

# Module 1: AWS Cloud

~ Abhijit Lalasaheb Zende

[For more such notes, visit my portfolio website](#)

## Tasks:

- 1) Demonstrate the AWS EC2 Ubuntu Instance Creation steps and connect to EC2 Instance using Mobaxterm agent**
- 2) Login to AWS Console and Create IAM User, Role, and Group**
- 3) Launch AWS EC2 Ubuntu Instance and configure the Security Group - Inbound Rule: 8080. Justify the usage of Inbound Rules**
- 4) Connect to the AWS EC2 Ubuntu Instance and Update default packages, install JDK, Maven, Git, and validate the versions**
- 5) Install Tomcat web application server in AWS EC2 Ubuntu Instance and access Tomcat using a web browser**
- 6) Create S3 Bucket and add folders and files**

# **1. L1 - Demonstrate the AWS EC2 Ubuntu Instance Creation steps and connect to EC2 Instance using Mobaxterm/putty agent**

Ans –

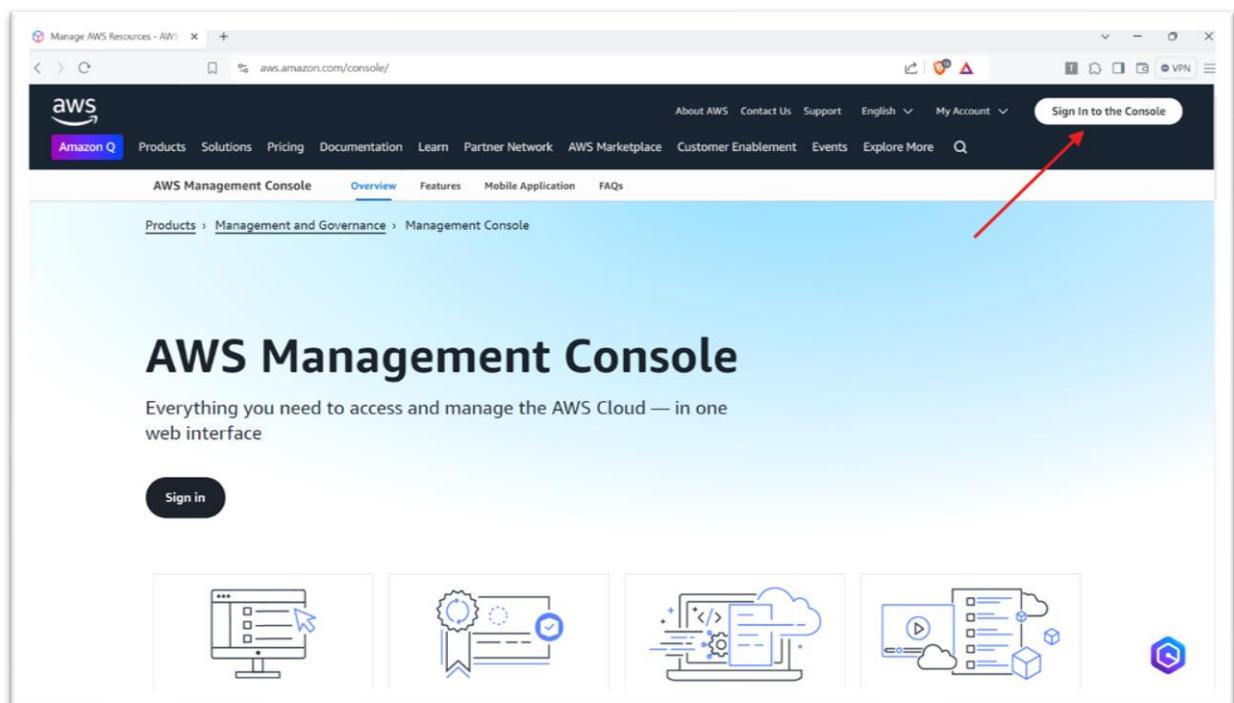
## **A] AWS EC2 Ubuntu Instance creation:**

### **1. AWS Account creation (If not already created):**

To create an AWS account if not already created, follow the steps provided in the official AWS documentation: [Create and activate an AWS account.](#)

### **2. Login to AWS Console:**

After account creation, open your favourite browser and log in to the [AWS management console](#) using your credentials



*Fig 1.01: AWS management console page*

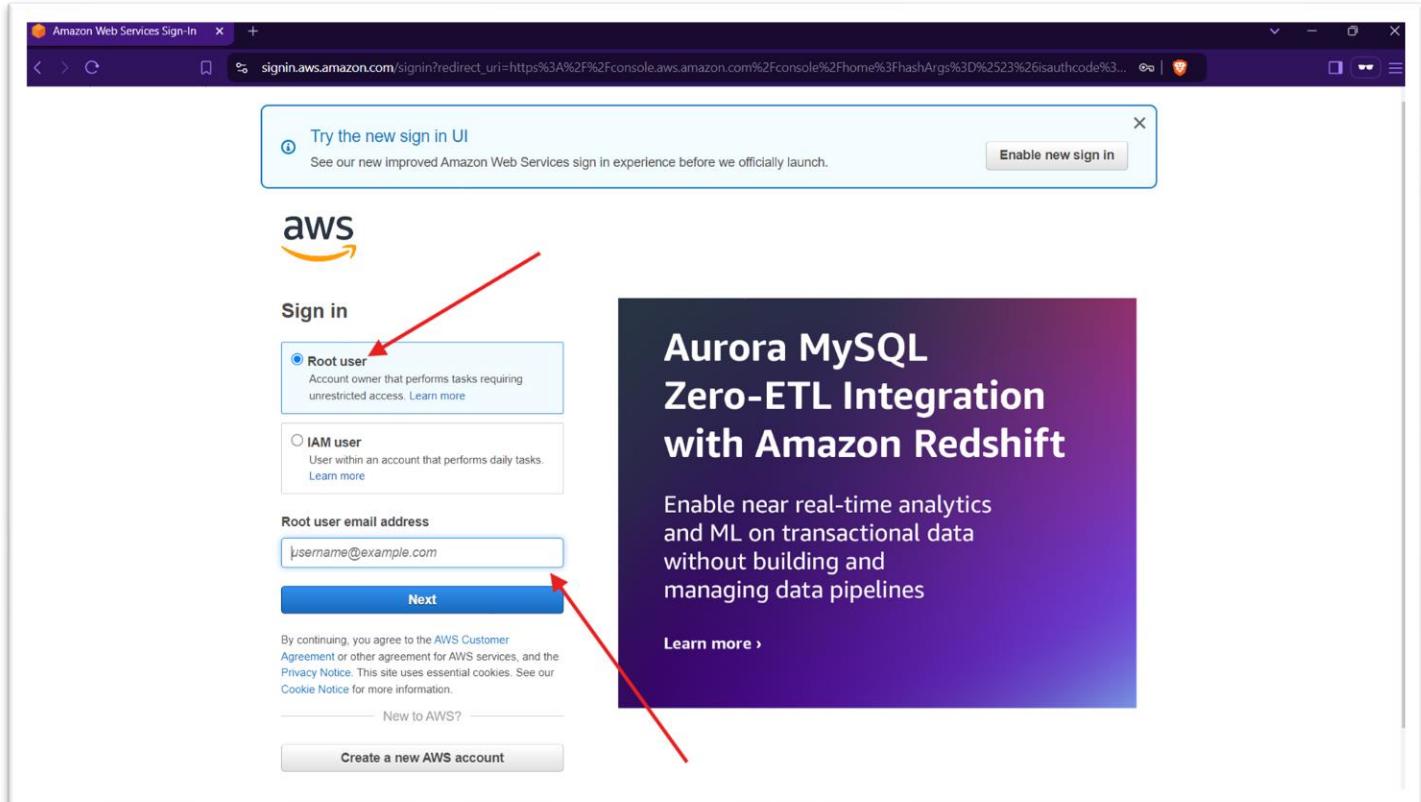


Fig. 1.02: AWS login page

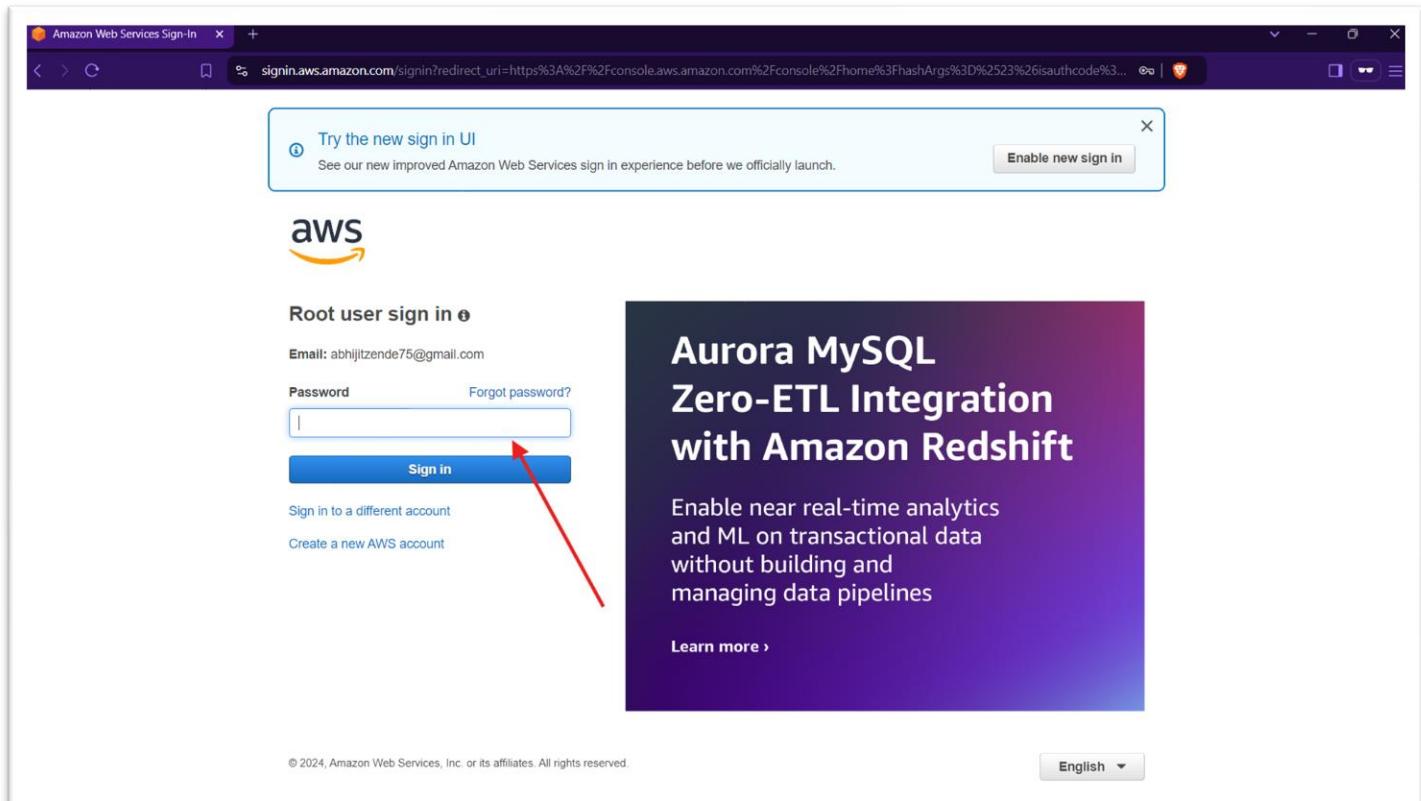


Fig. 1.03: Password authentication page

### 3. Navigate to EC2 Dashboard:

In the AWS Management Console, First make sure you have selected the nearest region to you as all the resources are region specific. Then search for "EC2" in the search bar and select EC2 from the services dropdown.

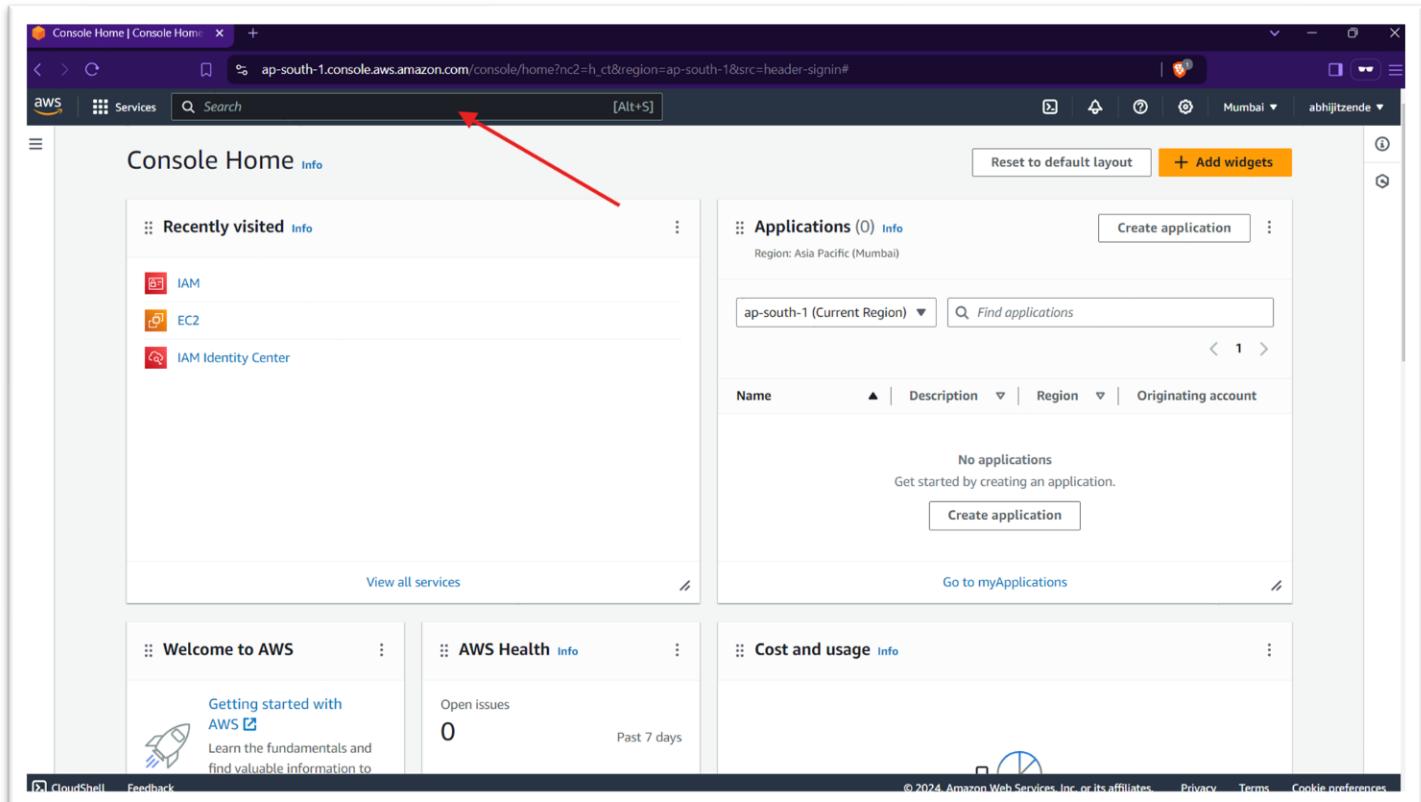
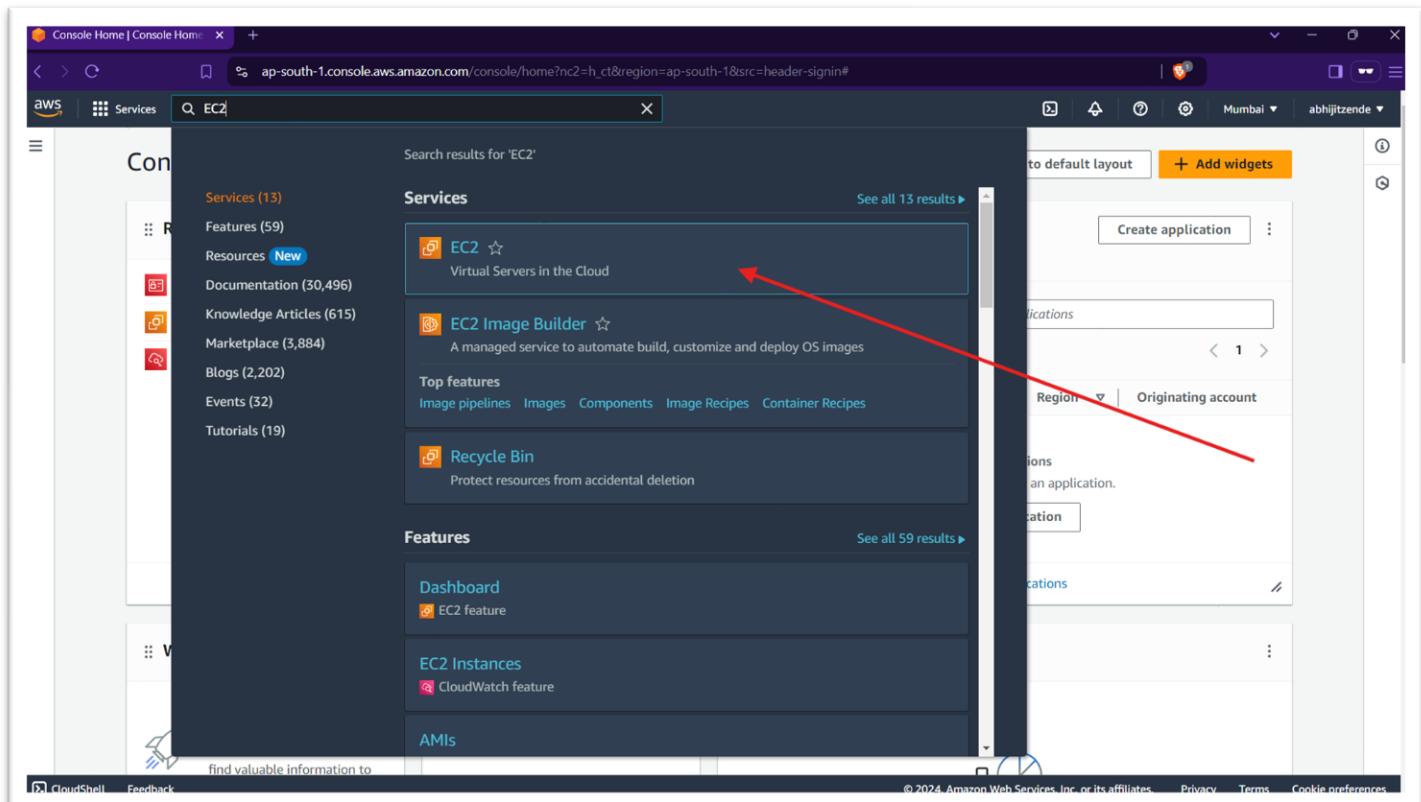


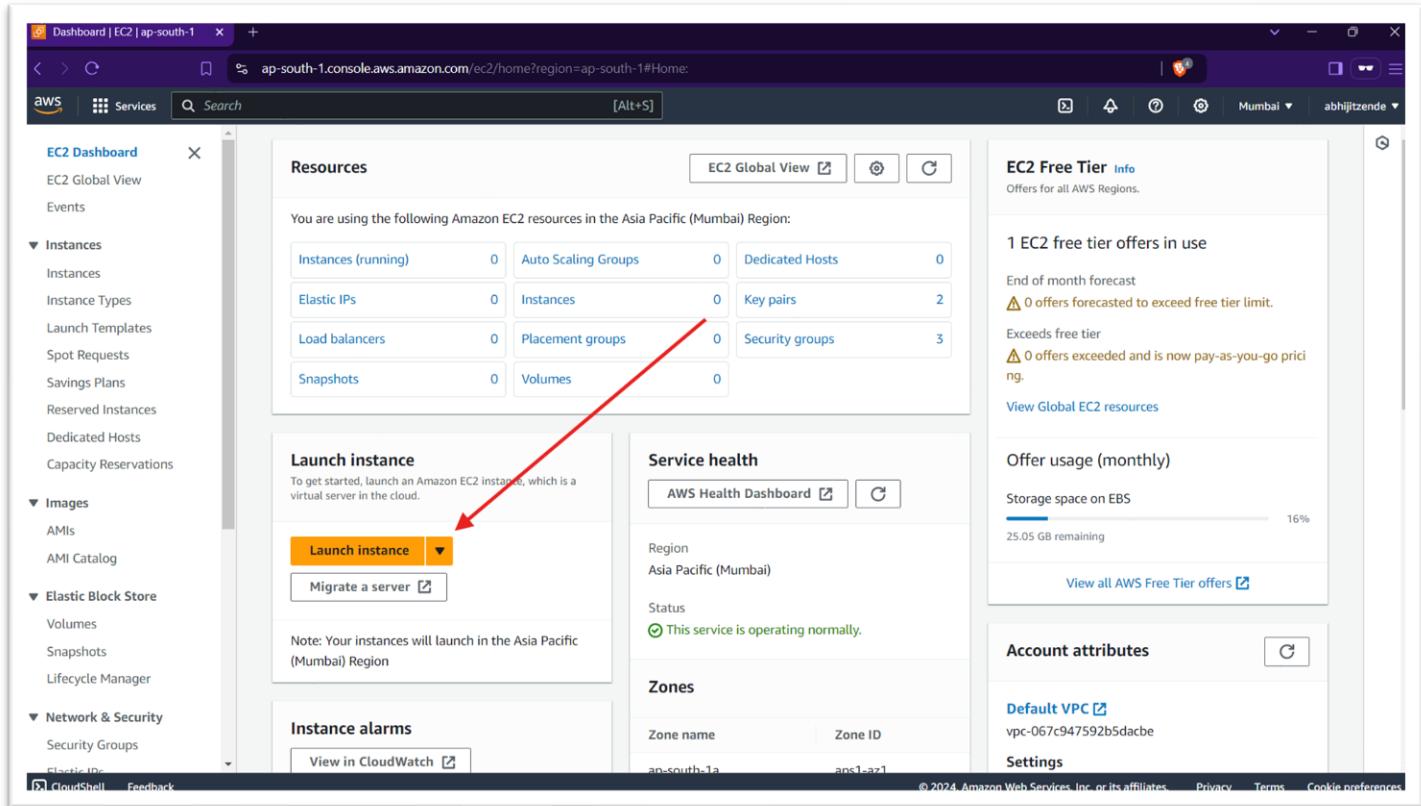
Fig. 1.04: AWS management page after successful login



*Fig. 1.05: AWS management search page*

#### 4. Launch Instance:

Click on "Launch Instance" [[Launch instance link](#)] from the EC2 dashboard.



*Fig.1.06: AWS Launch instance option*

#### 5. Configure the instance:

- Provide name and tag for the instance
- Specify number of instances to launch
- Choose your required Amazon Machine Image (AMI)
- Select "t2.micro" (eligible for the free tier) for basic testing or choose another instance type as per your needs.
- Keep remaining network to by default
- Choose "Create a new key pair", name it, and download the key pair (a .pem file). Keep this file safe as it is needed to connect to the instance. Click "Launch Instances"
- Review your instance settings and click "Launch".
- Click "View Instances" to go to the instance dashboard. Wait until the instance state shows "running".
- Once the status of your instance is running, congratulations on creating your new instance

**Note:** All instances are region specific. Therefore always keep an eye on region before any operations

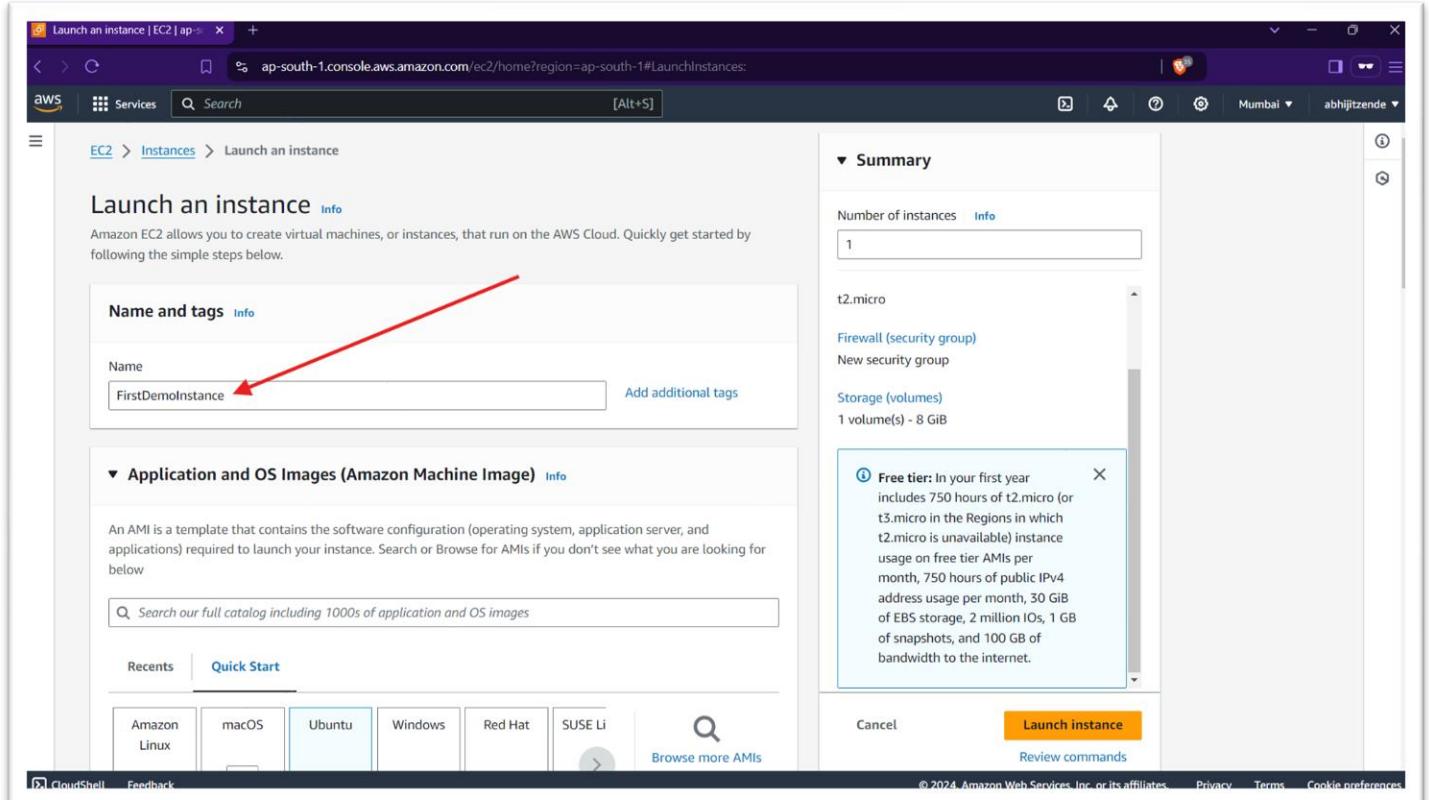
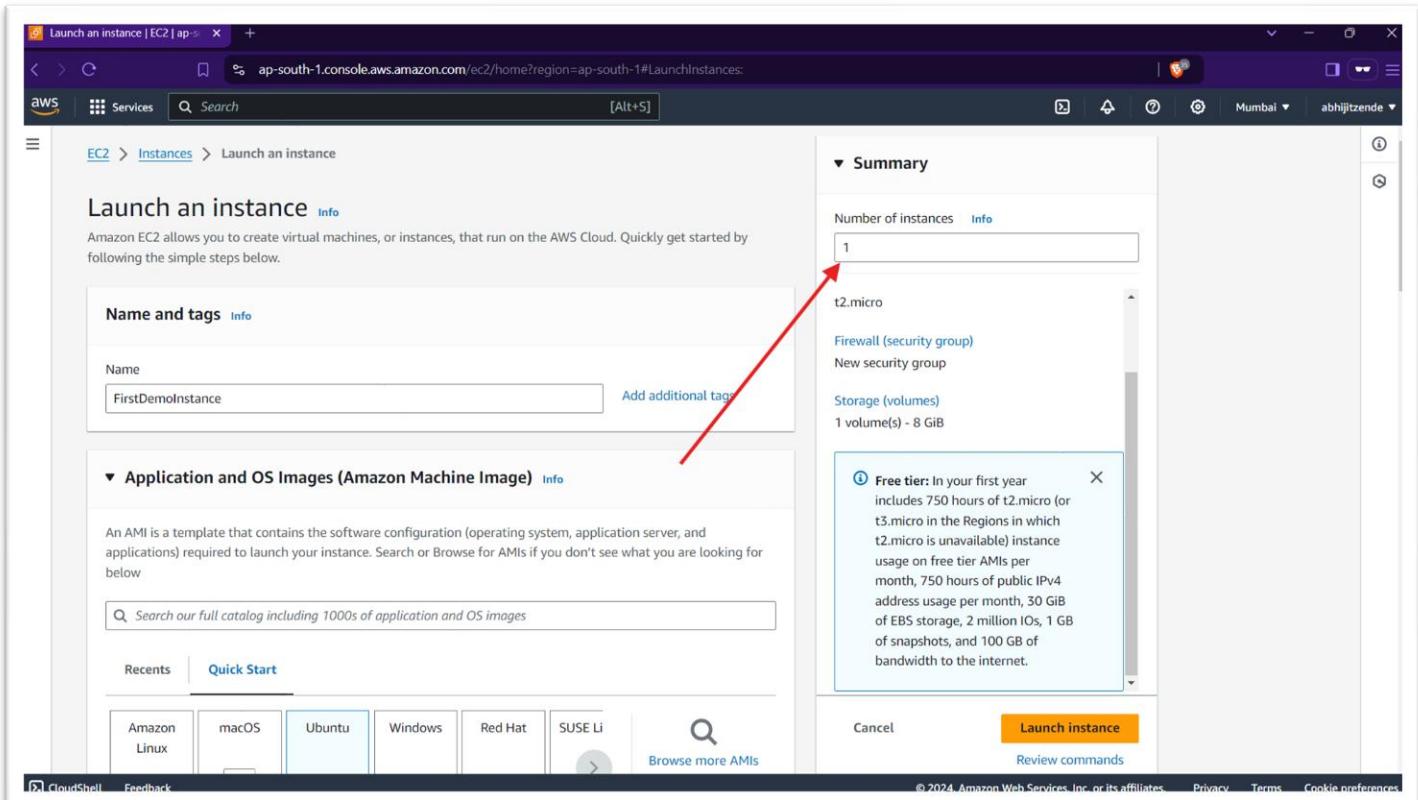
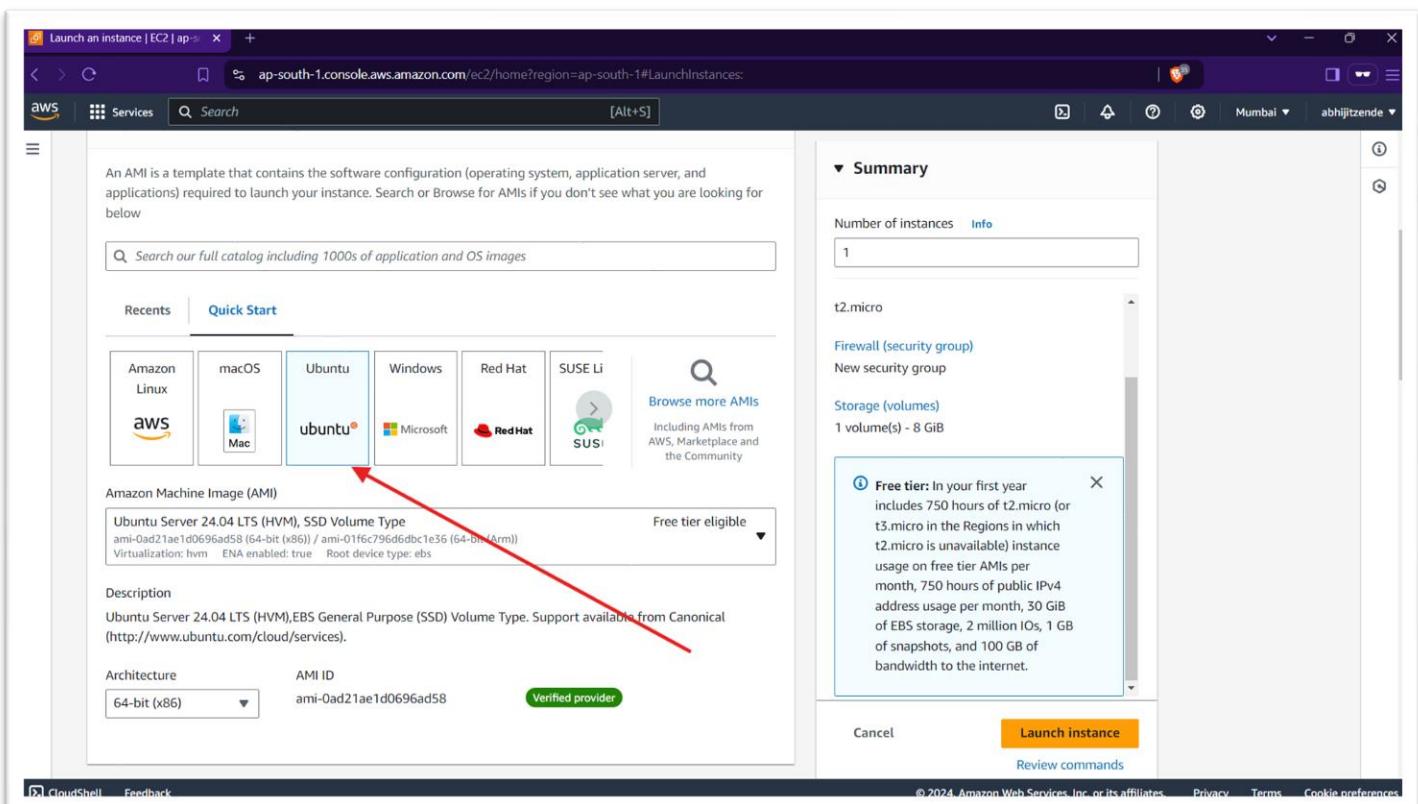


