Lab Exercise 1- Docker Fundamental Commands

Objective

Learn and practice the fundamental Docker CLI commands to:

- Manage images and containers
- Understand container lifecycle
- Build and run applications using Docker

---

1. Setup

Prerequisites

- Docker installed on your system

  (Check using: docker --version)
- Internet access to pull images

---

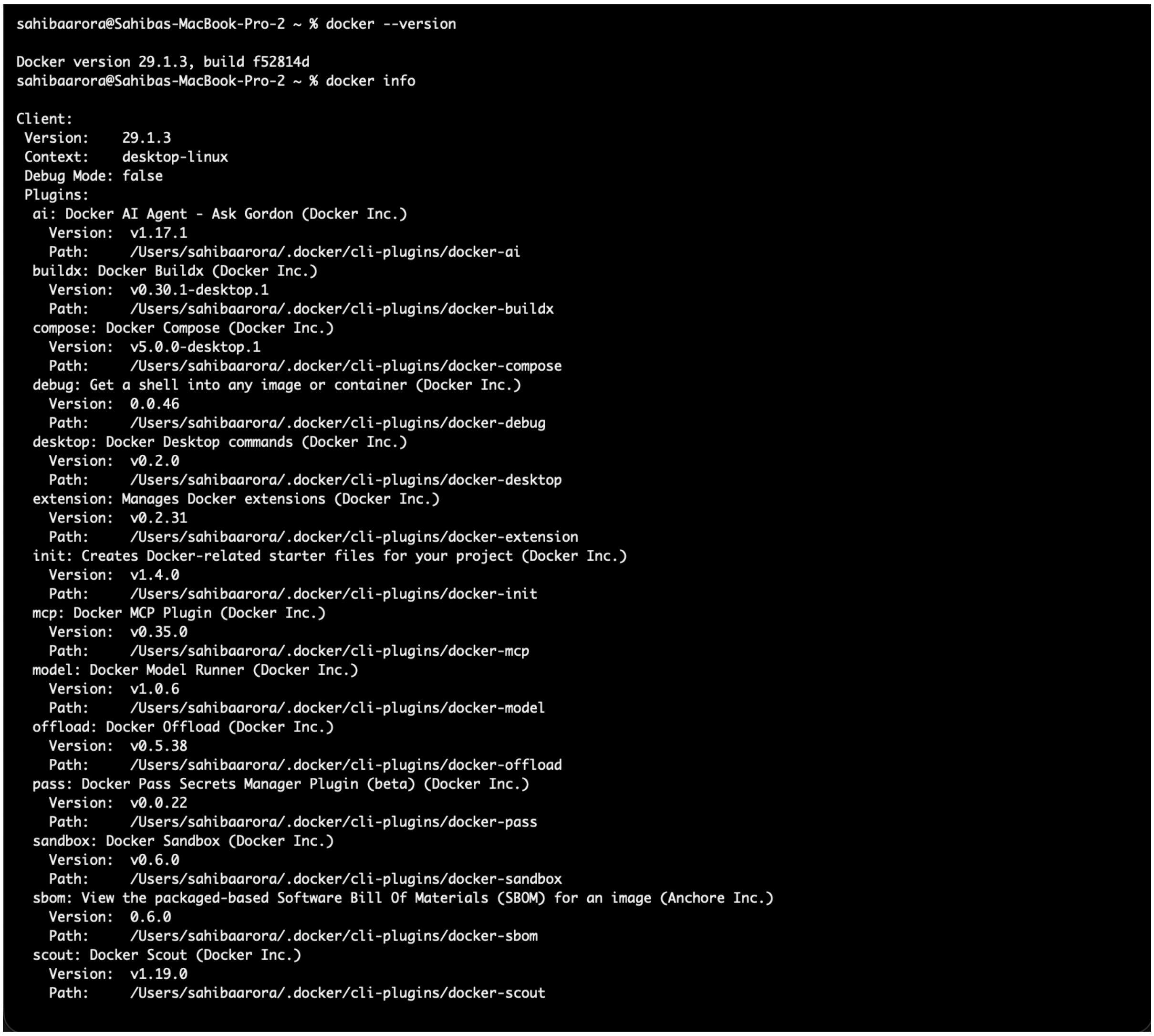
2. Basic Docker Commands

Step 1: Verify Installation

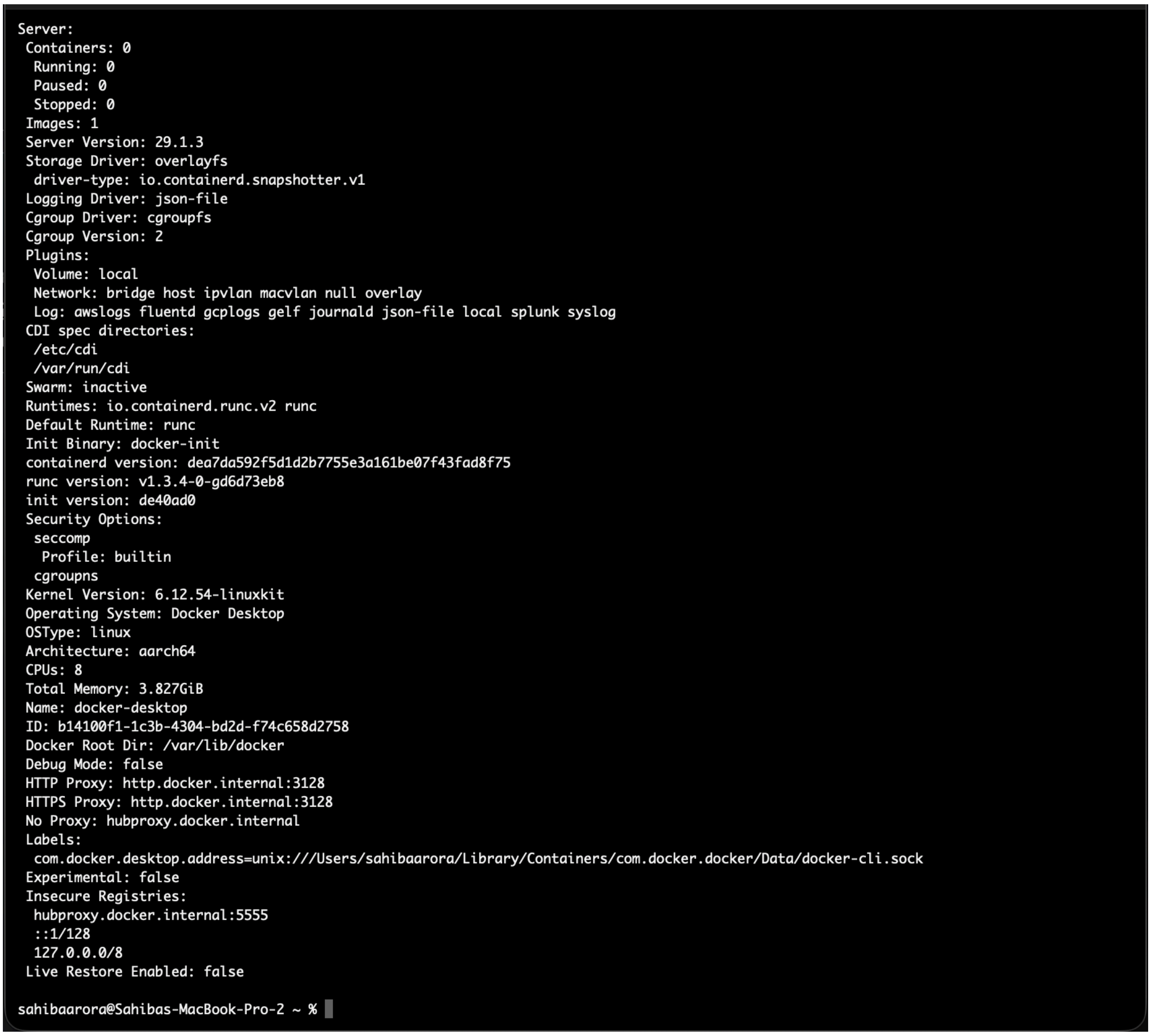
```
docker --version

docker info
```

Expected Output: Docker version and system details.

OUTPUT :-



```
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker --version

Docker version 29.1.3, build f52814d
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker info

Client:
 Version:    29.1.3
 Context:    desktop-linux
 Debug Mode: false
 Plugins:
  ai: Docker AI Agent - Ask Gordon (Docker Inc.)
    Version:  v1.17.1
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-ai
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.30.1-desktop.1
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version:  v5.0.0-desktop.1
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-compose
  debug: Get a shell into any image or container (Docker Inc.)
    Version:  0.0.46
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-debug
  desktop: Docker Desktop commands (Docker Inc.)
    Version:  v0.2.0
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-desktop
  extension: Manages Docker extensions (Docker Inc.)
    Version:  v0.2.31
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-extension
  init: Creates Docker-related starter files for your project (Docker Inc.)
    Version:  v1.4.0
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-init
  mcp: Docker MCP Plugin (Docker Inc.)
    Version:  v0.35.0
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-mcp
  model: Docker Model Runner (Docker Inc.)
    Version:  v1.0.6
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-model
  offload: Docker Offload (Docker Inc.)
    Version:  v0.5.38
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-offload
  pass: Docker Pass Secrets Manager Plugin (beta) (Docker Inc.)
    Version:  v0.0.22
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-pass
  sandbox: Docker Sandbox (Docker Inc.)
    Version:  v0.6.0
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-sandbox
  sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc.)
    Version:  0.6.0
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-sbom
  scout: Docker Scout (Docker Inc.)
    Version:  v1.19.0
    Path:     /Users/sahibaarora/.docker/cli-plugins/docker-scout
```



