

Introduktion till Docker

Och kontainrar i allmänhet

Simon Rydell

June 18, 2019

Kontainer?

- "Kappsla in" processer
- Garanterar samma miljö för ett program



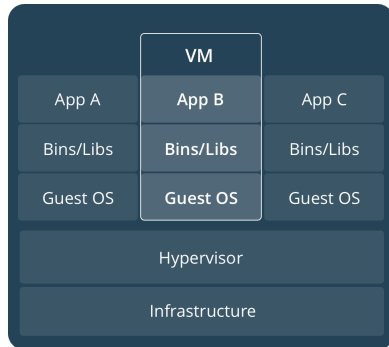
github.com/srydell/docker-intro

1. Virtual machine vs container
2. Använda Docker
3. Exempel

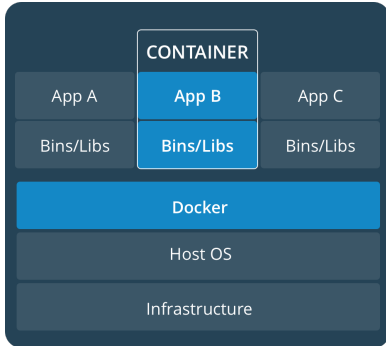
Virtual machine vs container

Virtual Machine (VM)

- Virtualisera allt!
(Nästan)
- Tunga processer
- Som en dator



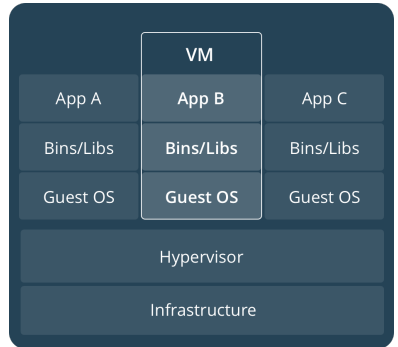
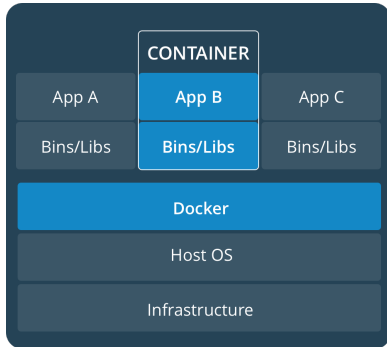
Container - Docker



- Virtualisera det som behövs
- "Lättviktiga" processer
- Som en samling processer

github.com/srydell/docker-intro

Virtual machine vs container

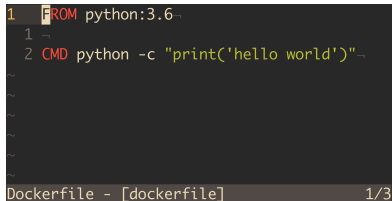


github.com/srydell/docker-intro

Använda Docker

Dockerfile

- Som okompilerad kod - måste byggas för att köras
- Om du kan göra det via terminal kan du göra det genom Docker



```
1 FROM python:3.6
2 CMD python -c "print('hello world')"
```

Dockerfile - [dockerfile] 1/3

Docker registry - Dockerhub

- Som ett git server där varje repository är Dockerfiles (eller images)
- Används oftast direkt från *FROM* i en Dockerfile

Exempel

Exempel

- Bygga en Dockerfile
 - Command line intro
- Jenkins
 - Exponera portar
 - Daemons
 - Volumes

github.com/srydell/docker-intro

Vad jag inte tog upp

- Network - Hur kontainrar ser varandras nätverk
- Logging - Hur samlar man fel/loggar på ett effektivt sätt
- Orchestration - Hur man kopplar ihop kontainrar

Frågor