Fig. 1.07: Launch instance page[Specify name and tag for instance]



*Fig. 1.08: Launch instance page[Specify number of instances to launch]*



*Fig. 1.09: Launch instance page[Select Amazon Instance Image for instance]*

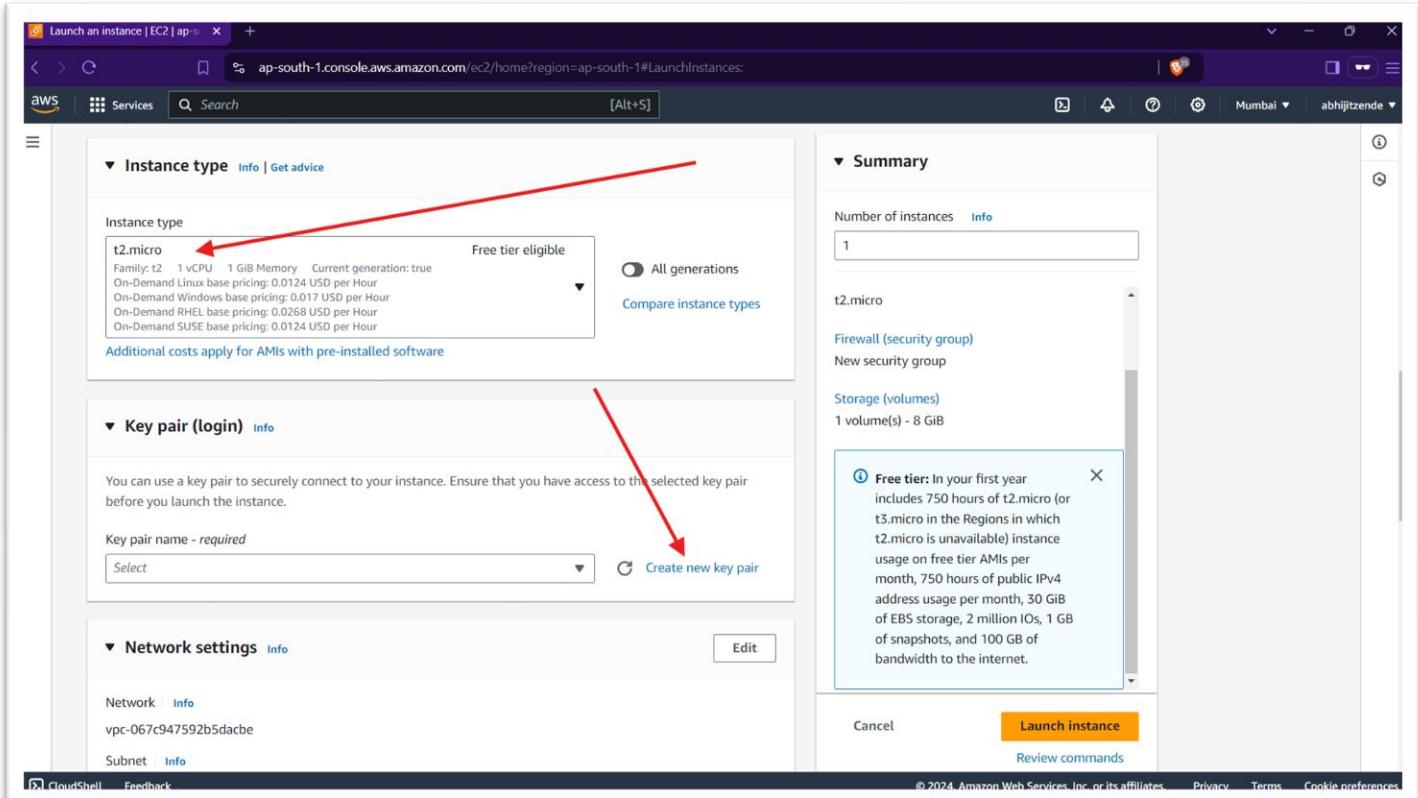


Fig. 1.10: Launch instance page[Select Instance type and Create new key pair for instance]

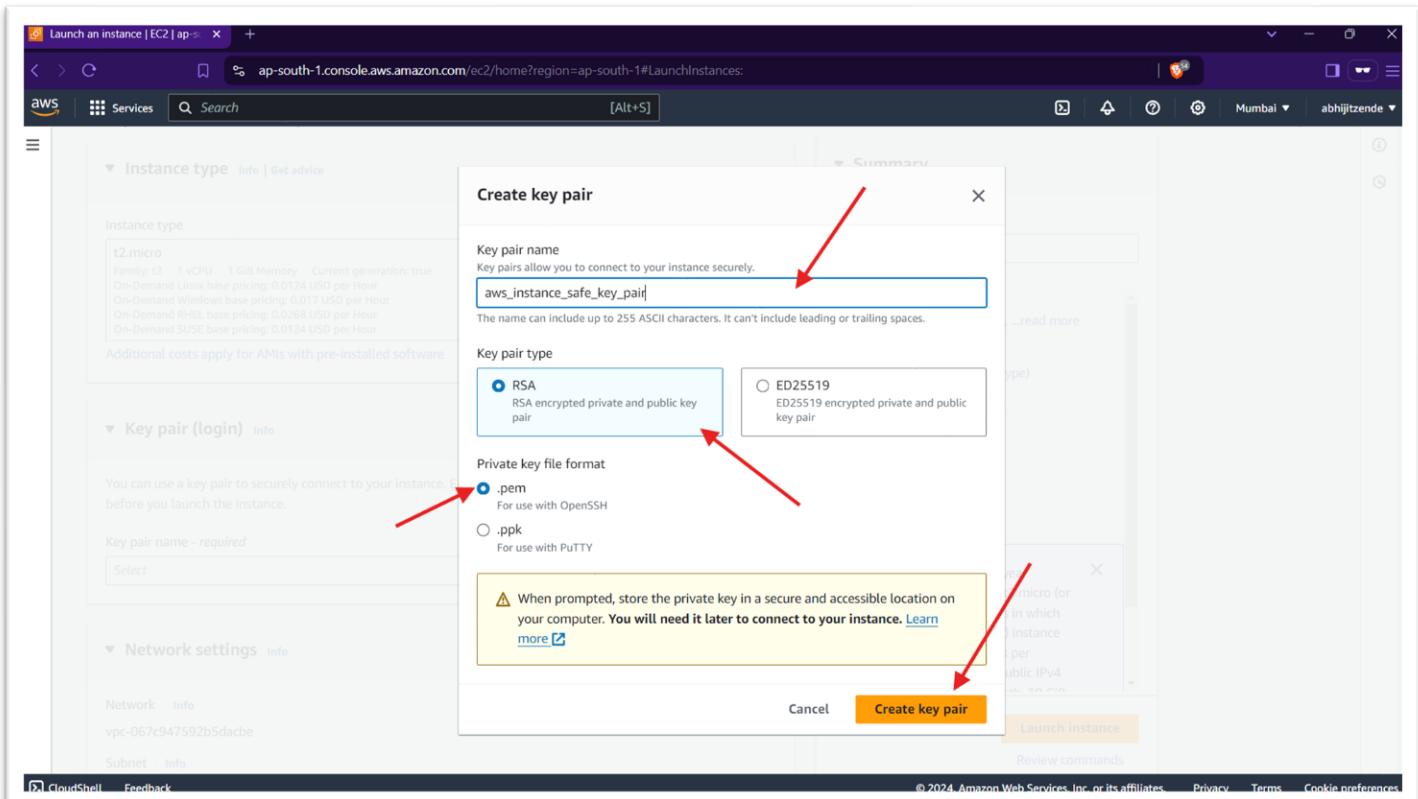


Fig. 1.11: Launch instance page[Specify configurations as above for key pair for instance]

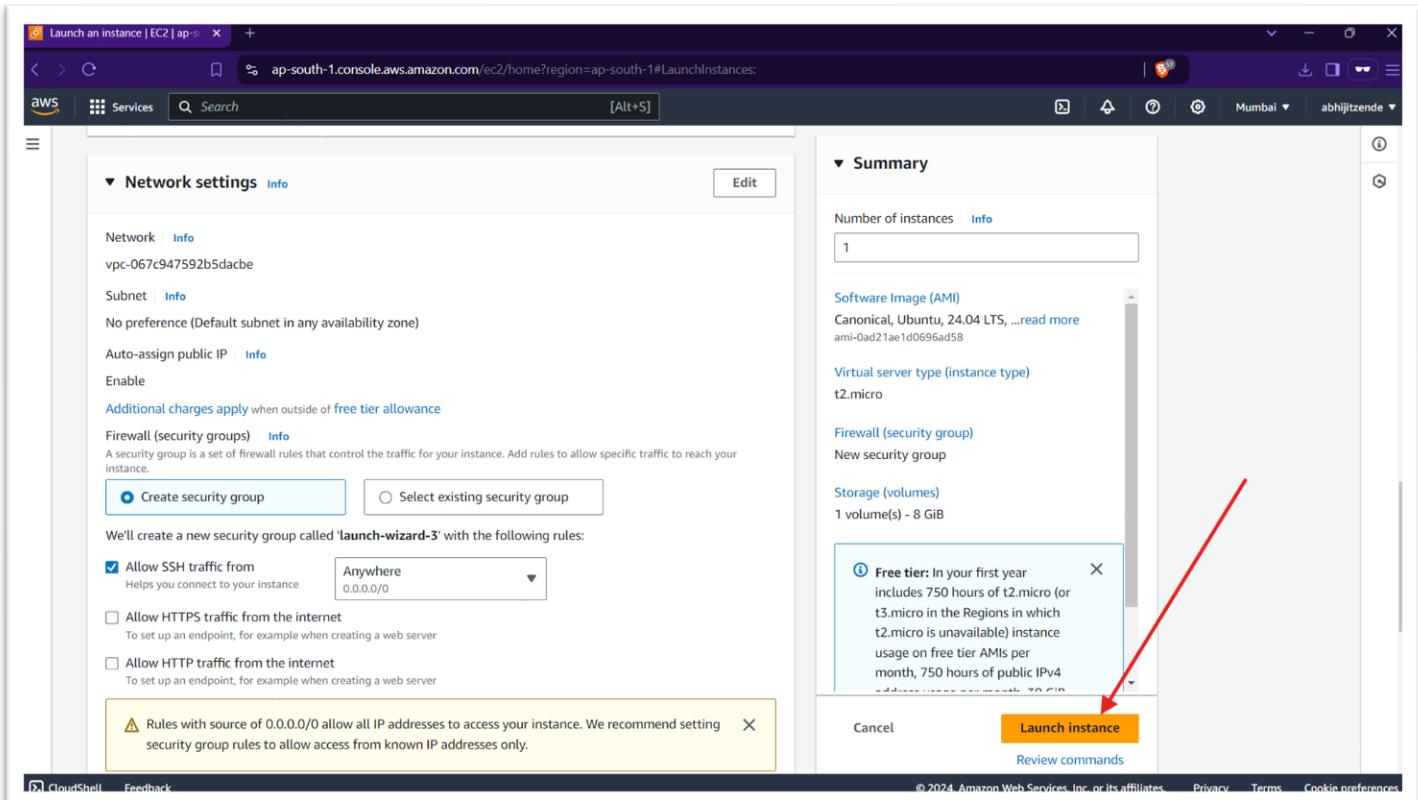


Fig. 1.12: Launch instance page[Keep network settings to default and click on “Launch Instance”]

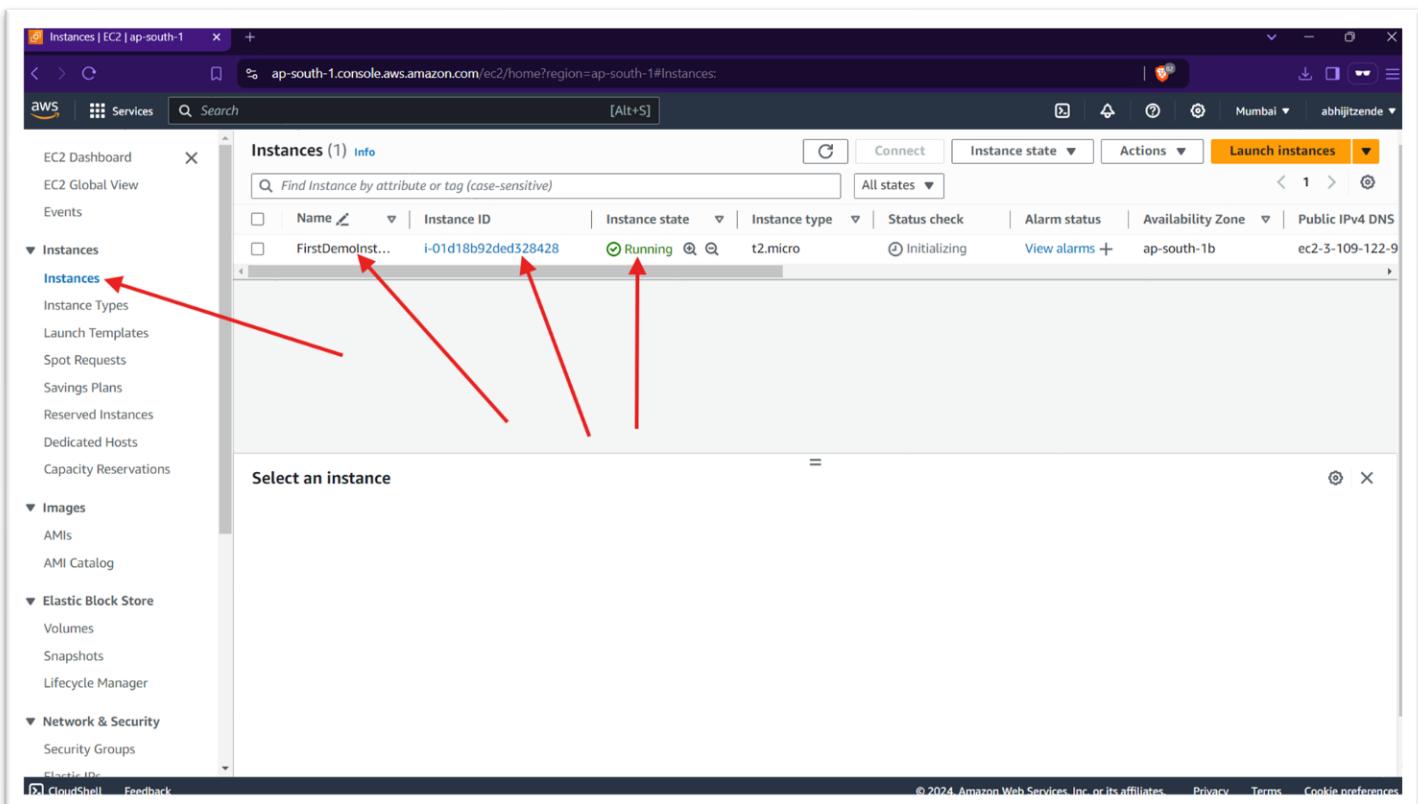


Fig. 1.13: Instance page[Checking if instance launched successfully]

## B] Connecting to EC2 instance using MobaXterm agent:

### 1. Download and Install MobaXterm:

If you do not have it already, download and install MobaXterm from the [Official download link](#).

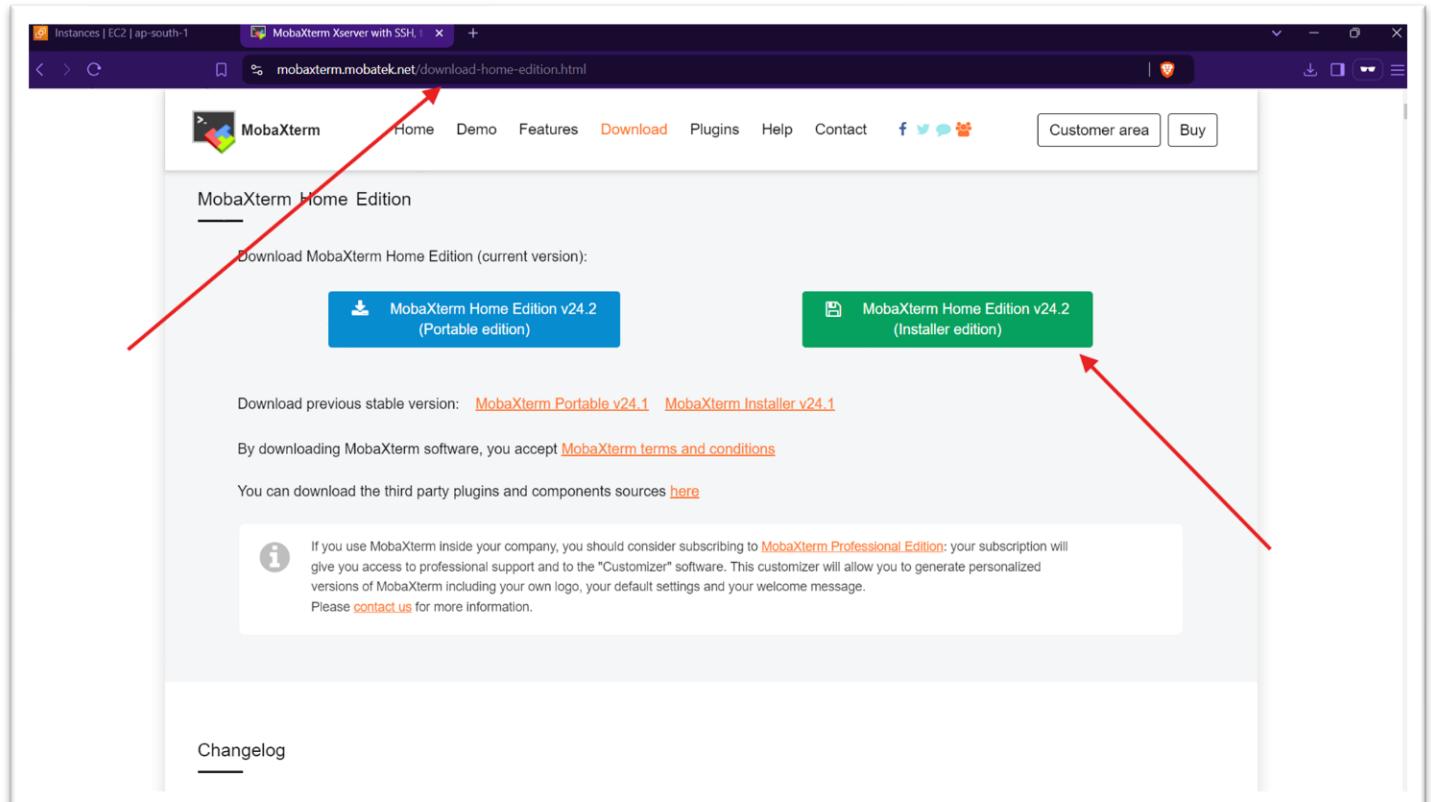


Fig. 1.14: Official MobaXterm download page

### 2. Check if status of EC2 instance is running:

From the instance page of AWS management console check if the status of the required instance is running. If the status is other than ‘running’ then select the instance and click the ‘instance state’ option and select ‘Start instance’ option to start the instance

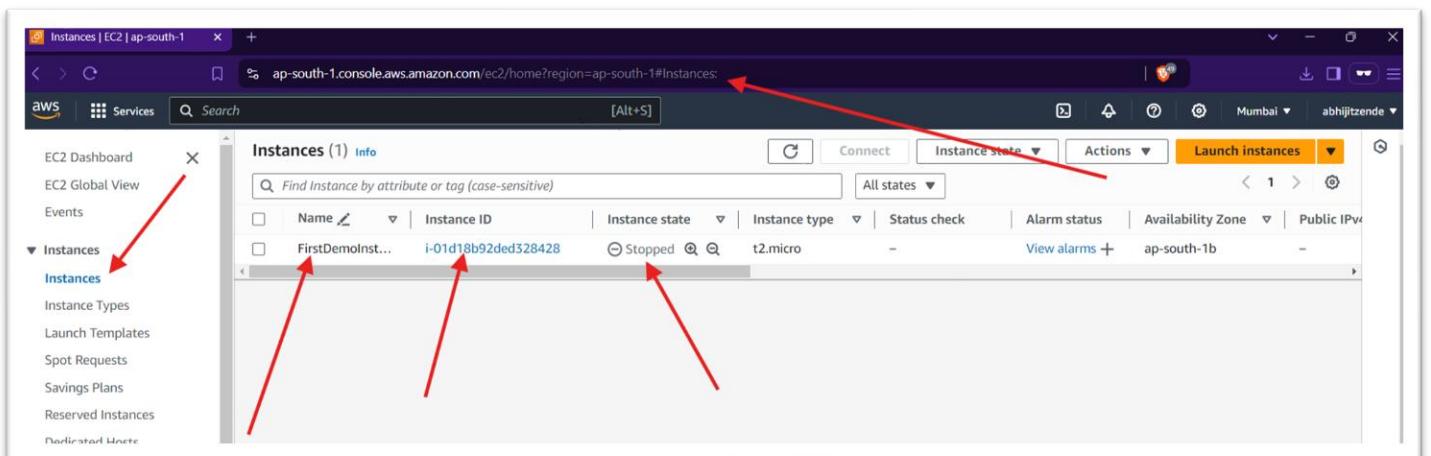


Fig. 1.15: Instance page in AWS management console[status check]

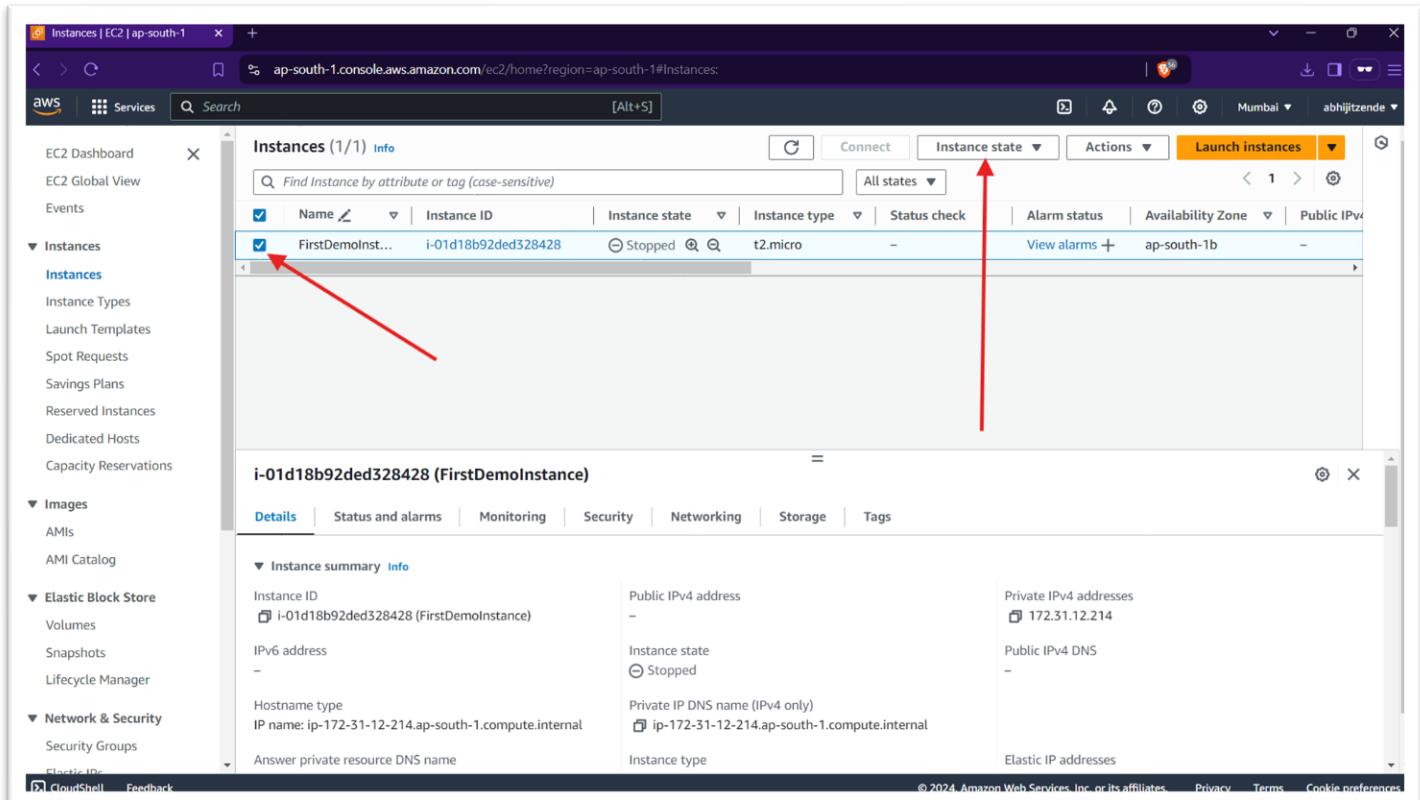


Fig. 1.16: Selecting instance to change instance status

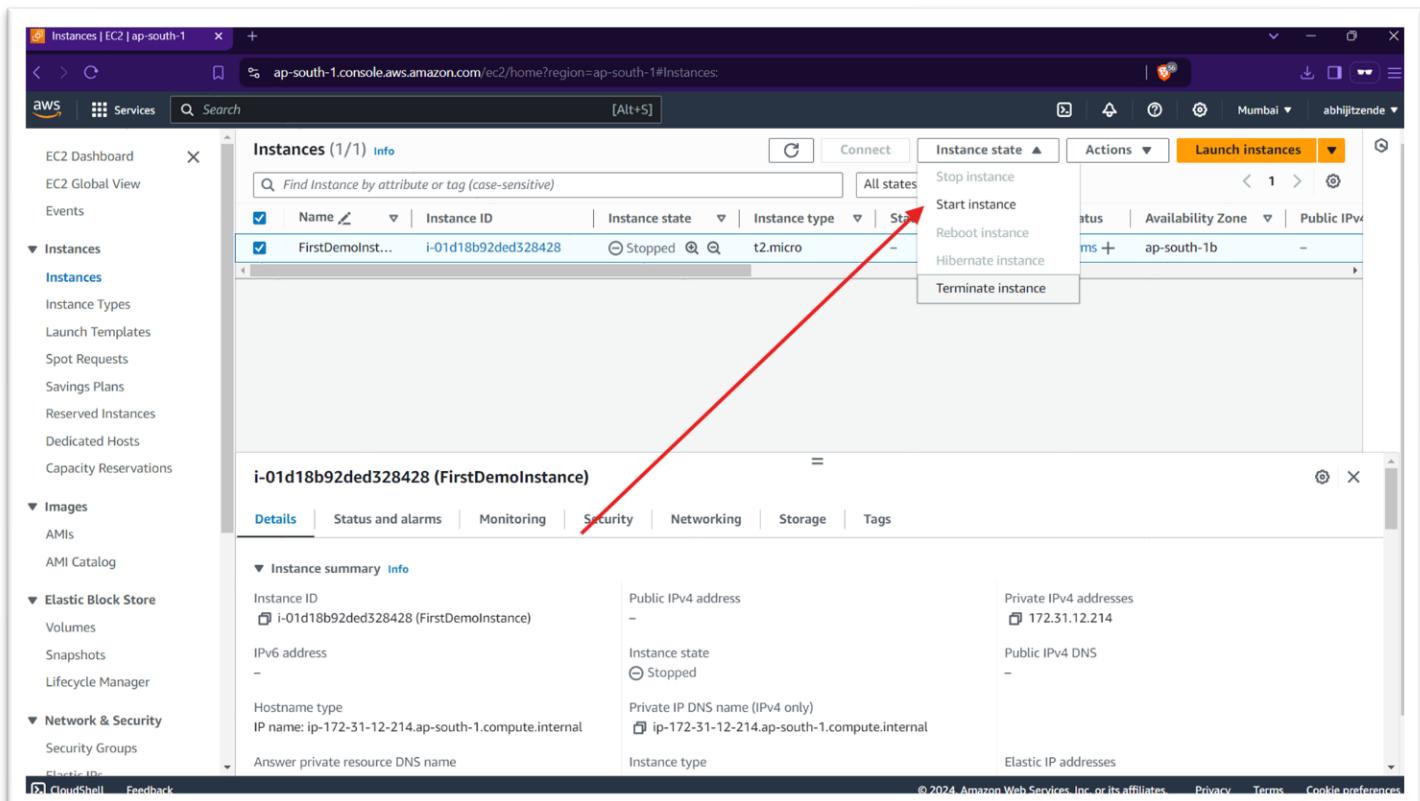
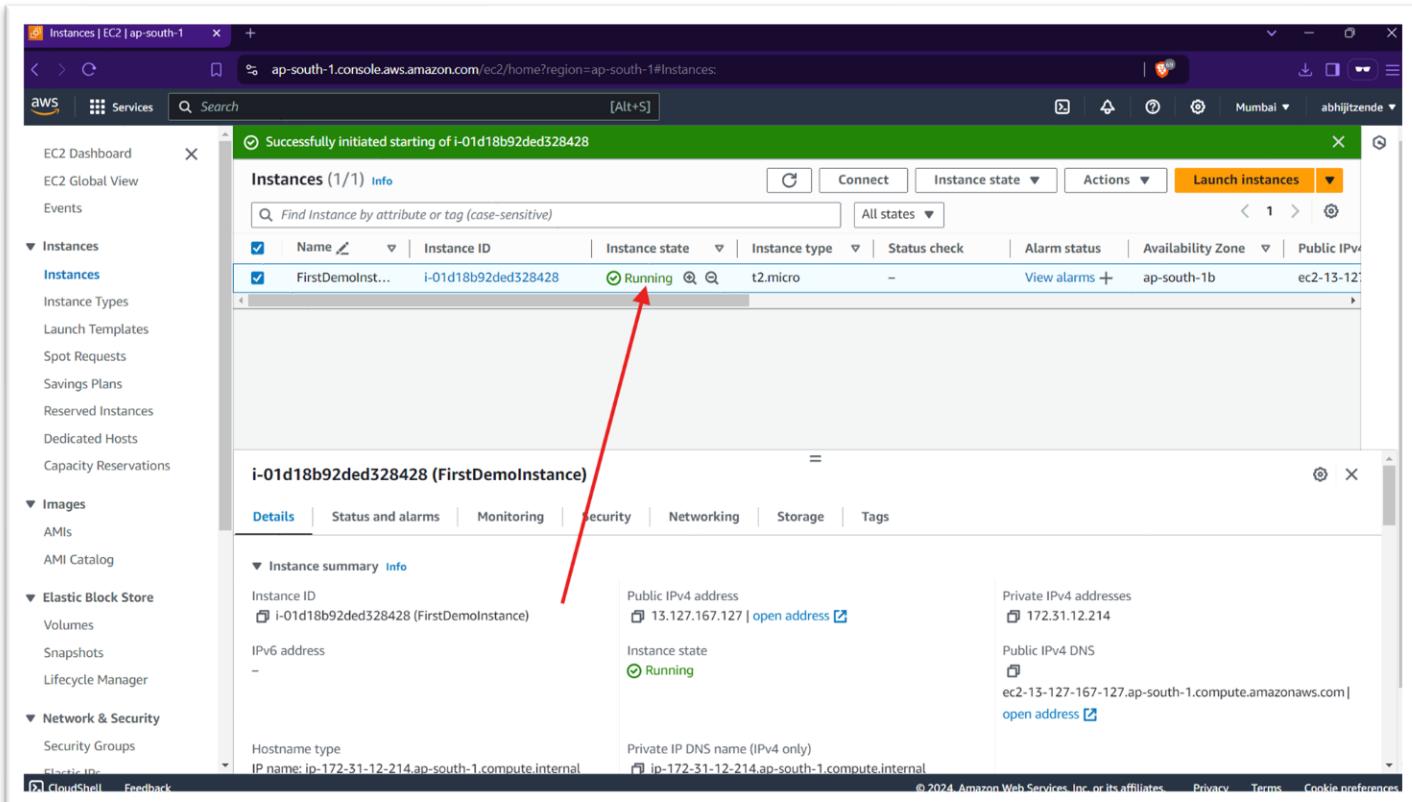


Fig. 1.17: Select Start instance option to start the selected instance



*Fig. 1.18: Instance successfully running*

### 3. Load the Key Pair:

- Open MobaXterm, go to ‘Session>SSH’ option
- Copy the IPv4 address of the instance from AWS management console/instance
- Paste the address in the ‘Remote host’ section.
- Enable the ‘Specify username’ checkbox and enter ‘ubuntu’ as username
- Keep the ‘port’ as ‘22’ as it is default port for ssh connection
- Click on ‘Advanced SSH settings’ and enable ‘Use private key’ checkbox
- Provide path to the previously saved key-pair
- Click ‘Ok’ to connect.

