



docker

# på tredje sal...

Thorbjørn Ravn Andersen - [tra@kb.dk](mailto:tra@kb.dk)

# Arkitekturgruppen....

- Har i foråret diskuteret hvordan vi tæmmer (især) integrationstestning.
- *(Nogen) nøgleord:* Reproducerbarhed, brugbart fra kommandolinien og CI-server, velunderstøttet "out there". Osv osv.
- Konklusionen var at eneste fornuftige teknologibud på nuværende tidspunkt ser ud til at være "docker".

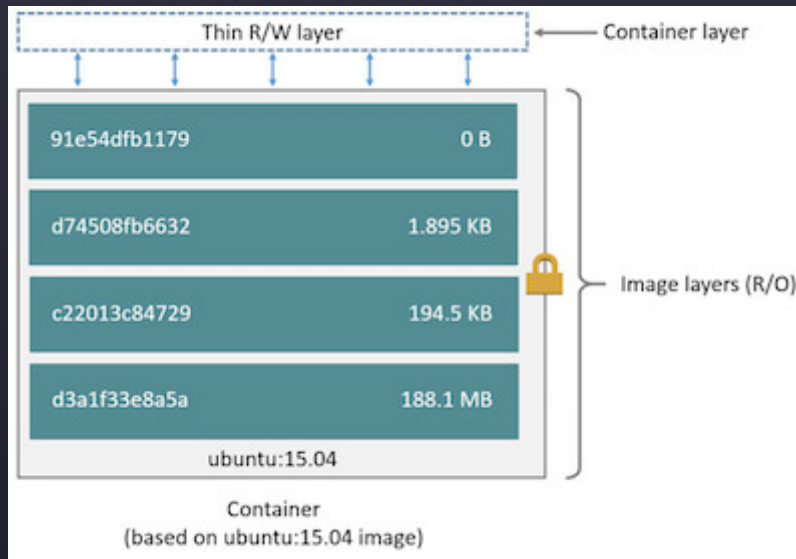
Hvad *kendetegner*  
docker?

# Docker tillader at køre en "gæst" i en Linux container på en "vært"

Dette er i modsætning til en fuld virtuel maskine som fx VirtualBox/vmware.

Helt præcist er Docker et API ovenpå et emuleringslag - LXC - der har været en del af Linux længe. "Container"-tankegangen er ikke ny.

# Unionfs - flere lag samles



R/W layer nulstilles når gæsten stopper. Filer kan persisteres på flere måder afhængigt af behov.

<https://docs.docker.com/engine/userguide/storagedrivers/and-layers>

# Hurtig opstart på de rå filer.

TRA har målt tiden fra start af docker til JVM begynder at køre bytekode til 0,4 sekunder.

# DEMO: nginx på redigerbar værtsfolder

# Masser af software lettilgængelig.

Det er ret nemt selv at bygge diskimages og gøre tilgængeligt for andre på fx Docker Hub, eller internt på tredie.

Eksempler på officielle images: Tomcat, PostgreSQL, mysql, Jetty, Maven, logstash, httpd, wordpress, centos, ubuntu, python, kibana, jenkins, memcached, solr, ibmjava...

<https://hub.docker.com/explore/>



Docker Engine er den samlede pakke. Er lige i 2017 splittet i CE og EE.

Klientdelen kører på Linux, MacOS og Windows. Ubuntu 17.04 har "nye nok" pakker.

Gæster kan afvikles på:

- Lokalt: Linux, Mac (evt via VirtualBox), Windows (evt via VirtualBox)
- Amazon EC2
- Google Compute Engine
- Microsoft Azure
- Rackspace
- Triton (giver mulighed for DTrace!)

og mange flere der implementerer Docker API'et.

# En gæst, én opgave

Docker-mentaliteten er at hver gæst har netop en opgave, og at flere gæster snakker sammen via netværk.

**docker-compose** orkestrerer dette!

Kan køres fra Linux command line. Asger har fået Bitmagasinet til at køre på denne måde.

# Maven build

Svært at gøre ensartet.

- Maven skal installeres manuelt. De pakkede udgaver er altid for gamle.
- Sippet med JAVA\_HOME
- Subtile forskelle mellem versioner.
- Selv med maven kan der være stygge afhængigheder der skal være på plads.

# Maven build

Indledende observationer:

- "maven:3.3.9-alpine" image kan bruges direkte.
- Tomt lokalt repository: Henter det halve internet hver gang :-(
- Starter hurtigt - sammenlignelig køretid.
- Kan startes fra kommandolinien. Kan bruges som **mvn** direkte.

# Maven build

Beskidte tricks:

- importer læsebeskyttet .git folder fra vært som kloner ud
- importer læsebeskyttet .m2/repository folder fra vært som filbaseret remote maven repository.
- lokal nexus er nok ikke nødvendig.

(`mvn dependency:go-offline` er godt)

\$HOME/.m2/settings.xml:

```
<settings>
  <mirrors>
    <mirror>
      <id>external-m2</id>
      <name>external m2 folder</name>
      <url>file:///root/external-m2/repository</url>
      <mirrorOf>external:*</mirrorOf>
    </mirror>
  </mirrors>
</settings>
```

**DEMO: Maven build**



# Status

Der har været skævet til docker længe:

- abr har bygget rigtige ting med det som alternativ til vagrant.
- nkhs er i gang med at lave introduktionsmateriale til det
- mmj har snakket om det længe.

Det er Arkitekturgruppens mål at lave et oplæg "inden længe" hvor vi gerne skulle kunne demonstrere at alle på tredie kan have *umiddelbar* fordele i det daglige ved at bruge den her teknologi.