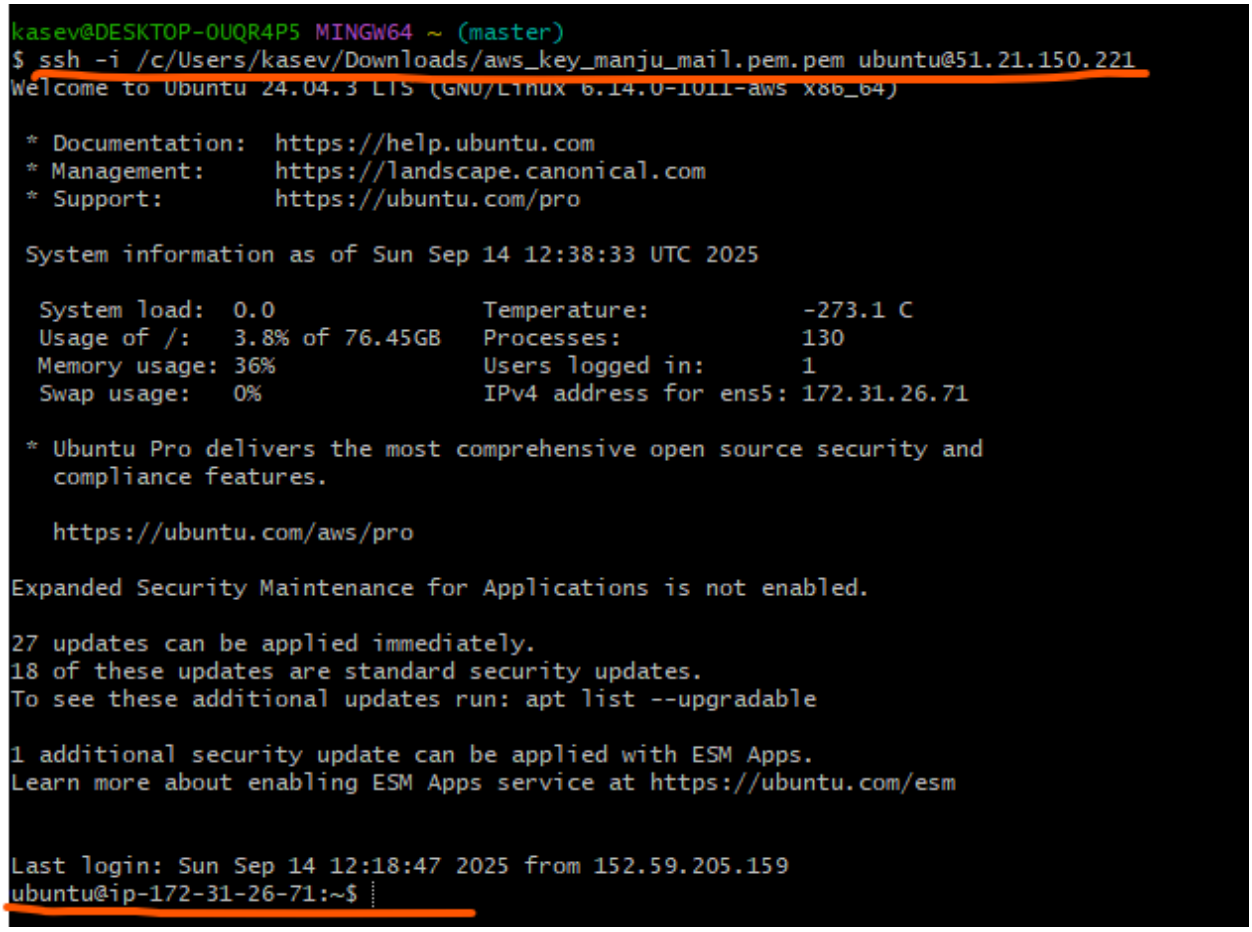
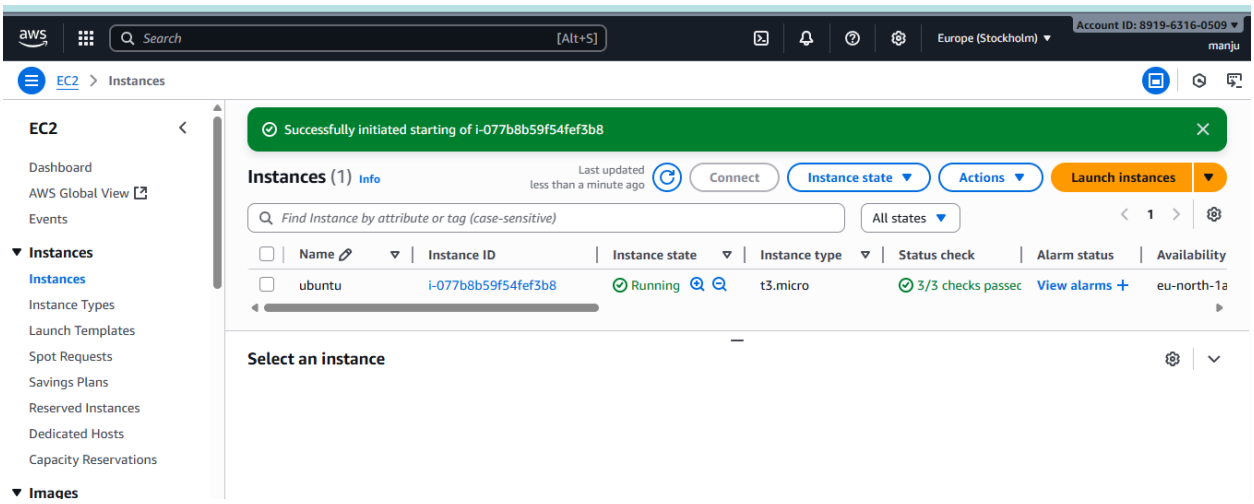
HOW TO SETUP DOCKER AND CREATE, BUILD, RUN and PUSH THE DOCKER IMAGES using UBUNTU OS