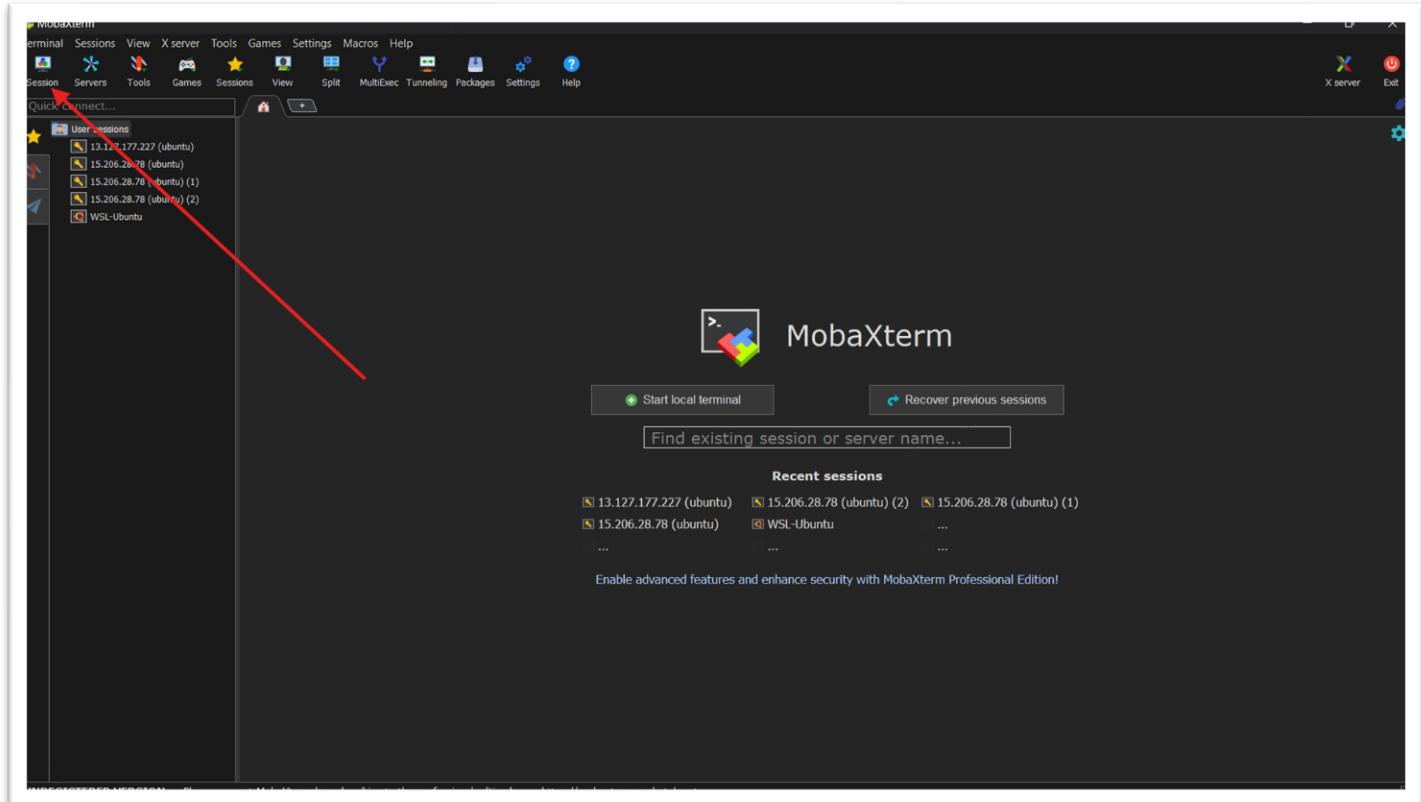


Fig. 1.19: MobaXterm home page

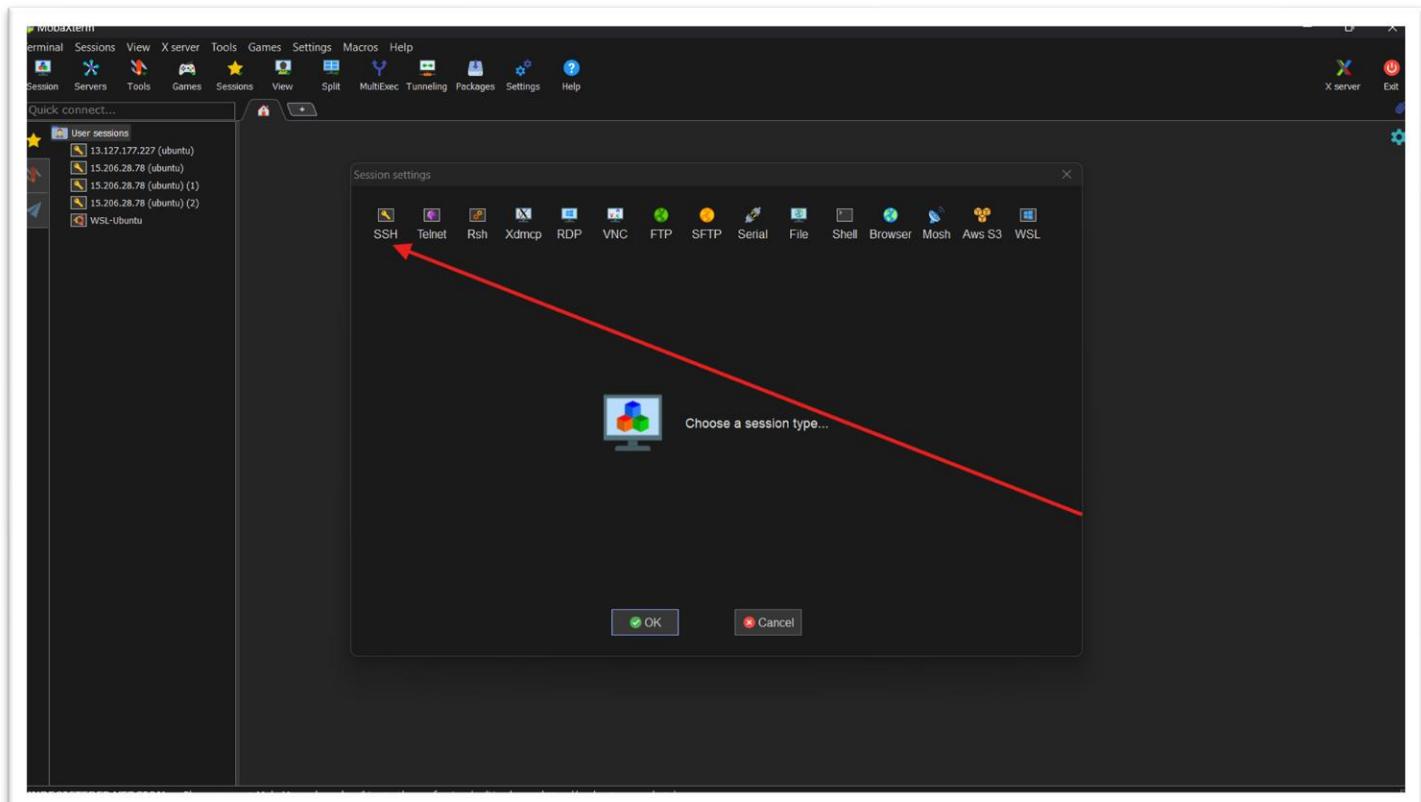


Fig. 1.20: MobaXterm connecting through SSH

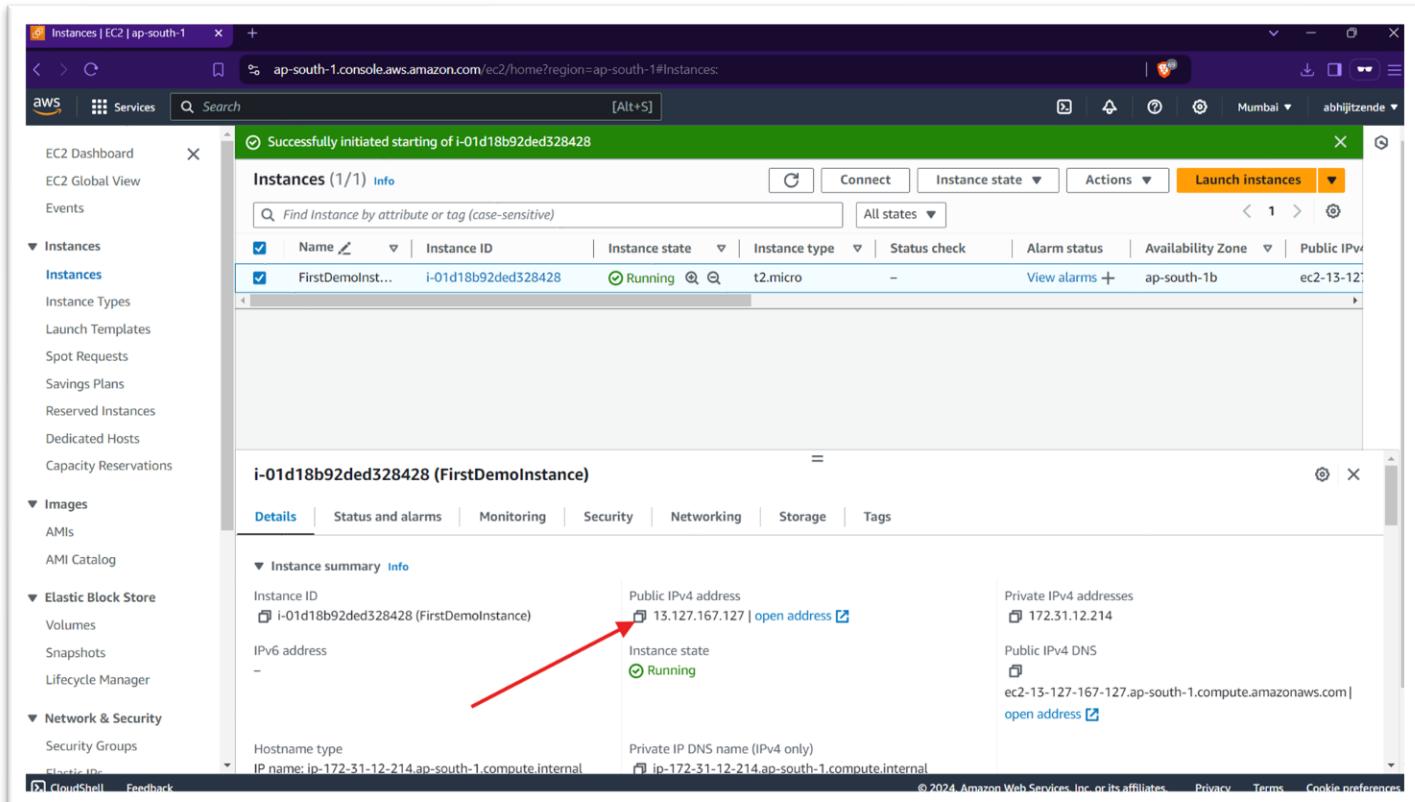


Fig. 1.21: Copying IPv4 address from AWS management console/Instance

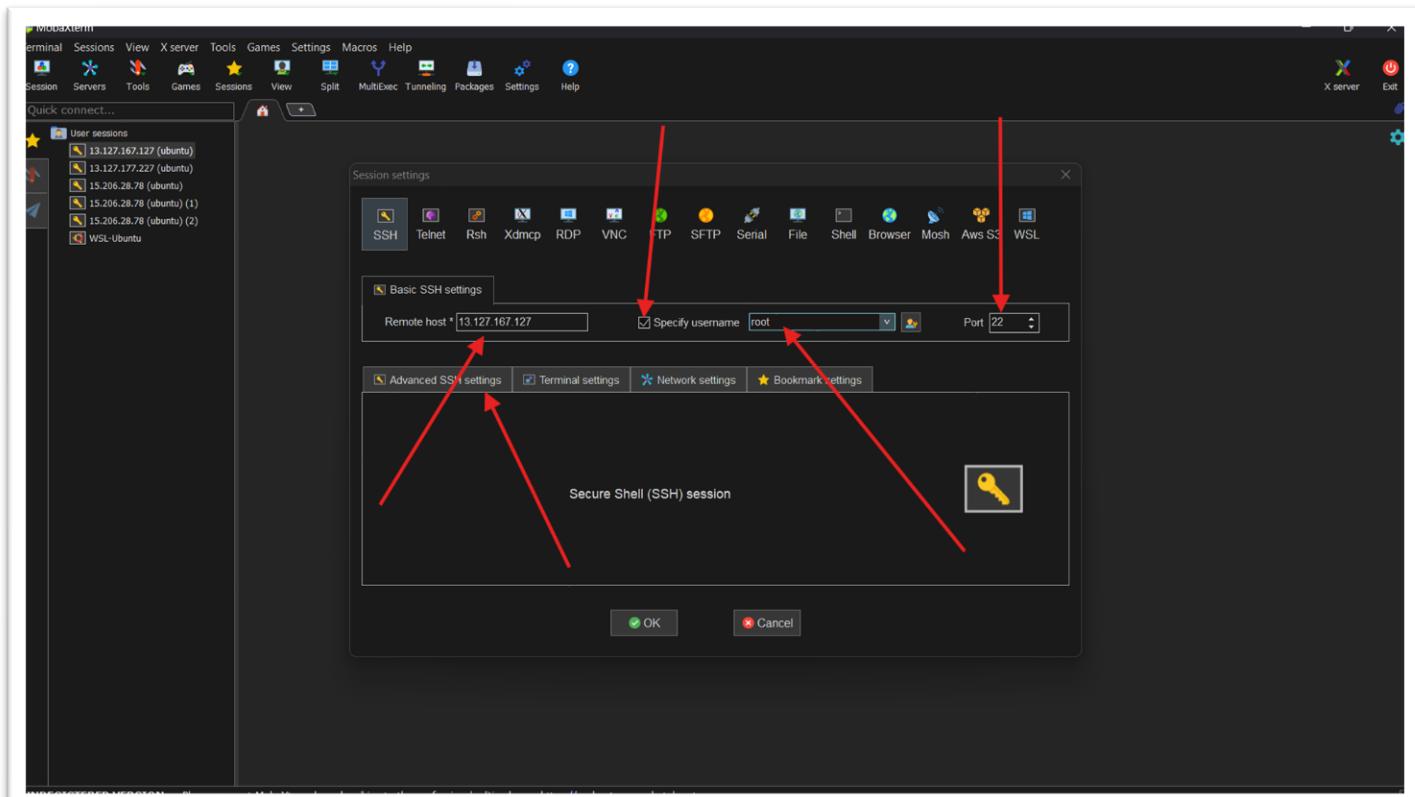


Fig. 1.21: Configuring MobaXterm

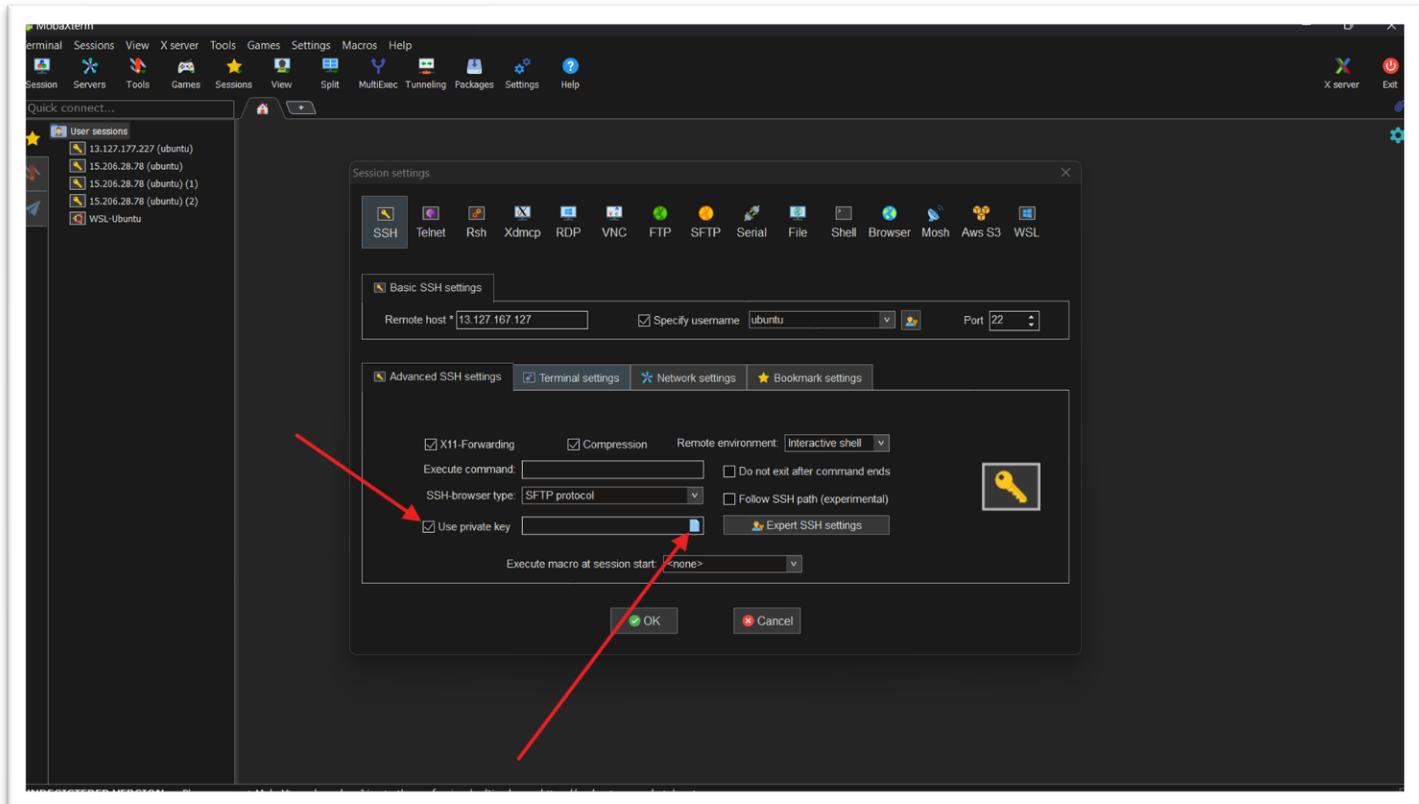


Fig. 1.22: Configuring MobaXterm advanced SSH settings

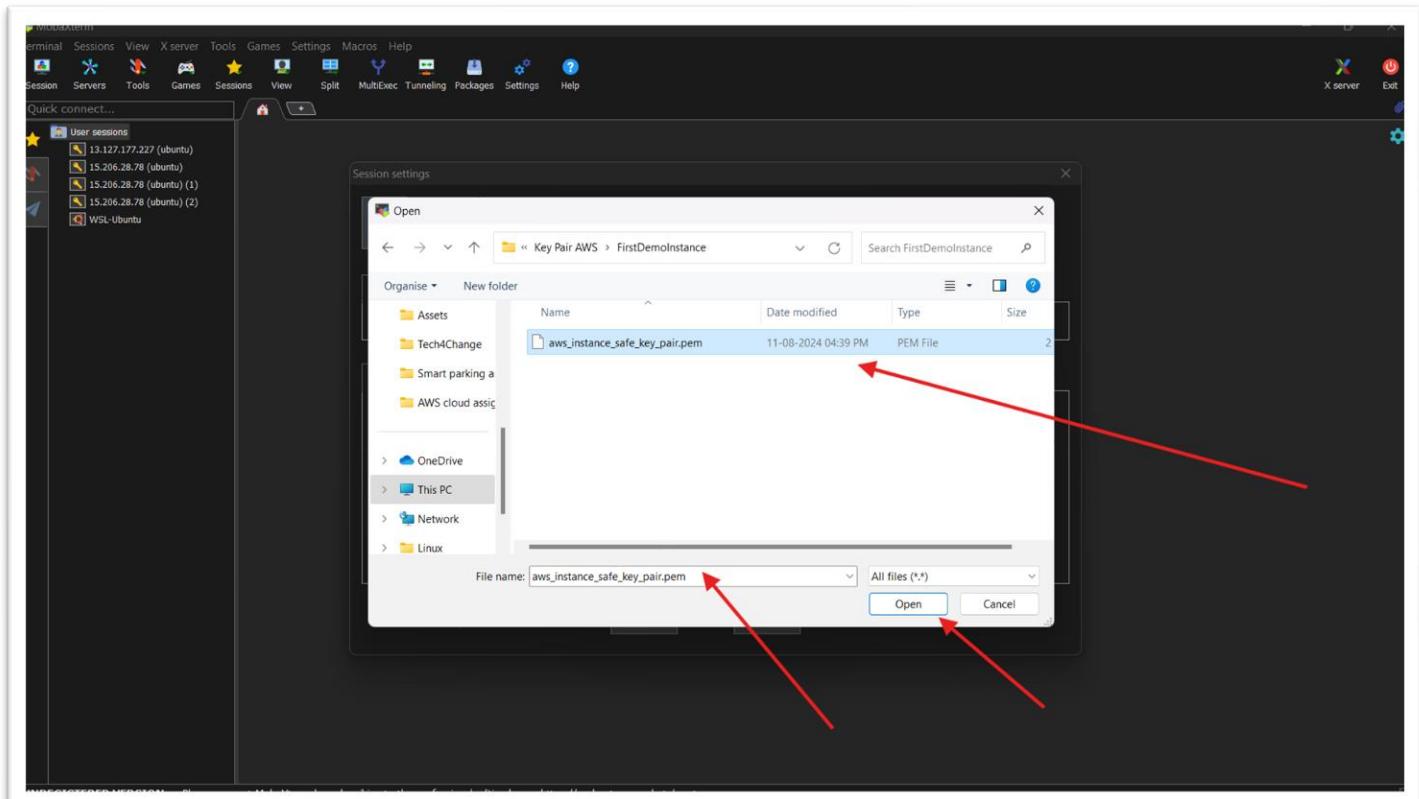


Fig. 1.23: Selecting key-pair to connect to instance in MobaXterm

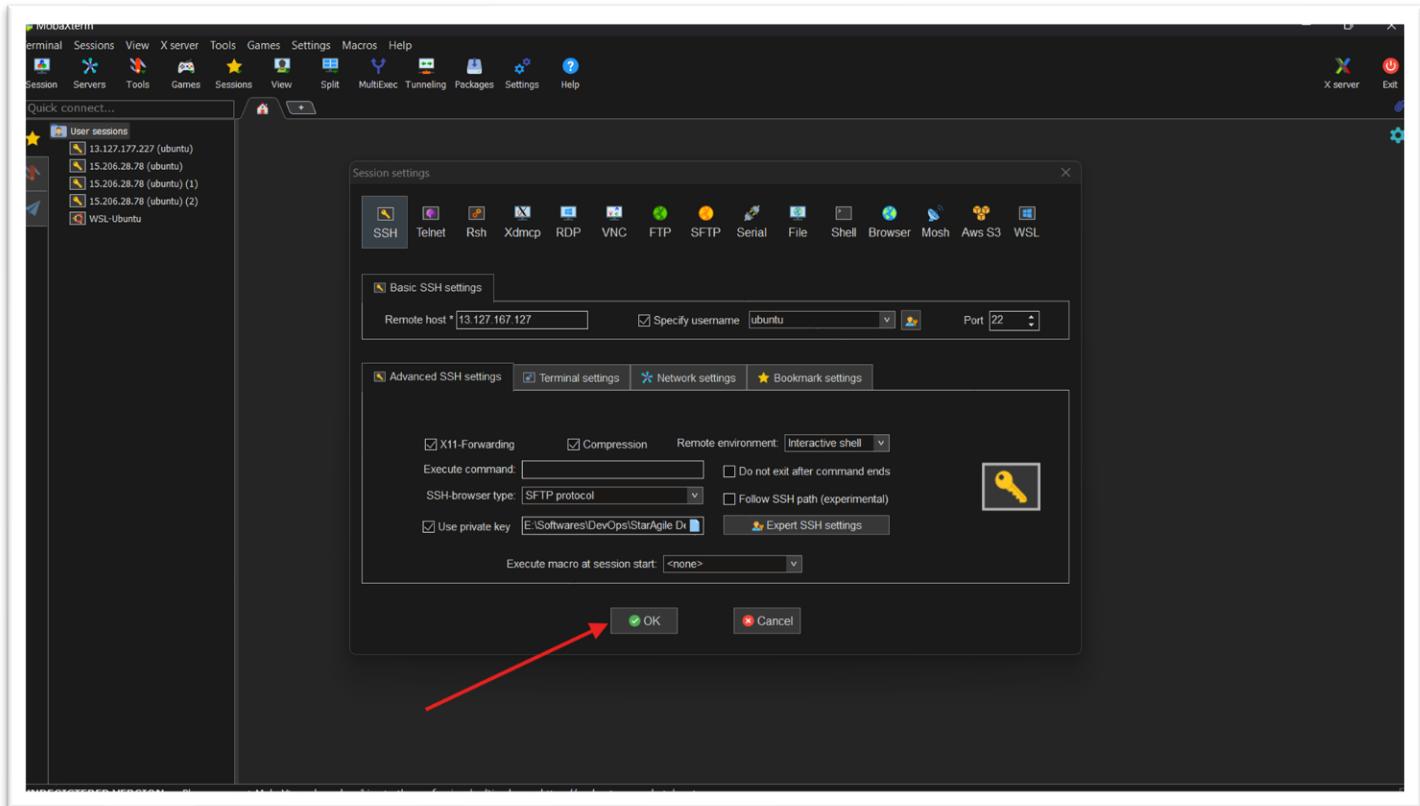


Fig. 1.23: Click Ok to connect to instance in MobaXterm through SSH

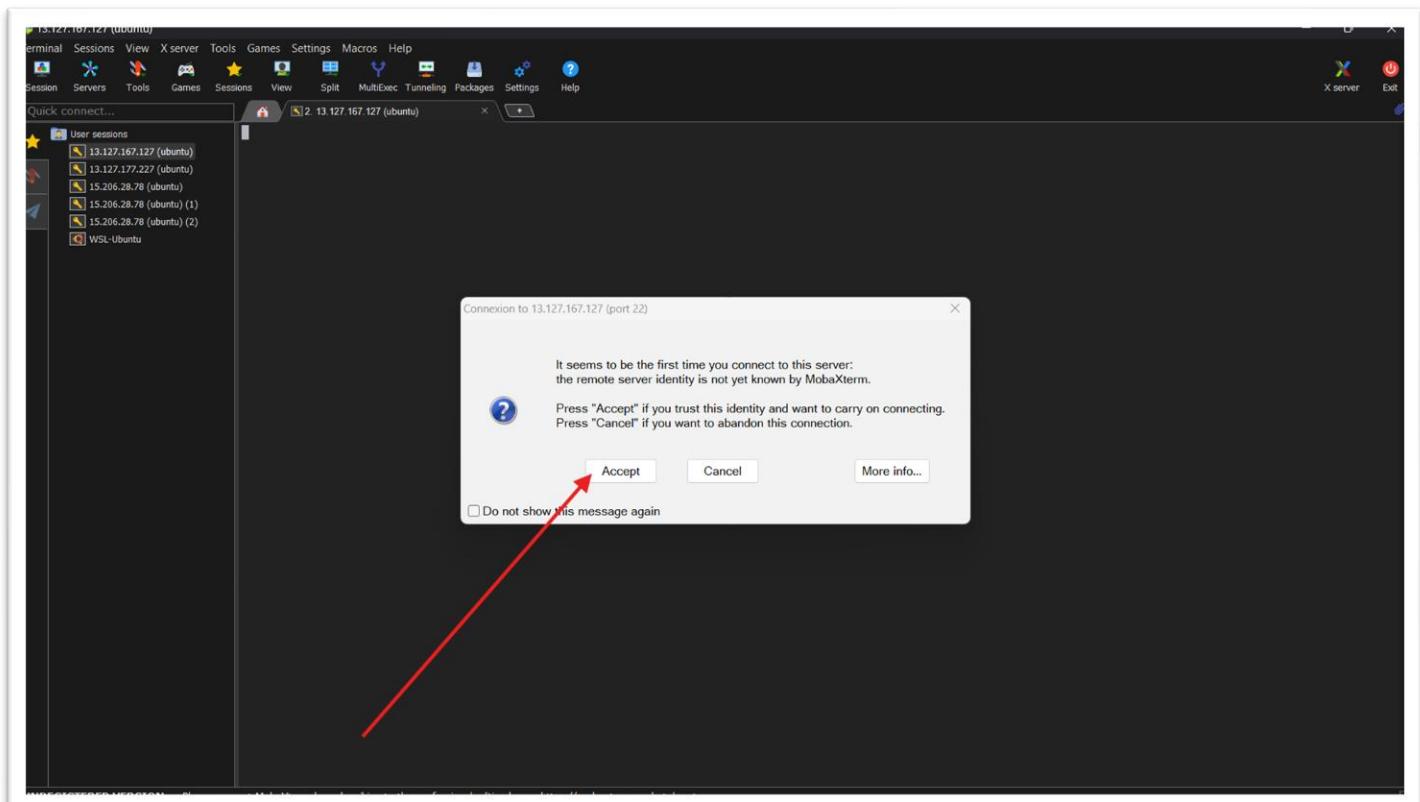


Fig. 1.24: Accept to connect to instance in MobaXterm to instance through SSH

```

13.127.107.127 (ubuntu)
terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
Session 4: 13.127.167.127 (ubuntu)
  • MobaXterm Personal Edition v24.2 •
  (SSH client, X server and network tools)

  ▶ SSH session to ubuntu@13.127.167.127
    • 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 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

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

System information as of Sun Aug 11 12:14:14 UTC 2024
System load: 0.0 Processes: 111
Usage of /: 23.1% of 6.71GB Users logged in: 0
Memory usage: 39% IPv4 address for enX0: 172.31.12.214
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

Last login: Sun Aug 11 12:06:15 2024 from 152.58.17.184
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-12-214:~$ █

```

*Fig. 1.25: Successfully connected to instance*

## 2. L2 - Login to AWS Console and Create IAM User, Role, and Group

Ans.

### 1. Access the AWS Management Console:

Visit the [AWS Management Console](#) and log in using your credentials.

### 2. A] Create IAM Groups:

#### i) Navigate to IAM:

In the AWS Management Console, search for "IAM" in the search bar and select IAM from the services dropdown.

