

CANONICAL



LXCコンテナ - ハンズオン

Presentation by
Yoshi Kadokawa

February 2015



- LXCコンテナ入門
- cgroupを使ったリソース管理
- LXCコンテナの扱い方
- 非特権コンテナ
- OpenStackとLXC

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



LXCコンテナ入門

LXCのインストールとコンテナ作成



LXCのインストール

```
$ sudo apt-get -y install lxc
```

LXCコンテナ作成

\$

```
$ sudo lxc-create -t ubuntu-cloud -n handson-test
```

詳細情報付きでコンテナ作成

```
$ sudo lxc-create -t ubuntu-cloud -n test-details -- \  
-r precise -a amd64
```

downloadテンプレートを使用したコンテナ作成

```
$ sudo lxc-create -t download -n test-interactive
```

LXCコンテナの主な操作



現在のコンテナのリスト

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

コンテナの詳細

```
$ sudo lxc-info -n handson-test
```

コンテナの開始

```
$ sudo lxc-start -n handson-test -d
```

コンテナの停止

```
$ sudo lxc-stop -n handson-test
```

コンテナの削除

```
$ sudo lxc-destroy -n handson-test
```

コンテナへのアクセス①



コンソールアクセス

```
$ sudo lxc-console -n handson-test
```

** Quit from console with [Ctrl a + q]*

コンテナへのアタッチ

```
$  
$ sudo lxc-attach -n handson-test
```

コンテナへのアクセス②



SSH アクセスする準備

- SSHキーを事前に作成
- SSHキーを指定してLXCコンテナを作成

SSH キーの作成

```
$ ssh-keygen -t rsa
```

SSH キーを指定してコンテナ作成

```
# Create lxc container with ssh public key
$ sudo lxc-create -t ubuntu-cloud -n container-with-key -- \
-r trusty -a amd64 --auth-key $HOME/.ssh/id_rsa.pub
# Check IP
$ sudo lxc-info -n handson-test
# SSH access with the IP address
$ ssh ubuntu@<ip address>
```

LXCコンテナのネットワーク

ネットワークの追加



ネットワークの作成

```
# Install libvirt-bin
$ sudo apt-get -y install libvirt-bin
# Add virtual network
$ virsh net-define demo-br0.xml
$ virsh net-start demo-br0
$ virsh net-autostart demo-br0
```

LXCコンテナにネットワークインターフェースを追加する

```
# Add the following configuration to /var/lib/lxc/<container-name>/config
$ cat << EOF > /var/lib/lxc/hands-on-test/config
lxc.network.type=veth
lxc.network.link=demo-br0
lxc.network.flags=up
lxc.network.ipv4 = 10.81.0.2/24
EOF
```



cgroupでリソース管理

メモリのリソース管理



メモリ使用量制限

```
# Check the current limit
$ sudo lxc-cgroup -n handson-test memory.limit_in_bytes
# Limit memory to 1GB for a container
$ sudo lxc-cgroup -n handson-test memory.limit_in_bytes 1G
# Check if it's applied
$ cat /sys/fs/cgroup/memory/lxc/handson-test/memory.limit_in_bytes
```

メモリ最大使用値

```
$ sudo lxc-cgroup -n handson-test memory.max_usage_in_bytes
```

CPUのリソース管理



使用するCPUコアの制限

```
# Check the current limit
$ sudo lxc-cgroup -n handson-test cpuset.cpus
# Limit CPU core to 0 and 1
$ sudo lxc-cgroup -n handson-test cpuset.cpus 0,1
# Check if it's applied
$ cat /sys/fs/cgroup/cpuset/lxc/handson-test/cpuset.cpus
```

CPUの使用割合を制限

```
# Check the current limit
$ sudo lxc-cgroup -n handson-test cpu.shares
# Limit CPU share to 256 (25%)
$ sudo lxc-cgroup -n handson-test cpu.shares 256
# Check if it's applied
$ cat /sys/fs/cgroup/cpu/lxc/handson-test/cpu.shares
```

```
# You can check other cgroup configuration here
$ ls /sys/fs/cgroup/<resource>/lxc/
```

ディスクIOのリソース管理



ディスクIOの制限

```
# Check the current limit
$ sudo lxc-cgroup -n handson-test blkio.weight
# Limit blkio weight to 250
$ sudo lxc-cgroup -n handson-test blkio.weight 250
# Check if it's applied
$ cat /sys/fs/cgroup/blkio/lxc/handson-test/blkio.weight
```

