


Debian / Ubuntu ユーザーミーティング in 札幌 2017.07

lxc について

Norimitsu Sugimoto (杉本 典充)
dictoss@live.jp

2017-07-14

アジェンダ

- 自己紹介
 - 仮想化技術について
 - lxc とは
 - lxc のインストール
 - lxc のコマンド解説
 - lxc を実用する
 - おわりに
 - 参考資料
- 

自己紹介

- Norimitsu Sugimoto (杉本 典充)
- dictoss@live.jp
- Twitter: @dictoss
- Debian-3.1、FreeBSD-6.2 の頃から使っています
- Debian GNU/kFreeBSD が気になっておりウォッチ中
- 仕事はソフトウェア開発者をやっています



仮想化技術 について

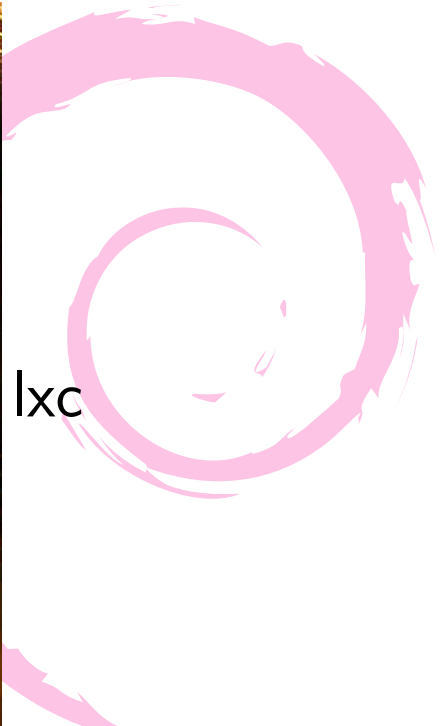
仮想化技術の分類

- コンテナ型仮想化
- 準仮想化型
- 完全仮想化型 (エミュレーション型)
- 完全仮想化型 (ハイパーバイザ型)

仮想化技術のメリット・デメリット

- 準仮想化型、完全仮想化型
 - 物理マシンをエミュレートした仮想マシンとして動作する
 - 仮想マシン上でもカーネルを動作させる
 - 物理マシンで動かしていたプログラムはほぼそのまま動く
 - CPU、メモリ、ディスクを多く消費する
- コンテナ型仮想化
 - ゲスト環境の動作にカーネルは不要 (ホスト環境のカーネル上で動作)
 - ゲスト環境は、ホスト環境から見るとプロセスとして扱われる
 - ゲスト環境が利用できるリソースに制約がつく場合がある

- chroot システムコールと chroot コマンド
- 1982 年にビル・ジョイが開発したとされている
- # chroot rootfsdir でコンテナ環境に入れることができる



lxc

lxc とは

- Linux Containers のことで、省略して lxc と読んでいる
- あるディレクトリ配下に実行ファイル、ライブラリ、設定ファイルを適切に配置した rootfs を準備する
- rootfs を chroot 環境で起動し、仮想マシンのように動かすことができる
- Debian 9 Stretch では lxc-2.0.5 を採用



lxc のイン
ストール

インストールの流れ

- ブリッジネットワークと libvirtd の準備
- lxc のインストール

ブリッジネットワークと libvirtd の準備

パッケージのインストール

```
# apt-get install libvirt-clients \  
libvirt-daemon-system ebttables dnsmasq
```

仮想ネットワークの設定

```
# virsh net-autostart default  
# virsh net-start default
```

ブリッジネットワークと libvirtd の準備

確認

```
$ sudo virsh net-info default
Name:                default
UUID:                78564864-f237-4059-a12a-3ec04369a27b
Active:              yes
Persistent:          yes
Autostart:           yes
Bridge:              virbr0

$ ip a show virbr0
    192.168.122.1/24 が付与されている
```

lxcのインストール

コンテナ内のリソース制約を処理する cgroup の確認

```
# mount | grep cgroup  
stretch は標準で mount されている
```

