D Y PATIL
INTERNATIONAL
UNIVERSITY
AKURDI PUNE

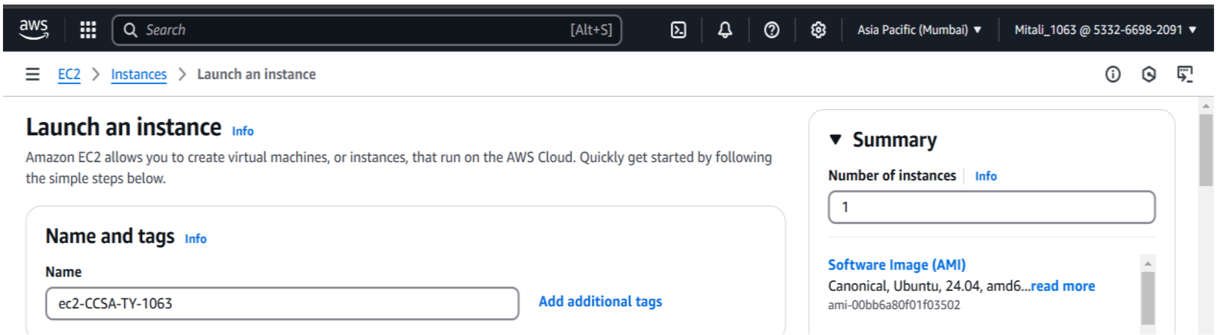
School of Computer Science, Engineering and Applications(SCSEA)
B.C.A. TY (CCSA)
Subject: Containers & Orchestration

Name of the Student: **Mitali Bhattad** **PRN: 20220801063**

Title of Practical: **Deploying Nginx and Apache2 in Docker Containers: Monitoring Running and Exited Containers**

Step1: Launch an EC2 Instance

- Name the Instance



Launch an instance

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

Name and tags

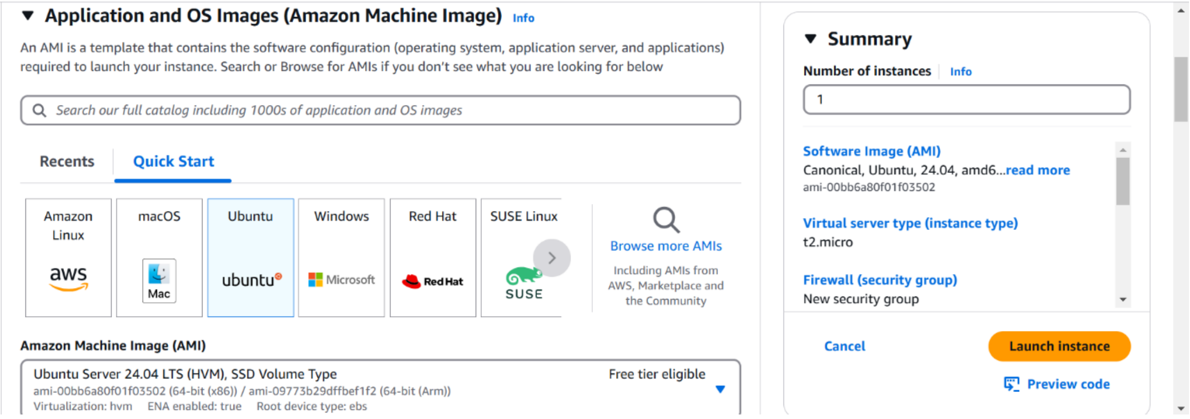
Name: ec2-CCSA-TY-1063

▼ Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

- Choose AMI: Ubuntu



▼ Application and OS Images (Amazon Machine Image)

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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-00bb6a80f01f03502 (64-bit (x86)) / ami-09773b29dffbef1f2 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

