

Useful Docker commands

Stop all Docker containers

To stop all Docker containers from the command line, you can use the following command:

```
docker stop $(docker ps -q)
```

Explanation:

- **docker ps -q**: This command lists the container IDs of all running containers.
- **docker stop**: This stops the containers provided as arguments.

So, **\$(docker ps -q)** will pass all running container IDs to docker stop, effectively stopping all of them.

Remove ALL Docker containers Networks

WARNING (USED / UNUSED Networks) This will remove all Networks from ALL container even the one you use.

```
docker network rm $(docker network ls -q)
```

Explanation:

- **docker network ls -q**: Lists all network IDs (including those that are in use).
- **docker network rm**: Removes networks by their IDs.

This will attempt to remove all networks, including the default ones if they are not in use by containers. *However, if a network is in use by a container, Docker won't allow you to remove it without first disconnecting the containers.*

If you really want to remove all networks including the ones in use, you'd have to stop and remove the containers using those networks first.

Remove all Docker volumes (including unused ones)

To remove all Docker volumes (including unused ones), you can use the following command:

```
docker volume prune -f -a
```

Explanation:

- **docker volume prune**: Removes all unused volumes (volumes that are not currently associated with any container).
- **-f or --force**: This flag forces the operation without asking for confirmation.
- **-a or --all**: Remove all unused volumes, not just anonymous ones

If you want to remove all volumes — even the ones in use by containers (which is dangerous and typically not recommended) — you'd need to remove the containers and then delete the volumes.

Here's how you can do it:

1. Stop and remove all containers:

```
docker stop $(docker ps -q)
docker rm $(docker ps -aq)
```

2. Remove all volumes:

```
docker volume rm $(docker volume ls -q)
```

Caution:

- Removing volumes that are in use by containers can result in data loss, as volumes often contain persistent data for applications running in containers.
- Always double-check what volumes are being used and whether you need to keep any before removing them.

Remove all Docker images (including images that are not being used by any containers)

To remove all unused images (not in use by any container), including dangling images, you can run:

```
docker image prune -f -a
```

Explanation:

- **docker image prune**: This will remove all dangling images.
- **-f or --force**: This flag forces the operation without asking for confirmation.
- **-a or --all**: Remove all unused images, not just anonymous ones

To remove all Docker images (including images that are not being used by any containers), you can use the following command:

```
docker rmi $(docker images -q)
```

Explanation:

- **docker images -q**: Lists the IDs of all images on your system.
- **docker rmi**: Removes the images by their IDs.

If you have containers using some of these images:

If there are containers still using these images, *Docker will not let you remove them. To force removal*, you'd need to stop and remove the containers first.

Here's how to do it:

1. Stop all running containers:

```
docker stop $(docker ps -q)
```

2. Remove all containers:

```
docker rm $(docker ps -aq)
```

3. Remove all images:

```
docker rmi $(docker images -q)
```

Caution:

- Removing all images will mean you lose access to those images unless you pull them again, and you may need to rebuild any images you were working on.
- Double-check before running these commands if you're in a production environment, as this can disrupt your workflows!

Remove all stacks of containers in Docker

To remove all stacks of containers in Docker, you'll typically be dealing with Docker Swarm. A stack in Docker refers to a collection of services deployed in a Docker Swarm, often using `docker stack deploy`.

If you want to remove all stacks (and the services, networks, and containers associated with them), you can do so with the following command:

```
docker stack rm $(docker stack ls -q)
```

Explanation:

- **docker stack ls -q**: Lists the names of all deployed stacks.
- **docker stack rm**: Removes the stack by name, along with the services and containers related to it.

This will remove all stacks deployed in your Docker Swarm environment.

Important Notes:

- This command removes the entire stack — containers, networks, services, and volumes that are part of the stack will be removed.
- If you only want to remove specific stacks, replace `$(docker stack ls -q)` with the name of the stack you want to remove.

InsOmniA

Revision #4

Created 2025-05-06 01:46:56 EEST by Green

Updated 2025-09-04 02:11:30 EEST by Green