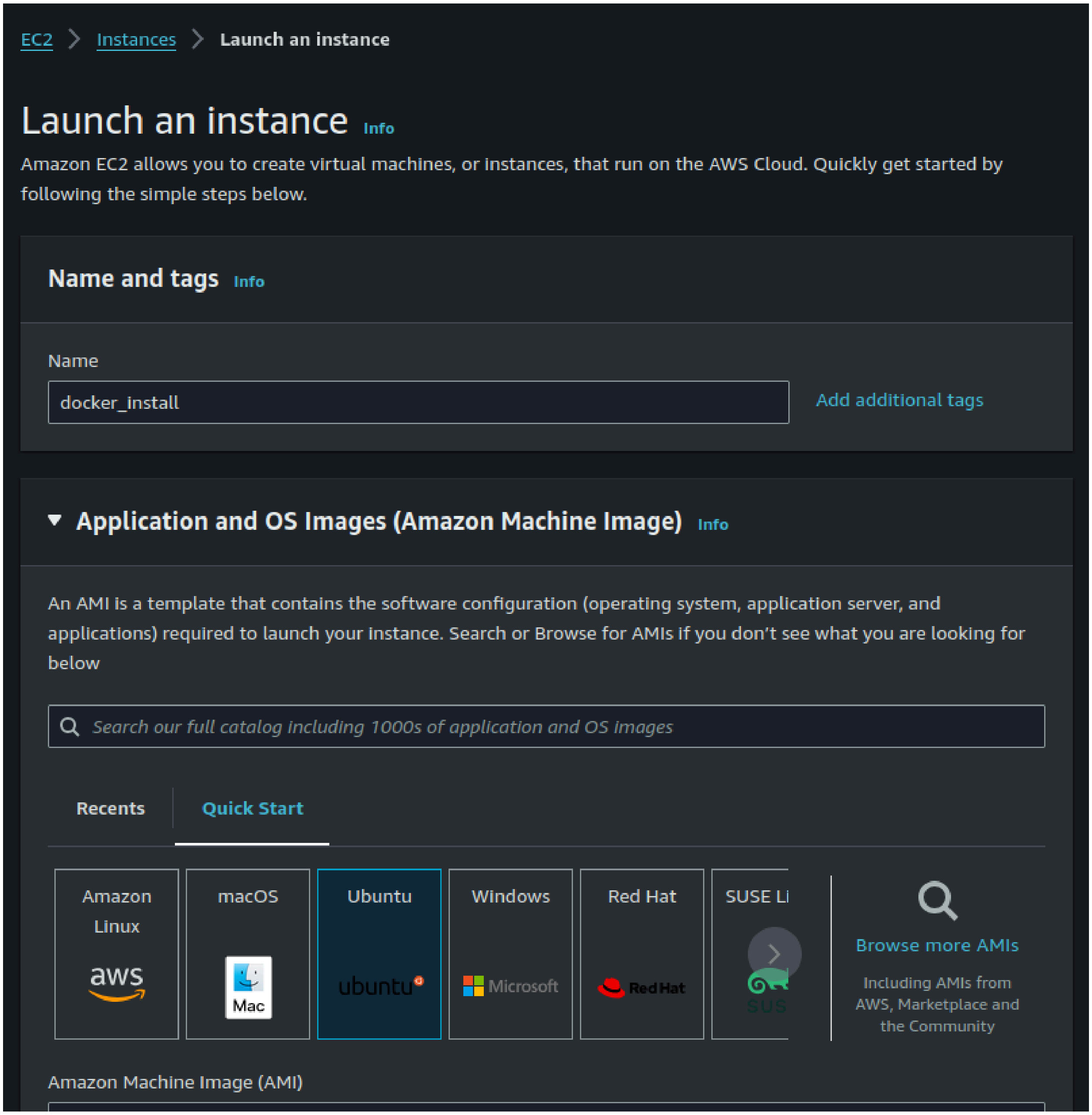
Task : Docker Essentials: Installation and Basic Commands on Ubuntu.

