1. Diiference between virtual machine and docker containers.

DOCKER:

- **Docker is a container-based model where containers are software packages used for executing an application on any operating system.**
- **In Docker, the containers share the host OS kernel.**
- **Multiple workloads can run on a single OS.**
- **Docker containers result in high-performance as they use the same operating system with no additional software (like hypervisor)**
- **With docker containers, users can create an application and store it into a container image. Then, he/she can run it across any host environment.**
- **Docker container is smaller than VMs, because of which the process of transferring files on the host's filesystem is easier The application in Docker containers starts with no delay since the OS is already up and running.**
- **These containers were basically designed to save time in the deployment process of an application.**

VM MACHINE:

- **A virtual machine (VM) is a computing environment or software that aids developers to access an operating system via a physical machine.**
- **there is a layer of Hypervisor which sits between physical hardware and operating systems. In a broader view, Hypervisor is used to virtualize the hardware which is then configured with the way a user wants it to.**
- **Virtual Machines our physical machine is divided into following parts :-**
- **Example:**
  Using a system having 8GB RAM. If you create 2 VMs each with 4GB RAM, you are basically dividing your server into two components - each with 4GB RAM and would never be able to use that underlying 8GB RAM altogether again.
- **They use user space along with the kernel space of an OS .**
- **Each workload needs a complete OS or hypervisor.**

- **Since VM uses a separate OS; it causes more resources to be used**
  It has known portability issues.
- **VMs don't have a central hub and it requires more memory space to store data.**
- **While transferring files, VMs should have a copy of the OS and its dependencies because of which image size is increased and becomes a tedious process to share data.**
- **It takes a much longer time than it takes for a container to run applications**
- **To deploy a single application, Virtual Machines need to start the entire OS, which would cause a full boot process.**

DOCKER	VM MACHINE
It boots in a few seconds.	It takes a few minutes for VMs to boot.
Pre-built docker containers are readily available.	Ready-made VMs are challenging to find.
Docker has a complex usage mechanism consisting of both third-party and docker-managed tools.	Tools are easy to use and more straightforward to work with. third-party.
Limited to Linux.	Can run a variety of guest OS.
Dockers make use of the execution engine.	VMs make use of the hypervisor.
It is lightweight.	It is heavyweight.

