

INFOSÜSTEEMIDE ARENDAMINE III - HAJUSRAKENDUSED

Loeng 5 – Docker

Tarvo Treier

Tarkvarateaduse instituut

02.10.2023

ERINEVAD TEADAANDED

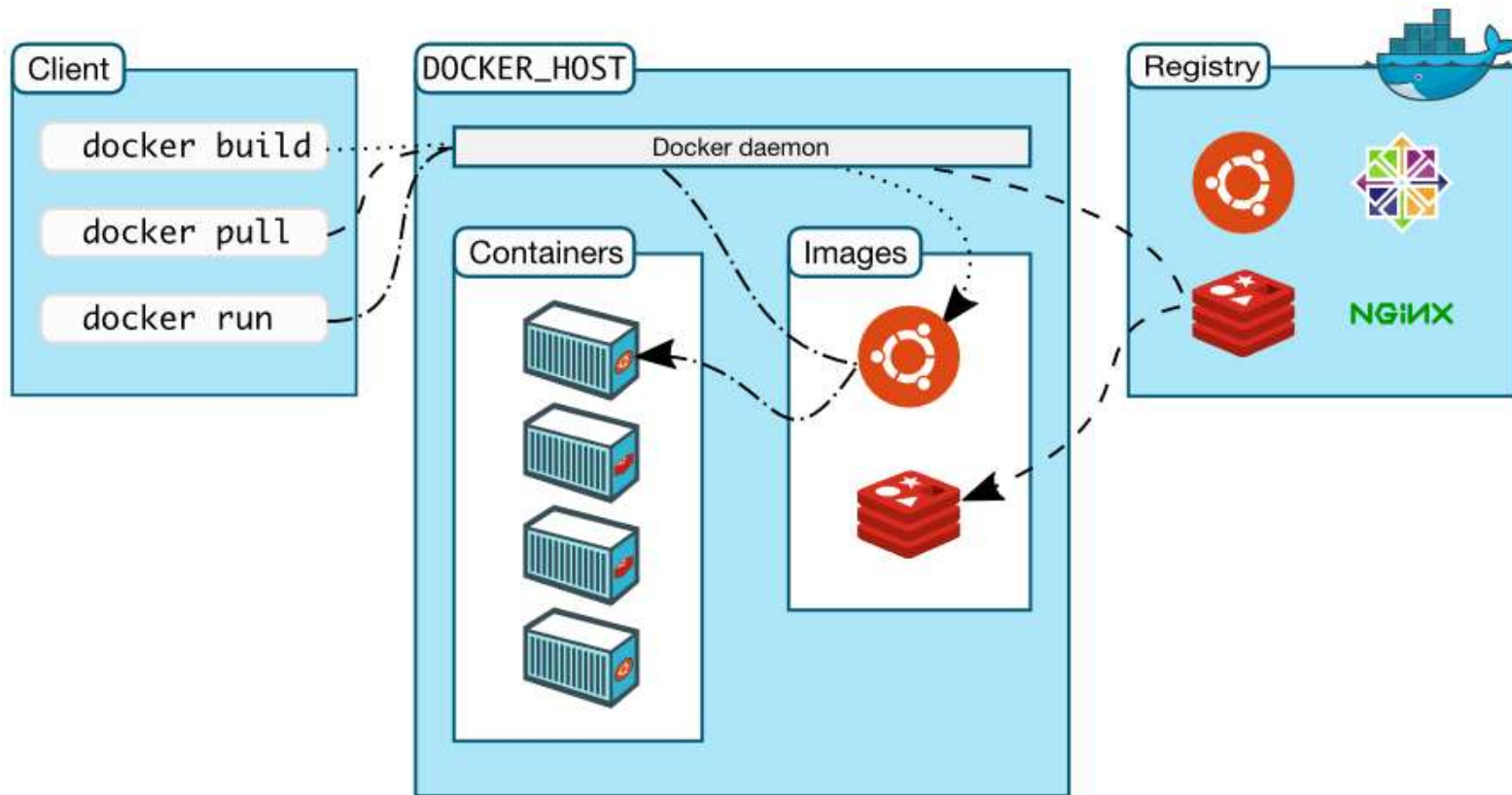
- **Järgmine nädal (6.) kõik kohtumised ülikoolis kohapeal!**
- Meeskonnaprojekti esimese sprint käib!
- Moodles 2. iseseisev kodutöö.
- Pane võimalusel **docker** enda arvutisse.
- Küsige julgelt abi, kui vajate!