I have created AWS custom linux ubuntu OS LIKE T3 MICRO Connected to instance with .pem key > below is the command ssh -i /c/Users/kasev/Downloads/aws_key_manju_mail.pem.pem ubuntu@51.21.150.221





Configured and installed the Docker in ubuntu instance

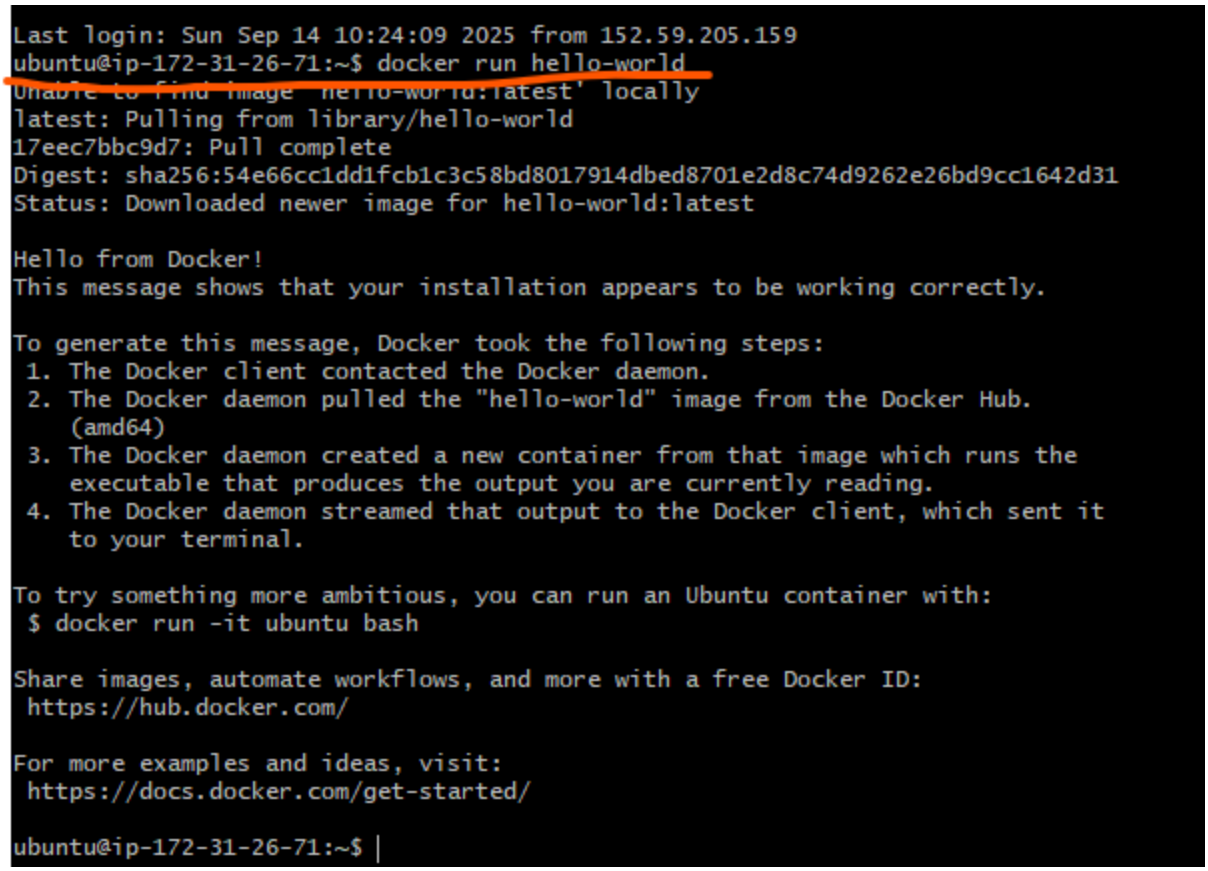
Follow below commands:

sudo apt update

sudo apt install docker.io -y

sudo systemctl status docker

```
Setting up bridge-utils (1.7.1-1ubuntu2) ...
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.27-0ubuntu1~24.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16+24.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (27.5.1-0ubuntu3~24.04.2) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group `docker' (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

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No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.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-172-31-26-71:~/docker$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-09-14 10:24:39 UTC; 52s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2455 (dockerd)
      Tasks: 9
     Memory: 32.0M (peak: 32.1M)
        CPU: 393ms
     CGroup: /system.slice/docker.service
             └─2455 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 14 10:24:38 ip-172-31-26-71 systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 14 10:24:38 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:38.662888242Z" level=info msg="Starting up"
Sep 14 10:24:38 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:38.663633489Z" level=info msg="OTEL tracing is not config
Sep 14 10:24:38 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:38.663755924Z" level=info msg="detected 127.0.0.53 nameser
Sep 14 10:24:38 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:38.856111485Z" level=info msg="Loading containers: start."
Sep 14 10:24:39 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:39.211858731Z" level=info msg="Loading containers: done."
Sep 14 10:24:39 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:39.233296982Z" level=info msg="Docker daemon" commit="27.
Sep 14 10:24:39 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:39.233422728Z" level=info msg="Daemon has completed initi
Sep 14 10:24:39 ip-172-31-26-71 dockerd[2455]: time="2025-09-14T10:24:39.256115024Z" level=info msg="API listen on /run/docker.
Sep 14 10:24:39 ip-172-31-26-71 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-22/22 (END)
```



docker run hello-world (if you get daemon error run below cmd to fix the the issue)