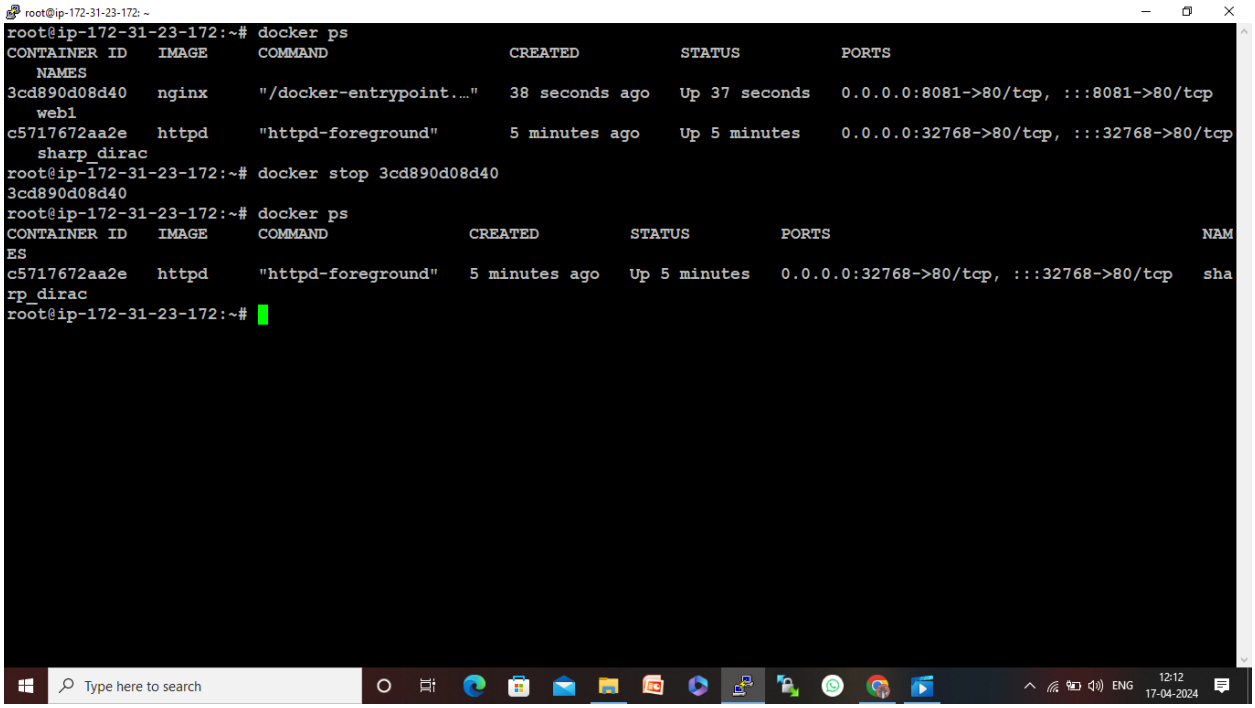
Host OS can be different from container OS.	**Host OS can be different from guest OS.**
Can run many docker containers on a laptop.	**Cannot run more than a couple of VMS on an average laptop.**
Docker can get a virtual network adapter. It can have separate IPs ad Ports.	**Each VMS gets its virtual network adapter.**
Sharing of files is possible.	**Sharing library and files are not possible.**
Lacks security measures.	**Security depends on the hypervisor.**
A container is portable.	**VMS is dependent on a hypervisor.**
