

Docker in kontejnerji



Davor Gutierrez

3 GEN d.o.o.

davor.gutierrez@3gen.si

Agenda



- Kaj so kontejnerji?
- Kako delujejo?
- Kaj pa prenosljivost?
- Kaj je virtualizacija?
- Kaj je Docker?
- Dockerjev pristop

Kaj so kontejnerji?



- Zelo stara zadeva
- Imajo jih vsi UNIX »like« OS
- FreeBSD ima Jail
- Solaris ima Zone
- Linux ima LXC in še kar nekaj drugih implementacij
 - OpenVZ
 - VServer

Kako delujejo?



- Kontejnerji vseh oblik izolirajo in enkapsulirajo aplikacijski nivo od gostujočega sistema
- Kontejner si lahko predstavljate kot OS v gostujočem OS-u, v katerega lahko nameščate in poganjate aplikacije in se v vseh pogledih obnaša kot virtualna mašina
- To emulacijo omogoča Linux kernel in LXC projekt, ki nam prinaša tudi predloge različnih gostov in uporabniške vmesnike za celoten ti. »Life Cycle Management«

Kaj pa prenosljivost?



- Kontejnerji dobesedno »odklopijo« aplikacijo od gostujočega sistema in jo s tem naredijo prenosljivo med vsemi sistemi, ki uporabljajo določeno vrsto kontejnerjev
- Uporabnik ima lahko tako nameščen minimalen »host« gostujoč OS. LAMP in e-pošto pa poganja v kontejnerju
- Aplikacije so izolirane in uporabniki lahko uporabljajo različne verzije PHP, Apache ipd., s čimer pridobimo fleksibilnost instanc v oblaku, katere lahko enostavno in hitro kloniramo, selimo in arhiviramo

Kaj pa virtualizacija?



- A ne omogoča vsega navedenega tudi virtualizacija?
- Da, ampak na račun »performans« in »fleksibilnosti«
- Kontejnerji ne emulirajo HW nivoja in uporabljajo ti. cgrupe in namespace Linux kernela, da kreirajo ti. »lightweight« virtualizacijsko okolje OS s skoraj »bare metal« hitrostjo
- To nam v osnovi povsem spremeni način virtualizacije strežnikov in aplikacij, saj so kontejnerji enostavno mnogo hitrejši

... kaj pa virtualizacija? (2)



- Torej virtualizacija izumira?
 - ne, v kolikor potrebujemo (ne)Linux OS, bomo še naprej uporabljali virtualizacijo

Kaj je Docker?



- Projekt, narejen leta 2013 s strani dotCloud-a
- Sprva baziran na LXC
- Za kreiranje posamičnih »single« aplikacijskih kontejnerjev
- Sedaj ne uporablja več LXC knjižnic, ampak libcontainer knjižnice, ki tesno sodelujejo s kernelom Linux OS
- Docker nam je predstavil različne koncepte uporabe in pristope do nam že znanih kontejnerjev

Dockerjev pristop



- Miselnost »container as an app«
- Terminologija, ki si jo velja zapomnit:
 - Layered containers
 - Single Application Containers
 - Separate States
 - Registry

Layered containers



- Izredno kompleksna zadeva za navadne smrtnike :)
- Docker uporablja AUFS/devicamp/ **btrfs** za izgradnjo kontejnerjev
- Docker kontejnerji so RO nivoji FS
- Docker kontejner je RO in samo vrhnji nivo je RW, vrhnji nivo ima prehodne podatke, ki se zapišejo ob potrditvi »commit«

Single Application Containers



- Docker kontejner je omejen na en proces
- Ni dizajniran za »multiapp«, procese ali servise kot so init, cron, syslog ali ssh
- Ko se prvič srečamo z Dockerjem, se nam podre celoten koncept delovanja strežnika, kot smo ga imeli do sedaj
- Primer: želimo postaviti LAMP – potrebujemo 3 kontejnerje
- Lahko bi ga postavili v en kontejner, ampak s pomočjo shell skript in proces managerjev, vendar s tem Docker izgubi smisel

Separate states



- Docker loči podatkovni del od aplikacije
- Ti. »persistent data« vzdržujemo izven Docker kontejnerja v ti. »data volume only« kontejnerjih

Registry



- Docker omogoča ti. javne in privatne registre slik
- Iz le-teh lahko delamo »pull« in »push« slik
- Slike so RO nivoji, s katerimi zaganjamo aplikacijo
- S pomočjo registra enostavno vzdržujemo in delimo različne aplikacije med uporabniki
- Microsoft je pred časom objavil, da bo Docker kaj kmalu dosegljiv tudi v MS svetu

DEMO



- Primer uporabe ...