▼ Summary

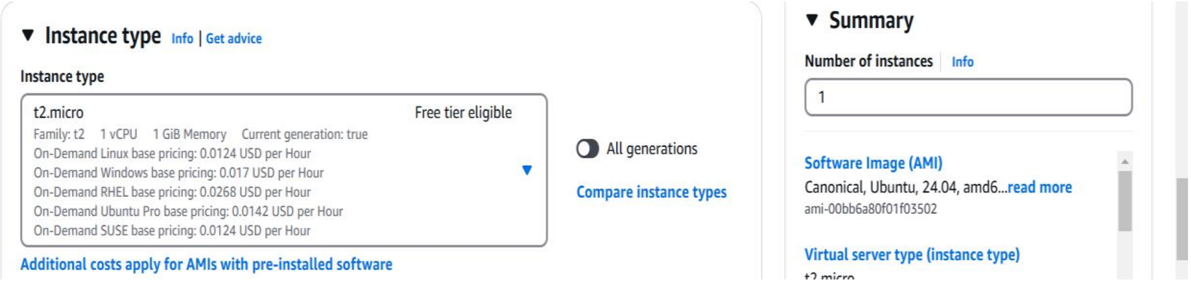
Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

- Select the instance type: t2 micro



▼ Instance type

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

Additional costs apply for AMIs with pre-installed software

▼ Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type)
t2.micro

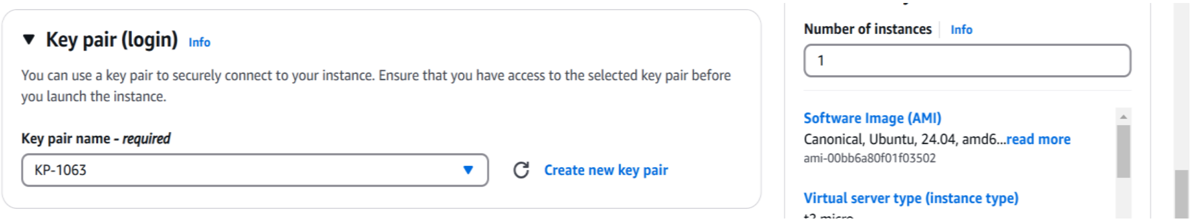
D Y PATIL
INTERNATIONAL
UNIVERSITY
AKURDI PUNE

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 Nginx and Apache2 in Docker Containers: Monitoring Running and Exited Containers**

- 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

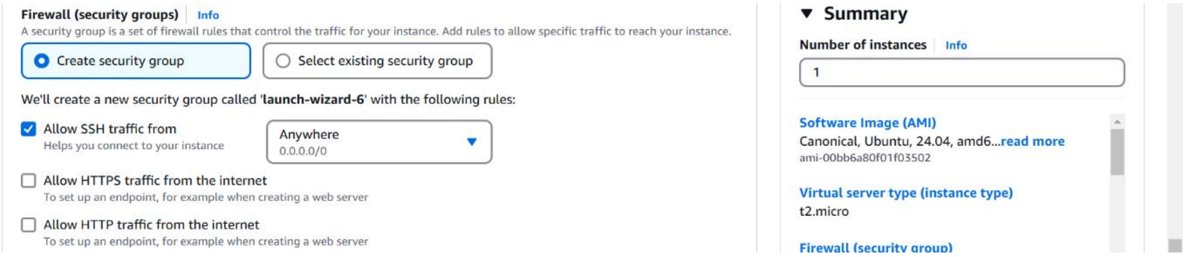
Number of instances Info

1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type)

- Security Group



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

● Create security group ○ Select existing security group

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

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

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

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

▼ Summary

Number of instances Info

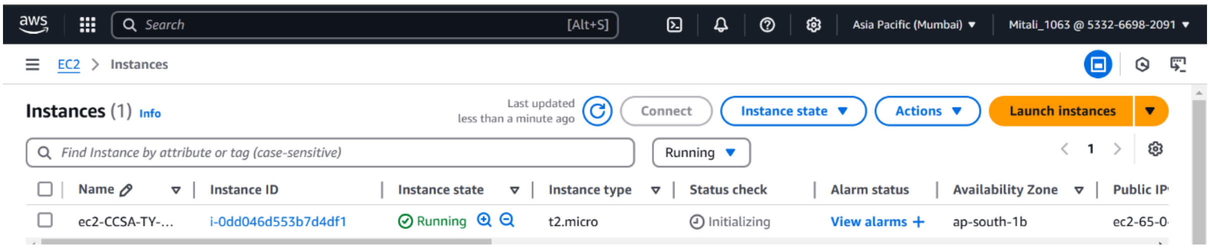
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type)
t2.micro

Firewall (security group)