Only one VM can be started from one set of VMX and VMDX files.	**Multiple docker containers can be started from one docker image.**

2.DOCKER STOP AND DOCKER PAUSE:

Docker stop: it sends a signal to the container to stop all processes gracefully and then shuts down the container.

When you stop a container, its state is not preserved, and any changes made during its runtime may be lost.

A stopped container cannot be resumed from its previous state. It needs to be started again, which would launch a new instance of the container.



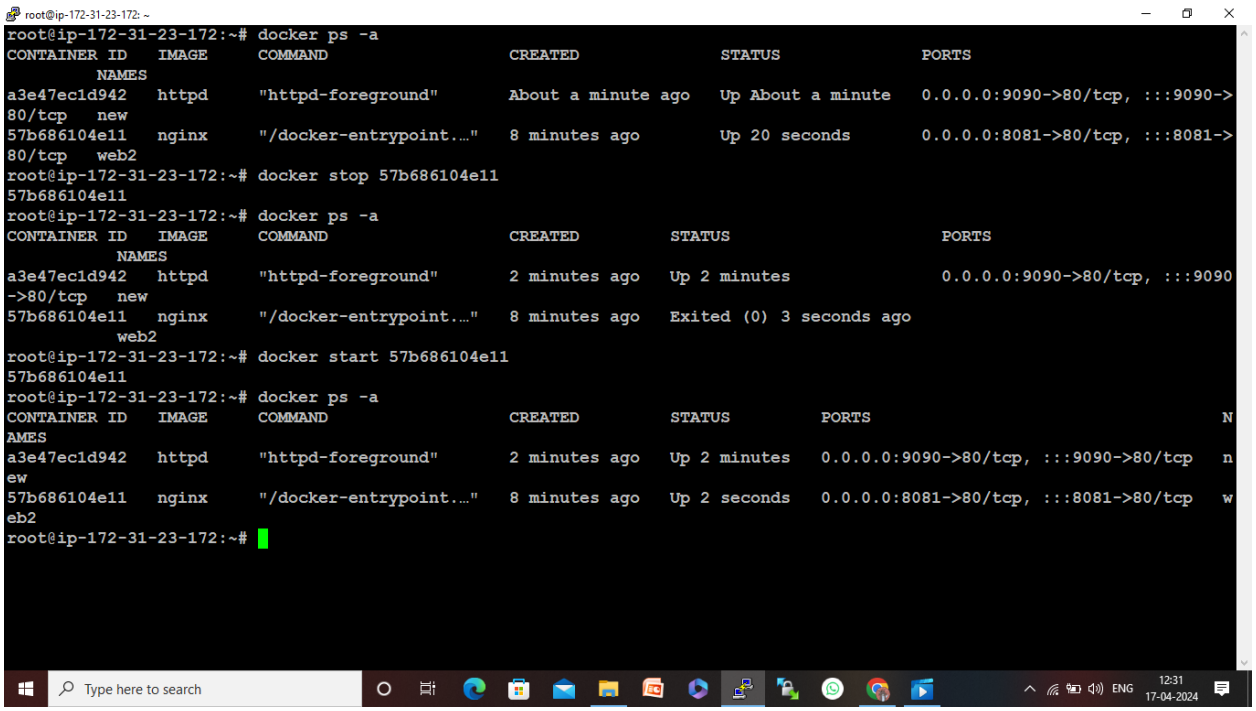
Can also start stopped containers in Docker.

