

# 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
```

*Output:*

```
PS C:\Users\rhyth> docker --version
Docker version 28.3.2, build 578ccf6
PS C:\Users\rhyth> docker info
Client:
  Version: 28.3.2
  Context: desktop-linux
  Debug Mode: false
  Plugins:
    ai: Docker AI Agent - Ask Gordon (Docker Inc.)
      Version: v1.6.0
      Path: C:\Program Files\Docker\cli-plugins\docker-ai.exe
    buildx: Docker Buildx (Docker Inc.)
      Version: v0.25.0-desktop.1
      Path: C:\Program Files\Docker\cli-plugins\docker-buildx.exe
    cloud: Docker Cloud (Docker Inc.)
      Version: v0.4.2
      Path: C:\Program Files\Docker\cli-plugins\docker-cloud.exe
    compose: Docker Compose (Docker Inc.)
      Version: v2.38.2-desktop.1
      Path: C:\Program Files\Docker\cli-plugins\docker-compose.exe
    debug: Get A shell into any image or container (Docker Inc.)
      Version: 0.0.41
      Path: C:\Program Files\Docker\cli-plugins\docker-debug.exe
    desktop: Docker Desktop commands (Docker Inc.)
      Version: v0.1.11
      Path: C:\Program Files\Docker\cli-plugins\docker-desktop.exe
    extension: Manages Docker extensions (Docker Inc.)
      Version: v0.2.29
      Path: C:\Program Files\Docker\cli-plugins\docker-extension.exe
    init: Creates Docker-related starter files for your project (Docker Inc.)
      Version: v1.4.0
      Path: C:\Program Files\Docker\cli-plugins\docker-init.exe
    mcp: Docker MCP Plugin (Docker Inc.)
      Version: v0.9.9
      Path: C:\Program Files\Docker\cli-plugins\docker-mcp.exe
    model: Docker Model Runner (EXPERIMENTAL) (Docker Inc.)
      Version: v0.0.33
      Path: C:\Program Files\Docker\cli-plugins\docker-model.exe
    sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc.)
      Version: 0.6.0
      Path: C:\Program Files\Docker\cli-plugins\docker-sbom.exe
  scout: Docker Scout (Docker Inc.)
      Version: v1.18.1
      Path: C:\Program Files\Docker\cli-plugins\docker-scout.exe
```

```
PS C:\Users\rhyth>
Server:
  Containers: 36
  Running: 34
  Paused: 0
  Stopped: 2
  Images: 12
  Server Version: 28.3.2
  Storage Driver: overlayfs
  Driver-BaseFS: overlayd.snapshotter.v1
  Log Driver: json-file
  Cgroup Driver: cgroups
  Cgroup Version: 2
  Plugins:
    Volume: local
    Network: bridge host ipvlan macvlan null overlay
    Log: awslogs fluentd gelf journald json-file local splunk syslog
  CDI spec directories:
    /var/run/cdi
  /var/run/cdi
  Discovered Devices:
    cdi: docker.com/gpu=webgpu
  Env: 0
  Runtimes: io.containerd.runc.v2 nvidia runc
  Default Runtime: runc
  Init Binary: docker-init
  Containerd Version: v2.0.14-0g04ec8a9a75232cad458027ca83437aae3f4da
  runc Version: v1.2.5-0-g59923ef
  init Version: dea8ad8
  Security Options:
    seccomp
      Profile: builtin
      cgroups
  Kernel Version: 6.6.87.2-microsoft-standard-WSL2
  Operating System: Docker Desktop
  OSType: linux
  Architecture: x86_64
  CPUs: 12
  Total Memory: 11.59GiB
  Max Memory: desktop
  ID: a95b7731-658c-424f-8d99-e6b7f65ac68b4
  Docker Root Dir: /var/lib/docker
  Debug Mode: false
  HTTP Proxy: http://docker.internal:3128
  HTTPS Proxy: https://docker.internal:3128
  No Proxy: hubproxy.docker.internal
  Labels:
  cook.docker.desktop.address=npipe://\\.\pipe\docker_cli
  Experimental: false
  Insecure Registries:
    hubproxy.docker.internal:5555
    127.0.0.1
    127.0.0.0/8
  Live Restore Enabled: false
WARNING: DOCKER_INSECURE_NO_IPTABLES_RAW is set
PS C:\Users\rhyth>
```

## Step 2: Pull an Image from Docker Hub

