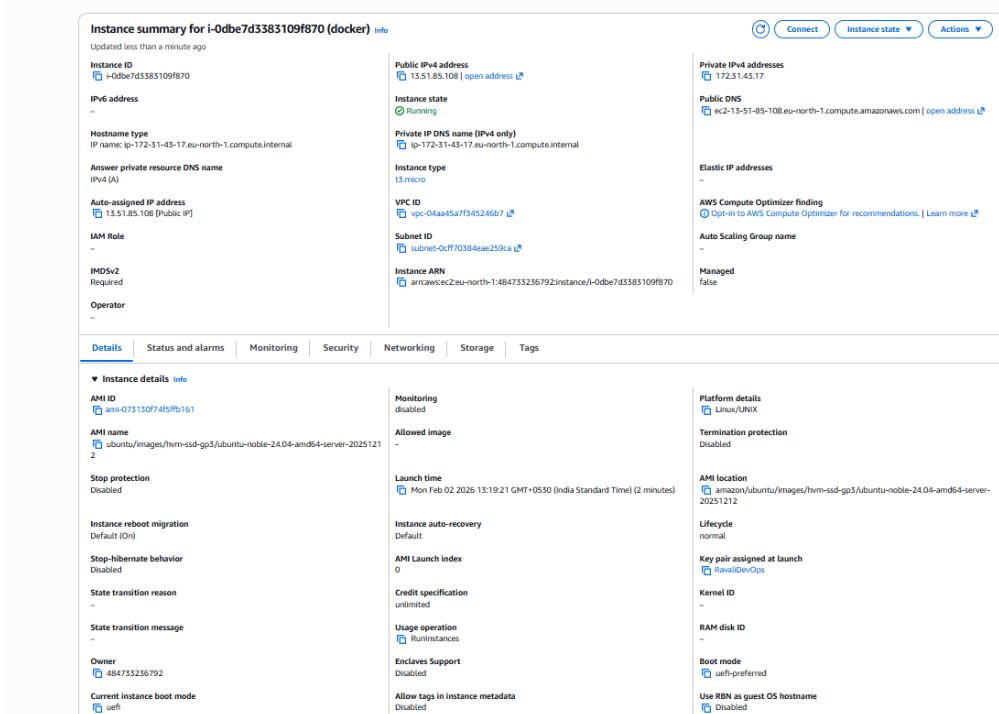


Task

Task 1: Install docker on EC2 and explore the docker commands (docker images, containers, volumes, network).

1. Create EC2 instances.



The screenshot shows the AWS CloudWatch Metrics interface. A single metric named 'CPU Utilization' is displayed across three dimensions: Region (eu-north-1), Service (AmazonEC2), and Metric Type (CPUUtilization). The chart shows values ranging from approximately 0% to 100% over a period of about 10 hours. The Y-axis is labeled 'CPU Utilization (%)' and the X-axis shows time from 10:00 to 20:00. The legend indicates the series is 'CPU Utilization'.

2. Installing docker on EC2

```
ubuntu@ip-172-31-43-17:~$ docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
ubuntu@ip-172-31-43-17:~$ |
```

3. explore the docker commands

```
ubuntu@ip-172-31-43-17:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
119d43eec815: Pull complete
700146c8ad64: Pull complete
d989100b8a84: Pull complete
500799c30424: Pull complete
10b68cfefee1: Pull complete
57f0dd1befc2: Pull complete
eaf8753feae0: Pull complete
Digest: sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf4aec
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
ubuntu@ip-172-31-43-17:~$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
nginx           latest       4af177a024eb   2 weeks ago   161MB
```

4. Creating containers.

```
nginx      /var/www/html   2 weeks ago  101MB
ubuntu@ip-172-31-43-17:~$ docker run -d --name mynginx -p 80:80 nginx
67c45fefba124f1af331870d7ff2e91294dba43981cf93a234e0f82a88187f7e
ubuntu@ip-172-31-43-17:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
67c45fefba12 nginx "/docker-entrypoint..." 8 seconds ago Up 7 seconds
0.0.0.0:80->80/tcp, [::]:80->80/tcp mynginx
ubuntu@ip-172-31-43-17:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
67c45fefba12 nginx "/docker-entrypoint..." 2 minutes ago Up 2 minutes 0.0.0.0:80->80/tcp, [::]:80->80/tcp mynginx
ubuntu@ip-172-31-43-17:~$ docker stop mynginx
```

```
ubuntu@ip-172-31-43-17:~$ docker stop mynginx
mynginx
ubuntu@ip-172-31-43-17:~$ |
```



This site can't be reached

13.51.85.108 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Reload

```
mynginx
ubuntu@ip-172-31-43-17:~$ docker start mynginx
mynginx
ubuntu@ip-172-31-43-17:~$ |
```

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

```
mynginx
ubuntu@ip-172-31-43-17:~$ docker rm mynginx
mynginx
ubuntu@ip-172-31-43-17:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@ip-172-31-43-17:~$ |
```

5. Docker volumes.

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@ip-172-31-43-17:~$ docker volume create myvolume
myvolume
ubuntu@ip-172-31-43-17:~$ docker volume ls
DRIVER VOLUME NAME
local myvolume
ubuntu@ip-172-31-43-17:~$ docker run -d \
--name volcontainer \
-v myvolume:/usr/share/nginx/html \
nginx
53a3e79f0cbfdb9cfbba640683d7b52d23d53e7665a295fc28562105661843a4
ubuntu@ip-172-31-43-17:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
53a3e79f0cbf nginx "/docker-entrypoint..." 11 seconds ago Up 10 seconds 80/tcp volcontainer
```

6. Docker network

```
ubuntu@ip-172-31-43-17:~$ docker network create mynetwork
fac8b3e86b3e465698204fff5fe8103d34b735d85d164bb944063b6cf5db8165
ubuntu@ip-172-31-43-17:~$ docker run -d --name netcontainer --network mynetwork nginx
a118487cdc375fae929d3edc25d9cd4e46a24b01e7ad6559610f17137c284171
ubuntu@ip-172-31-43-17:~$ docker network ls
NETWORK ID     NAME      DRIVER      SCOPE
19ddc20fb2f7   bridge    bridge      local
7eb787865b22   host      host       local
fac8b3e86b3e   mynetwork  bridge      local
53ab184ab9a2   none      null      local
ubuntu@ip-172-31-43-17:~$ docker network inspect mynetwork
[
  {
    "Name": "mynetwork",
    "Id": "fac8b3e86b3e465698204fff5fe8103d34b735d85d164bb944063b6cf5db8165",
    "Created": "2026-02-02T08:24:22.12012087Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "a118487cdc375fae929d3edc25d9cd4e46a24b01e7ad6559610f17137c284171": {
        "Name": "netcontainer",
        "EndpointID": "0f676334c698840c31a807b00b15a1a031203e18e25901e10604c3ca7619c56d",
        "MacAddress": "5a:92:99:fe:e7:0c",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
```