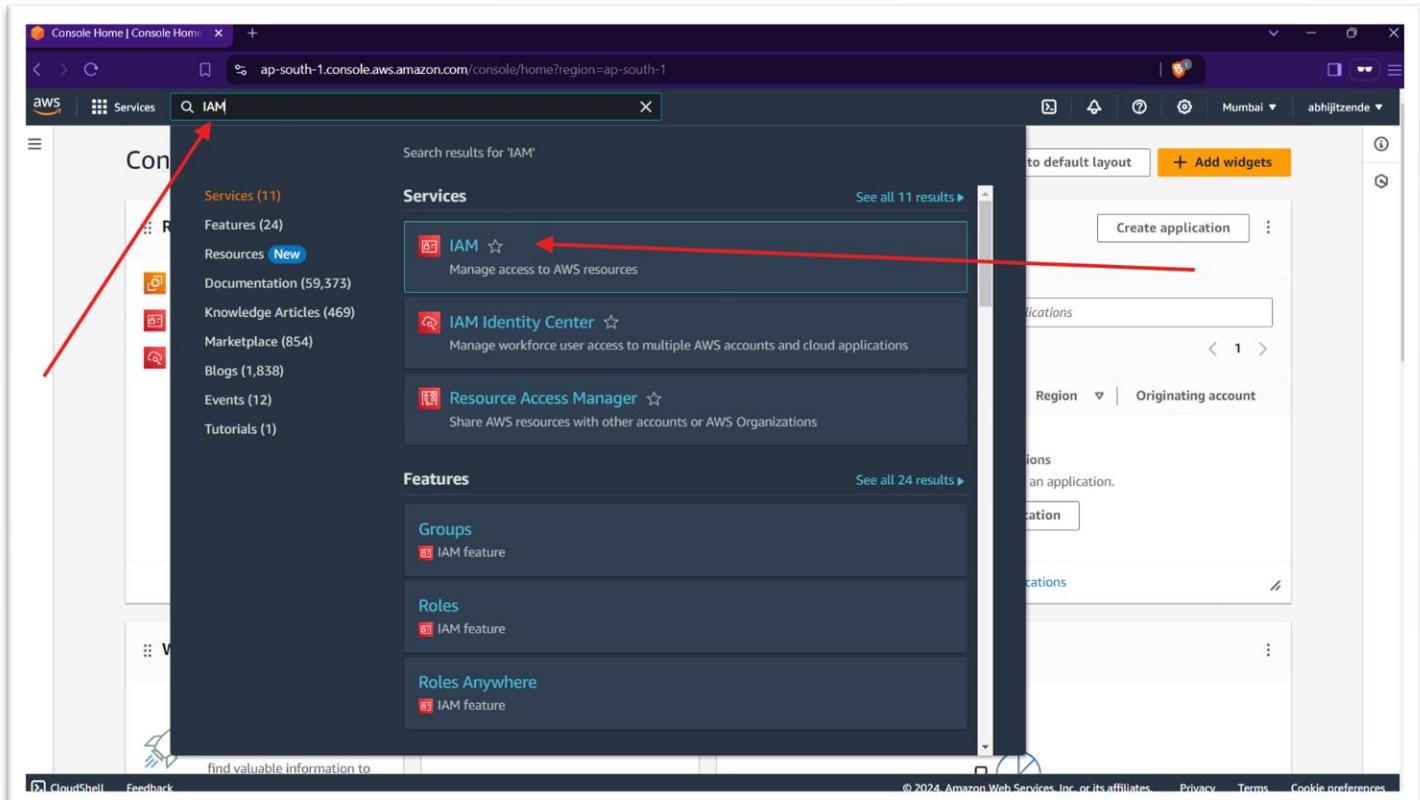


Fig. 2.1. Search IAM in search bar of AWS management console

ii) Create the Developer Group:

- In the IAM dashboard, click on "User groups" from 'Access management' which is on the left-hand menu
- Click "Create New Group".
- Enter the group name as "Developer".
- In the "Attach Permissions Policies" section, you can either choose an existing policy (e.g., "AmazonEC2FullAccess" for developers who need full access to EC2) or skip for now and attach policies later.
- Click "Create Group".

The screenshot shows the AWS IAM Dashboard. On the left, a sidebar menu includes 'User groups' under 'Access management'. A red arrow points from this menu item to the 'User groups' section in the main content area. The main content area features a 'Security recommendations' section with two items: 'Root user has MFA' (status: good) and 'Root user has no active access keys' (status: good). Below this is an 'IAM resources' section showing counts for User groups (0), Users (0), Roles (2), Policies (0), and Identity providers (0). At the bottom, there's a 'What's new' section and a footer with copyright information.

Fig. 2.2: IAM Dashboard

The screenshot shows the 'User groups' page under the IAM service. A red arrow points to the 'Create group' button in the top right corner of the table header. The table itself is currently empty, displaying the message 'No resources to display'. The left sidebar menu shows 'User groups' selected under 'Access management'.

Fig. 2.3: IAM > User group

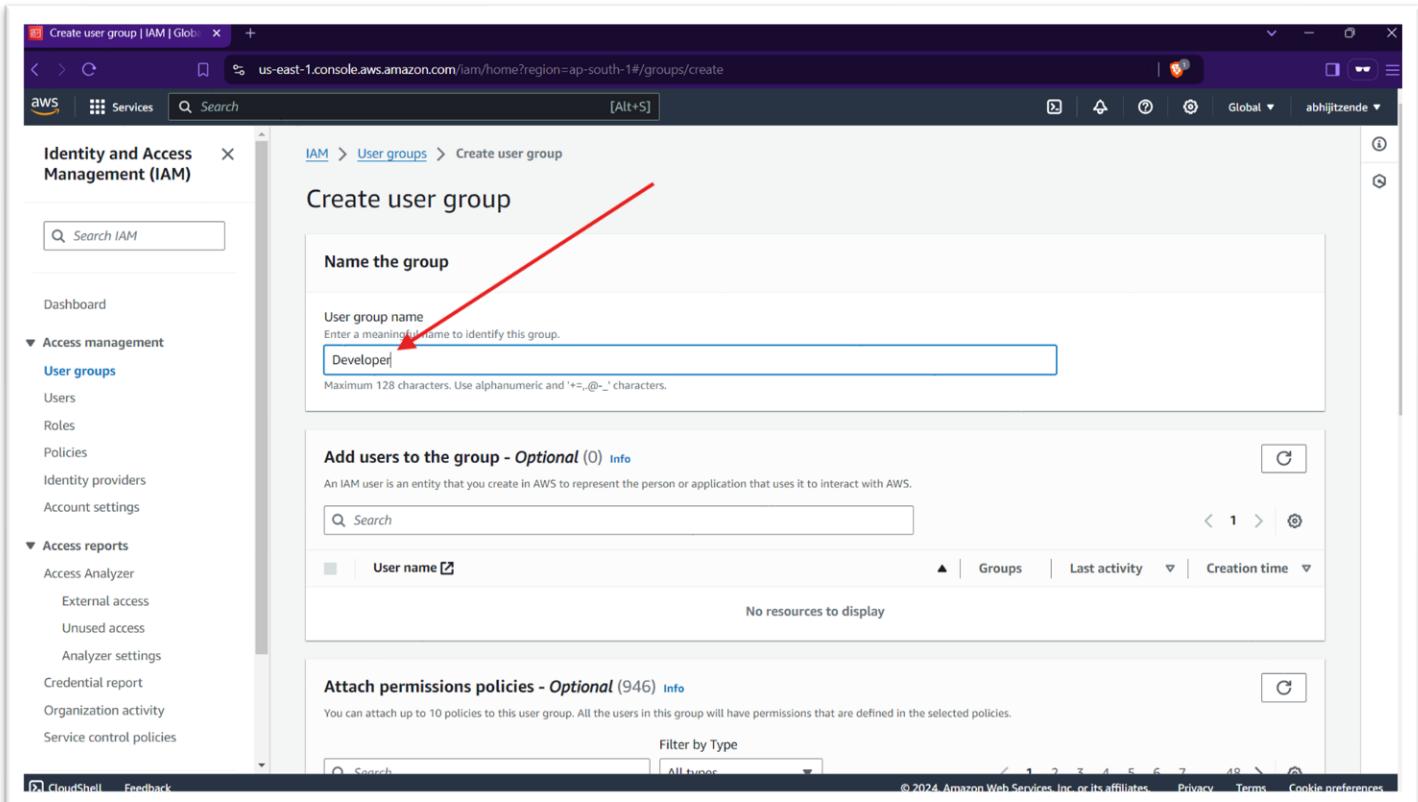


Fig. 2.4: IAM > User groups > Create user group

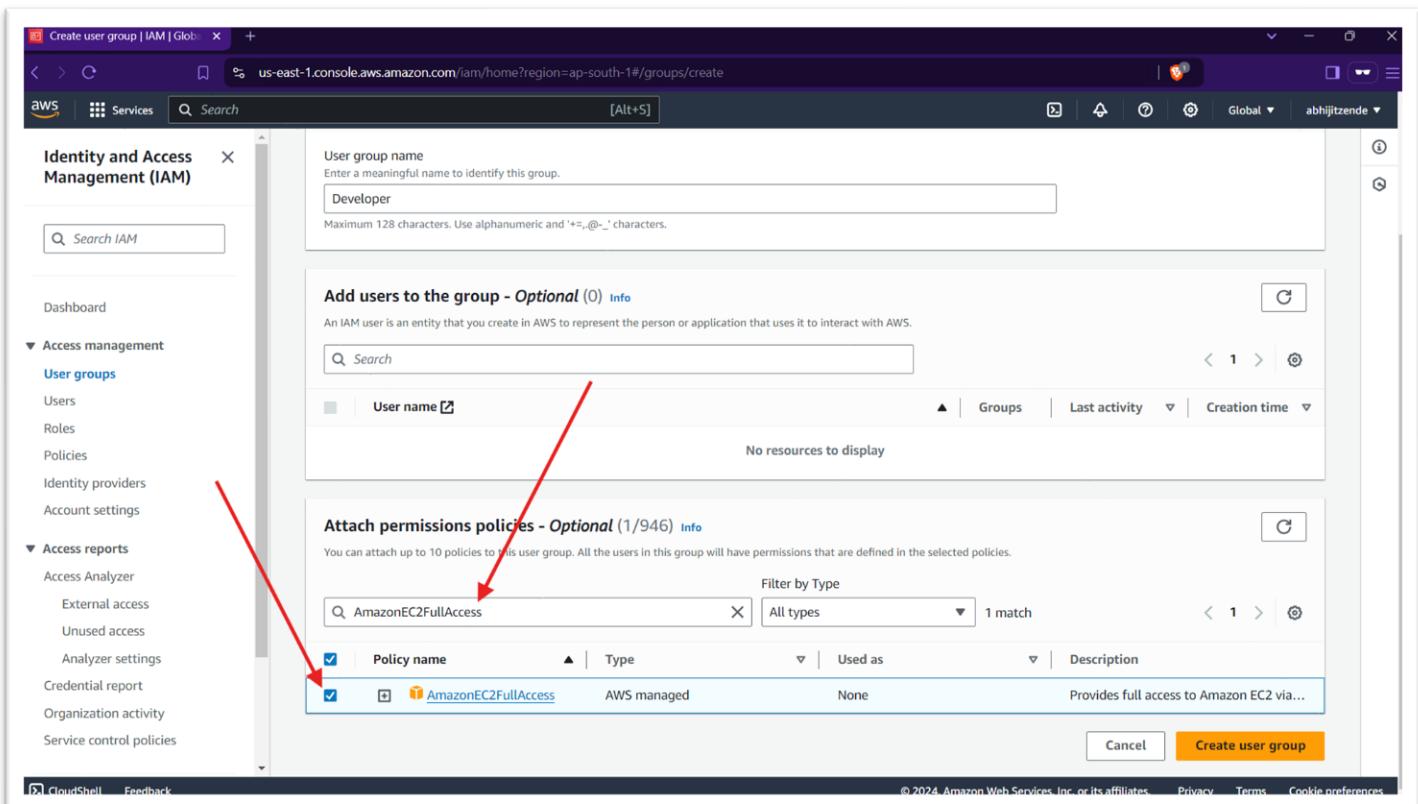


Fig. 2.5: IAM > User groups > Create user group [Attach required permission to group]

## B] Create IAM User:

### i) Create the User 'dev1':

- In the IAM dashboard, click on "Users" from the left-hand menu.
- Click "Add User".
- Enter "dev1" as the username.
- Under "Select AWS access type", select "Programmatic access" (for CLI, SDKs) and/or "AWS Management Console access" (for console access).
- If you select console access, set a custom password.

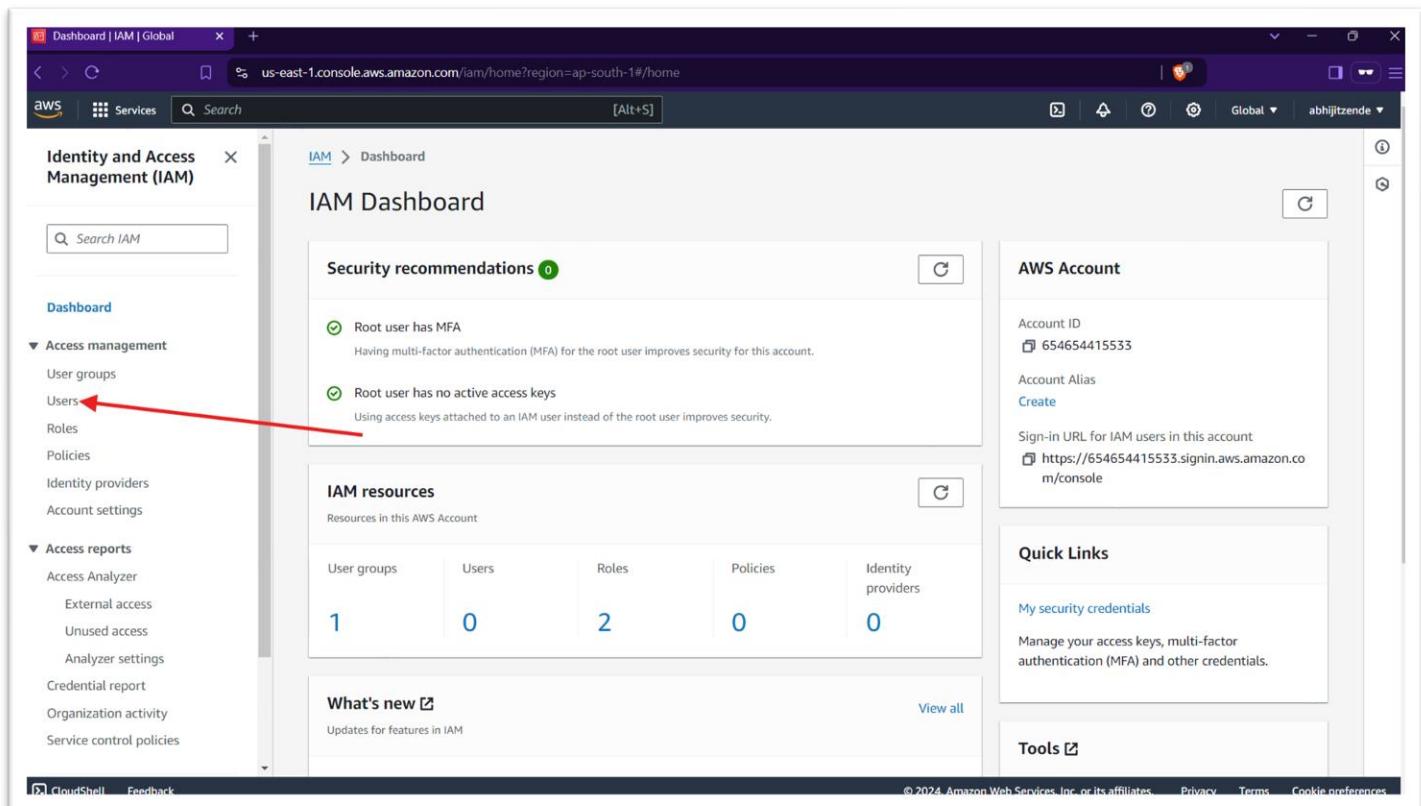


Fig. 2.6: IAM dashboard > Users

The screenshot shows the AWS IAM service in the AWS Management Console. The left sidebar is titled 'Identity and Access Management (IAM)' and includes sections for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access Analyzer, External access, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies). The main content area is titled 'Users (0) Info' and contains a table header with columns: User name, Path, Group, Last activity, MFA, Password age, and Console last sign-in. A red arrow points to the 'Create user' button in the top right corner of the main content area.

Fig. 2.7: Create user page

The screenshot shows the 'Create user' wizard in the AWS Management Console. The left sidebar shows steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area is titled 'Specify user details' and has a sub-section 'User details'. It shows a 'User name' field containing 'dev1'. Below it is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)'. There is a checked checkbox 'Provide user access to the AWS Management Console - optional' with a note: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' A large blue callout box highlights the 'I want to create an IAM user' option under 'Are you providing console access to a person?'. A red arrow points from the text above to this callout box. At the bottom, there are options for 'Console password' (Autogenerated or Custom password) and a 'Create user' button.

Fig. 2.8: Specify user details page

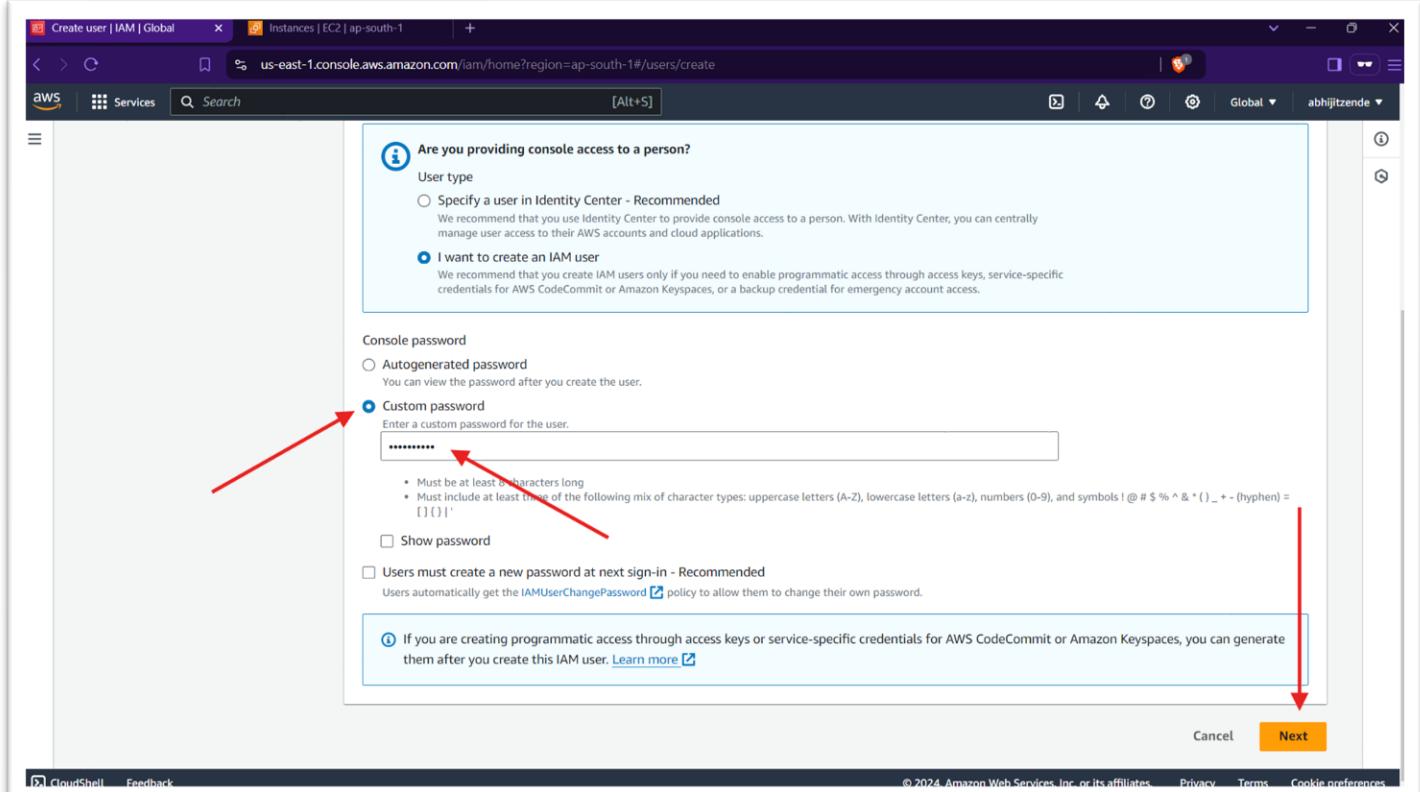


Fig. 2.9: Specify user details page

iii) Attach 'dev1' to the Developer Group:

- In the "Set permissions" section, choose "Add user to group".
- Select the "Developer" group created earlier.
- Click "Next: Tags", then "Next: Review", and finally "Create user".

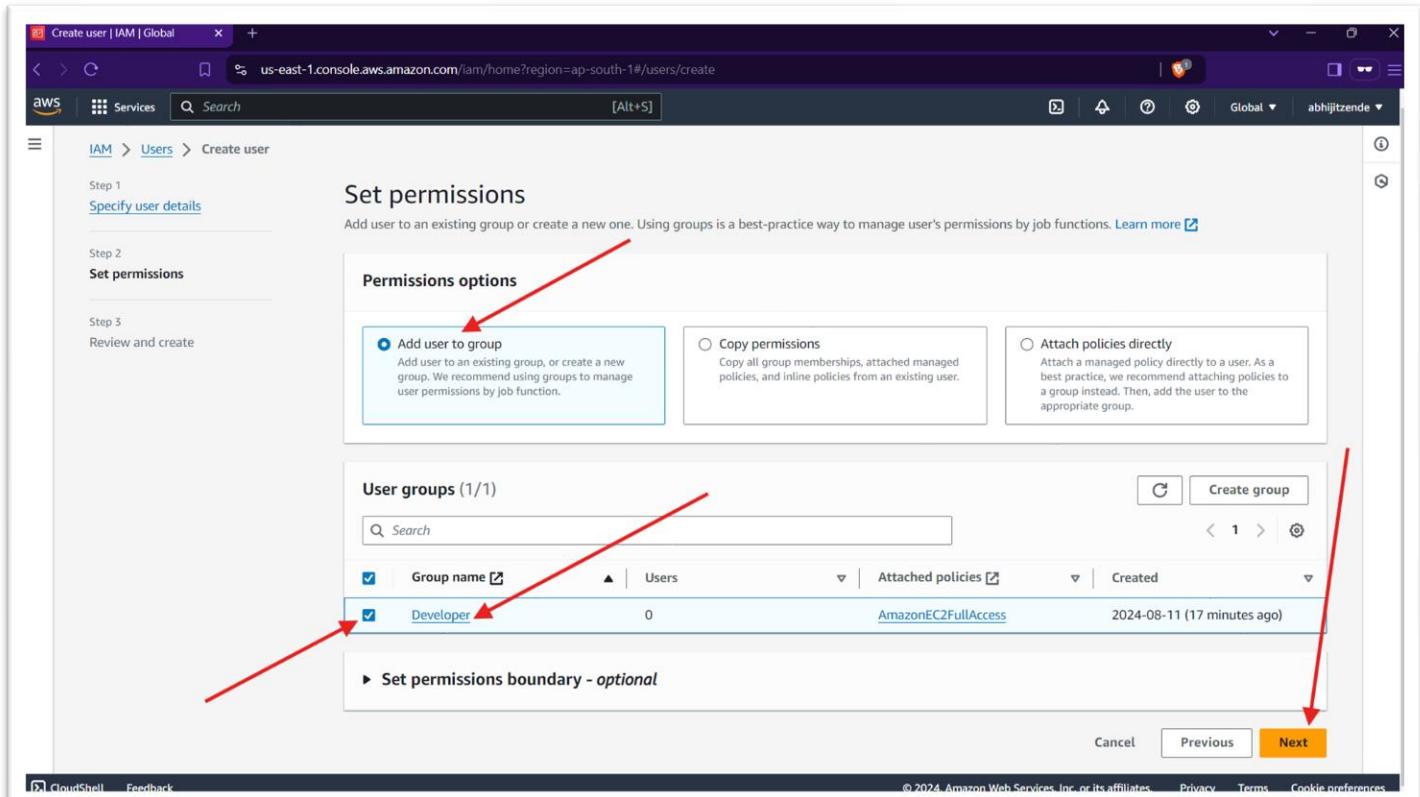


Fig. 2.10: Set permission of IAM > Users > Create user

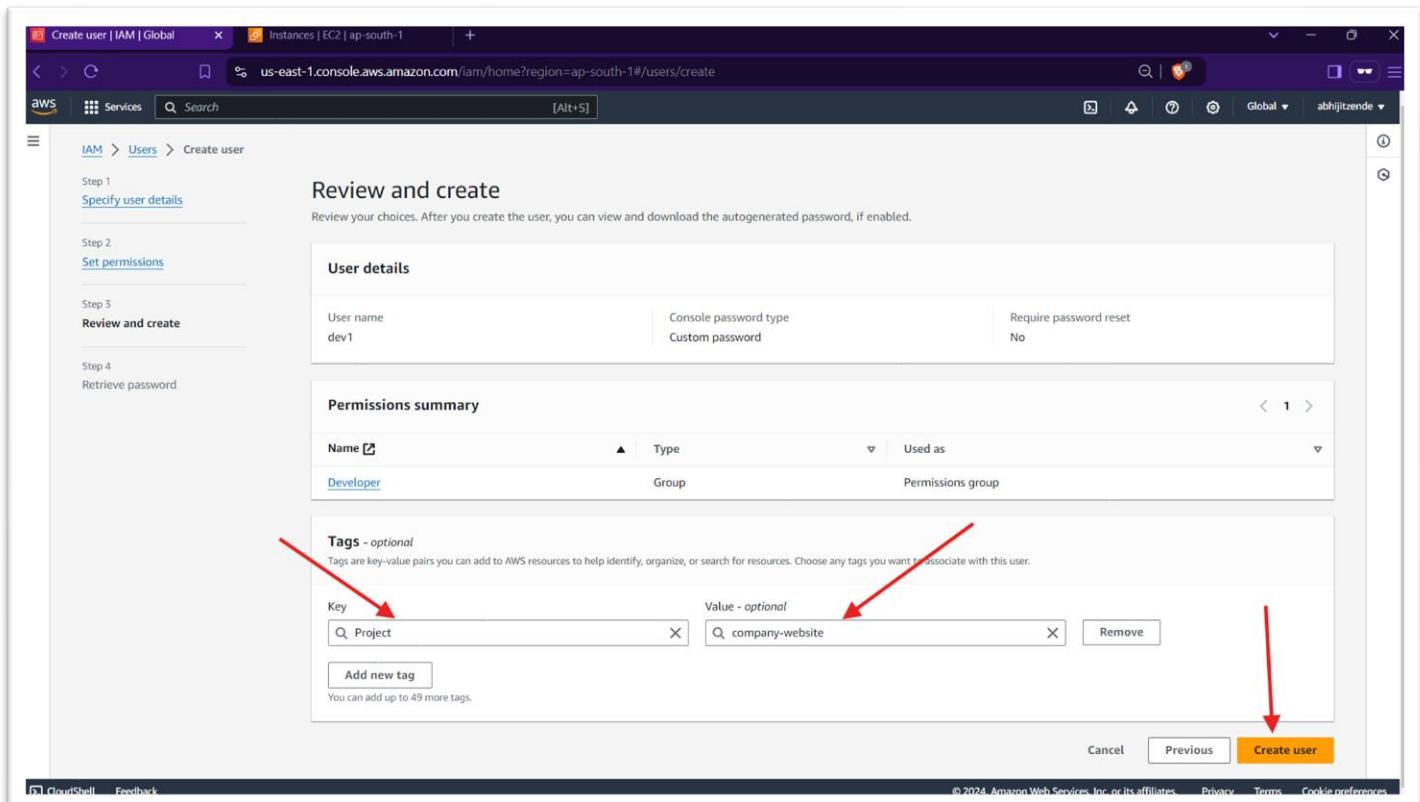


Fig. 2.11: Review and create page

iv) Download the Credentials:

- Once the user is created, download the .csv file containing the access key, secret key, and login information for the 'dev1' user.

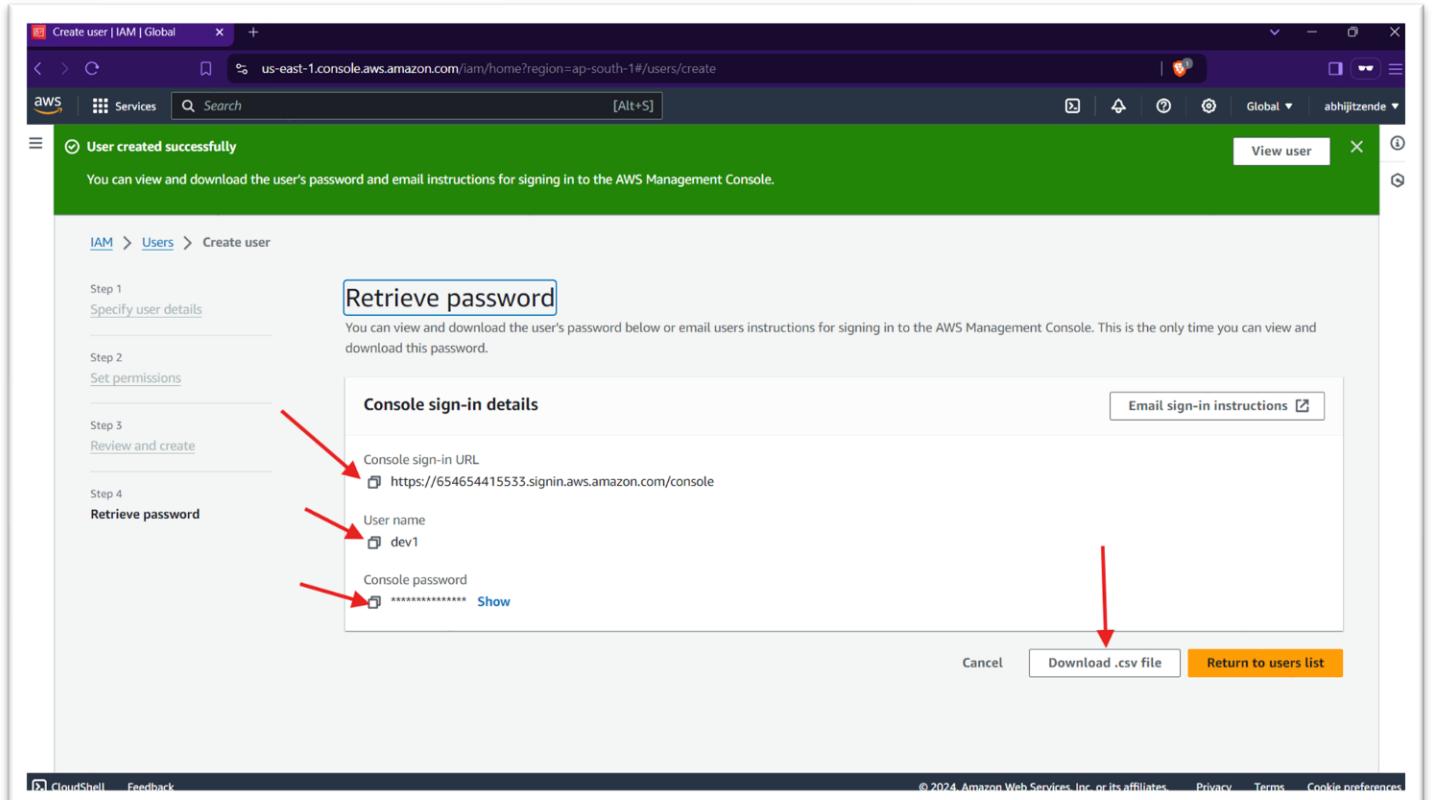


Fig. 2.12: Retrieve password page

C] Create role:

i) Create a Role:

- In the IAM dashboard, click on "Roles" from the left-hand menu.
- Click "Create role".
- Choose the AWS service that will use this role (e.g., EC2 for granting permissions to EC2 instances).
- Click "Next: Permissions".

The screenshot shows the AWS IAM Roles page. On the left, a navigation menu is open under 'Access management', with 'Roles' highlighted by a red arrow. The main area displays a table of roles:

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-

At the top right, there is a 'Create role' button, which is also highlighted by a red arrow. Below the table, there are sections for 'Roles Anywhere' and 'Temporary credentials'.

Fig. 2.13: IAM > Roles

The screenshot shows the 'Create role' wizard, Step 1: Select trusted entity. The left sidebar shows steps 1, 2, and 3: 'Select trusted entity', 'Add permissions', and 'Name, review, and create'. The main area is titled 'Select trusted entity' and contains a 'Trusted entity type' section with five options:

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

Below this is a 'Use case' section with the sub-section 'Service or use case' containing a dropdown menu labeled 'Choose a service or use case'. A red arrow points from the 'Custom trust policy' option to the 'Service or use case' dropdown.

Fig. 2.14: Create role page

Screenshot of the AWS IAM 'Create role' wizard Step 1: Select trusted entity. The 'Trusted entity type' section shows four options: AWS service (selected), AWS account, SAML 2.0 federation, and Web identity. Below it, the 'Use case' section shows a dropdown for 'Service or use case' containing 'EC2'. A red arrow points from the 'Service or use case' dropdown to the 'EC2' option under 'Use case'.

## 2.15: Selecting Use case of role

Screenshot of the AWS IAM 'Create role' wizard Step 1: Select trusted entity. The 'Use case' section shows a dropdown for 'Service or use case' containing 'EC2'. Under 'Use case', the 'EC2' option is selected (indicated by a blue circle). A red arrow points to the 'EC2' option. Another red arrow points to the 'Next' button at the bottom right.

## 2.16: Selecting Use case of role

## ii) Attach Policies to the Role:

- Select a policy that suits the role's purpose (e.g., "AmazonS3FullAccess" if the role requires full access to S3).
- Click "Next: Tags", then "Next: Review".

