Conteneurs

M1 RÉSEAUX & TÉLÉCOMS – RT0702

OLIVIER FLAUZAC

Isolation

Isolation des exécutions

Gestion des contextes

- Définition d’éléments systèmes spécifiques
- Mémoire propre
- CPU / part de CPU ...

Application des contextes à la demande

Mise en place de multi-instances

Plusieurs degrés d'Isolation

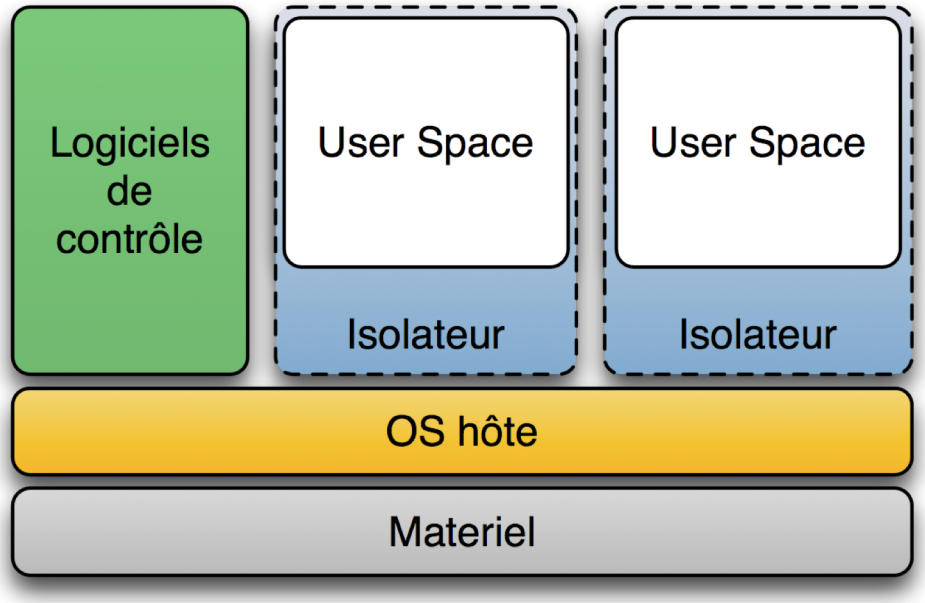
Exploitation cohérente : Linux dans Linux

Schéma global de l’isolation

Couche matérielle

Système d'exploitation

Environnement confinés



Isolateur

Mécanisme système assurant l'indépendance de plusieurs exécutions

Implique la gestion d'environnement séparés

Mise en place d'un *super scheduler*

Protection des ressources des éléments isolés

Nécessité de définir

- La base commune entre les environnements isolés
- La base commune entre les environnements isolés et le système hôte
- Les éléments spécifiques à chaque système isolé
- Les éléments spécifiques à l'hôte

Différents degrés d'isolation

Cas du système de fichiers

Nécessité d'installer un *système de fichiers local*

- *Debootstrap*

Système de fichiers de l'invité accessible depuis l'hôte

Système de fichiers de l'hôte inaccessible depuis l'invité

Chroot

Isolation de ressources disque

Solution le plus ancienne (1979 UNIX V)

Outil GNU élément des *coreutils*

Isolation par changement de racine

Modification du répertoire de la racine pour un processus

Permet de limiter un processus à un ensemble de répertoires

Etendu par les BSD *jails*

Linux V-Server

Isolation de contexte de sécurité

- Mémoire
- Disque, Quota
- Routage

Patch du noyau linux

- Nécessite une recompilation du noyau
- Toujours actif (*kernel* 4.9.113)

Définition d'espaces utilisateurs : *user spaces*

Partage du noyau entre les *Vserver* et l'hôte

Utilisable pour exécuter des serveurs

OpenVz

Généralités

`http://www.openvz.org`

Isolateur niveau noyau

Hôte et invité basés sur linux

Hautes performances

- environ 3 % de perte de performance

Basé sur Virtuozzo

Intégration dans un noyau 2.6

intégration en cours dans un noyau 3.x

Noyau spécifique

Fonction de virtualisation

Isolement

Gestion des ressources

Noyau 2.6

Jeu de commandes utilisateur

- pilotage des espaces

- pilotage des exécutions

Isolation et environnement spécifiques

Eléments virtualisés

- fichiers
- utilisateurs / groupes
- processus
- réseau
- périphérique
- messages système

Gestion au niveau noyau

Gestion au niveau noyau

Isolation : éviter d'empiéter sur les autres

Configuration des différents éléments

Eléments gérés

- quota disque double niveau
  - limite totale
  - limite utilisateur

scheduling fair CPU

- ordonnancement entre les environnements / hôtes
- ordonnancement interne à chaque environnement

Fonctionnalités

Scalabilité

- Jusqu'à 64 processeurs
- Jusqu'à 64 Go de mémoire
- Environnement virtuel extensible à la totalité de la machine

