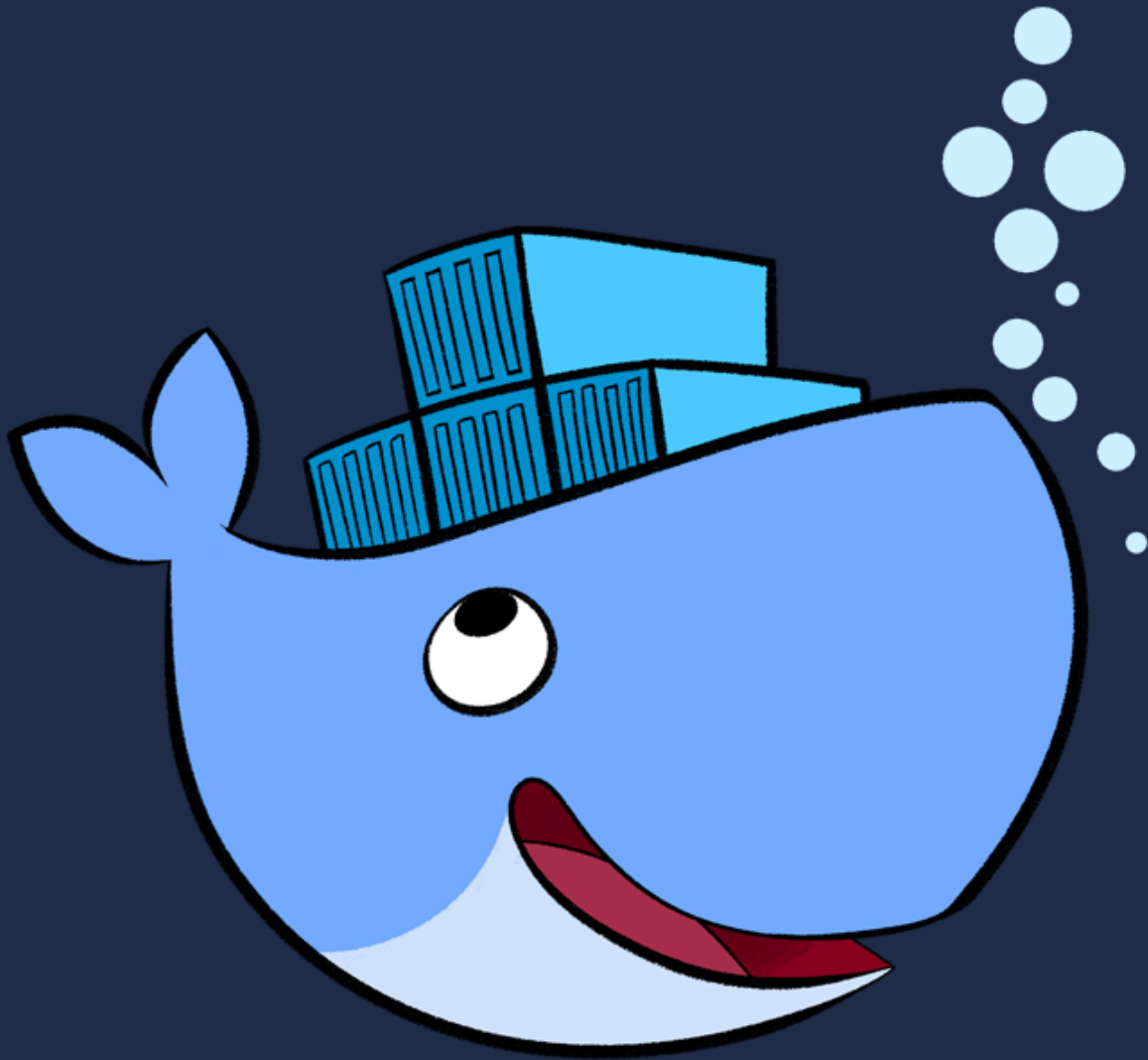




Docker Cheatsheet

Common commands and their functions.



By Albert Arakelyan

1. docker build

```
docker build -t your_image_name .
```

- **Purpose:** Creates an image from a **Dockerfile**.
- **Explanation:** This command builds a Docker image based on the instructions in a **Dockerfile**, usually located in the current directory **(.)**. The **t** flag allows you to tag the image with a name.

2. docker run

```
docker run -d -p 80:80 --name webserver nginx
```

- **Purpose:** Runs a new container from an image.
- **Explanation:** Creates a new container from an image (e.g., **nginx**) and runs it. Options include **d** (detached mode, runs in the background), **p** (maps a port, e.g., **80:80**), and **-name** (gives the container a name, e.g., **webserver**).

