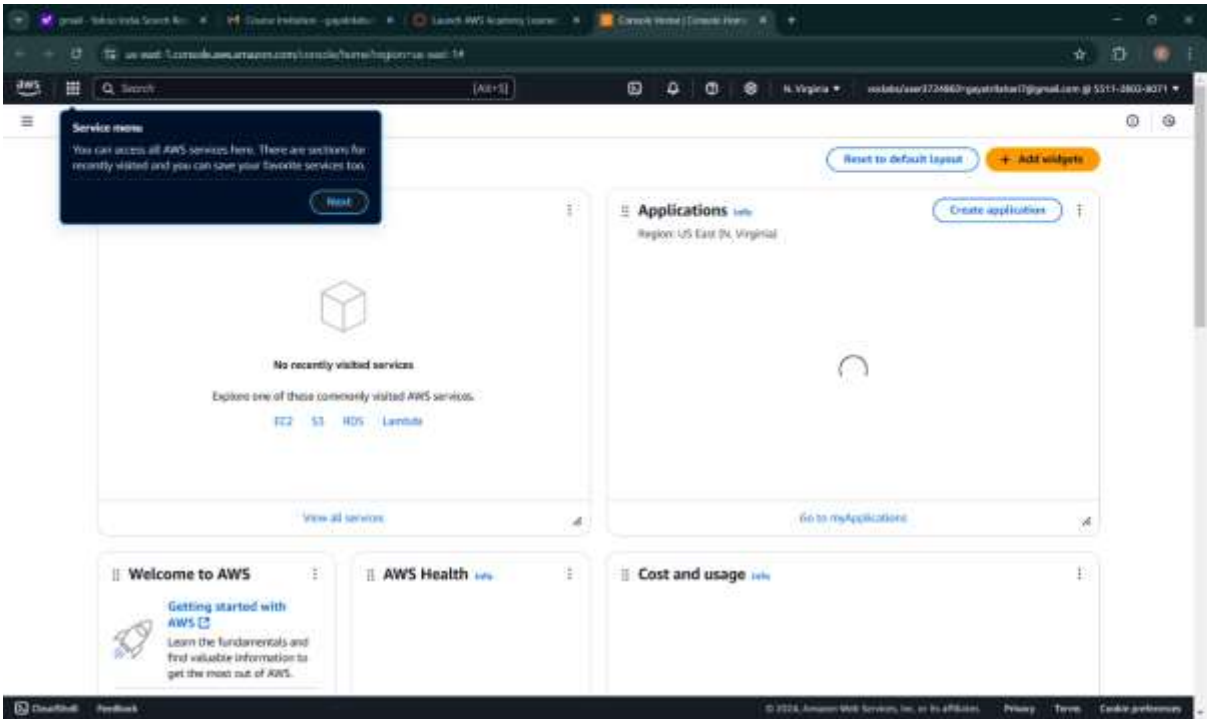
DEPLOYING A WEBPAGE USING EC2 INSTANCE ON CLOUD

1. Enter to the AWS Dashboard



You can access all AWS services here. There are sections for recently visited and you can save your favorite services too.

Region: US East (N. Virginia)

No recently visited services

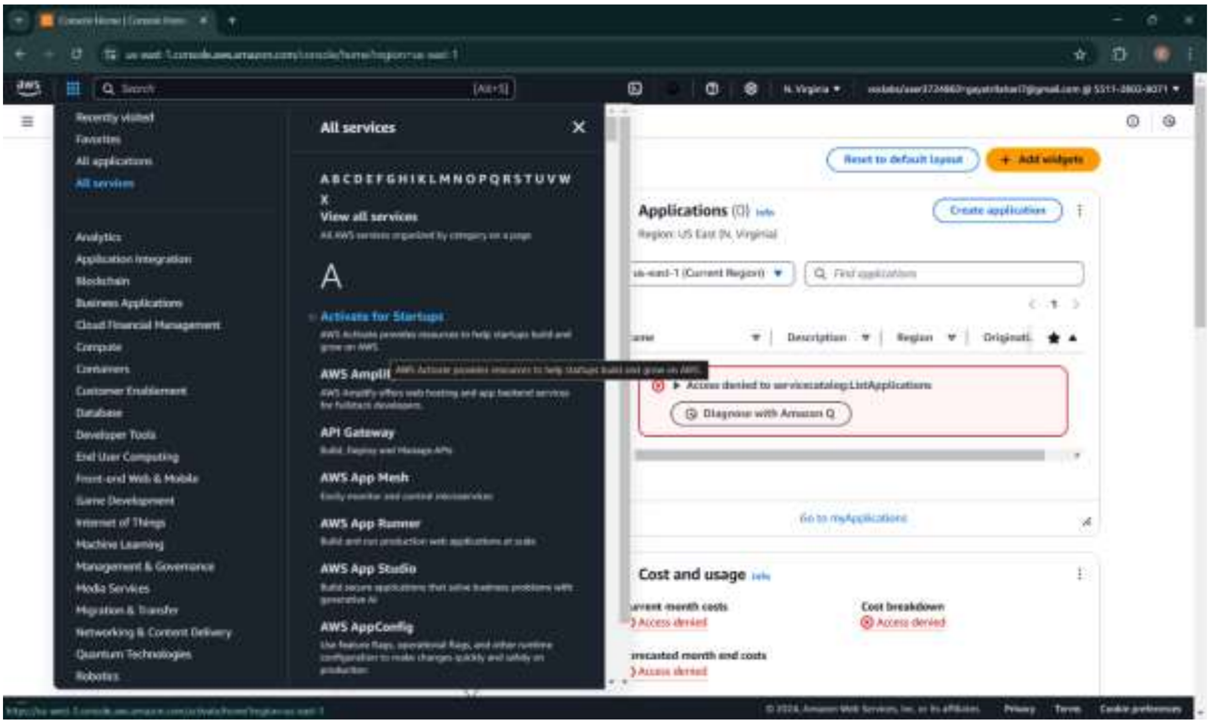
Explore one of these commonly visited AWS services.

EC2 S3 RDS Lambda

Learn the fundamentals and find valuable information to get the most out of AWS.

© 2024, Amazon Web Services, Inc. or its affiliates.

2. Click on Services on top left corner→ then click on All services→ then click on View all services



All AWS services organized by category on a page

AWS Activate provides resources to help startups build and grow on AWS.

AWS Amplify offers web hosting and app backend services for fullstack developers.

Build, Deploy and Manage APIs

Easily monitor and control microservices

Build and run production web applications at scale

Build secure applications that solve business problems with generative AI

Use feature flags, operational flags, and other runtime configuration to make changes quickly and safely on production

Region: US East (N. Virginia)

us-east-1 (Current Region)

Access denied to servicecatalog:ListApplications

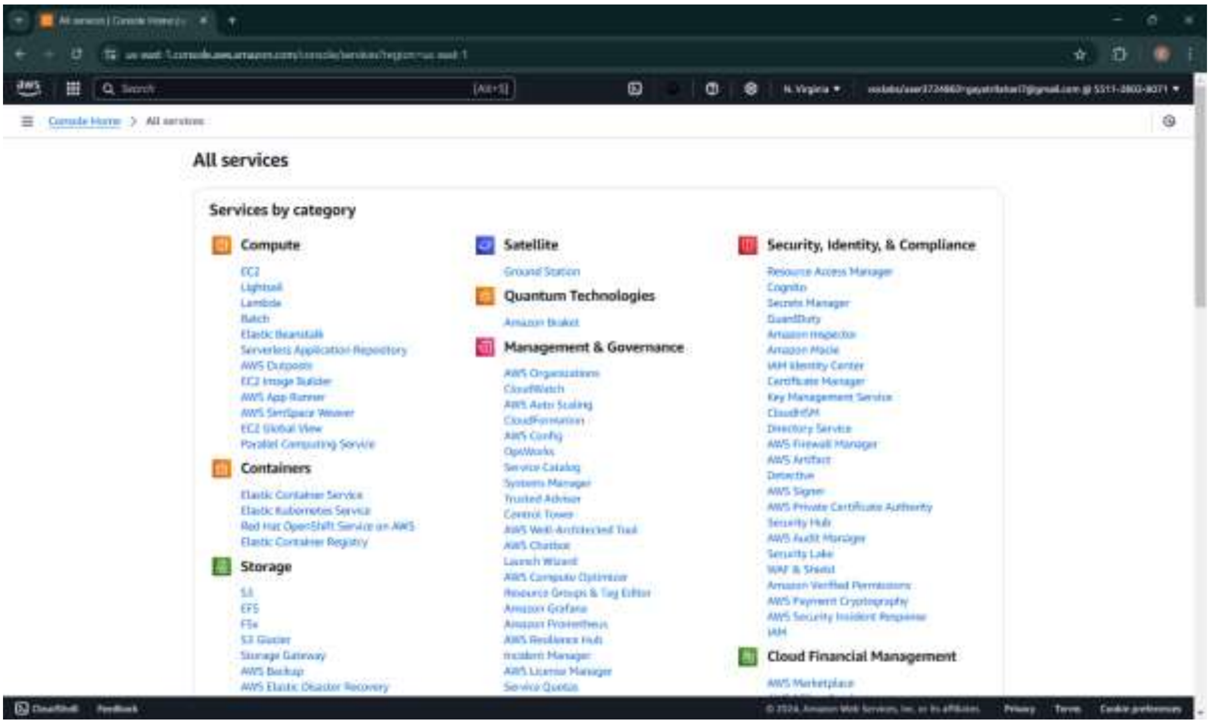
Current month costs: Access denied

Cost breakdown: Access denied

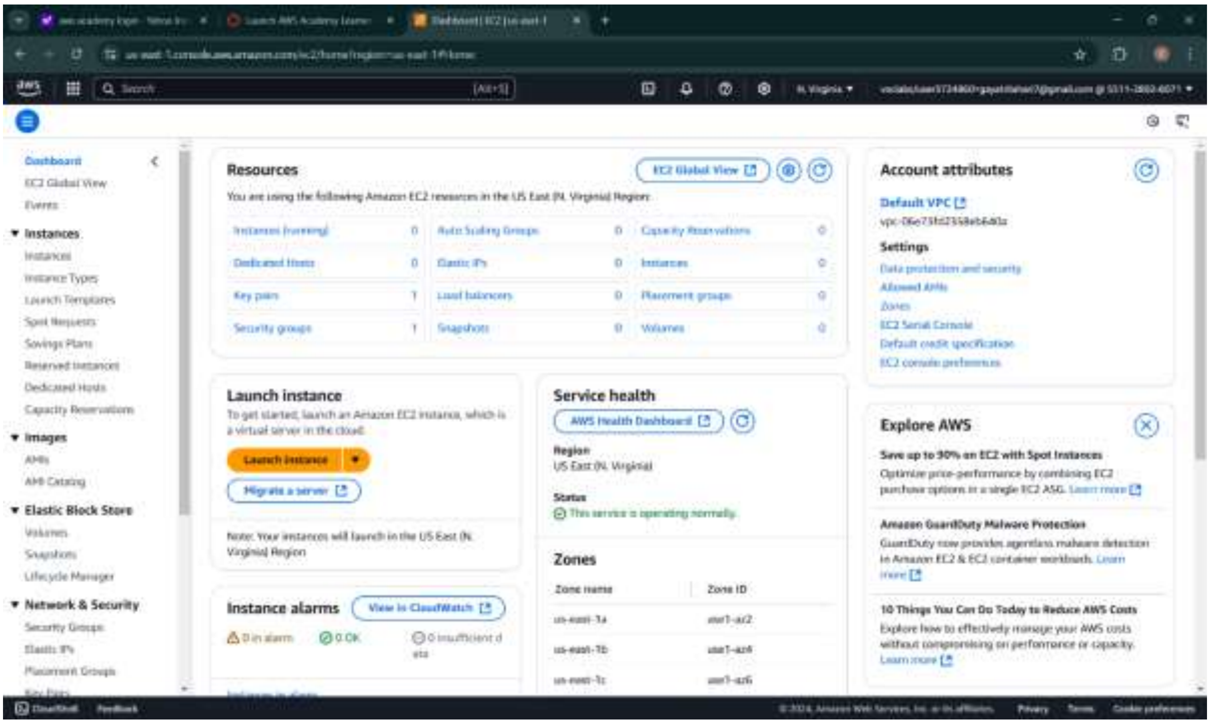
Forecasted month end costs: Access denied

© 2024, Amazon Web Services, Inc. or its affiliates.

3. These are the various services provided by AWS



4. Click on EC2
5. This is the EC2 Dashboard



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	1	Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

Instance alarms

0 in alarm 0 OK 0 insufficient data

Service health

Region
US East (N. Virginia)

Status
This service is operating normally.

