

# **CSE 3035**

## **PRINCIPLES OF CLOUD COMPUTING**



### **Lab Assessment – 5**

L15+L16 | SJT501  
Dr. Sivaprakash S

FALL SEMESTER 2022-23

by

**SHARADINDU ADHIKARI**  
**19BCE2105**

## Assessment 5: Docker & Grafana .

### Experiment: 1

**Title:** Docker®.

**Aim:**

- To install docker and run it in AWS.

### Background Theory:

Docker is a set of platforms-as-a-service products that use OS-level virtualization to deliver software in packages called containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine.

### Requirements for conducting the experiment:

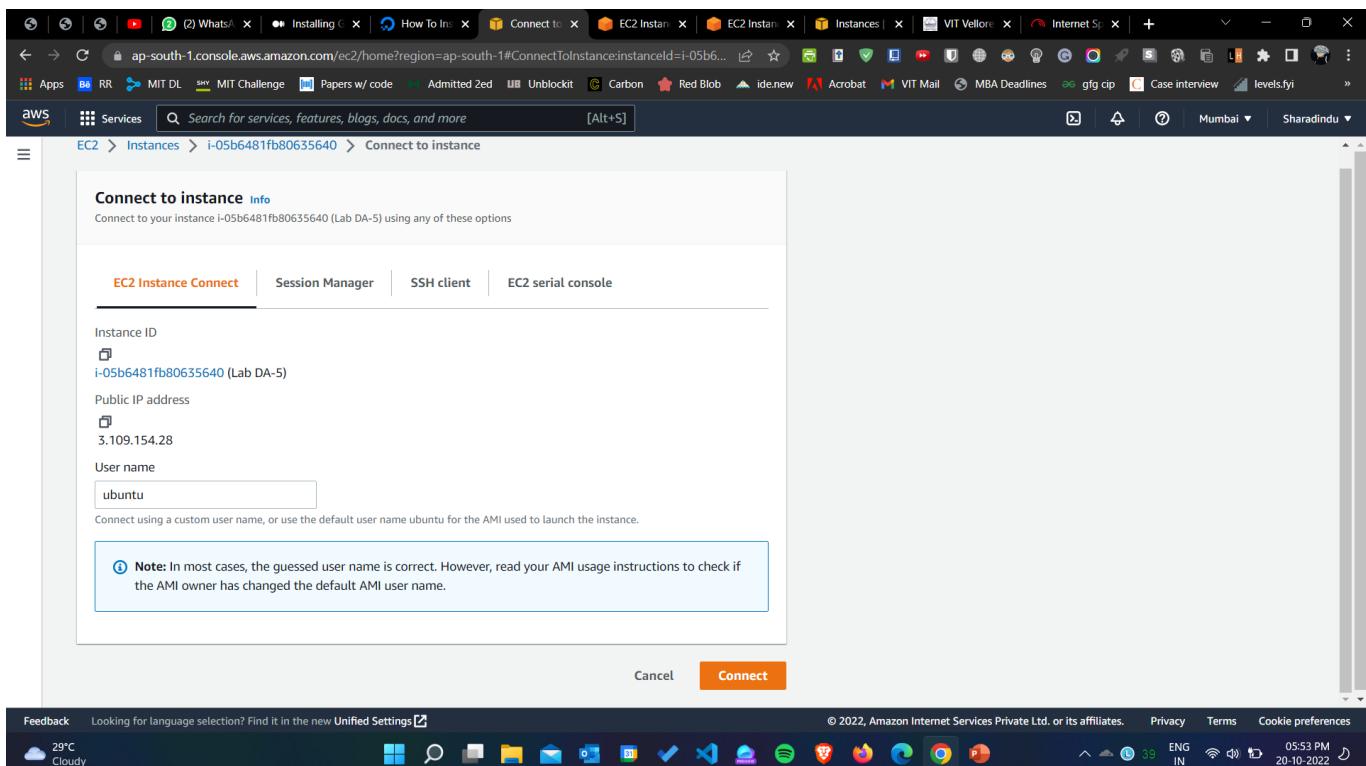
- AWS Licence

### Procedure & Screenshots:

**Step 1:** Create an instance in AWS and launch it. Use Ubuntu (normal) as the OS, while initiating it.

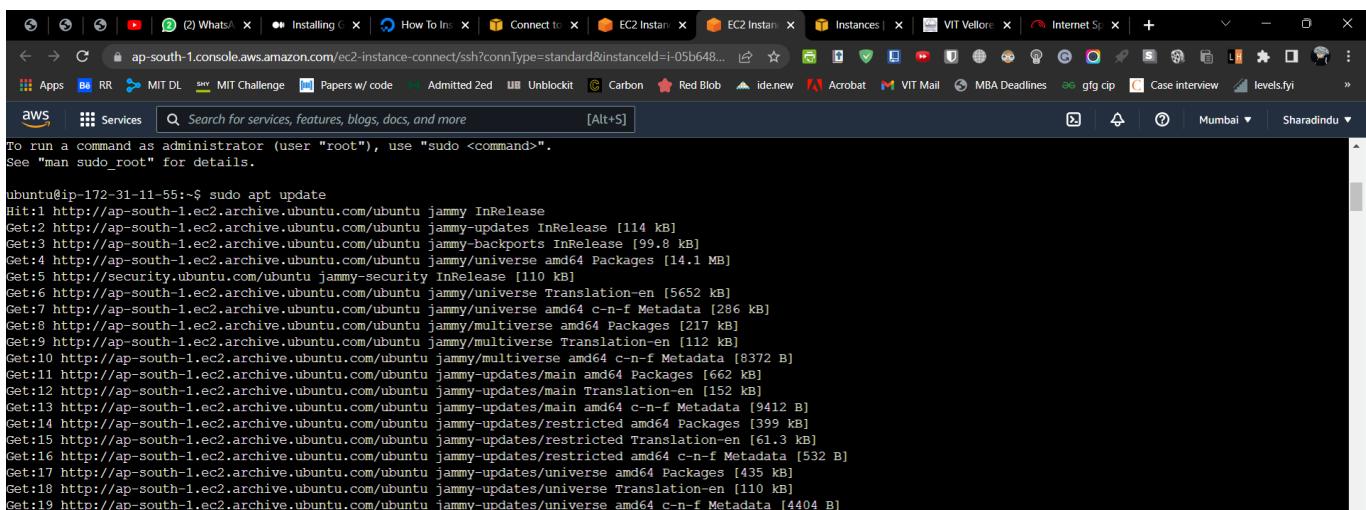
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'New EC2 Experience' selected. Under 'Instances', 'Instances New' is highlighted. The main area shows a table with one row for 'Lab DA-5'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The instance details page below shows the instance summary with fields like Instance ID (i-05b6481fb80635640), Public IPv4 address (3.109.154.28), Private IPv4 address (172.31.11.55), and Public IPv4 DNS (ec2-3-109-154-28.ap-south-1.compute.amazonaws.com).

**Step 2:** Connect to the instance.



### Step 3: Type in the following commands in the terminal to install Docker®:

```
$ sudo apt update
