

Week 12: Creation of virtual machine for Ubuntu OS and Deploying the web application

1. Creation of virtual machine
 2. Deploying the web application
 3. Accessing it publicly
-

Deploying an application into cloud

Steps for Deploying application into the cloud

- I. Create application and Push into github
- II. Create the virtual machine and connect to it.
- III. Clone the application from github, Write the Dockerfile
- IV. Create the image
- V. Run the image and access it public ip of virtual machine

I. Create Maven-web-java project in eclipse & push into github

II. Create the virtual machine (EC2--instance) in aws and connect to

Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster.

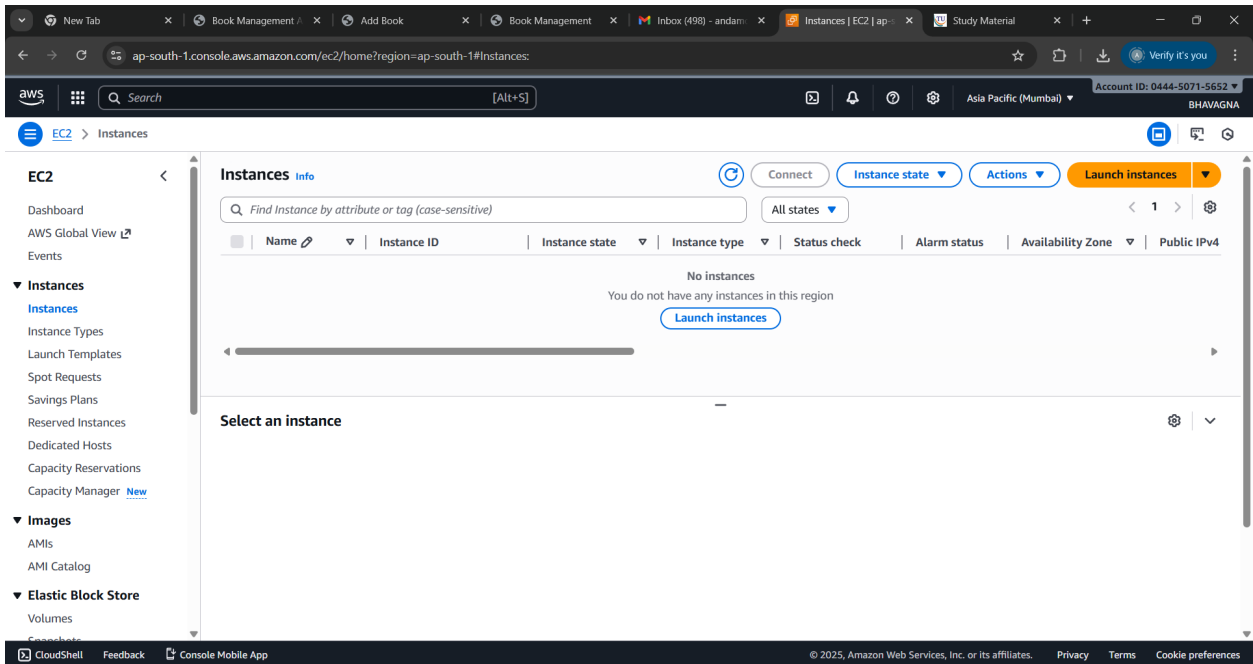
Ex : **Launch ubuntu instnace**

Step 1: Login to AWS /canvas account

Step 2: Services -- EC2

Step 3: Choose region which is near ?