- Launch the instance



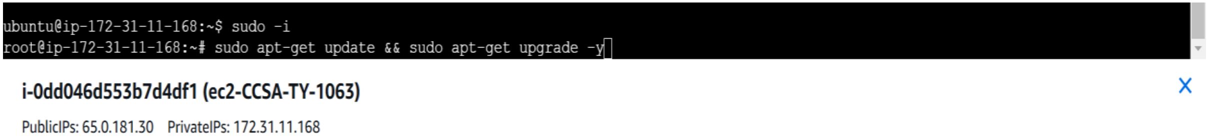
Instances (1) Info

Last updated less than a minute ago

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☐	ec2-CCSA-TY-...	i-0dd046d553b7d4df1	✓ Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-65-0

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

1. Switch to Root user and update and upgrade it
- sudo -i
- sudo apt-get update && sudo apt-get upgrade

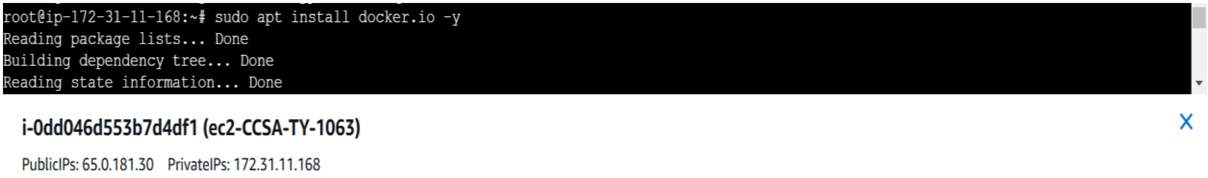


```
ubuntu@ip-172-31-11-168:~$ sudo -i
root@ip-172-31-11-168:~# sudo apt-get update && sudo apt-get upgrade -y
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

2. Install docker
- sudo apt install docker.io -y

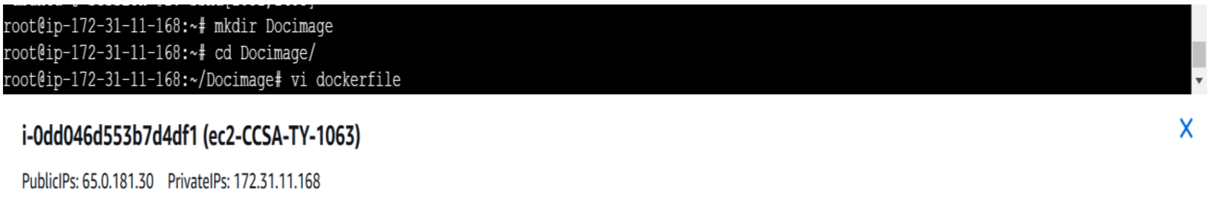


```
root@ip-172-31-11-168:~# sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