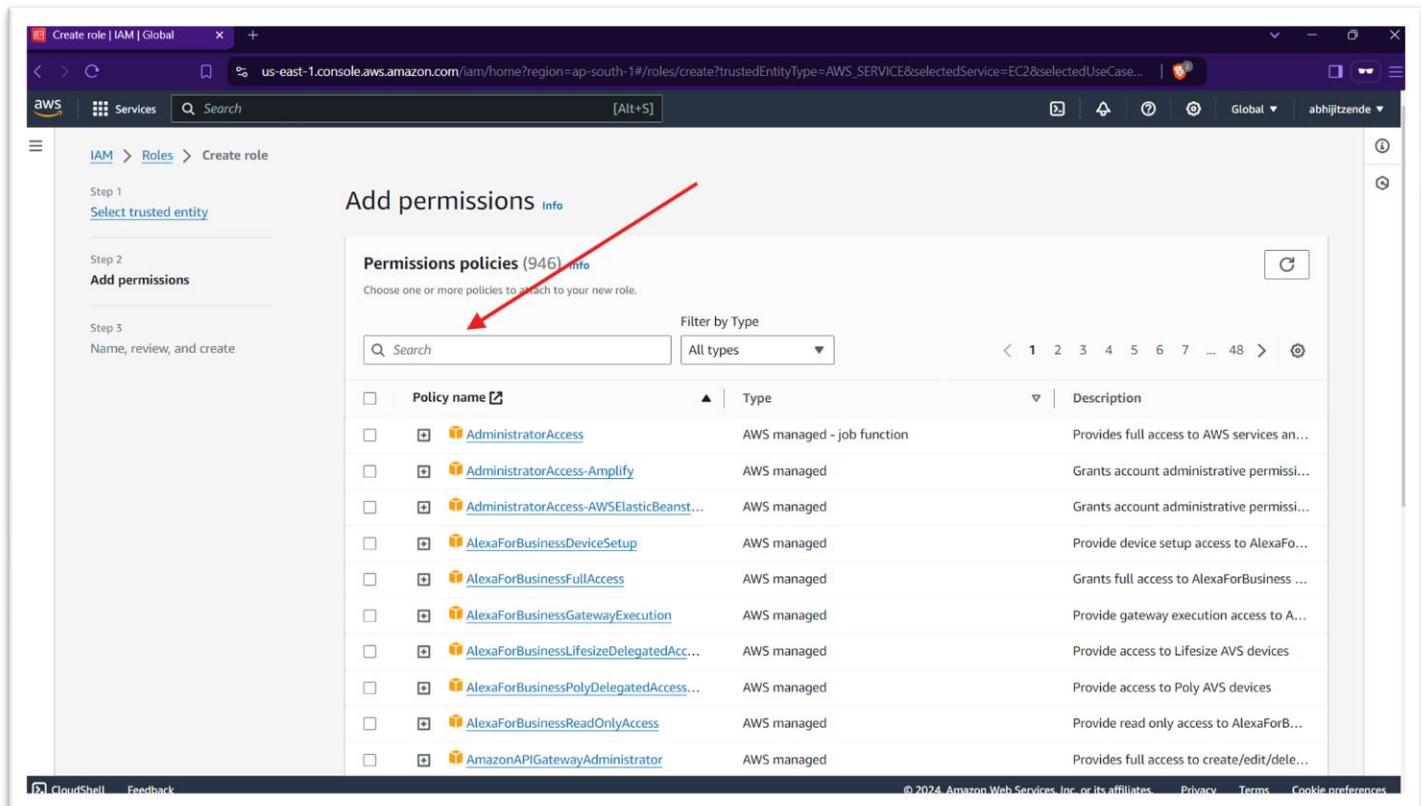


Fig. 2.21: IAM > Roles > Create roles. Add permissions

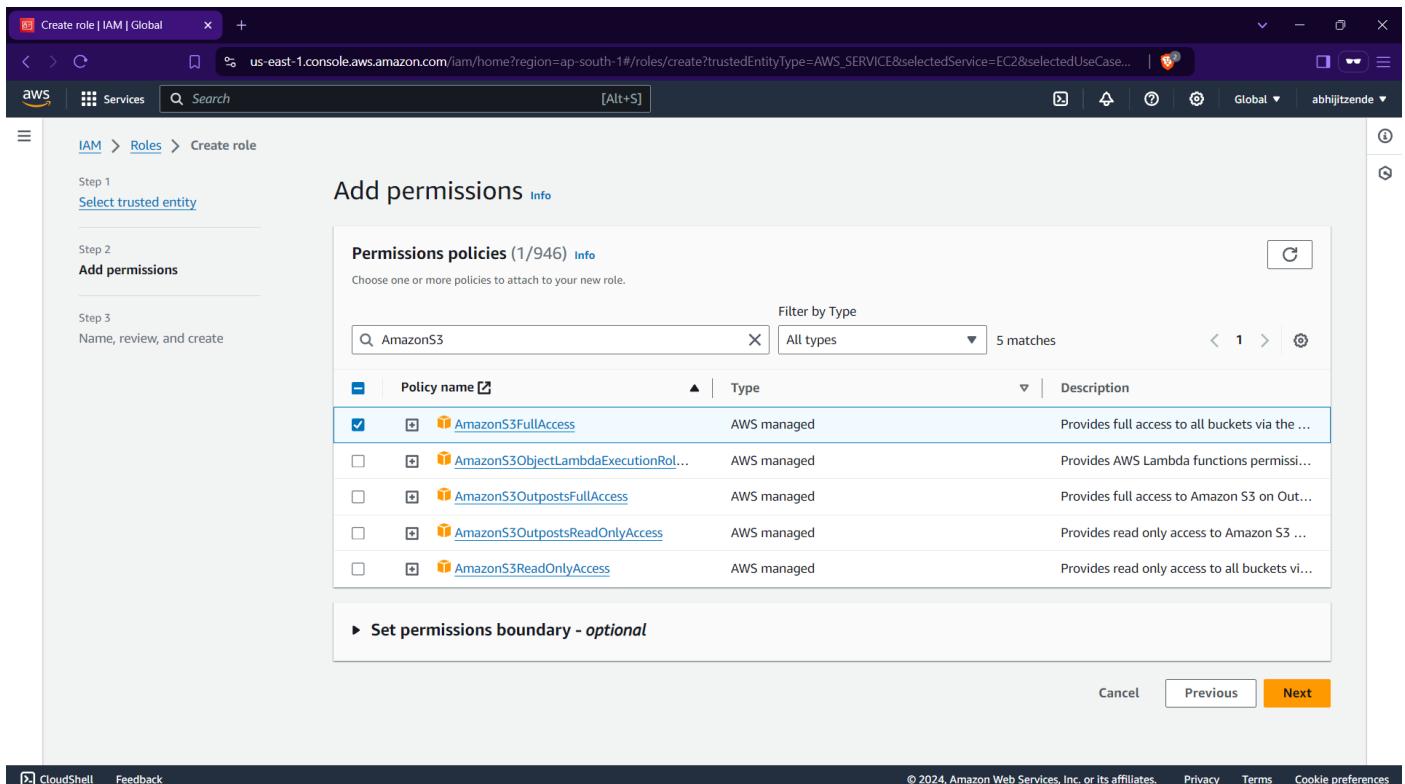
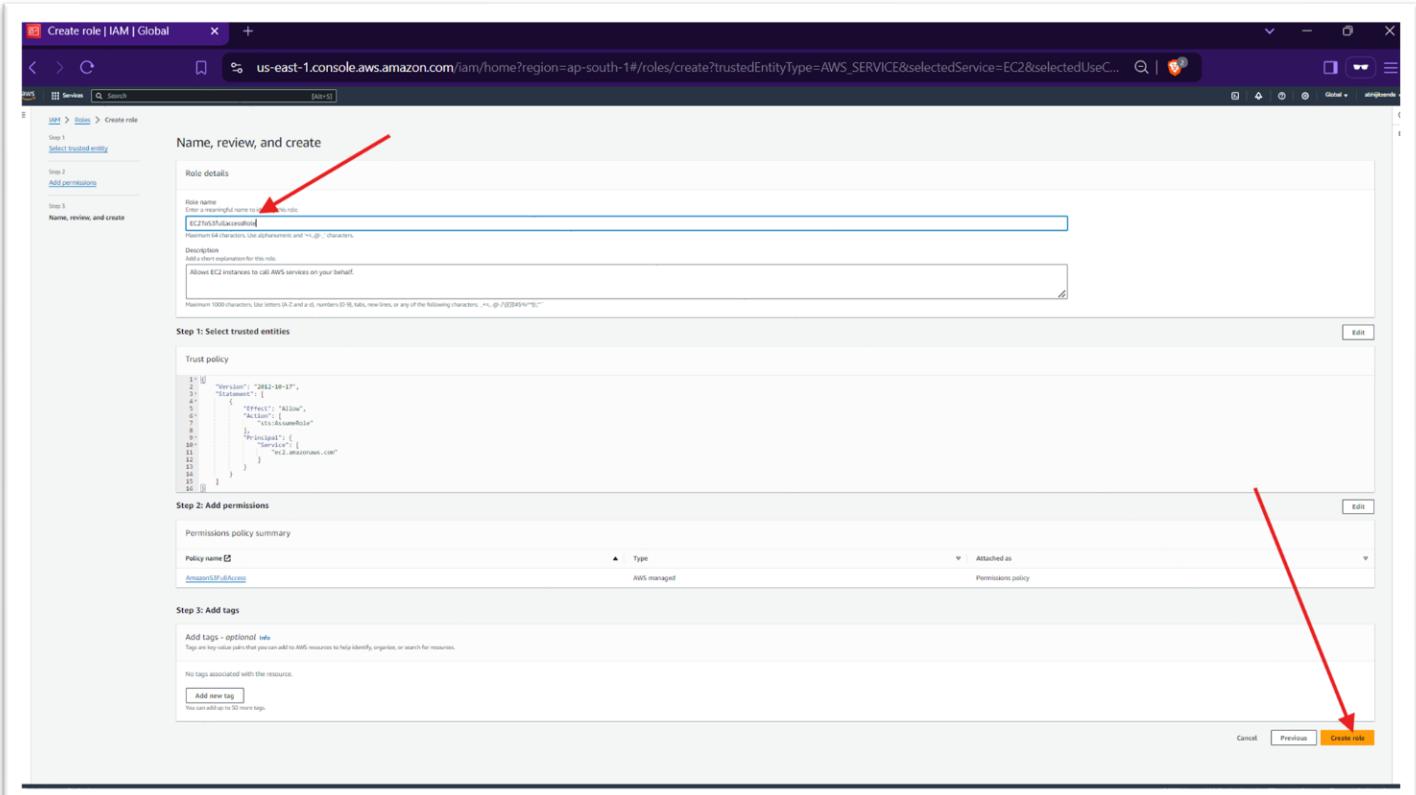


Fig. 2.23: IAM > Roles > Create roles. Add permissions

### iii) Name and Create the Role:

- Enter a role name (e.g., "EC2S3AccessRole").
- Click "Create role".



*Fig. 2.24: IAM > Roles > Create role. Name review and create*

The screenshot shows the details page for the 'EC2ToS3fullaccessRole'. The left sidebar shows 'Identity and Access Management (IAM)'. The main content area displays the 'Summary' of the role, including creation date (August 11, 2024), ARN (arn:aws:iam::654654415533:role/EC2ToS3fullaccessRole), and instance profile ARN (arn:aws:iam::654654415533:instance-profile/EC2ToS3fullaccessRole). Below the summary, the 'Permissions' tab is selected, showing one attached policy: 'AmazonS3FullAccess'. There is a 'Permissions policies (1)' section with a 'Search' bar and a 'Filter by Type' dropdown set to 'All types'. A red arrow points to the 'Add permissions' button in the top right corner of this section.

*Fig. 2.25: After role creation*

### 3. L3 - Launch AWS EC2 Ubuntu Instance and configure the Security Group - Inbound Rule: 8080. Justify the usage of Inbound Rules

Ans.

#### 1. Launching an AWS EC2 Ubuntu Instance:

Follow the steps mentioned in L1 ans until ‘Network settings’ by first signing in using newly created IAM user. Also now add the tag as key: Name and value: MyWebSever

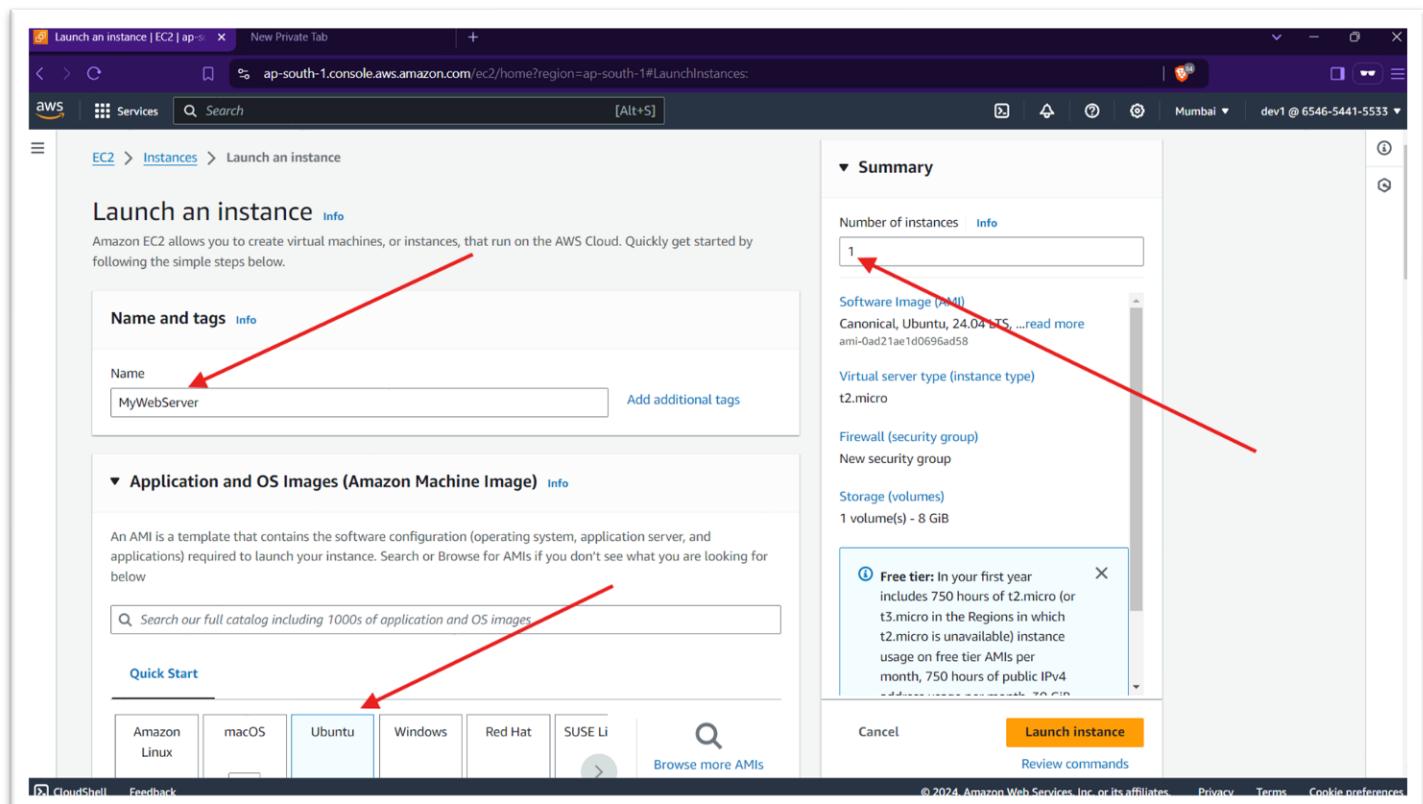


Fig. 3.01: EC2 > Instances > Launch an instance

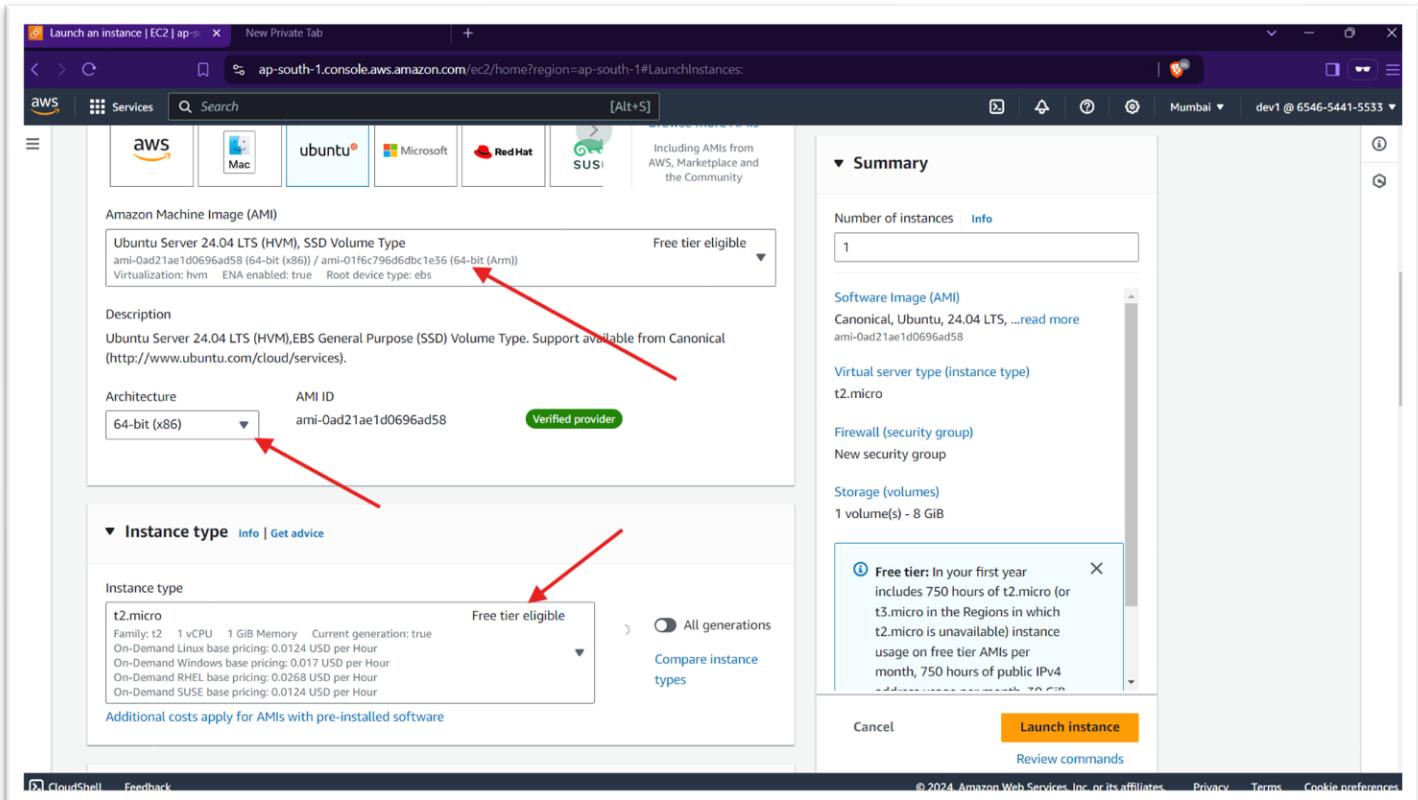


Fig. 3.02: EC2 > Instances > Launch an instance

## 2. Configure Security Group - Inbound Rule for Port 8080:

### i) Create a Security Group:

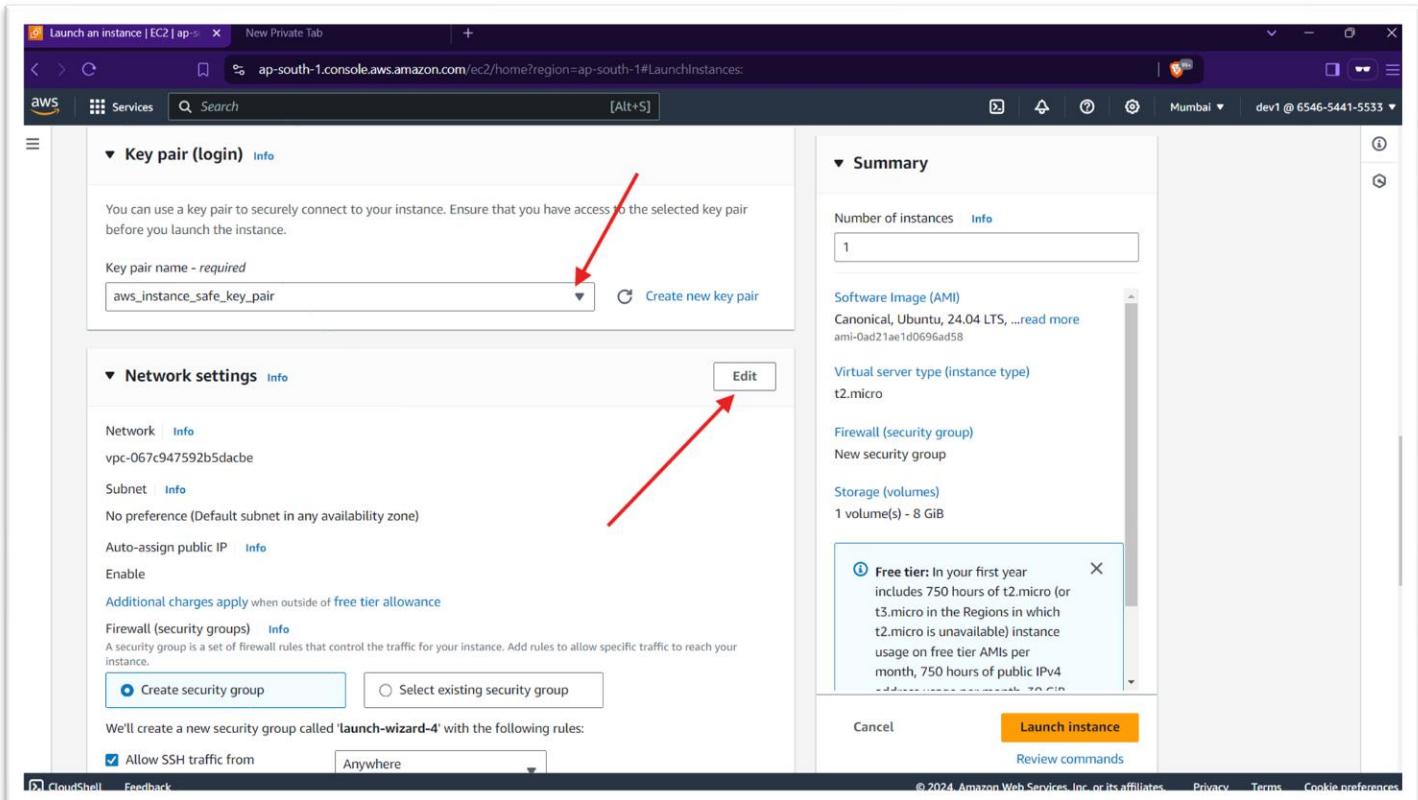
- Under the "Configure Security Group" section, create a new security group.
- Name the Security Group (e.g., "WebServerSG").
- In the "Inbound rules" section, click "Add Rule".

### ii) Set Inbound Rule for Port 8080:

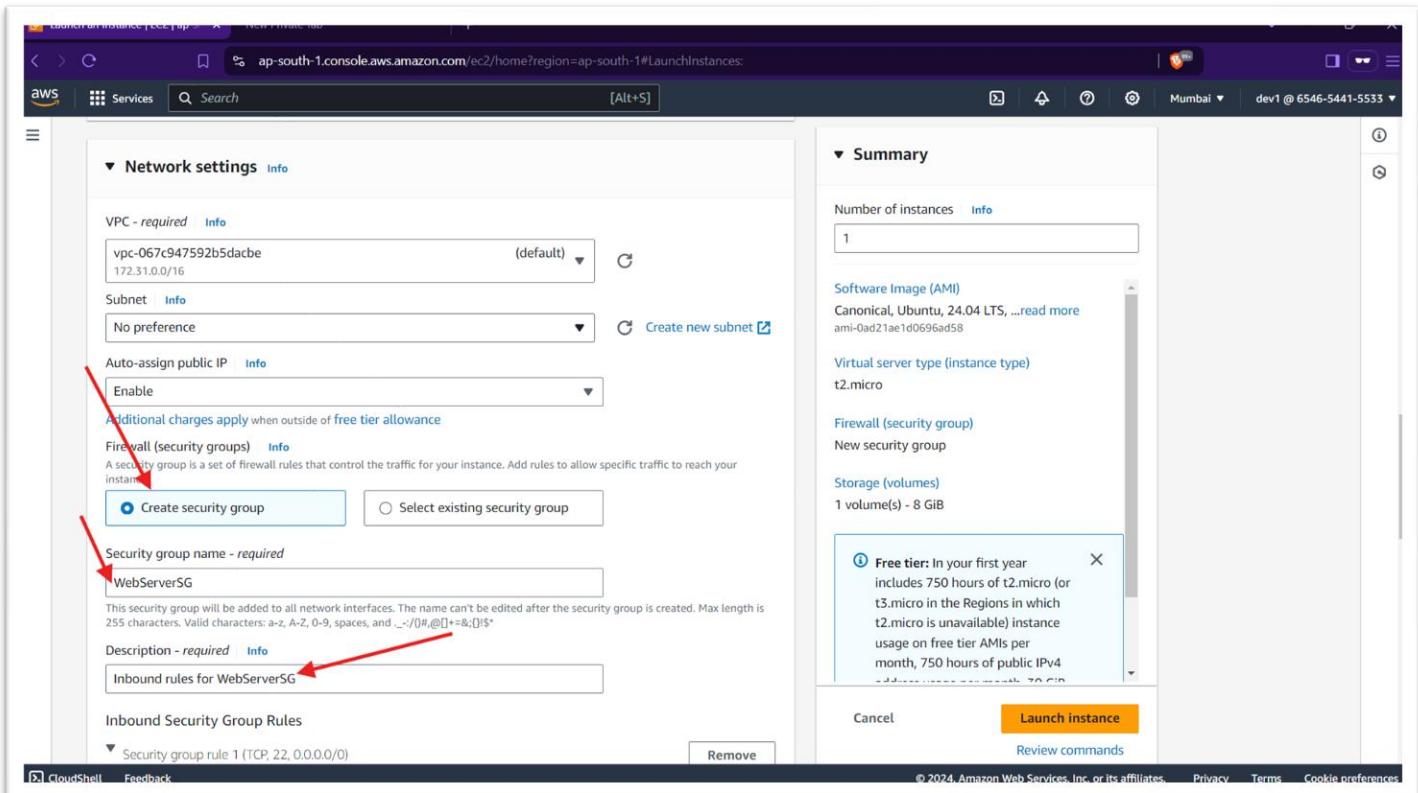
- Type: Custom TCP Rule
- Protocol: TCP
- Port Range: 8080
- Source: Custom (enter your IP address or choose "Anywhere" for testing purposes)
- This rule allows inbound traffic to reach your instance on port 8080, commonly used by web applications and servers.

### iii) Review and Launch:

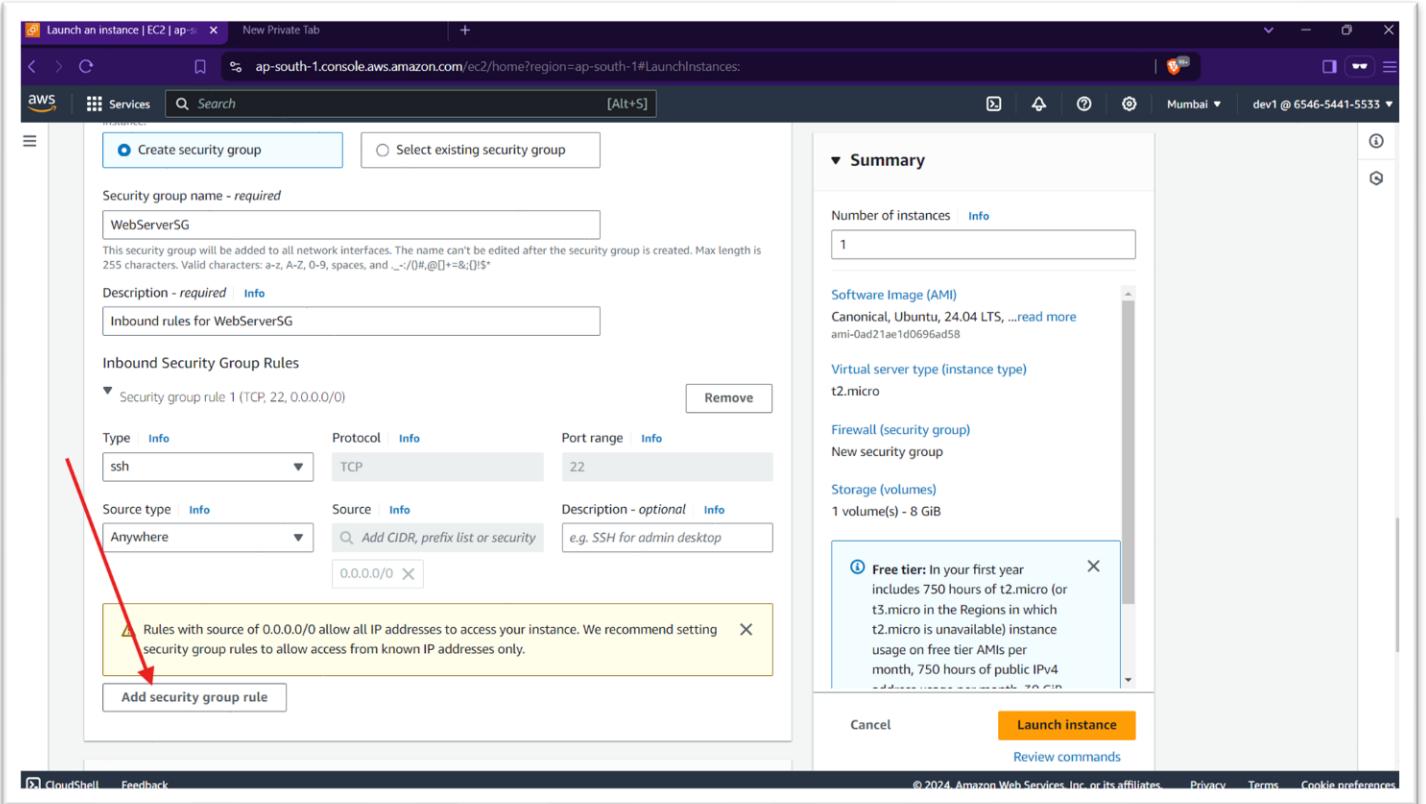
Launch the instance after successfully reviewing it.