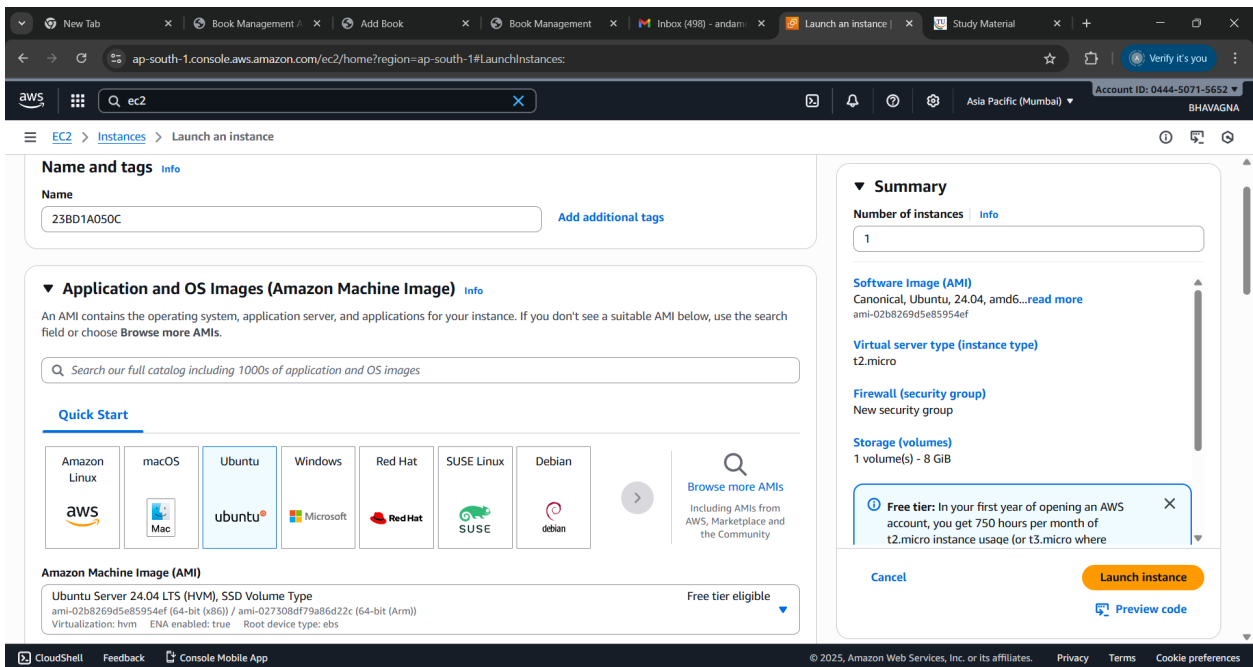
Services -- EC2 --- Launch Instance



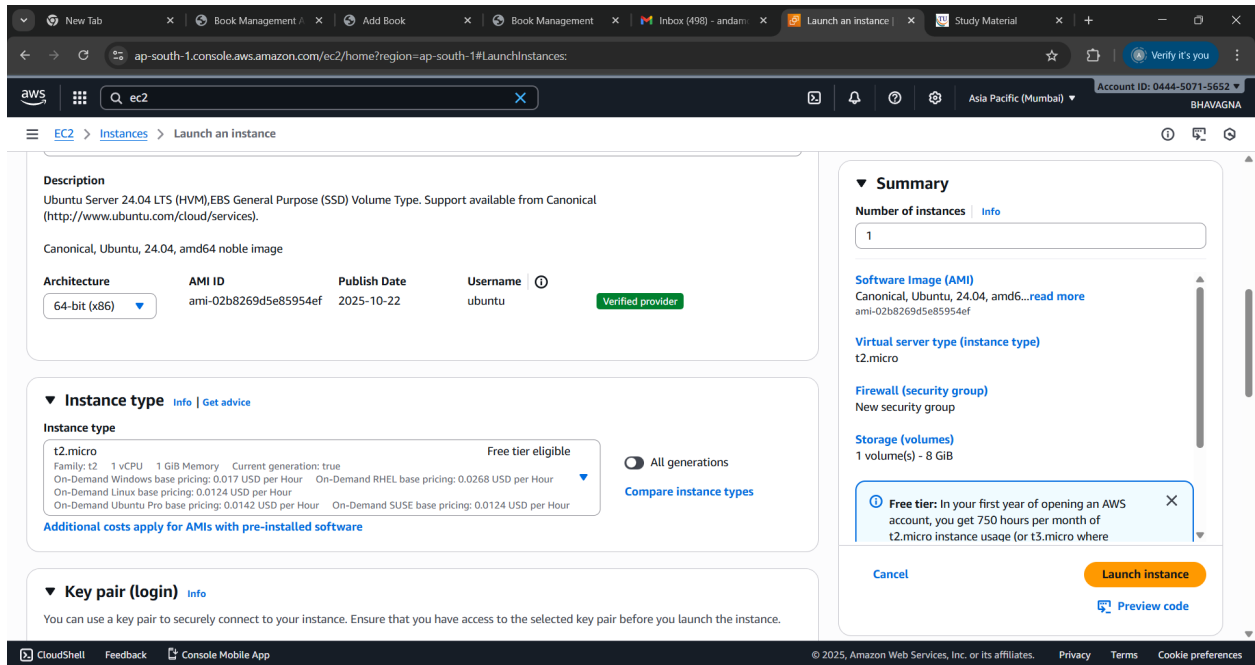
Stage 1 --Name (Giving name to the machine) ubuntu

Stage 2 -- Select AMI (Note: Select free tier eligible) ubuntu server

Stage 3 -- Architecture as 64-bit



Stage 4 -- Instance type ---- t2.micro(default 1 CPU,1 GB RAM)