DOCKER

- Docker on arvutiprogramm, mis teostab operatsioonisüsteemi tasemel virtualiseerimist ehk konteineriseerimist.
- See avaldati 2013. aastal ning seda arendab Docker, Inc. Ettevõtte asub San Franciscos California osariigis USA-s. Dockeri tarkvara looja on Solomon Hykes.

Viide: <https://et.wikipedia.org/wiki/Docker>

DOCKER-I ARHITEKTUUR



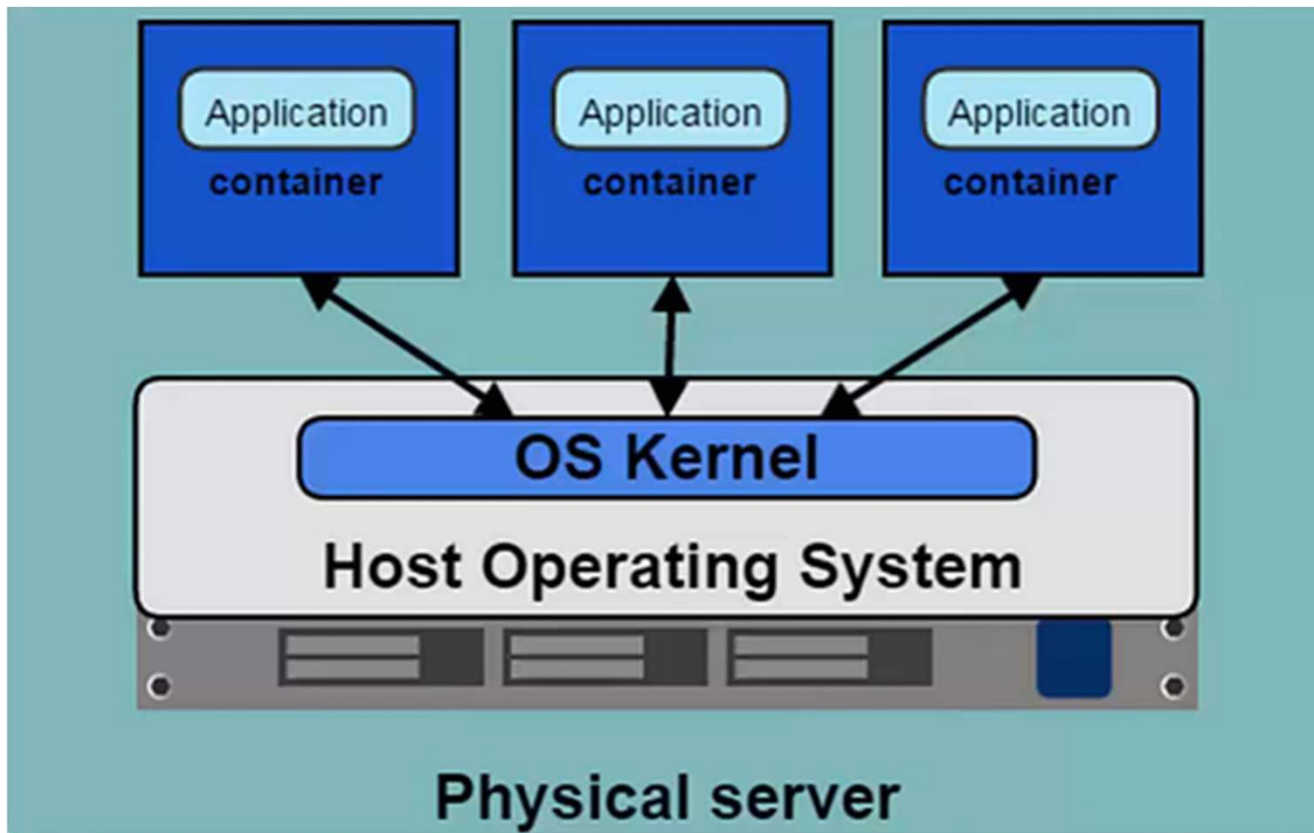
DOCKER-I KOMPONENDID

- Docker daemon – Dockeri server, mis haldab Dockeri objekte nagu *images*, *containers*, *networks* ja *volumes*
- Docker klient – vahend dockerile käskude edastamiseks nagu *build*, *run*
- Docker register – selles hoitakse Docker *images*. Docker hub on avalik register, kuskohast docker otsib *images*
- Docker klient ja daemon kasutavad suhtlemiseks REST API-t.