3. docker start

```
docker start container_name
```

- **Purpose:** Starts an existing, stopped container.
- **Explanation:** If you have a container that was created and stopped, this command will start it again without creating a new one.

4. docker stop



```
docker stop container_name
```

- **Purpose:** Stops a running container.
- **Explanation:** This gracefully stops a container by shutting down the processes running inside it.

5. docker ps



```
docker ps
```

- **Purpose:** Lists running containers.
- **Explanation:** Shows all currently running containers and their details like container ID, name, image, status, ports, etc. Use `docker ps -a` to list all containers, including stopped ones.

6. docker exec

```
docker exec -it container_name bash
```

- **Purpose:** Executes a command inside a running container.
- **Explanation:** This command allows you to run commands inside a container. The `-it` flag runs an interactive shell (like `bash`) inside the container.

7. docker rm

```
docker rm container_name
```

- **Purpose:** Removes a stopped container.
- **Explanation:** Deletes a container from your system. The container must be stopped first (or you can use `docker rm -f container_name` to forcefully remove a running container).

8. docker rmi

```
docker rmi image_name
```

- **Purpose:** Removes an image.
- **Explanation:** Deletes a Docker image from your local system. You may want to use ``docker images`` to list images first. Note: You cannot remove an image if containers based on it are still running.

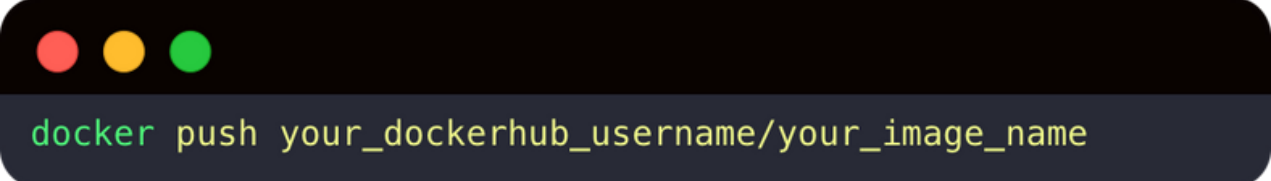
9. docker pull



```
docker pull mysql
```

- **Purpose:** Downloads an image from Docker Hub (or another registry).
- **Explanation:** Retrieves the specified image (e.g., ``mysql``) from Docker Hub and saves it locally so you can use it later.

10. docker push



```
docker push your_dockerhub_username/your_image_name
```

- **Purpose:** Uploads an image to a Docker registry.
- **Explanation:** Pushes an image from your local system to a Docker registry (e.g., Docker Hub) so others can download and use it.

11. docker commit



```
docker commit container_name new_image_name
```

- **Purpose:** Creates a new image from a container's changes.
- **Explanation:** If you've made changes to a container (like installing software), you can save those changes as a new image with this command.

12. docker logs



```
docker logs container_name
```

- **Purpose:** Shows the logs from a running or stopped container.
- **Explanation:** Displays the output (logs) generated by a container's processes. This is useful for debugging or tracking what's happening inside the container.

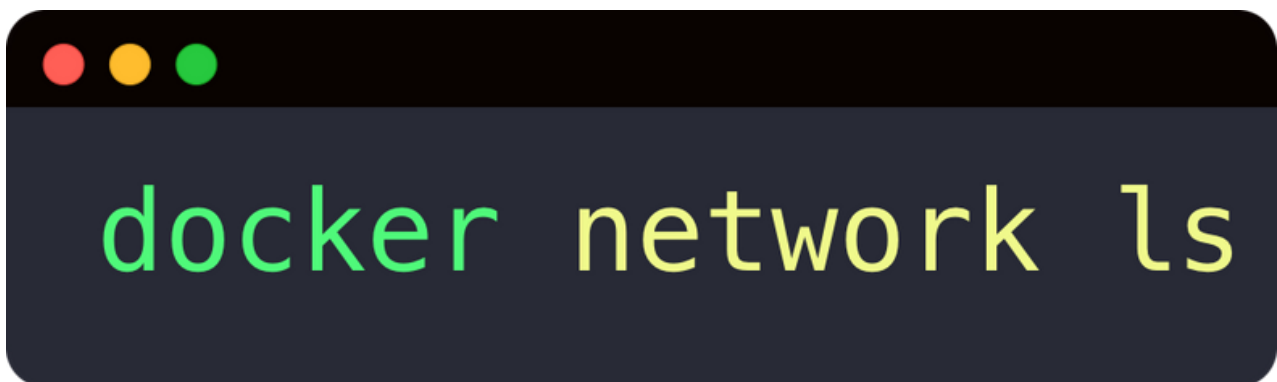
13. docker inspect



```
docker inspect container_name
```