3. Create a docker file and name it with vi dockerfile
- mkdir (name-image)
- cd docimage
- vi dockerfile

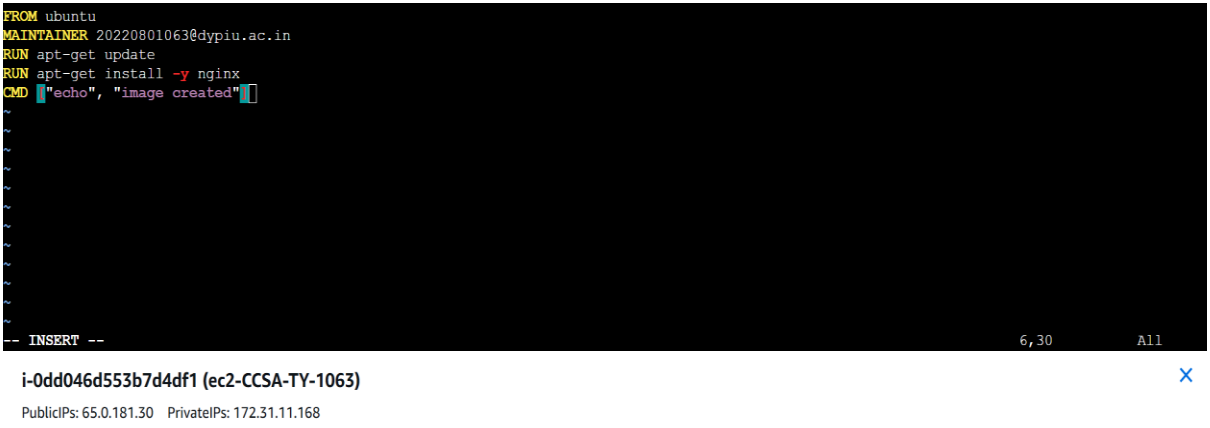


```
root@ip-172-31-11-168:~# mkdir Docimage
root@ip-172-31-11-168:~# cd Docimage/
root@ip-172-31-11-168:~/Docimage# vi dockerfile
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

4. Write a script, press esc:wq to exit

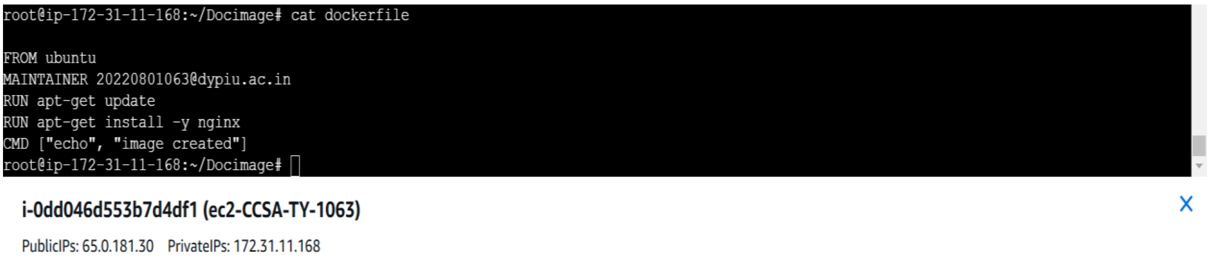


```
FROM ubuntu
MAINTAINER 20220801063@dypiu.ac.in
RUN apt-get update
RUN apt-get install -y nginx
CMD ["echo", "image created"]
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                                          6,30          All
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

5. To verify content of the file
- cat dockerfile



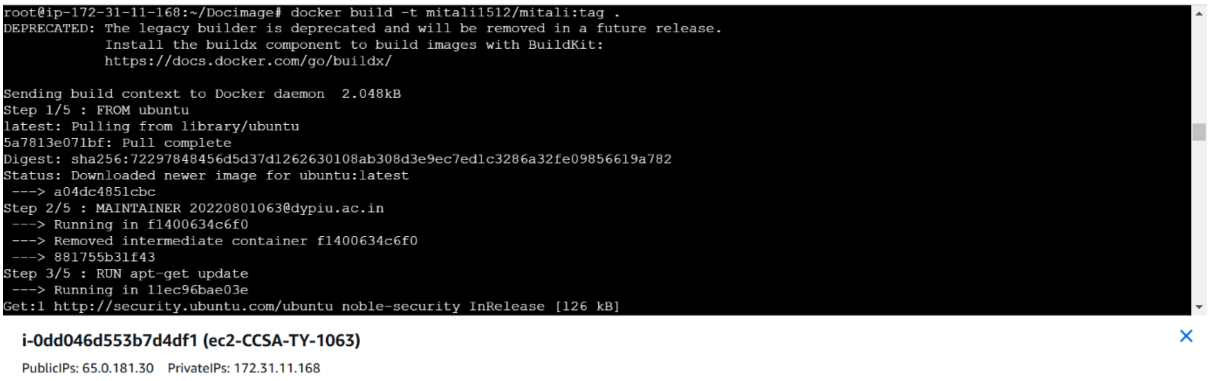
```
root@ip-172-31-11-168:~/Docimage# cat dockerfile

FROM ubuntu
MAINTAINER 20220801063@dypiu.ac.in
RUN apt-get update
RUN apt-get install -y nginx
CMD ["echo", "image created"]
root@ip-172-31-11-168:~/Docimage# 
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

6. To build an image
- docker build -t image username/name: tag .



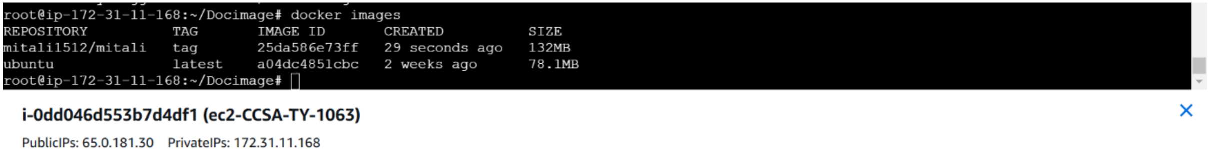
```
root@ip-172-31-11-168:~/Docimage# docker build -t mitali1512/mitali:tag .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  2.048kB
Step 1/5 : FROM ubuntu
latest: Pulling from library/ubuntu
5a7813e071bf: Pull complete
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
 ---> a04dc4851cbc
Step 2/5 : MAINTAINER 20220801063@dypiu.ac.in
 ---> Running in f1400634c6f0
 ---> Removed intermediate container f1400634c6f0
 ---> 881755b31f43
Step 3/5 : RUN apt-get update
 ---> Running in 11ec96bae03e
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

