

IaaS, PaaS & SaaS (hint: Nist) – infrastructure as a service, Platform as a service, Software as service

- OS-level virtualization – virtualiserar inte kärnan
- cgroups – funktion i linux kärnan som begränsar cpu användning
- Copy-on-Write (COW) & Snapshots – copy on write: data modifieras inte, utan förändringar sparas istället. snapshots: punkter i ett copy on write system dit man kan gå tillbaka.
- High-Availability – en miljö går inte ner bara för att en maskin går sönder.
- Idempotency – en funktion ger alltid samma resultat
- Mutable vs Immutable Infrastructure – mutable är att server infrastructure kan ändras efter man driftsatt den. immutable är att man inte kan ändra nåt
- Configuration Management vs. Orchestration – config mgmt är att installera paket. orchestration kan partitionera diskar.
- Procedural vs Declarative – procedural betyder datorn vet vad den ska göra, declarative betyder att man berättar för datorn vilket problem man vill datorn ska lösa
- Git Submodule – betyder att man kan lägga till andra git arkiv under ett git arkiv och hålla commits åtskilda
- Ansible – definierar hosts och grupper av hosts på vilka kommandon, moduler och uppgifter i en spellista fungerar. Filen kan vara i ett av många format beroende på din Ansible-miljö och plugins.
- Inventory File – definierar hosts och grupper av hosts på vilka kommandon, moduler och uppgifter i en spellista fungerar. Filen kan vara i ett av många format beroende på din Ansible-miljö och plugins.
- Playbook – beskriver allt som Ansible ska göra med skripting
- Kubernetes – en verktyg för virtualiserad klustring
- Compare Stateful and Stateless Applications – stateful sparar data. stateless arbetar bara med data som kan återskapas
- ReplicaSets, Deployments, Pods & Services – replicasets ser till att ett visst antal podreplikationer körs vid en viss tidpunkt. deployments är deklarativa uppdateringar av pods och replica sets och de ersätter automatiskt instanser som misslyckas eller inte svarar. Pods är en samling containrar samma värd. En service är en grupp av poddar som körs på ett kluster.

Why might you want to make a multi-stage build ?

Med en multistage build kan man skapa en docker image som innehåller resultatet av tidigare operationer. Man kan till exempel kompilera kod med Go i en Go kontainer och sedan driftsätta den i en ubuntu kontainer. Då slängs föregående kontainer.

Mounting database dumps like this can be a good solution to use for a test database, but why might you not want to do it like this for a production

database?

För databasens innehåll förändras och det som läggs in i imagen är det som kommer komma med när man driftsätter kontainern. Om kontainern går ner kommer man förlora datan i den. Istället bör man sätta upp en kontainer och en volym till den där databasen spara data. Sedan bör man ta regelbundna backuper.