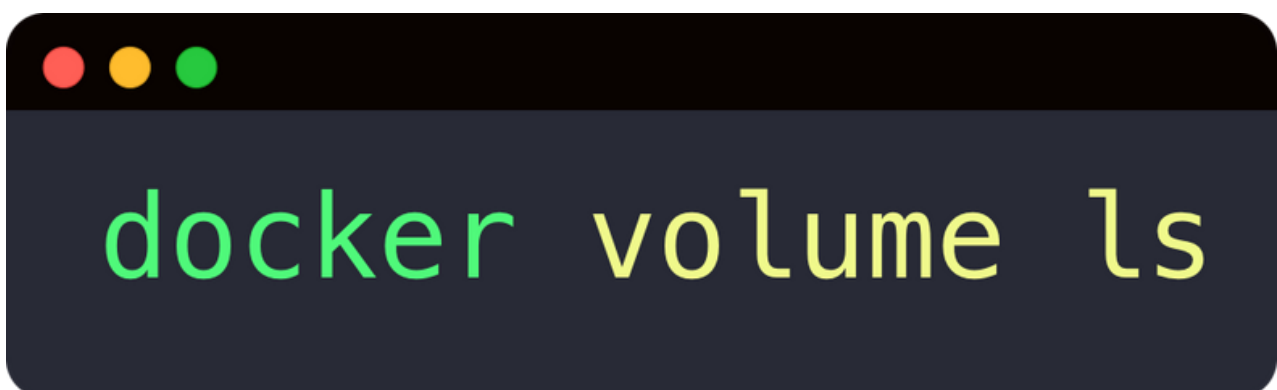
- **Purpose:** Shows detailed information about a container or image.
- **Explanation:** Provides detailed, low-level information (in JSON format) about a container, image, or other Docker object.

14. docker network ls



- **Purpose:** Lists all Docker networks.
- **Explanation:** Displays the available Docker networks on your system. Networks allow containers to communicate with each other.

15. docker volume ls



- **Purpose:** Lists all Docker volumes.
- **Explanation:** Shows all volumes, which are used to store persistent data outside of a container's lifecycle.

16. docker-compose up



```
docker-compose up -d
```

- **Purpose:** Starts services defined in a `docker-compose.yml` file.
- **Explanation:** Runs multiple containers as defined in a `docker-compose.yml` file. The `-d` flag runs it in detached mode. Containers are automatically networked and can share resources.

17. docker-compose down



```
docker-compose down
```

- **Purpose:** Stops and removes all containers defined in a `docker-compose.yml`.
- **Explanation:** Shuts down the running containers and removes networks, volumes, and containers created by `docker-compose up`.

18. docker image prune



```
docker image prune
```

- **Purpose:** Removes unused Docker images.
- **Explanation:** Cleans up unused images (those not referenced by any containers) to free up disk space.

19. docker system prune



```
docker system prune
```

- **Purpose:** Cleans up unused resources (containers, networks, images, etc.).
- **Explanation:** This command removes all unused containers, networks, images, and optionally, volumes, freeing up space.

20. docker tag

```
docker tag source_image new_image_name
```

- **Purpose:** Creates a tag for an image.
- **Explanation:** Tags an image with a new name (and optionally a version). This is useful for versioning or preparing an image for a registry push.

Summary Table

Command	Purpose	Example
docker build	Build an image from a Dockerfile	docker build -t my_image .
docker run	Run a new container from an image	docker run -d -p 80:80 nginx
docker start	Start an existing container	docker start container_name
docker stop	Stop a running container	docker stop container_name
docker ps	List running containers	docker ps
docker exec	Run a command inside a container	docker exec -it container_name bash
docker rm	Remove a stopped container	docker rm container_name
docker rmi	Remove an image	docker rmi image_name
docker pull	Download an image from a registry	docker pull mysql
docker push	Upload an image to a registry	docker push your_image_name
docker commit	Create a new image from a container	docker commit container_name new_image
docker logs	View container logs	docker logs container_name
docker inspect	Inspect a container or image	docker inspect container_name
docker network ls	List Docker networks	docker network ls
docker volume ls	List Docker volumes	docker volume ls
docker-compose up	Start services from a docker-compose.yml	docker-compose up -d
docker-compose down	Stop services defined in docker-compose.yml	docker-compose down
docker image prune	Remove unused images	docker image prune
docker system prune	Clean up unused containers, images, etc.	docker system prune
docker tag	Tag an image	docker tag source_image new_image_name