# Docker Notes

## 1. Docker Commands for Windows

Docker commands are the same across Windows, Linux, and macOS when using the command line. Below are some essential Docker commands:

### General Commands:

# Check Docker version

docker --version

# Display system-wide information

docker info

### Container Management:

# Run a container from an image

docker run -d --name mycontainer image\_name

# List running containers

docker ps

# List all containers (including stopped ones)

docker ps -a

# Stop a running container

docker stop container\_id

# Start a stopped container

docker start container\_id

# Remove a container

docker rm container\_id

### Image Management:

# List all downloaded images

docker images

# Remove an image

docker rmi image\_id

# Pull an image from Docker Hub

docker pull image\_name

### Volume Management:

# List all volumes

docker volume ls

# Create a volume

docker volume create myvolume

### Network Management:

# List all networks

docker network ls

# Create a network

docker network create mynetwork

## 2. What is Docker Compose?

Docker Compose is a tool that helps define and run multi-container Docker applications using a YAML configuration file (docker-compose.yml).

### Key Features:

* Simplifies multi-container deployment.
* Uses YAML for configuration.
* Allows defining networks, volumes, and services easily.

### Example docker-compose.yml:

version: '3.8'

services:

web:

image: nginx

ports:

- "8080:80"

database:

image: mysql

environment:

MYSQL\_ROOT\_PASSWORD: root

### Basic Commands:

# Start all services in detached mode

docker-compose up -d

# Stop and remove all services

docker-compose down

## 3. What is DOCKER\_PORT\_EXPOSE\_AND\_PUBLISH?

Docker handles ports in two ways:

* **EXPOSE**: Used in a Dockerfile to indicate that a container listens on a specific port.
* **PUBLISH**: When running a container, the -p flag explicitly maps a container’s port to the host.

### Example:

#### Dockerfile:

FROM nginx

EXPOSE 80

#### Run Container:

# Expose internally but not accessible from host

docker run -d nginx

# Expose and publish to host

docker run -d -p 8080:80 nginx

## 4. How to Install Docker on Windows and Ubuntu

### Install Docker on Windows:

1. Download and install **Docker Desktop** from <https://www.docker.com/products/docker-desktop/>.
2. Enable **WSL 2 Backend** (if using Windows 10/11).
3. Run the following command in PowerShell to verify installation:

docker --version

### Install Docker on Ubuntu:

1. Update system packages:

sudo apt update && sudo apt upgrade -y

1. Install required dependencies:

sudo apt install ca-certificates curl gnupg -y

1. Add Docker’s official GPG key:

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/keyrings/docker.asc > /dev/null

sudo chmod a+r /etc/apt/keyrings/docker.asc

1. Add Docker repository:

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

1. Install Docker Engine:

sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y

1. Start and enable Docker service:

sudo systemctl start docker

sudo systemctl enable docker

1. Verify installation:

docker --version

## Conclusion

* **Docker** simplifies application deployment with containerization.
* **Docker Compose** manages multi-container applications.
* **Port Exposure and Publishing** allow external access to containers.
* **Installation** differs slightly for Windows and Ubuntu but is straightforward.