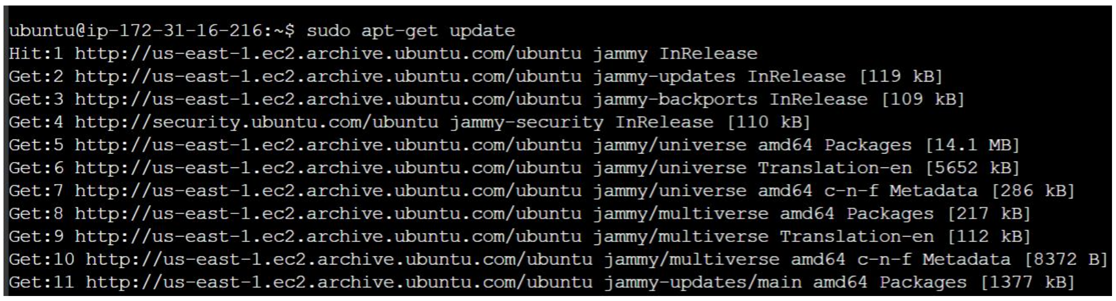
DHARANESH T K
ACE11651

DOCKER - Creation

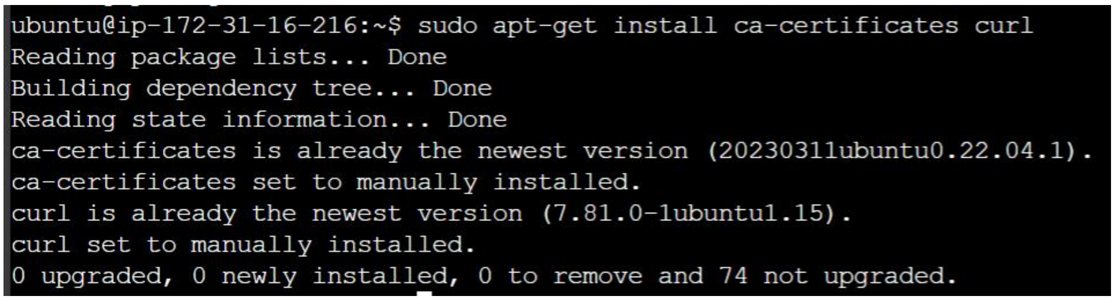
- First create a instance in Amazon Web service.

- **sudo apt-get update**: This command updates the package lists for all repositories in the APT sources configuration. It fetches the latest information about available packages and their versions.



```
ubuntu@ip-172-31-16-216:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1377 kB]
```

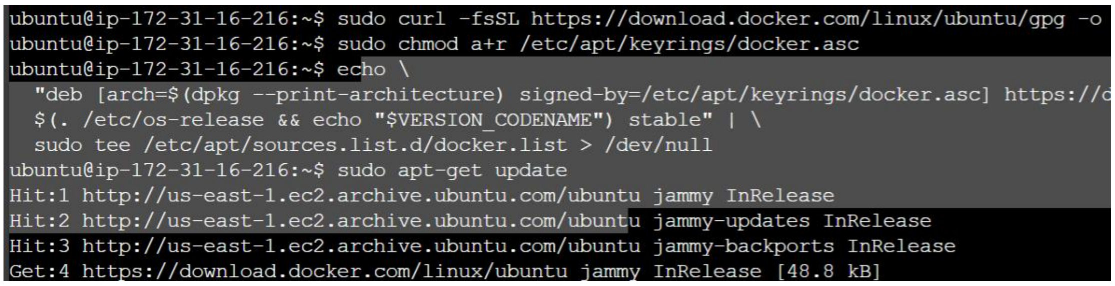
- **sudo apt-get install ca-certificates curl**: This command installs the “ca-certificates” package, which contains the public keys for various Certificate Authorities, and “curl”, a command-line tool for transferring data using various protocols, including HTTP and HTTPS.



```
ubuntu@ip-172-31-16-216:~$ sudo apt-get install ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.15).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
```

