Create a container (without starting it):
docker create [IMAGE]

Rename an existing container:
docker rename [CONTAINER_NAME] [NEW_CONTAINER_NAME]

Run a command in a new container:
docker run [IMAGE] [COMMAND]

docker run --rm [IMAGE] – removes a container after it exits.

docker run -td [IMAGE] – starts a container and keeps it running.

docker run -it [IMAGE] – starts a container, allocates a pseudo-TTY connected to the container's stdin, and creates an interactive bash shell in the container.

docker run -it-rm [IMAGE] – creates, starts, and runs a command inside the container. Once it executes the command, the container is removed.

Delete a container (if it is not running):
docker rm [CONTAINER]

Update the configuration of one or more containers:
docker update [CONTAINER]

Starting and Stopping Containers

Start a container:
docker start [CONTAINER]

Stop a running container:
docker stop [CONTAINER]

Stop a running container and start it up again:
docker restart [CONTAINER]

Pause processes in a running container:
docker pause [CONTAINER]

Unpause processes in a running container:
docker unpause [CONTAINER]

Block a container until others stop (after which it prints their exit codes):
docker wait [CONTAINER]

Kill a container by sending a SIGKILL to a running container:
docker kill [CONTAINER]

Attach local standard input, output, and error streams to a running container:
docker attach [CONTAINER]

Docker Image Commands

Create an image from a Dockerfile:
docker build [URL]

docker build -t :– builds an image from a Dockerfile in the current directory and tags the image

Pull an image from a registry:
docker pull [IMAGE]

Push an image to a registry:
docker push [IMAGE]

Create an image from a tarball:
docker import [URL/FILE]

Create an image from a container:
docker commit [CONTAINER] [NEW_IMAGE_NAME]

Remove an image:
docker rmi [IMAGE]

Load an image from a tar archive or stdin:
docker load [TAR_FILE/STDIN_FILE]

Save an image to a tar archive, streamed to STDOUT with all parent layers, tags, and versions:
docker save [IMAGE] > [TAR_FILE]

List running containers:
docker ps
docker ps -a – lists both running containers and ones that have stopped

List the logs from a running container:
docker logs [CONTAINER]

List low-level information on Docker objects:
```
docker inspect [OBJECT_NAME/ID]
```

List real-time events from a container:
```
docker events [CONTAINER]
```

Show port (or specific) mapping for a container:
```
docker port [CONTAINER]
```

Show running processes in a container:
```
docker top [CONTAINER]
```

Show live resource usage statistics of containers:
```
docker stats [CONTAINER]
```

Show changes to files (or directories) on a filesystem:
```
docker diff [CONTAINER]
```

List all images that are locally stored with the docker engine:
```
docker image ls
```

Show the history of an image:
```
docker history [IMAGE]
```

Networks Command

One of the most valuable features of Docker software is the ability to connect containers to each other and to other non-Docker workloads. This section covers network-related commands.

List networks:
```
docker network ls
```

Remove one or more networks:
```
docker network rm [NETWORK]
```

Show information on one or more networks:
```
docker network inspect [NETWORK]
```

Connects a container to a network:
```
docker network connect [NETWORK] [CONTAINER]
```

Disconnect a container from a network:
```
docker network disconnect [NETWORK] [CONTAINER]
```