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
$ sudo apt update
$ apt-cache policy docker-ce
$ sudo apt install docker-ce
$ sudo systemctl status docker
```



```

0 upgraded, 9 newly installed, 0 to remove and 54 not upgraded.
Need to get 102 MB of archives.
After this operation, 398 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu/bionic/stable amd64 containerd.io amd64 1.6.8-1 [28.1 MB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:6 https://download.docker.com/linux/ubuntu/bionic/stable amd64 docker-ce-cli amd64 5:20.10.20-3-0~ubuntu-bionic [41.5 MB]
Get:7 https://download.docker.com/linux/ubuntu/bionic/stable amd64 docker-ce amd64 5:20.10.20-3-0~ubuntu-bionic [20.5 kB]
Get:8 https://download.docker.com/linux/ubuntu/bionic/stable amd64 docker-ce-rootless-extras amd64 5:20.10.20-3-0~ubuntu-bionic [8391 kB]
Get:9 https://download.docker.com/linux/ubuntu/bionic/stable amd64 docker-scan-plugin amd64 0.17.0~ubuntu-bionic [3521 kB]
Fetched 102 MB in 2s (64.6 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 63667 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.6.8-1_amd64.deb ...
Unpacking containerd.io (1.6.8-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../2-docker-ce-cli_5%3a20.10.20-3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce-cli (5:20.10.20-3-0~ubuntu-bionic) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../3-docker-ce_5%3a20.10.20-3-0~ubuntu-bionic_amd64.deb ...