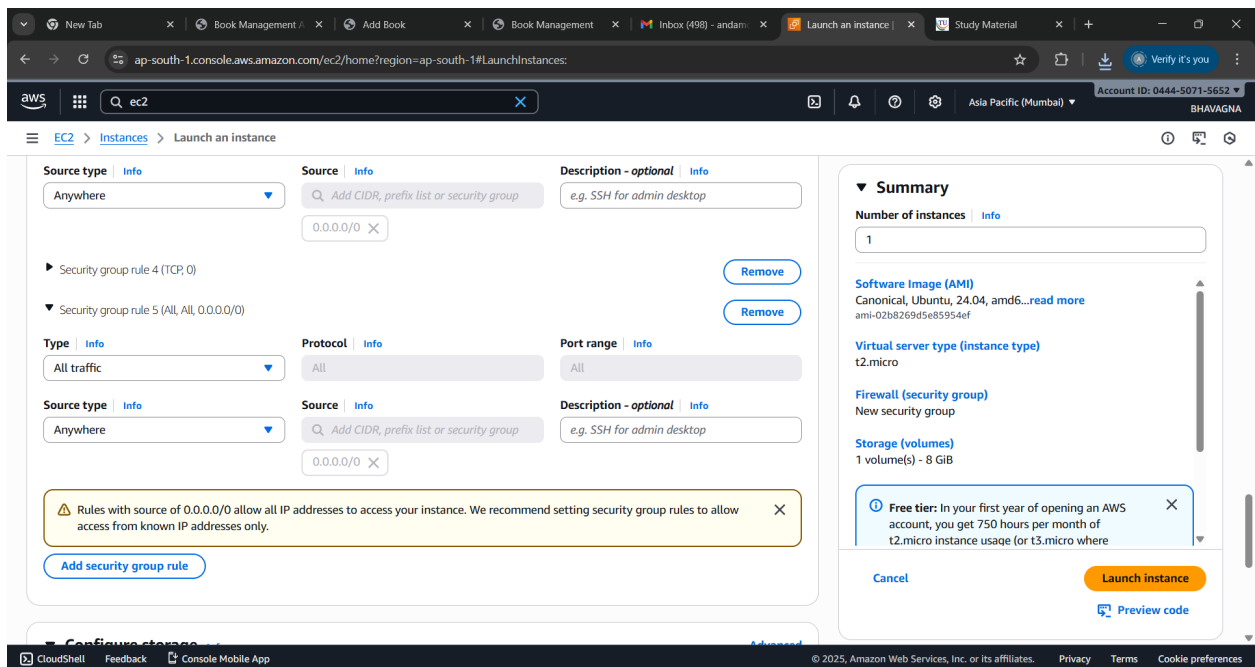
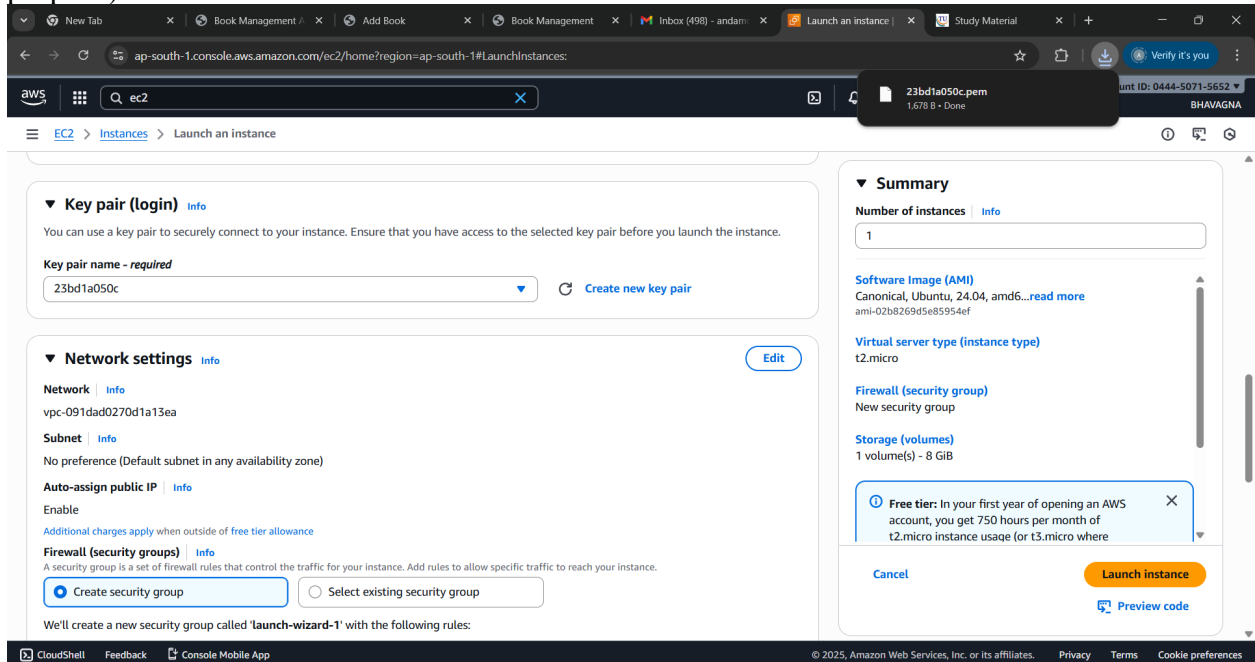