```
docker pull ubuntu:latest
```

*Output:*

```
PS C:\Users\rhyth> docker pull ubuntu:latest
latest: Pulling from library/ubuntu
a3629acbbf41: Pull complete
Digest: sha256:8e3606cef4fe3c4228f026b521fb76978881737d24f208828b2b
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\Users\rhyth> |
```

Check available images:

```
docker images
```

*Output:*

```
PS C:\Users\rhyth> docker images
REPOSITORY          TAG      IMAGE ID            CREATED             SIZE
ubuntu              latest   c010146a1b58    9 days ago        177MB
registry.k8s.io/kube-apiserver   latest   f48bf6a11e5a    5 months ago       220MB
hello-world          latest   f7931603f79e    5 months ago       20.3MB
docker/desktop-kubernetes   kubernetes-v1.32.2-cni-v1.6.0-critools-v1.31.1-cri-dockerd-v0.3.16-1-debian   fdd17202efdc   11 months ago      596MB
registry.k8s.io/kube-apiserver   v1.32.2  c47849f3e793    11 months ago      129MB
registry.k8s.io/kube-controller-manager   v1.32.2  393a0a15a343    11 months ago      159MB
registry.k8s.io/kube-proxy         v1.32.2  83c625ff9faa6   11 months ago      129MB
registry.k8s.io/kube-scheduler     v1.32.2  45710d7cfd5   11 months ago      93.5MB
registry.k8s.io/etcd              3.5.16-0  c6a90115c50    16 months ago      211MB
registry.k8s.io/coredns           v1.16.3  9ca3a6f65c50   20 months ago      55.1MB
registry.k8s.io/pause             3.1.0   tba5521f290b2   20 months ago      1.66MB
docker/desktop-vpnkit-controller   dc331cb22850be0cd97c84a9cfecaf44a1af06e  7ecf567ea870   2 years ago       47MB
docker/desktop-storage-provisioner   v2.0   115d77e7e62    4 years ago       59.2MB
PS C:\Users\rhyth> |
```

### Step 3: Run a Container

Run Ubuntu interactively:

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

*Output:*

```
PS C:\Users\rhyth> docker run -it --name myubuntu ubuntu bash
root@6f14448d055e:/# |
```

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

Exit:

```
exit
```

*Output:*

```
root@0f14448d033e:/# exit  
exit  
PS C:\Users\rhythth> |
```

## Step 4: List Containers

- Show running containers:

```
docker ps
```

*Output:*

```
PS C:\Users\rhythth> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
PS C:\Users\rhythth> |
```

- Show all containers (including stopped):

```
docker ps -a
```

*Output:*

```
PS C:\Users\rhythth> docker ps -a  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
0f14448d033e ubuntu "bash" 2 minutes ago Exited (0) About a minute ago  
d85ef3922bd2 hello-world "/hello" 2 months ago Exited (0) 2 months ago  
PS C:\Users\rhythth> |
```

## 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:*

---

```
PS C:\Users\rhyth> docker start myubuntu
myubuntu
PS C:\Users\rhyth> ^C
PS C:\Users\rhyth> docker stop myubuntu
myubuntu
PS C:\Users\rhyth> docker rm myubuntu
myubuntu
PS C:\Users\rhyth> docker ps -a
CONTAINER ID        IMAGE             COMMAND            CREATED           STATUS            PORTS      NAMES
d85ef3922bd2        hello-world      "/hello"          2 months ago     Exited (0) 2 months ago   affectionate_bhaskara
PS C:\Users\rhyth> |
```

## Step 6: Remove Images

```
docker rmi ubuntu:latest
```

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

*Output:*

```
PS C:\Users\rhyth> docker rmi ubuntu:latest
Untagged: ubuntu:latest
Deleted: sha256:cdd1bad51b3088c3686ecf4e3c4220f02eb521fb76978801737d24f206828b2b
PS C:\Users\rhyth> |
```