```

i-05b6481fb80635640 (Lab DA-5)

PublicIPs: 3.109.154.28 PrivateIPs: 172.31.11.55

```

No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-11-55:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-10-20 12:21:36 UTC; 7s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 3362 (dockerd)
    Tasks: 7
   Memory: 31.2M
      CPU: 260ms
     CGroup: /system.slice/docker.service
             └─3362 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.140866375Z" level=info msg="scheme \"unix\" not registered, fallback to default scheme" module=grpc
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.141039284Z" level=info msg="ccResolverWrapper: sending update to cc: {[{unix:///run/containerd/con>
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.141203961Z" level=info msg="ClientConn switching balancer to \"pick_first\"" module=grpc
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.186789342Z" level=info msg="Loading containers: start."
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.371481764Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16.>
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.468107436Z" level=info msg="Loading containers: done."

```

i-05b6481fb80635640 (Lab DA-5)

PublicIPs: 3.109.154.28 PrivateIPs: 172.31.11.55

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-11-55:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-10-20 12:21:36 UTC; 7s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 3362 (dockerd)
    Tasks: 7
   Memory: 31.2M
      CPU: 260ms
     CGroup: /system.slice/docker.service
             └─3362 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.140866375Z" level=info msg="scheme \"unix\" not registered, fallback to default scheme" module=grpc
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.141039284Z" level=info msg="ccResolverWrapper: sending update to cc: {[{unix:///run/containerd/con>
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.141203961Z" level=info msg="ClientConn switching balancer to \"pick_first\"" module=grpc
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.186789342Z" level=info msg="Loading containers: start."
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.371481764Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16.>
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.468107436Z" level=info msg="Loading containers: done."
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.533167780Z" level=info msg="Docker daemon" commit=03df974 graphdriver(s)=overlay2 version=20.10.20
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.533526025Z" level=info msg="Daemon has completed initialization"
Oct 20 12:21:36 ip-172-31-11-55 systemd[1]: Started Docker Application Container Engine.
Oct 20 12:21:36 ip-172-31-11-55 dockerd[3362]: time="2022-10-20T12:21:36.570272241Z" level=info msg="API listen on /run/docker.sock"
lines 1-22/22 (END)


```

i-05b6481fb80635640 (Lab DA-5)

## Step 4: Verify whether the docker engine has been installed properly, by running:

sudo docker run hello-world

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

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

System information as of Thu Oct 20 12:24:56 UTC 2022

System load: 0.09912109375 Processes: 112
Usage of /: 26.0% of 7.57GB Users logged in: 1
Memory usage: 28% IPv4 address for docker0: 172.17.0.1
Swap usage: 0% IPv4 address for eth0: 172.31.11.55

59 updates can be applied immediately.
31 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Oct 20 12:22:10 2022 from 13.233.177.3
ubuntu@ip-172-31-11-55:~$ sudo docker run hello-world
```



```
2db29710123e: Pull complete
Digest: sha256:18a657d0cc1c7d0678a3fbea8b7eb4918bba25968d3e1b0adebfa71caddbc346
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

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```



**Conclusion:** Docker® has been successfully installed in the AWS Console, and is working properly.

## Experiment: 2

**Title:** Grafana.

**Aim:**

- To install Grafana docker image in AWS EC2.

**Background Theory:**

Grafana is a multi-platform open-source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources.

**Requirements for conducting the experiment:**

- AWS Licence

**Procedure & Screenshots:**

**Step 1:** Create an instance in AWS and launch it. Use AWS-Ubuntu as the OS here, while initiating it.

The screenshot shows the AWS Management Console with the EC2 Instances page open. A single instance named "Lab DA-5 II" is listed, showing it is running (t2.micro) and has passed 2/2 checks. The instance ID is i-02c6e0c2843f4f553. The screenshot also shows the AWS navigation bar and various service links.

**Step 2:** Connect to the instance.

**Connect to instance Info**

Connect to your instance i-02c6e0c2843f4f553 (Lab DA-5 II) using any of these options

**EC2 Instance Connect** Session Manager SSH client EC2 serial console

Instance ID  
i-02c6e0c2843f4f553 (Lab DA-5 II)

Public IP address  
15.206.178.120

User name  
ec2-user

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

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

Cancel Connect

