Система управления контейнерами LXD

Денис Пынькин

1 октября 2016 г.

Linux Containers

Проект Linux Containers (`https://linuxcontainers.org`) уже долгое время занимается развитием набора утилит LXC, для управления локальными контейнерами.

При этом, основное отличие LXC от других подобных систем, это создание и управление контейнерами уровня операционной системы.

Недавнее прошлое

LXC v1.x

Набор средств для управления контейнерами на одном хосте:

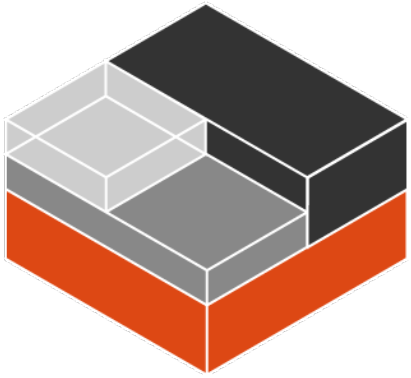
- библиотека
- утилиты управления
- скрипты запуска сервисов lxc и lxc-net
- темплейты (скрипты) для создания контейнеров

Настоящее

Проект состоит из 4 основных частей:

- `LXC` – ядро системы, обеспечивающее низкоуровневое управление контейнерами;
- `LXD` – "высокоуровневая" часть системы управления, предоставляющая единообразное API управления, а также утилиты командной строки;
- `CGManager` – демон для управления контролем групп, позволяющий создавать и использовать непривилегированные контейнеры;
- `lxcfs` – файловая система на базе FUSE, "скрывающая" реальные подсистемы `procfs`, `sysfs` и `cgroupfs`, изолируя доступ к ним от программ, работающих внутри контейнеров.

LXD



- Демон управления LXD
  - контейнерами
  - снапшотами
  - образами систем
  - профилями
- REST API
- CLI
- nova-плагин

CLI

lxc

- Одна утилита для управления
- Единообразное управление локальной и удаленными системами
- Взаимодействие с хранилищами образов

Интеграция с LXD

- Openstack
- Juju

1. Обзор
2. Установка и настройка
3. Images
4. Контейнеры
5. Примеры

Зависимости

- пакет `shadow-submap` – поддержка непривилегированных контейнеров
- `criu` – stateful-снапшоты, live-миграция
- `btrfs-progs` – поддержка бакенда `btrfs`
- `lvm2` – поддержка бакенда `LVM`

Установка

1. `apt-get install lxd`
2. `usermod -v 100000-165535 -w 100000-165535 root`
3. `lxd init`

Сеть

Не часть LXD!

LXD не пытается рассказать вам, что вам нужно делать со своей сетью.

Сеть

Не часть LXD!

LXD не пытается рассказать вам, что вам нужно делать со своей сетью.

bridge

Для разработчиков рекомендуется использовать локальный `bridge` интерфейс, на котором должны работать сервисы `dns` и `dhcp`. При желании можно настроить файерволл и машрутизацию,

Сеть

bridge

Для разработчиков рекомендуется использовать локальный `bridge` интерфейс, на котором должны работать сервисы `dns` и `dhcp`. При желании можно настроить файерволл и машрутизацию,

Для ленивых

Использовать сервис (скрипт) `lxd-bridge` либо `lxc-net`

Storage

Хранилище образов, контейнеров и снапшотов

- ZFS
- LVM
- btrfs
- Simple directory

Storage

Хранилище образов, контейнеров и снапшотов

- ZFS
- LVM
- btrfs
- Simple directory

Для ленивых

Использовать скрипт `lxd-setup-lvm-storage`

btrfs backend

На тему "стабильности" btrfs:

```
lxd[17632]: error: listen unix /var/lib/lxd/devlxd/sock: \
  bind: no space left on device
```

1. Обзор
2. Установка и настройка
3. Images
4. Контейнеры
5. Примеры

Образы систем

Образ системы

Архив с упакованным корнем системы и метаданными решает проблему неподготовленных пользователей.

