

School of Computer Science, Engineering and Applications (SCSEA)
B.C.A. TY (CCSA)
Subject: Containers & Orchestration

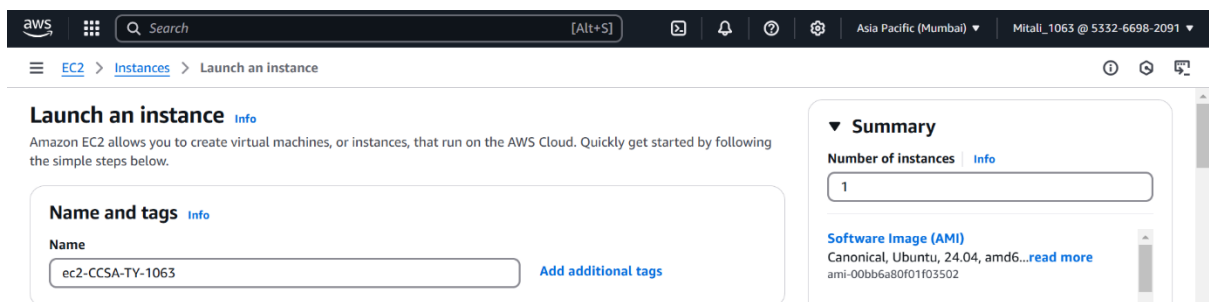
Name of the Student: Mitali Bhattad

PRN: 20220801063

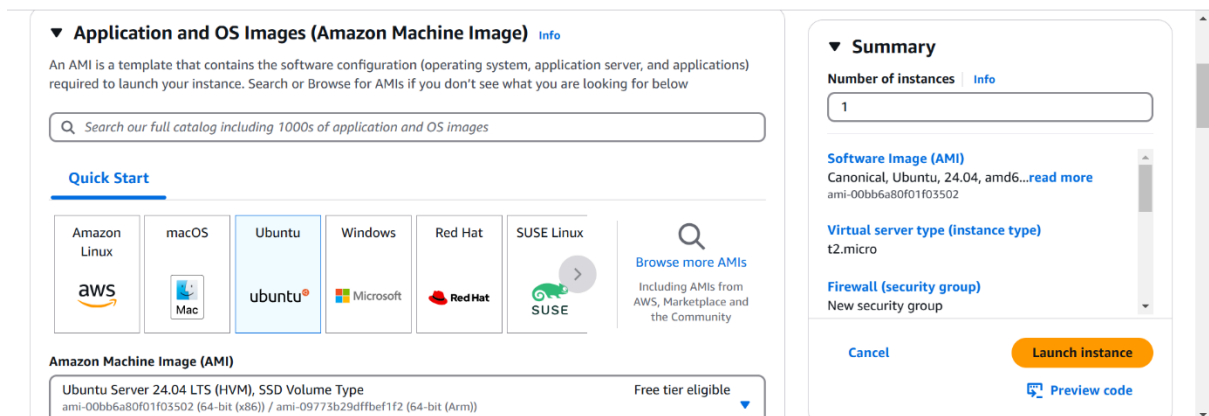
Title of Practical: Deploying and securing docker community edition on AWS linux EC2 instance.

Step1: Launch an EC2 Instance

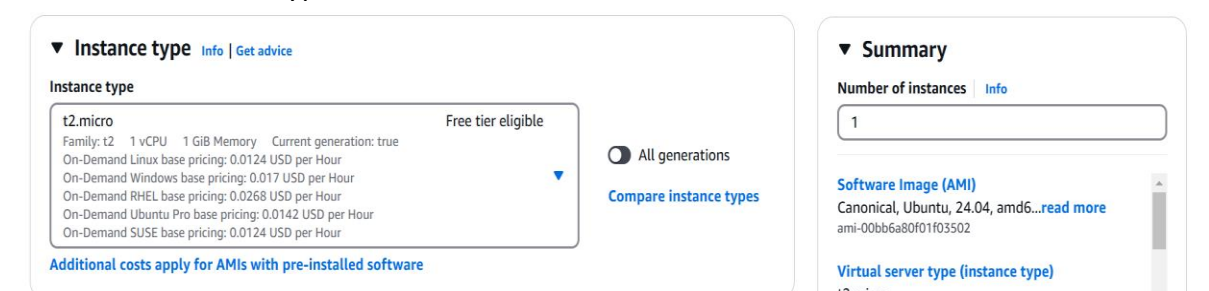
- Name the Instance



- Choose AMI: Ubuntu



- Select the instance type: t2 micro



School of Computer Science, Engineering and Applications (SCSEA)

B.C.A. TY (CCSA)

Subject: Containers & Orchestration

Name of the Student: Mitali Bhattad

PRN: 20220801063

Title of Practical: Deploying and securing docker community edition on AWS linux EC2 instance.

