

Lab Exercise 1- Docker Fundamental

Commands

Name:- Vansh Bhatt

Sap ID:- 500125395

Batch:- DevOps B1

To:- Hitesh Sharma Sir

Objective

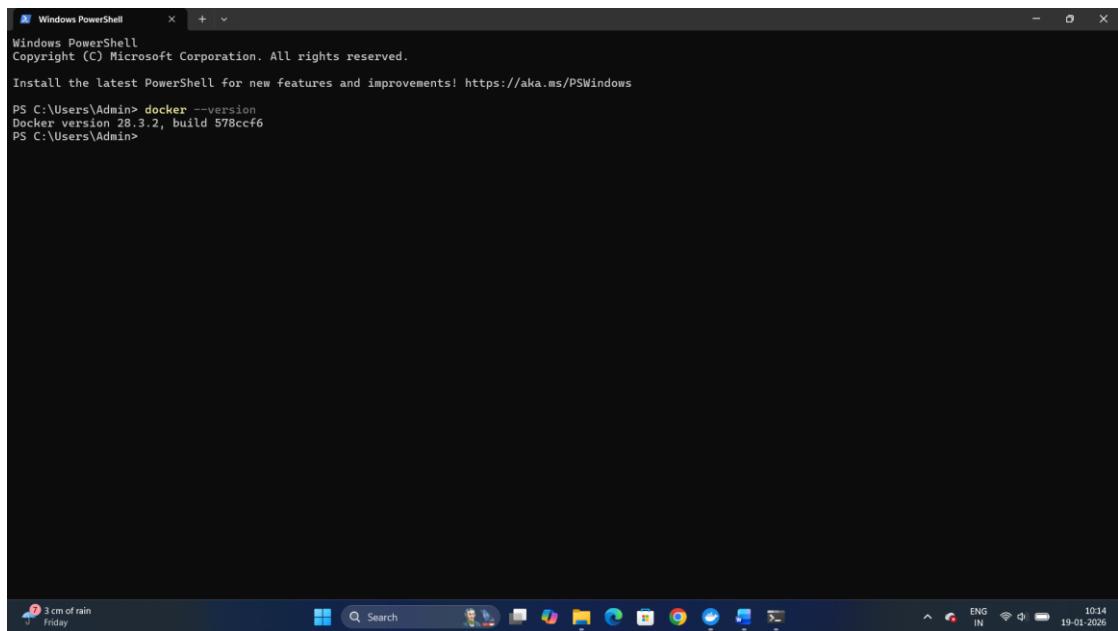
Learn and practice the fundamental Docker CLI commands to:

- 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



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

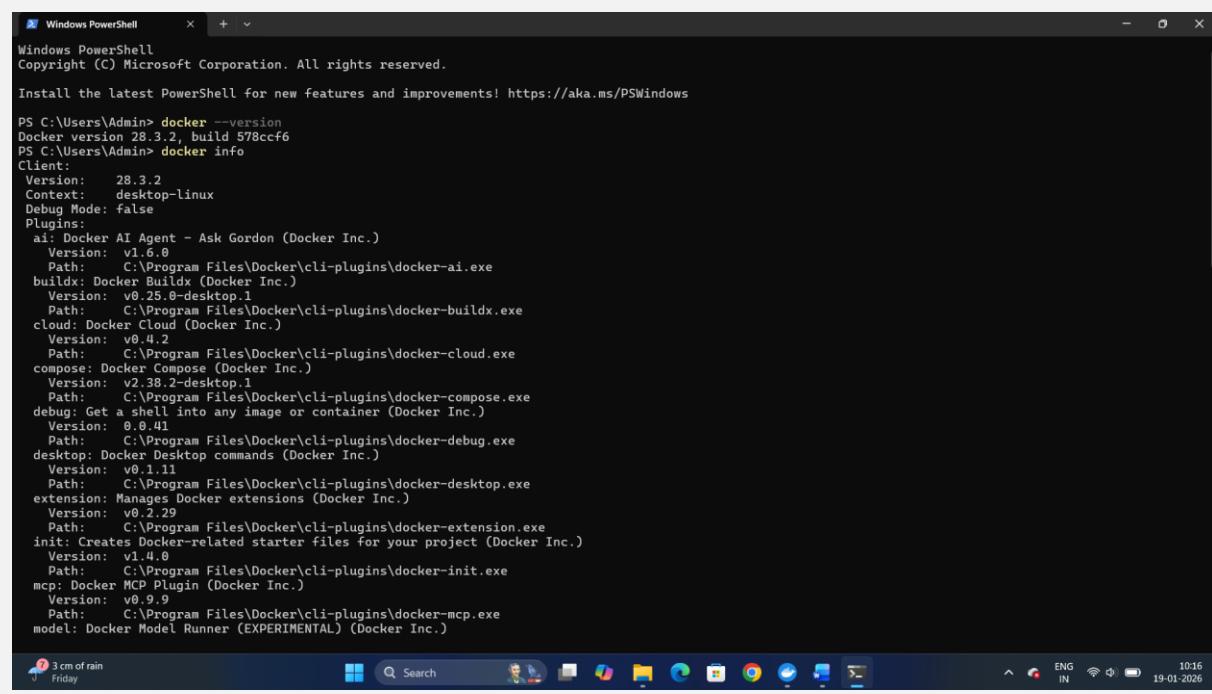
PS C:\Users\Admin> docker --version
Docker version 28.3.2, build 578ccf6
PS C:\Users\Admin>
```

2. Basic Docker Commands

Step 1: Verify Installation