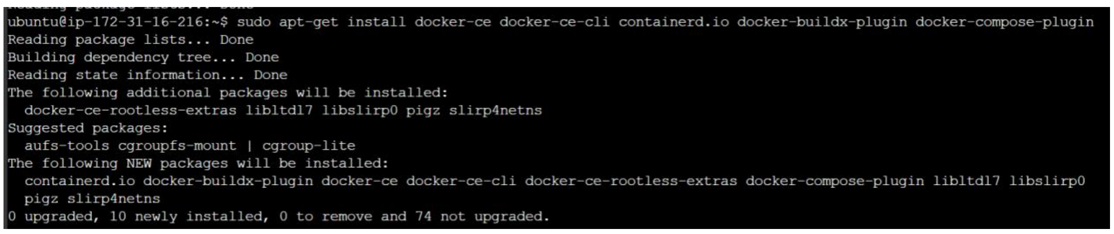
- **sudo chmod a+r /etc/apt/keyrings/docker.asc**: This command changes the permissions of the downloaded docker.asc file to make it readable (a+r) by all users (a stands for all). This step is necessary to allow APT to read the GPG key from this file during package verification.

- Next echo command : This command appends a new entry to the APT sources list by echoing a repository configuration line into /etc/apt/sources.list.d/docker.list



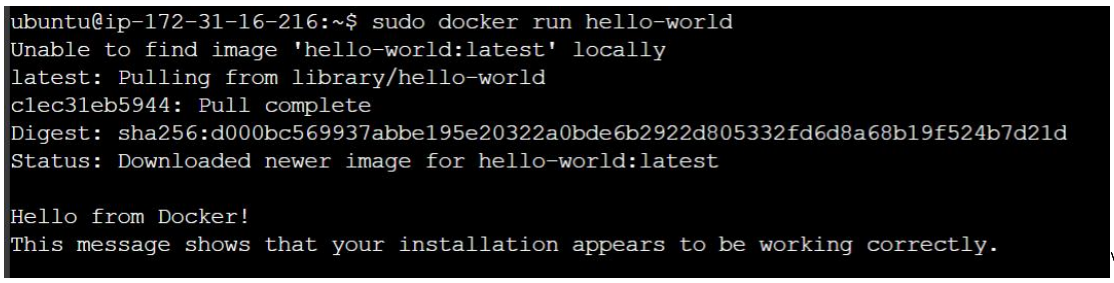
```
ubuntu@ip-172-31-16-216:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
ubuntu@ip-172-31-16-216:~$ sudo chmod a+r /etc/apt/keyrings/docker.asc
ubuntu@ip-172-31-16-216:~$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://d
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ubuntu@ip-172-31-16-216:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
```

➢ Below command installs Docker Community Edition (docker-ce), Docker CLI (docker-ce-cli), containerd (containerd.io), Docker Buildx plugin (docker-buildx-plugin), and Docker Compose plugin (docker-compose-plugin) using the APT package manager in one go.



```
ubuntu@ip-172-31-16-216:~$ 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
The following additional packages will be installed:
  docker-ce-rootless-extras libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0
  pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 74 not upgraded.
```

➢ Below command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits.