その他のcgroup設定値

```
# You can check other cgroup configuration here  
$ ls /sys/fs/cgroup/<resource>/lxc/
```

configファイルを指定してコンテナ作成

```
$ sudo lxc-create -t ubuntu-cloud -f <filename> -n <container-name>
```

```
lxc.cgroup.memory.limit_in_bytes = 1G  
lxc.cgroup.cpuset.cpus = 0  
lxc.cgroup.blkio.weight = 250
```



コンテナの扱い方

LXC Clone & Snapshot



Clone

```
# Clone container from A to B
$ sudo lxc-create -t ubuntu-cloud -n A
$ sudo lxc-clone A B
```

Snapshot

```
# Create a snapshot
$ sudo lxc-snapshot -n B

# Create a snapshot with comment
$ echo "Initial version" >> /tmp/snapshot-comment
$ sudo lxc-snapshot -c /tmp/snapshot-comment -n B

# List up snapshots with comments
$ sudo lxc-snapshot -C -L -n B

# Restore from snapshot "snap0"
$ sudo lxc-snapshot -r snap0 -n B
```

Container import & export



Export

```
# Stop the container
$ sudo lxc-stop -n A
# Archive the directory
$ sudo tar acfv container-A.tar.gz /var/lib/lxc/A
```

Import

```
# Extract the tar archive to appropriate place
$ sudo tar xfv container-A.tar.gz
$ sudo mv -r var/lib/lxc/A /var/lib/lxc/
# Check the container
$ sudo lxc-ls -f
# Start the container
$ sudo lxc-start -n A -d
```



CANONICAL

Exercise

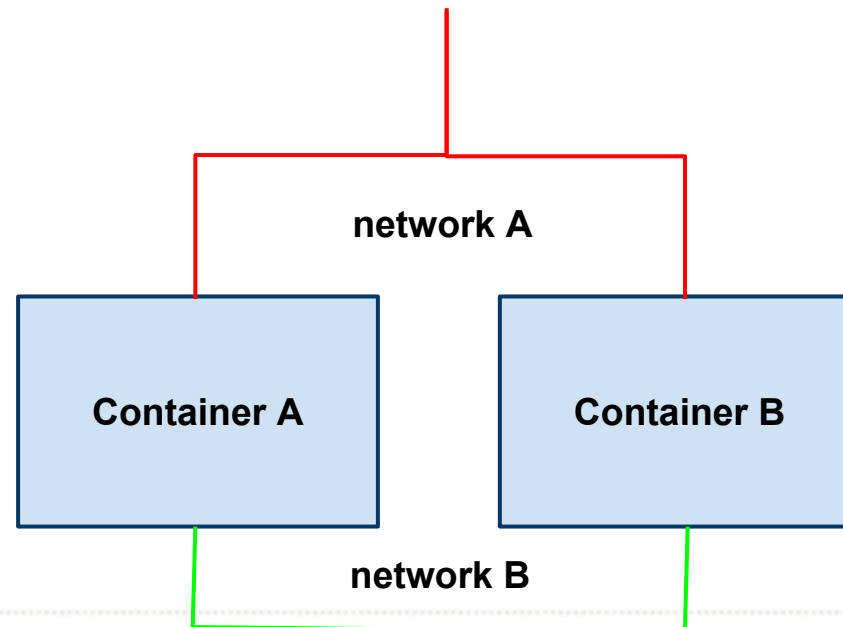
Exercise!



以下の**LXC**コンテナを作ってみましょう

- 2つ以上のコンテナを作成
- すべてのコンテナは**2つ以上のNIC**を持ち、それぞれ疎通がとれていること
- すべてのコンテナは以下のリソース配分を
 - 512MB memory
 - 1 vCPU

例:



非特権モードのLXCコンテナ

非特権モードのコンテナ



非特権モードのコンテナとは

名前空間が分離されているコンテナでも、
コンテナ内の**UID 0**は、コンテナ外の**UID 0**と同じであり、
つまりホスト側の**root**に昇格する懸念があった
非特権モードのコンテナは、**root**権限ではなく、一般ユーザの権限で
LXCのコンテナを作ることができるモードになる

