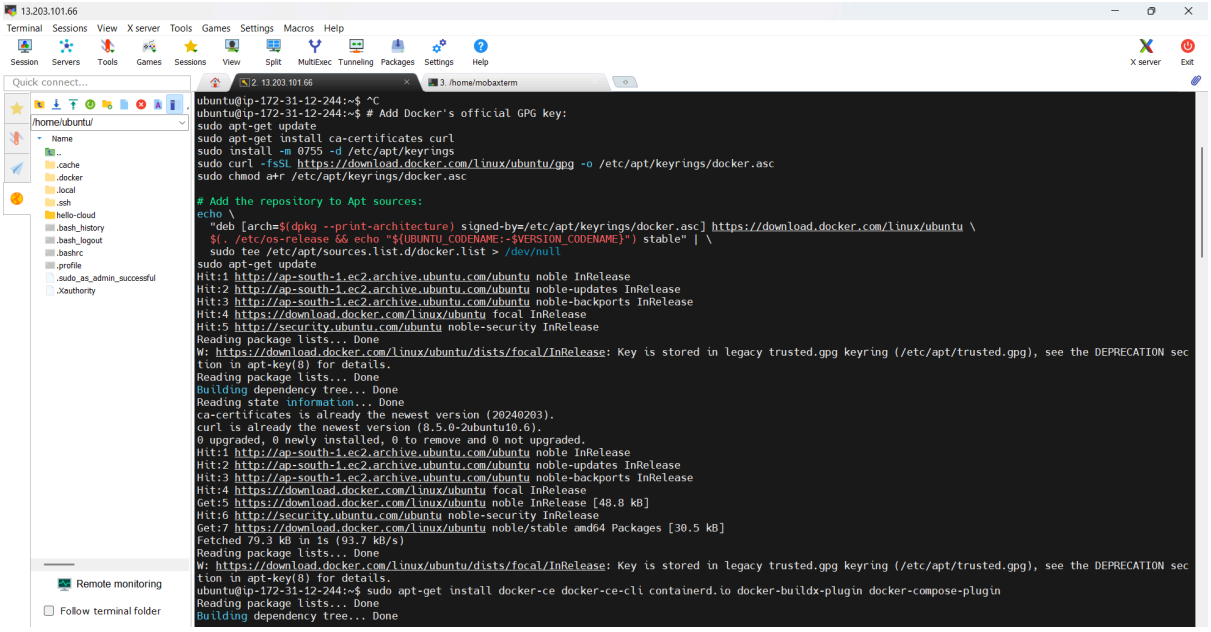
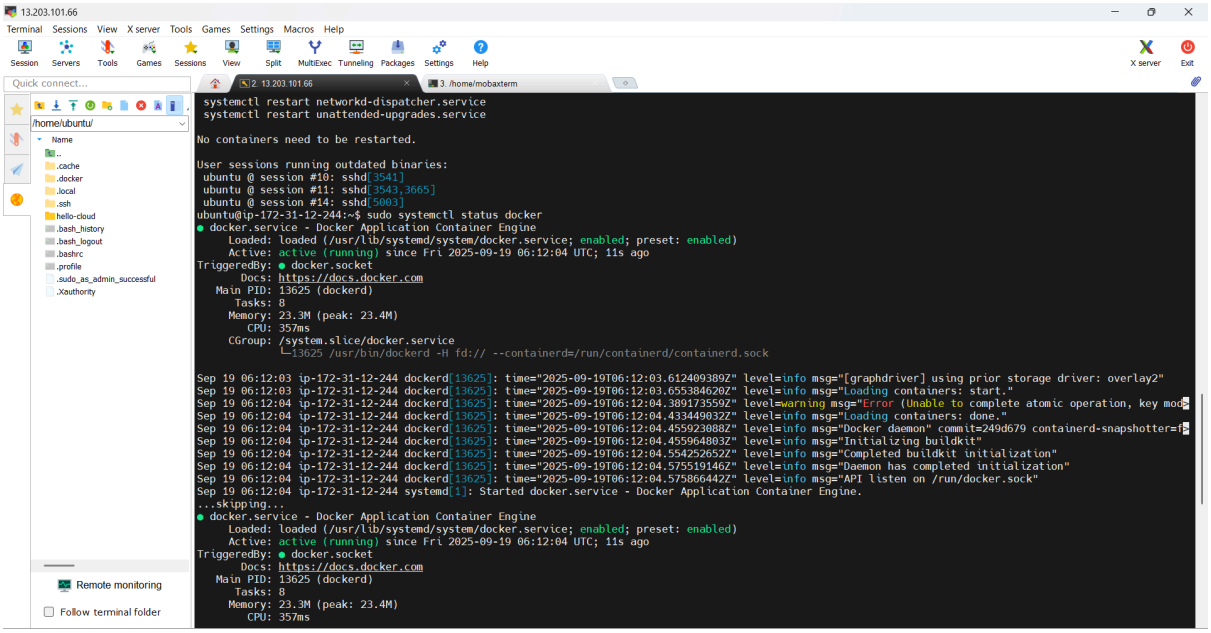
Documentations

- **Below the Screenshots is**

1.Docker installation & version check.