When you stop a container using the docker stop command, it does not delete the container.

it simply stops its execution.

You can start a stopped container using “docker stasrt container_id”

“docker ps –a” can list the all state of containers (running, stopped, paused).



Docker pause: When you use docker pause, it suspends all processes within the container.

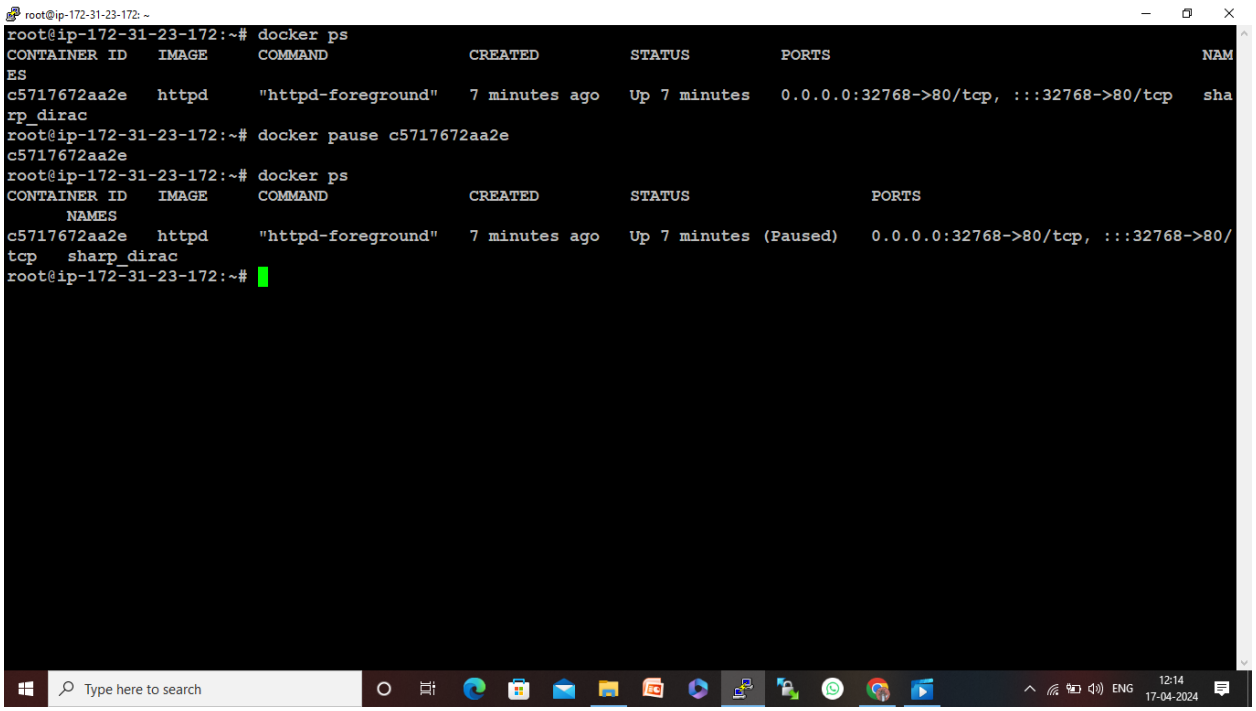
The container remains in a paused state, but its state and memory are preserved.