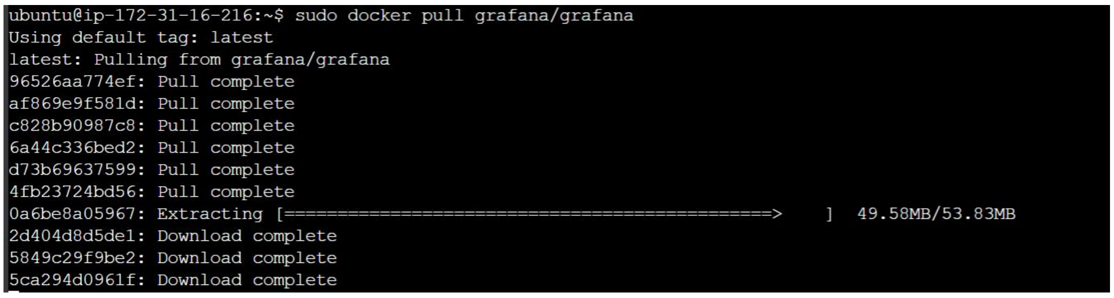
```
ubuntu@ip-172-31-16-216:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d000bc569937abbe195e20322a0bde6b2922d805332fd6d8a68b19f524b7d21d
Status: Downloaded newer image for hello-world:latest

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

\

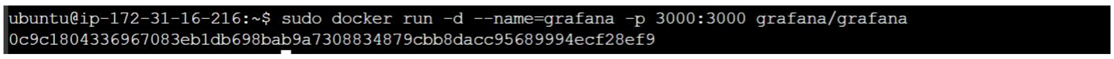
✓ Once docker setup is done, then move to container creation.

➢ Docker will download the latest version of the Grafana image from the official repository on Docker Hub and store it locally on your machine



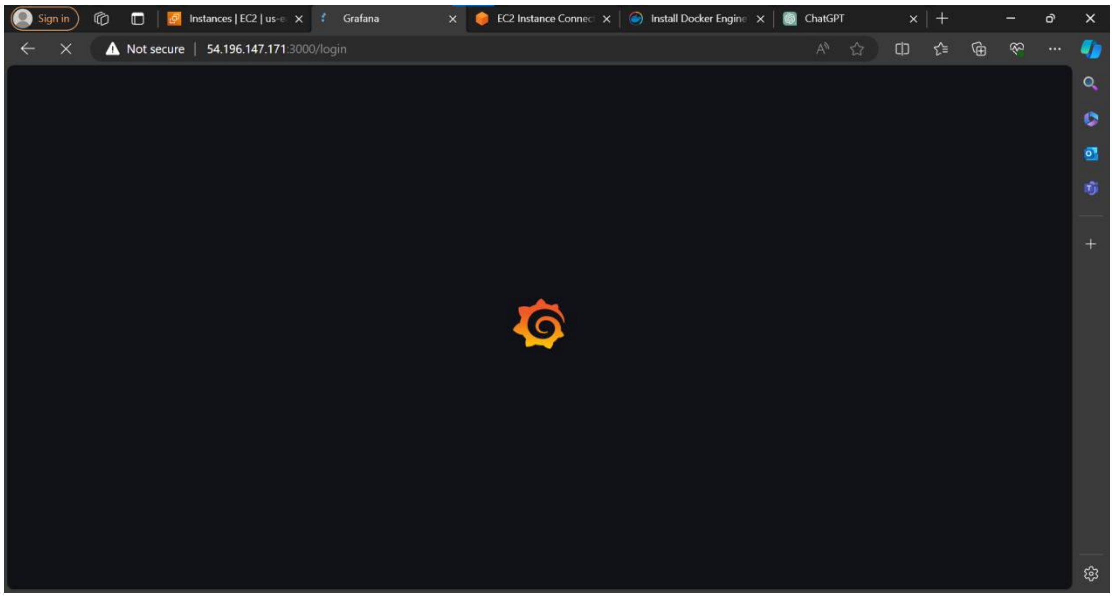
```
ubuntu@ip-172-31-16-216:~$ sudo docker pull grafana/grafana
Using default tag: latest
latest: Pulling from grafana/grafana
96526aa774ef: Pull complete
af869e9f581d: Pull complete
c828b90987c8: Pull complete
6a44c336bed2: Pull complete
d73b69637599: Pull complete
4fb23724bd56: Pull complete
0a6be8a05967: Extracting [=============================================>     ]  49.58MB/53.83MB
2d404d8d5de1: Download complete
5849c29f9be2: Download complete
5ca294d0961f: Download complete
```

➢ grafana/grafana is used to run a Grafana container in detached mode, assigning it the name "grafana" and mapping port 3000 of the container to port 3000 on the host machine.

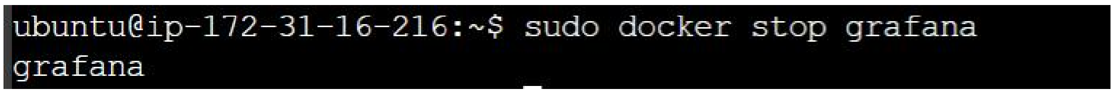