Stage 5 -- Create a new keypair---a keypair will downloaded with extension .pem

Store key in folder AWS

Stage 6 -- Network Setting ----Create Security group -- (It deals with ports)

(Note for understanding We have 0 to 65535 ports. Every port is dedicated to special purpose)



Do this step : HERE select http and https

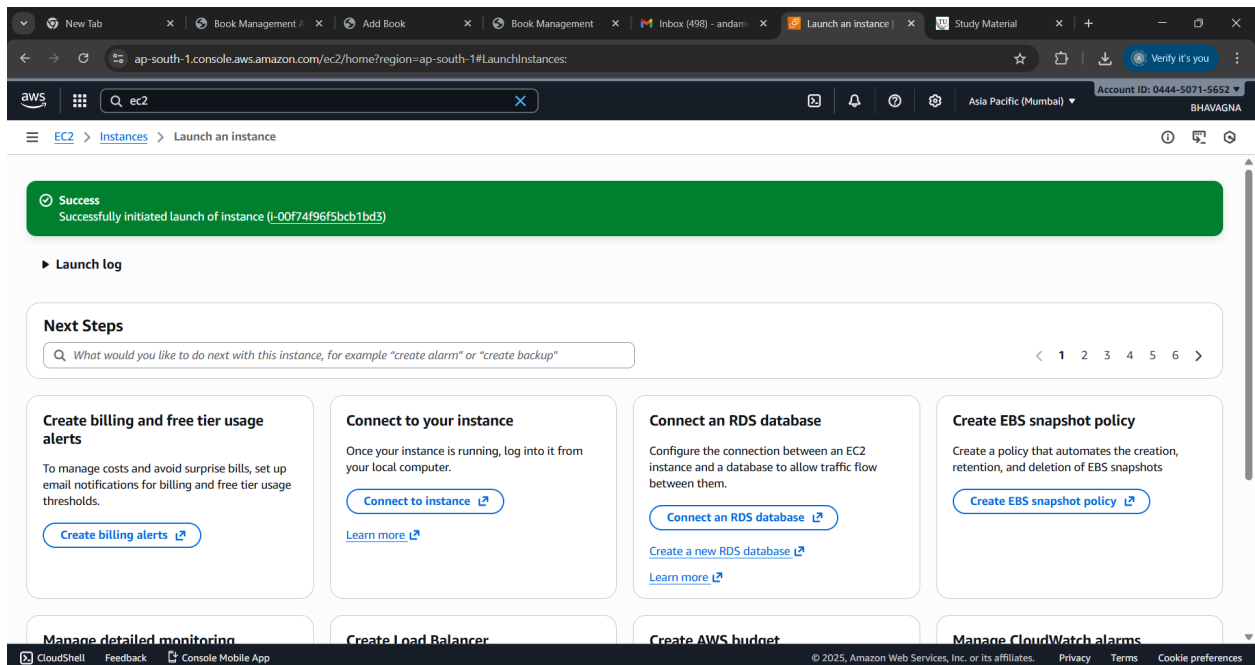
Stage 7 -- Storage - 8GB (Observation - we have root - it is same as C Drive)