非特権モードで作成したコンテナのパスは以下の様に

- `/etc/lxc/lxc.conf` => `~/.config/lxc/lxc.conf`
- `/etc/lxc/default.conf` => `~/.config/lxc/default.conf`
- `/var/lib/lxc` => `~/.local/share/lxc`
- `/var/lib/lxc/snaps` => `~/.local/share/lxc/snaps`
- `/var/cache/lxc` => `~/.cache/lxc`

非特権モードのコンテナを作成



非特権モードのための準備

```
# Assign yourself a set of uids and gids
$ sudo usermod -v 100000-200000 -w 100000-200000 $USER
$ sudo chmod +x $HOME
# Create ~/.config/lxc/default.conf
$ mkdir -p ~/.config/lxc .local/share/lxc .cache/lxc
$ cat << EOF > ~/.config/lxc/default.conf
lxc.network.type = veth
lxc.network.link = lxcbr0
lxc.id_map = u 0 100000 65536
lxc.id_map = g 0 100000 65536
EOF
# Add
$ echo "$(whoami) veth lxcbr0 10" | sudo tee --append /etc/lxc/lxc-usernet
```

Now it's ready to create !

```
$ lxc-create -t download -n unprivileged-cntr -- \
-d ubuntu -r trusty -a amd64
```

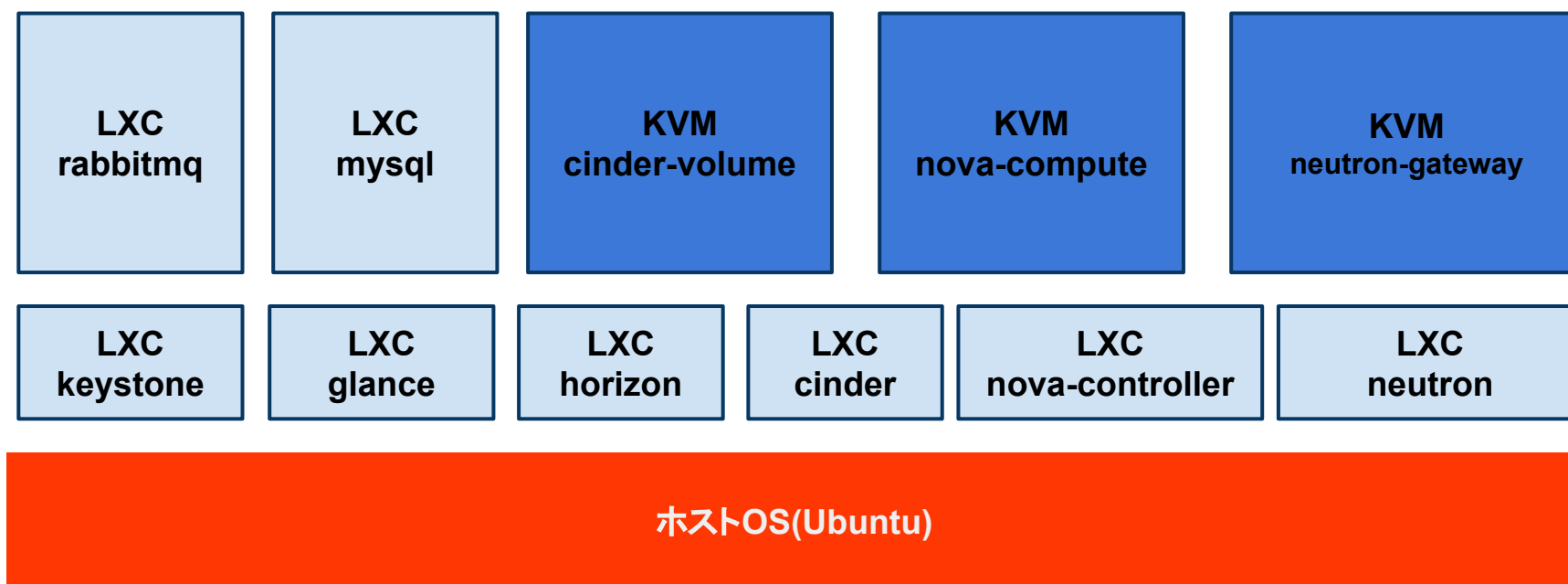


OpenStackとLXC

LXCを使ってOpenStackを構築



LXCをベースにOpenStack構築





CANONICAL

Thank you !