Zones

Zone name	Zone ID
us-east-1a	use1-az2
us-east-1b	use1-az4
us-east-1c	use1-az6

Account attributes

Default VPC
vpc-06e73fd2358eb6d0a

Settings
Data protection and security
Allowed AMIs
Zones
EC2 Serial Console
Default credit specification
EC2 console preferences

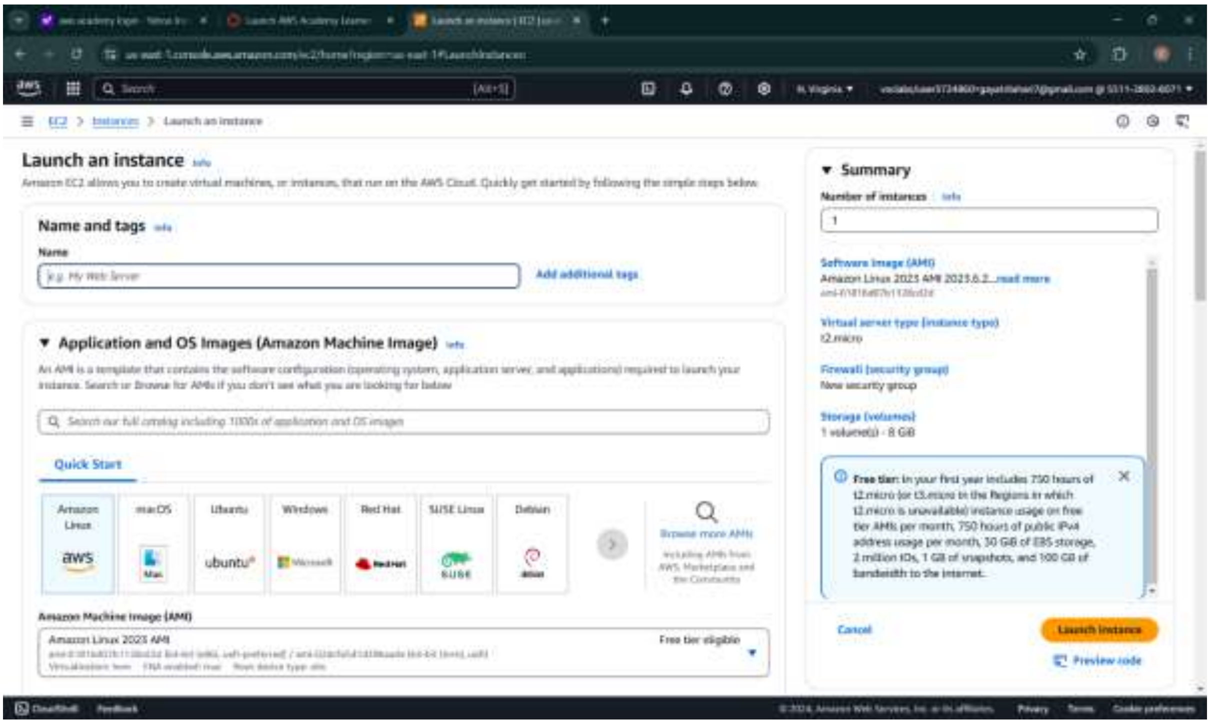
Explore AWS

Save up to 90% on EC2 with Spot Instances
Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. Learn more

Amazon GuardDuty Malware Protection
GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. Learn more

10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity. Learn more

6. click Launch instances.
7. This is the Launch an instance dashboard



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: e.g. My Web Server

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Machine Image (AMI): Amazon Linux 2023 AMI — Free tier eligible

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more

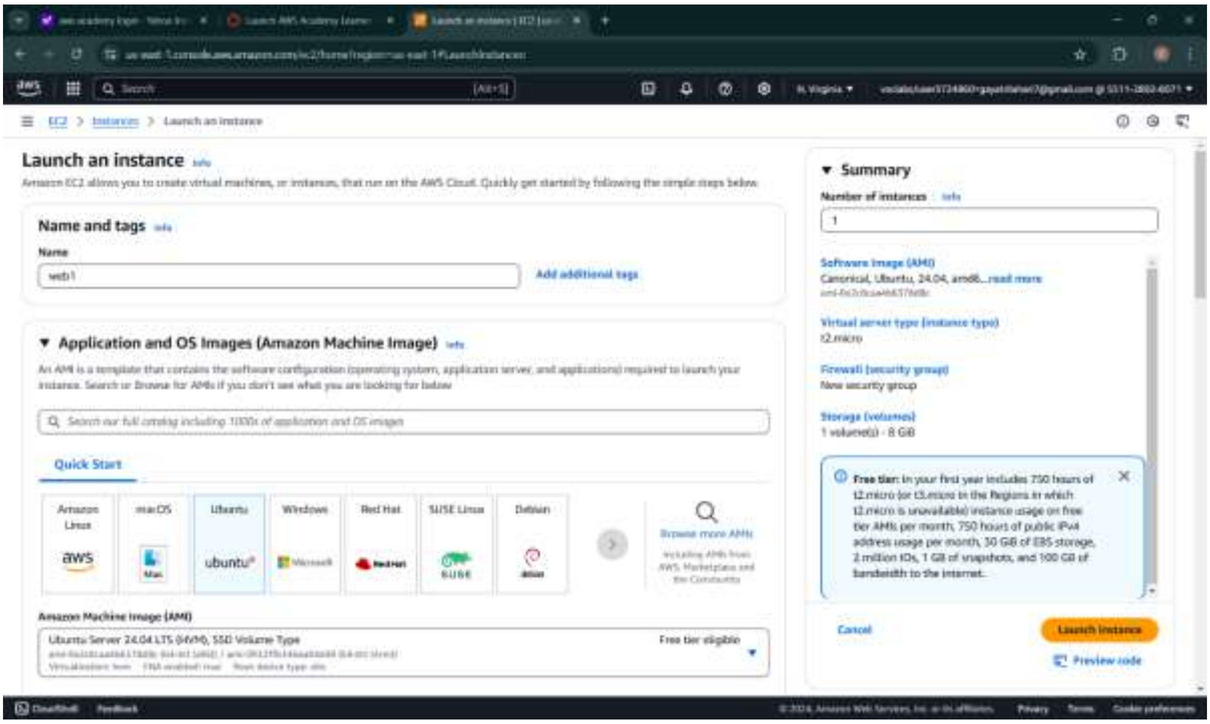
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

8. Give a name of your choice to the instance → Scroll down
9. Select the required OS→ select Ubuntu



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: web1

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Machine Image (AMI): Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more

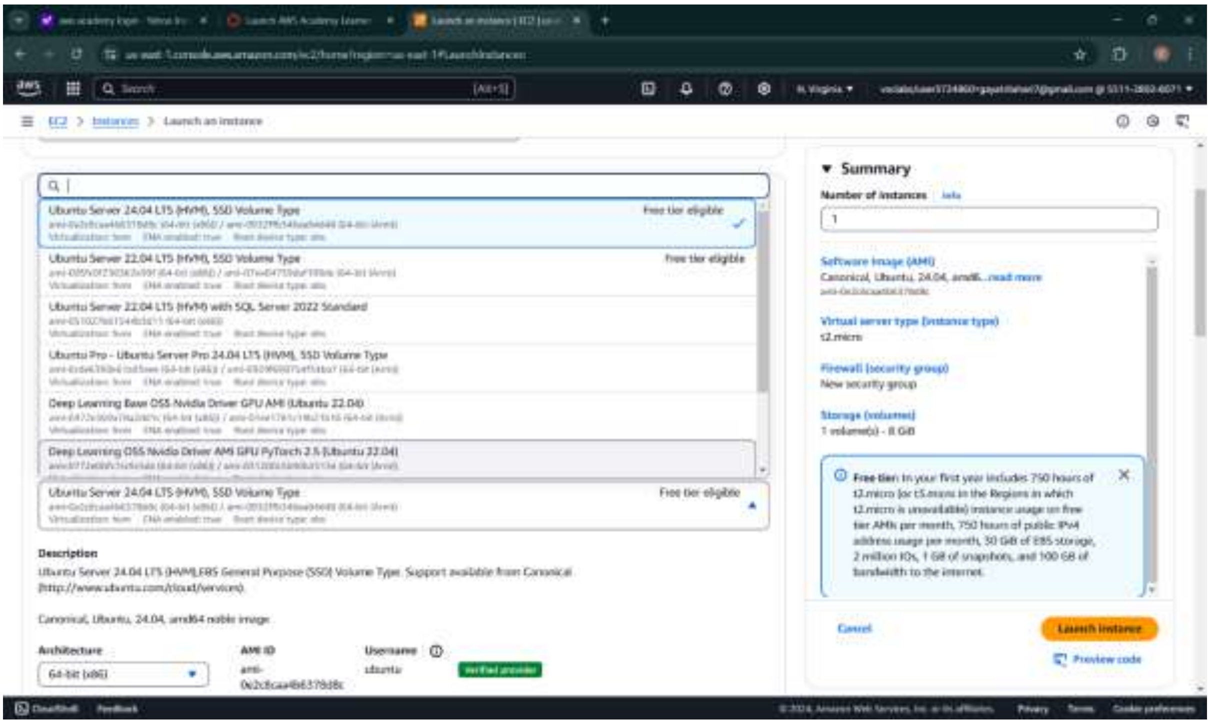
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

10. Under Ubuntu we have many versions→ select any Free tier to avoid billing→ better go for default



EC2 > Instances > Launch an instance

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible ✓

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible

Ubuntu Server 22.04 LTS (HVM) with SQL Server 2022 Standard

Ubuntu Pro - Ubuntu Server Pro 24.04 LTS (HVM), SSD Volume Type

Deep Learning Base OSS Nvidia Driver GPU AMI (Ubuntu 22.04)

Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.5 (Ubuntu 22.04)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

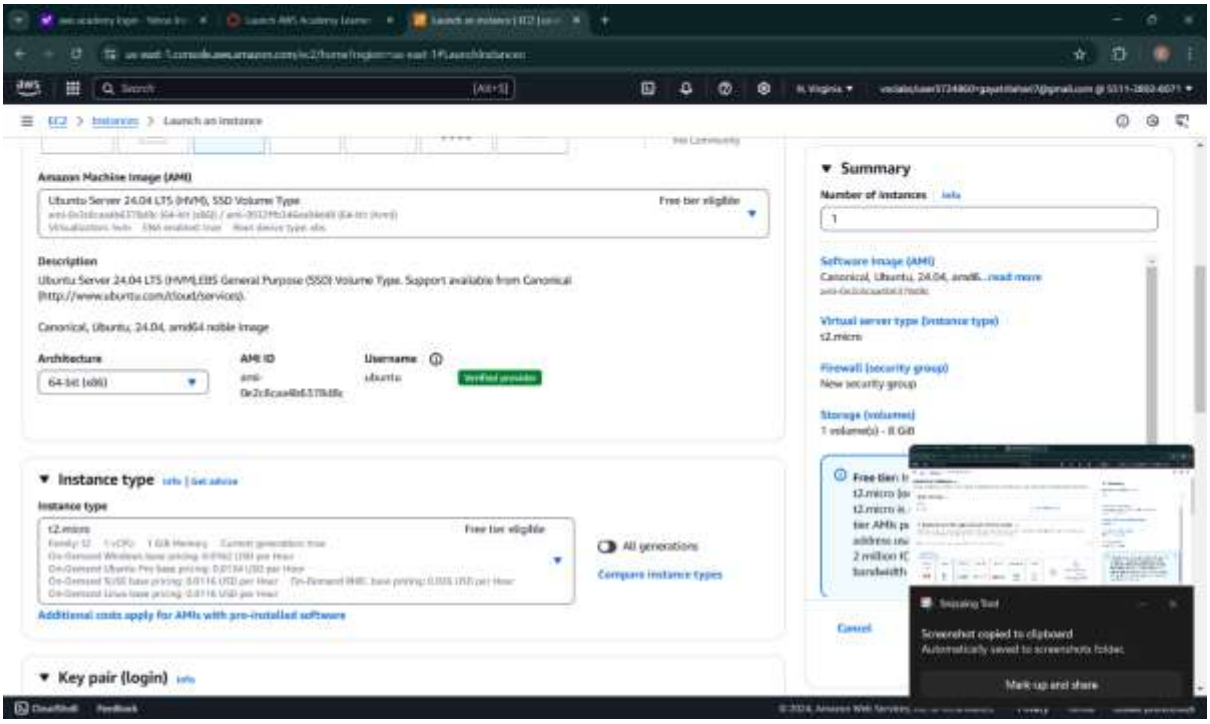
Canonical, Ubuntu, 24.04, amd64 noble image

Architecture: 64-bit (x86)
AMI ID: ami-0e2c8caa4b6378d8c
Username: ubuntu
Verified provider

Summary
Number of instances: 1
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
Virtual server type (instance type): t2.micro
Firewall (security group): New security group
Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

11. Select 64-bit (x86) which is default → x86 being a common processor architecture for traditional servers, while Arm is known for its energy-efficient design often used in mobile devices and newer cloud-based systems.



Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture: 64-bit (x86)
AMI ID: ami-0e2c8caa4b6378d8c
Username: ubuntu
Verified provider

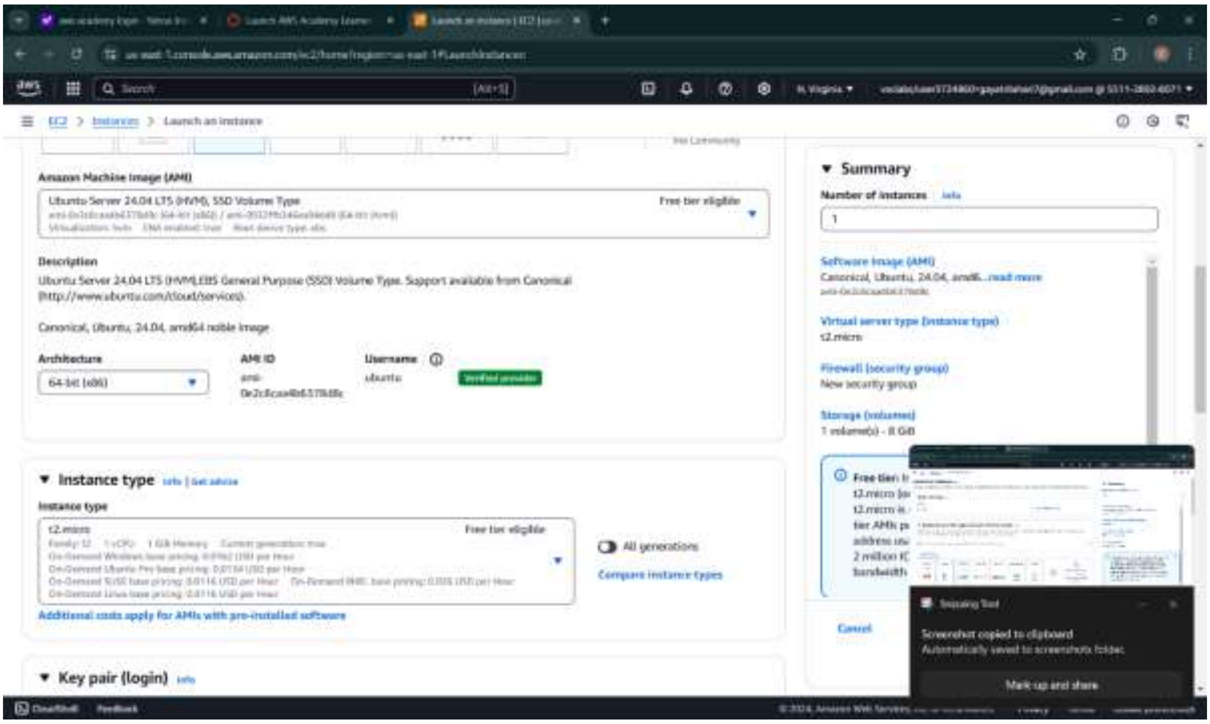
Instance type
Instance type: t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login)

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.

12. Next is Instance Type→ select default→ use Free tier only
    An instance type in AWS defines the hardware of a virtual server, specifying its CPU, memory, storage, and networking capacity to meet different application requirements.



Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture: 64-bit (x86)

AMI ID: ami-0e2c8caa4b6378d8c

Username: ubuntu

Instance type

t2.micro — Free tier eligible

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more

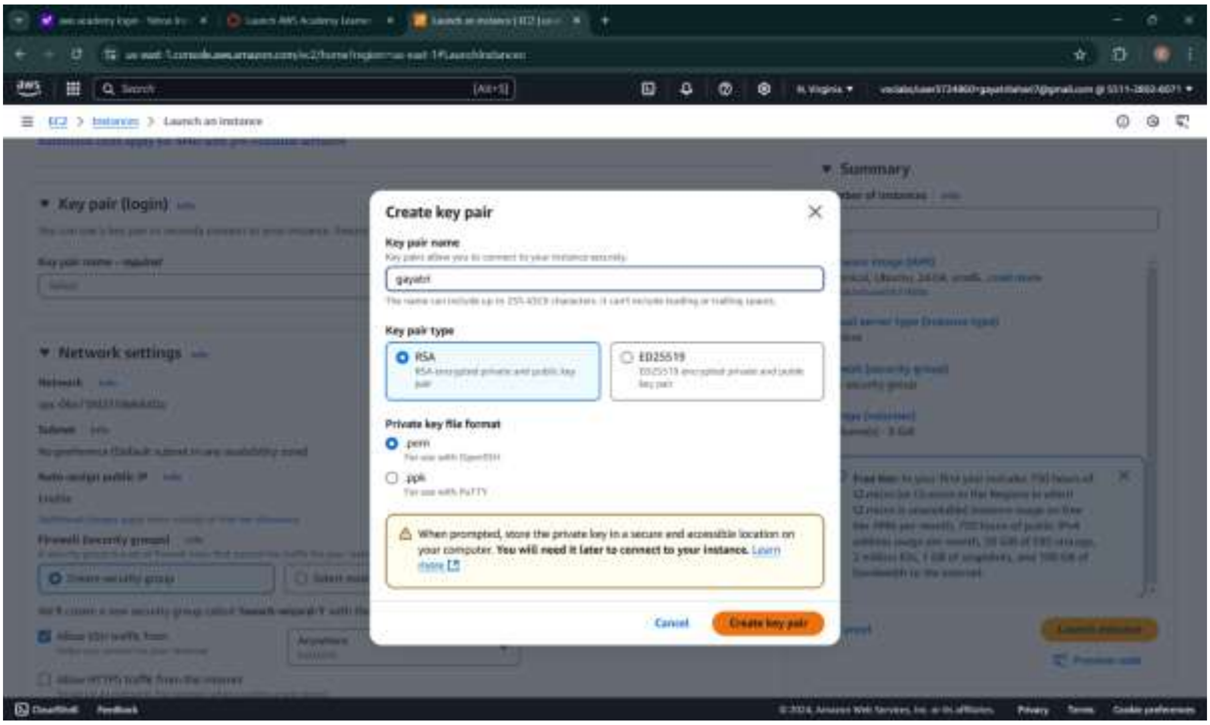
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login)
Cancel

13. Next is Key pair →Click on Create new key pair
 In AWS, a key pair is a set of security credentials, consisting of a public key used to encrypt information and a private key used to decrypt it, allowing secure access to instances and ensuring data protection.



Create key pair

Key pair name: gayatri

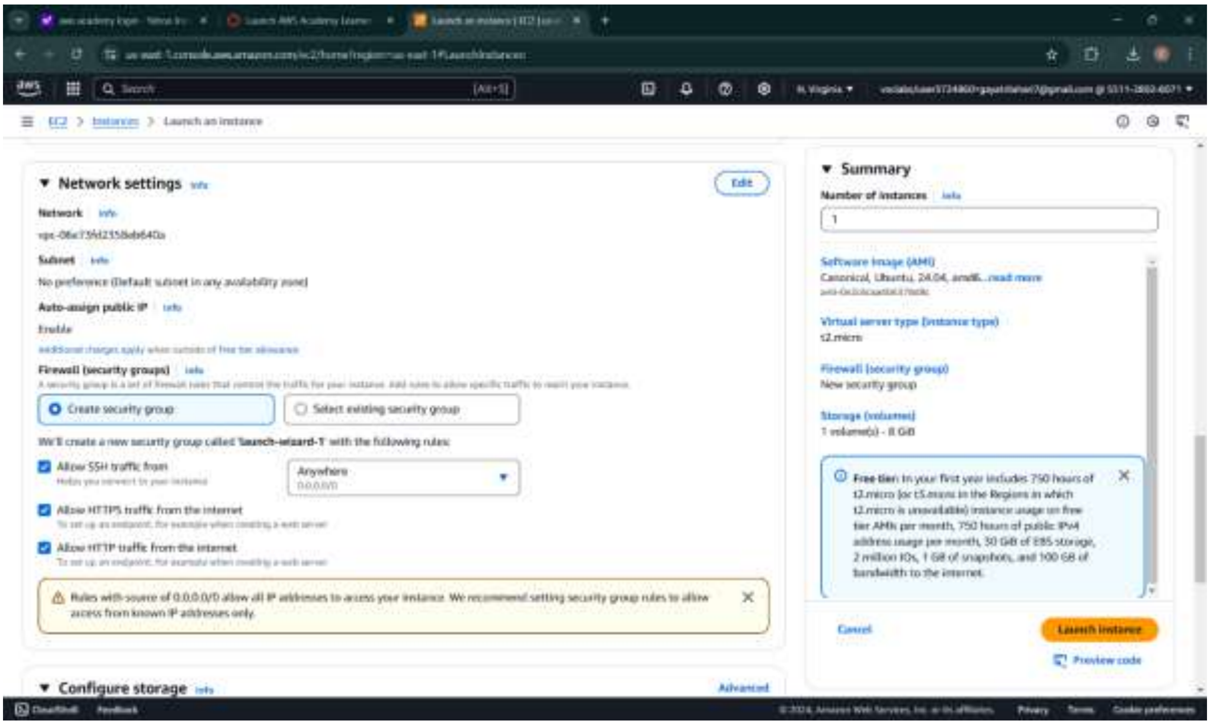
Key pair type: ● RSA ○ ED25519

Private key file format: ● .pem ○ .ppk

When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

14. After the keypair → next is Network settings

15. Check the two boxes: Allow HTTPS and Allow HTTP traffic
    Allowing HTTPS and HTTP traffic from the internet in AWS network settings enables secure and regular web communication, allowing users to access your web-based applications or websites while ensuring data confidentiality and accessibility



EC2 > Instances > Launch an instance

Network settings

Network: vpc-06a735d2358eb640a

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Firewall (security groups)

● Create security group ○ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☑ Allow SSH traffic from: Anywhere 0.0.0.0/0

☑ Allow HTTPS traffic from the internet

☑ Allow HTTP traffic from the internet

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more

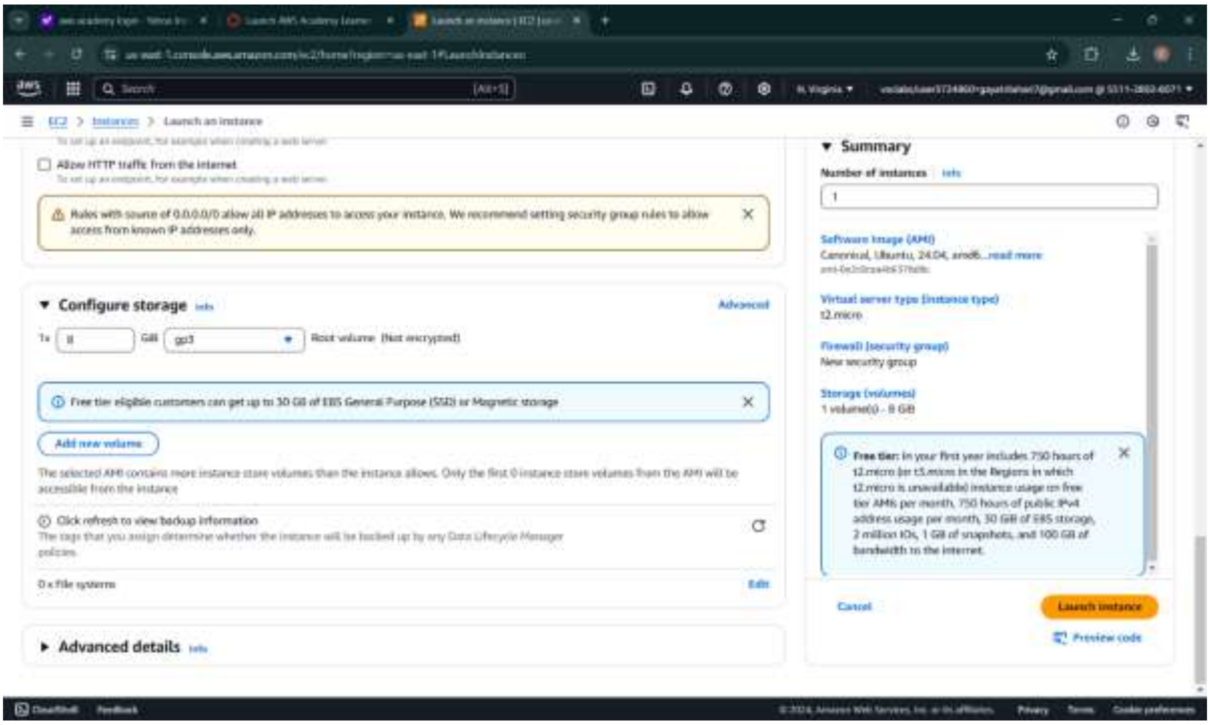
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

16. Next is the Configuration Storage: Default is 8GB, for Ubuntu we can extend up to 10GB
    Configuration storage in AWS refers to the secure and centralized management of settings, preferences, and parameters for various services and applications, ensuring consistent and organized control of your resources.