```
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-5-19 ~]$
```

**Step 3:** Before running commands in the AWS terminal, we need to add a port first. Go to the main dashboard of EC2 → Security → Security Groups → Inbound rules

Find instance by attribute or tag (case-sensitive)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
<input checked="" type="checkbox"/> Lab DA-5 II	i-02c6e0c2843f4f553	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	+ ap-south-1b	ec2-15-206-17
<input type="checkbox"/> Lab DA-5	i-05b6481fb0635640	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	+ ap-south-1b	ec2-3-109-154

**Instance: i-02c6e0c2843f4f553 (Lab DA-5 II)**

Details Security Networking Storage Status checks Monitoring Tags

**Security details**

IAM Role  
-

Owner ID  
917110574018

Launch time  
Thu Oct 20 2022 17:45:29 GMT+0530 (India Standard Time)

Security groups  
sg-0621558e0729693a6 (launch-wizard-15)

**Inbound rules**

Filter rules

## Step 4: Add a Custom TCP Port, of range 3000. And save the rules.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0e5caeb34905ed4c	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-061b17e6bb852bfae	SSH	TCP	22	Custom	0.0.0.0/0
sgr-0a792d61257232b38	HTTPS	TCP	443	Custom	0.0.0.0/0
-	Custom TCP	TCP	3000	Anywhere	0.0.0.0/0

**Add rule**

**Save rules**

**Inbound security group rules successfully modified on security group (sg-0621558e0729693a6 | launch-wizard-15)**

**Details**

**sg-0621558e0729693a6 - launch-wizard-15**

**Inbound rules**

You can now check network connectivity with Reachability Analyzer

**Run Reachability Analyzer**

## Step 5: Now come back to the terminal, and initiate the following commands to update the packages.

```
$ sudo yum update -y
$ sudo vim /etc/yum.repos.d/grafana.repo
```

## Step 6: Add the text below to the repo file. This will install the open-source Grafana.

```
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

The screenshot shows a terminal window with the configuration file for Grafana. The file contains the following content:

```
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

## Step 7: Install Grafana.