Densité

- Un grand nombre de machines exploitables en parallèle

Gestion individuelle

- Gestion des droits sur chaque machine

Templates

Principe

- Installations *pré-packées*
- Mode opératoire
  - Téléchargement d'une archive maître
  - Instanciation et création des invités
  - Autant d'instances indépendantes que voulu

Exploitation

Plusieurs types

- distributions brutes
  - `http://download.openvz.org/template/precreated/`
- distributions logicielles
  - `http://www.turnkeylinux.org`

Configuration du réseau

Différents modes

Venet
- mode réseau virtuel

Veth
- interface virtuelle

Permet la gestion différenciée des accès réseau

Sépare les aspects utilisation et sécurité

VeNet

Mode de base

Adresse attribuée par le serveur OpenVZ

- pas de configuration interne au conteneur

Pas d'accès au *broadcast*

Pas d'accès direct à la couche *ethernet*

Interface réseau *switchée* par le serveur OpenVZ

Administration complète par le gestionnaire du serveur OpenVZ

Veth

Accès au *broadcast*

Accès direct à la couche *ethernet*

Bridge possible avec l'interface du serveur OpenVZ

Possibilité de configuration interne au conteneur (DHCP)

Carte virtuelle avec adresse MAC

- adressage IPv6 possible

Mise en oeuvre

Ajout du dépôt openVz

Installation du noyau spécifique

- `linux-image-openvz-amd64`

Installation des outils spécifiques

- `Vzctl, vzquota, ploop, vzstats`

Mise en oeuvre

Modification de la configuration de `/etc/sysctl.conf`

```
# On Hardware Node we generally need
# packet forwarding enabled and proxy arp disabled
net.ipv4.ip_forward = 1
net.ipv4.conf.default.proxy_arp = 0
# Enables source route verification
net.ipv4.conf.all.rp_filter = 1
# Enables the magic-sysrq key
kernel.sysrq = 1
# We do not want all our interfaces to send redirects
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

Les templates

Téléchargement

- `http://openvz.org/Download/templates/precreated`
- `http://www.turnkeylinux.org`

Enregistrement dans

- `var/lib/vz/template/cache`

Possibilité de créer ses propres templates

Conteneurs openVz

Identifiants :

- Conteneurs identifiés pas le VEID
- 0 : noeud OpenVZ
- 1 - 100 : réservé par le système
- 101 - : Conteneurs utilisateur

Commande vzctl

- création / destruction `create / destroy`
- démarrage / arrêt `start / stop`
- exécution d'une commande dans un conteneur `exec`
- accès à un conteneur `enter`
- modification des propriétés d'un conteneur `set`

Création d’un conteneur

Création Minimale

- identifiant
- conteneur

Configuration générale

- limites (CPU, mémoire, disque ...)
- réseau

Exemple

```
vzctl create 105 --ostemplate debian-7.0-x86_64-minimal
vzctl enter 105
vzctl exec 105 apt-get install figlet
```

Modification des propriétés

Modification en cours d'exécution : `vzctl set VEID --param --value`

Modification et sauvegarde de la configuration : `vzctl set VEID --param --value --save`

- `--userpasswd user:pass`
- `--ipadd addr`
- `--hostname name`
- `--nameserver addr`
- `--cpus num`
- `--cpulimit num[\%]`
- `--diskspace num[:num]`
- `--ram bytes`
- `--swap bytes`

LXC

LXC : généralités

Isolation au niveau système d'exploitation

Basé sur `cgroups`

Partage :

- du noyau avec l'hôte
- d'une partie du disque

Virtualisation de l'environnement d'exécution :

- mémoire
- hiérarchie fichier
- processeur
- réseau

`https://linuxcontainers.org`

Cgroups

Control Groups

Fonctionnalité du noyau Linux (depuis 2.6.24)

Initié en 2006 chez Google

Ensemble de :

- fonctionnalités
- Contrôleurs

Utilisation par le biais de `libcgroup`

Fonctionnalités de cgroup

Limitation des ressources

- Mise en place du *sandboxing* ressources
- Mémoire
- Utilisation processeur
- Utilisation disque

Gestion de priorité

- Gestion d'un *scheduling* de ressources étendu
- Exploitation relative des du processeur
- Exploitation relative de la bande passante
- Extension *scheduler* à toutes les ressources
- Extension *scheduler* inter conteneurs

Fonctionnalités de cgroup

Gestion comptable

- Contrôle / mémorisation des ressources
- Enregistrement de la consommation : facturation possible

Isolation

- Mise en place d'espace de nommages des conteneurs
- Indépendance des conteneurs
- Blocage des interactions inter contrôleurs

Contrôle

- Arrêt / démarrage des contrôleur
- Mise en pause
- Prise / injection de *snapshots*

Mise en œuvre

Installation sur une Debian

Installation possible en virtuel

Installation sans modification du noyau

Installation

- lxc lxctl

Vérification

- lxc-checkconfig

Attention activation différent de utilisation !