Stage 8 --- click on launch instance

Stage 9: Number of instances ---1

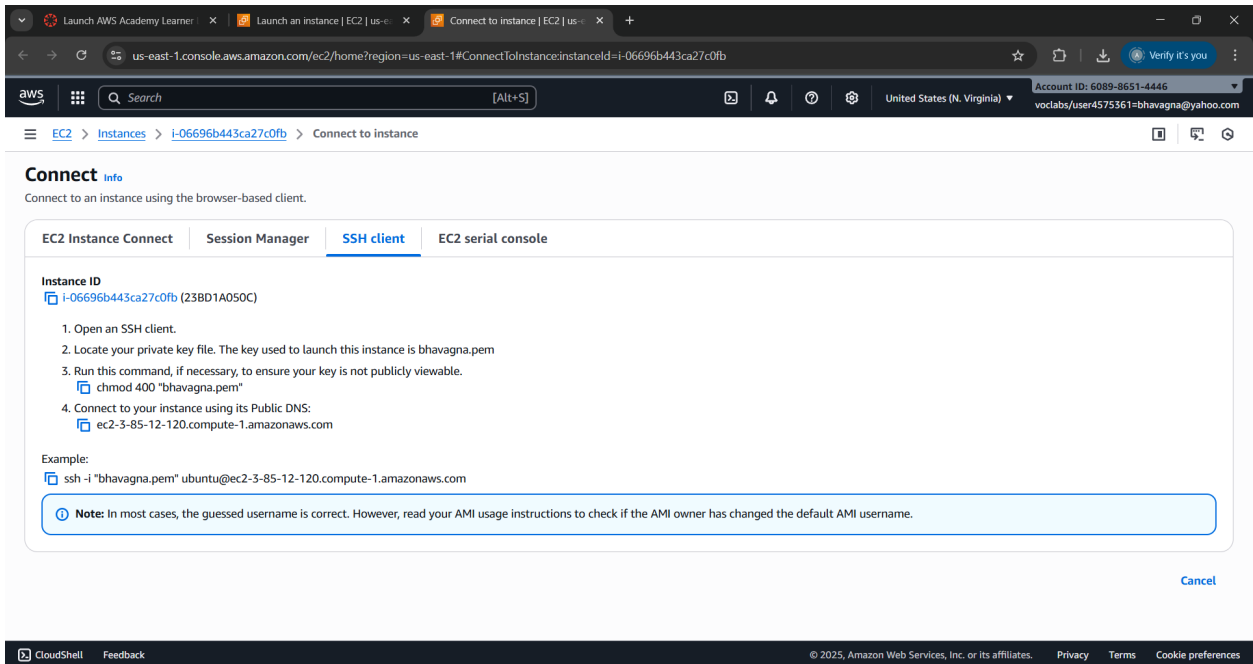
+++++

Observation - One machines created



Do this step:---once it is created **select that instance and click on connect**

Here copy the ssh – i command from SSH client connect tab



We can use powershell /gitbash /webconsole , to connect to ubuntu machine.

NOTE:- cd path of AWS folder // change path

To connect to above terminals we need to go into the path of the keypair.and

paste the

ssh -i command from the aws console

```
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..
PS C:\WINDOWS> cd..
PS C:\> cd users
PS C:\users> cd andam
PS C:\users\andam> cd downloads
PS C:\users\andam\downloads> ssh -i "23bd1a050c.pem" ubuntu@ec2-35-154-124-182.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-35-154-124-182.ap-south-1.compute.amazonaws.com (64:ff9b::239a:7cb6)' can't be established
ED25519 key fingerprint is SHA256:SL/a4YlEixiF1L3716mVtZWScxxRchIML6PWcmAWB4M.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-154-124-182.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
```

III. Clone the application from github, Write the Dockerfile

once connected to instance

Step 1:- **Run the following commands to install s/w**

1. Update all softwares in Ubuntu by command
sudo apt update

```
root@ip-172-31-5-60: /home/ x + v
System information as of Thu Nov 13 05:38:11 UTC 2025

System load: 0.16 Processes: 111
Usage of /: 25.8% of 6.71GB Users logged in: 0
Memory usage: 29% IPv4 address for enX0: 172.31.5.60
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.

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

ubuntu@ip-172-31-5-60:~$ sudo su
root@ip-172-31-5-60:/home/ubuntu# docker --version
Command 'docker' not found, but can be installed with:
apt install docker.io # version 28.2.2-0ubuntu1~24.04.1, or
apt install podman-docker # version 4.9.3+ds1-lubuntu0.2
root@ip-172-31-5-60:/home/ubuntu# sudo docker --version
sudo: docker: command not found
root@ip-172-31-5-60:/home/ubuntu# sudo apt-get update|
```

2. Install docker by command

sudo apt-get install docker.io

```
root@ip-172-31-5-60: /home/ x + v
Get:48 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2140 kB]
Get:49 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [486 kB]
Get:50 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [27.4 kB]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [5708 B]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [384 B]
Fetched 38.6 MB in 8s (4590 kB/s)
Reading package lists... Done
root@ip-172-31-5-60:/home/ubuntu# sudo apt-get 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 debotstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 14 not upgraded.
Need to get 76.0 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-lubuntu2 [33.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.2 [8815 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.1 [38.4 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801-ubuntu0.24.04.1 [5918 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 76.0 MB in 1s (65.9 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 71735 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-lubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-lubuntu2) ...
```