```
ubuntu@ip-172-31-16-216:~$ sudo docker run -d --name=grafana -p 3000:3000 grafana/grafana
0c9c1804336967083eb1db698bab9a7308834879cbb8dacc95689994ecf28ef9
```

Created a Grafana using Docker and hosted successfully.



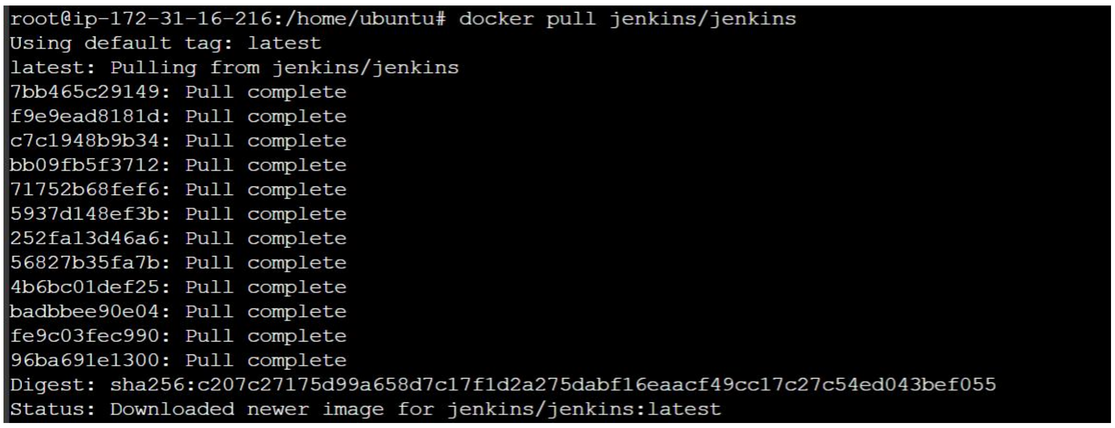
➢ After running **docker stop grafana**, the container will no longer be running, but it will still exist on your system.