EC2 > Instances > Launch an instance

□ Allow HTTP traffic from the internet

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Advanced details

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more

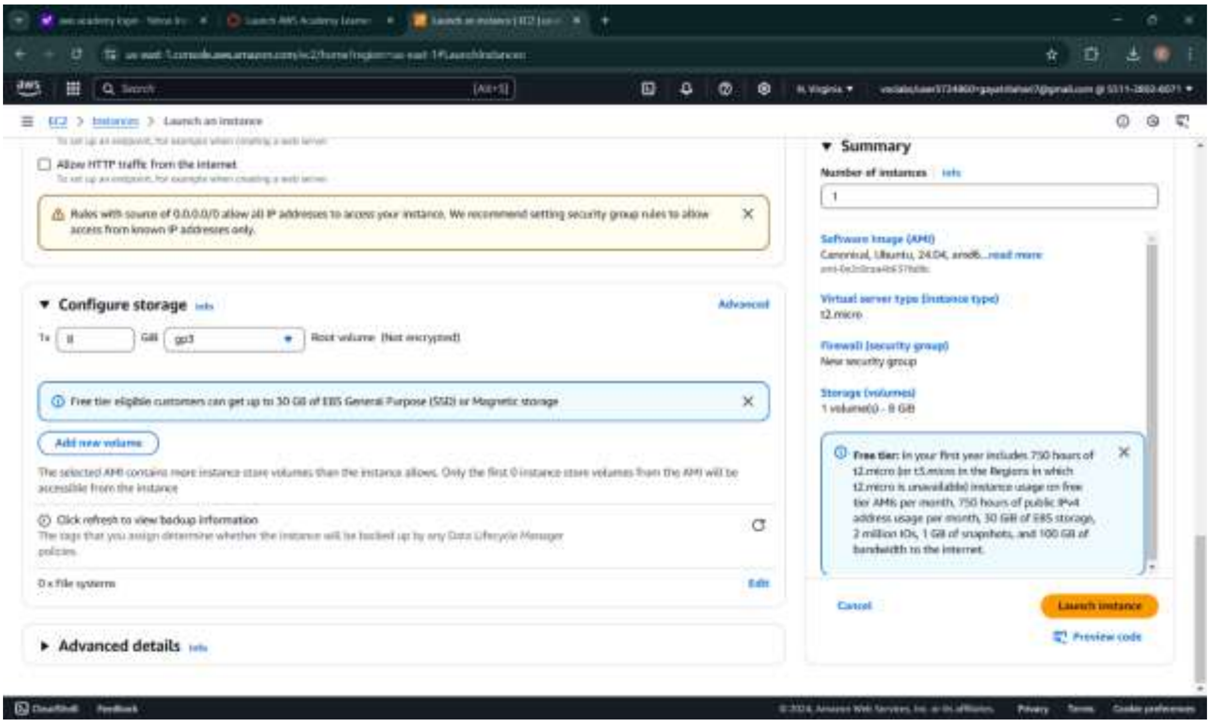
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

17. On the right side, we can see the summary of the instance created → now click on Launch instance in the bottom



EC2 > Instances > Launch an instance

☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Advanced details Info

Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more

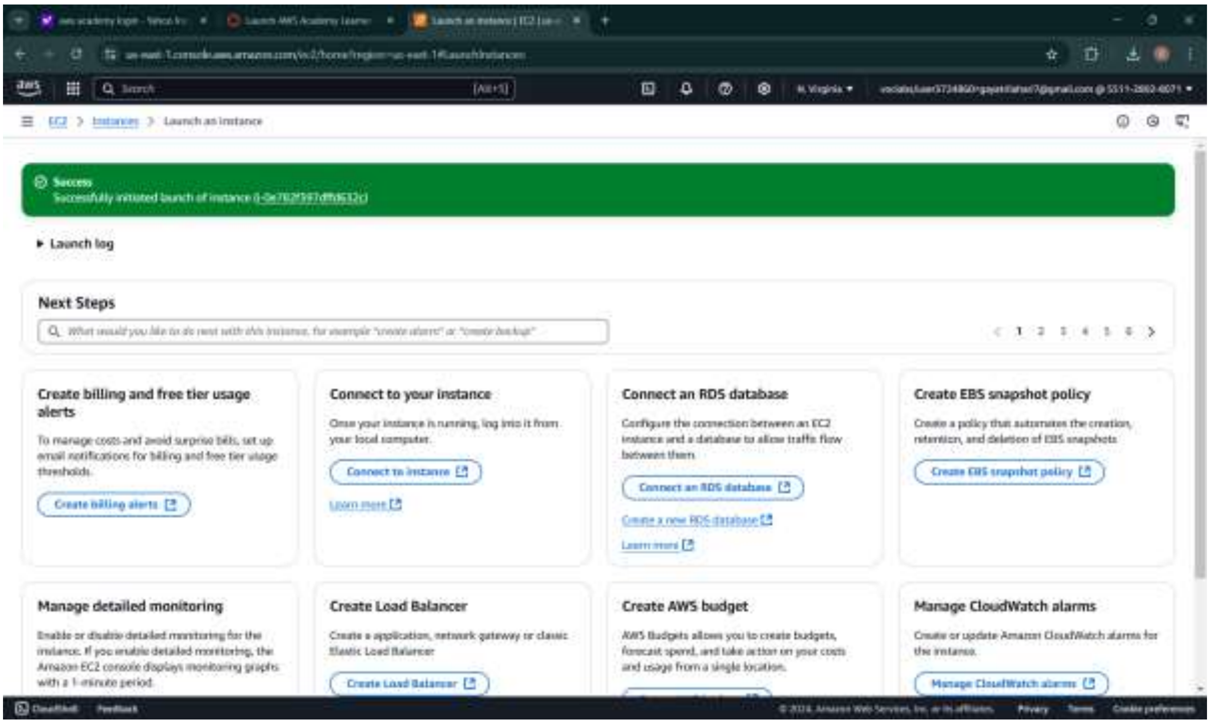
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

18. The instance is successfully launched→ we can either click on instances or scroll down



EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0e782f597dffd632c)

Launch log

Next Steps

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Connect to your instance
Once your instance is running, log into it from your local computer.

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots

Manage detailed monitoring
Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.

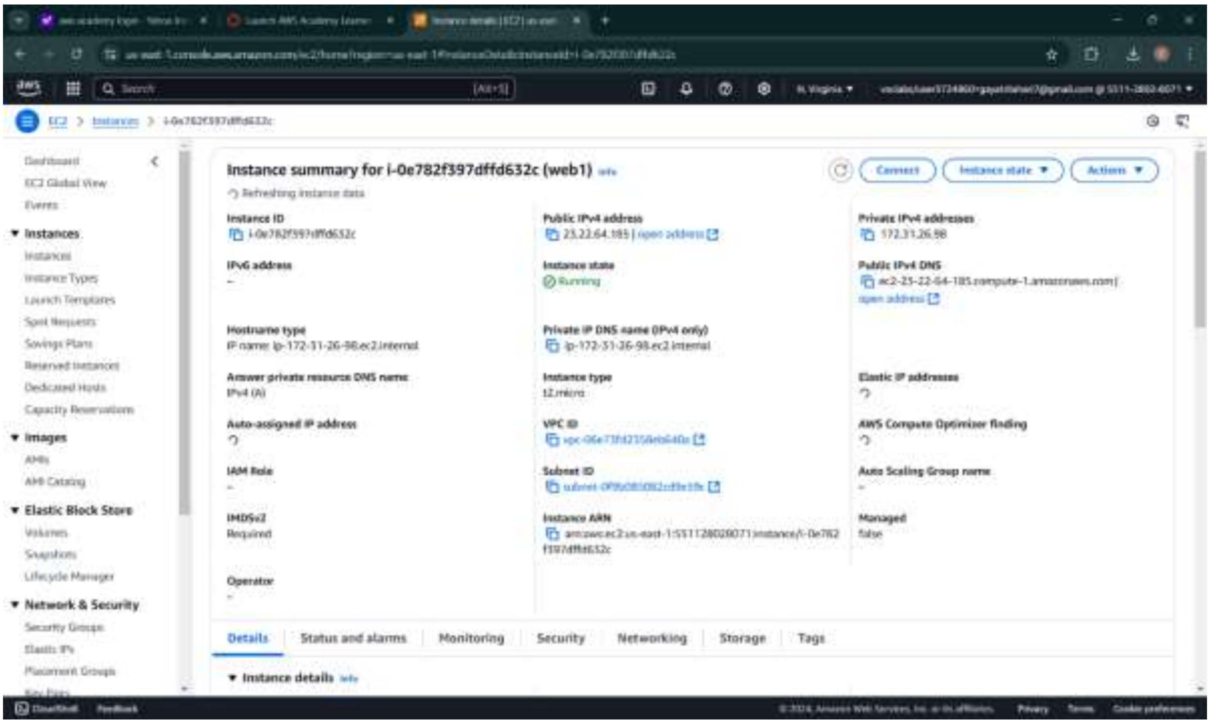
Create Load Balancer
Create a application, network gateway or classic Elastic Load Balancer

Create AWS budget
AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.

Manage CloudWatch alarms
Create or update Amazon CloudWatch alarms for the instance.

What would you like to do next with this instance, for example "create alarm" or "create backup"
1 2 3 4 5 6
Create billing alerts
Connect to instance
Learn more
Connect an RDS database
Create a new RDS database
Learn more
Create EBS snapshot policy
Create Load Balancer
Manage CloudWatch alarms

19. The instance will start running in few seconds→ check the instance state



Instance summary for i-0e782f397dffd632c (web1) Info

Refreshing instance data

Instance ID
i-0e782f397dffd632c

Public IPv4 address
23.22.64.185 | open address

Private IPv4 addresses
172.31.26.98

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-23-22-64-185.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-26-98.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-26-98.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses

Auto-assigned IP address

VPC ID
vpc-06e73fd2358eb640c

AWS Compute Optimizer finding

IAM Role
–

Subnet ID
subnet-0f9b085082cd9e1fe

Auto Scaling Group name
–

IMDSv2
Required

Instance ARN
arn:aws:ec2:us-east-1:551128028071:instance/i-0e782f397dffd632c

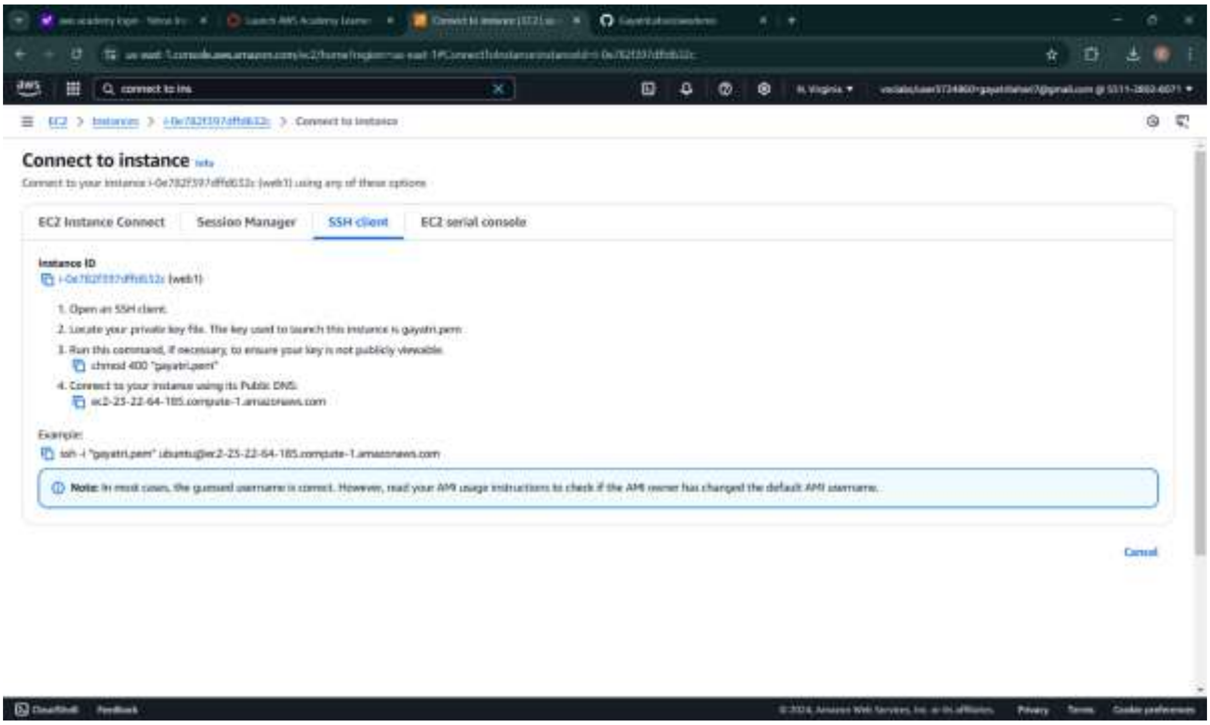
Managed
false

Operator
–

Instance details Info

© 2024, Amazon Web Services, Inc. or its affiliates.

20. We will connect using SSH client:
"Connecting to an instance in AWS means establishing a secure and remote access to your virtual server, allowing you to manage and interact with it from your local computer."
21. The command is copied.