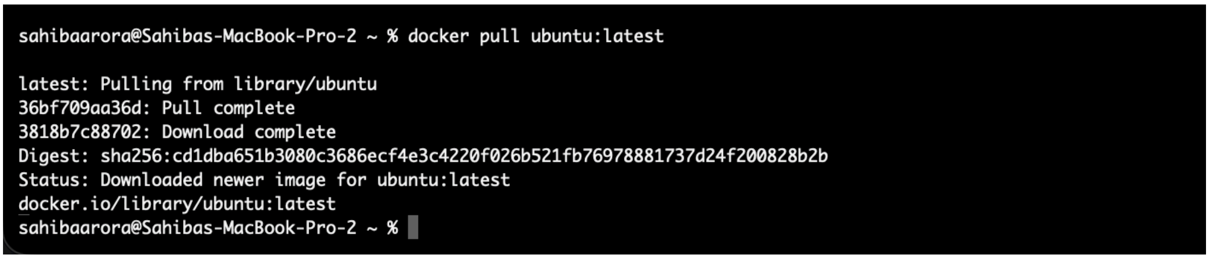
```
Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 1
 Server Version: 29.1.3
 Storage Driver: overlayfs
  driver-type: io.containerd.snapshotter.v1
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 2
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
 CDI spec directories:
  /etc/cdi
  /var/run/cdi
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: dea7da592f5d1d2b7755e3a161be07f43fad8f75
 runc version: v1.3.4-0-gd6d73eb8
 init version: de40ad0
 Security Options:
  seccomp
   Profile: builtin
  cgroupns
 Kernel Version: 6.12.54-linuxkit
 Operating System: Docker Desktop
 OSType: linux
 Architecture: aarch64
 CPUs: 8
 Total Memory: 3.827GiB
 Name: docker-desktop
 ID: b14100f1-1c3b-4304-bd2d-f74c658d2758
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 HTTP Proxy: http.docker.internal:3128
 HTTPS Proxy: http.docker.internal:3128
 No Proxy: hubproxy.docker.internal
 Labels:
  com.docker.desktop.address=unix:///Users/sahibaarora/Library/Containers/com.docker.docker/Data/docker-cli.sock
 Experimental: false
 Insecure Registries:
  hubproxy.docker.internal:5555
  ::1/128
  127.0.0.0/8
 Live Restore Enabled: false

sahibaarora@Sahibas-MacBook-Pro-2 ~ %
```

Step 2: Pull an Image from Docker Hub

```
docker pull ubuntu:latest
```

Explanation: Downloads the latest Ubuntu image from Docker Hub.

OUTPUT :-

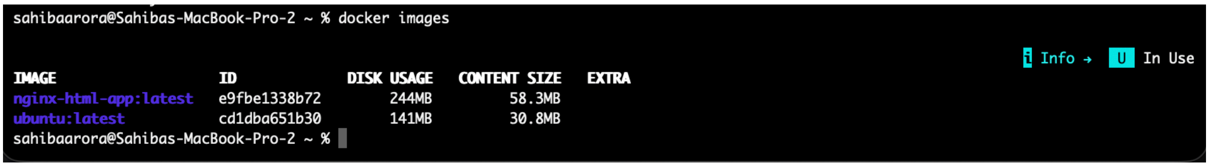


```
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker pull ubuntu:latest

latest: Pulling from library/ubuntu
36bf709aa36d: Pull complete
3818b7c88702: Download complete
Digest: sha256:cd1dba651b3080c3686ecf4e3c4220f026b521fb76978881737d24f200828b2b
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
sahibaarora@Sahibas-MacBook-Pro-2 ~ %
```

Check available images:

```
docker images
```

OUTPUT :-



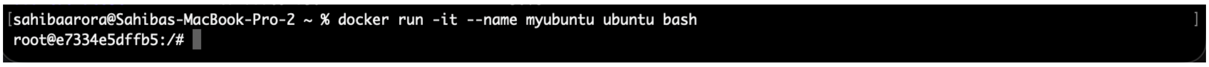
```
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker images

                                                                        i Info →  U In Use
IMAGE                  ID             DISK USAGE   CONTENT SIZE   EXTRA
nginx-html-app:latest  e9fbe1338b72        244MB         58.3MB
ubuntu:latest          cd1dba651b30        141MB         30.8MB
sahibaarora@Sahibas-MacBook-Pro-2 ~ %
```

