**DOCKER**

BASICS:

**List of the most commonly used docker commands:**

docker ps

docker images (ls – shoes images on the local machine)

docker logs

docker inspect

dcoker stop

docker rm (-- force removes running container)

docker rmi (removes image)

docker pull

docker build (-t)

docker run (-rm, -t , -it, -p, --restart, -v, -e, --link) create + start

docker login (-u, -p)

docker push

docker stats

docker exec

docke commit

docker volume

docker kill

docker network ls

docker container/volume/image prune -f - removes all stopped containers

All the commands can be found here:

https://docs.docker.com/engine/reference/commandline/docker/

**Basic objectives:**

Image – template for a container

Container – what we run and host – isolated machine

Registries – use for storing various images, storing tags for the same image, HTTP API to pull and push images, TSL-secured connection to the API)

**Basic topics:**

* DockerFile and its instructions (FROM, WORKDIR, ENV, COPY, RUN, ENRTYPOINT)

https://docs.docker.com/engine/reference/builder/

* Tagging
* Private registries and reducing their size (by base image and layers)
* Detached and attached mode container running
* Long-lived containers - server containers (listens for incoming network connection)
* Stateless containers
* Bind mount (use host filesystem and mount on the container)
* Volumes - mapping directory inside the container to a persistent storage (managed through drivers and depend on actual Docker host) used mostly for sharing data in different containers

(docker run -v /your/dir:/var/lib/mysql -d mysql:5.7)

* Docker networks

**Advanced topics:**

**Docker - Compose**: tool for running multi-container application

* Compose file in in YAML format looks like:

version : version of the Compose file

services: all the containers that should be created

name of the container:

image: specifies image location of there is no Dockerfile

build : specifies Docker file location

ports: maps container ports to the host

volumes: attaches code files directory to the containers

links: links one service to another

environment: env variables that need to be set up

* Commands:

docker-compose build

docker-compose images

docker-compose run

docker-compose up build + run

docker-compose stop

docker-compose rm

docker-compose start

docker-compose restart

docker-compose ps

docker-compose down (like docker system prune but it stopped all the services and then cleans up the containers, networks, images used and created by the compose file)

docker-compose logs

* Multiple Dockerfiles :

In case there are multiple docker files then we should specify:

context: specifies directory of Dockerfile or alternate DockerFile relative to docker-compose.yaml

dockerfile: name of the Dockerfile

depends\_on: informs docker-compose about all the dependencies