Step 1: Create an ec2 instance and launch it .



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: docker_install

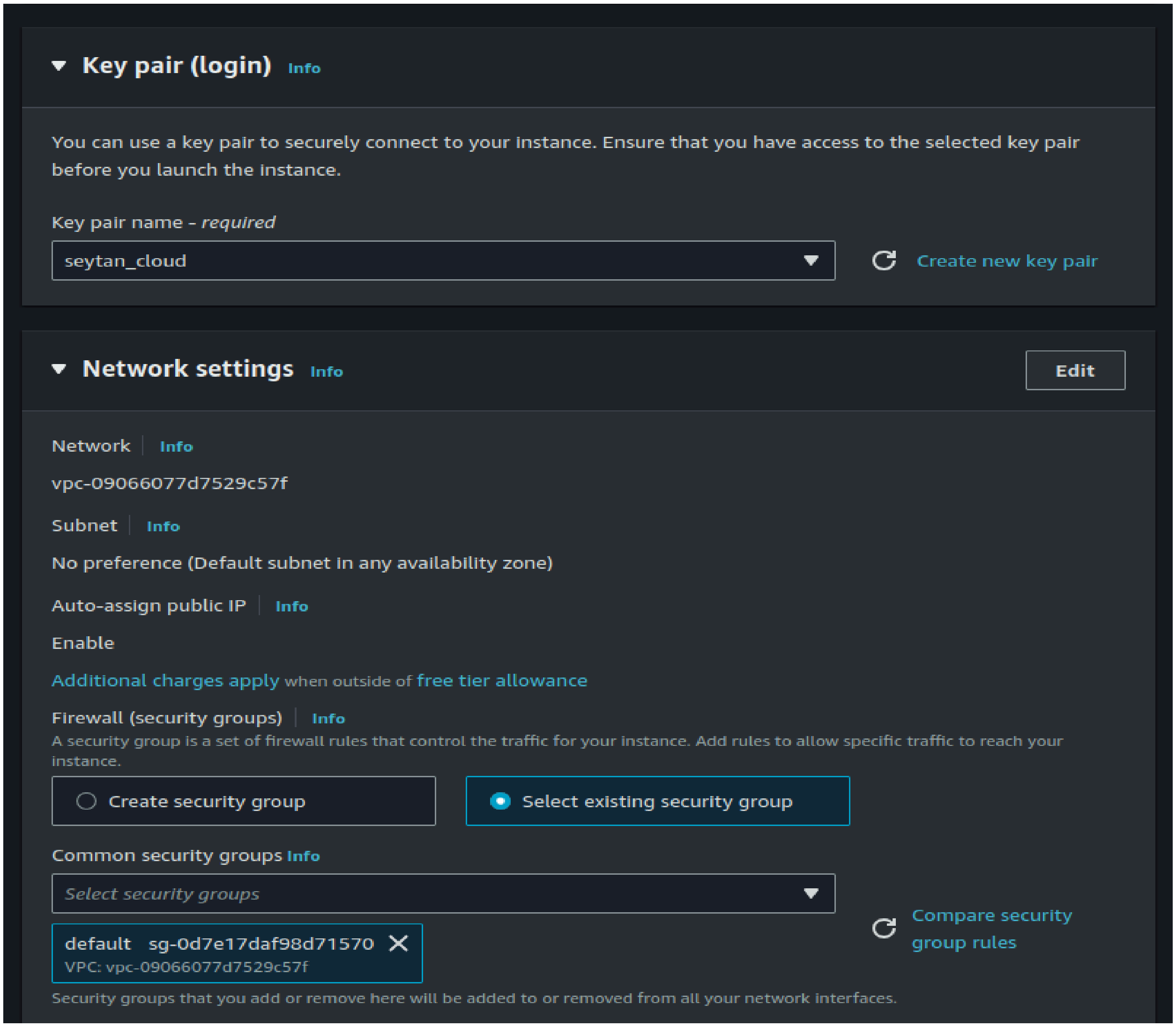
Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)



▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

seytan_cloud

▼ **Network settings** Info

Network | Info

vpc-09066077d7529c57f

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

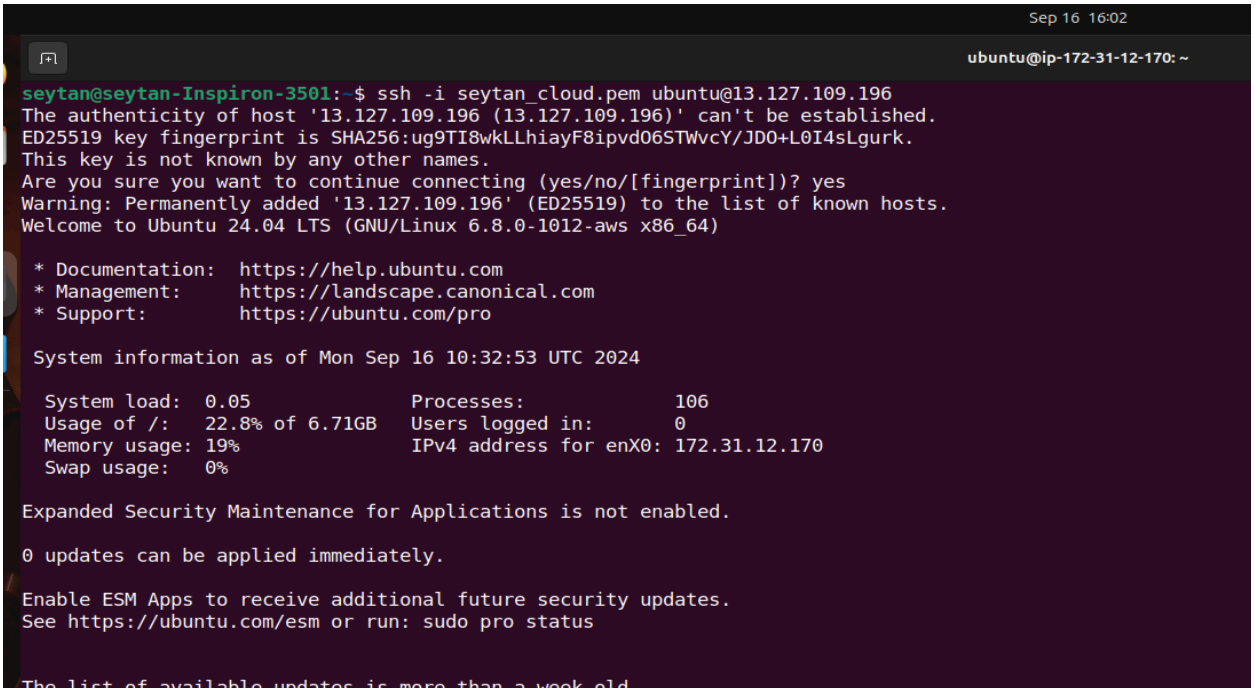
Common security groups Info

Select security groups

default sg-0d7e17daf98d71570 ✕
VPC: vpc-09066077d7529c57f

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Step 2: Take ssh of the instance.