3. Install git by command

sudo apt install git

4. Install nano(text editor) by command

Sudo apt install nano

step 2:- git clone <paste the github link of maven-web-java project>


```
root@ip-172-31-5-60: /home/ x + v
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.
root@ip-172-31-5-60:/home/ubuntu# sudo docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
root@ip-172-31-5-60:/home/ubuntu# git clone https://github.com/bhavagna06/ecommerce.git
Cloning into 'ecommerce'...
Username for 'https://github.com': bhavagna06
Password for 'https://bhavagna06@github.com':
remote: Invalid username or token. Password authentication is not supported for Git operations.
fatal: Authentication failed for 'https://github.com/bhavagna06/ecommerce.git/'
root@ip-172-31-5-60:/home/ubuntu# git clone https://github.com/bhavagna06/ecommerce.git
Cloning into 'ecommerce'...
Username for 'https://github.com': bhavagna06
Password for 'https://bhavagna06@github.com':
remote: Invalid username or token. Password authentication is not supported for Git operations.
fatal: Authentication failed for 'https://github.com/bhavagna06/ecommerce.git/'
root@ip-172-31-5-60:/home/ubuntu# git clone https://github.com/bhavagna06/ecommerce.git
Cloning into 'ecommerce'...
Username for 'https://github.com': bhavagna06
Password for 'https://bhavagna06@github.com':
remote: Invalid username or token. Password authentication is not supported for Git operations.
fatal: Authentication failed for 'https://github.com/bhavagna06/ecommerce.git/'
root@ip-172-31-5-60:/home/ubuntu# git clone https://github.com/bhavagna06/internalmaven.git
Cloning into 'internalmaven'...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 51 (delta 6), reused 44 (delta 4), pack-reused 0 (from 0)
Receiving objects: 100% (51/51), 4.91 MiB | 5.95 MiB/s, done.
Resolving deltas: 100% (6/6), done.
root@ip-172-31-5-60:/home/ubuntu#
```

step 3:- navigate to the maven-web-java project

VI. Create the image

Step 4:- nano Dockerfile

```
minigv047/c/Users/archa/OneDrive/Desktop/web/maven_web_application/we-app
GNU nano 8.2 Dockerfile
# Use an official Java runtime as a parent image
FROM tomcat:9-jdk11

# Copy the built WAR file to the Tomcat webapps directory
COPY **/*.war /usr/local/tomcat/webapps/

# Expose the port Tomcat is running on
# EXPOSE 8080

# Start Tomcat server
# CMD ["catalina.sh", "run"]
```

V. Run the image and access it with public ip of virtual machine

Step 1:- build your image

docker build -t <imagename> .(dot)

```
root@ip-172-31-5-60: /home/ x + v
root@ip-172-31-5-60:/home/ubuntu# cd internalmaven
root@ip-172-31-5-60:/home/ubuntu/internalmaven# sudo docker build -t image .
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 10.65MB
Step 1/3 : FROM tomcat:9.0
9.0: Pulling from library/tomcat
4b3ffd8ccb52: Pulling fs layer
b48f960b388d: Pulling fs layer
58424d8c3e86: Pulling fs layer
4f4fb700ef54: Pulling fs layer
37b617836889: Pulling fs layer
891b6ad931b7: Pulling fs layer
ac0beccecf50: Pulling fs layer
4f4fb700ef54: Waiting
37b617836889: Waiting
891b6ad931b7: Waiting
ac0beccecf50: Waiting
4b3ffd8ccb52: Verifying Checksum
4b3ffd8ccb52: Download complete
b48f960b388d: Verifying Checksum
b48f960b388d: Download complete
58424d8c3e86: Verifying Checksum
58424d8c3e86: Download complete
4f4fb700ef54: Verifying Checksum
4f4fb700ef54: Download complete
37b617836889: Verifying Checksum
37b617836889: Download complete
4b3ffd8ccb52: Pull complete
891b6ad931b7: Verifying Checksum
891b6ad931b7: Download complete
ac0beccecf50: Verifying Checksum
ac0beccecf50: Download complete
b48f960b388d: Pull complete
58424d8c3e86: Pull complete
4f4fb700ef54: Pull complete
37b617836889: Pull complete
891b6ad931b7: Pull complete
ac0beccecf50: Pull complete
```