Pausing a container essentially freezes it in its current state, meaning that it stops all processes within the container from running.

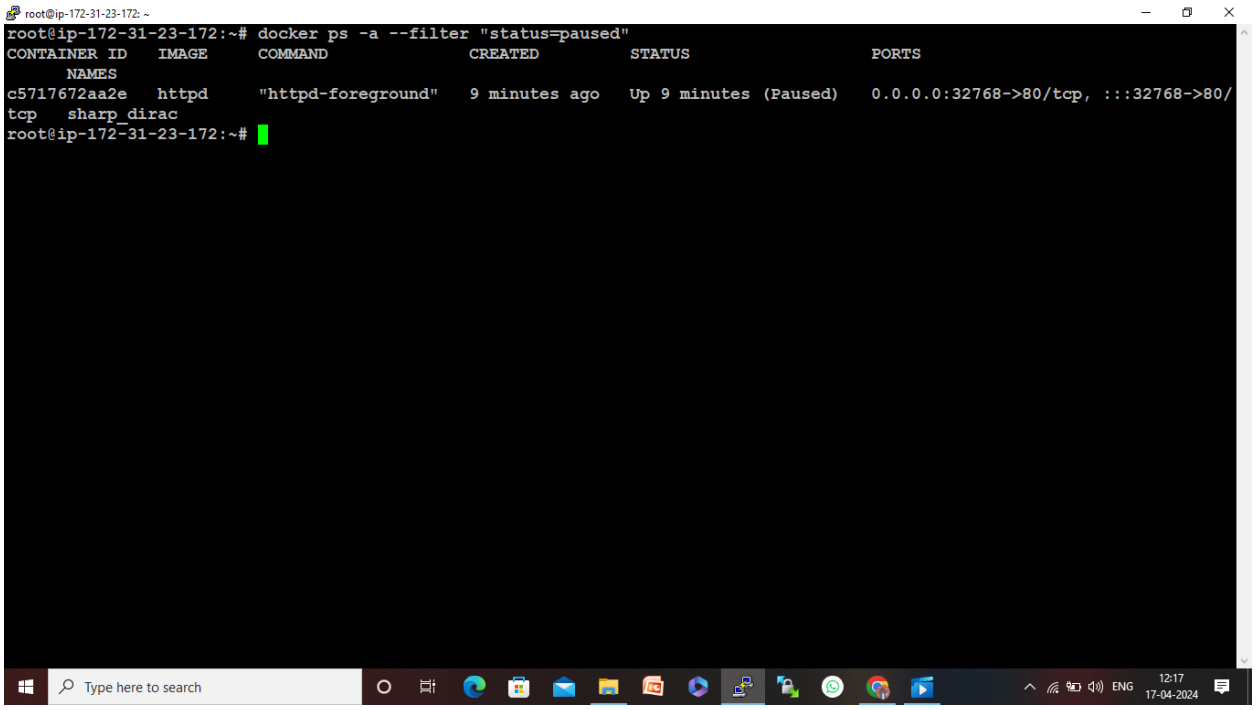
However, the container's file system and network connections remain intact, and it can be resumed later to continue execution from where it was paused.