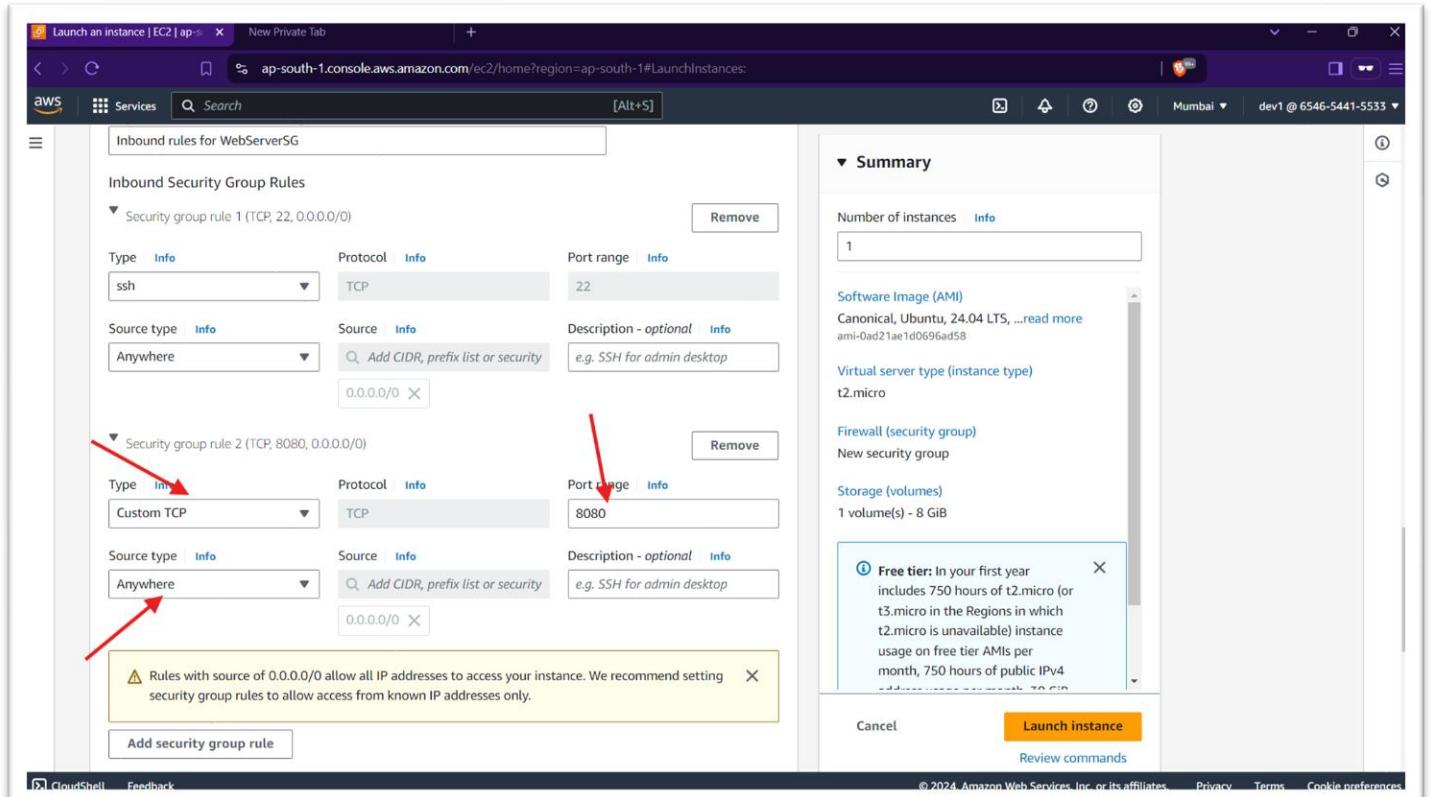
*Fig. 3.03: Edit Network settings*



*Fig. 3.04: Figure illustrating creation of new security group*



*Fig. 3.05: Figure illustrating creation of new security group*



*Fig. 3.06: Figure illustrating creation of new security group*

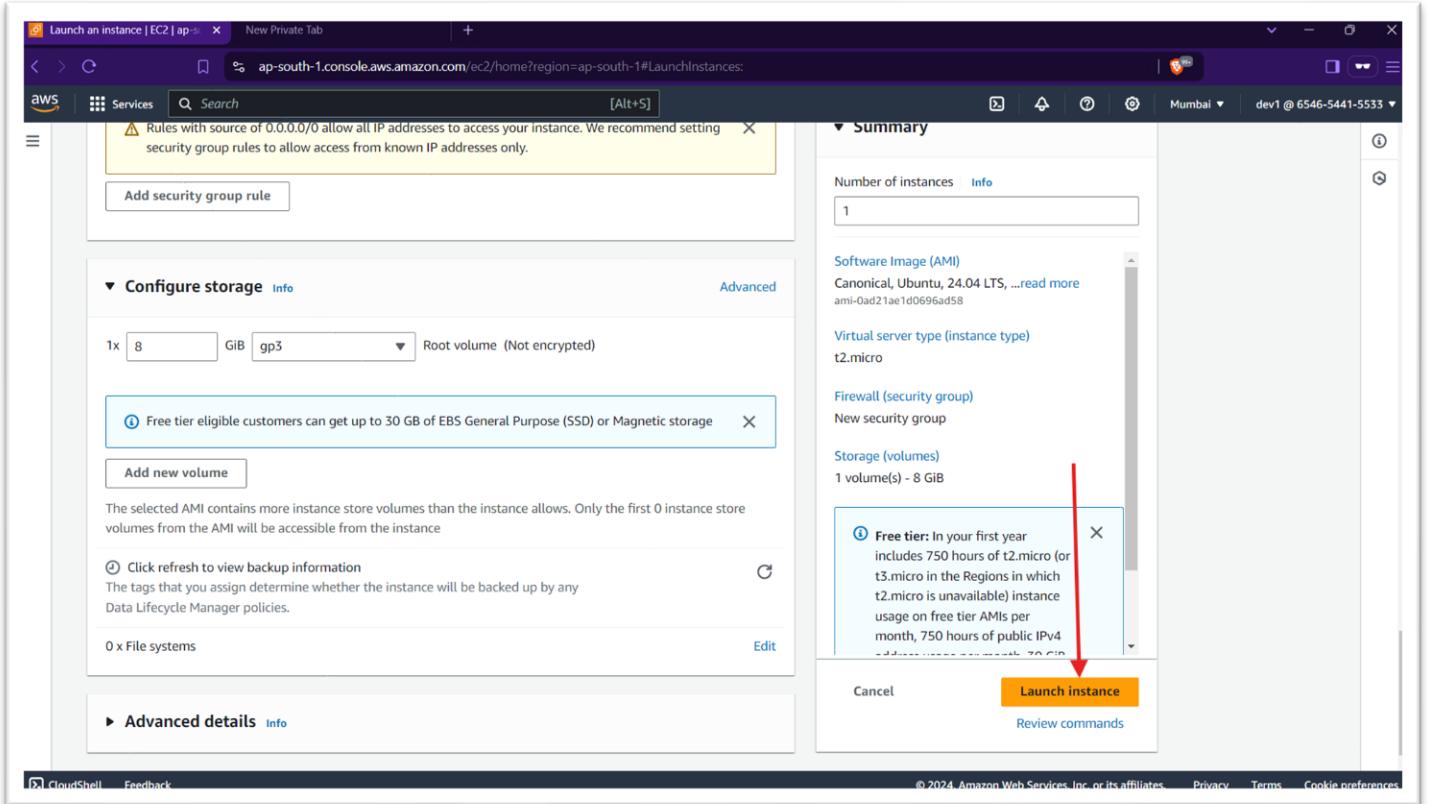


Fig. 3.07: EC2 > Instances > Launch new instances

### 3. Justification of Inbound Rules:

#### A] Purpose of Inbound Rules:

- Inbound rules in a security group control the incoming traffic to your EC2 instance. These rules determine who can connect to your instance and on which ports.

#### B] Justification for Port 8080:

- Web Servers & Applications: Port 8080 is commonly used for web servers or web applications, particularly in development environments. Configuring this port allows you to host web services that can be accessed by users or other services.
- Security Considerations: By specifying port 8080 in the inbound rules, you limit access to only the necessary service, reducing the attack surface of your instance. Unnecessary ports can remain closed to prevent unauthorized access.
- Controlled Access: The inbound rule also allows you to control the source of the traffic, meaning you can limit access to your web

application to specific IP addresses or ranges, further enhancing security.

### C] General Security Group Best Practices:

- Principle of Least Privilege: Only open the ports you absolutely need. If port 8080 is the only one required, ensure that others are closed unless necessary.
- Monitoring and Auditing: Regularly review and audit your security group settings to ensure they align with your security policies.

## 4. L4 - Connect to the AWS EC2 Ubuntu Instance and Update default packages, install JDK, Maven, Git, and validate the versions

Ans.

### 1. Connect to the AWS EC2 Ubuntu Instance:

- Go to the EC2 Dashboard in the AWS Management Console.
- Select your running instance and copy the Public DNS (or IPv4 address).

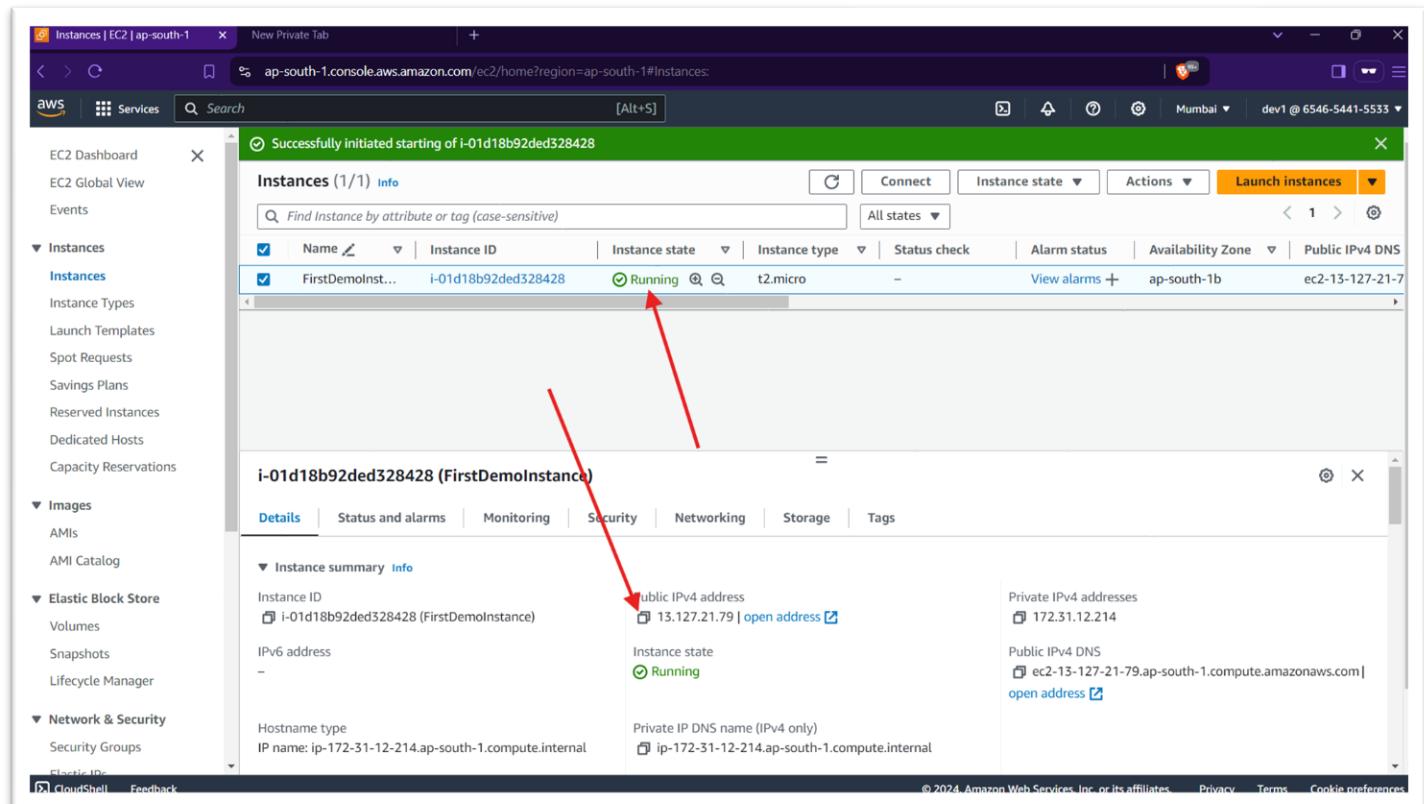


Fig. 4.01: EC2 > Instances

## 2. Connect to the running instance:

Follow L1 ans ‘Load key and pair’ step to connect to running EC2 instance using MobaXterm

## 3. Update Default Packages:

### i) Update Package Lists:

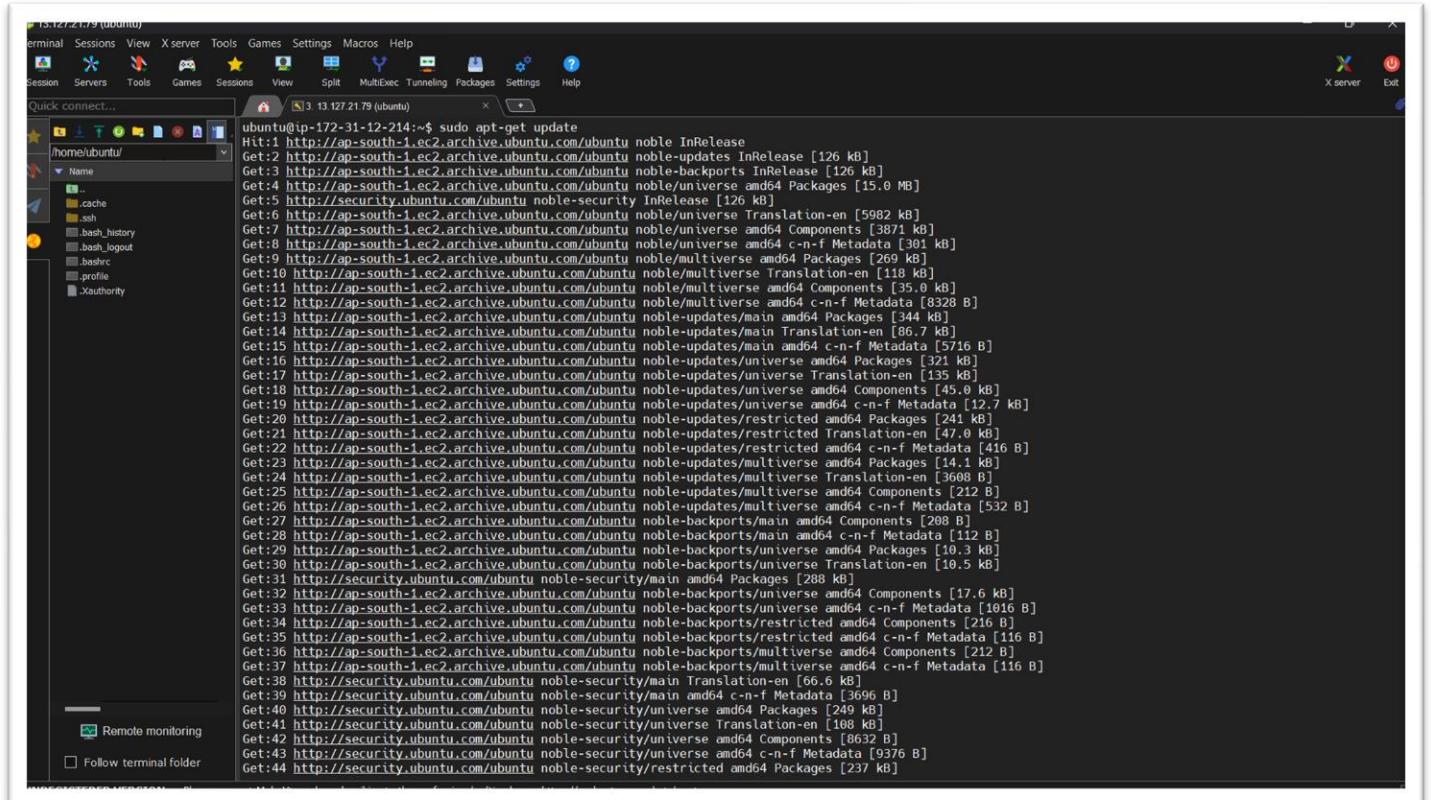
Once connected, update the package lists for upgrades and new package installations:

```sudo apt-get update```

### ii) Upgrade Installed Packages:

Upgrade the existing packages to the latest versions:

``` sudo apt-get upgrade -y```



The screenshot shows a MobaXterm session titled '13.127.21.79 (ubuntu)'. The terminal window displays the command 'sudo apt-get update' being run, followed by a long list of package download details. The list includes numerous GET requests for files from 'http://ap-south-1.ec2.archive.ubuntu.com/ubuntu' and 'http://security.ubuntu.com/ubuntu', indicating the download of InRelease, universe, multiverse, and security components for various packages like 'noble', 'noble-updates', 'noble-security', etc., in both amd64 and i386 architectures. The terminal interface includes a file manager sidebar on the left and various system status icons at the top.

```
ubuntu@ip-172-31-12-214:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [344 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [86.7 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [5716 B]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [321 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [135 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [12.7 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [241 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [47.0 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [288 B]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.3 kB]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.5 kB]
Get:31 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [288 kB]
Get:32 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:33 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1016 B]
Get:34 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:37 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:38 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [66.6 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [3696 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [249 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [108 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9376 B]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [237 kB]
```

Fig. 4.02: Updating list

```

ubuntu@ip-13-127-21-79:~$ sudo apt-get upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws
The following packages will be upgraded:
  apparmor apport apport-core-dump-handler bind9-host bind9-libs chrony curl dracut-install krb5-locales landscape-common libapparmor1
  libcurl3g4-gnutls libcurl4gtl4 libgssapi-krb5-2 libk5crypto3 libkrb5-3 libkrb5support0 libnss-systemd libopeniscsiusr libpam-systemd libpython3.12-minimal
  libpython3.12-stdlib libpython3.12t64 libssl3t64 libsystemd-shared libsystemd0 libudev1 linux-tools-common lxd-installer needrestart open-iscsi
  openssh-client openssh-server openssh-sftp-server openssl python3-apport python3-problem-report python3.12 python3.12-minimal snapd systemd systemd-dev
  systemd-resolved systemd-sysv thin-provisioning-tools udev xkb-data
48 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 55.6 MB of archives.
After this operation, 156 KB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12t64 amd64 3.12.3-1ubuntu0.1 [2339 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libssl3t64 amd64 3.0.13-0ubuntu3.2 [1940 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12 amd64 3.12.3-1ubuntu0.1 [651 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-stdlib amd64 3.12.3-1ubuntu0.1 [2069 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12-minimal amd64 3.12.3-1ubuntu0.1 [2334 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-minimal amd64 3.12.3-1ubuntu0.1 [832 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libnss-systemd amd64 255.4-1ubuntu8.2 [159 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 systemd-dev all 255.4-1ubuntu8.2 [104 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 systemd-resolved amd64 255.4-1ubuntu8.2 [296 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libsystemd-shared amd64 255.4-1ubuntu8.2 [2072 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libsystemd0 amd64 255.4-1ubuntu8.2 [433 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 systemd-sysv amd64 255.4-1ubuntu8.2 [11.9 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpam-systemd amd64 255.4-1ubuntu8.2 [235 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apport all 2.28.1-0ubuntu3.1 [3471 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 udev amd64 255.4-1ubuntu8.2 [1873 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libudev1 amd64 255.4-1ubuntu8.2 [175 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libapparmor1 amd64 4.0.1really4.0.0-beta3-0ubuntu0.1 [50.3 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-problem-report all 2.28.1-0ubuntu3.1 [24.3 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-apport all 2.28.1-0ubuntu3.1 [92.3 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apport-core-dump-handler all 2.28.1-0ubuntu3.1 [17.3 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libopeniscsiusr amd64 2.1.9-3ubuntu5.1 [49.1 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libudev amd64 2.1.9-3ubuntu5.1 [327 kB]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 open-iscsi amd64 2.1.9-3ubuntu5.1 [81.7 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libk5crypto3 amd64 1.26.1-0ubuntu2.1 [143 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libgssapi-krb5-2 amd64 1.26.1-0ubuntu2.1 [33.6 kB]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libkrb5support0 amd64 1.26.1-0ubuntu2.1 [347 kB]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libkrb5-3 amd64 1.26.1-0ubuntu2.1 [347 kB]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-sftp-server amd64 1:9.6p1-0ubuntu13.4 [37.4 kB]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-server amd64 1:9.6p1-0ubuntu13.4 [509 kB]

```

*Fig. 4.03: Upgrade the existing packages to the latest versions*

#### 4. Install JDK (Java Development Kit)

##### i) Install JDK:

Install OpenJDK (replace 22 with another version if required)

``` sudo apt-get install openjdk-22-jdk -y````

##### ii) Verify JDK Installation:

Check the installed Java version

``` java -version ````

```

ubuntu@ip-172-31-12-214:~$ sudo apt-get install openjdk-21-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
adwaita-icon-theme alsaa-topology-conf alsacm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig
fontconfig-config fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme
humanity-icon-theme java-common libasound2-data libasound2t64 libatk-bridge2.0-0t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64
libatspi2.0-0t64 libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcolord2 libcurl3t64 libdatrie1 libdconf1
libdeflate0 libdrm-amdpv1 libdrm-intel libdrm-nouveau2 libdrm-radeon1 libepoxy0 libfontconfig libgd-pixbuf-2.0-0 libgd-pixbuf2.0-0t64
libgdk-pixbuf2.0-common libgit7 libgl libgl1-amber-dri libglapi-mesa libglvnd libglx-mesa libglx0 libgraphite2-3 libgtk-3-0t64
libgtk-3-0t64 libgtk-3-common libharfbuzz0b libice-dev libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblerc4 libllyvm17t64 libpango-1.0-0
libpangocairo-1.0-0 libpangoft2-1.0-0 libpiciaccess0 libpcslite1 libpixman-1-0 libpthread-stubs0-dev librsvg2-2 librsvg2-common libsharpyuv0 libsm-dev
libsm0 libthal-data libthai0 libtiff libulkani libwayland-client0 libwayland-cursor0 libwayland-egl1 libwevp7 libx11-dev libx11-xcb1 libxau-dev libxaw7
libxcb-dr2-0 libxcb-dri3-0 libxcb-glx libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev
libxcomposite1 libxcursor1 libxdamage1 libxdmp-dev libxf86-fixes3 libxft2 libx16 libxinerama1 libxbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
libxsmfencel libxt-dev libxt6t64 libxtst6 libxv1 libxf86dg1 libxf86vml mesa-vulkan-drivers openjdk-21-jdk openjdk-21-jdk-headless openjdk-21-jre
openjdk-21-jre-headless session-migration ubuntu-mono x11-common x11-utils x1proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 132 newly installed, 0 to remove and 3 not upgraded.
Need to get 202 MB of archives.
After this operation, 632 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libgd-pixbuf2.0-common all 2.42.10+dfsg-3ubuntu3.1 [8024 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libjpeg-turbo8 amd64 2.1.5-2ubuntu2 [150 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libjpeg8 amd64 8c-2ubuntu11 [2148 B]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libdeflate0 amd64 1.19-1build1 [43.8 kB]

```

*Fig. 4.04: Installing JDK latest version*

```

ubuntu@ip-172-31-12-214:~$ java -version
openjdk version "21.0.4" 2024-07-16
OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-1ubuntu224.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-1ubuntu224.04, mixed mode, sharing)
ubuntu@ip-172-31-12-214:~$

```

*Fig. 4.05: Verifying JDK/Java installation*

## 5. Install Maven:

### i) Install Maven:

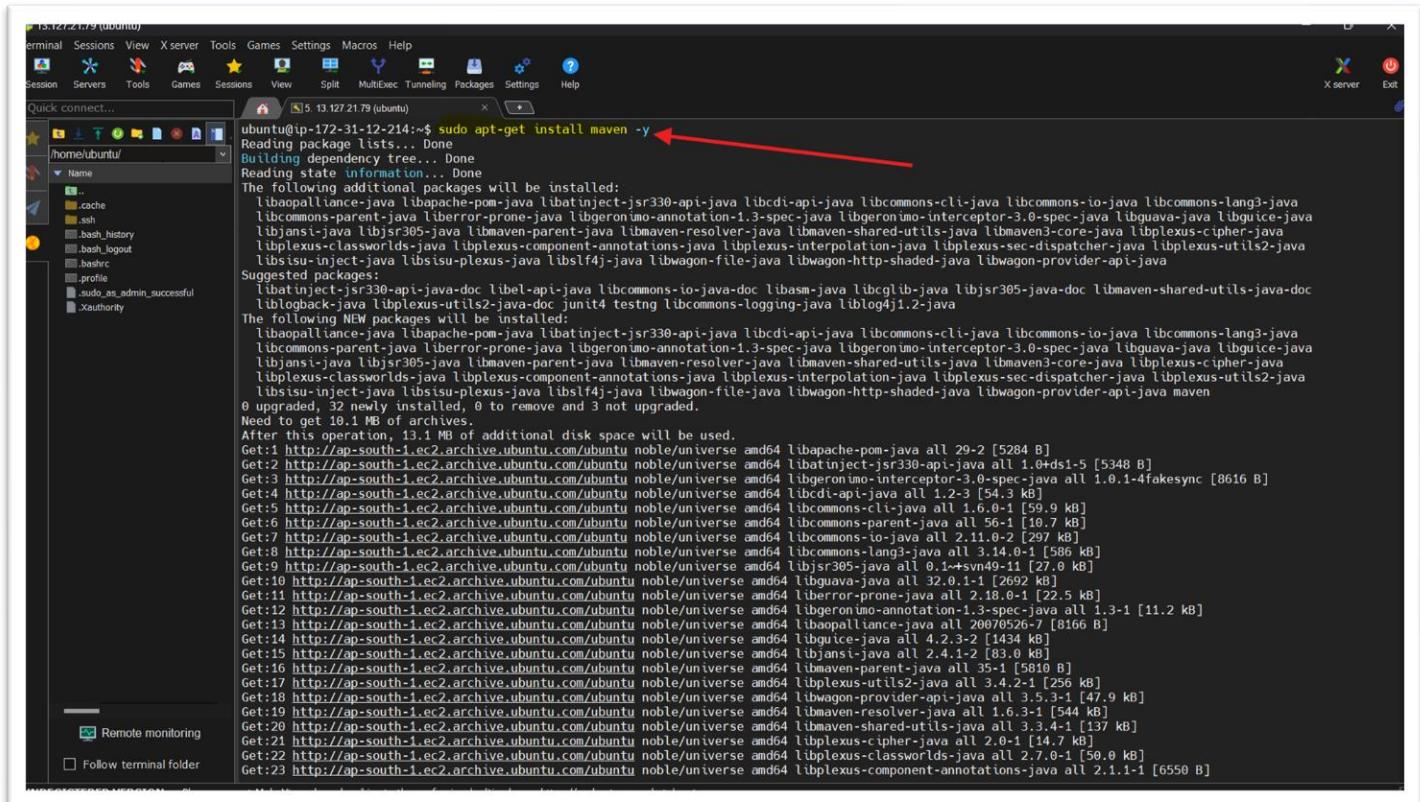
Install Maven using the package manager:

``` sudo apt-get install maven -y```

### ii) Verify Maven Installation:

Check the installed Maven version:

``` mvn -version```



The screenshot shows a terminal window titled '13.127.21.79 (Ubuntu)' with the command 'sudo apt-get install maven -y' being run. A red arrow points to the command line. The terminal output shows the package list, dependency resolution, and the download and installation process for Maven and its dependencies.

```
ubuntu@ip-172-31-12-214:~$ sudo apt-get install maven -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapolliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java libcommons-cli-java libcommons-io-java libcommons-lang3-java
  libcommons-parent-java liberror-prone-java liberonimo-annotation-1.3-spec-java liberonimo-interceptor-3.0-spec-java libguava-java libguice-java
  libjansi-java libjsr305-java libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java libmaven3-core-java libplexus-cipher-java
  libplexus-classworlds-java libplexus-component-annotations-java libplexus-interpolation-java libplexus-sec-dispatcher-java libplexus-utils2-java
  libsisu-inject-java libsisu-plexus-java libslf4j-java libwagon-file-java libwagon-http-shaded-java libwagon-provider-api-java
Suggested packages:
  libatinject-jsr330-api-java-doc libel-api-java libcommons-io-java-doc libasm-java libcglib-java libjsr305-java-doc libmaven-shared-utils-java-doc
  liblogback-java libplexus-utils2-java-doc junit4 testng libcommons-logging-java liblog4j1.2-java
The following NEW packages will be installed:
  libapolliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java libcommons-cli-java libcommons-io-java libcommons-lang3-java
  libcommons-parent-java liberror-prone-java liberonimo-annotation-1.3-spec-java liberonimo-interceptor-3.0-spec-java libguava-java libguice-java
  libjansi-java libjsr305-java libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java libmaven3-core-java libplexus-cipher-java
  libplexus-classworlds-java libplexus-component-annotations-java libplexus-interpolation-java libplexus-sec-dispatcher-java libplexus-utils2-java
  libsisu-inject-java libsisu-plexus-java libslf4j-java libwagon-file-java libwagon-http-shaded-java libwagon-provider-api-java maven
0 upgraded, 32 newly installed, 0 to remove and 3 not upgraded.
Need to get 10.1 MB of archives.
After this operation, 13.1 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libapache-pom-java all 29-2 [5284 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libatinject-jsr330-api-java all 1.0+ds1-5 [5348 B]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 liberonimo-interceptor-3.0-spec-java all 1.0.1-4faesync [8616 B]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libcdi-api-java all 1.2-3 [54.3 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libcommons-cli-java all 1.6.0-1 [59.9 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libcommons-parent-java all 56-1 [10.7 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libcommons-io-java all 2.11.0-2 [297 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libcommons-lang3-java all 3.14.0-1 [586 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libjsr305-java all 0.1+svn49-11 [27.0 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libguava-java all 32.0.1-1 [2692 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 liberror-prone-java all 2.18.0-1 [22.5 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 liberonimo-annotation-1.3-spec-java all 1.3-1 [11.2 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libapolliance-java all 20070526-7 [8166 B]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libguice-java all 4.2.3-2 [1434 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libjansi-java all 2.4.1-2 [83.0 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libmaven-parent-java all 35-1 [5810 B]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libplexus-util2-java all 3.4.2-1 [256 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libwagon-provider-api-java all 3.5.3-1 [47.9 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libmaven-resolver-java all 1.6.3-1 [544 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libmaven-shared-utils-java all 3.3.4-1 [137 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libplexus-cipher-java all 2.0-1 [14.7 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libplexus-classworlds-java all 2.7.0-1 [50.0 kB]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libplexus-component-annotations-java all 2.1.1-1 [6550 B]
```

Fig. 4.06: Maven installation guide

A screenshot of a terminal window titled '13.127.21.79 (Ubuntu)'. The window shows a file browser on the left with a path of '/home/ubuntu/'. The main pane displays the output of the command 'mvn -version'. A red arrow points to the output line 'Apache Maven 3.8.7'. The terminal interface includes a menu bar with 'terminal', 'Sessions', 'View', 'X server', 'Tools', 'Games', 'Settings', 'Macros', 'Help', and various icons for session management and X server control. At the bottom, there are checkboxes for 'Remote monitoring' and 'Follow terminal folder'.

```
ubuntu@ip-172-31-12-214:~$ mvn -version
Apache Maven 3.8.7
Java version: 21.0.4, vendor: Ubuntu, runtime: /usr/lib/jvm/java-21-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1009-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-12-214:~$
```

Fig. 4.07: Verifying maven installation

## 6. Install Git:

### i) Install Git:

Install Git using the package manager:

``` sudo apt-get install git -y```

### ii) Verify Git Installation:

Check the installed Git version:

``` git --version```

```

13.127.21.79 (Ubuntu)
terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 5 13.127.21.79 (ubuntu) X server Exit
/home/ubuntu/
Name
.. .cache .ssh .bash_history .bash_logout .bashrc .profile .sudo_as_admin_successful .Xauthority
ubuntu@ip-172-31-12-214:~$ sudo apt-get install git -y
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 3 not upgraded.
ubuntu@ip-172-31-12-214:~$ git --version
git version 2.43.0
ubuntu@ip-172-31-12-214:~$ 

```

Remote monitoring  Follow terminal folder

*Fig. 4.08: Installing Git and verifying its installation*

## 5. L5 - Install Tomcat web application server in AWS EC2 Ubuntu Instance and access Tomcat using a web browser

Ans:

1. Connect to the AWS EC2 Ubuntu Instance

2. Install Tomcat:

- i) Update the Package List:
- ii) Install Java (if not already installed):
  - a. Tomcat requires Java, so ensure it is installed:
- iii) Create a Tomcat User:
  - a. For security, it's best to run Tomcat under its own user:  
``` sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat ```

- iv) Download and Install Tomcat:

- a. Download the latest version of Tomcat (replace 9.0.XX with the latest version):

``` wget <https://downloads.apache.org/tomcat/tomcat-9/v9.0.XX/bin/apache-tomcat-9.0.XX.tar.gz> ```

- v) Extract the Tomcat package:

```
``` sudo tar xf apache-tomcat-9.0.XX.tar.gz -C /opt/tomcat```
```

- vi) Create a symbolic link to the Tomcat installation:

```
``` sudo ln -s /opt/tomcat/apache-tomcat-9.0.XX /opt/tomcat/latest```
```

- vii) Change ownership of the Tomcat files:

```
``` sudo chown -R tomcat: /opt/tomcat```
```

### 3. Configure Tomcat

#### a. Set Up Permissions:

- i. Give execute permissions to the Tomcat scripts:

```
``` sudo sh -c 'chmod +x /opt/tomcat/latest/bin/*.sh```
```

#### b. Create a Systemd Service File:

- i. Create a systemd service file to manage Tomcat as a service:

```
``` sudo nano /etc/systemd/system/tomcat.service```
```

- ii. Add the following configuration to the file:

```
```
```

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

User=tomcat

Group=tomcat

Environment="JAVA\_HOME=/usr/lib/jvm/java-21-openjdk-amd64"

Environment="CATALINA\_PID=/opt/tomcat/latest/temp/tomcat.pid"

Environment="CATALINA\_HOME=/opt/tomcat/latest"

Environment="CATALINA\_BASE=/opt/tomcat/latest"

Environment="CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

Environment="JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/.urandom"

ExecStart=/opt/tomcat/latest/bin/startup.sh

ExecStop=/opt/tomcat/latest/bin/shutdown.sh

[Install]

```
WantedBy=multi-user.target
```

```
```
```

Save and exit the file (Ctrl + O, Enter, Ctrl + X).

c. Reload Systemd and Start Tomcat:

- i. Reload systemd to recognize the new service:

```
``` sudo systemctl daemon-reload ```
```

- ii. Start the Tomcat service:

```
``` sudo systemctl start tomcat ```
```

- iii. Enable Tomcat to start on boot:

```
``` sudo systemctl enable tomcat ```
```

4. Verify Security Group to Allow Traffic on Port 8080

5. Access Tomcat via a Web Browser:

a. Open Your Web Browser:

