Essential Docker Commands

1. Image Management Commands

Pull an Image

bash

```
docker pull <image_name>:<tag>
docker pull nginx:latest
docker pull ubuntu:20.04
```

Downloads an image from Docker Hub to your local machine.

List Images

bash

```
docker images
# or
docker image ls
```

Shows all images stored on your system.

Build an Image

bash

```
docker build -t <image_name>:<tag> <path>
docker build -t myapp:1.0 .
```

Creates an image from a Dockerfile. The `.` means current directory.

Remove an Image

bash

```
docker rmi <image_name>
docker rmi nginx:latest
docker rmi <image_id>
```

Deletes an image from your system.

Remove Unused Images

bash

```
docker image prune
docker image prune -a  # Remove all unused images
```

---

2. Container Management Commands

Run a Container

bash

```
docker run <image_name>
docker run nginx
docker run -d nginx                # Run in detached mode (background)
docker run -d -p 8080:80 nginx     # Map port 8080 to container port 80
docker run -d --name my-nginx nginx   # Give container a custom name
docker run -it ubuntu bash         # Interactive mode with terminal
```

Creates and starts a container from an image.

Common flags:

- `-d` : Detached mode (runs in background)
- `-p` : Port mapping (host:container)
- `--name` : Give container a name
- `-it` : Interactive terminal
- `-v` : Mount volume
- `-e` : Set environment variables

List Containers

bash

```
docker ps          # Running containers only
docker ps -a       # All containers (including stopped)
docker container ls    # Same as docker ps
```

Start/Stop/Restart Containers

bash

```
docker start <container_id/name>
docker stop <container_id/name>
docker restart <container_id/name>
```

Remove a Container

bash

```
docker rm <container_id/name>
docker rm -f <container_id>        # Force remove (even if running)
```

```
docker container prune          # Remove all stopped containers
```

Execute Command in Running Container

bash

```
docker exec -it <container_id/name> <command>
docker exec -it my-nginx bash       # Open bash shell in container
docker exec my-nginx ls /usr/share  # Run ls command
```

Runs a command inside a running container.

---

3. Container Inspection & Logs

View Container Logs

bash

```
docker logs <container_id/name>
docker logs -f <container_id>       # Follow logs (like tail -f)
docker logs --tail 100 <container_id> # Last 100 lines
```

Inspect Container Details

bash

```
docker inspect <container_id/name>
```

Shows detailed configuration and information in JSON format.

View Container Resource Usage

bash

```
docker stats
docker stats <container_id>         # Stats for specific container
```

Shows CPU, memory, network usage in real-time.

View Running Processes

bash

```
docker top <container_id/name>
```

Shows processes running inside a container.

---

4. Volume Management Commands

Create a Volume

bash

```
docker volume create <volume_name>
docker volume create my-data
```

List Volumes

bash

```
docker volume ls
```

Use Volume with Container

bash

```
docker run -v <volume_name>:<container_path> <image>
docker run -v my-data:/var/lib/mysql mysql
docker run -v /host/path:/container/path nginx  # Bind mount
```

Remove Volume

bash

```
docker volume rm <volume_name>
docker volume prune              # Remove unused volumes
```

---

5. Network Commands

List Networks

bash

```
docker network ls
```

Create a Network

bash

```
docker network create <network_name>
docker network create my-network
```

Connect Container to Network

bash

```
docker network connect <network_name> <container_id>
```

Inspect Network

bash

```
docker network inspect <network_name>
```

---

6. Docker Compose Commands

Start Services

bash

```
docker-compose up
docker-compose up -d          # Detached mode
```

Stop Services

bash

```
docker-compose down
docker-compose down -v        # Also remove volumes
```

View Logs

bash

```
docker-compose logs
docker-compose logs -f        # Follow logs
```

List Services

bash

```
docker-compose ps
```

---

7. System & Cleanup Commands

View Disk Usage

bash

```
docker system df
```

Clean Up Everything

bash

```
docker system prune           # Remove unused data
docker system prune -a        # Remove all unused images too
```

```
docker system prune --volumes       # Include volumes
```

Docker Version & Info

bash

```
docker --version
docker version              # Detailed version info
docker info                 # System-wide information
```

---

8. File Operations

Copy Files Between Host and Container

bash

```
docker cp <container_id>:/path/in/container /host/path
docker cp /host/path <container_id>:/path/in/container
```

---

9. Common Workflow Examples

Example 1: Run a Web Server

bash

```
# Pull nginx image
docker pull nginx

# Run nginx on port 8080
docker run -d -p 8080:80 --name my-web nginx

# View logs
docker logs my-web

# Stop and remove
docker stop my-web
docker rm my-web
```

Example 2: Run MySQL Database

bash

```
docker run -d \
  --name my-mysql \
```

```
  -e MYSQL_ROOT_PASSWORD=mypassword \
  -v mysql-data:/var/lib/mysql \
  -p 3306:3306 \
  mysql:8.0
```

Example 3: Interactive Ubuntu Container

bash

```
docker run -it --rm ubuntu:20.04 bash
# --rm automatically removes container when you exit
```

Quick Reference Cheat Sheet

Task	Command
Pull image	`docker pull nginx`
Run container	`docker run -d -p 8080:80 nginx`
List running containers	`docker ps`
List all containers	`docker ps -a`
Stop container	`docker stop <id>`
Remove container	`docker rm <id>`
View logs	`docker logs <id>`
Execute command	`docker exec -it <id> bash`
List images	`docker images`
Remove image	`docker rmi <image>`
Clean up	`docker system prune -a`

These commands cover 95% of daily Docker usage! Practice them to become proficient with Docker.