

lxc とは何ぞや？

今年最後のネタです。

まえだこうへい mkouhei@debian.or.jp
IRC nick: mkouhei

2009 年 12 月 21 日

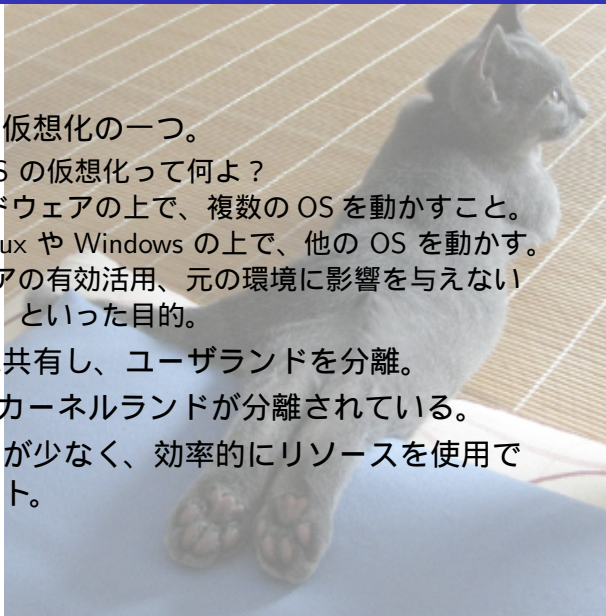
lxc とは何ぞや？

- 正式名称 Linux Containers
- lxc の構成
 - コンテナ自体が稼働するカーネルの機能
 - lxc を管理するための lxc-* コマンドラインツール
- Daniel Lezcano 氏が開発、メンテナンス。GPL2 で配布。
- Linux Kernel 2.6.29 で lxc に必要なパッチが mainline にマージされた。

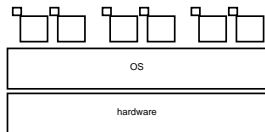


そもそもコンテナって何よ？

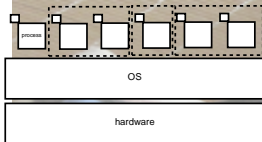
- いわゆる OS の仮想化の一つ。
 - そもそも OS の仮想化って何よ？
 - 1つのハードウェアの上で、複数の OS を動かすこと。
 - 例えば、Linux や Windows の上で、他の OS を動かす。
 - ハードウェアの有効活用、元の環境に影響を与えないようにする、といった目的。
- カーネル空間は共有し、ユーザランドを分離。
- Xen や KVM はカーネルランドが分離されている。
- オーバーヘッドが少なく、効率的にリソースを使用できるのがメリット。



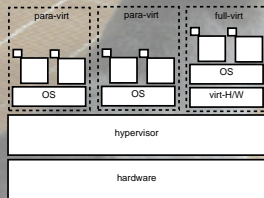
他の仮想化技術との比較



通常の OS



コンテナ



ハイパーバイザー
(Xen, KVM など)

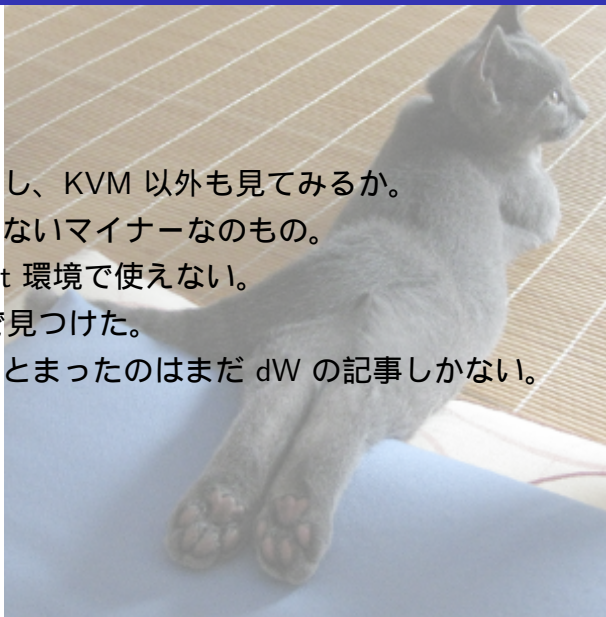
他のコンテナ技術との比較

- OpenVZ, Linux-VServer
- Solaris Containers
- FreeBSD jail
- lxc で提供するサービス
 - システムコンテナ：いわゆる OS の仮想化
 - アプリケーションコンテナ：chroot でのプロセスの分離



lxc を使ってみた動機。

- 連載も終わったし、KVM 以外も見てみるか。
- あまり使われてないマイナーなもの。
- lguest だと 64bit 環境で使えない。
- たまたま dW で見つけた。
- でもある程度まとまったのはまだ dW の記事しかない。
- 何より手軽！



導入する



パッケージインストール

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



確認する

```
$ lxc-checkconfig
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup namespace: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled

--- Misc ---
Veth pair device: enabled
Macvlan: enabled
File capabilities: enabled
```