Connect to instance Info

Connect to your instance i-0e782f397dffd632c (web1) using any of these options

Instance ID
i-0e782f397dffd632c (web1)

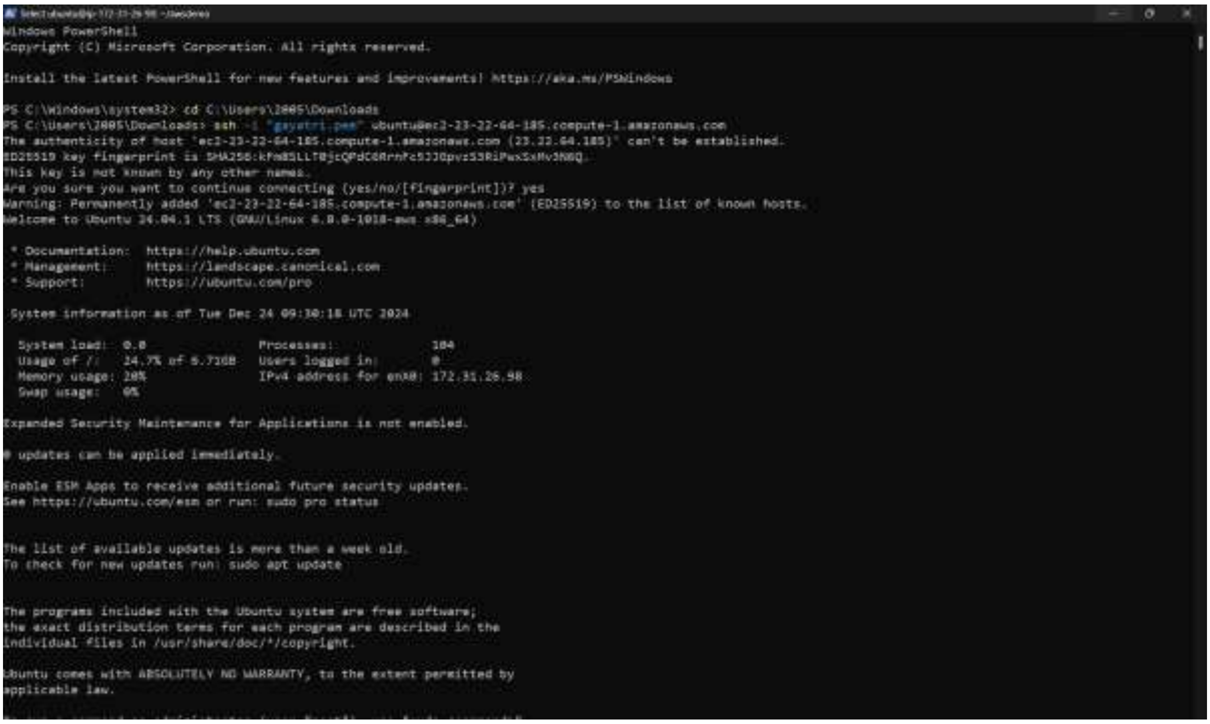
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is gayatri.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
```
chmod 400 "gayatri.pem"
```
4. Connect to your instance using its Public DNS:
```
ec2-23-22-64-185.compute-1.amazonaws.com
```

Example:
```
ssh -i "gayatri.pem" ubuntu@ec2-23-22-64-185.compute-1.amazonaws.com
```

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

© 2024, Amazon Web Services, Inc. or its affiliates.

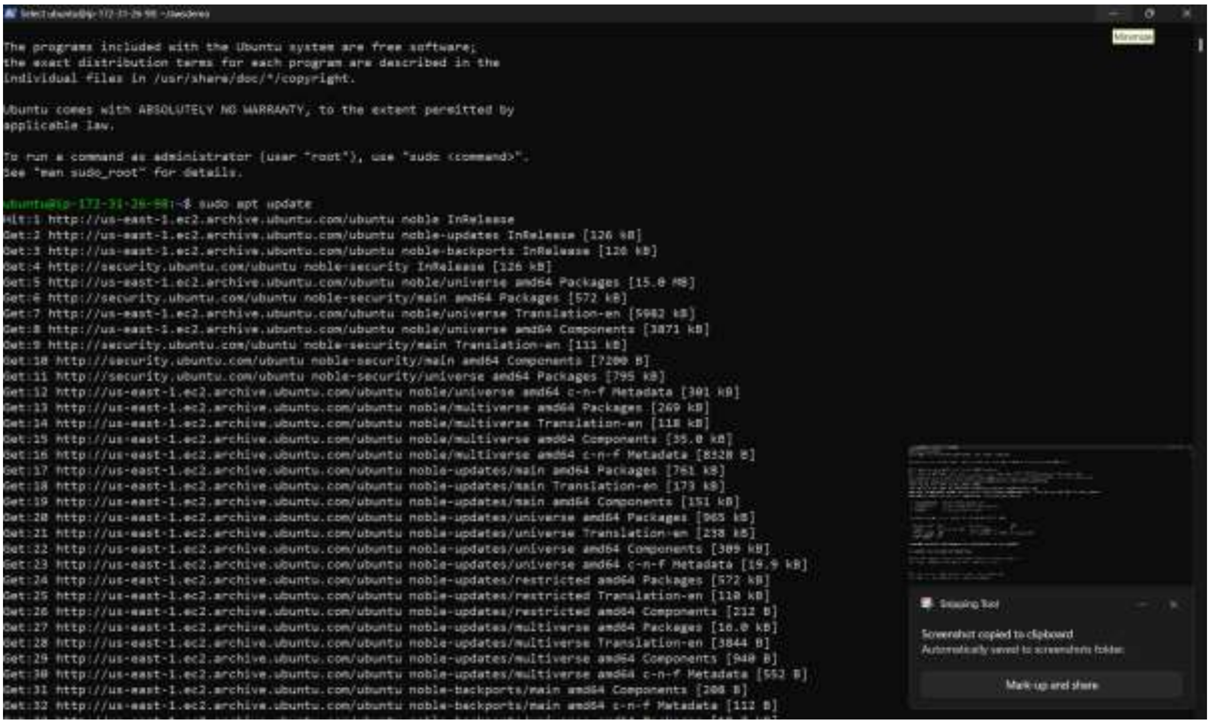
22. Now open PowerShell in Administrator mode
23. Paste the example command copied from the AWS instance



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> cd C:\Users\2805\Downloads
PS C:\Users\2805\Downloads> ssh -i "gayatri.pem" ubuntu@ec2-23-22-64-185.compute-1.amazonaws.com
The authenticity of host 'ec2-23-22-64-185.compute-1.amazonaws.com (23.22.64.185)' can't be established.
ED25519 key fingerprint is SHA256:kFm85LLT0jcQPdC6RrnFc5JJ0pvz53RiPwxSxMv3N8Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-23-22-64-185.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

 System information as of Tue Dec 24 09:30:18 UTC 2024

  System load:  0.0               Processes:             104
  Usage of /:   24.7% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.26.98
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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.
```

Now the Ubuntu is successfully launched into our local system

24. The highlighted is the private IP address to access the instance.   GO back to the PowerShell and Update Ubuntu using the highlighted command: 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.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-26-98:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [572 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [111 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7200 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [795 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [761 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [173 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [965 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [238 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [309 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [572 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [110 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [16.0 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3844 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
```

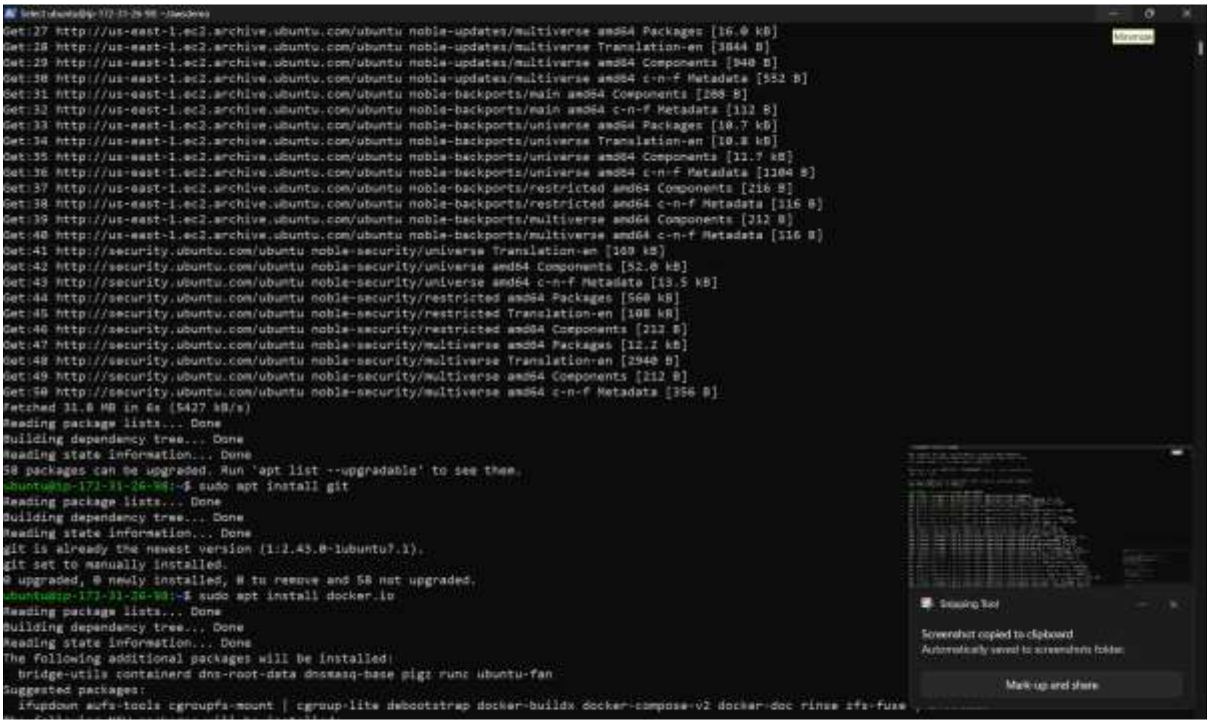
Snipping Tool

Screenshot copied to clipboard

Automatically saved to screenshots folder.

25. After updating Ubuntu install Docker using the command:
    sudo apt-get install docker.io
    Click Y to continue

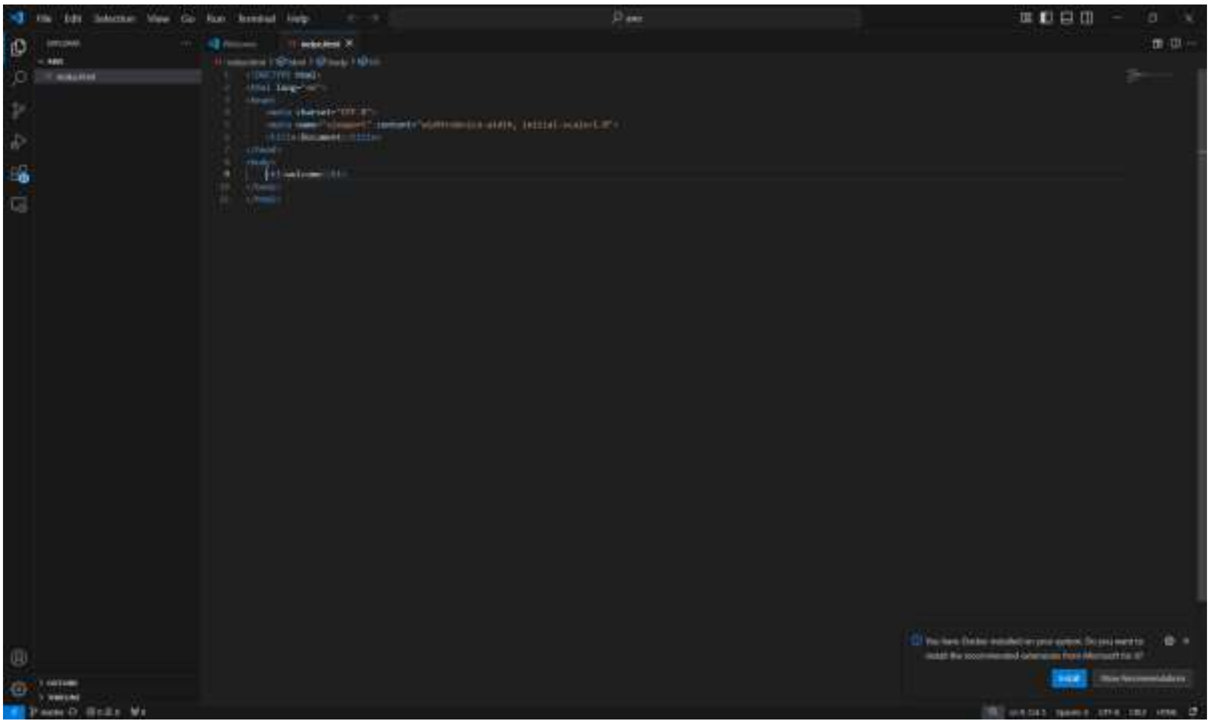
26. Install git using the following command:
    sudo apt install git



```
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [16.0 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3844 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.7 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.7 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [169 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [13.5 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [560 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [108 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [12.2 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2940 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 31.8 MB in 6s (5427 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
58 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-26-98:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.1).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-26-98:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse
```

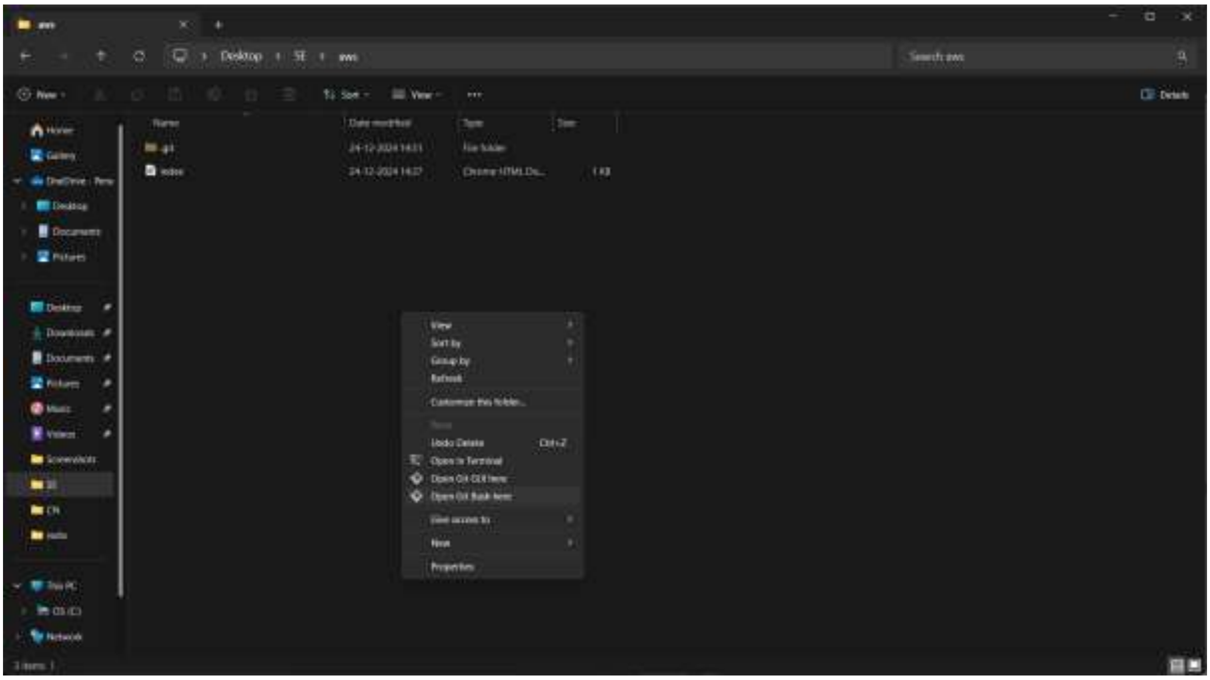
Screenshot copied to clipboard
Automatically saved to screenshots folder.

