コンテナ型ハイパーバイザー「LXD」入門

柴田 充也

Ubuntu Japanese Team

2016 年 2 月 27 日

LXD について

LXD（lex-dee）: Go 言語製のコンテナ型ハイパーバイザー

- LXC を使って複数のホスト上の複数のコンテナを管理する仕組み [a]
- クライアントサーバーモデル
- 設定管理システム
- イメージ管理システム
- （クライアントの）マルチプラットフォーム対応
- OpenStack Nova 向けプラグイン

[a] https://linuxcontainers.org/ja/lxd/introduction/

LXC：Linux のコンテナ機能を使うためのインターフェース

- Docker や systemd でも使われているカーネルの namespace や cgroups などを使用
- 「コンテナ」と呼ばれる軽量な仮想環境を構築
- CPU の仮想化支援機構が不要
- KVM に比べると起動が高速でイメージサイズが小さい
- 各種言語向けのバインディングが存在
- Linux でしか動作しない

詳しいことは：

「LXC で学ぶコンテナ入門 ー軽量仮想化環境を実現する技術」[a]

[a] http://gihyo.jp/admin/serial/01/linux_containers

Docker

- 主にアプリケーションコンテナを作る
- １コンテナに１プロセス
- 一度作ったら中身は変えない

Docker

- 主にアプリケーションコンテナを作る
- １コンテナに１プロセス
- 一度作ったら中身は変えない

LXC

- 主にシステムコンテナを作る
- １コンテナに init 以下のフルシステムが動く
- コンテナ内部にログインして作業を行う

Docker

- 主にアプリケーションコンテナを作る
- １コンテナに１プロセス
- 一度作ったら中身は変えない

LXC

- 主にシステムコンテナを作る
- １コンテナに init 以下のフルシステムが動く
- コンテナ内部にログインして作業を行う

上記はあくまで一般論で、異なる使い方も可能です。

LXC の不満点

- 何をするにしても管理者権限が必要
- 同一ホストのコンテナしか操作できない
- コンテナの設定を変更する方法がわかりにくい
- 複数のホスト間でコンテナインスタンスを共有しづらい

LXC の不満点

- 何をするにしても管理者権限が必要
- 同一ホストのコンテナしか操作できない
- コンテナの設定を変更する方法がわかりにくい
- 複数のホスト間でコンテナインスタンスを共有しづらい

LXC の機能よりはユーザー向けのインターフェースに問題がある。

LXC の不満点

- 何をするにしても管理者権限が必要
- 同一ホストのコンテナしか操作できない
- コンテナの設定を変更する方法がわかりにくい
- 複数のホスト間でコンテナインスタンスを共有しづらい

LXC の機能よりはユーザー向けのインターフェースに問題がある。

Docker のやり方をパクろう！

LXC の不満点

- 何をするにしても管理者権限が必要
- 同一ホストのコンテナしか操作できない
- コンテナの設定を変更する方法がわかりにくい
- 複数のホスト間でコンテナインスタンスを共有しづらい

LXC の機能よりはユーザー向けのインターフェースに問題がある。

LXC のハイパーバイザーとそのクライアントを作ろう！

改めて LXD について

- LXC を使って複数のホスト上の複数のコンテナを管理する仕組み
- クライアントサーバーモデル
- 設定管理システム
- イメージ管理システム
- （クライアントの）マルチプラットフォーム対応
- OpenStack Nova 向けプラグイン

改めて LXD について

- LXC を使って複数のホスト上の複数のコンテナを管理する仕組み
- クライアントサーバーモデル
  - サーバー：コンテナを管理する「lxd」
  - クライアント：サーバーにリクエストする「lxc」
    - lxc コマンドは管理者権限が不要
    - lxd を操作できる lxd グループに入っていればいい
    - サブコマンド方式で統一性のある UI になった
  - ネットワーク透過性
  - REST API の提供
    - 異なるホストのコンテナを操作できるようになった
- 設定管理システム
- イメージ管理システム
- （クライアントの）マルチプラットフォーム対応
- OpenStack Nova 向けプラグイン

改めて LXD について

- LXC を使って複数のホスト上の複数のコンテナを管理する仕組み
- クライアントサーバーモデル
- 設定管理システム
  - コンテナの設定を sqlite3 に保存
  - 設定の更新・表示は lxc コマンド経由で行う
- イメージ管理システム
- （クライアントの）マルチプラットフォーム対応
- OpenStack Nova 向けプラグイン