Типы образов:

- `unified` – все в одном, хорошо подходит для распространения готового продукта
- `split` – раздельно metadata и rootfs, что очень удобно при разработке

metadata

- Описание образа
- Темплейты файлов (в формате pongo2)

Пример минимального файла-описания `metadata.yaml`:

```
architecture: "x86_64"
creation_date: 1470920887
properties:
  architecture: "x86_64"
  description: "Custom development image"
  os: "fedora"
  release: "1.2.3"
templates:
```

Хранилища образов

Протоколы

- lxd
- simplestreams

Хранилища по-умолчанию:

```
# lxc remote list
+-----------------+-----------------------------------------+---------------+-
|      NAME       |                   URL                   |   PROTOCOL    |
+-----------------+-----------------------------------------+---------------+-
| images          | https://images.linuxcontainers.org:8443 | lxd           |
+-----------------+-----------------------------------------+---------------+-
| local (default) | unix://                                 | lxd           |
+-----------------+-----------------------------------------+---------------+-
| ubuntu          | https://cloud-images.ubuntu.com/releases | simplestreams |
+-----------------+-----------------------------------------+---------------+-
| ubuntu-daily    | https://cloud-images.ubuntu.com/daily   | simplestreams |
+-----------------+-----------------------------------------+---------------+-
```

1. Обзор
2. Установка и настройка
3. Images
4. Контейнеры
5. Примеры

Пример использования контейнера

Запуск

```
lxc launch images:ubuntu/yakkety/amd64
```

Использование

```
lxc exec container /bin/bash
```

Остановка

```
lxc stop container
```

Экспорт контейнера в имидж

```
lxc publish container -alias test
lxc image edit test
```

Типы контейнеров

- постоянные
- ”эфемерные”
- привилегированные
- непривилегированные

Конфигурация и профили

Конфигурация контейнера

- Метаданные – имя, описание
- Конфигурация – настройки разрешений и ограничений [a]
- Устройства – диски, сетевые интерфейсы

[a] в том числе прямые вставки для LXC

Профили

Профиль – это часть конфигурации, которая вынесена в отдельную "библиотеку"и может использоваться множеством контейнеров

1. Обзор
2. Установка и настройка
3. Images
4. Контейнеры
5. Примеры

«Багофичи»

swapfile

Созданный и активированный swapfile на сравнительно старом ядре приводит к невозможности удаления контейнера.

/dev/pts

Привилегированный контейнер. Запускаем от рута:

```
mount /dev/pts -o ro,remount
```

и внезапно *хостовый* /dev/pts становится R/O

LXC в chroot

```
mount("", "/", NULL, MS_REC|MS_SLAVE, NULL)
```

не работает в chroot без монтирования корня для chroot

LXC вместо KVM

Решаемые задачи

- Изоляция хостовой системы от некоторых процессов.
- Лимитирование потребления ресурсов.

Занимательные факты

Связка `libvirt` и `LXC` – могут быть проблемы с loopback дисками при потере драйвера

LXC для сборки продукта

Решаемые задачи

- Унификация сборочного окружения
- Запуск более одного процесса сборки на хосте

Занимательные факты

Каждый разработчик считает, что именно его подход – лучший!

OBS в LXD

Решаемые задачи

- Быстрое прототипирование
- Нахождение проблем сборки в пакетах

Занимательные факты

Понадобилось мне как-то добавить в продукт порядка 150 перловых пакетов...

LXC для запуска продукта

Решаемые задачи

- Быстрое прототипирование и проверка новых фич
- Полноценная работа с сетью

Занимательные факты

Понадобилось добавить поддержку новой архитектуры

TODO

Интегрировать с LXD и осчастливить QA

Вопросы?

1. Обзор
2. Установка и настройка
3. Images
4. Контейнеры
5. Примеры