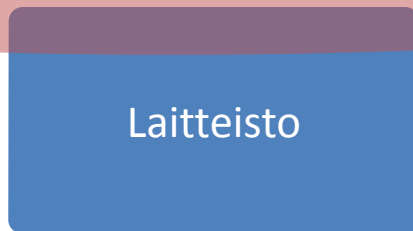


Docker - kevyempää virtualisointia

Jussi Talaskivi

Jyväskylän yliopisto

Docker ei ole virtualisointia!



Mikä on Docker?

- Docker on teknologia rakentaa, jaella ja ajaa sovelluksia
- Sovellus rakennetaan Dockerfile kuvausella
- Sovellukset jaellaan sovelluskuvina (image)
- Sovellukset ajetaan konttissa (container).
- Docker itse on asiakas-palvelin ohjelmisto, joka on kirjoitettu Go-kielellä.

Rakentaminen: Dockerfile

```
FROM centos:centos7
MAINTAINER Jussi Talaskivi <jussi.talaskivi@jyu.fi>

RUN rpm -Uhv http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-2.noarch.rpm
RUN yum install -y nodejs npm make msgpack-devel zeromq3-devel
RUN mkdir /app
WORKDIR /app
RUN npm install zerorpc
ADD . /app
CMD node server.js
EXPOSE 4242
```

```
$ docker build -t oma-image hakemisto
```

Jakelu: Hub tai oma rekisteri

- <https://hub.docker.com> - Docker Inc tarjoama keskitetty palvelu imagejen jakelemiseen.
 - Julkiset repositoryt ovat ilmaisia. Yksityisistä pitää maksaa (vrt. GitHub)
- <https://github.com/docker/docker-registry> - Avoimen lähdekoodin rekisteri
- Jos käytetään omaa rekisteriä lisäätään imagen nimeen rekisterin osoite alkuun.
 - Esim. docker.oma.domain/oma-image

```
$ docker push oma-image
```

Ajaminen: Käyttöjärjestelmä

- Vaatii kohtalaisen tuoreen Linuxin.
 - RHEL/CENTOS 6 tai 7
 - Ubuntu 12.04 tai uudempi
- Windows tai OS X alustalla voidaan käyttää Boot2Dockeria.
 - VirtualBoxissa ajettava TinyCoreLinux jossa Docker palvelin.
 - Client toimii natiivisti

```
$ docker run --name=oma oma-image
```

Dockerin ominaisuuksia

- Prosessien ajo Linux containereissa
- Imagen perintä toisesta
- Tiedostojärjestelmän abstrahointi kerroksiksi
- Containeroiden verkkoyhteyksien hallinta
- Datat jakaminen hostista containeriin tai niiden välillä

Meidän Docker ympäristö

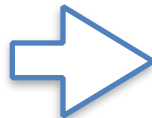
Versionhallinta

- RhodeCode (Mercurial)
- Laukaisee Jenkins työn



Jenkins

- Imagen rakennus
- Imagen rekisteriin tallennus
- Puppet Hiera



Tuotantopalvelin

- Puppet



Docker rekisteri

- Tagatyt imaget (2014-11-04.3)

Docker tuotantoympäristössä

- Sama image sekä kehitysympäristössä että tuotannossa
- Konfiguraatio ympäristömuuttujien kautta
- Yksi prosessi / container
- Linkitetään containereita
- Pyritään tekemään containereista tilattomia
 - Dataa ei haluta tallentaa sovelluksen kanssa samaan containeriin
 - Data joko hostin puolelle tai omaan containeriinsa (jossa ei olekaan sitten juuri muuta)

Kokemuksia Dockerista

- Yksinkertaistaa palvelinasennuksia merkittävästi
 - Tarvitsee kirjaston x version y. Ei ole enää ongelma.
- Pienempi kynnyks ottaa uutta teknologia käyttöön
- Mahdollistaa helpomman continuous deliveryn
 - Sama image sekä kehityksessä että tuotannossa
- Syntyy helposti roskaa (imageja, pysähtyneitä containereita)

Dockerin puutteita

- Useamman containerin orkestrointi
 - Tulossa orkestrointi API
 - [Fig.sh](#) integroidaan Dockeriin
 - Kubernetes, Mesos, OpenStack
- Toimii vain Linuxilla
 - Docker API on geneerinen
 - [Microsoft tukee Dockeria](#)

Kiitos!

Kysymyksiä?