

LXC

Įrankis eksperimentams



Sergej Kurakin

LXC - Linux Containers

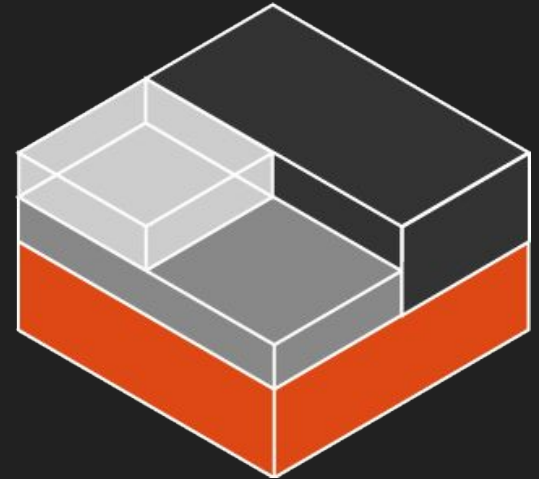
LXC - Virtualizacijos sprendimas gebantis paleisti kelias izoliuotas Linux sistemas (konteinerius) viename Linux "host".

Viskas veikia operacinės sistemos lygyje.

Veikia su "vanilla Linux kernel" - žymiai lengviau įdiegti.

Dabar tapo pakankamai populiarus.

<https://linuxcontainers.org/>



Kam man reikalingas LXC?

Noriu išmokti kažko naujo.

- Aš jau išbandžiau: VMWare, VirtualBox, KVM.

Man tenka daryti eksperimentus su programine įranga:

- Diegti skirtingas PĮ versijas.
- Kompiliuoti iš “source” su skirtiniais “patch”.
- Diegti ir išmokti konfiguruoti didelio prieinamumo sprendimus.

Kaip man padeda LXC?

Sukuria izoliuotą konteinerį, kuriame aš galiu atlikti eksperimentus.

Eksperimentas neužteršia savo veiksmais “host” sistemos.

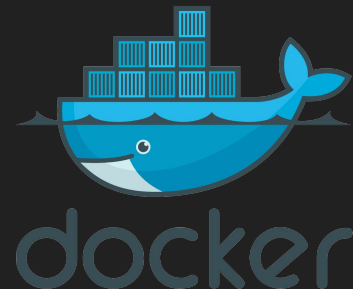
Minimalus “overhead” švaistant resursus virtualizacijai.

Kodėl ne “full virtualization”?

- Kartais reikalauja per daug “host” resursų.
- “Guest” paruošimas užima per daug laiko.
- Eksperimentams nereikia visų jos galimybių.
- Kartais tai tik resursų švaistimas.

Kodėl ne Docker?

- Dažniausiai, kai aš darau eksperimentą, aš dar nežinau, kaip teisingai padaryti “build” su Docker.
- Dažniausiai aš nežinau kur bus daromas “ship” ir ar ten bus Docker.
- Man nepatinka tas teorinis apribojimas: “one container - one process” ir siūlomi “workaround”.
- Darant eksperimentus man žymiai lengviau kažką tvarkyti per SSH.
- Man nereikia galvoti apie “data container”.
- Man nereikia taikytis prie Docker tinklo.



Diegimas

Diegimas yra labai paprastas. Bent Ubuntu 14.04 ir Debian 8.

```
sudo apt-get install lxc
```

Konteinerio sukūrimas

```
sudo lxc-create -t download \  
  -n conaname -- --dist ubuntu \  
  --release trusty --arch amd64
```


Konteinerio sunaikinimas

```
sudo lxc-destroy -n contname
```

Konteinerio paleidimas

```
sudo lxc-start -n contname -d
```

Konteinerio sustabdymas

```
sudo lxc-stop -n contname
```

Prisijungti prie konteinerio

```
sudo lxc-attach -n contname
```

```
sudo lxc-console -n contname
```

Naudingos komandos

```
sudo lxc-ls -f
```

```
sudo lxc-info -n contname
```

Konteinerių tipai

- unprivileged container, created as user
- unprivileged container, created as root
- privileged container, created as root

Kokios yra galimybės

Klonavimas (clone)

Įšaldymas (freeze)

Momentinė nuotrauka (snapshot)

Efemeriškas paleidimas (ephemeral)

Palaikomos distribucijos

- alpine
- altlinux
- archlinux
- busybox
- centos
- cirros
- debian
- fedora
- gentoo
- openmandriva
- opensuse
- oracle
- plamo
- ubuntu
- ubuntu-cloud

Kur viskas saugoma? backingstores!

