Docker General

Display the Docker version information.

```
docker version
docker version
```

Display system-wide information about Docker and its components

```
docker info
docker info
```

Clean up unused networks, volumes, and images

```
docker system prune
docker system prune
```

Docker Build and Run

Build Image from Dockerfile

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

Build Image with No Cache

```
docker build --no-cache -t <image_name>:<tag> <path_to_dockerfile>
docker build --no-cache -t myapp:1.0 .
```

Build Image with Build Argument

```
docker build --build-arg <key>=<value> -t <image_name>:<tag> <path_to_dockerfile>
docker build --build-arg ENVIRONMENT=production -t myapp:1.0 .
```

Build Image with Docker Compose

```
docker-compose build
docker-compose build web
```

Run Container Interactively

```
docker run -it <image_name>:<tag> /bin/
docker run -it myapp:1.0 /bin/
```

Run a Docker container in the background (detached mode)

```
docker run -d -p <host_port>:<container_port> <image_name>:<tag>
docker run -d -p 8080:80 myapp:latest
```

Run a Docker container with a volume mounted

```
docker run -v <host_path>:<container_path> <image_name>:<tag>
docker run -v /path/on/host:/path/in/container myapp:1.0
```

Run a Docker container with an environment variable

```
docker run -e <key>=<value> <image_name>:<tag>
docker run -e DATABASE_URL=postgresql://user:password@host:port/db myapp:1.0
```

Maps a host port to a container port when running a Docker container

```
docker run -p <host_port>:<container_port> <image_name>:<tag>
docker run -p 8080:80 myapp:latest
```

By Sanjay Lalwani

Docker Login

Logs in to a Docker registry

```
docker login
```

```
docker login
```

You will be prompted to enter your Docker Hub username, password, and optionally, an access token.

Docker Image

DownLoad a Docker image from a registry

```
docker pull <image_name>
```

```
docker pull nginx:latest
```

Display a list of locally available Docker images

```
docker images
```

```
docker images
```

Remove a Docker image from the local machine

```
docker rmi <image_name>
```

```
docker rmi nginx:latest
```

Builds a Docker image from a Dockerfile

```
docker build -t <image_name>:<tag> <path_to_dockerfile>
```

```
docker build -t myapp:1.0 .
```

Assigns a new tag to an existing Docker image

```
docker tag <source_image>:<source_tag> <target_image>:<target_tag>
```

```
docker tag myapp:1.0 myapp:latest
```

UpLoad a Docker image to a container registry

```
docker push <image_name>:<tag>
```

```
docker push myapp:latest
```

Display detailed information about a Docker image

```
docker image inspect <image_name>
```

```
docker image inspect myapp:latest
```

Search for Docker images on a container registry

```
docker search <search_term>
```

```
docker search ubuntu
```

Save a Docker image to a tar archive file

```
docker save -o <output_file.tar> <image_name>:<tag>
```

```
docker save -o myapp.tar myapp:latest
```

Load a Docker image from a tar archive file

```
docker load -i <input_file.tar>
```

```
docker load -i myapp.tar
```

By Sanjay Lalwani

Docker Container

Start a new container from a Docker image.

```
docker run <image_name>:<tag>
docker run nginx:latest
```

Run a container in the background (detached mode).

```
docker run -d <image_name>:<tag>
docker run -d nginx:latest
```

List the currently running containers.

```
docker ps
docker ps
```

List all containers, including stopped ones.

```
docker ps -a
docker ps -a
```

Display detailed information about a container.

```
docker inspect <container_id_or_name>
docker inspect my_container
```

Attaches to a running container, allowing interaction with its console.

```
docker attach <container_id_or_name>
docker attach my_container
```

Stop a running container gracefully.

```
docker stop <container_id_or_name>
docker stop my_container
```

Forcefully Stop a running container.

```
docker kill <container_id_or_name>
docker kill my_container
```

Remove a stopped container.

```
docker rm <container_id_or_name>
docker rm my_container
```

Executes a command inside a running container.

```
docker exec -it <container_id_or_name> <command>
docker exec -it my_container
```

By Sanjay Lalwani