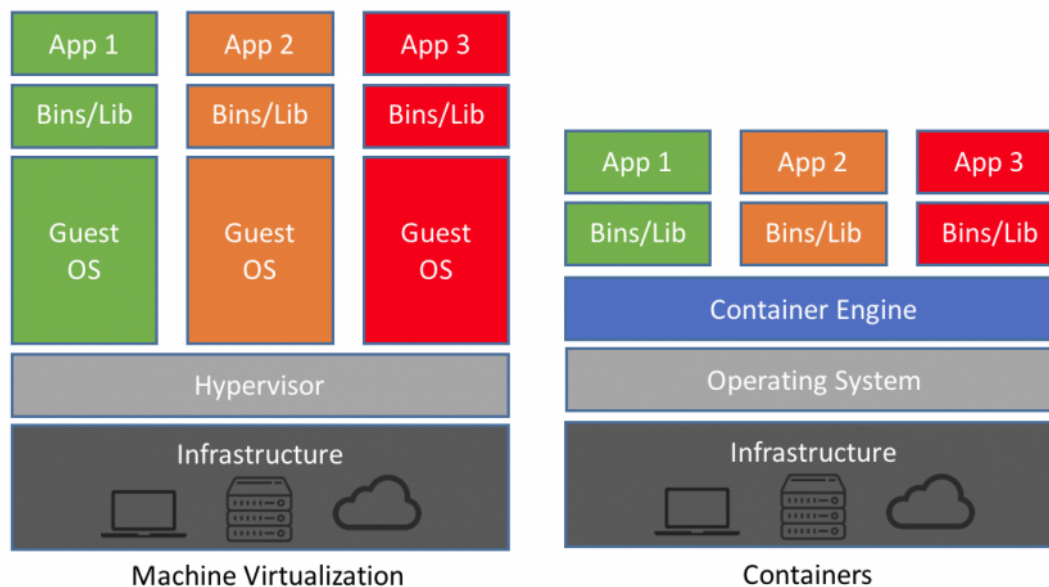


Singularity

- Uvod
 - Singularity
- Izgradnja kontejnera
 - Singularity recepti
 - Header
 - Sections
 - %help
 - %setup
 - %files
 - %labels
 - %environment
 - %post
 - %runscript
- Korištenje kontejnera
 - Podnošenje poslova na klasteru

Uvod

Kontejner je standardna jedinica softvera koja pakira kod i sve njegove ovisnosti tako da se aplikacija može brzo i pouzdano pokretati u različitim okruženjima. Oni "sjede" na fizičkom serveru i njegovom operacijskom sustavu i s njim dijele kernel. Slični su virtualnim strojevima (eng. *virtual machines*) utoliko što su također virtualizirani sustavi. Međutim, za razliku od virtualnih strojeva koji sadrže cijele operacijske sustave i sve ovisnosti koje im trebaju za izvršavanje zadanih operacija (te su zbog toga zahtjevne za resurse), kontejneri s hostom dijele kernel, te u sebi sadržavaju samo aplikacije, njihove ovisnosti i potrebne biblioteke (eng. *library*).



Slika 1: Hipervizor sjedi na fizičkom stroju i omogućuje kreiranje virtualnih strojeva. Svaki virtualni stroj ima svoj operacijski sustav, i zauzima određene resurse potrebne za pokretanje odvojenih OS-a. S druge strane, kontejneri su povezani s operacijskim sustavom fizičkog računala te dijele njegov kernel i odvojene su samo aplikacije i njihove ovisnosti. Zbog toga kontejneri troše značajno manje resursa od virtualnih strojeva.

Singularity

Singularity je jedno od kontejnerskih rješenja. Glavna svojstva su mu:

- mobilnost
- reproducibilnost
- korisnička sloboda

- podrška za HPC

Mobilnost računanja omogućuje da se workflow definira u kontejneru te korisnik može biti siguran da će se on moći izvršavati na različitim strojevima, neovisno o operacijskom sustavu (dok god je Linux! Singularity se oslanja na Linux kernel te ne radi na drugim operacijskim sustavima.) i infrastrukturi. Mobilnost je osigurana time što se Singularity kontejneri spremaju u slike (eng. *image*) koje su jedna datoteka koja obuhvaća sve datoteke u kontejneru. Slike se mogu izgraditi kao *read-only* datoteke (tako se grade po defaultu) te je tako osigurana i reproducibilnost.

