LES CONTENEURS SOUS LINUX
UN ÉTAT DE L'ART

Grégory Colpart & Jérémy Lecour – VVT 2018

VISION SUBJECTIVE

```
$ who
Jérémy Lecour <jlecour@evolix.fr>
Grégory Colpart <reg@evolix.fr>

$ whois Evolix
EVOLIX-AS : AS 197696

$ man Evolix
Open Source managed hosting provider

$ uptime
up 14 years, 20 users

$ whereis Evolix
/fr/Marseille, /fr/Aix, /fr/Paris, /ca/Montréal

$ top
Linux/BSD servers: 800, customers: 120
```

- Infogérance / Hébergement dédié et cloud / Conseil et Formation
- Sysadmins à Marseille et Montréal = 24/7
- Linux, infra web, HA, virtualisation, conteneurs, Ansible
- Clients : agences web, SaaS, médias

CONTEXTE / HISTORIQUE

VIRTUALISATION

- VMWare, KVM, Xen, Virtualbox…
- émulation du matériel
- indépendance de l'OS virtualisé
- para-virtualisation

ISOLATION DE PROCESSUS

- enfermer le processus dans une prison (chroot)
- accès limité au filesystem
- technique ancienne
- OS/noyau homogène

CONTENEURS

- utilisent les "namespaces"
- partitionnement des ressources du Kernel

CARACTÉRISTIQUES ET TECHNOLOGIES

RÉSEAU

- NAT ou bridge
- partage avec l'hôte
- redirection de ports

STOCKAGE

- montages classiques
- accès au filesystem de l'hôte

CGROUPS

- limites de ressources
- priorisation
- mesure d'usage
- contrôle d'exécution

CONTENEURS PRIVILÉGIÉS, OU NON

- privilégié : accès total
- non privilégié : accès limité

TYPES DE CONTENEURS

- système : init et arbre de processus
- applicatif : un seul processus

LXC

- LinuX Container
- utilise les cgroups et namespaces
- très facile à manipuler

CRÉATION D'UN CONTENEUR

```
# apt-get install lxc
# lxc-create --template download --name foo
Setting up the GPG keyring
Downloading the image index
[…]

Distribution: debian
Release: stretch
Architecture: amd64

Downloading the image index
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created an Debian stretch amd64 (20180611_05:25) container.

To enable SSH, run: apt install openssh-server
No default root or user password are set by LXC.
```

LISTER, DÉMARRER, S'ATTACHER À UN CONTENEUR

```
# lxc-ls
foo

# lxc-start --name foo

# lxc-info --name foo

# lxc-attach --name foo
```

UNE VRAIE CONSOLE, SI BESOIN

```
# lxc-console --name foo

Connected to tty 1
Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself

Debian GNU/Linux buster/sid foo pts/0

foo login:
```

UTILISATION DE SNAPSHOTS

```
# lxc-stop --name foo
# lxc-snapshot --name foo

# du -sch /var/lib/lxc/foo/snaps/snap0/
354M    /var/lib/lxc/foo/snaps/snap0/

# lxc-snapshot --name foo --list
snap0 (/var/lib/lxc/foo/snaps) 2018:06:12 03:24:48
snap1 (/var/lib/lxc/foo/snaps) 2018:06:12 03:26:24

# lxc-snapshot --name foo --restore snap1
```

LIMITER LES RESSOURCES ALLOUÉES (À CHAUD)

```
# lxc-cgroup --name foo cpuset.cpus 0

# lxc-cgroup --name foo memory.limit_in_bytes 4G
```

CHACUN SES PROCESSUS

```
# ps auwx
root         1  […] /sbin/init
root      5310  […] /sbin/init
root      2658  […] /lib/systemd/systemd-journald
root       389  […] /lib/systemd/systemd-journald
root      1039  […] /lib/systemd/systemd-logind
root      2704  […] /lib/systemd/systemd-logind
systemd+  2664  […] /lib/systemd/systemd-networkd
systemd+  2712  […] /lib/systemd/systemd-resolved
systemd+   719  […] /lib/systemd/systemd-timesyncd
root       420  […] /lib/systemd/systemd-udevd
message+  1045  […] /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfil
message+  2698  […] /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfil
```

STOCKAGE

- dir – stockage au sein d'un dossier classique
- lvm – un volume logique par conteneur
- btrfs, zfs – exploite les fonctionnalités natives
- loop – une partition dédiée
- rbd – un block-device Ceph

LXD

- s'appuie sur LXC (liblxc)
- simplifie l'interface et les outils
- fonctionnalités plus avancées

FONCTIONNALITÉS MARQUANTES

- Démon en Go avec API REST
- Sécurisé par défaut
- permet une gestion de cluster
- https://linuxcontainers.org/lxd/try-it/

DÉMO LXD

```
# lxc list
+------+---------+-----------------------+-----------------------------------------------+
| NAME |  STATE  |         IPV4          |                     IPV6                      |
+------+---------+-----------------------+-----------------------------------------------+
| toto | RUNNING | 10.219.150.156 (eth0) | 2001:470:b368:1070:216:3eff:fe96:da14 (eth0)  |
+------+---------+-----------------------+-----------------------------------------------+

# free -m
              total        used        free      shared  buff/cache   available
Mem:            256          26          70         189         158         229

# lxc config set toto limits.memory 128MB
# lxc exec toto -- free -m
              total        used        free      shared  buff/cache   available
Mem:            128           5          89         189          33         122
```

DOCKER

- Initialement basé sur LXC
- Orienté conteneurs applicatifs
- fonctionnalités beaucoup plus avancées

FONCTIONNALITÉS MARQUANTES

- Démon en Go avec API REST
- Gestion des images

GESTION DES IMAGES

- Dockerfile
- registry / Dockerhub
- Orchestration

DOCKERFILE

```
FROM debian:stretch

MAINTAINER Evolix

ENV DEBIAN_FRONTEND noninteractive
RUN apt-get update \
    && apt-get install -y --no-install-recommends apache2 \
    && rm -rf /var/lib/apt/lists/*

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

```
# docker run debian:stretch
Unable to find image 'debian:stretch' locally
stretch: Pulling from library/debian
cc1a78bfd46b: Pull complete
Digest: sha256:de3eac...
Status: Downloaded newer image for debian:stretch

# docker images
REPOSITORY TAG     IMAGE ID     CREATED     SIZE
debian     stretch 8626492fecd3 7 weeks ago 101 MB

# docker ps -a
CONTAINER ID IMAGE          COMMAND CREATED       STATUS                    PORTS NAMES
f302ad884767 debian:stretch "bash"  5 minutes ago Exited (0) 5 minutes ago       quirky_
```

DOCKERS

- docker compose
- Orchestration avec Swarm / Kubernetes
- K8S… futur standard ?

SURCOUCHES

LIBVIRT

- démon de gestion d'hyperviseurs
- KVM, Xen, VMWare, QEMU, LXC…
- nombreux bindings (Python, Ruby, Perl…)
- interfaces graphiques ou CLI

VAGRANT

- automate de gestion de VM
- VirtualBox, VMWare, LXC, libvirt…
- orienté "développeur"
- facilite le provisionnement (Ansible…)

SÉCURITÉ

- séparation imparfaite
- garder le contrôle des images
- limiter les privilèges au maximum

AU FINAL, LES CONTENEURS

- plus légers, performants, simples que des VM
- de l'usage très basique à la plateforme complète
- une histoire ancienne qui continue de s'écrire

MERCI