27. Open any editor like notepad and write a simple web-application using html and save the file as index.html only

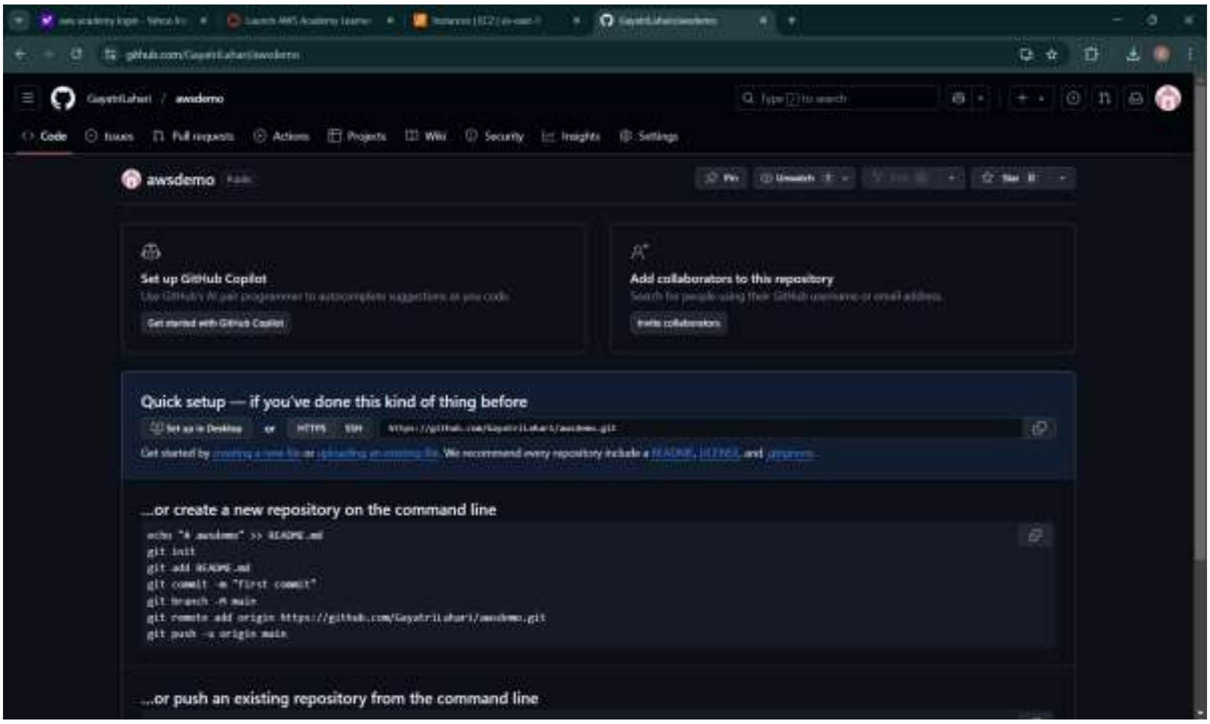


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>welcome</h1>
</body>
</html>
```

28. Open the folder where your index.html file is located → right click and open GitBash here



29. Type the following Git commands:
    git init
    git add .
    git commit
30. Create a repository in GitHub



Quick setup — if you've done this kind of thing before

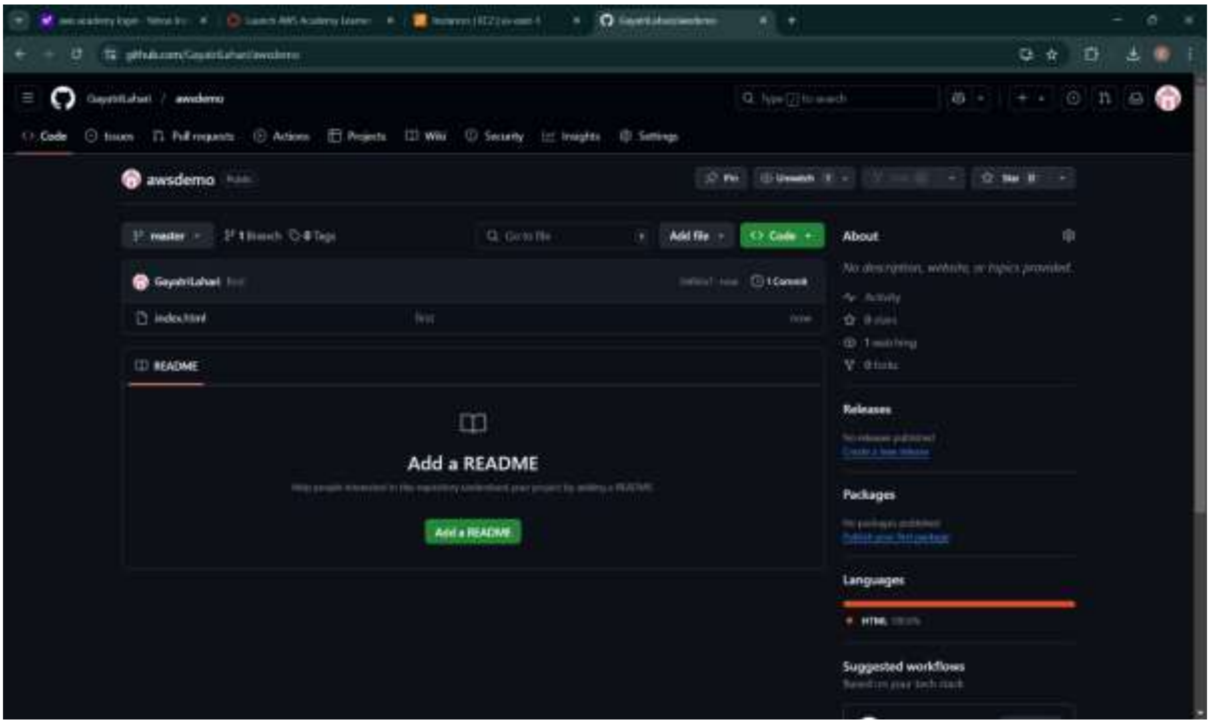
Set up in Desktop or HTTPS SSH https://github.com/GayatriLahari/awsdemo.git

…or create a new repository on the command line

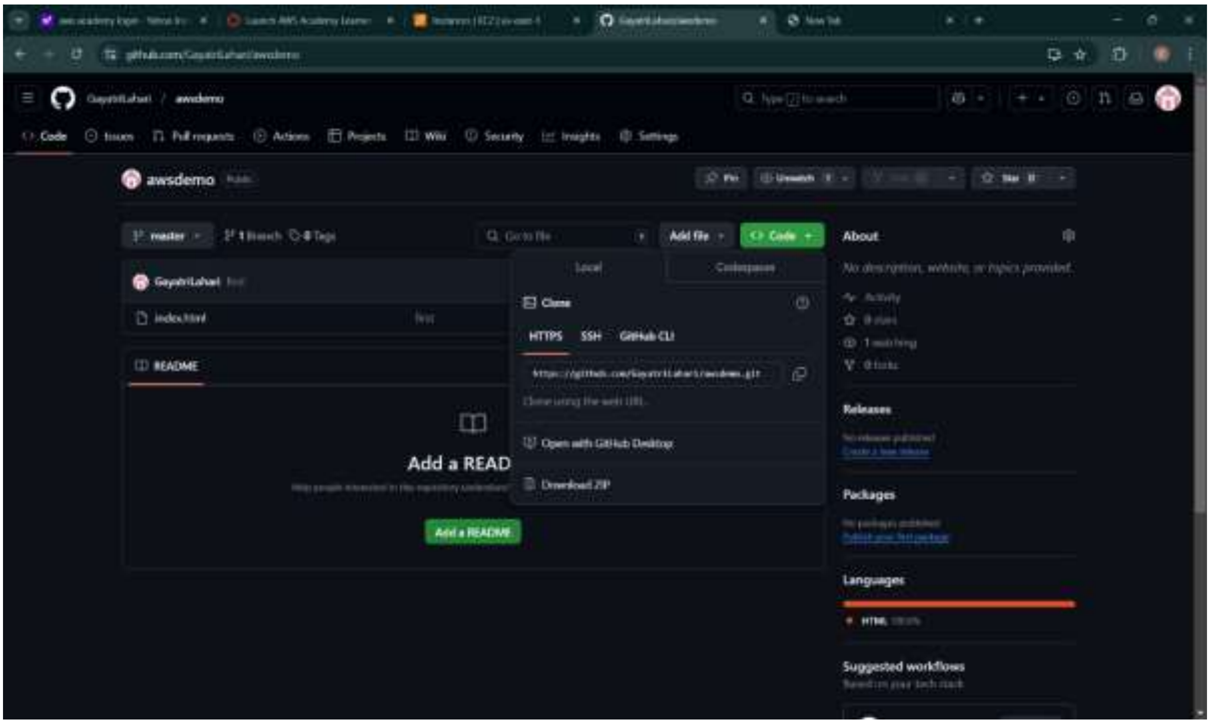
```
echo "# awsdemo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/GayatriLahari/awsdemo.git
git push -u origin main
```

…or push an existing repository from the command line

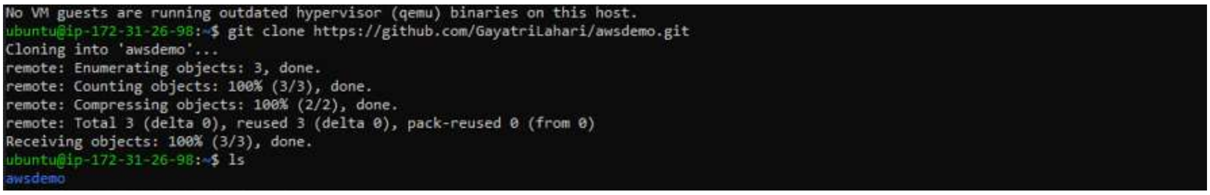
31. Add the remote repository as shown:
    git remote add origin <paste the HTTPS link of GitHub>.push
32. Refresh the GitHub page to see the pushed file.



33. Now copy the HTTPS URL as shown

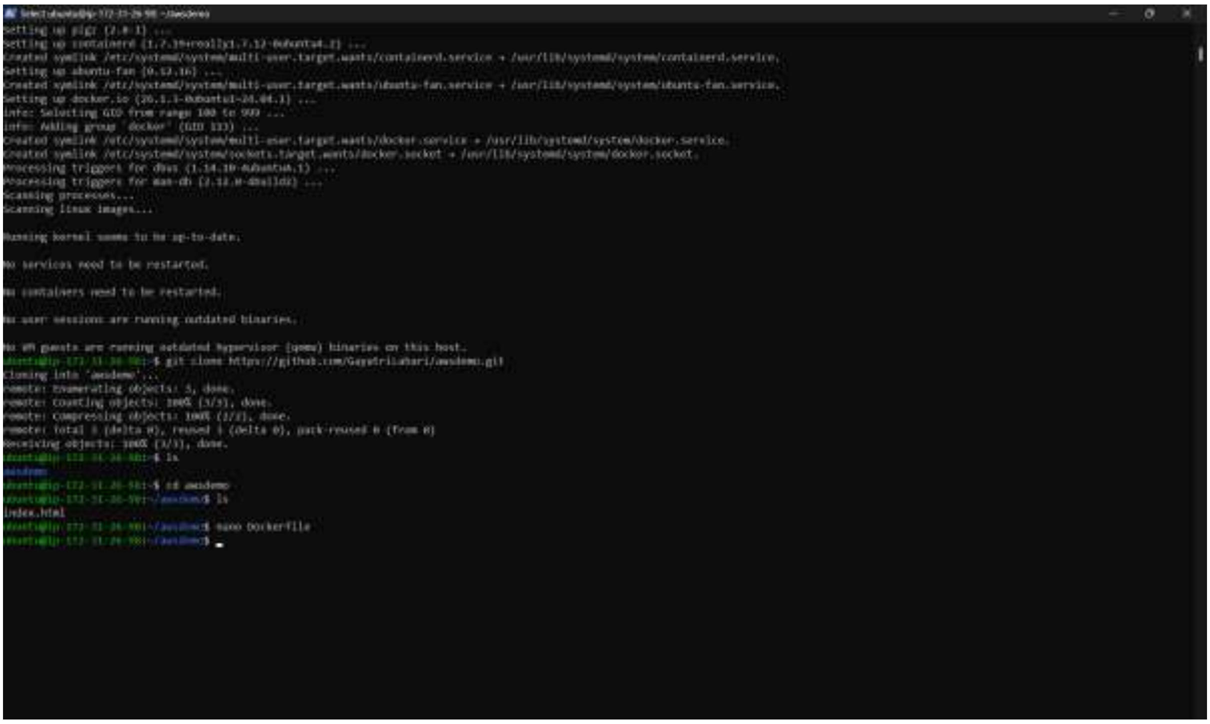


34. Go back to PowerShell and paste the URL as shown

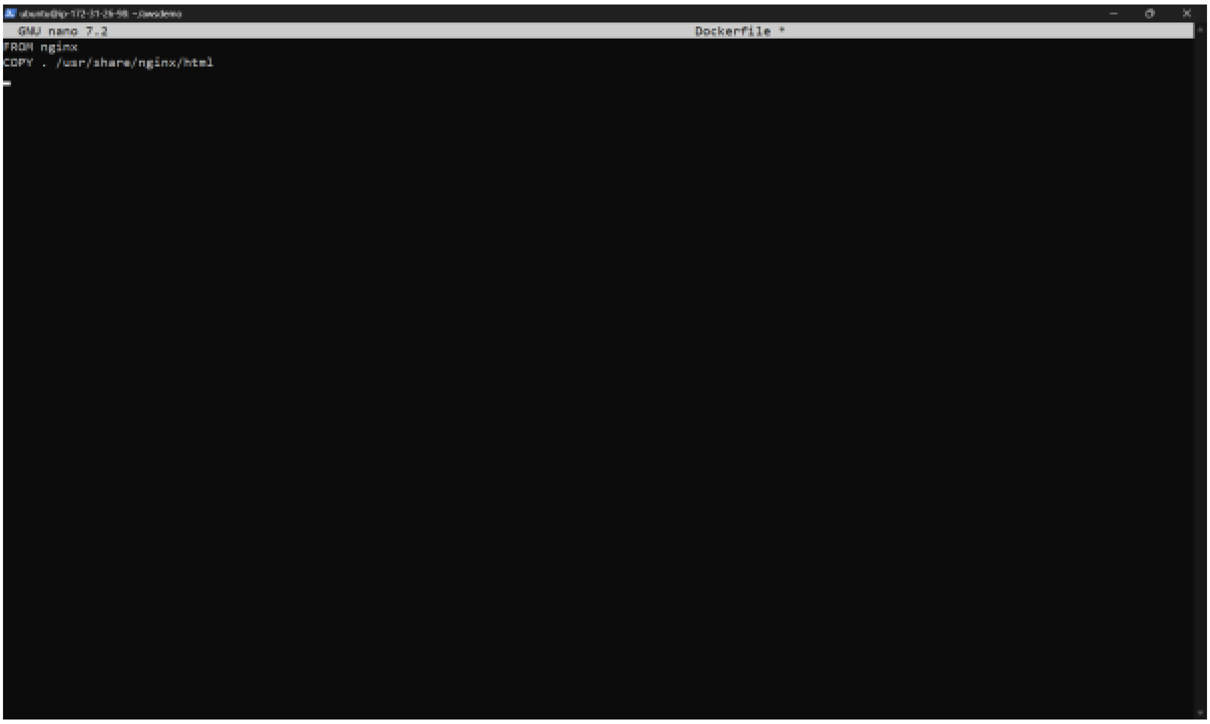