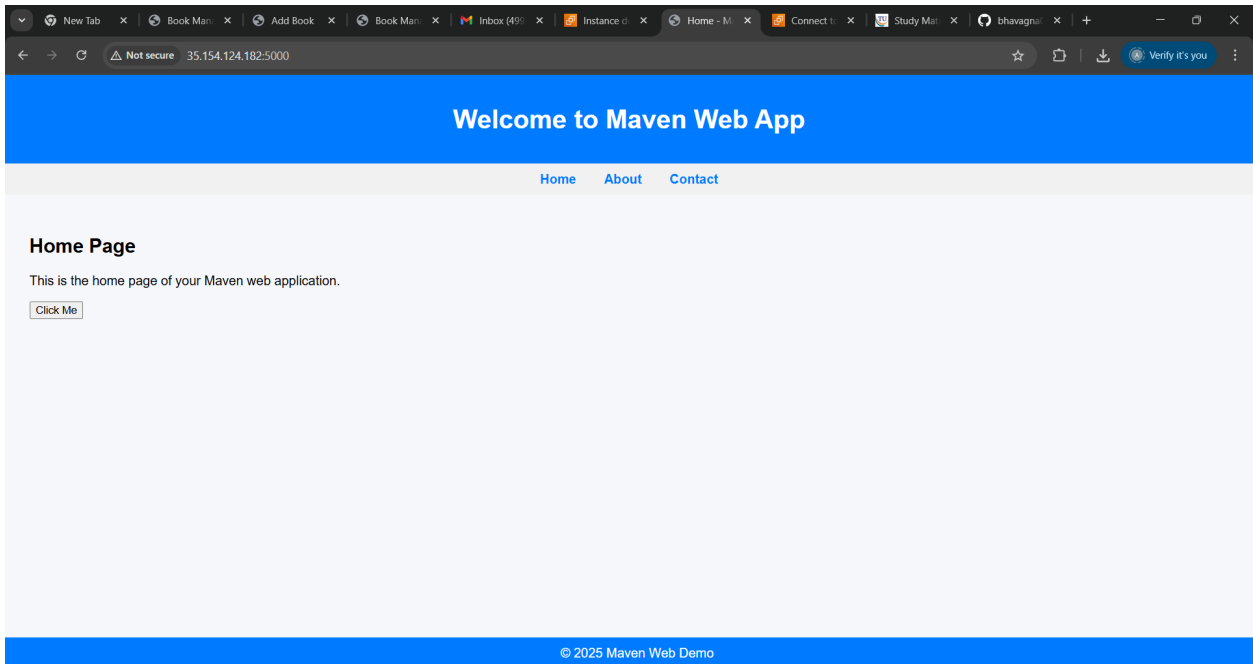
Step 2:- check for images

Step 3:- run image

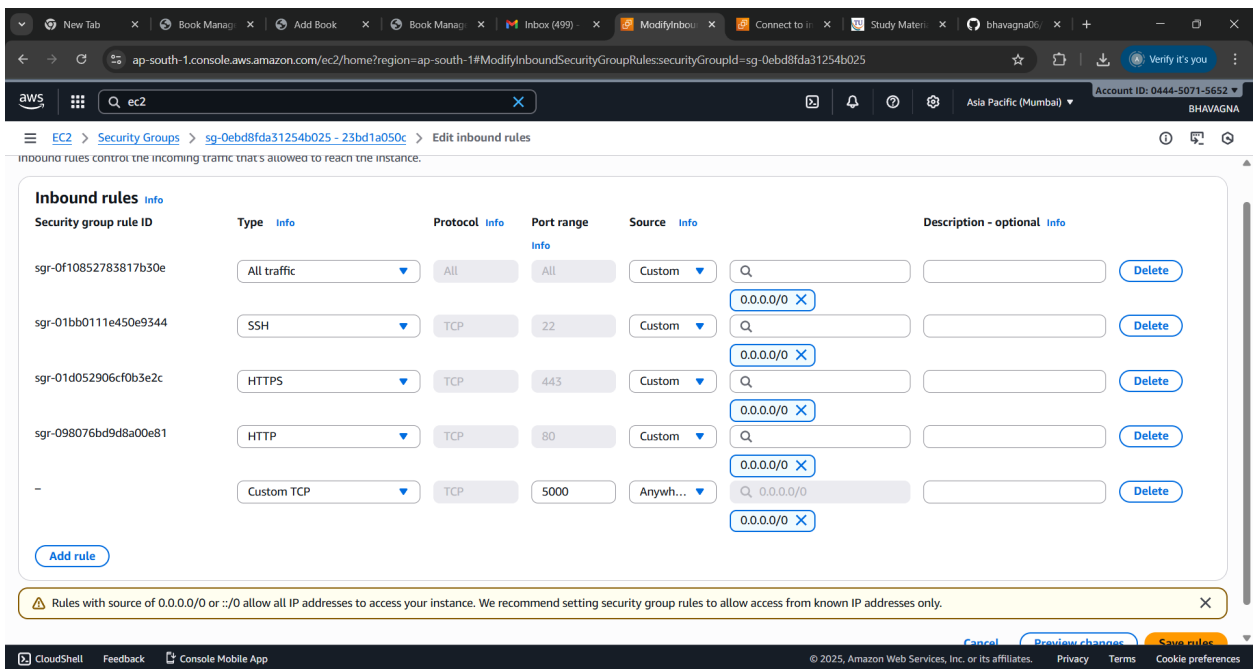
`docker run -d --name app-demo -p 6060:8080 <image name>`

```
root@ip-172-31-5-60: /home/ x + v
891b6ad931b7: Pulling fs layer
ac0beccecf50: Pulling fs layer
4f4fb700ef54: Waiting
37b617836889: Waiting
891b6ad931b7: Waiting
ac0beccecf50: Waiting
4b3ffd8ccb52: Verifying Checksum
4b3ffd8ccb52: Download complete
b48f960b388d: Verifying Checksum
b48f960b388d: Download complete
58424d8c3e86: Verifying Checksum
58424d8c3e86: Download complete
4f4fb700ef54: Verifying Checksum
4f4fb700ef54: Download complete
37b617836889: Verifying Checksum
37b617836889: Download complete
4b3ffd8ccb52: Pull complete
891b6ad931b7: Verifying Checksum
891b6ad931b7: Download complete
ac0beccecf50: Verifying Checksum
ac0beccecf50: Download complete
b48f960b388d: Pull complete
58424d8c3e86: Pull complete
4f4fb700ef54: Pull complete
37b617836889: Pull complete
891b6ad931b7: Pull complete
ac0beccecf50: Pull complete
Digest: sha256:f3fa67518217a4ec196414e49dcc593b2875bd2ac83b712e2122df05593c95a0
Status: Downloaded newer image for tomcat:9.0
--> 2e4887a16e43
Step 2/3 : COPY ./target/*.war /usr/local/tomcat/webapps/ROOT.war
--> 6c05a7511950
Step 3/3 : CMD ["catalina.sh", "run"]
--> Running in a9dc0b6bbd11
--> Removed intermediate container a9dc0b6bbd11
--> 162c60854b81
Successfully built 162c60854b81
Successfully tagged image:latest
root@ip-172-31-5-60:/home/ubuntu/internalmaven# sudo docker run -d -p 5000:8080 image
30f2e372a34b038de80121a547614392af0c15585beda68dfc17f8f8b3b47923
root@ip-172-31-5-60:/home/ubuntu/internalmaven#
```

Step:-4 **Accessing the app by public ip of virtual machine**



Note:-if your are not able to connect change the inbound rules..



Scenerio based questions

1. You have a simple index.html file on your laptop and you launched an EC2 instance with Amazon Linux 2. The instance is running but when you open the public IP in browser, the page doesn't load.
What steps will you take to host the index.html?