- none/dir - katalogas kietajame diske
- btrfs - “b-tree” “copy-on-write” failų sistema
- lvm - Logical Volume Management
- overlayfs - “union mount” failų sistema
- zfs - ZFS failų sistemą

none/dir - /var/lib/lxc/<container>

Tinklo tipai

- empty - tik loopback
- veth - peer network device
- vlan - vlan interface
- macvlan - mac vlan interface
- phys - tikra tinklo korta “perduodama” į konteinerį

Labai geras straipsnis apie tinklo konfigūravimą <http://containerops.org/2013/11/19/lxc-networking/>

Demonstracija

Sukurkime vieną konteinerį

```
sudo lxc-create -t download \  
  -n redis1 -- --dist ubuntu \  
  --release trusty --arch amd64
```

Sukurkime kitą konteinerį

```
sudo lxc-create -t download \  
  -n redis2 -- --dist ubuntu \  
  --release trusty --arch amd64
```

“Ļjunkime” mūsu konteinerius

```
sudo lxc-start -n redis1 -d  
sudo lxc-start -n redis2 -d
```

Prisijunkime su “lxc-attach”

```
sudo lxc-attach -n redis1  
sudo lxc-attach -n redis2
```

Įdiekite Redis ir jo įrankius

```
apt-get install redis-server  
apt-get install redis-tools
```


Prikelkime Redis replicacija

Bandymo metu “redis1” bus “Master”, “redis2” bus “Slave”.

Abu Redis serverius perkeliame ant “public” tinklo.

Prisijngiam prie Redis: redis-cli ir paleidžiam “info” kad įsitikintume kad jie veikia.

“redis2” paleidžiame komanda:

```
SLAVEOF redis1 6379
```

Patikrinam, ar replikacija veikia.

Prisijunkime su “lxc-console”

```
sudo lxc-console -n redis1  
sudo lxc-console -n redis2
```

Kaip išeiti iš “lxc-console”

<Ctrl+a q>

Sukurkime dar vieną konteinerį

```
sudo lxc-create -t download \  
  -n web -- --dist ubuntu \  
  --release trusty --arch amd64
```

“ljunkime” mūsų naują konteinerį

```
sudo lxc-start -n web -d  
sudo lxc-attach -n web
```

İdiekime NGINX ir PHP

```
apt-get install nginx php5-fpm php5-redis
```

Paprastas PHP ir Redis panaudojimas

```
$redis = new Redis();  
$redis->connect('redis1', 6379);  
  
echo $redis->incr('test-  
increment');  
  
$redis->close();
```

Informacija apie konteinerius

```
sudo lxc-info -n web
```

```
sudo lxc-info -n redis1
```

```
sudo lxc-info -n redis2
```


“Išjungiam” konteinerius

```
sudo lxc-stop -n web
```

```
sudo lxc-stop -n redis1
```

```
sudo lxc-stop -n redis2
```

Sunaikinam konteinerius

```
sudo lxc-destroy -n web  
sudo lxc-destroy -n redis1  
sudo lxc-destroy -n redis2
```

O kam teko susidurti su:
Vagrant + VirtualBox + NFS

Létoka?

O gal pabandom kitaip ir su LXC

LXC konteineris išnaudoja “host” failų sistemą konteinerio failams saugoti:
`/var/lib/lxc/<container name>/rootfs`

O ką jei butu galima įkelti katalogą iš “host” į konteinerį?

Pasirodo tai yra įmanoma!

Tik reikia įdėti truputi pastangų. Ir ateiti tai gali pasikeisti...

Kaip tai padaryti?

“Host” sukuriame katalogą

```
mkdir ~/hostfolder
```

Ir reikia žinoti savo UID ir GID

Tam kad kiltu kuo mažiau problemų dėl “permissions” mes sinhronizusime savo vartotoją su “konteinerių”:

```
id -u
```

```
id -g
```


Sukurkime konteinerį

```
sudo lxc-create -t download \  
  -n linkingtest -- --dist ubuntu \  
  --release trusty --arch amd64
```

Paruoškime konteinerio “vidurius”

```
sudo lxc-start -n linkingtest -d
```

```
sudo lxc-attach -n linkingtest
```

```
mkdir /outsideworld
```

```
deluser ubuntu
```

```
adduser --uid 1000 <your username>
```

```
<Ctrl+D>
```

```
sudo lxc-stop -n linkingtest
```

Pakeiskime konteinerio konfigūraciją

```
sudo -i
```

```
cd /var/lib/lxc/linkingtest
```

```
vim config
```

```
# Custom mount
```

```
lxc.mount.entry = /home/skurakin/lxclinktest outsideworld  
none bind 0 0
```

Paleiskime mūsų konteinerį

```
sudo lxc-start -n linkingtest -d
```

Kaip man padėjo LXC?

Jo pagalba esu paleidęs MongoDB klasterį, kur buvo 3 “Shard” po 3 “Replica Set” ir 3 “Config” serveriai ir 1 “Router”.

Jo pagalba dariau “Demo” per Redis pranešimus.

Jo pagalba mokiausi RabbitMQ integracijos su PHP ir NodeJS.

Jo pagalba studijuojų Redis Sentinel ir Redis Cluter.

Dabar bandau jį adaptuoti darbui, kad pakeistu VirtualBox.

Laikas diskusijai



Sergej Kurakin

Darbinis paštas: sergej.kurakin@nfq.lt

Asmeninis paštas: sergej@kurakin.info

<https://www.linkedin.com/in/sergejkurakin>