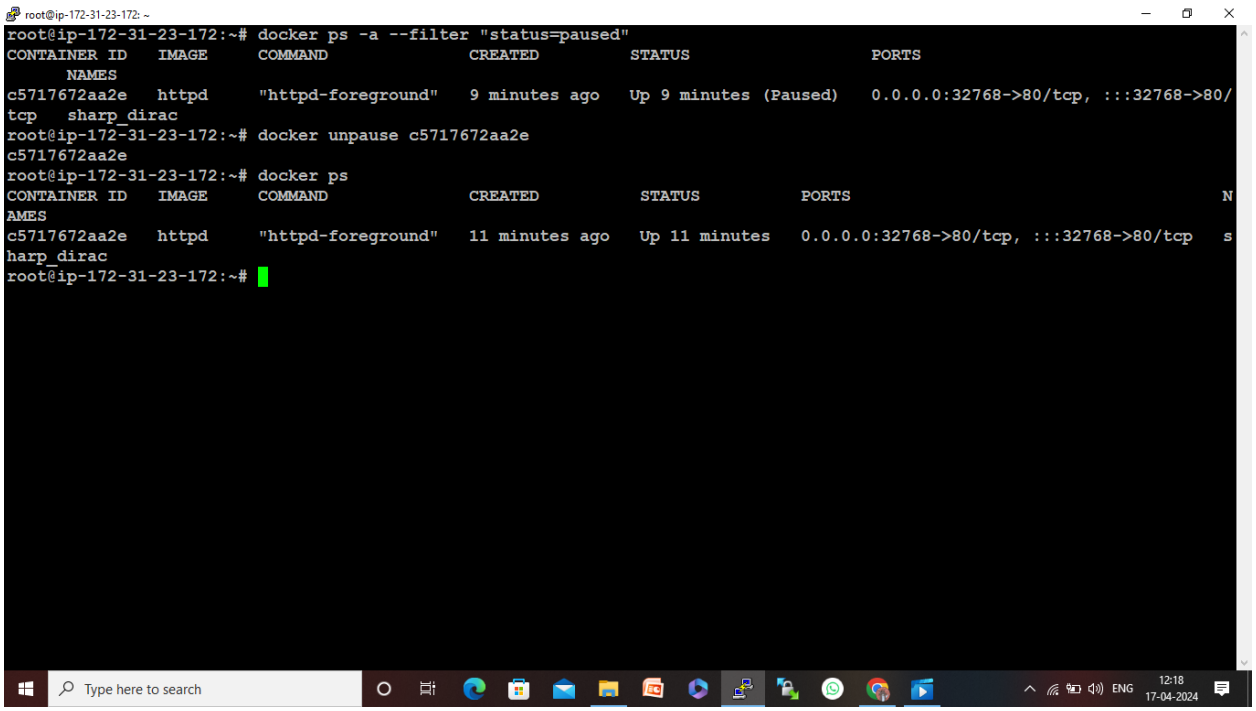
We can also start a paused container in Docker,

you can use the “docker unpause” command followed by the container's ID



docker ps -a --filter "status=paused" : This command will display a list of all paused containers along with their IDs, names, and other details.