```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-26-98:~$ git clone https://github.com/GayatriLahari/awsdemo.git
Cloning into 'awsdemo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-26-98:~$ ls
awsdemo
```

35. Once cloning is completed→ Move into the folder as shown. Inside this folder→ create a Dockerfile as shown

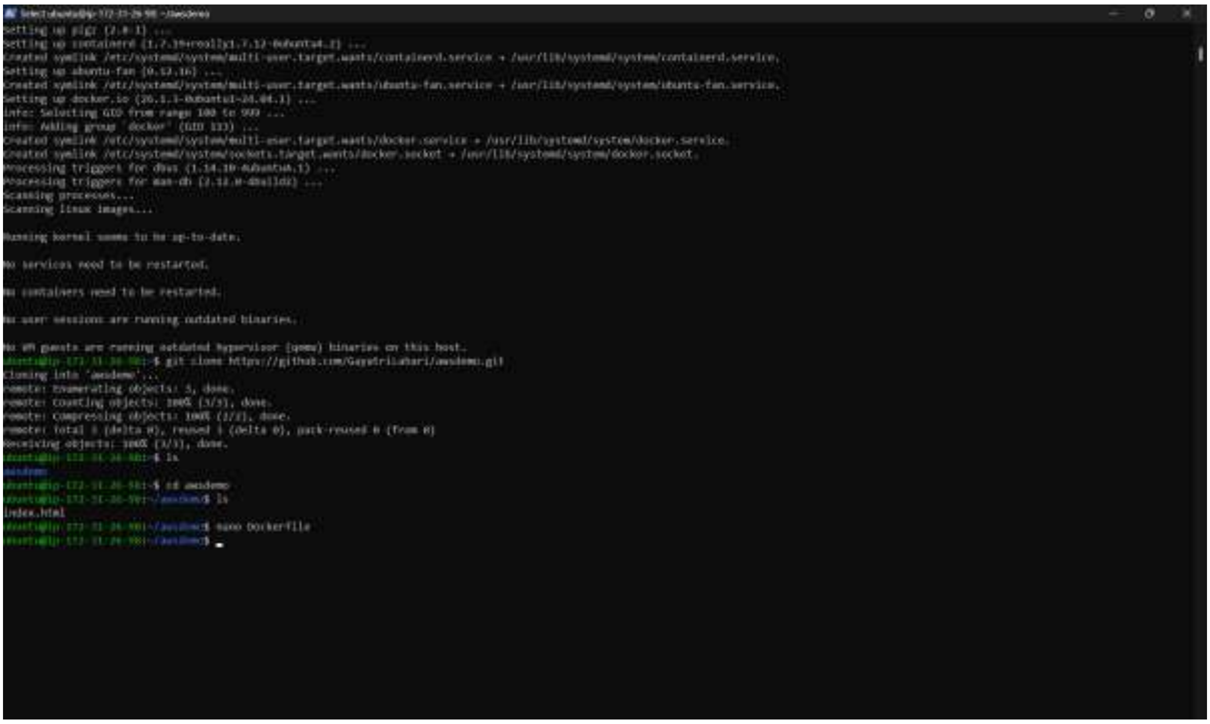


```
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.19+really1.7.12-0ubuntu4.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (26.1.3-0ubuntu1~24.04.1) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group `docker' (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-26-98:~$ git clone https://github.com/Gayatrilahari/awsdemo.git
Cloning into 'awsdemo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-26-98:~$ ls
awsdemo
ubuntu@ip-172-31-26-98:~$ cd awsdemo
ubuntu@ip-172-31-26-98:~/awsdemo$ ls
index.html
ubuntu@ip-172-31-26-98:~/awsdemo$ nano Dockerfile
ubuntu@ip-172-31-26-98:~/awsdemo$
```



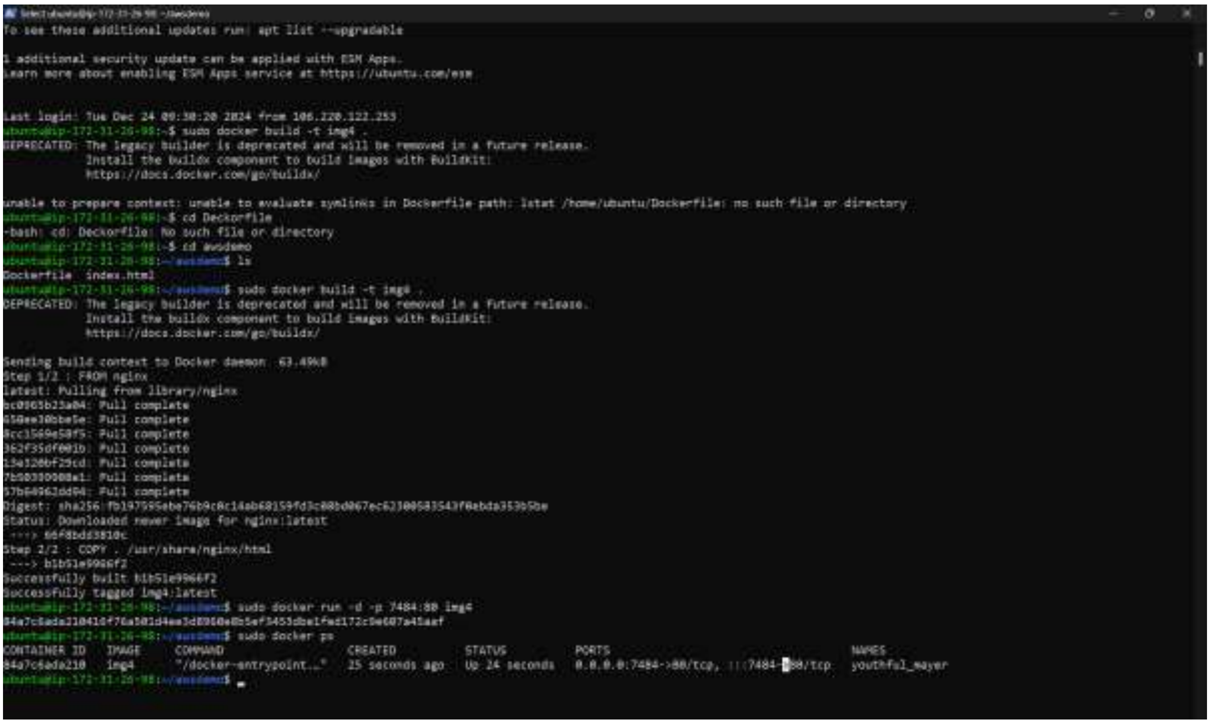
```
  GNU nano 7.2                                         Dockerfile *
FROM nginx
COPY . /usr/share/nginx/html
```

36. Clone the repostiroy.



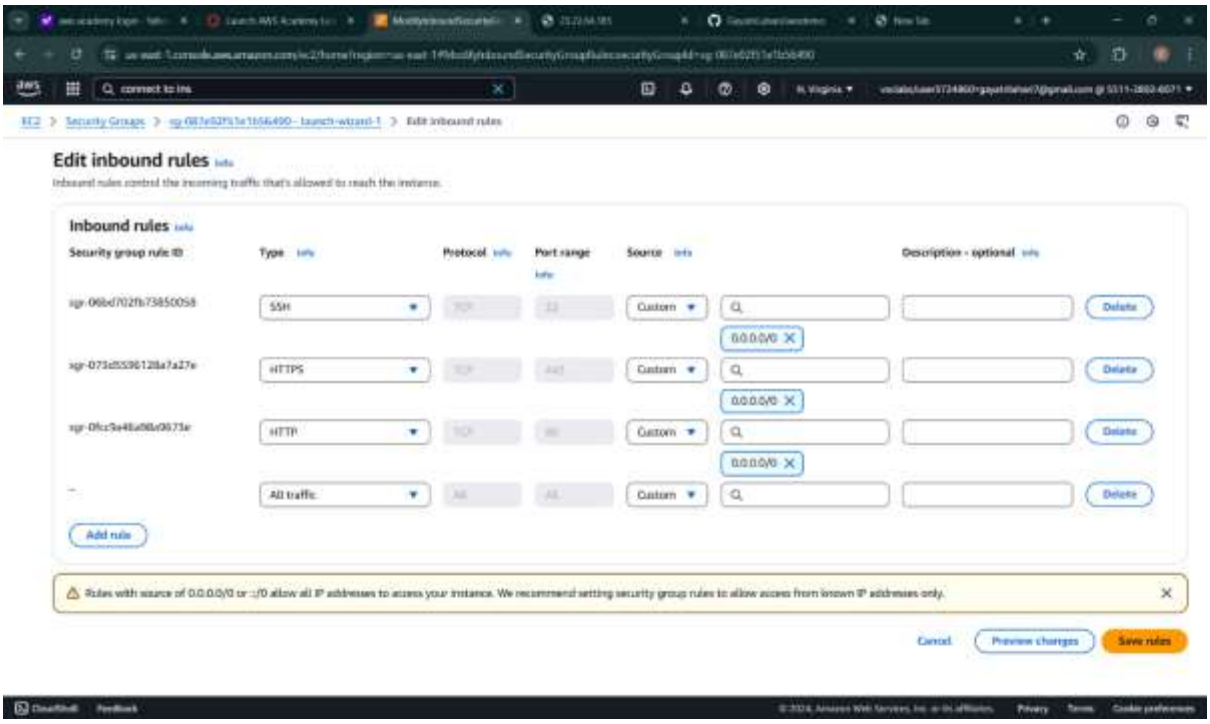
```
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.19+really1.7.12-0ubuntu4.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (26.1.3-0ubuntu1~24.04.1) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group `docker' (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-26-98:~$ git clone https://github.com/GayatriLahari/awsdemo.git
Cloning into 'awsdemo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-26-98:~$ ls
awsdemo
ubuntu@ip-172-31-26-98:~$ cd awsdemo
ubuntu@ip-172-31-26-98:~/awsdemo$ ls
index.html
ubuntu@ip-172-31-26-98:~/awsdemo$ nano Dockerfile
ubuntu@ip-172-31-26-98:~/awsdemo$ 
```

37. Build a docker image.
38. Next→ Run the Docker container as shown:



