DOCKER

CHEET SHEET BY @CHARLSTOWN

CREATOR: SOLOMON HYKES & SEBASTIEN PAHL
YEAR: 2013
STEABLE RELEASE (2020): 19.03
TOP DOCKER HUB: ubuntu, redis, node, postgres, traefik, mongo, alpine, nginx, hello-world, python

THE DOCKERFILE — Creating a container

FROM image	> Sets the base image
MAINTAINER name	> Sets the Author
RUN apt install x	> Executes any cmnd for the image
CMD python app.py	> Default execution when container is up
ENTRYPOINT ...	> Run as executable
ENV key value	> Sets an environment variable to be passed
COPY /src /dest	> Copy files inside container
ADD url /dest	> Copy files from url inside container
VOLUME /path	> External mount point from container
WORKDIR path/to	> Sets the working dir inside container
USER user	> Sets username
ARG name=value	> Define a var to be passed at build time
ONBUILD instruction	> Executes inst. when img used with FROM

BUILDING IMAGES [*] = optional

docker build [opts] path > New docker image
E.g. **$docker build -t app ./main**

-t /--tag image_name	> Image name
-f / --file path	> Path to Dockerfile
--build-arg var=value	> Build an argument
--label namespace var	> Sets metadata
-q /--quite	> Supress the output
--rm	> delete intermediate container

MANAGING CONTAINERS [*] = optional

docker run [opts] img [command]
>> Creates and run a new container
E.g. **$docker run -it --name app img_name**

-i / --interactive	> stdin is always open
-t / --tty	> tty is active (enables the terminal)
--name cont_name	> Assign a name to the container
-v / --volume host:cont	> Mount a volume
-d / --detach	> Run container in the background
-e, --env var=value	> Sets env variable
--env-file file_name	> Read a file with vars
-h, --hostname="name	> Cont. host name
--add-host=host:ip	> Add host to ip map.
--rm	> Remove when exits

docker create [opts] img cmd
>> Create a new stopped container, same opts

docker start [opts] container
>> Start one or more existant containers
E.g. **$docker start container_name**

-a, --attach	> attach stdout
-i, --interactive	> attach stdin

docker stop [opts] container
>> Stop one or more existant containers

-t, --time	> count down to stop

docker kill [opts] container
>> Kill one or more running containers

RUNNING CONTAINERS [*] = optional

docker exec [opts] container command
>> Run a process in a running container
E.g. **$docker exec -it container bin/bash**

-i, --interactive	> stdin is always open
-t / --tty	> tty is active
-d / --detach	> Run detached

docker cp container:path host_path
>> Copy files from container to the outside
E.g. **$docker cp cont:/data ./files**

docker logs container **> Show all logs**

LISTING IMAGES & CONT. [*] = optional

docker images [opts] >Show all images

-a, --all	> Show all
-q, --quite	> Show only IDs
--no-trunc	> Full output

docker ps [opts] >Show all images
*Same as before: -a /-q /--no trunc

-l, --latest	> Show last container

REMOVING IMAGES & CONT. [*] = optional

docker rm [opts] container
>> Remove one or more containers

-f, --force	> Show all
-v, --volume	> Show only IDs

docker rmi [opts] >Remove images
*Same as before: -f