```
ubuntu@ip-172-31-16-216:~$ sudo docker stop grafana
grafana
```

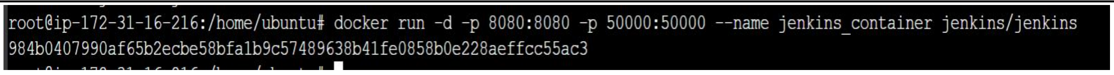
✓ Installing another container in same docker.

➢ Docker will download the latest version of the Jenkins image from the official repository on Docker Hub and store it locally on your machine



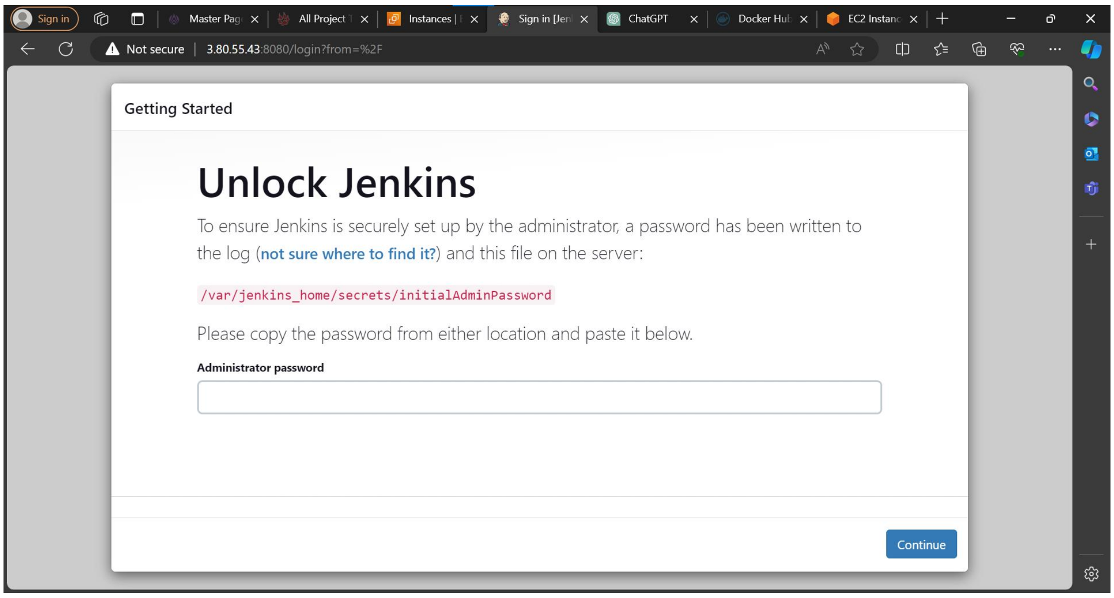
```
root@ip-172-31-16-216:/home/ubuntu# docker pull jenkins/jenkins
Using default tag: latest
latest: Pulling from jenkins/jenkins
7bb465c29149: Pull complete
f9e9ead8181d: Pull complete
c7c1948b9b34: Pull complete
bb09fb5f3712: Pull complete
71752b68fef6: Pull complete
5937d148ef3b: Pull complete
252fa13d46a6: Pull complete
56827b35fa7b: Pull complete
4b6bc01def25: Pull complete
badbbee90e04: Pull complete
fe9c03fec990: Pull complete
96ba691e1300: Pull complete
Digest: sha256:c207c27175d99a658d7c17f1d2a275dabf16eaacf49cc17c27c54ed043bef055
Status: Downloaded newer image for jenkins/jenkins:latest
```

This command starts a Docker container named "jenkins_container" based on the jenkins/jenkins image. It runs the container in detached mode (-d), meaning it runs in the background. It also maps port 8080 of the host to port 8080 of the container (-p 8080:8080) for accessing Jenkins web interface and maps port 50000 of the host to port 50000 of the container for Jenkins agent communication.



```
root@ip-172-31-16-216:/home/ubuntu# docker run -d -p 8080:8080 -p 50000:50000 --name jenkins_container jenkins/jenkins
984b0407990af65b2ecbe58bfa1b9c57489638b41fe0858b0e228aeffcc55ac3
```

Created a Jenkins using Docker and hosted successfully.



Getting Started

Unlock Jenkins

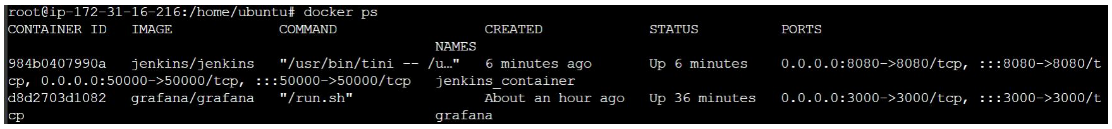
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

The command **docker ps** is used to list all the currently running Docker containers.



```
root@ip-172-31-16-216:/home/ubuntu# docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED             STATUS          PORTS
                                                    NAMES
984b0407990a   jenkins/jenkins   "/usr/bin/tini -- /u…"   6 minutes ago       Up 6 minutes    0.0.0.0:8080->8080/tcp, :::8080->8080/t
cp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp   jenkins_container
d8d2703d1082   grafana/grafana   "/run.sh"                About an hour ago   Up 36 minutes   0.0.0.0:3000->3000/tcp, :::3000->3000/t
cp                                                  grafana
```