```
docker --version
```

```
docker info
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

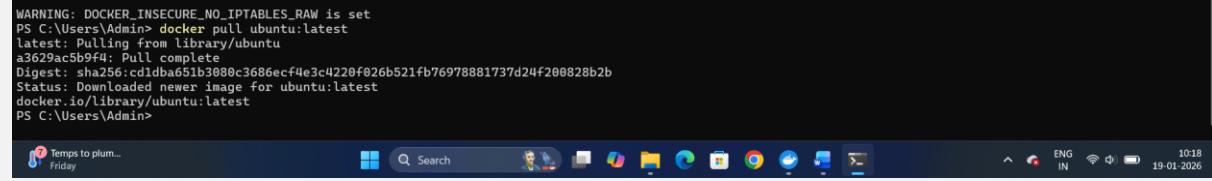
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin> docker --version
Docker version 28.3.2, build 578ccf6
PS C:\Users\Admin> 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: v0.0.41
      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.)
```

Expected Output: Docker version and system details.

Step 2: Pull an Image from Docker Hub

```
docker pull ubuntu:latest
```



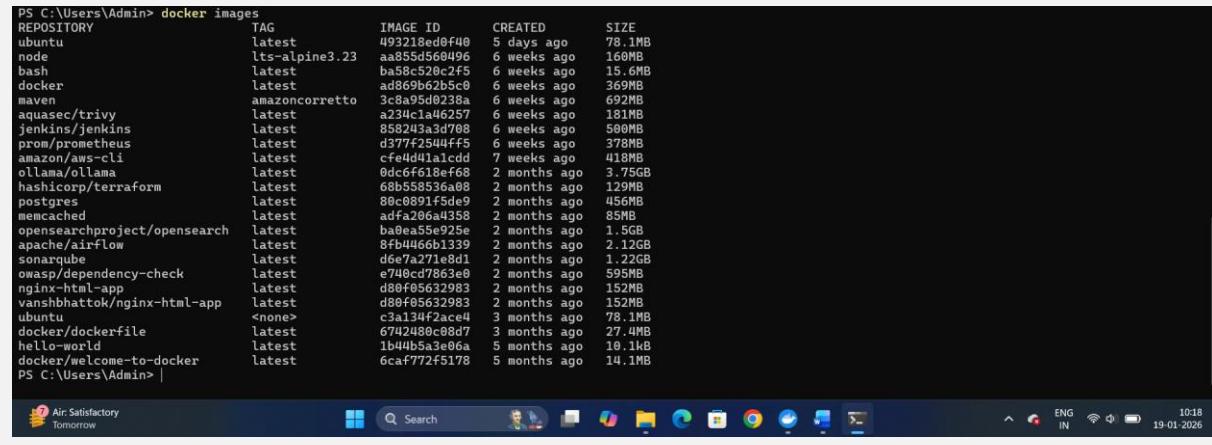
```
PS C:\Users\Admin> docker pull ubuntu:latest
latest: Pulling from library/ubuntu
a3629ac5b9f4: Pull complete
Digest: sha256:cdd1ba651b3080c3686ecf4e3c4220f026b521fb76978881737d24f200828b2b
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\Users\Admin>
```

The screenshot shows a Windows taskbar at the bottom with various icons. The terminal window is titled 'cmd' and contains the command 'docker pull ubuntu:latest'. The output shows the image is being pulled from the 'library/ubuntu' repository, with the digest and status message 'Downloaded newer image for ubuntu:latest'.

Explanation: Downloads the latest Ubuntu image from Docker Hub.

Check available images:

```
docker images
```



```
PS C:\Users\Admin> docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
ubuntu              latest   493218ed0f40  5 days ago   78.1MB
node                lts-alpine3.23  aae855d560a96  6 weeks ago   160MB
bash                latest   ba58c528c2f5  6 weeks ago   15.6MB
docker               latest   ad869b62b5c0  6 weeks ago   369MB
maven               amazoncorretto latest   3c8a95d8238a  6 weeks ago   692MB
aquasec/trivy       latest   a234c1a46257  6 weeks ago   181MB
jenkins/jenkins     latest   858243a3d788  6 weeks ago   580MB
prom/prometheus     latest   d377f42504ff5  6 weeks ago   378MB
amazon/aws-cli      latest   cfe4d11a1cdd  7 weeks ago   418MB
ollama/ollama        latest   0dc6f618ef68  2 months ago  3.75GB
hashicorp/terraform latest   68b558536a08  2 months ago  129MB
postgres            latest   80c0891f5de9  2 months ago  456MB
memcached           latest   adfa206a4558  2 months ago  85MB
opensearchproject/opensearch latest   ba0ea556925e  2 months ago  1.5GB
apache/airflow        latest   8fb4466b1339  2 months ago  2.12GB
sonarqube           latest   dbe7a271e8d1  2 months ago  1.22GB
owasp/dependency-check latest   e740cd7863e0  2 months ago  595MB
nginx-html-app      latest   d80f05632983  2 months ago  152MB
vanshahattok/nginx-html-app latest   d80f05632983  2 months ago  152MB
ubuntu               <none>  c3a134f2ace4  3 months ago  78.1MB
docker/dockerfile    latest   67f42480c08d7  3 months ago  27.4MB
hello-world          latest   1b44b5a3e06a  5 months ago  16.1kB
docker/welcome-to-docker latest   6caf772f5178  5 months ago  14.1MB
PS C:\Users\Admin>
```

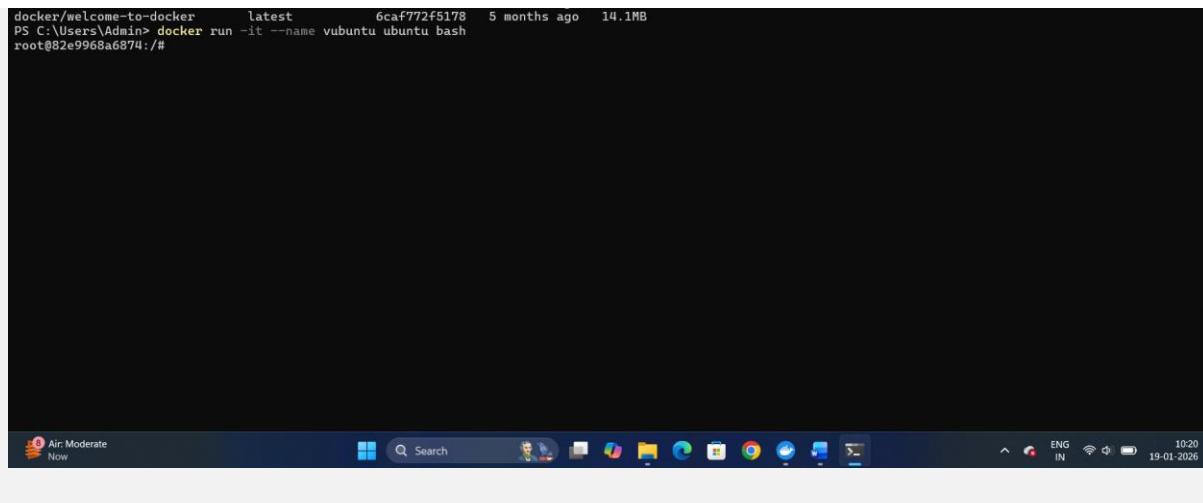
The screenshot shows a Windows taskbar at the bottom with various icons. The terminal window is titled 'cmd' and contains the command 'docker images'. The output lists several Docker images along with their tags, creation time, and size. The images include 'ubuntu', 'node', 'bash', 'docker', 'maven', 'aquasec/trivy', 'jenkins/jenkins', 'prom/prometheus', 'amazon/aws-cli', 'ollama/ollama', 'hashicorp/terraform', 'postgres', 'memcached', 'opensearchproject/opensearch', 'apache/airflow', 'sonarqube', 'owasp/dependency-check', 'nginx-html-app', 'vanshahattok/nginx-html-app', 'ubuntu', 'docker/dockerfile', 'hello-world', and 'docker/welcome-to-docker'.

Step 3: Run a Container

Run Ubuntu interactively:

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

