



## Basic Commands

### Image Management

<code>docker pull &lt;image&gt;</code>	Download an image from Docker Hub.
<code>docker images</code>	List available images locally.
<code>docker rmi &lt;image&gt;</code>	Remove an image.
<code>docker build -t &lt;image&gt; .</code>	Build an image from a Dockerfile in the current directory.
<code>docker tag &lt;source_image&gt; &lt;target_image&gt;</code>	Tag an image.
<code>docker push &lt;image&gt;</code>	Push image to Docker Hub or registry.

### Container Management

<code>docker run &lt;image&gt;</code>	Create and start a container.
<code>docker ps</code>	List running containers.
<code>docker ps -a</code>	List all containers (running and stopped).
<code>docker stop &lt;container&gt;</code>	Stop a running container.
<code>docker start &lt;container&gt;</code>	Start a stopped container.
<code>docker restart &lt;container&gt;</code>	Restart a container.
<code>docker rm &lt;container&gt;</code>	Remove a stopped container.
<code>docker exec -it &lt;container&gt; &lt;command&gt;</code>	Execute a command inside a container.

### Networking

<code>docker network create &lt;network&gt;</code>	Create a new network.
<code>docker network ls</code>	List available networks.
<code>docker network connect &lt;network&gt; &lt;container&gt;</code>	Connect a container to a network.
<code>docker port &lt;container&gt;</code>	List port mappings for a container.

## Dockerfile Instructions

### Essential Instructions

<code>FROM &lt;image&gt;</code>	Specifies the base image for the Dockerfile.
<code>RUN &lt;command&gt;</code>	Executes commands during the image build process.
<code>CMD &lt;command&gt;</code>	Specifies the default command to run when the container starts.
<code>EXPOSE &lt;port&gt;</code>	Declares the port the container listens on.
<code>ENV &lt;key&gt; &lt;value&gt;</code>	Sets environment variables.
<code>COPY &lt;src&gt; &lt;dest&gt;</code>	Copies files/directories from the host to the container.
<code>ADD &lt;src&gt; &lt;dest&gt;</code>	Similar to COPY, but can also extract archives and fetch URLs.
<code>WORKDIR &lt;path&gt;</code>	Sets the working directory for subsequent instructions.

### User and Volume Management

<code>USER &lt;user&gt;</code>	Sets the user for subsequent RUN, CMD, and ENTRYPOINT instructions.
<code>VOLUME &lt;path&gt;</code>	Creates a mount point for persistent storage.
<code>STOPSIGNAL &lt;signal&gt;</code>	Signal to be used to stop the container
<code>ARG &lt;name&gt;[=&lt;default value&gt;]</code>	Defines a build argument.

## Docker Compose

### Compose File Structure

A <code>docker-compose.yml</code> file defines services, networks, and volumes for a multi-container Docker application.	
Key elements include:	
• <code>version</code>	: Specifies the Compose file version.
• <code>services</code>	: Defines individual containers.
• <code>networks</code>	: Defines networks used by the services.
• <code>volumes</code>	: Defines persistent data volumes.

Common Compose Commands

<code>docker compose up</code>	Builds, (re)creates, starts, and attaches to containers defined in a <code>docker-compose.yml</code> file.
<code>docker compose up -d</code>	Runs the containers in detached mode (background).
<code>docker compose down</code>	Stops and removes containers, networks, and volumes defined in the Compose file.
<code>docker compose ps</code>	Lists the containers defined in the Compose file.
<code>docker compose logs</code>	View the logs of the containers.
<code>docker compose stop</code>	Stop services
<code>docker compose start</code>	Start services
<code>docker compose restart</code>	Restart services

Service Configuration

<code>image</code>	Specifies the image to use for the service.
<code>build</code>	Specifies the path to the Dockerfile to build the image.
<code>ports</code>	Exposes ports from the container to the host.
<code>volumes</code>	Mounts volumes to the container.
<code>environment</code>	Sets environment variables for the service.
<code>depends_on</code>	Defines service dependencies.
<code>networks</code>	Attaches the service to a network.

Docker Swarm

Swarm Initialization and Management

<code>docker swarm init</code>	Initializes a new Swarm cluster. Run this on the manager node.
<code>docker swarm join</code>	Joins a node to an existing Swarm cluster. Use the token provided by <code>docker swarm init</code> .
<code>docker swarm leave</code>	Leaves the Swarm cluster. Use <code>--force</code> if the node is unreachable.
<code>docker node ls</code>	Lists the nodes in the Swarm cluster.
<code>docker node inspect &lt;node_id&gt;</code>	Inspect a specific node.
<code>docker node update &lt;node_id&gt;</code>	Update node's role and availability.

Service Deployment and Scaling

<code>docker service create</code>	Creates a new service in the Swarm cluster.
<code>docker service ls</code>	Lists the services running in the Swarm cluster.
<code>docker service update</code>	Updates an existing service (e.g., scale, update image).
<code>docker service scale &lt;service_id&gt;=&lt;number_of_replicas&gt;</code>	Scales a service to the specified number of replicas.
<code>docker service inspect &lt;service_id&gt;</code>	Inspect a specific service.
<code>docker service logs &lt;service_id&gt;</code>	View logs for a specific service.
<code>docker service rm &lt;service_id&gt;</code>	Removes a service from the Swarm cluster.

Stack Deployment

<code>docker stack deploy -c &lt;compose_file&gt;.yaml &lt;stack_name&gt;</code>	Deploy a stack based on a compose file.
<code>docker stack ls</code>	List deployed stacks.
<code>docker stack rm &lt;stack_name&gt;</code>	Remove a deployed stack.
<code>docker stack ps &lt;stack_name&gt;</code>	List the tasks in the stack.