改めて LXD について

- LXC を使って複数のホスト上の複数のコンテナを管理する仕組み
- クライアントサーバーモデル
- 設定管理システム
- イメージ管理システム
  - 非特権システムコンテナを簡単に構築
  - マイグレーション機能
  - 他ホスト上の lxd にも移動可能
- （クライアントの）マルチプラットフォーム対応
- OpenStack Nova 向けプラグイン

改めて LXD について

- LXC を使って複数のホスト上の複数のコンテナを管理する仕組み
- クライアントサーバーモデル
- 設定管理システム
- イメージ管理システム
- （クライアントの）マルチプラットフォーム対応
  - Windows/Mac から Ubuntu 上のコンテナを操作できる
- OpenStack Nova 向けプラグイン

改めて LXD について

- LXC を使って複数のホスト上の複数のコンテナを管理する仕組み
- クライアントサーバーモデル
- 設定管理システム
- イメージ管理システム
- （クライアントの）マルチプラットフォーム対応
- OpenStack Nova 向けプラグイン
  - KVM の代わりに LXD をコンピュートノードとして追加できる

非特権システムコンテナ

- root が UID=0/GID=0 ではないコンテナ

[a] https://github.com/lxc/lxd/blob/master/specs/userns-idmap.md

非特権システムコンテナ

- root が UID=0/GID=0 ではないコンテナ
- 特権（privileged）コンテナはホストと UID/GID を共有する
  - ホストとゲストは隔離されているが、ゲストの root がホストをまったく操作できないというわけではない

[a] https://github.com/lxc/lxd/blob/master/specs/userns-idmap.md

非特権システムコンテナ

- root が UID=0/GID=0 ではないコンテナ
- 特権（privileged）コンテナはホストと UID/GID を共有する
  - ホストとゲストは隔離されているが、ゲストの root がホストをまったく操作できないというわけではない
- 非特権（unprivileged）コンテナはホストと異なる UID/GID を利用する
  - User Namespace を使ってホスト・ゲスト間のマッピングを行う [a]
  - ゲストの root である UID=0 はホストでは UID=100000
  - ゲストのユーザーである UID=1000 はホストでは UID=101000
  - ホスト・ゲスト間のファイルコピー用 UI も用意してある

[a] https://github.com/lxc/lxd/blob/master/specs/userns-idmap.md

LXD/LXC に向いている仕事

- KVM や VMWare、Xen が行っていたシステムの仮想化
- リソースが少ない環境での仮想化
  - CPU の仮想化支援機構がない
  - ディスク・メモリが少ない
- 特定のアプリを隔離領域で動かす
  - 将来の Ubuntu デスクトップで採用予定

LXD/LXC に向いている仕事

- KVM や VMWare、Xen が行っていたシステムの仮想化
- リソースが少ない環境での仮想化
  - CPU の仮想化支援機構がない
  - ディスク・メモリが少ない
- 特定のアプリを隔離領域で動かす
  - 将来の Ubuntu デスクトップで採用予定

LXD/LXC に向いていない仕事

- ホスト・ゲストで異なるカーネルを動かす必要がある
- 既に Docker 資産が充実しているソフトウェア環境の構築
  - 素直に Docker を使いましょう
  - Docker on LXD はまだ動かないようです

LXD 入門

バージョンについて

- 0.x 系と 2.0 系がある
- 0.x 系はこれまでのリリース版
  - あくまで Techinical Preview という位置づけ
- 2.0 系は次期正式リリースとして開発中
  - LXC のメジャーバージョンにあわせてバージョン番号調整
  - 2 月時点ではまだベータ版
- Ubuntu 16.04 LTS では 2.0 が入る（予定）
- 2.0 には後方互換性がない部分がある
- 今から評価するなら 2.0 を使った方がいい

必要なもの

- ホストマシン：Ubuntu 14.04 LTS 以上
  - VirtualBox 上でも化
  - ZFS を使いたいなら Ubuntu 16.04 LTS 以上
- クライアントマシン：Ubuntu 14.04 LTS 以上
  - ホストと共用でも可

必要なもの

- ホストマシン：Ubuntu 14.04 LTS 以上
  - VirtualBox 上でも化
  - ZFS を使いたいなら Ubuntu 16.04 LTS 以上
- クライアントマシン：Ubuntu 14.04 LTS 以上
  - ホストと共用でも可

Ubuntu なんてないよ！

- デモサービスがあります
- `https://linuxcontainers.org/ja/lxd/try-it/`
- ブラウザだけで操作可能
- 30 分の時間制限あり

