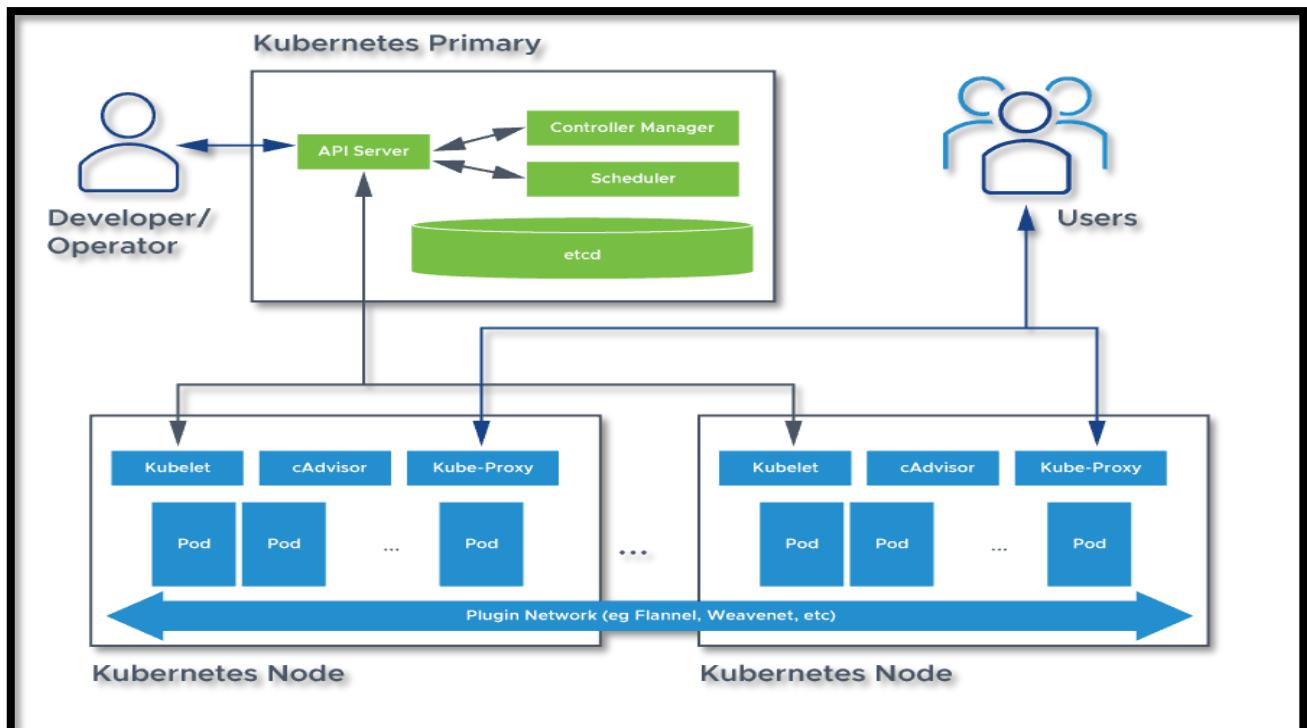
Q3. What is the difference between deploying applications on hosts and containers?

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

for closing your Amazon account (Optional)

Yes, I want to pe

Amazon Account and delete my data.

Close My Account