なんか足りなければ、カーネルリビルドしとけ。

3分もいないぜ。

即席コンテナで稼働確認する。

```
$ sudo lxc-execute -n hoge -f \  
/usr/share/doc/lxc/examples/lxc/lxc-macvlan.conf /bin/bash  
$ sudo lxc-info -n hoge  
'hoge' is RUNNING  
$ sudo lxc-stop -n hoge <-別のシェルで実行。
```

システムコンテナを使う。

簡単に Debian 環境を増やせて、簡単に起動できるのが良い。
やること。

- ネットワークの設定
- cgroup ファイルシステムのマウント
- コンテナの作成
- コンテナのファイルの微調整
- コンテナの起動

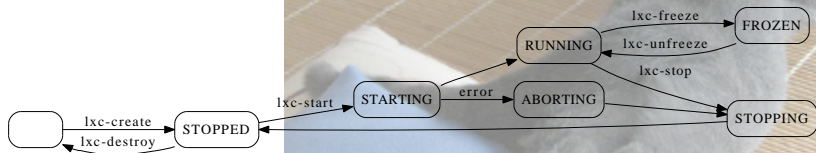


lxc の仕組みの話



lxc のライフサイクル

他の仮想マシンのライフサイクルと大差ない。



lxc のライフサイクル

status	コマンド	実行後の状態	備考
コンテナの 起動	lxc-start	STARTING, RUNNING	システムコンテナ
	lxc-execute		アプリケーションコンテナ
コンテナの 一時停止	lxc-freeze	FROZEN	
コンテナの 再開	lxc-unfreeze	RUNNING	
コンテナの 停止	lxc-stop	STOPPING, STOPPED	
コンテナの 再起動	lxc-restart	STOPPING, STOPPED STARTING, RUNNING	
コンテナの 作成	lxc-create	STOPPED	lxc-debian は内部で lxc-create を実行。
コンテナの 破棄	lxc-destroy		

リソースの管理

- cgroup(Control Group) で行う。
- インタフェースは lxc-cgroup コマンド
- マウントした cgroup ファイルシステムからアクセスできる。

```
cgroup /var/local/cgroup cgroup defaults 0 0
```

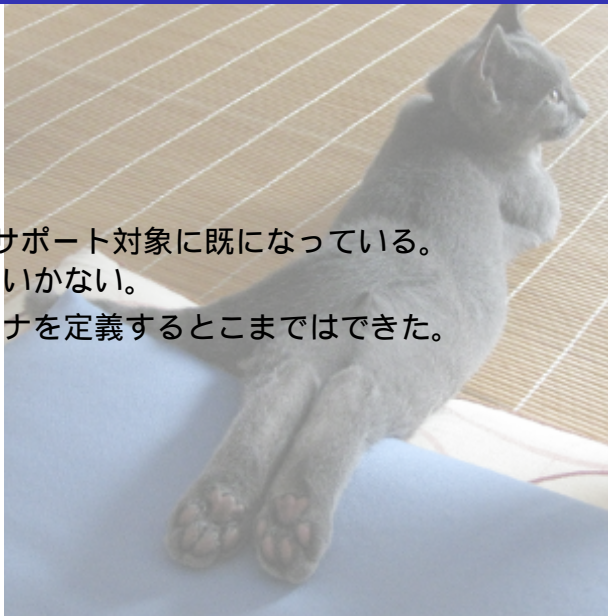
- システムコンテナの rootfs を置くのはどこでも良い。
- Read, Write の権限は この /var/local/cgroup 以下のファイルのアクセス権で決まる。

おまけ



libvirt の話

- lxc は libvirt のサポート対象に既になっている。
んだが、うまくいかない。
- システムコンテナを定義するところまではできた。



libvirt の話

```
(test.xml)
<domain type='lxc'>
  <name>test</name>
  <memory>500000</memory>
  <os>
    <type>exe</type>
    <init>/sbin/init</init>
  </os>
  <vcpu>1</vcpu>
  <clock offset='utc' />
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/lib/libvirt/libvirt_lxc</emulator>
    <filesystem type='mount'>
      <source dir='/var/cache/lxc/debian/rootfs.test' />
      <target dir='/' />
    </filesystem>
    <interface type='bridge'>
      <source bridge='br0' />
      <target dev='veth_99999' />
      <mac address='00:11:22:33:44:55' />
    </interface>
    <console type='pty' />
  </devices>
</domain>
```

libvirt の話

```
$ sudo virsh --connect lxc:/// define text.xml
$ sudo virsh --connect lxc:/// start lxc
$ sudo virsh --connect lxc:/// list
```

Id	Name	State
3	test	no state

なんでやねん。

lxc-* コマンドでの運用を考える。

- lxc-* のコマンドは、1 コンテナに対するアクセスしかできない。
- 全体を把握するには不便。
- lxc-ls コマンドあたりを活用すると便利。



lxc-*コマンドでの運用を考える。

```
#!/bin/sh

for i in `lxc-ls`
do
if (lxc-info -n $i | grep -q "STOPPED" ) then
    echo -en $i"\t:"
    lxc-start -n $i -d
    test $? -eq 0 && echo starting ok.
    sleep 3
fi
done

exit 0
```

まとめ

- lxc とは コンテナ。
- lxc を導入してみる。ほとんど手間いらず。
- lxc の仕組み。cgroup の具体的なサンプル。
- libvirt はもうちょい先かな。