Edit the inbound rules in security group and allow the port no of index.html and reload the page you view the output

2. You deployed your index.html to /var/www/html/ directory on EC2, but the web page still isn't loading. What are two possible issues you would check?

Security group inbound rule

Container port forward

3. You installed Apache HTTP server on EC2 to host index.html, but the service stops after instance reboot. What command should you run to ensure it auto-starts on boot?

`sudo systemctl enable httpd`

4. You are deploying a Maven web application onto an EC2 instance. Maven is not installed on the instance. What commands or steps will you follow to install Maven on Amazon Linux/Ubuntu EC2?

`sudo apt update`

`sudo apt install maven -y`

5. You built a Maven project on EC2, and a .war file generated inside target/.

You want to deploy it using Tomcat. Where will you place the .war file and why?

/usr/local/tomcat/webapps/

The webapps directory is where Tomcat automatically deploys WAR files. Tomcat will extract the WAR and create the application folder

6. Your Maven web app is deployed to Tomcat on EC2, but accessing it via browser gives 404 error. What configuration or path issues will you check?

Correct url path

War file

7. You can access your web application locally on the EC2 instance using curl localhost:8080 but not from your browser. What AWS setting is likely missing or misconfigured?

Security Group inbound rule for port 8080

8. You have deployed your web app on EC2 successfully, but after shutting down your local Wi-Fi and reconnecting, the public IP changed and the app is not opening. What AWS feature helps avoid this issue?

Elastic IP remains constant even if instance stops/restarts

9. You want to automate deployment of your index.html whenever you restart the EC2 instance. Which EC2 feature can be used to run commands automatically during instance setup?

EC2 User Data (Cloud-init)

10. Your Maven application needs external dependencies during build but EC2 has no internet.

What AWS service or change can allow the EC2 instance to download dependencies securely?

NAT Gateway

Attach Public IP