```
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Tue Dec 24 09:38:20 2024 from 106.220.122.253
ubuntu@ip-172-31-26-98:~$ sudo docker build -t img4 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/ubuntu/Dockerfile: no such file or directory
ubuntu@ip-172-31-26-98:~$ cd Deckorfile
-bash: cd: Deckorfile: No such file or directory
ubuntu@ip-172-31-26-98:~$ cd awsdemo
ubuntu@ip-172-31-26-98:~/awsdemo$ ls
Dockerfile  index.html
ubuntu@ip-172-31-26-98:~/awsdemo$ sudo docker build -t img4 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  63.49kB
Step 1/2 : FROM nginx
latest: Pulling from library/nginx
bc0965b23a04: Pull complete
650ee30bbe5e: Pull complete
8cc1569e58f5: Pull complete
362f35df001b: Pull complete
13e320bf29cd: Pull complete
7b50399908e1: Pull complete
57b64962dd94: Pull complete
Digest: sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be
Status: Downloaded newer image for nginx:latest
 ---> 66f8bdd3810c
Step 2/2 : COPY . /usr/share/nginx/html
 ---> b1b51e9966f2
Successfully built b1b51e9966f2
Successfully tagged img4:latest
ubuntu@ip-172-31-26-98:~/awsdemo$ sudo docker run -d -p 7484:80 img4
84a7c6ada210410f76a501d4ee3d8960e0b5ef3453dbe1fed172c9e607a45aef
ubuntu@ip-172-31-26-98:~/awsdemo$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                   NAMES
84a7c6ada210   img4      "/docker-entrypoint.…"   25 seconds ago   Up 24 seconds   0.0.0.0:7484->80/tcp, :::7484->80/tcp   youthful_mayer
ubuntu@ip-172-31-26-98:~/awsdemo$ 
```

39. Now, we can access the web-application using the Public IP address as shown→ Copy the address and paste in any browser



EC2 > Security Groups > sg-087e02f51e1b56490 - launch-wizard-1 > Edit inbound rules

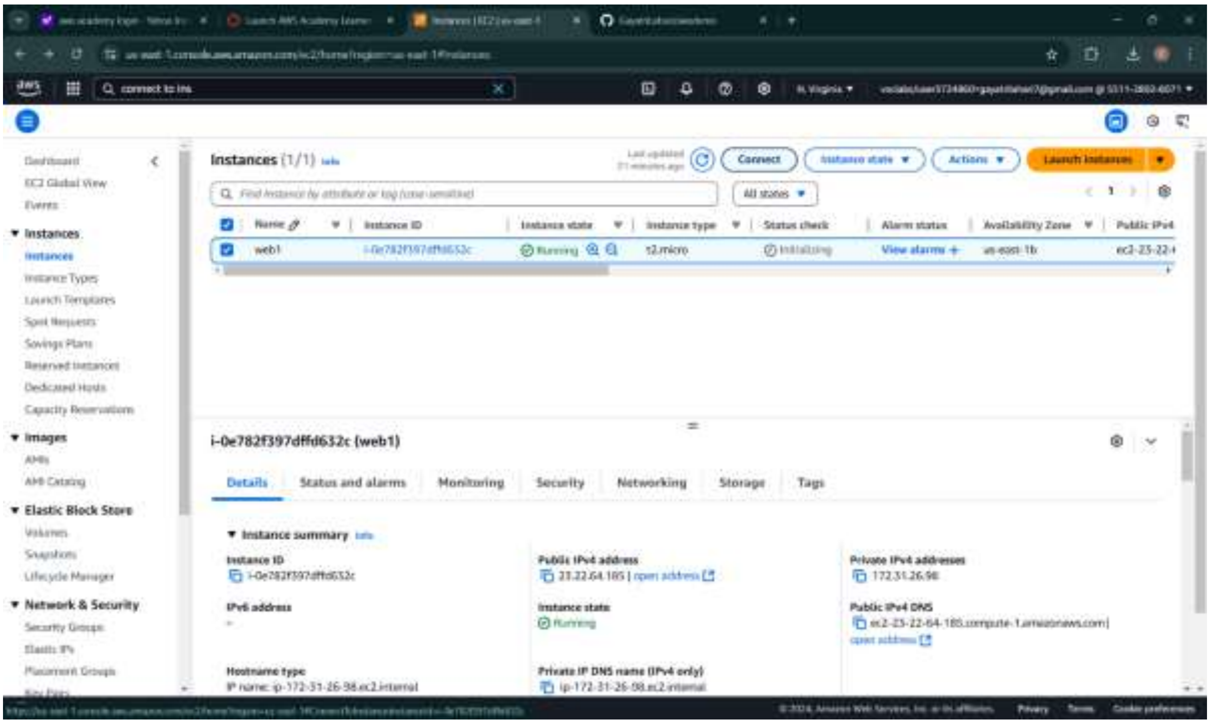
Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-06bd702fb73850058	SSH	TCP	22	Custom	0.0.0.0/0	
sgr-073d5536128a7a27e	HTTPS	TCP	443	Custom	0.0.0.0/0	
sgr-0fcc5e48a08d0673e	HTTP	TCP	80	Custom	0.0.0.0/0	
-	All traffic	All	All	Custom		

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.



Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
web1	i-0e782f397dffd632c	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-23-22-4

i-0e782f397dffd632c (web1)

Instance summary

Instance ID: i-0e782f397dffd632c

Public IPv4 address: 23.22.64.185 | open address

Private IPv4 addresses: 172.31.26.98

IPv6 address: -

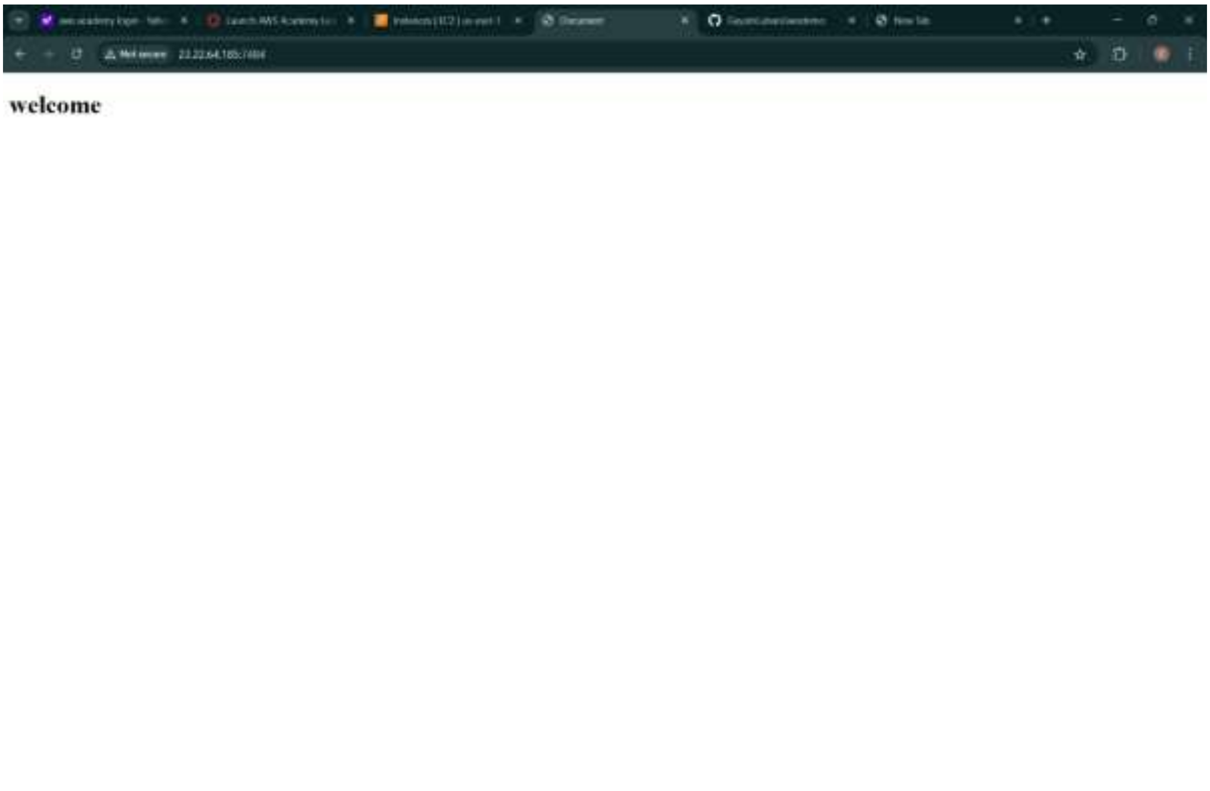
Instance state: Running

Public IPv4 DNS: ec2-23-22-64-185.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-26-98.ec2.internal

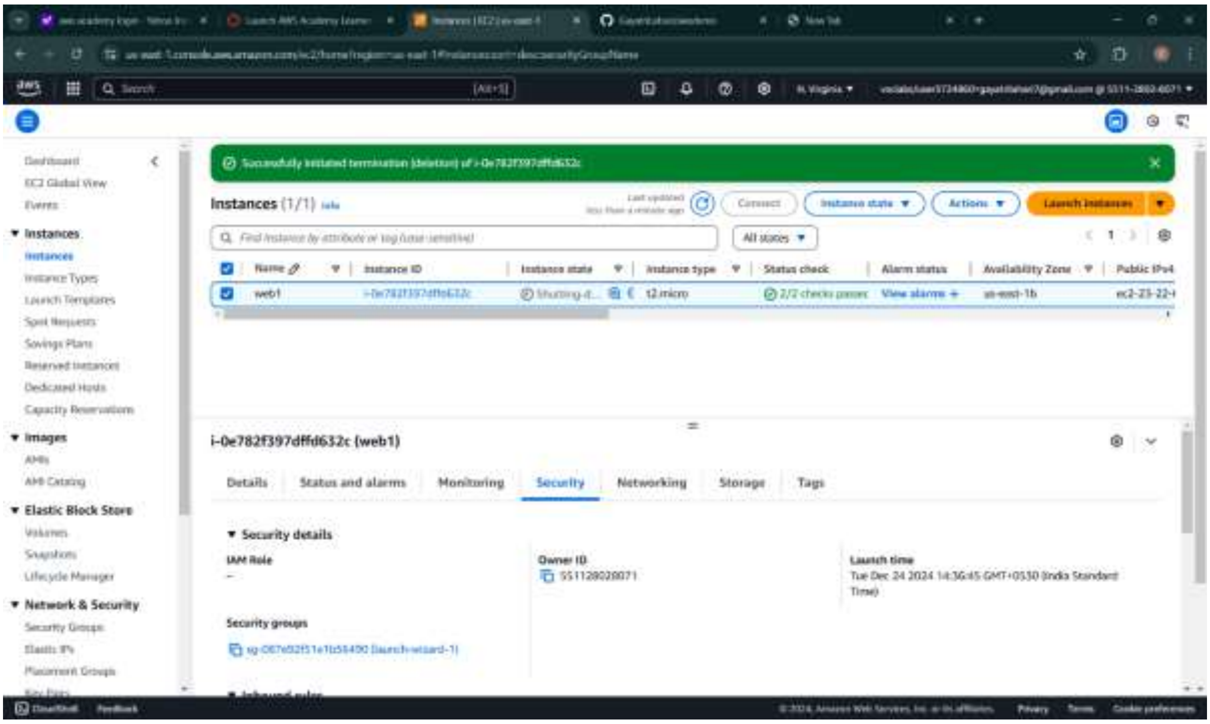
Private IP DNS name (IPv4 only): ip-172-31-26-98.ec2.internal

40. We can now see the deployed web application



welcome

41. Go back to the AWS → click on the instance checkbox. Go to Instance State and click on Terminate Instance. Click on terminate to delete the instance. The instance is successfully terminated.



Successfully initiated termination (deletion) of i-0e782f397dffd632c

Instances (1/1)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	web1	i-0e782f397dffd632c	Shutting-d...	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-23-22-(

i-0e782f397dffd632c (web1)

Security details

IAM Role
–

Owner ID
551128028071

Launch time
Tue Dec 24 2024 14:36:45 GMT+0530 (India Standard Time)

Security groups
sg-067e02f51e1b55490 (launch-wizard-1)