Korisnička sloboda se odnosi na mogućnost korisnika da unutar kontejnera instaliraju aplikacije s njima poželjnim verzijama i bibliotekama, neovisno o stanju na poslužitelju na kojem će taj kontejner kasnije pokretati. Dopremanjem kontejnera na bilo koji resurs, korisnici mogu koristiti aplikacije koje god žele.

Singularity je kompatibilan s HPC-om. Podržava InfiniBand, ima podršku za MPI, radi bez problema s većinom raspoređivača poslova te ima podršku i za rad s GPU-ovima.

Korisnik unutar Singularity kontejnera je isti onaj koji je i izvan kontejnera te ima jednake ovlasti unutar i izvan kontejnera. Ako se kontejner koristi na klasteru, korisnik ne može unutar kontejnera napredovati do root ovlasti te nema opasnosti da nehotično (ili namjerno) napravi štetu sustavu. Ako korisnik mora mijenjati kontejner, morat će to raditi na lokalnom računalu, na kojem ima root ovlasti, pa promijenjeni kontejner ponovno dopremiti na klaster.

Singularity je vrlo jednostavan za korištenje. Koristi se kao bilo koja druga aplikacija (ne pokreće se servis kao npr. u slučaju Dockera - zato dobro radi s raspoređivačima poslova!).

Ima svega nekoliko osnovnih naredbi:

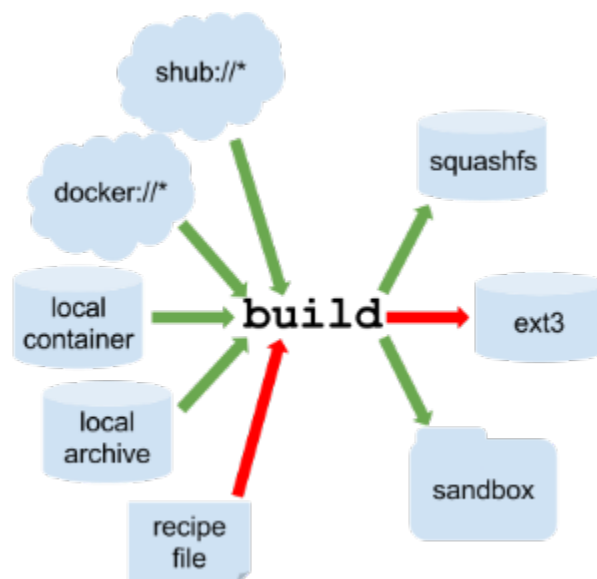
- `build`: naredba za izgradnju kontejnera
- `exec`: naredba za izvršavanje naredbe u kontejneru
- `inspect`: naredba za gledanje labela, runscripti, test scripti te varijabli okoline
- `pull`: naredba za povlačenje slika sa Singularity ili Docker Huba
- `run`: naredba za pokretanje slike kao izvršne datoteke
- `shell`: naredba za pokretanje ljuske u kontejneru

Izgradnja kontejnera

`build` može kreirati kontejnere u tri različita formata. Prvi i defaultni format je **squashfs** - kontejneri u tom formatu su *read-only* i sve promjene koje se naprave u kontejneru **nisu** spremljene nakon što se izađe iz kontejnera. Još su dva formata: **writable ext3** format koji je pogodan za interaktivno kreiranje kontejnera (opcija `--writable`) te **sandbox** direktorij (također *writable*, koristi se opcija `--sandbox`).

Naredba `build` se, osim za izgradnju kontejnera, može koristiti i za konvertiranje formata kontejnera između tri navedena formata te za izgradnju kontejnera iz Singularity recepta.

Izgradnja kontejnera iz Singularity recepta se preporuča, budući da je tako osigurana reproducibilnost (svi koraci izgradnje kontejnera su sadržani u Singularity receptu koji je vidljiv u kontejneru). Singularity recept se sastoji od dva glavna dijela: zaglavlje (eng. *header*) i sekcija (eng. *sections*). Zaglavlje je obavezan dio recepta, dok su sekcije opcionalne. U zaglavlju se definira bazni OS koji se želi instalirati u kontejneru, a u sekcijama se navode svi ostali koraci koje želimo iskoristiti prilikom izgradnje kontejnera (npr. kopirati datoteke, instalirati softver, kreirati varijable okoline i sl.).