In your web browser, navigate to:

```
``` http://your-instance-public-dns:8080 ```
```

Replace your-instance-public-dns with the actual public DNS of your EC2 instance.

b. Verify Tomcat Installation:

If everything is configured correctly, you should see the Tomcat welcome page.

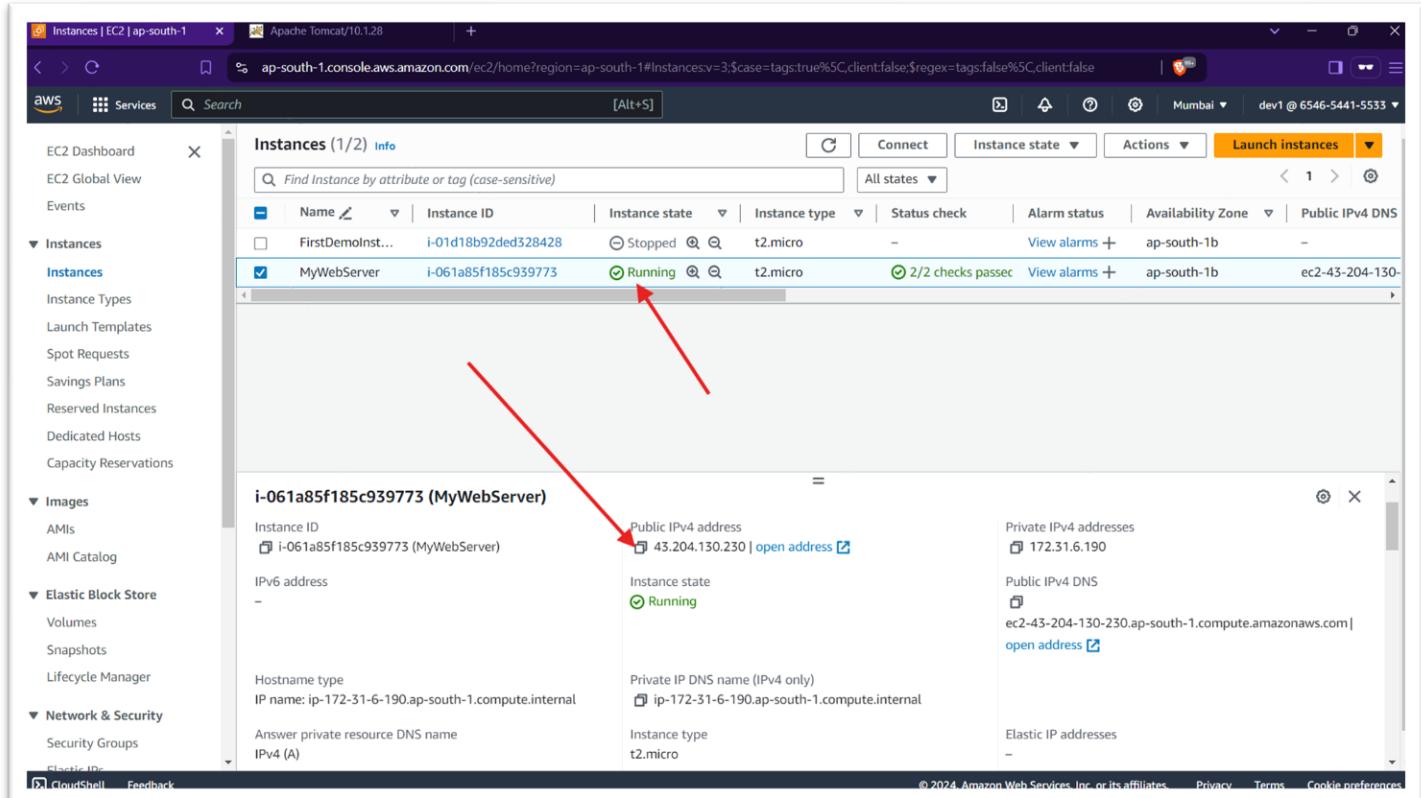


Fig. 5.01: Copy running instance IPv4 address

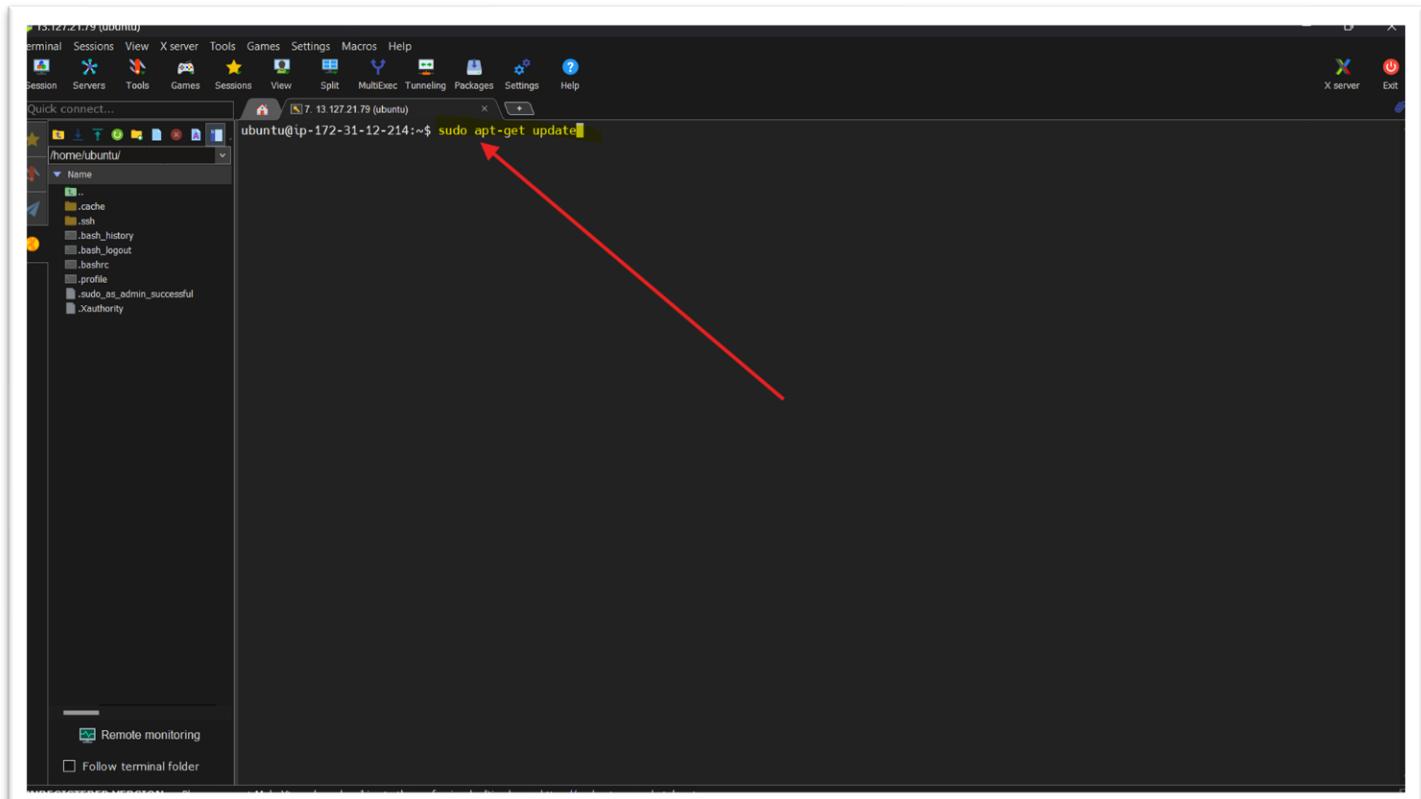
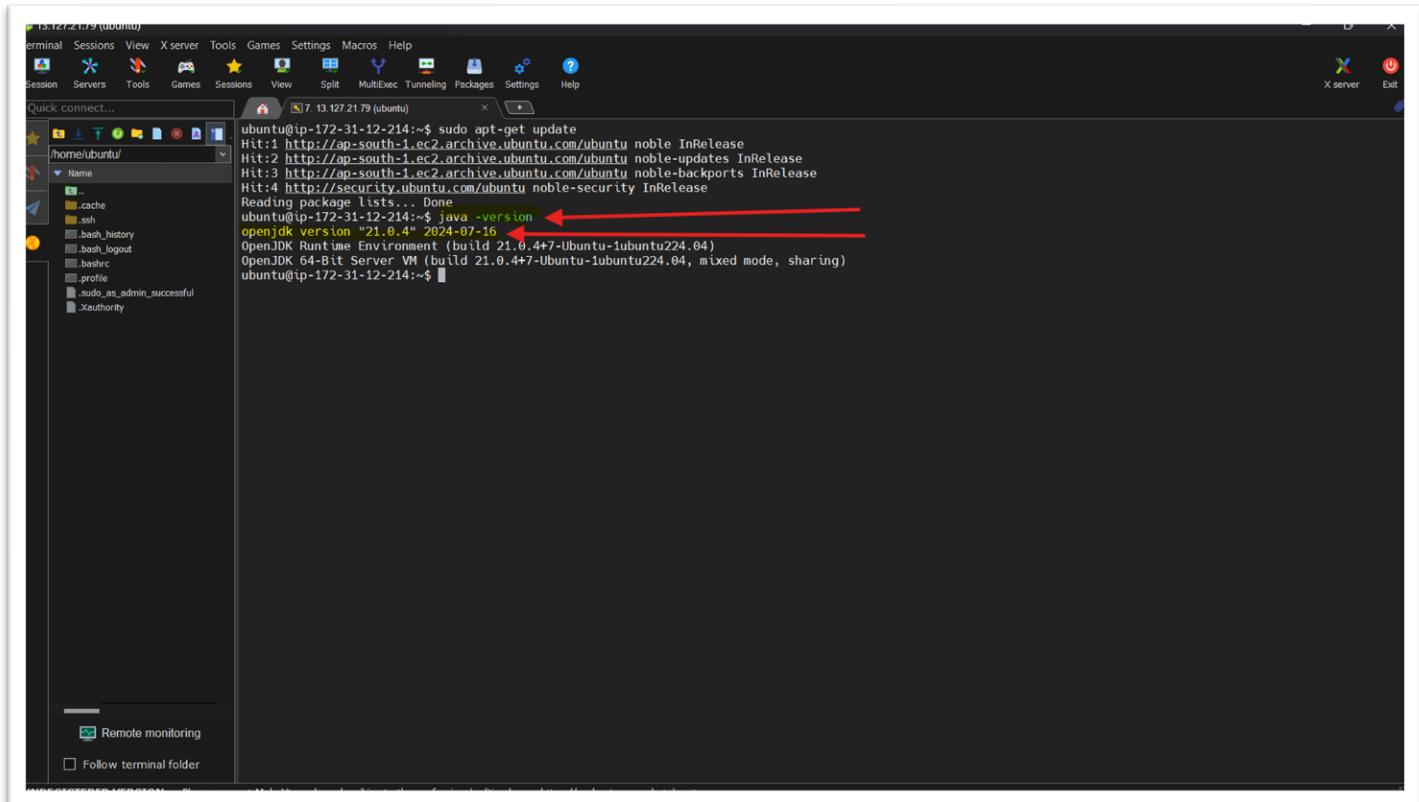


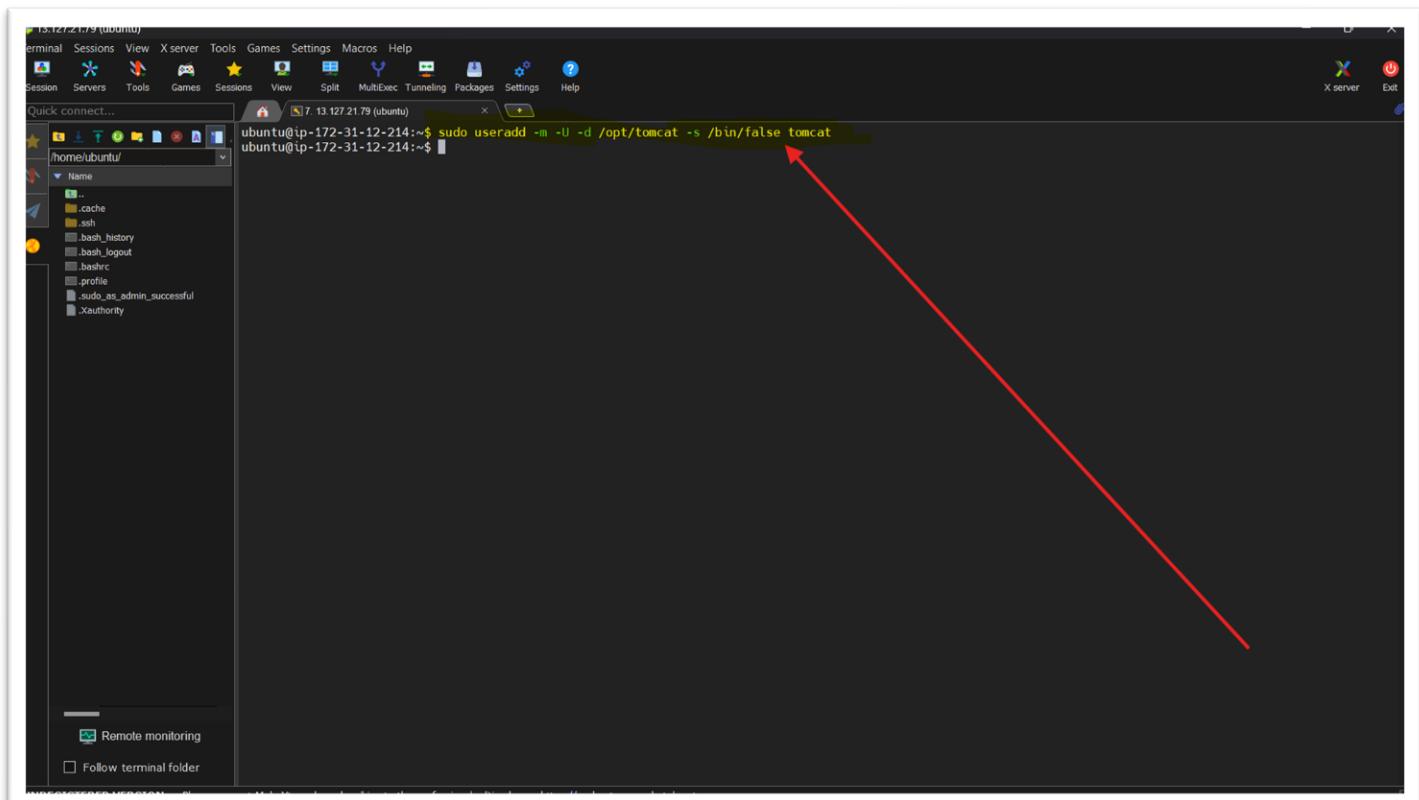
Fig. 5.02: After SSH connection established update apt-get



A screenshot of a terminal window titled "13.127.21.79 (Ubuntu)". The window shows a file tree on the left and a terminal session on the right. In the terminal, the command "java -version" is run, and the output shows OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-1ubuntu224.04) and OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-1ubuntu224.04, mixed mode, sharing). A red arrow points to the "openjdk" line in the output.

```
ubuntu@ip-172-31-12-214:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
ubuntu@ip-172-31-12-214:~$ java -version
openjdk version "21.0.4" 2024-07-16
OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-1ubuntu224.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-1ubuntu224.04, mixed mode, sharing)
ubuntu@ip-172-31-12-214:~$
```

Fig. 5.03: Verifying JDK/Java installation



A screenshot of a terminal window titled "13.127.21.79 (Ubuntu)". The window shows a file tree on the left and a terminal session on the right. In the terminal, the command "sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat" is run. A red arrow points to the command line.

```
ubuntu@ip-172-31-12-214:~$ sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat
ubuntu@ip-172-31-12-214:~$
```

Fig. 5.04: Create new user

```

45.204.150.230 (ubuntu)
terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 9 43.204.130.230 (ubuntu)
/home/ubuntu/
Name
.. .cache .ssh .bash_history .bash_logout .profile .Xauthority
ubuntu@ip-172-31-6-190:~$ sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat
ubuntu@ip-172-31-6-190:~$ cd /tmp
ubuntu@ip-172-31-6-190:~/tmp$ wget https://downloads.apache.org/tomcat/tomcat-10/v10.1.28/bin/apache-tomcat-10.1.28.tar.gz
--2024-08-11 20:53:37-- https://downloads.apache.org/tomcat/tomcat-10/v10.1.28/bin/apache-tomcat-10.1.28.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181.214.104, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13056103 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-10.1.28.tar.gz'

apache-tomcat-10.1.28.tar.gz      100%[=====] 12.45M 6.48MB/s   in 1.9s
2024-08-11 20:53:40 (6.48 MB/s) - 'apache-tomcat-10.1.28.tar.gz' saved [13056103/13056103]

ubuntu@ip-172-31-6-190:~/tmp$ sudo tar xf apache-tomcat-10.1.28.tar.gz -C /opt/tomcat
ubuntu@ip-172-31-6-190:~/tmp$ sudo ln -s /opt/tomcat/apache-tomcat-10.1.28 /opt/tomcat/latest
ubuntu@ip-172-31-6-190:~/tmp$ sudo chown -R tomcat: /opt/tomcat
ubuntu@ip-172-31-6-190:~/tmp$ sudo sh -c 'chmod +x /opt/tomcat/latest/bin/*.sh'
ubuntu@ip-172-31-6-190:~/tmp$ sudo nano /etc/systemd/system/tomcat.service
ubuntu@ip-172-31-6-190:~/tmp$ sudo systemctl daemon-reload
ubuntu@ip-172-31-6-190:~/tmp$ sudo systemctl start tomcat
ubuntu@ip-172-31-6-190:~/tmp$ sudo systemctl enable tomcat
Created symlink /etc/systemd/system/multi-user.target.wants/tomcat.service → /etc/systemd/system/tomcat.service.
ubuntu@ip-172-31-6-190:~/tmp$ 

```

Remote monitoring

Follow terminal folder

*Fig. 5.05: Downloading, Extracting, Creating symlink, Changing owner permission to tomcat, making tomcat service and enabling tomcat on startup*

```

13.127.21.79 (ubuntu)
terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 7 13.127.21.79 (ubuntu)
/home/ubuntu/
Name
.. .cache .ssh .bash_history .bash_logout .profile .sudo_as_admin_successful .Xauthority
GNU nano 7.2          /etc/systemd/system/tomcat.service
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking
User=tomcat
Group=tomcat
Environments="JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64"
Environments="CATALINA_PID=/opt/tomcat/latest/temp/tomcat.pid"
Environments="CATALINA_HOME=/opt/tomcat/latest"
Environments="CATALINA_BASE=/opt/tomcat/latest"
Environments="CATALINA_OPTS=-Xms512M -XX:+UseParallelGC"
Environments="JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom"

ExecStart=/opt/tomcat/latest/bin/startup.sh
ExecStop=/opt/tomcat/latest/bin/shutdown.sh