```
$ sudo yum install grafana -y
$ sudo systemctl daemon-reload
$ sudo systemctl start grafana-server
$ sudo systemctl status grafana-server
$ sudo systemctl enable grafana-server.service
```

The screenshot shows a terminal window with the command history and output of the Grafana installation process. The output includes:

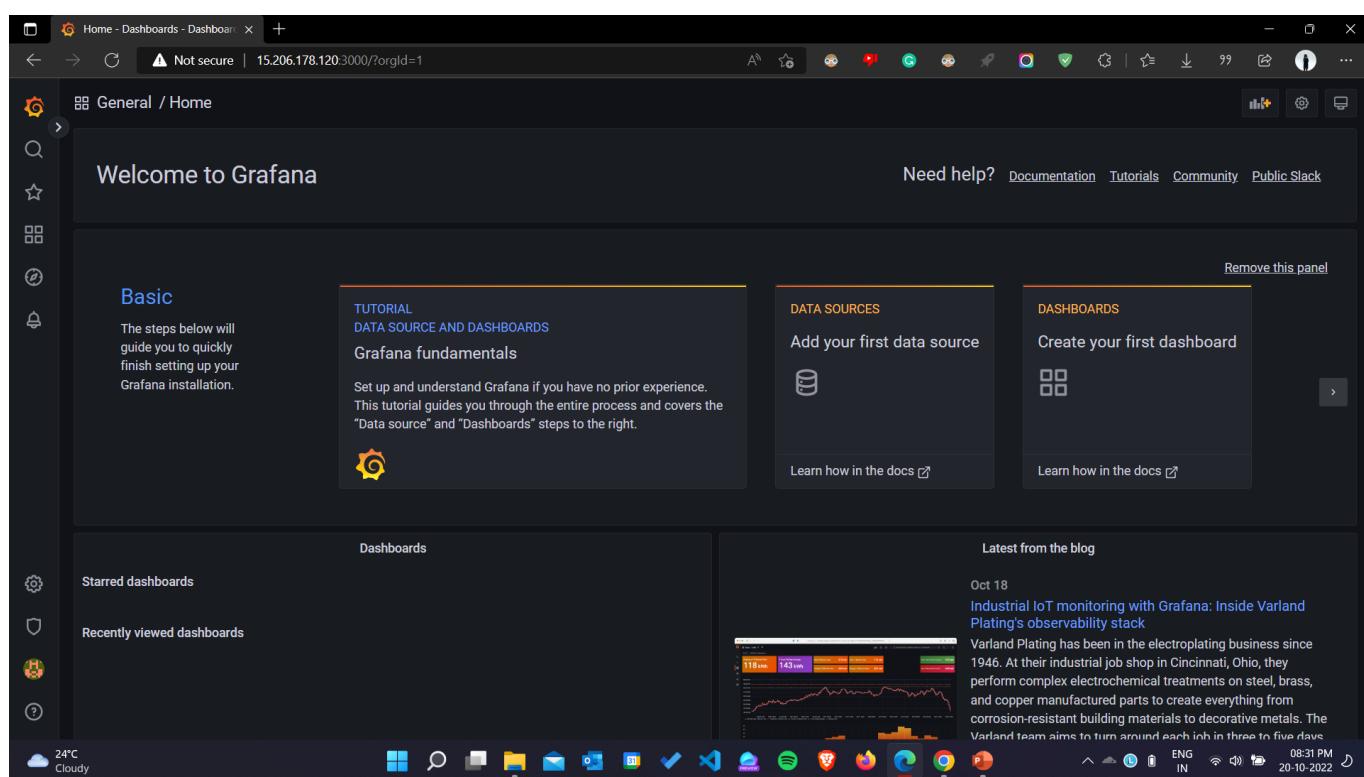
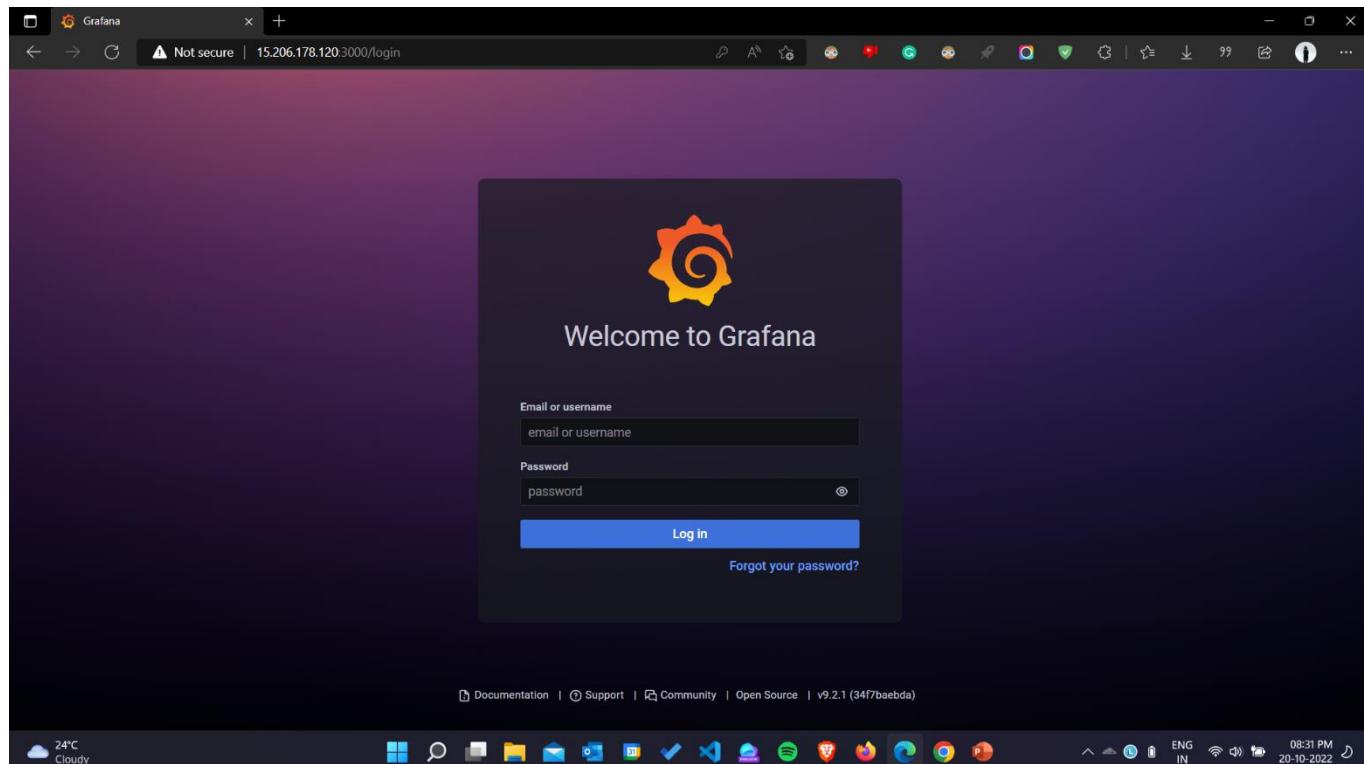
```
xorg-x11-font-utils.x86_64 1:7.5-21.amzn2
xorg-x11-server-utils.x86_64 0:7.7-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-5-19 ~]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-5-19 ~]$ sudo systemctl start grafana-server
[ec2-user@ip-172-31-5-19 ~]$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-10-20 12:31:17 UTC; 6s ago
     Docs: http://docs.grafana.org
Main PID: 7088 (grafana-server)
   CGroup: /system.slice/grafana-server.service
           └─7088 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/var/run/grafana/grafana-server.pid --packaging=rpm cfg:default.paths.logs=/var/lo...

Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=infra.usagestats.collector t=2022-10-20T12:31:17.0694213Z level=info msg="...rsLen=2
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=server t=2022-10-20T12:31:17.070028624Z level=info msg="Writing PID file" ...id=7088
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=provisioning.alerting t=2022-10-20T12:31:17.070854157Z level=info msg="start...
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=provisioning.alerting t=2022-10-20T12:31:17.071175232Z level=info msg="fini...
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal systemd[1]: Started Grafana instance.
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=http.server t=2022-10-20T12:31:17.089796796Z level=info msg="HTTP Server L...socket=
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=ngealert t=2022-10-20T12:31:17.090159974Z level=info msg="warming cache for startup"
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=ticker t=2022-10-20T12:31:17.09060101Z level=info msg="starting first_tick=....31:20Z
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=grafanaStorageLogger t=2022-10-20T12:31:17.099644243Z level=info msg="stor...
Oct 20 12:31:17 ip-172-31-5-19.ap-south-1.compute.internal grafana-server[7088]: logger=ngealert.multiorg.alertmanager t=2022-10-20T12:31:17.100639885Z level=info ...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-5-19 ~]$
```

At the bottom of the terminal window, there is a message: "i-02c60c2843f4f553 (Lab DA-5 II)" and "PublicIPs: 15.206.178.120 PrivateIPs: 172.31.15.19".

**Step 8:** Take the public IPV4 address of the instance and add a :3000 at the end, i.e. 15.206.178.120:3000, and insert it into our browser URL. This will bring us to a login screen and our Username and Password will be *admin*. We will be prompted to create a new password.



**Conclusion:** Grafana has been installed successfully in AWS EC2, and is running properly.