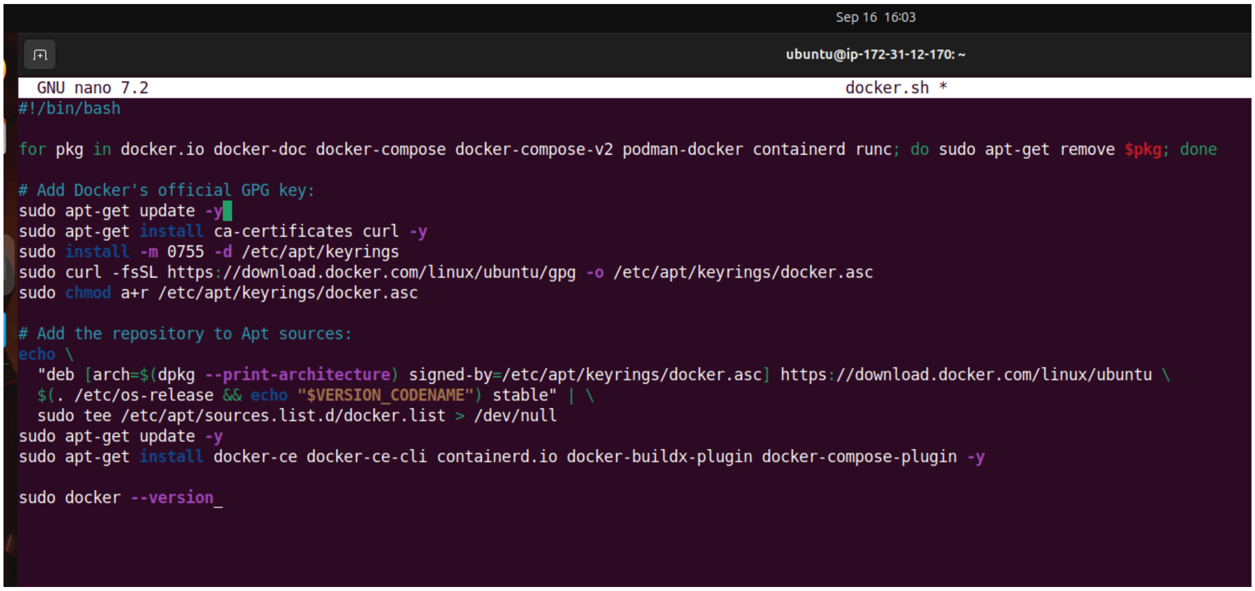
```
seytan@seytan-Inspiron-3501:~$ ssh -i seytan_cloud.pem ubuntu@13.127.109.196
The authenticity of host '13.127.109.196 (13.127.109.196)' can't be established.
ED25519 key fingerprint is SHA256:ug9TI8wkLLhiayF8ipvdO6STWvcY/JDO+L0I4sLgurk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.127.109.196' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Mon Sep 16 10:32:53 UTC 2024

  System load:  0.05              Processes:             106
  Usage of /:   22.8% of 6.71GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for enX0: 172.31.12.170
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
```

Step 3: Now you create installation script for docker or you can manually enter each command