[Install]
WantedBy=multi-user.target

```

Remote monitoring

Follow terminal folder

Help Exit Write Out Read File Replace Cut Paste Read 20 lines Execute Location Go To Line Undo Redo Set Mark Copy To Bracket Where Was

25°C 01:57 AM

*Fig. 5.06: Contents of tomcat.service*

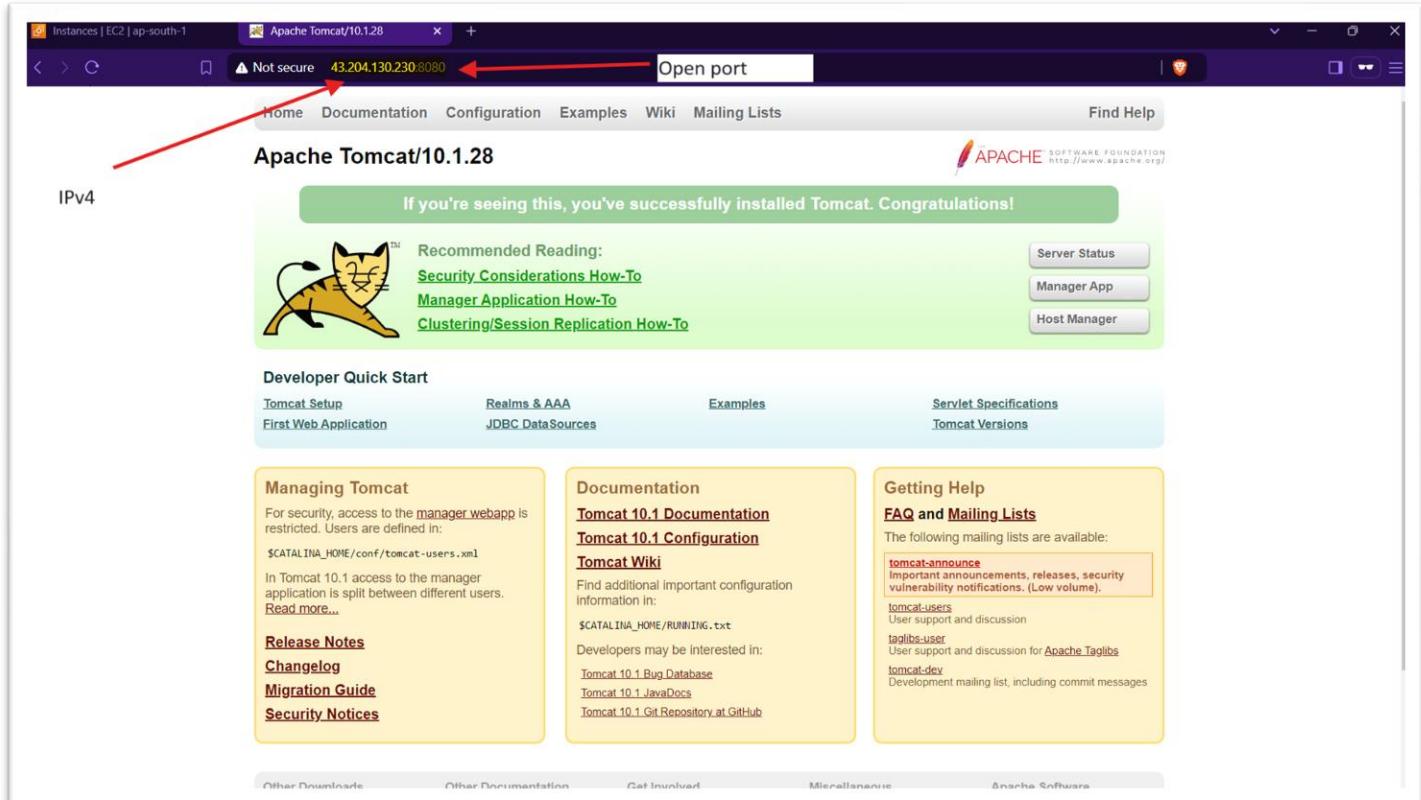


Fig. 5.07: Accessing Tomcat server through web browser

## 6. L6 - Create S3 Bucket and add folders and files

Ans:

1. Verify if the current IAM user has access to create and edit S3 bucket

The screenshot shows the AWS IAM User Groups page. On the left, the navigation menu includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups' selected), 'Access reports', and 'Service control policies'. The main content area shows the 'Developer' user group under 'User groups > Developer'. The 'Summary' section displays the group name 'Developer', creation time 'August 11, 2024, 19:29 (UTC+05:30)', and ARN 'arn:aws:iam::654654415533:group/Developer'. Below this, there are tabs for 'Users (1)', 'Permissions' (selected), and 'Access Advisor'. The 'Permissions policies' section lists two policies: 'AmazonEC2FullAccess' (AWS managed, attached to 1 entity) and 'AmazonS3FullAccess' (AWS managed, attached to 2 entities). A red arrow points to the 'AmazonS3FullAccess' policy.

*Fig. 6.01: Verify S3 access rights to current IAM user*

## 2. Create an S3 Bucket:

- Log in to the AWS Management Console:
- Navigate to S3:

In the AWS Management Console, type "S3" in the search bar and select S3 from the list of services.

- Create a New Bucket:

- Click on the Create bucket button.
- Enter a Bucket name. The name must be unique across all of AWS. For example, my-unique-bucket-name.
- Choose the AWS Region where you want the bucket to be created.
- Configure the Bucket settings for Block Public Access (leave it as default unless you want the bucket to be public).
- Leave the other settings as default, or adjust them based on your needs.
- Click Create bucket.

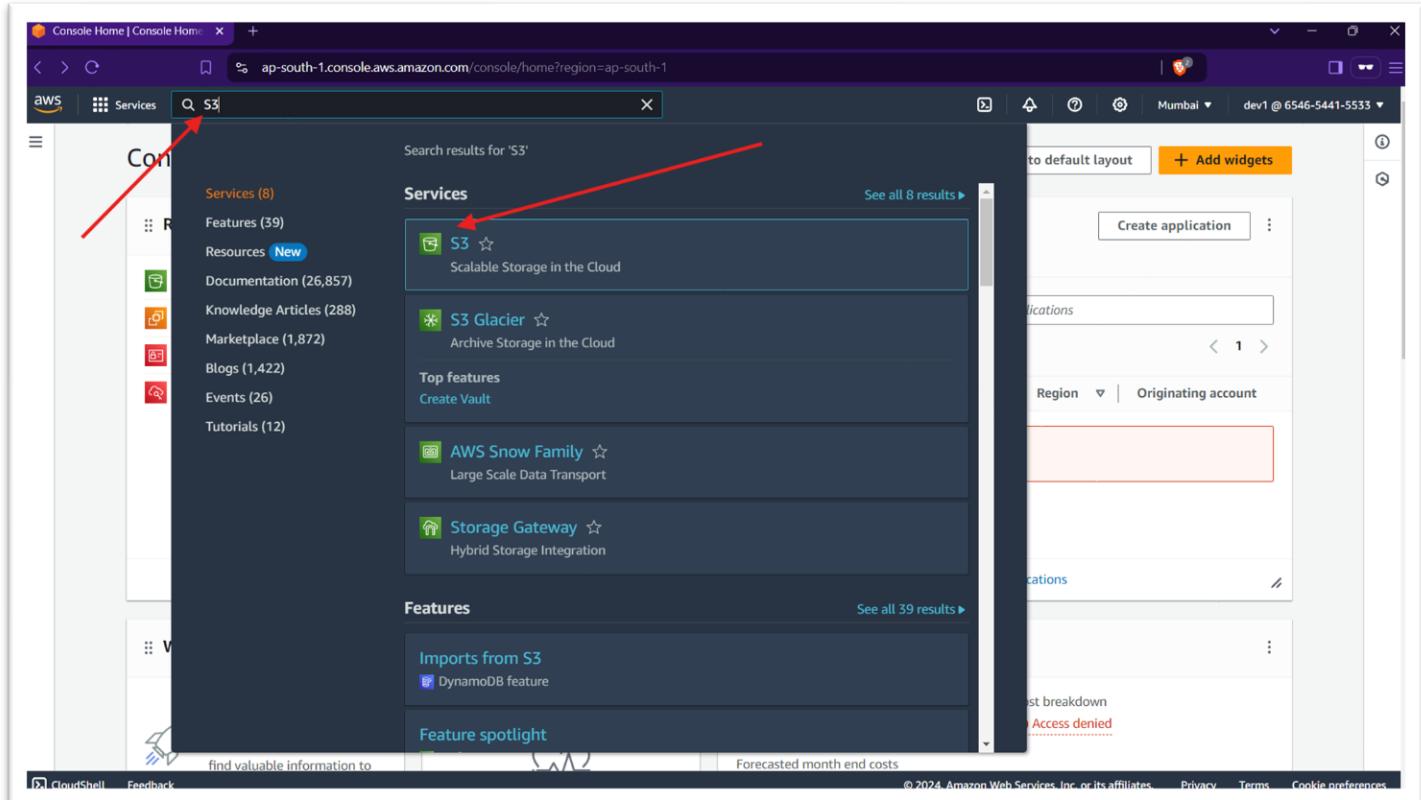


Fig. 6.02: Search S3 in AWS dashboard

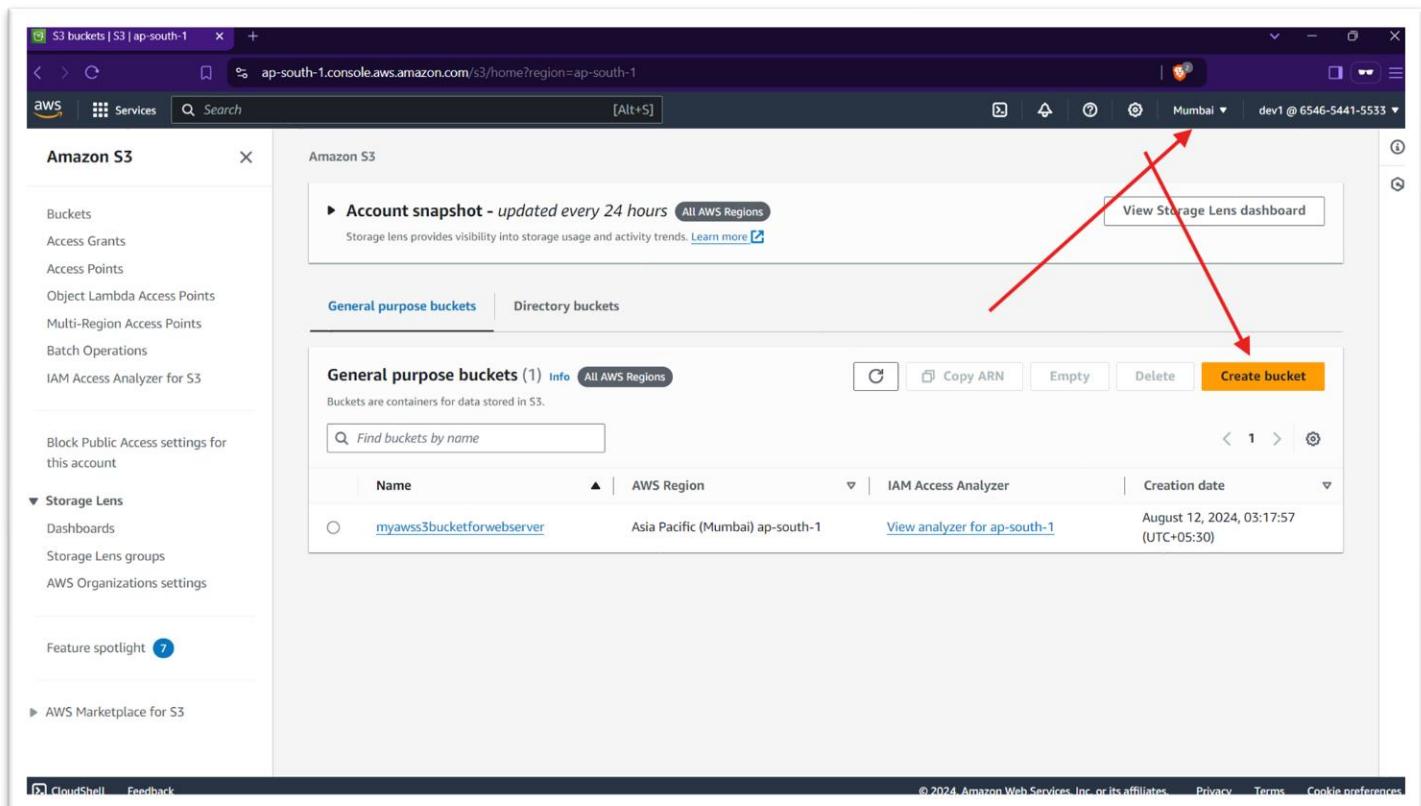


Fig. 6.03: Create new S3 bucket

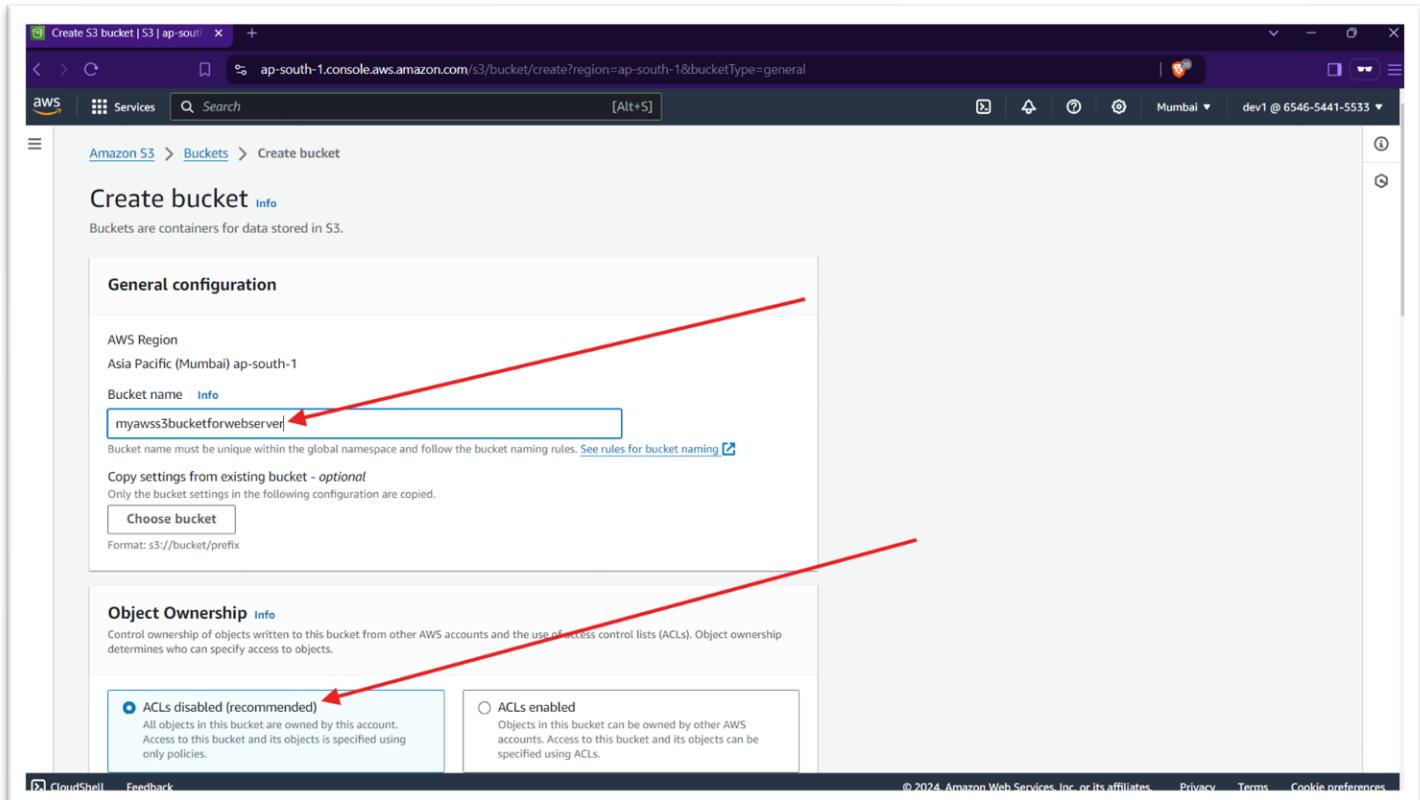


Fig. 6.04: Give appropriate bucket name

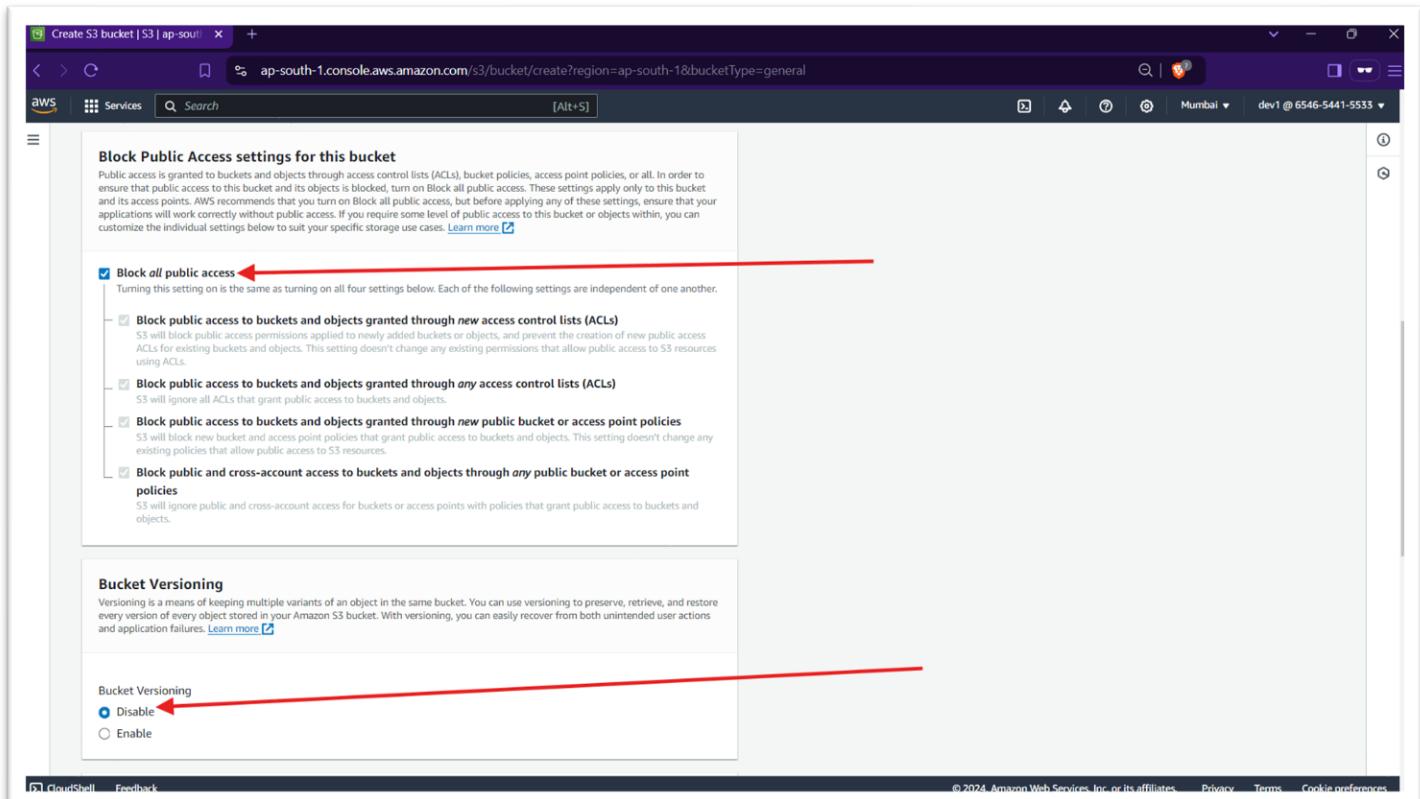


Fig. 6.05: Block public access

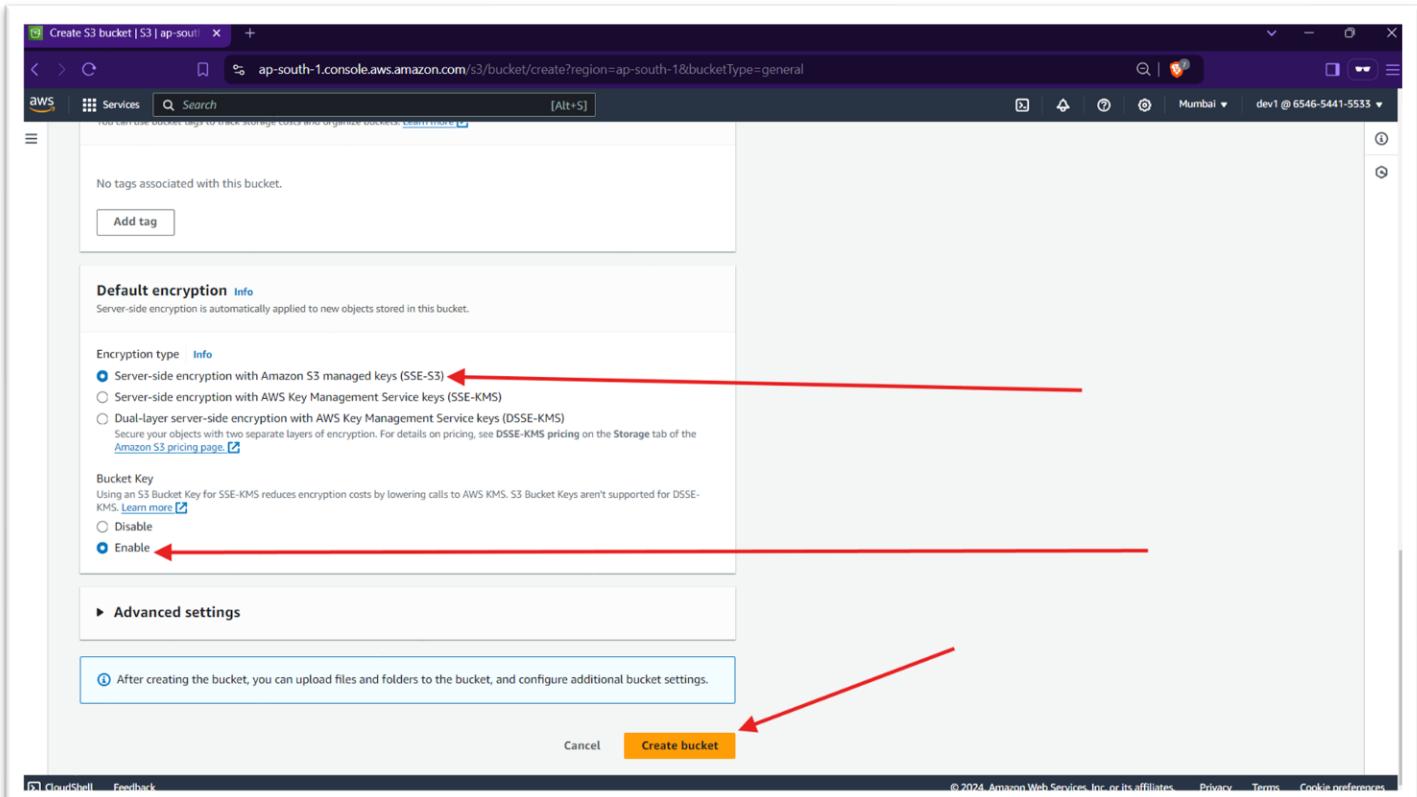


Fig. 6.06: Setting up encryption for bucket

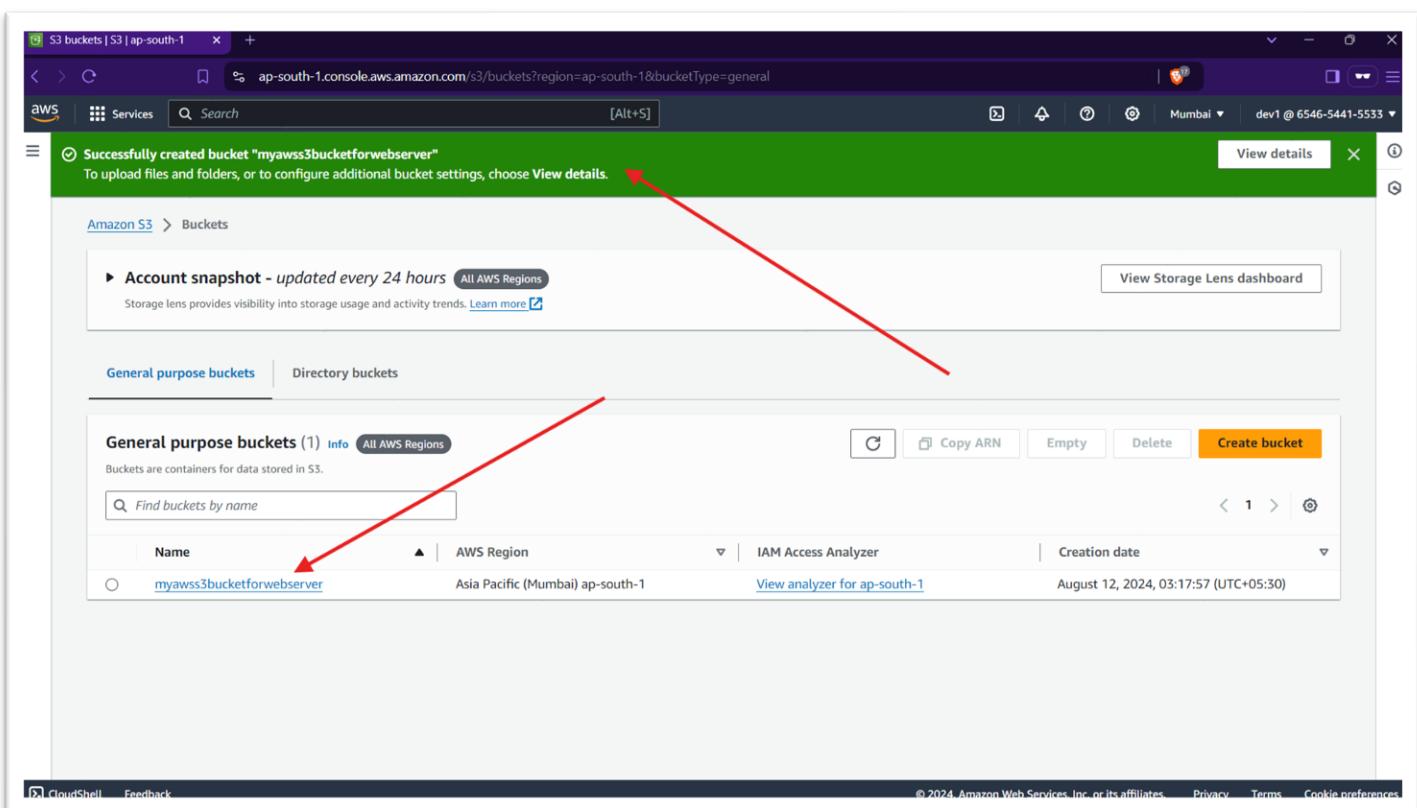


Fig. 6.07: After successful bucket creation

### 3. Add Folders and Files to the S3 Bucket:

#### a. Open the Bucket:

i. Once the bucket is created, click on the bucket name to open it.

#### b. Create a Folder:

i. In the Objects tab, click on the Create folder button.

ii. Enter a name for the folder, e.g., my-folder.

iii. Click Create folder.

#### c. Upload Files:

i. Inside the folder, click on the Upload button.

ii. In the upload screen, either drag and drop files from your local machine or click on Add files to browse your local files and select them.

iii. Optionally, you can also click on Add folder to upload an entire folder from your local machine.

iv. After adding the files, click Upload to start uploading them to the S3 bucket.

### 4. Verify the Upload:

a. After the upload is complete, you can see the files and folders listed under the bucket in the S3 console.

b. To download or manage these files, click on their names.

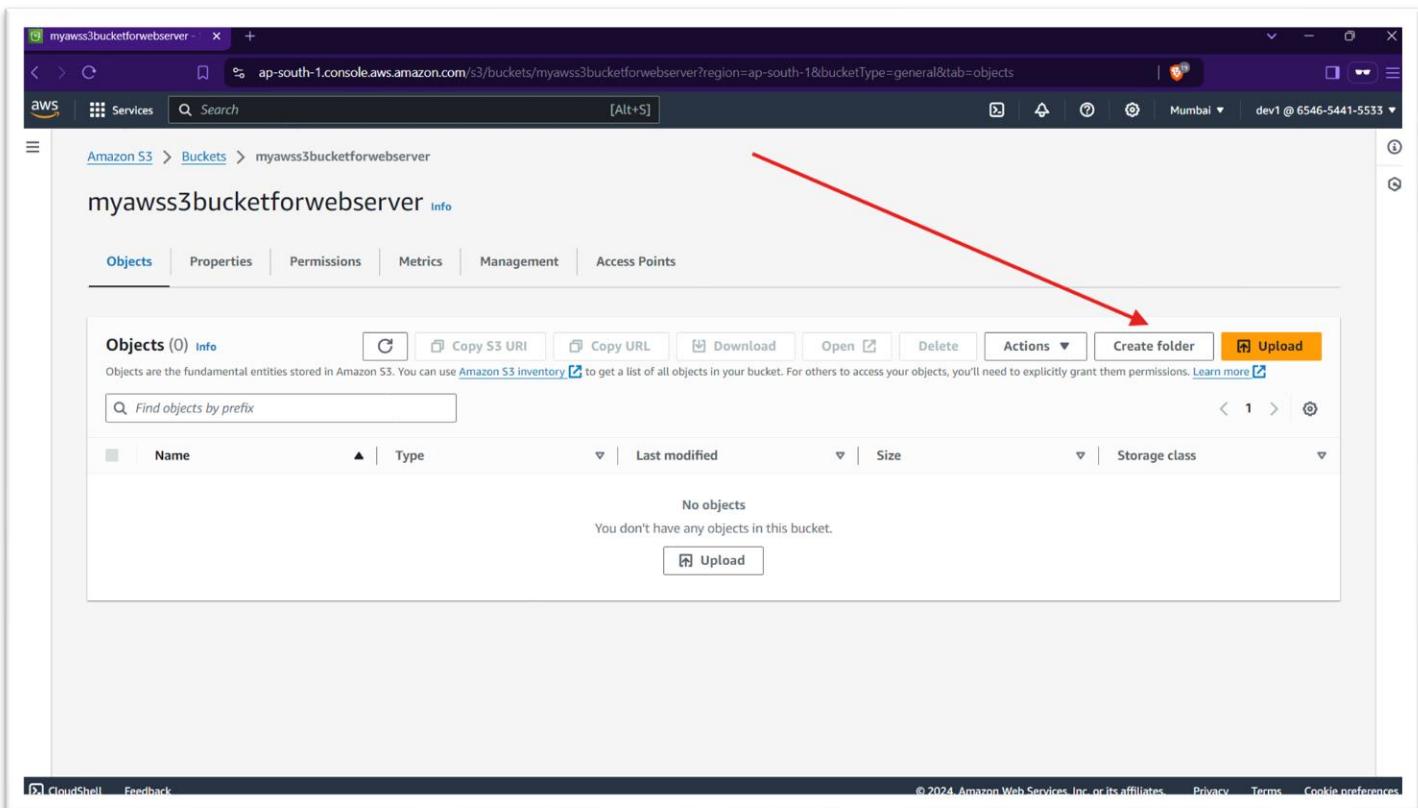


Fig. 6.08: Create folder inside S3 bucket

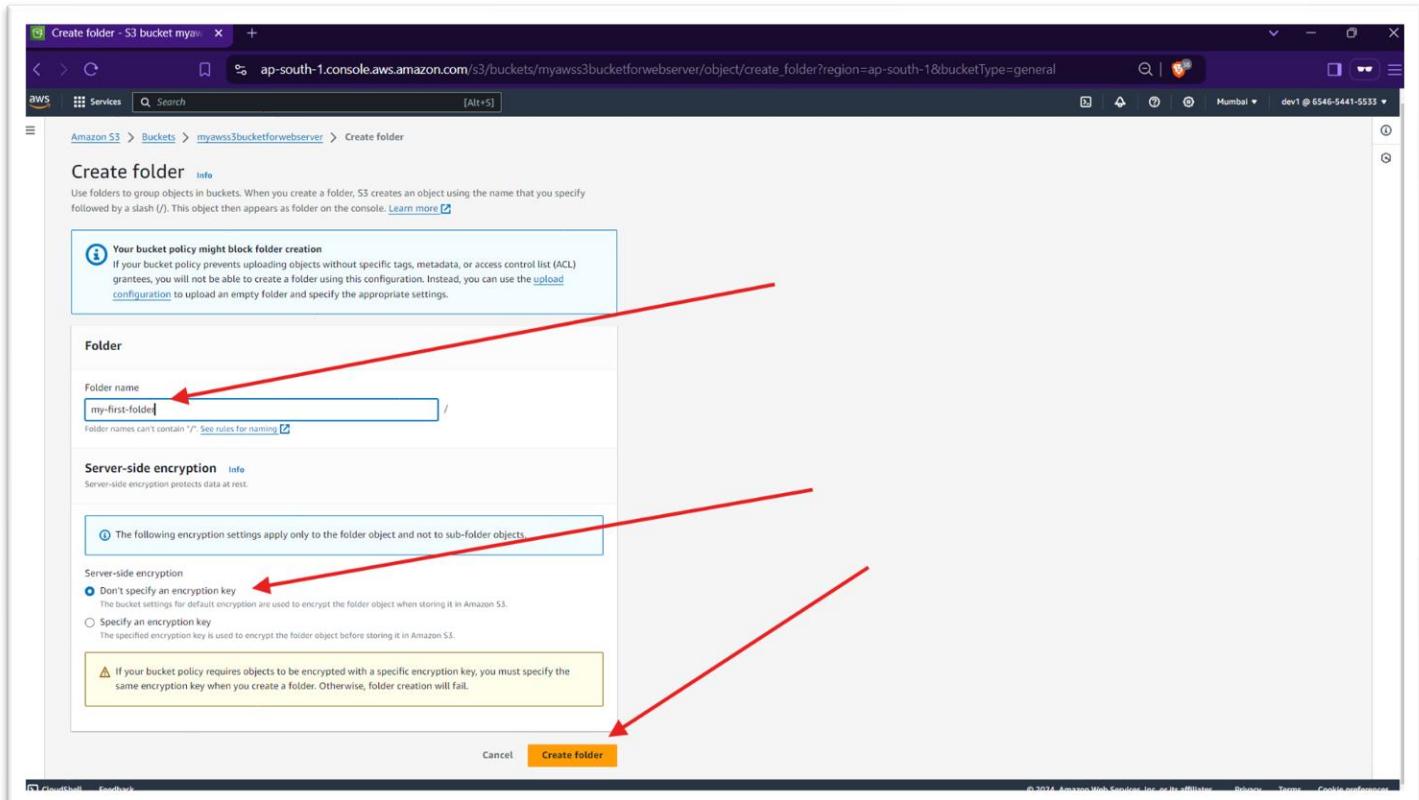


Fig. 6.09: Folder naming

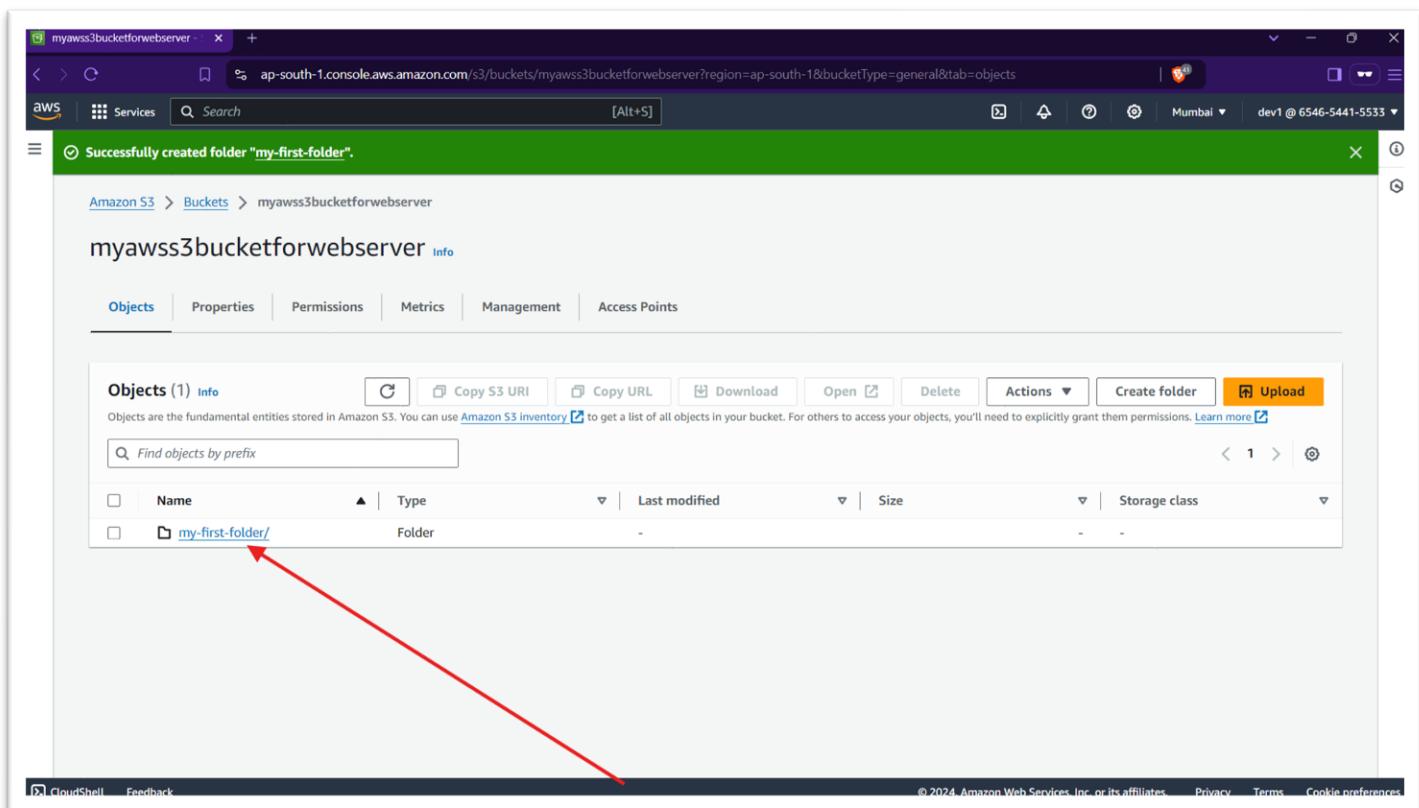


Fig. 6.10: Opening already created folder

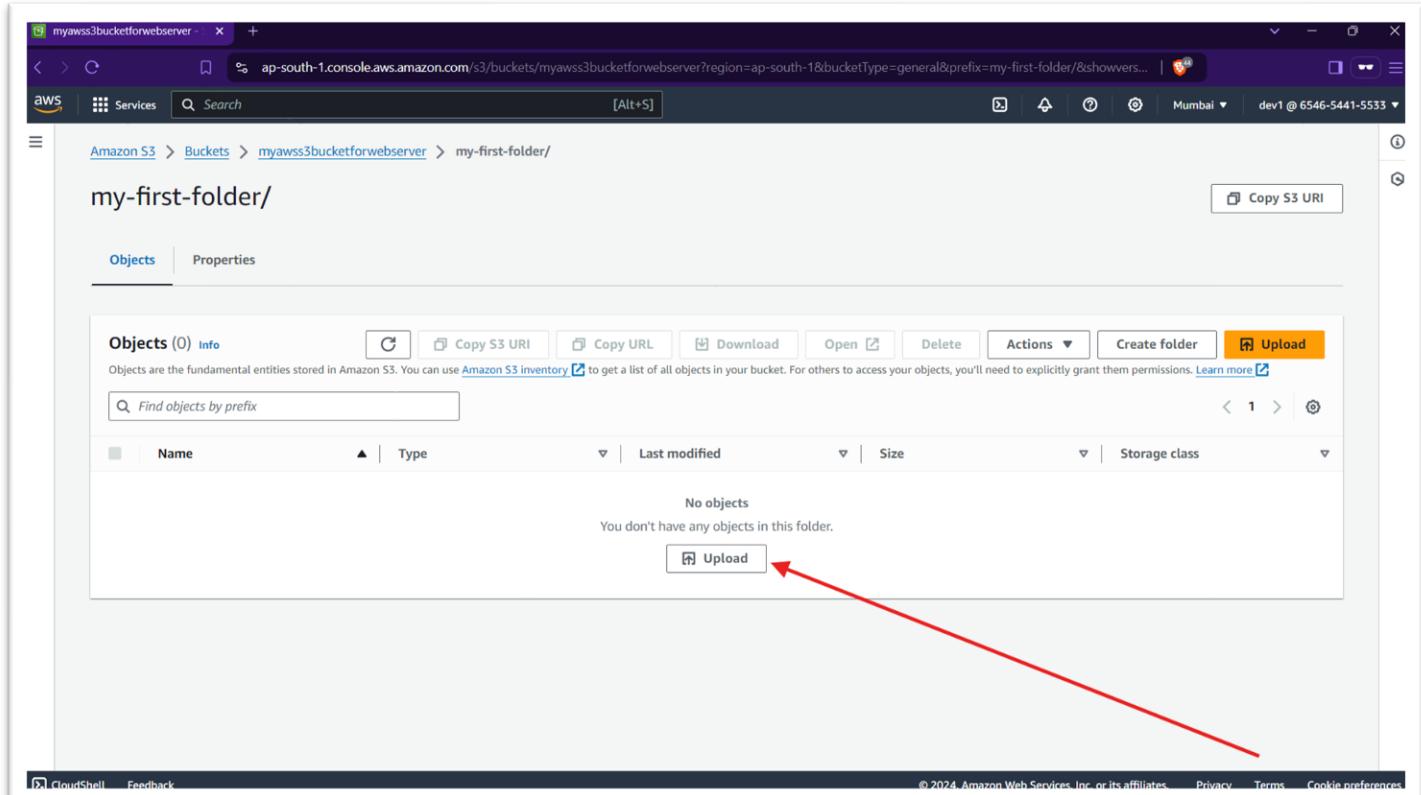


Fig. 6.11: Uploading files in the folder

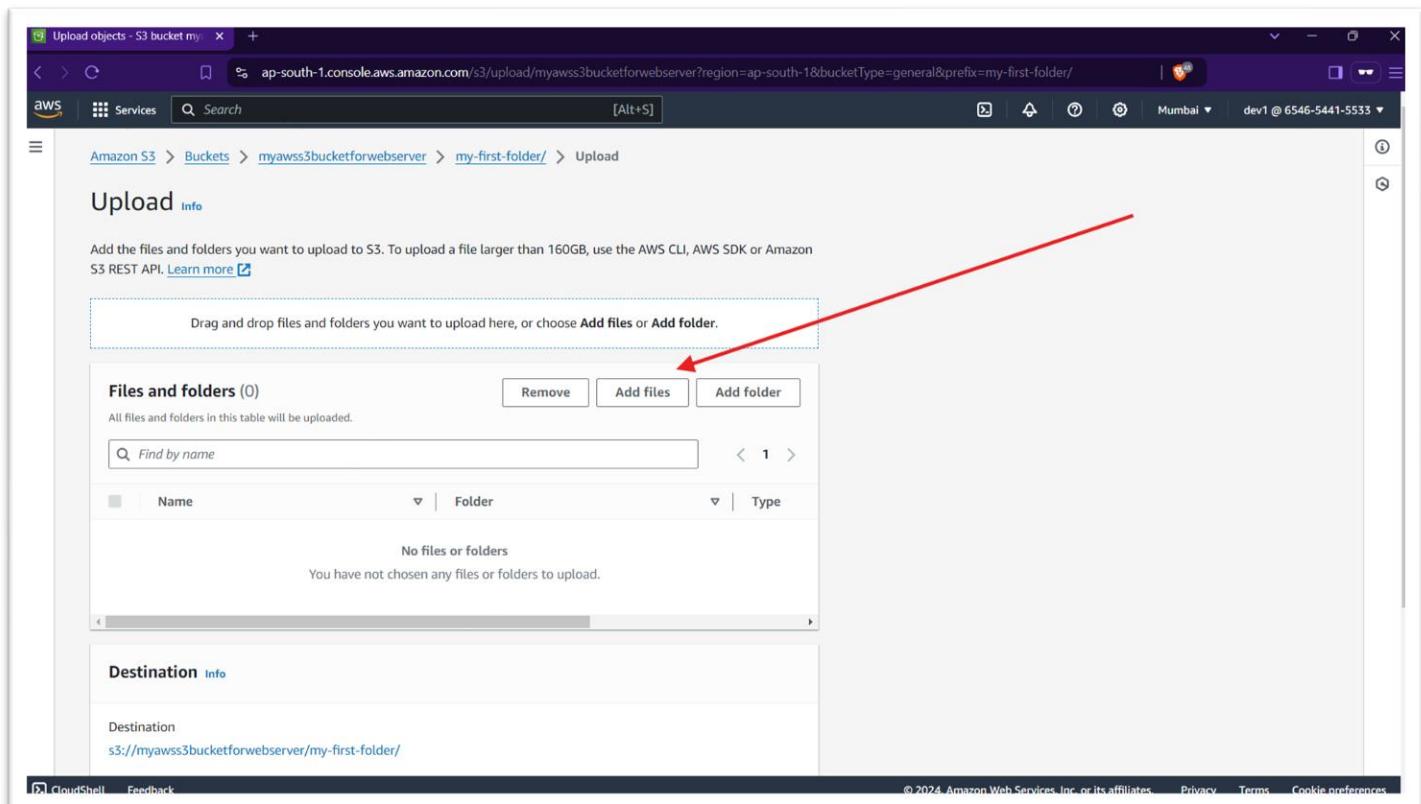


Fig. 6.12: Drag and Drop or Select files from local repository

The screenshot shows the AWS S3 'Upload objects' interface. At the top, a green banner displays the message 'Upload succeeded' with a link 'View details below.' A red arrow points from this message towards the 'Files and folders' section. Below the banner, the title 'Upload: status' is shown. A note indicates that the information will no longer be available after navigating away. The 'Summary' section shows the destination 's3://myawss3bucketforwebserver/my-first-folder/' with a 'Succeeded' status for 1 file (140.2 KB) and 0 failed files. The 'Files and folders' tab is selected, displaying a table with one row: 'B4 upload fi...' (image/png, 140.2 KB, Succeeded). The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and copyright information.

Fig. 6.13: After successful file upload

The screenshot shows the AWS S3 'my-first-folder' dashboard. The URL in the address bar is 'ap-south-1.console.aws.amazon.com/s3/buckets/myawss3bucketforwebserver?region=ap-south-1&bucketType=general&prefix=my-first-folder/'. The main area displays the 'Objects (1)' section with a single item: 'B4 upload files.png' (png, 140.2 KB, Standard storage class). A red arrow points from this file entry towards the left edge of the table. The top navigation bar shows the bucket name 'myawss3bucketforwebserver'. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and copyright information.

Fig. 6.14: Final dashboard view after creating folder and uploading file