LXD のインストール

```
$ sudo add-apt-repository ppa:ubuntu-lxc/lxd-stable
$ sudo apt update
$ sudo apt full-upgrade
$ sudo apt install lxd zfsutils-linux
$ newgrp lxd
```

PPA について

Ubuntu 16.04 LTS/Xenial の場合は PPA の追加は不要です。ただし 16.04 のリリース後により新しい LXD を使いたい場合は、PPA が必要になるかもしれません。

ZFS について

zfsutils-linux パッケージのインストールと zfs モジュールのロードはストレージバックエンドに ZFS を使いたい時のみ。

最初の一歩：LXD の初期設定

```
$ sudo lxd init
Name of the storage backend to use (dir or zfs): zfs
  ゲストコンテナをどのように保存するのか
Create a new ZFS pool (yes/no)? yes
  新しい ZFS プールを作るのか
Name of the new ZFS pool: lxd
  作成する ZFS プールの名前
Would you like to use an existing block device (yes/no)? no
  既存のブロックデバイスを使って ZFS を構築するのか
Size in GB of the new loop device (1GB minimum): 2
  新規に作るループバックデバイスのサイズ
Would you like LXD to be available over the network (yes/no)? no
  LXD サービスをホストの外に公開するか
LXD has been successfully configured.
```

ZFS の状態

```
$ sudo zpool list
NAME   SIZE  ALLOC   FREE  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
lxd   1.98G   469K  1.98G         -     0%     0%  1.00x  ONLINE  -
$ sudo zpool status
  pool: lxd
 state: ONLINE
  scan: none requested
config:

        NAME                    STATE     READ WRITE CKSUM
        lxd                     ONLINE       0     0     0
          /var/lib/lxd/zfs.img  ONLINE       0     0     0

errors: No known data errors
```

イメージの取り込み

```
$ lxd-images import ubuntu --alias ubuntu
  images.linuxcontainers.org からベースイメージのダウンロード
Setup alias: ubuntu
$ lxc image list
```

イメージの起動（イメージのインスタンス化）

```
$ lxc launch ubuntu first
Creating first
Starting first
```

```
$ lxc list
```

起動しているインスタンスのリスト

```
$ lxc info first
```

指定したインスタンスの情報表示

```
$ lxc config show first
```

指定したインスタンスの設定表示

ホストとゲストのプロセスの状態

```
$ lxc exec first -- ps axjf
```

コンテナの中で「`ps axjf`」を実行する
`/sbin/init` が動いていることを確認

```
$ ps axjf
```

コンテナの中のプロセスの `PID` や `UID` が
マッピングされていることを確認

リソースの制限

```
$ free -m
$ lxc exec first -- free -m
$ lxc config set first limits.memory 256MB
$ lxc exec first -- free -m
```

制限されていない？

今回の例だとゲスト側で lxcfs が動いていないせいか、ホストの/proc の情報がそのまま free で表示されてしまっています。

スナップショット

```
$ lxc snapshot first clean
```
first コンテナを clean という名前でスナップショット化
```
$ lxc list
```
SNAPSHOT フィールドの数が増えていることを確認
```
$ lxc exec first -- touch /home/ubuntu/vaporfile
$ lxc exec first -- ls /home/ubuntu/
$ lxc restore first clean
```
ファイル作成前の状態にリストア
```
$ lxc exec first -- ls /home/ubuntu/
```

インスタンスをイメージに

```
$ lxc stop first
```

インスタンスを停止

新しい LXD なら publish すると一時的にコンテナを停止します

```
$ lxc publish first --alias first-base
```

first コンテナの内容を first-base として保存

イメージサーバーの構築@ubuntu2 ホスト

```
$ lxc config set core.https_address 192.168.122.24
$ lxc config set core.trust_password ubuntujp
```

他のマシンからイメージサーバーにアクセスするための設定
パスワード認証ではなく証明書を使うことも可能

イメージサーバーの追加@ubuntu1 ホスト

```
$ lxc remote add ubuntu2 192.168.122.24
```

フィンガープリントの確認とパスワードの入力

```
$ lxc image list ubuntu2:
```

ubuntu2 マシン上の lxd に登録されたイメージを表示できる

ライブマイグレーション

```
$ lxc move ubuntu2:nginx ubuntu1:nginx
```

ubuntu2 マシンから ubuntu1 マシンへインスタンスを移動