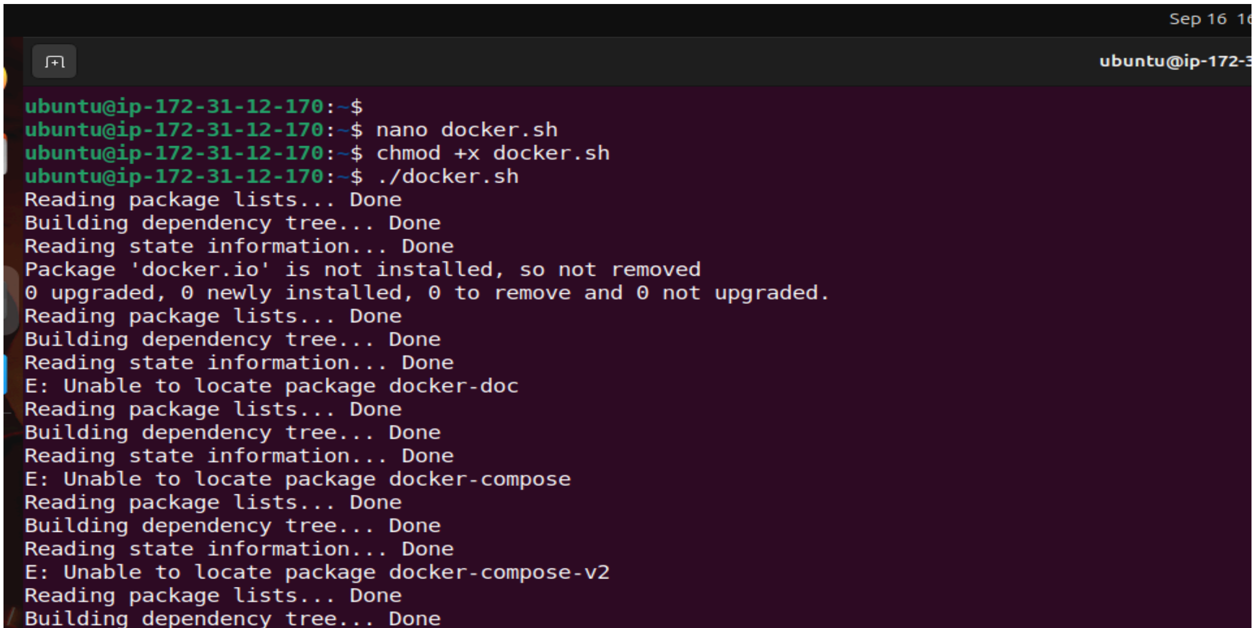
Reference: https://docs.docker.com/engine/install/ubuntu/

What is docker ?

→ Docker is an open-source platform that automates the deployment of applications in lightweight, portable containers. It enables developers to package code, dependencies, and configurations, ensuring consistent environments across different systems.



```
#!/bin/bash

for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get remove $pkg; done

# Add Docker's official GPG key:
sudo apt-get update -y
sudo apt-get install ca-certificates curl -y
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y

sudo docker --version
```



```
ubuntu@ip-172-31-12-170:~$
ubuntu@ip-172-31-12-170:~$ nano docker.sh
ubuntu@ip-172-31-12-170:~$ chmod +x docker.sh
ubuntu@ip-172-31-12-170:~$ ./docker.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker.io' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-doc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-compose-v2
Reading package lists... Done
Building dependency tree... Done
```

Step 4: Run some basic commands of docker .

You can run the following basic commands of docker to get some hands-on docker.

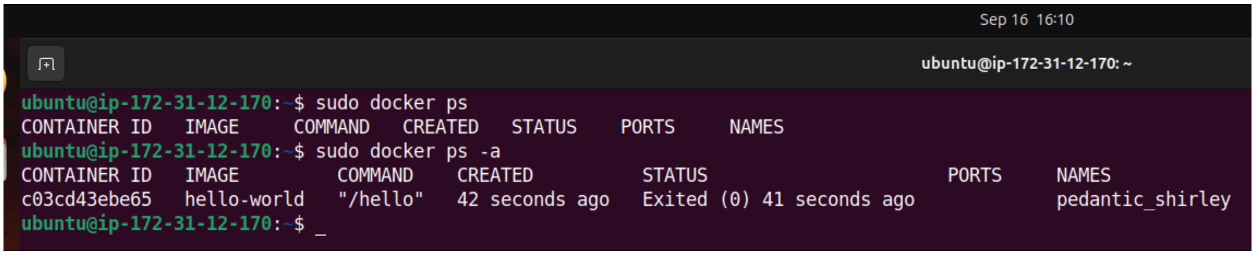
E.g
Docker Commands:

1. docker run <image name> - your container will be created.
2. docker run -d <image name> - runs your image in detached mode.
3. docker ps - running container.
4. docker ps -a - all containers (exited and running).
5. docker top <cont id> - check process inside the container.
6. docker inspect <container id> - detailed information of container.
7. docker run -d -p 8080:80 <image id> - assign specific port expose outside.
8. docker run -d -P <image name> - assign random ports.
9. netstat -tulpn - check ports.
10. docker ps -q - all the container IDs (running).
11. docker ps -qa - all containers IDs (exited or running).
12. docker run -d --name <container_name> <image_name> - gives name to your container(while container is running).
13. docker rename <cont id> <newname> - rename an existing container.
14. docker start <container id> - start the container.
15. docker stop <container id> - stop the container.
16. docker rm <container id> - remove container.