DOCKER-I OBJEKTID

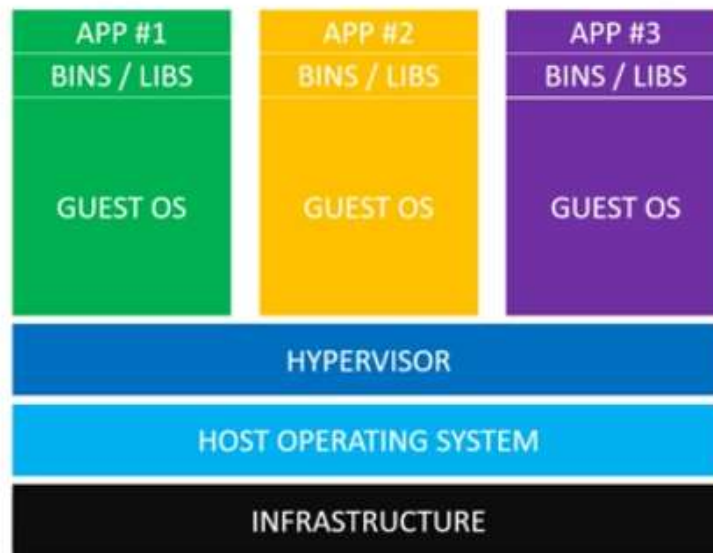
- Images – *read-only* mall koos instruktsioonidega Dockeri konteineri loomiseks. Sageli *images* baseerub mõnel teisel *images-l*
 - Dockerfile – sammud/käsud enda *image* loomiseks. Iga käsk teeb enda kihi ja kui käske muudetakse, siis *rebuild* alates kihist, mis muutus (kiirem, väiksem, õhem)
- Containers – *runnable instane of an image*. Saab luua, käivitada, peatada, kolida, kustutada
- dockerfile -> (build) -> image -> (run) -> container

DOCKER-I KONTEINERID KASUTAVAD ÜHTE KERNELIT



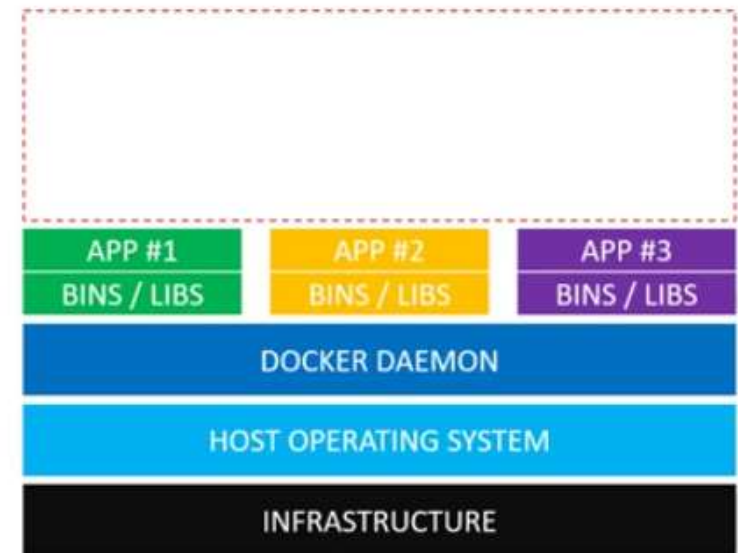
VIRTUAALMASINAD VS DOCKERI KONTEINERID

Isolate systems



Virtual Machines

Isolate applications ⁱ



Docker Containers

DOCKERI EELISED VÕRRELDES VIRTUAALMASINAGA

- Kasutab ära host OS-i kernelit ja kasutab seetõttu vähem ressursse
 - Kettamaht, mälu..
- Isoleerib rakendused, mitte kogu süsteemi
- Käivitub millisekundite, mitte minutitega
- Parema jõudlusega
- Lihtne skaleerida
- Andmeid on võimalik jagada erinevate konteinerite vahel
- Pidevintegratsiooni võimalus

VIRUAALMASINAD VS DOCKER – MAJA VS KORTER

DOCKER – NR1 VALIK MIKROTEENUSTE KORRAL, SEST

- see aitab erinevate teenuste arendusmeeskondadel kiiresti kogu vajalikku keskkonda tööle panna enda arenduskeskkonnas
- kiirendab toodangus versioonivahetuse protsessi ja aitab seda automatiseerida
- ühe serveri kõiki mikroteenuseid saab vajadusel ühe korraga lihtsalt tõsta teise serverisse
- kasutab paremini ära serveri ressursse (odavam)
- lubab ühel serveril asuvatel teenustel kasutada samadest komponentidest erinevaid versioone ilma konfliktideta

LEGACY CODE

- mingi tarkvara kood, mis on seotud mõne aegunud tehnoloogiaga
- negatiivne kuvand, aga tegelikult vastab kliendi nõudmistele
- Docker ei pruugi olla *legacy code*-i puhul esimene valik, aga kindlasti valik, mida kaaluda.