```
PS C:\Users\Admin> docker run -it --name vubuntu ubuntu bash
root@82e9968a6874:/#
```

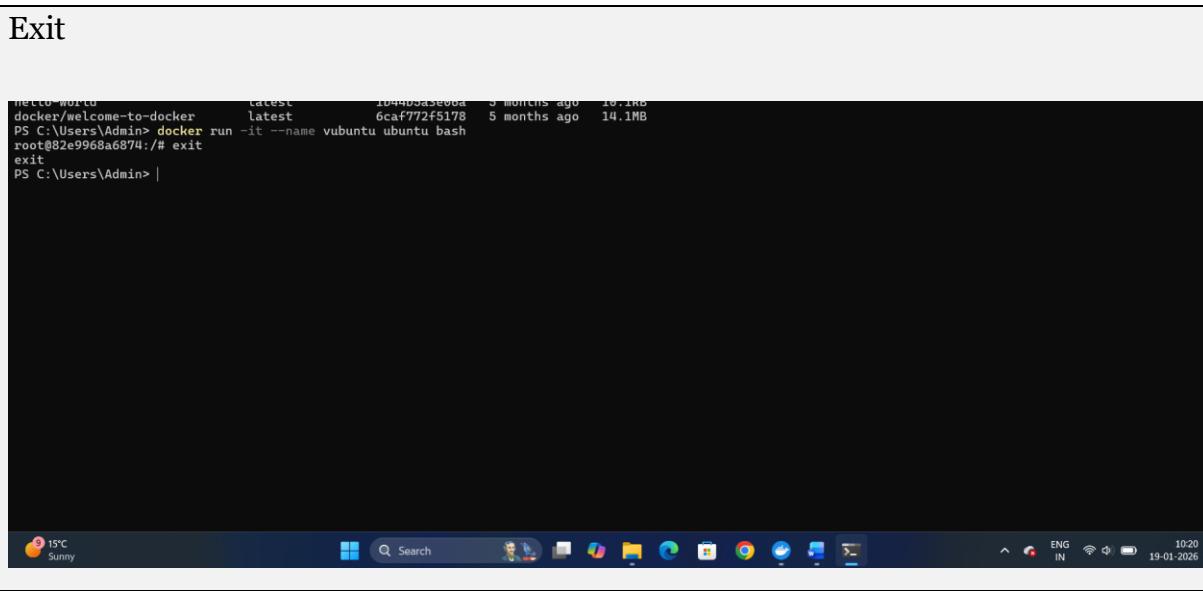


Now you're inside a running container (prompt will change).

Exit:

Exit

```
PS C:\Users\Admin> docker run -it --name vubuntu ubuntu bash
root@82e9968a6874:/# exit
PS C:\Users\Admin>
```



Step 4: List Containers

- Show running containers:

```
PS C:\Users\Admin> docker ps
```

```
exit  
PS C:\Users\Admin> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

- Show all containers (including stopped):

```
docker ps -a
```

```
PS C:\Users\Admin> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
82e9968a6874	ubuntu	"bash"	About a minute ago	Exited (0) 39 seconds ago	vubuntu	
0671aa52a7de	hello-world	"/hello"	2 months ago	Exited (0) 2 months ago	gallant_edison	
3fce5d5fb3c4	nginx-html-app	"/docker-entrypoint..."	2 months ago	Exited (255) 2 months ago	flamboyant_khayyam	

```
PS C:\Users\Admin>
```

Step 5: Start / Stop / Remove Containers

```
# Start a stopped container
```

```
docker start <container_id>
```

```
0.0.0:8006->80/tcp    flamboyant_khayyam  
PS C:\Users\Admin> docker start 82e9968a6874  
82e9968a6874  
PS C:\Users\Admin> |
```

```
# Stop a running container
```

```
docker stop <container_id>
```

```
PS C:\Users\Admin> docker stop 82e9968a6874  
82e9968a6874  
PS C:\Users\Admin> |
```



Remove a container

```
docker rm <container_id>
```

```
PS C:\Users\Admin> docker rm 82e9968a6874  
82e9968a6874
```



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

```
PS C:\Users\Admin> docker ps -aq  
0671aa52a7de  
3fce5d5fb3c4  
PS C:\Users\Admin> |
```



Step 6: Remove Images

```
docker rmi ubuntu:latest
```

```
PS C:\Users\Admin> docker rmi ubuntu:latest
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:cd1dba651b3080c3686ecf4e3c4220
f026b521fb76978881737d24f200828b2b
Deleted: sha256:493218ed0f404132311952996fea8ce85e50c4
9f5a717f26f25c52a25fcb2e56
Deleted: sha256:123a078714d5ea9382d4d9f550753aefce8b34
ec5ae11ae8273038d3bcbb943f
PS C:\Users\Admin> |
```

Focus - + 100%



^ ENG
IN 10:27
19-01-2026

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

Thank You