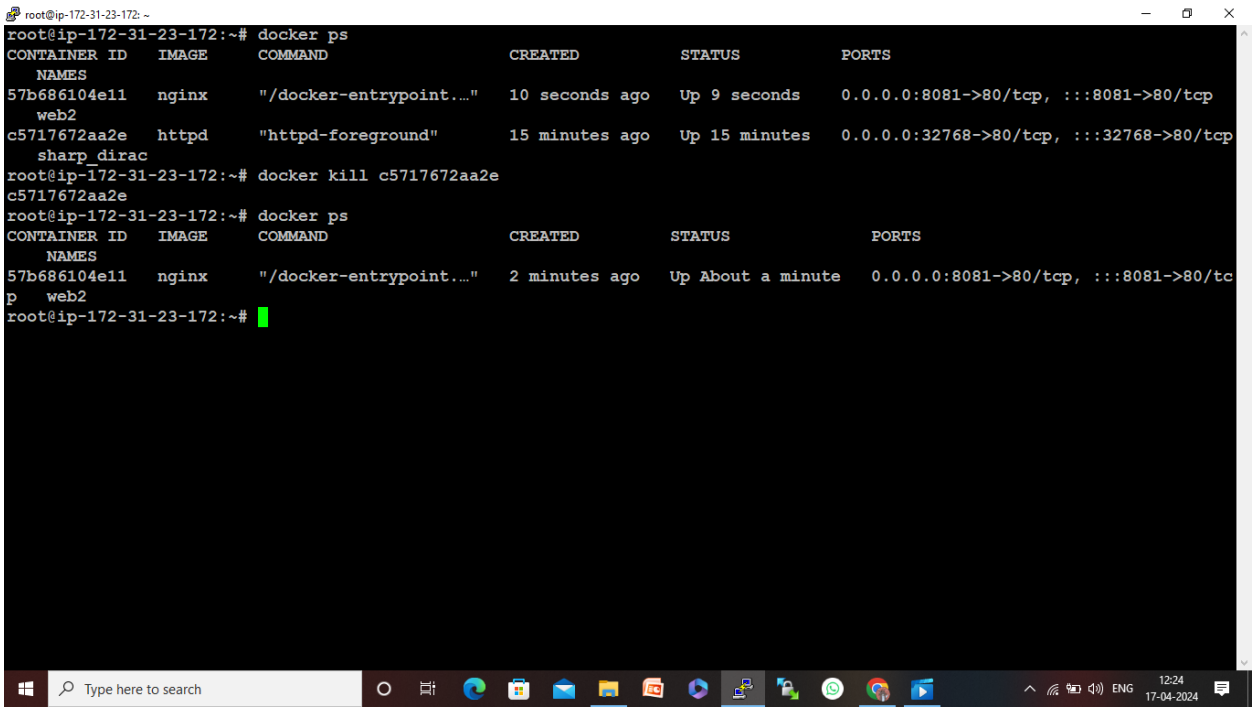




3. DOCKER KILL AND DCOKER DELETE(REMOVE):

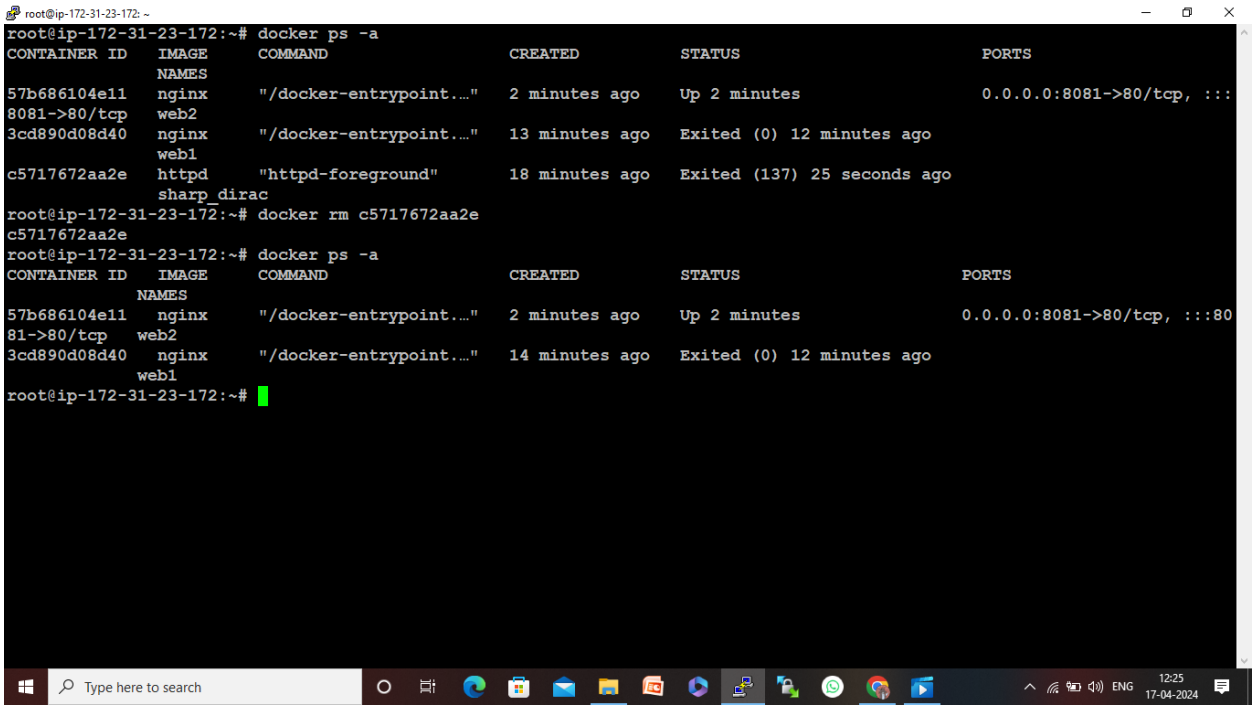
Docker kill : The docker kill command is used to send a signal to a running container to stop it forcefully.