7. To verify image creation
- docker images



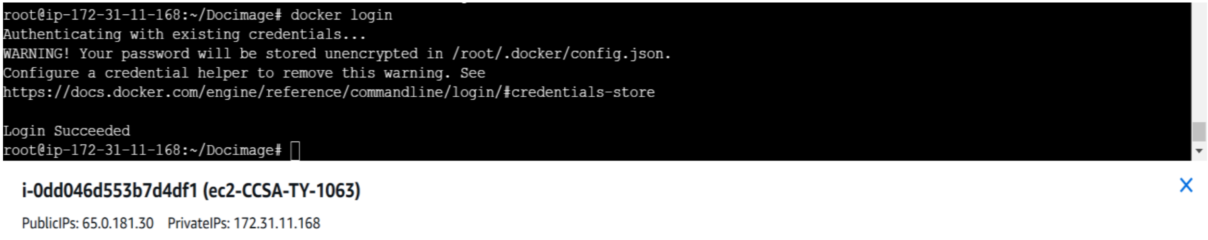
```
root@ip-172-31-11-168:~/Docimage# docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
mitali1512/mitali   tag       25da586e73ff   29 seconds ago   132MB
ubuntu              latest    a04dc4851cbc   2 weeks ago      78.1MB
root@ip-172-31-11-168:~/Docimage# 
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

Step 3: Create docker hub account and insert username and password (password won't be visible but will show you "login succeeded", it means you are into your account)

1. Login and enter username and password
- docker login



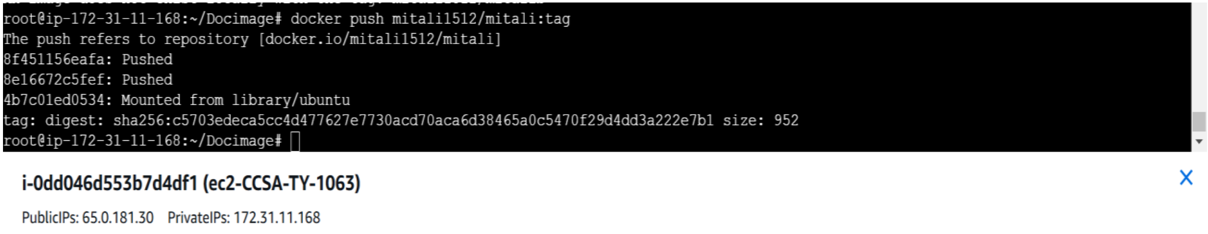
```
root@ip-172-31-11-168:~/Docimage# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@ip-172-31-11-168:~/Docimage# 
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