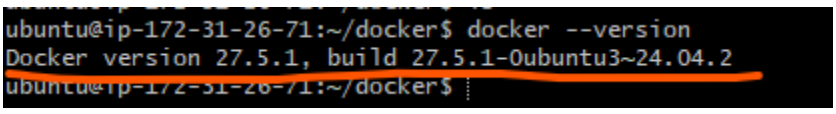
sudo usermod –aG docker ubuntu (this comd represent the adding the ubuntu user to docker group)

Next must logout from the instance and reconnect again and run below comd

docker run hello-world

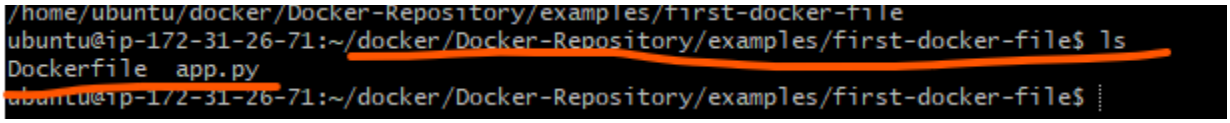
Now error would be resolved.

Check the docker version



Next how to write and execute the docker files:

1) I have cloned repository of
   git clone https://github.com/KASEVADDI/Docker-Repository.git
2) After cloned
   cd /home/ubuntu/docker/Docker-Repository/examples/first-docker-file
   Here Dockerfile i have created



Vim Dockerfile

```
FROM ubuntu:latest

 # Set the working directory in the image

WORKDIR /app

 # Copy the files from the host file system to the image file system

COPY . /app

 # Install the necessary packages

RUN apt-get update && apt-get install -y python3 python3-pip

 ## Set environment variables

 ENV NAME World

 # Run a command to start the application

CMD ["python3", "app.py"]

:wq!
```

vim app.py

print ("Hello World")

:wq!

Once create the dockerfile and app.py need to execute the dockerfile using below comd

```
docker build –it kmuthyal/my-first-docker-image:latest .
```

Must run above comd to build a docker image from dockerfile

Why must provide the tag becoz in our system or docker registory there would be many images so to remain the images we have to create image with tag.

Once execute the docker build comd the tasks would be executed in the sequence which contains in the Dockerfile

3) Run the container from the image:

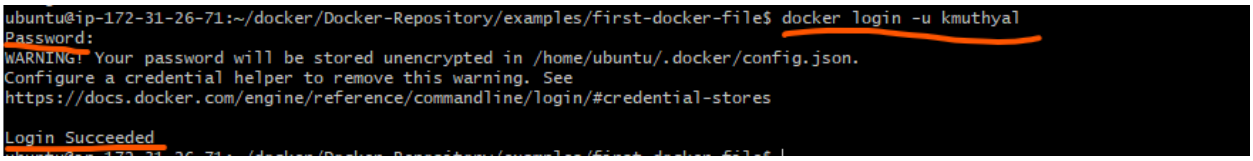
```
docker run –it kmuthyal/my-first-docker-image:latest .
```

docker images – to check the images

4) How push the docker image:

   The **docker push** command is used to **upload (push) your local Docker image to a remote registry** like **Docker Hub.**

To push docker image to docker hub must login to the docker registry through command line mode:



After login to the docker hub run below comd to push the image from **local Docker image to a remote registry** like **Docker Hub.**

```
docker push  kmuthyal/my-first-docker-image:latest
```

```
ubuntu@ip-172-31-26-71:~/docker/Docker-Repository/examples/first-docker-file$ docker push kmuthyal/my-first-docker-image:latest
The push refers to repository [docker.io/kmuthyal/my-first-docker-image]
15b685c4f3ab: Pushed
e91dd433de7b: Pushed
9dcc08c023c0: Pushed
9d592720ced4: Mounted from library/ubuntu
latest: digest: sha256:03805d843eeb4cb191c15b259c524124f18c9938d80308867c657e169267a359 size: 1155
ubuntu@ip-172-31-26-71:~/docker/Docker-Repository/examples/first-docker-file$
```