Unlike docker stop, which sends a graceful termination signal, docker kill sends a signal that immediately terminates the container without giving it a chance to shut down gracefully.

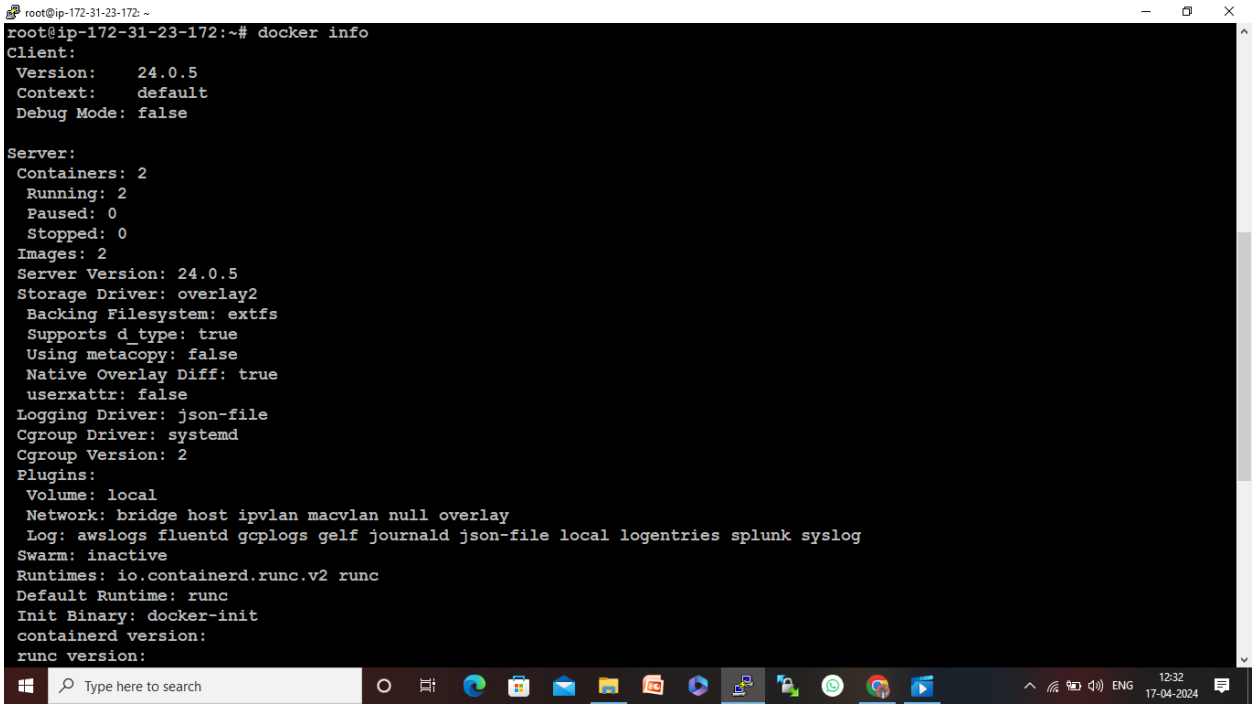


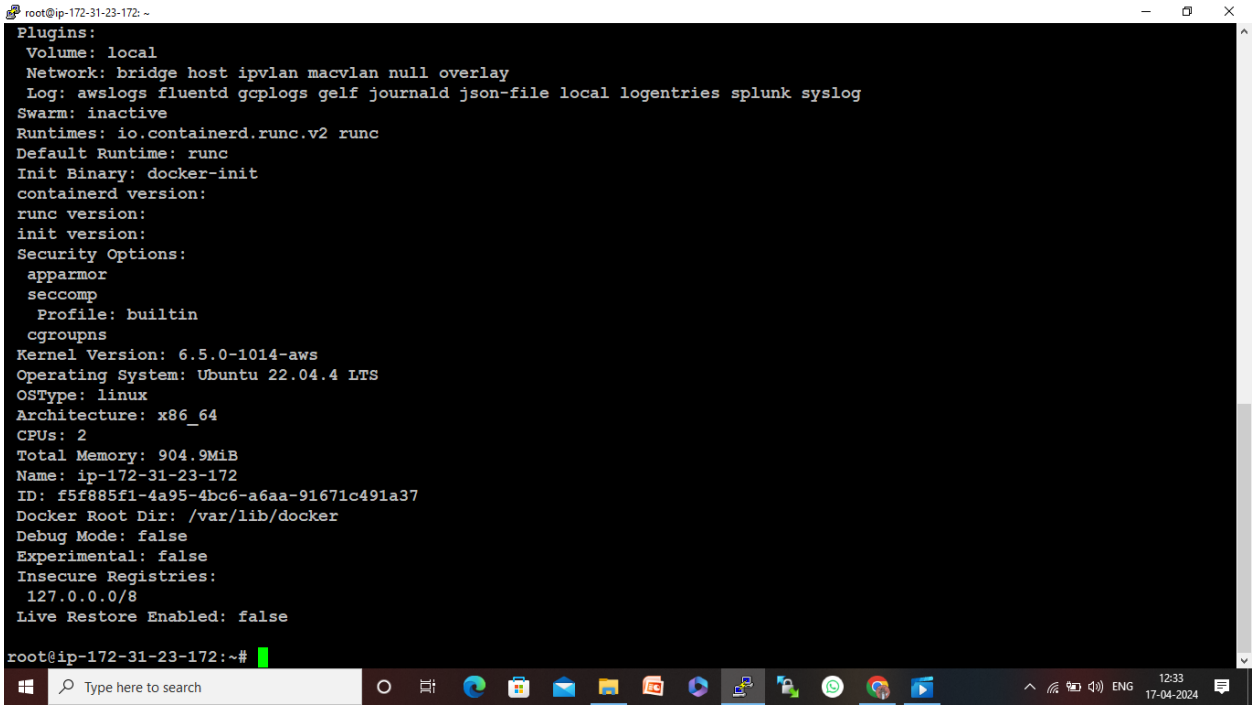
Docker rm : The docker rm command is used to delete one or more stopped containers from the system.

This command permanently removes containers from the Docker host.



4. Docker info





The docker info command provides detailed information about the Docker system.

Docker version: The version of Docker Engine installed on your system.

Containers: The number of containers currently running and the total number of containers.

Images: The number of images downloaded or built and the total number of images.

Storage driver: The storage driver used by Docker to manage container and image data.

Operating system: Information about the underlying operating system running Docker.

Kernel version: The version of the Linux kernel running on the host system (if applicable).

API version: The version of the Docker Remote API supported by the Docker Engine.