Step 3: Run a Container

Run Ubuntu interactively:

```
docker run -it --name myubuntu ubuntu bash
```

OUTPUT :-



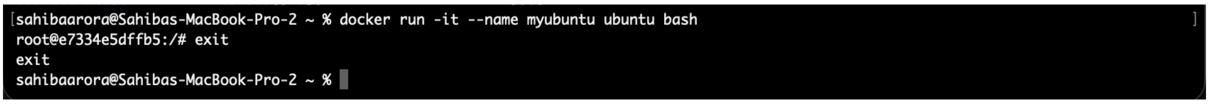
```
[sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker run -it --name myubuntu ubuntu bash        ]
root@e7334e5dffb5:/#
```

Now you’re inside a running container (prompt will change).

Exit:

```
exit
```

OUTPUT :-



```
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker run -it --name myubuntu ubuntu bash
root@e7334e5dffb5:/# exit
exit
sahibaarora@Sahibas-MacBook-Pro-2 ~ %
```

Step 4: List Containers

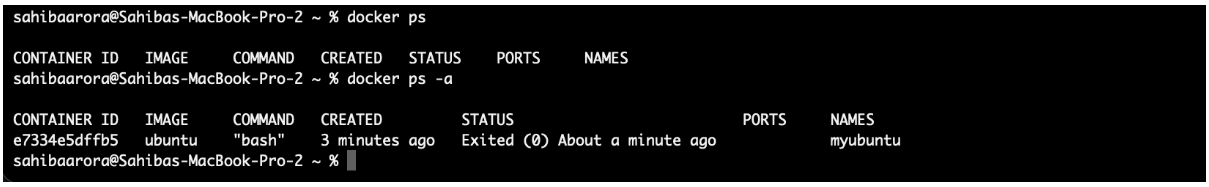
- Show running containers:

```
docker ps
```

- Show all containers (including stopped):

```
docker ps -a
```

OUTPUT :-



```
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker ps

CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker ps -a

CONTAINER ID   IMAGE     COMMAND   CREATED         STATUS                          PORTS     NAMES
e7334e5dffb5   ubuntu    "bash"    3 minutes ago   Exited (0) About a minute ago             myubuntu
sahibaarora@Sahibas-MacBook-Pro-2 ~ %
```

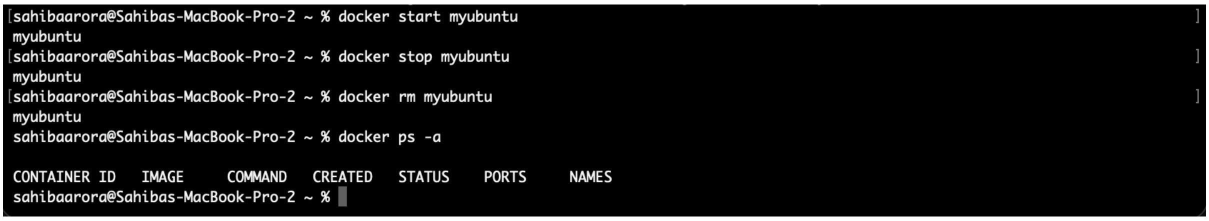
Step 5: Start / Stop / Remove Containers

```
# Start a stopped container

docker start <container_id>

# Stop a running container

docker stop <container_id>

# Remove a container
```

```
docker rm <container_id>
```

Tip: Use docker ps -a to get container IDs.

OUTPUT :-



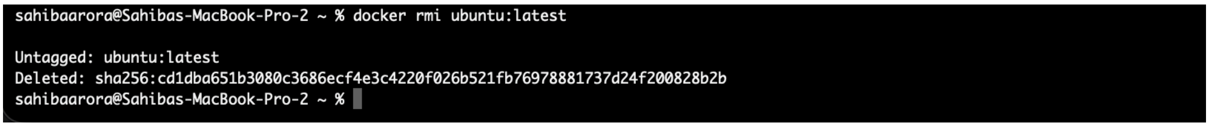
```
[sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker start myubuntu
 myubuntu
[sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker stop myubuntu
 myubuntu
[sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker rm myubuntu
 myubuntu
 sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker ps -a

 CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
 sahibaarora@Sahibas-MacBook-Pro-2 ~ %
```

Step 6: Remove Images

```
docker rmi ubuntu:latest
```

Note: You must stop and remove all containers using that image first.

OUTPUT :-



```
sahibaarora@Sahibas-MacBook-Pro-2 ~ % docker rmi ubuntu:latest

Untagged: ubuntu:latest
Deleted: sha256:cd1dba651b3080c3686ecf4e3c4220f026b521fb76978881737d24f200828b2b
sahibaarora@Sahibas-MacBook-Pro-2 ~ %
```
