# Docker cheat sheet 2.0

## Build Docker image

Usage:

**docker build [options] [context]**

Example:

```
docker build -f Dockerfile --build-arg API_URL=http://localhost:3333 --tag=myapp .
```

-f : Dockerfile to build

--build-arg : Arguments to pass as environment variables during the build process. Must be defined inside Dockerfile using the ARG directive.

--tag : Tag (name) of Docker image to build

[context] (dot) : build context to use for building Docker image (i.e. folder which contains assets to be copied).

## Run Docker image

Usage:

**docker run [options] [image] [command] [arg...]**

Example:

```
docker run --rm -d -p 8080:8080 -e HOST=0.0.0.0 -v my-volume:/myapp/data --name=myapp
myapp:latest
```

--rm : Delete container upon container exit / Daemon termination.

-d : Detached mode (container runs in the background)

-p : Port range mapping. <HOST_PORTS>:<CONTAINER_PORTS>

-e : Enviornment variable to inject. <KEY_>:<VALUE_>

-v : Mount docker volume / bind-mount path on local filesystem.
<HOST_PATH>:<CONTAINER_PATH>

--name : Name of the newly created container.

[image]:latest : image to be run at revision (latest)

# List running Docker containers

Usage:

**docker ps [arg...]**

Example:

```
docker ps -a
```

-a : Show all containers (including non-running).

# Stop (delete) Docker containers

Usage:

**docker stop [container] [arg...] docker rm [container] [arg...]**

Example:

```
docker stop slek-client
docker rm slek-client
```

# Execute into Docker image

Usage:

**docker exec [options] [container] [command] [arg...]**

Example:

```
docker exec -it myapp /bin/bash
```

-it : Interactive mode (useful for opening interactive command prompt with shell, e.g. bash)

[command] : command to run inside the container. In this case, we open interactive bash command prompt.

# Alter Docker container's state

Usage:

**docker restart / start / stop / pause [options] [container]**

Example:

```
docker restart myapp
```

stop : Stop container using SIGKILL signal.

pause : Stop container using SIGSTOP signal.

# Clean-up Docker

Usage:

**docker system / container / image / volume prune [options]**

Example:

```
docker system prune
```

system : Delete all stopped containers, dangling / unreferenced images, unused networks, optionally unmounted volumes (--volumes argument).

volune : Delete all volumes not attached to a container.

image : Delete all dangling and unreferenced images.

system prune -a : Deletes everything including unused Docker images (non-dangling).

# Docker-compose

Usage:

**docker-compose [options] [command] [arg...]**

Example:

```
docker-compose -f my-app-compose.yml up / down
```

-f : Docker Compose YAML file to use (default: docker-compose.yml).

# Create Docker volume

Usage:

**docker volume create [volume] [arg...]**

Example:

```
docker volume create vol-slek-server
```

# List (delete / prune) Docker containers / images / volumes

Usage:

**docker container (image / volume / network) ls (rm) [container (image / volume / network)] [arg...]**

Example:

```
docker image ls
docker image rm slek-client-image
docker container prune
```

prune : Deletes all unused (stopped) containers (images / volumes / networks)

---

Revision #10
Created Sat, Nov 19, 2022 1:20 PM by Adam Puskas
Updated Sat, Nov 19, 2022 12:20 PM by Adam Puskas