- Create a Key pair

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

KP-1063 [Create new key pair](#)

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...[read more](#)

- Security Group: Allow all the traffic

Additional charges apply when outside of [free tier allowance](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

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

☒ Allow SSH traffic from

☒ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...[read more](#)

ami-00bb6a80f01f03502

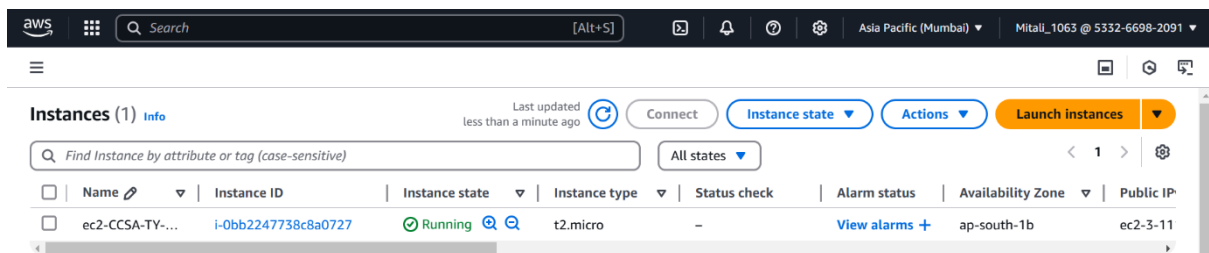
Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

- Launch the instance



The screenshot shows the AWS Management Console interface. At the top, there's a search bar and navigation icons. Below that, the 'Instances' section is active, showing a table with one instance. The instance is named 'ec2-CCSA-TY-...' and is in a 'Running' state. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. The instance is located in the 'ap-south-1b' availability zone and has a public IP of 'ec2-3-11'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
ec2-CCSA-TY-...	i-0bb2247738c8a0727	Running	t2.micro	-	View alarms +	ap-south-1b	ec2-3-11

Step 2: Connect the EC2 Instance and run the following commands:

1. Set up Docker's apt repository.

-sudo apt-get update -y

```
ubuntu@ip-172-31-3-152:~$ sudo apt-get update -y
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152

-sudo apt-get install ca-certificates curl -y

```
Fetches 32.2 MB in 12s (2622 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-3-152:~$ sudo apt-get install ca-certificates curl -y
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152



School of Computer Science, Engineering and Applications (SCSEA)

B.C.A. TY (CCSA)

Subject: Containers & Orchestration

Name of the Student: Mitali Bhattad

PRN: 20220801063

Title of Practical: Deploying and securing docker community edition on AWS linux EC2 instance.

```
-sudo install -m 0755 -d /etc/apt/keyrings
```

```
-sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o  
/etc/apt/keyrings/docker.asc
```

```
-sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
-echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]  
https://download.docker.com/linux/ubuntu \  
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
ubuntu@ip-172-31-3-152:~$ sudo install -m 0755 -d /etc/apt/keyrings  
ubuntu@ip-172-31-3-152:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc  
ubuntu@ip-172-31-3-152:~$ sudo chmod a+r /etc/apt/keyrings/docker.asc  
ubuntu@ip-172-31-3-152:~$ echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \  
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
ubuntu@ip-172-31-3-152:~$
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152

```
-sudo apt-get update
```

```
ubuntu@ip-172-31-3-152:~$ 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  
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]  
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease  
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [18.9 kB]  
Fetched 67.7 kB in 1s (88.2 kB/s)  
Reading package lists... Done  
ubuntu@ip-172-31-3-152:~$
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152

School of Computer Science, Engineering and Applications (SCSEA)
B.C.A. TY (CCSA)
Subject: Containers & Orchestration

Name of the Student: Mitali Bhattad

PRN: 20220801063

Title of Practical: Deploying and securing docker community edition on AWS linux EC2 instance.

2. Install the Docker packages.

-sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

```
ubuntu@ip-172-31-3-152:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

3. To verify the docker installation:

Run the following commands:

- docker -v

```
ubuntu@ip-172-31-3-152:~$ docker -v
Docker version 27.5.1, build 9f9e405
ubuntu@ip-172-31-3-152:~$
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152

4. Verify that the installation is successful by running the hello-world image

- sudo docker run hello-world

```
ubuntu@ip-172-31-3-152:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:d715f14f9eca81473d9112df50457893aa4d099adeb4729f679006bf5ea12407
Status: Downloaded newer image for hello-world:latest
```

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

To generate this message, Docker took the following steps:

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

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152

5. Now, check the docker images

- sudo docker images

```
ubuntu@ip-172-31-3-152:~$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest 74cc54e27dc4 2 weeks ago 10.1kB
ubuntu@ip-172-31-3-152:~$
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152



School of Computer Science, Engineering and Applications (SCSEA)

B.C.A. TY (CCSA)

Subject: Containers & Orchestration

Name of the Student: Mitali Bhattad

PRN: 20220801063

Title of Practical: Deploying and securing docker community edition on AWS linux EC2 instance.

6. Now, to check running containers

- `sudo docker container ls`

7. To check all the container including exited one also (-a stands for all, live and dead also)

- `sudo docker container ls -a`

```
ubuntu@ip-172-31-3-152:~$ sudo docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
ubuntu@ip-172-31-3-152:~$ sudo docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
b70502166939   hello-world   "/hello"   2 minutes ago   Exited (0) 2 minutes ago           dazzling_moser
ubuntu@ip-172-31-3-152:~$
```

i-0bb2247738c8a0727 (ec2-CCSA-TY-1063)

PublicIPs: 3.111.55.38 PrivateIPs: 172.31.3.152