- Docker was successfully installed and verified using the “docker –version” command, confirming that the system is ready for containerization.





```
Remote monitoring
Follow terminal folder
ubuntu@ip-172-31-12-244:~$ sudo systemctl start docker
ubuntu@ip-172-31-12-244:~$ docker --version
Docker version 28.4.0, build d8eb465
ubuntu@ip-172-31-12-244:~$
```

2.**Docker file creation & Container build**

- A Dockerfile was created and used to build a custom image(hello-cloud:1.0).
- This define the environment, dependencies, and instructions needed to run the application inside a container.

```
ubuntu@ip-172-31-12-244:~$ cd hello-cloud
ubuntu@ip-172-31-12-244:~/hello-cloud$ nano requirements.txt
ubuntu@ip-172-31-12-244:~/hello-cloud$ nano app.py
ubuntu@ip-172-31-12-244:~/hello-cloud$ nano Dockerfile
ubuntu@ip-172-31-12-244:~/hello-cloud$ ls
Dockerfile  app.py  requirements.txt
ubuntu@ip-172-31-12-244:~/hello-cloud$ docker build -t hello-cloud:1.0 .
[+] Building 2.2s (10/10) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 421B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.11-slim                                                           2.0s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [1/5] FROM docker.io/library/python:3.11-slim@sha256:a0939570b38cddeb861b8e75d20b1c8218b21562b18f301171904b544e8cf228      0.0s
 => [internal] load build context                                                                                             0.0s
 => => transferring context: 738B                                                                                             0.0s
 => CACHED [2/5] WORKDIR /app                                                                                                 0.0s
 => CACHED [3/5] COPY requirements.txt .                                                                                      0.0s
 => CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt                                                           0.0s
 => CACHED [5/5] COPY . .                                                                                                     0.0s
 => exporting to image                                                                                                        0.0s
 => => exporting layers                                                                                                       0.0s
 => => writing image sha256:02c513ac735592e96bf1d8f79348ca03f30336878723ceda3e651b0adaf9c24a                                  0.0s
 => => naming to docker.io/library/hello-cloud:1.0                                                                            0.0s
```

3.**Container running and webpage accessible in browser**

- **Container Status Check:**

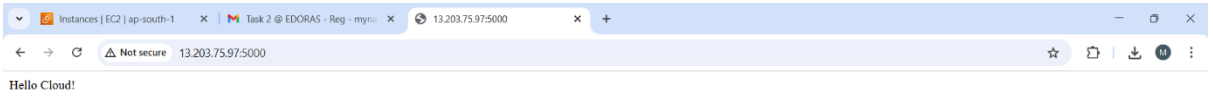
This is important for ensuring that the container is actually running. You can verify this with a command like ***"docker ps"*** ,which helps confirm that your container is active and serving the webpage.

- **Accessing the Webpage:**

Providing the exact URL format for accessing the webpage**(e.g.,http://localhost:5000**)would be important because it guides users on how to open the browser and connect to the containerized service

```
ubuntu@ip-172-31-12-244:~/hello-cloud$ docker run -d -p 5000:5000 --name hello-cloud hello-cloud:1.0
ff7d58ffe3c4f5b115cdc0bd9d304f8658b66a8a79e1db8267c628c30d38f628
ubuntu@ip-172-31-12-244:~/hello-cloud$ docker ps
CONTAINER ID   IMAGE             COMMAND           CREATED          STATUS          PORTS                                       NAMES
ff7d58ffe3c4   hello-cloud:1.0   "python app.py"   10 seconds ago   Up 10 seconds   0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp   hello-cloud
ubuntu@ip-172-31-12-244:~/hello-cloud$ curl http://localhost:5000
Hello Cloud!ubuntu@ip-172-31-12-244:~/hello-cloud$
```

- Include the public URL/IP:port link to the web app:
  - To access the deployed web application, include the public URL or IP address with the port number in your documentation. For example: http://13.203.75.97.5000.
  - When visited, the app should display a confirmation message like **"Hello Cloud!"** indicating successful deployment.



Hello Cloud!