```
Kernel configuration not found at /proc/config.gz;
searching...
Kernel configuration found at /boot/config-3.16.0-4-amd64
--- Namespaces ---
Namespaces: enabledUtsname namespace: enabled
Ipc namespace: enabledPid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled
--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: 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
Vlan: enabled
File capabilities: enabled
```

Commandes de base

Installation

- `lxc-create` : création d'un conteneur
- `lxc-start` : démarrage d'un conteneur

téléchargement si nécessaire

- `lxc-info` : information sur un conteneur
- `lxc-ls` : liste des Conteneurs
- `lxc-console` : création d'une console dans un contenu
- `lxc-attach` : exécution d'une commande dans un conteneur
- `lxc-stop` : arrêt d'un conteneur
- `lxc-destroy` : destruction d'un conteneur

Utilisation

```
root@debLxc:~# lxc-create -n ctn01 -t debian
... TELECHARGEMENT SI NECESSAIRE ...
root@debLxc:~# lxc-start -n ctn01 -d
root@debLxc:~# lxc-console -n ctn01

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

Debian GNU/Linux 8 ctn01 tty1

ctn01 login:

....
On quitte
```

```
root@debLxc:~# lxc-info -n ctn01
Name:           ctn01
State:          RUNNING
PID:            1140
CPU use:        0.12 seconds
BlkIO use:      0 bytes
root@debLxc:~# lxc-ls
ctn01
root@debLxc:~# lxc-ls --fancy ctn01
NAME   STATE    IPV4  IPV6  AUTOSTART
-------------------------------------
ctn01  RUNNING  -     -     NO
root@debLxc:~#
```

```
root@debLxc:~# lxc-attach -n ctn01 -- ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2  28120  4280 ?        Ss   20:52   0:00 /sbin/init
root          19  0.0  0.1  32960  3416 ?        Ss   20:52   0:00 /lib/systemd/systemd-journald
root          68  0.0  0.2  55164  5440 ?        Ss   20:52   0:00 /usr/sbin/sshd -D
root          73  0.0  0.0  12656  1796 tty2     Ss+  20:52   0:00 /sbin/agetty --noclear tty2 linux
root          74  0.0  0.0  12656  1744 tty4     Ss+  20:52   0:00 /sbin/agetty --noclear tty4 linux
root          75  0.0  0.0  12656  1756 tty3     Ss+  20:52   0:00 /sbin/agetty --noclear tty3 linux
root          76  0.0  0.1  63300  2948 tty1     Ss   20:52   0:00 /bin/login --
root          77  0.0  0.1  14228  2220 console  Ss+  20:52   0:00 /sbin/agetty --noclear --keep-baud
console 115200 38400 9600 vt102
root          95  0.0  0.1  21828  3716 tty1     S+   20:53   0:00 -bash
root         212  0.0  0.0  12656  1812 ?        Ss   21:02   0:00 /sbin/agetty --noclear tty5 linux
root         213  0.0  0.0  12656  1744 ?        Ss   21:02   0:00 /sbin/agetty --noclear tty6 linux
root         214  0.0  0.1  19092  2588 ?        R+   21:02   0:00 ps aux
```

Configuration /var/lib/lxc/nomConteneur/config

```
# Template used to create this container: /usr/share/lxc/templates/lxc-debian
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)
lxc.network.type = empty
lxc.rootfs = /var/lib/lxc/ctn01/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.mount = /var/lib/lxc/ctn01/fstab
lxc.utsname = ctn01
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0
```

Définition de limites

Types de limites
- Mémoire
- CPU
- Utilisation du disque

Mise en place des limites
- Au démarrage
  - dans la ligne de commande
  - dans un fichier de configuration
- En ligne lors de l'exécution

Attention à la configuration :

Gestion mémoire dans LXC
- Modification de /etc/default/grub
  - `GRUB_CMDLINE_LINUX="cgroup_enable=memory »`
- Ne pas oublier la mise à jour de GRUB !!!

Limites en cours d’exécution

```
lxc-cgroup -n ctn01 cpuset.cpus 0,1

lxc-cgroup -n ctn01 memory.soft_limit_in_bytes 268435456
lxc-cgroup -n ctn01 memory.limit_in_bytes 53687091
```

Limites en fichier de configuration

```
lxc.cgroup.cpuset.cpus=0,1

lxc.cgroup.memory.soft_limit_in_bytes=268435456
lxc.cgroup.memory.limit_in_bytes=53687091
```

Connexion au réseau

Définition dans le fichier de configuration

Types de connexions

- empty : pas de réseau
- **phys** : lien direct avec l'interface de l'hôte
- veth : mise en place d'un pont
- Vlan : intégration de l'invité dans un vlan
- macvlan : mise en place d'un VEPA : *Virtual Ethenet Port Agreggator*

Configuration physique

Pas de configuration sur l'hôte

Configuration de l'invité

Perte du réseau sur l'hôte !

attention à la connexion SSH !!!

Fichier de configuration

```
# Template used to create this container: /usr/share/lxc/templates/lxc-debian
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)
lxc.network.type = empty
lxc.rootfs = /var/lib/lxc/ctn01/rootfs
# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf
# Container specific configuration
lxc.mount = /var/lib/lxc/ctn01/fstab
lxc.utsname = ctn01
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0

lxc.network.type = phys
lxc.network.link = eth0
```