

**Build Commands:** Docker uses the build command for building images from a Docker file. Some of the most common commands include:

Command	Explanation
<code>docker build</code>	Builds an image from a Dockerfile in the current directory
<code>docker build https://github.com/docker/rootfs.git#container:docker</code>	Builds an image from a remote GIT repository
<code>docker build -t imagename/tag</code>	Builds and tags an image for easier tracking
<code>docker build https://yourserver/file.tar.gz</code>	Builds an image from a remote tar archive
<code>docker build -t image:1.0 -&lt;&lt;EOFFFROM busyboxRUN echo "hello world"EOF</code>	Builds an image via a Dockerfile that is passed through STDIN

**Clean Up Commands:** To keep your system clean and save disk space, it's a great idea to clean up unused images, containers, and volumes. Check the commands below for more details:

Command	Explanation
<code>docker image prune</code>	Clears an unused image
<code>docker image prune -a</code>	Clears all images that are not being used by containers
<code>docker system prune</code>	Removes all stopped containers, all networks not used by containers, all dangling images, and all build cache
<code>docker image rm image</code>	Removes an image
<code>docker rm container</code>	Removes a running container

<code>docker kill \$(docker ps -q)</code>	Stops all running containers
<code>docker swarm leave</code>	Leaves a swarm
<code>docker stack rm stackname</code>	Removes a swarm
<code>docker volume rm \$(docker volume ls -f dangling=true -q)</code>	Removes all dangling volumes
<code>docker rm \$(docker ps -a -q)</code>	Removes all stopped containers
<code>docker kill \$(docker ps -q)</code>	Stops all running containers

**Container Interaction Commands:** Interact with your Docker container with the following common commands:

Command	Explanation
<code>docker start container</code>	<a href="#">Starts a new container</a>
<code>docker stop container</code>	Stops a container
<code>docker pause container</code>	Pauses a container
<code>docker unpause container</code>	Unpauses a container
<code>docker restart container</code>	Restarts a container
<code>docker wait container</code>	Blocks a container
<code>docker export container</code>	Exports container contents to a tar archive
<code>docker attach container</code>	Attaches to a running container

<code>docker wait container</code>	Waits until the container is terminated and shows the exit code
<code>docker commit -m "commit message" -a "author" container username/image_name: tag</code>	Saves a running container as an image
<code>docker logs -ft container</code>	Follows container logs
<code>docker exec -ti container script.sh</code>	Runs a command in a container
<code>docker commit container image</code>	Creates a new image from a container
<code>docker create image</code>	Creates a new container from an image

**Container Inspection Commands:** Sometimes, you need to inspect your containers for quality assurance or troubleshooting purposes. These commands help you get an overview of what different containers are doing:

Command	Explanation
<code>docker ps</code>	Lists all running containers
<code>docker -ps -a</code>	Lists all containers
<code>docker diff container</code>	Inspects changes to directories and files in the container filesystem
<code>docker top container</code>	Shows all running processes in an existing container
<code>docker inspect container</code>	Displays low-level information about a container
<code>docker logs container</code>	Gathers the logs for a container

```
docker stats
container
```

Shows container resource usage statistics

**Manage Images Commands:** Some of the most common image management commands include:

Command	Explanation
<code>docker image ls</code>	Lists images
<code>docker image rm mysql</code>	Removes an image
<code>docker tag image tag</code>	Tags an image
<code>docker history image</code>	Displays the image history
<code>docker inspect image</code>	Displays low-level information about an image

**Run Commands:** Docker uses the run command to create containers from provided images. The default syntax for this command looks like this:

```
docker run (options) image (command) (arg...)
```

After the default syntax, use one of the following flags:

Flag	Explanation
<code>--detach , -d</code>	Runs a container in the background and prints the container ID
<code>--env , -e</code>	Sets environment variables
<code>--hostname , -h</code>	Sets a hostname to a container
<code>--label , -l</code>	Creates a meta data label for a container
<code>--name</code>	Assigns a name to a container

<code>--network</code>	Connects a container to a network
<code>--rm</code>	Removes container when it stops
<code>--read-only</code>	Sets the container filesystem as read-only
<code>--workdir</code> , <code>-w</code>	Sets a working directory in a container

**Registry Commands:** If you need to interact with Docker Hub, use the following commands:

Command	Explanation
<code>docker login</code>	Logs in to a registry
<code>docker logout</code>	Logs out from a registry
<code>docker pull mysql</code>	Pulls an image from a registry
<code>docker push repo/ rhel-httpd:latest</code>	Pushes an image to a registry
<code>docker search term</code>	Searches Docker Hub for images with the specified term

**Service Commands:** Manage all Docker services with these basic commands:

Command	Explanation
<code>docker service ls</code>	Lists all services running in a swarm
<code>docker stack services stackname</code>	Lists all running services
<code>docker service ps servicename</code>	Lists the tasks of a service
<code>docker service update servicename</code>	Updates a service
<code>docker service create image</code>	Creates a new service

```
docker service scale  
servicename=10
```

Scales one or more replicated services

```
docker service logs stackname  
servicename
```

Lists all service logs

**Network Commands:** If you need to interact with the Docker network, use one of the following commands:

Command	Explanation
docker network create networkname	Creates a new network
docker network rm networkname	Removes a specified network
docker network ls	Lists all networks
docker network connect networkname container	Connects a container to a network
docker network disconnect networkname container	Disconnects a container from a network
docker network inspect networkname	Displays detailed information about a network