MILLAL KASUTADA DOCKERIT JA MILLAL MITTE



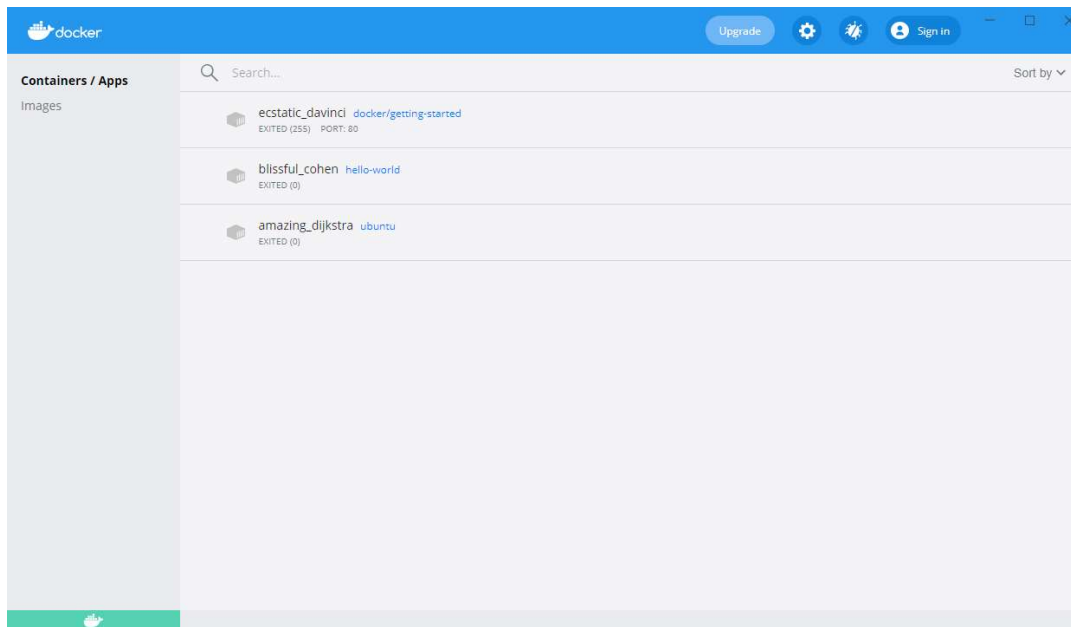
Viide: <https://www.linode.com/docs/applications/containers/when-and-why-to-use-docker/>

DOCKERI INSTALLEERIMINE

- <https://docs.docker.com/install/>
 - <https://hub.docker.com/editions/community/docker-ce-desktop-windows>
- *bios-s* peab virtualiseerimise lubama
- *PowerShelliga* saab käske anda
- *docker run hello-world*
- *docker ps*

DOCKERI KONTEINER TÖÖTAB AINULT SIIS, KUI SEAL SEES MÕNI RAKENDUS TÖÖTAB

- *docker run ubuntu*
- *docker ps*
- *docker ps -a*



DOCKER-I PEAMISED KÄSUD

- <https://www.edureka.co/blog/docker-commands/>

DOCKER-COMPOSE

- <https://docs.docker.com/compose/>
- <https://devhints.io/docker-compose>

DOCKER-I KASUTAMINE MEESKONNAPROJEKTIS

ERINEVAD LAHENDUSE KOMPONDID, MIDA DOCKERISSE VIIA

- paneme ainult *postgresi* andmebaasi *dockerisse* (Miks?)
docker run -d -p 5432:5432 --name my-db -e POSTGRES_PASSWORD=parool postgres
- paneme lisaks baasile *pgAdmin-i* *dockerisse* (Miks?)
docker compose up
volumes
networks (pg_network)
- paneme lisaks eelnevale API-i *dockerisse* (Miks?)
dockerfile, docker compose (eraldi esimesest)
external networks (pg_network)
#postgres ports
- paneme lisaks eelnevale *node-i* ja *Vue-i* *dockerisse* (Miks?)
dockerfile, docker compose (eraldi)
external networks (api_network)
#api ports

DOCKER-COMPOSE.YML NÄIDE

VALIK DOCKER-COMPOSE-IGA SEOTUD KÄSKE

Docker compose up

Docker compose down -v

Docker compose down --volumes

docker volume ls

docker volume rm XXX_postgres-data