

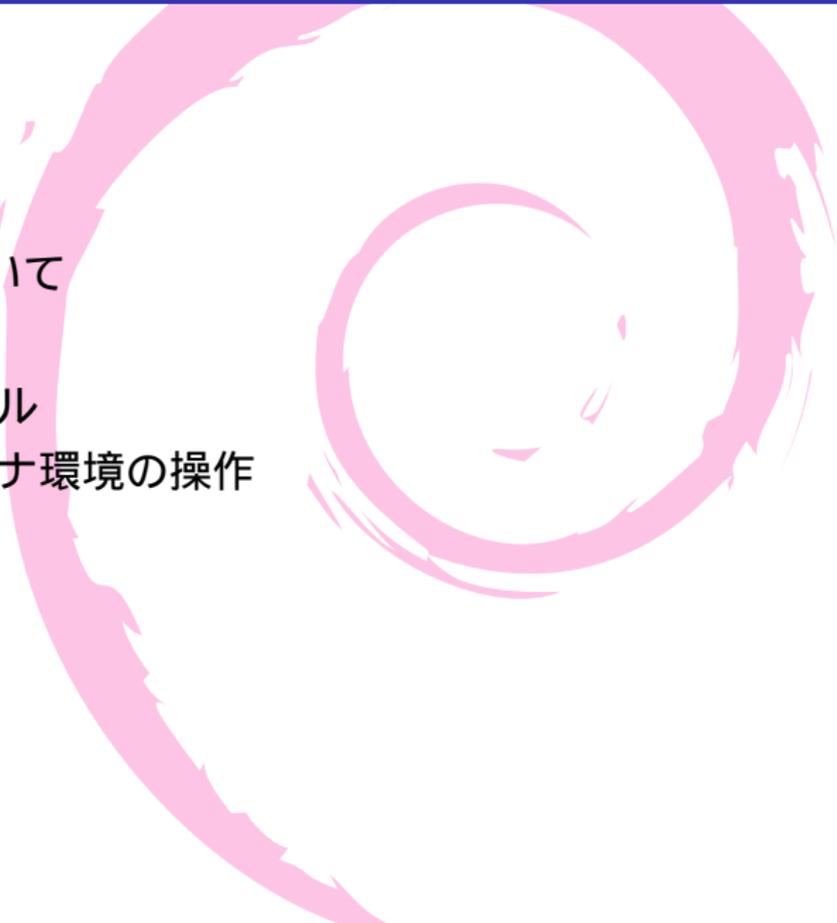
第142回東京エリア Debian 勉強会

Debian で lxc をセットアップしてみよう

Norimitsu Sugimoto (杉本 典充)
dictoss@live.jp

2016-08-20

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 によるコンテナ環境の操作
 - 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 使って10年以上、FreeBSD 使って5年以上
- Debian GNU/kFreeBSD が気になっておりウォッチ中
- tilegx に少し手を出したがなかなか手ごわい
- 仕事はソフトウェア開発者をやってます



仮想化技術 について

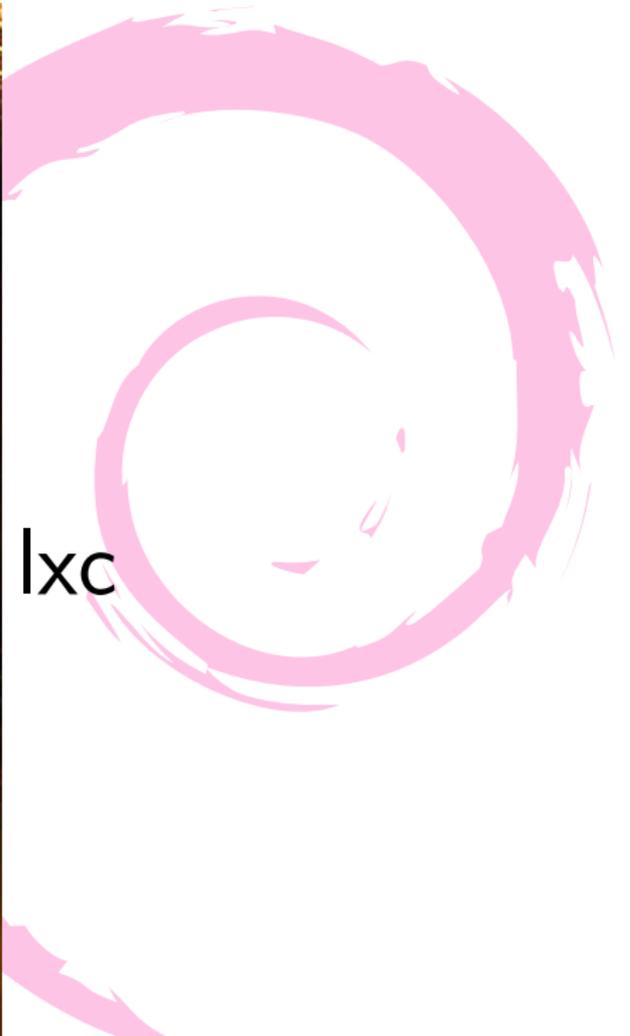
仮想化技術の分類

- 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



lxc のイン
ストール

lxcのインストール

- apt-get install lxc bridge-utils libvirt-bin
- lxc-*コマンドが使えるようになります
- 今回は libvirtd で NAT ネットワークを lxc ゲスト環境に割り当てます
- cgroups の mount 設定
- lxc-checkconfig で動作環境チェック



lxcを使う

lxc-create(1)

- LANG=C SUITE=jessie
MIRROR=http://ftp.jp.debian.org/debian lxc-create -n
debstudy1 -t debian
- 中で debootstrap を実行して rootfs をダウンロードする
- できた lxc のゲスト環境のディレクトリの中
 - config
 - fstab
 - rootfs

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 = 192.168.122.203/24`
- `lxc.network.ipv4.gateway = 192.168.122.1`

- lxc-ls で /var/lib/lxc 配下にいるコンテナ環境の一覧を表示します
- -active, -running、-stopped などのオプションを指定すると、その状態に絞って一覧を表示できます

- `lxc-destroy -n debstudy1`
- コンテナ環境を削除できます
- コンテナが実行中の場合は削除できないのがデフォルト
- `-force` オプションを指定すれば、コンテナが実行中なら `stop` させて削除する

- `lxc-start -n debstudy1` または `lxc-start -n debstudy1 -d`
- `-d` または `-daemon` オプションを指定するとバックグラウンドで `lxc` のゲスト環境が起動する
- `-d` または `-daemon` を指定しない場合はそのターミナル上で `lxc` のゲスト環境が起動し、コンソールに接続します

- `lxc-stop -n debstudy1`
- 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、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 debstudy1`
- 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 を実用
する

コンテナ環境のセットアップの流れ

- lxc-create 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
- chroot でゲスト環境に入る
 - passwd で root パスワード変更
 - adduser
 - apt-get install sudo vim-tiny
 - visudo
- lxc-start
- ssh ログインしてお好みに設定します

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
- 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
- ホスト環境は systemd、ゲスト環境は sysvinit と使い分ける
- 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
- ホスト環境と異なる CPU アーキテクチャのクロス環境がほしい場合

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>
- 「debootstrap を有効活用してみよう」
<http://tokyodebian.alioth.debian.org/pdf/debianmeetingresum201304.pdf>