Slika 2: Shematski prikaz procesa izgradnje kontejnera. Zelenim strelicama su označene akcije koje se mogu izvoditi bez sudo ovlasti, za akcije označene crvenim strelicama potrebno je imati sudo ovlasti.

Singularity recepti

Header

U headeru se definira bazni operacijski sustav koji će se koristiti u kontejneru. Osnovne ključne riječi su: `Bootstrap` referencira tip baze koji će se koristiti. To može biti, na primjer, `shub` za slike koje su na Singularity Hubu, `docker` za slike na Docker Hubu, `localimage` za slike spremljeno lokalno. Ključna riječ `From` je ime kontejnera (ili referenca na *layer*e u slučaju Dockera) koji će se koristiti.

```
Bootstrap: shub
From: vsoch/hello-world
```

Ovisno o vrijednosti za ključnu riječ `Bootstrap`, dodatne ključne riječi mogu biti potrebne. Primjer za instalaciju Centosa-7 s mirror URL-om:

```
Bootstrap: yum
OSVersion: 7
MirrorURL: http://mirror.centos.org/centos-%{OSVERSION}%{OSVERSION}/os/$basearch/
Include: yum
```

Više informacija o Singularity receptima dostupno je u [Singularity korisničkim uputama](#).

Sections

%help

Tekst koji se prikazuje kad se pozove `singularity help`.

%setup

Naredbe u ovoj sekciji se izvršavaju na host sustavu nakon što se bazni sustav instalira u kontejneru.

%files

U ovoj sekciji se kopiraju datoteke s host sustava u kontejner. Svaka linija je kombinacija izvor destinacija.

%labels

Ovdje se spremaju metapodaci vezani za kontejner.

%environment

Ovdje se dodaju varijable okoline. Treba imati na umu da se varijable u ovoj sekciji kreiraju tijekom runtime-a, tako da u slučaju da su potrebne tijekom procesa izgradnje kontejnera, treba ih također staviti u `%post` sekciju kao u primjeru:

```
%post

echo 'export TEST_VAR=test' >> $SINGULARITY_ENVIRONMENT
```

%post

Naredbe u ovoj sekciji se izvršavaju unutar kontejnera nakon što je bazni operacijski sustav instaliran. Tu se radi kreiranje direktorija, instaliranje softvera i biblioteka. U ovoj sekciji se ne mogu kopirati datoteke, ali se mogu downloadati s naredbama `git clone`, `wget` ili `curl`.

%runscript

Ne izvršava se tijekom izgradnje kontejnera, nego ostaje unutar kontejnera te se izvršava kad se kontejner pokrene (naredbom `singularity run` ili direktno kao izvršna datoteka).

Korištenje kontejnera

Kada jednom imamo gotov kontejner, postoji nekoliko naredbi za interagiranje s njim. Naredbom `shell` pokreće se ljuska unutar kontejnera, te se interagira s kontejnerom kao da je pokrenuta ljuska na malom virtualnom stroju.

```
$ singularity shell lolcow.simg
```

Naredbom `exec` izvršava se naredba unutar kontejnera (npr. može se pokrenuti program koji je instaliran u kontejneru).

```
$ singularity exec lolcow.simg ls /
```

Naredbom `run` pokreće se kontejner pozivanjem *runscripte* definirane u Singularity receptu.

```
$ singularity run lolcow.simg
```

Kada se kontejner koristi, on automatski povezuje tri direktorija: `/home/$USER`, `/tmp` i `$PWD`. Kontejner navedene direktorije "vidi" kao da su njegovi direktoriji. Dodatni direktoriji se mogu povezati naredbom `--bind`. U tom slučaju, mora postojati direktorij unutar kontejnera koji će Singularity koristiti za povezivanje s direktorijem na hostu.

Podnošenje poslova na klasteru

U slučaju da se koristi Singularity kontejner na klasteru, podnošenje poslova je jednako kao i u slučaju aplikacija koje su instalirane direktno na klasteru, budući da klaster vidi Singularity kao običnu aplikaciju. Na primjer, ako želimo pokrenuti kontejner `lolcow.simg`, minimalna SGE skripta će biti:

```
singularity-test.sh
```

```
#!/bin/sh
#$ -N singularity-test
#$ -cwd

singularity run lolcow.simg
```

A posao se podnosi (ako se gore navedena skripta zove `singularity-test.sh`) standardno sa:

```
$ qsub singularity-test.sh
```