パッケージのインストール

```
# apt-get install lxc libvirt0 libpam-cgroup \  
libpam-cgfs
```

環境を確認

```
$ ls /usr/bin | grep lxc  
# lxc-checkconfig
```

lxcのインストール

- lxc-create
- lxc-destroy
- lxc-start
- lxc-stop
- lxc-console
- lxc-attach





lxc のコマ
ンド解説

lxc-create(1)

```
# lxc-create -n demo1 -t debian -- \  
--release=stretch --arch=amd64 \  
--mirror=http://ftp.jp.debian.org/debian
```

- 実行するとテンプレートが debian の場合は debootstrap を実行して rootfs をダウンロードする
- lxc のゲスト環境のディレクトリは、`/var/lib/lxc/" コンテナ名 "`。中身は以下。
 - config (設定ファイル)
 - rootfs (コンテナの中身)

lxc-create(2)

- config を修正して、ネットワークの設定を行う
- `lxc.network.type = veth`
- `lxc.network.flags = up`
- `lxc.network.link = virbr0`
- `lxc.network.name = eth0`
- `lxc.network.ipv4 = 192.168.122.60/24`
- `lxc.network.ipv4.gateway = 192.168.122.1`

```
# lxc-destroy -n demo1
```

- コンテナを削除します

```
# lxc-start -n demo1
```

- 実行して何もエラーが表示されなければ、バックグラウンドで lxc コンテナが動き出します
- 起動したコンテナへの接続は、後述する lxc-console または lxc-attach で行います
- コンテナへのログインは ssh でもログインできますが、ユーザを作成する必要があります

```
# lxc-stop -n demo1
```

- コンテナ環境を終了するよう指示を出します
- コンテナ環境の終了とは、コンテナ内の init プログラムを終了することをいいます
- コンテナ環境で shutdown 命令は実行できません

```
# lxc-console -n demo1
```

- lxc のゲスト環境のコンソールに接続します
- コンソールを抜ける場合は、「Ctrl+a q」の順に入力してください

```
# lxc-attach -n demo1 {command}
```

- lxc のゲスト環境でコマンドを実行します
- コマンドを指定しない場合は、コンテナ内のユーザのデフォルトシェルが実行されます
- lxc-console でログインすることに比べ、いきなりコンテナ内でシェルを実行できるため、lxc-attachの方がコンテナ内の整備がしやすいです



lxc を実用
する

コンテナ環境のセットアップの流れ

- lxc-create を実行してコンテナを生成する
- lxc のゲスト環境の config を書き換えてネットワークを設定する
- lxc-start してコンテナを起動する
- lxc-attach でゲスト環境に入る

```
# passwd  
# adduser username  
# apt-get install sudo vim-tiny  
# visudo
```

- ssh ログインしてお好みに設定する

何に lxc を使うか

- 一時的な検証で、ホスト環境にいろいろインストールしたくない場合
- アプリケーションのクリーンビルドやクリーンインストールをテストする場合
- ホスト環境は systemd、ゲスト環境は sysvinit と使い分ける場合
- python2 系と python3 系の wsgi アプリを 1 つのホストで動かしたい場合
- ホスト環境と異なる CPU アーキテクチャのエミュレーション環境がほしい場合
 - apt-get install qemu qemu-user-static binfmt-support
 - その後、lxc-create を実行してください
 - 詳しくは CrossDebootstrap を調べてみてください

おわりに

- Debian 上で lxc を試してみました
- 発展系である LXD や docker へつなげていきましょう
- コンテナは便利ですので試してみてください

- 「LXC」 <https://linuxcontainers.org/>
- 「LXC - Debian Wiki」 <https://wiki.debian.org/LXC>
- 「LXC LibVirt Default Network」 <https://wiki.debian.org/LXC/LibVirtDefaultNetwork>