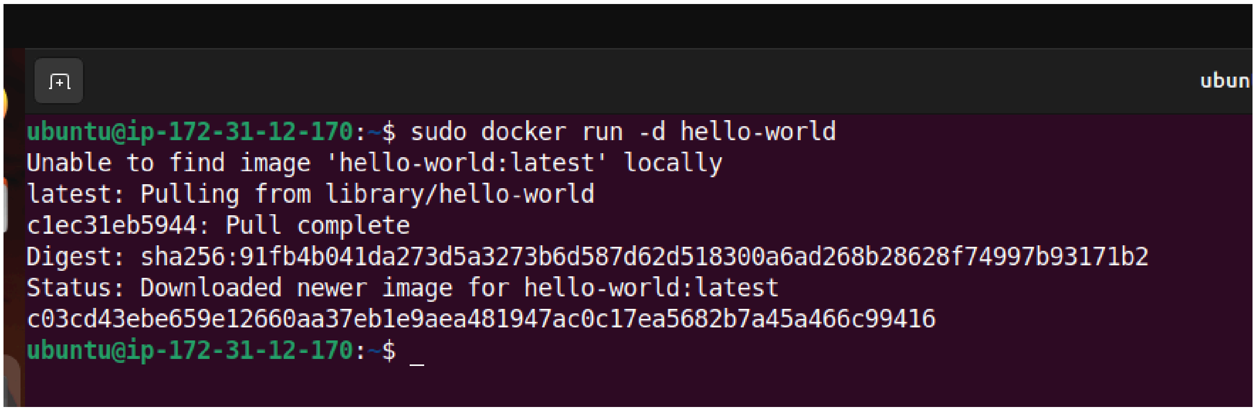
**17. docker exec -it <container id> bash - enter into existing container or run a command inside
it.
18. docker exec <container id> <command> - execute command inside container without going
inside.
19. docker run -it <image name> command (bash or sh) - enter inside container or execute any
command in container.
20. docker run <image name> <command> - execute command in running container.
21. docker stats <container_id> - display container stats.
22. docker run --name <container_name> -d -P <image_name> - set name for container.
23. docker container status - check the status of containers.
24. docker cp index.html <cont id>:/usr/share/nginx/html/index.html - copy file to the container.**

**Working with Container Images:
25. docker images - list all images in host machine.
26. docker images ls - same as docker images (list).
27. docker pull <image name> - pull docker image.
28. docker image rm <image name> - remove docker image by name.
29. docker rmi <image id> - remove docker image by ID.
30. docker prune - remove unused containers and images.
31. docker tag <container id> newname
- tag container with new name.**

Step 5: Running some commands and its output.



```
ubuntu@ip-172-31-12-170:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-12-170:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                      PORTS     NAMES
c03cd43ebe65   hello-world   "/hello"   42 seconds ago   Exited (0) 41 seconds ago             pedantic_shirley
ubuntu@ip-172-31-12-170:~$ _
```



```
ubuntu@ip-172-31-12-170:~$ sudo docker run -d hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:91fb4b041da273d5a3273b6d587d62d518300a6ad268b28628f74997b93171b2
Status: Downloaded newer image for hello-world:latest
c03cd43ebe659e12660aa37eb1e9aea481947ac0c17ea5682b7a45a466c99416
ubuntu@ip-172-31-12-170:~$ _
```

Conclusion:

Docker simplifies containerization, allowing applications to run in isolated environments. Installing Docker on Ubuntu and mastering basic commands is essential for efficient development and deployment.