2. Now, push the image that we created into the docker hub
- docker push username/name:tag

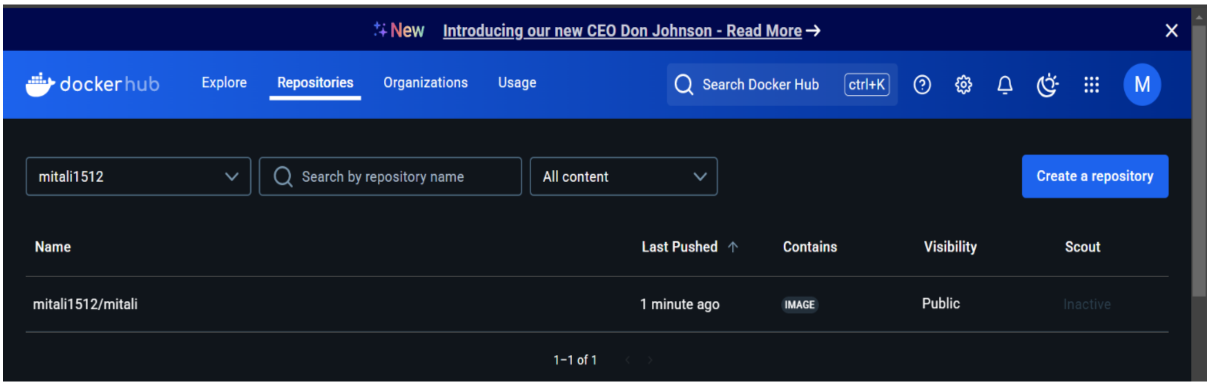


```
root@ip-172-31-11-168:~/Docimage# docker push mitali1512/mitali:tag
The push refers to repository [docker.io/mitali1512/mitali]
8f451156eafa: Pushed
8e16672c5fef: Pushed
4b7c01ed0534: Mounted from library/ubuntu
tag: digest: sha256:c5703edeca5cc4d477627e7730acd70aca6d38465a0c5470f29d4dd3a222e7b1 size: 952
root@ip-172-31-11-168:~/Docimage# 
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

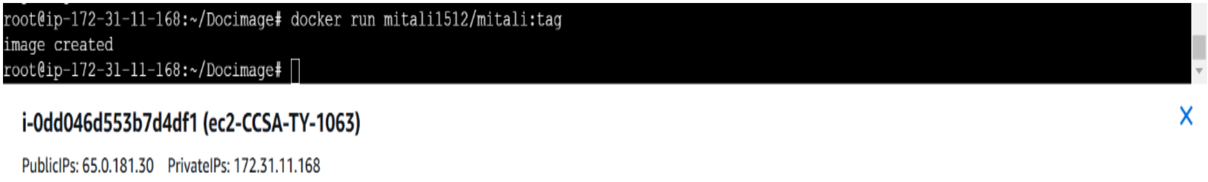
Check in the Docker Hub A/C



Name	Last Pushed	Contains	Visibility	Scout
mitali1512/mitali	1 minute ago	IMAGE	Public	Inactive

1–1 of 1

3. Run the image
- docker run username/name:tag

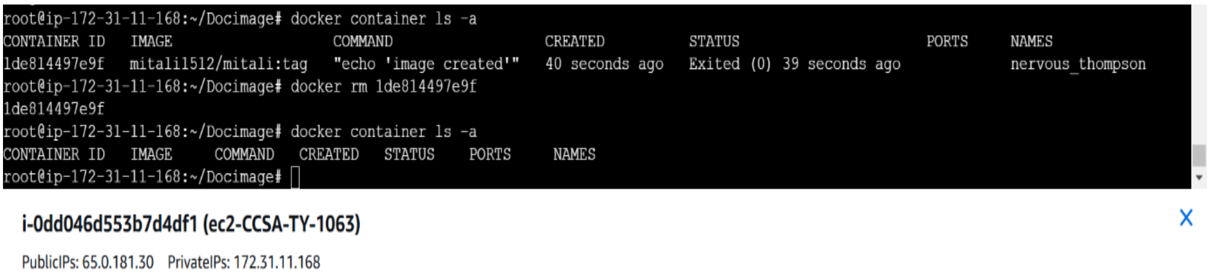


```
root@ip-172-31-11-168:~/Docimage# docker run mitali1512/mitali:tag
image created
root@ip-172-31-11-168:~/Docimage# 
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168

4. To remove the container
- docker rm container-id



```
root@ip-172-31-11-168:~/Docimage# docker container ls -a
CONTAINER ID   IMAGE                COMMAND                 CREATED          STATUS                     PORTS     NAMES
1de814497e9f   mitali1512/mitali:tag   "echo 'image created'"   40 seconds ago   Exited (0) 39 seconds ago             nervous_thompson
root@ip-172-31-11-168:~/Docimage# docker rm 1de814497e9f
1de814497e9f
root@ip-172-31-11-168:~/Docimage# docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-11-168:~/Docimage# 
```

i-0dd046d553b7d4df1 (ec2-CCSA-TY-1063)

PublicIPs: 65.0.181.30 PrivateIPs: 172.31.11.168