~~CONFIDENTIEL DÉFENSE~~ ~~SPECIAL FRANCE~~

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PREMIER MINISTRE

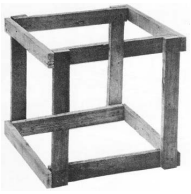
Secrétariat général de la défense et de la sécurité nationale

Agence nationale de la sécurité des systèmes d'information

DÉCLASSIFIÉ
par décision n°15699/ANSSI/SDE/ST/LAM
du 18 juillet 2018

Documentation CLIP
1103

Guide d'installation de l'environnement de développement



ANSSI, 51 boulevard de la Tour Maubourg, 75700 Paris 07 SP.

~~CONFIDENTIEL DÉFENSE~~

HISTORIQUE

Révision	Date	Auteur	Commentaire
2.3	07/02/2014	Tony Cheneau, Hugo Chargois	Ajout d'une procédure de mise à jour du SDK, réorganisation de l'ordre du texte et corrections de certaines commandes
2.2	14/02/2013	Mickaël Salaün	Utilisation de plusieurs SDK et correction sur les chemins de sources
2.1	31/01/2013	Benjamin Morin	Modifications de forme et précisions
2.0	07/01/2013	Hugo Chargois, Mickaël Salaün	Nouvelle version de mise en place de l'environnement de développement avec LXC
1.0	26/06/2008	Olivier Levillain	Version initiale (DVD-ROM d'installation du poste de développement)

Table des matières

Résumé

Ce document décrit la procédure d'installation d'un environnement de développement CLIP.

L'environnement de développement est un système d'exploitation basé sur Gentoo, prévu pour fonctionner dans un conteneur de la solution de virtualisation légère LXC. À noter que cette configuration n'est pas obligatoire. Il est également possible d'exécuter l'environnement de développement dans un *chroot*.

1 Prérequis

1.1 Rôles intervenant dans la génération

Les différentes opérations décrites dans le présent document sont réalisées par un utilisateur du réseau de développement CLIP ayant le profil *Développeur* ([CLIP-DCS-120]).

1.2 Installation de l'OS hôte

Il faut initialement installer un poste avec une distribution GNU/Linux quelconque (Gentoo, Debian, Ubuntu). Cette documentation est prévue pour un système hôte Debian. Penser à créer une partition pouvant contenir au moins 100 Go en plus du système hôte.

Un miroir Debian est disponible sur le réseau de développement ; celui-ci peut être renseigné dans le fichier `/etc/apt/sources.list` :

```
deb https://clip.ssi.gouv.fr/debian/ sid main non-free contrib
```

La configuration réseau se fait par DHCP.

Ce poste sera présent sur un réseau CD, il convient donc de prendre les dispositions nécessaires pour qu'il soit sécurisé (règles iptables, etc.).

2 Procédure d'installation de l'environnement de développement

Afin d'avoir des branches de stabilisation, il est fortement recommandé d'avoir autant de versions du SDK que de versions à maintenir (i.e. une version stable et une version de développement).

Les étapes de mise en place d'un *clip-sdk-unstable* sont identiques à celles du *clip-sdk-stable*. Seule la version des logiciels et bibliothèques installées doivent varier. Pour cela, il faut bien veiller à maintenir à jour chaque SDK à partir de la branche *clip-int* souhaitée (e.g. *stable-4.3.5* et *clip4*) grâce à un lien symbolique `/opt/clip-int` correct.

Les fichiers de configuration des conteneurs sont similaires sauf pour le nom, les chemins de fichiers et le réseau.

2.1 Installation de LXC

Suivre la procédure dépendante de la distribution utilisée. Pour consacrée à la distribution Debian est disponible à l'adresse `http://wiki.debian.org/LXC`.

2.2 Création du conteneur LXC

Le fichier de configuration pour LXC est disponible sur le SVN (`clip-dev/clip-devutils/branches/clip4/-share/lxc/clip-sdk-stable.conf` pour la version stable de CLIP). Pour créer le conteneur LXC avec cette configuration :

```
# lxc-create -f clip-sdk-stable.conf -n clip-sdk-stable
```

Cela crée le répertoire */var/lib/lxc/clip-sdk-stable* et y copie le fichier de configuration.

Modifiez le fichier de configuration du conteneur créé (*/var/lib/lxc/clip-sdk-stable/config*) pour renseigner le véritable point de montage de clip-src (qui doit contenir une copie valable des dépôts Subversion).

2.3 Installation du *stage-5* clip-sdk

1. Récupérer l'archive du système clip-sdk
2. La décompresser (en *root*) dans */var/lib/lxc/clip-sdk-stable/rootfs* [1].
3. Si *clip-layout/baselayout-sdk[clip-dev-lxc]* n'est pas installé dans le *stage-5* effectuer à la main les actions de l'ebuild (création des périphériques/dossiers, gestion des services et options par défaut, notamment : `mkdir /dev/pts`).
4. Changer le mot de passe root, on peut pour cela faire un *chroot* (si besoin, utiliser *linux32* puis un *passwd*).
5. Si ce n'est pas déjà fait par OpenRC, décommenter et remplacer la variable *rc_sys* du fichier */var/lib/lxc/clip-sdk-stable/rootfs/etc/rc.conf* par :

```
rc_sys="lxc"
```

2.4 Configuration du réseau

La façon la plus simple de configurer le réseau consiste à ne rien renseigner dans le fichier de configuration du conteneur LXC, auquel cas la machine hôte et le conteneur partagent le même *namespace* réseau (le conteneur voit les mêmes interfaces réseau que l'hôte). Il est alors nécessaire de faire écouter les services qui s'exécutent dans le conteneur sur des ports différents de ceux exécutés par l'hôte (par exemple, mettre un serveur SSH du conteneur en écoute sur le port 2200 au lieu de 22).

La suite de cette section décrit la configuration recommandée (surtout dans le cas d'utilisation de services réseaux tel que SSH), dans laquelle une configuration réseau virtuelle (`veth`) est dédiée au conteneur. Il est possible de passer directement à la section 2.5 si la configuration réseau décrite précédemment est utilisée.

À noter que le fichier de configuration du conteneur LXC [2] fourni à titre d'exemple dans le SVN [3] et le SDK [4] configure le réseau en mode `veth`.

Lors du lancement du conteneur LXC, une interface est créée sur le système hôte (elle possède un nom aléatoire *vethXXX*), et le conteneur dispose aussi d'une interface (*lxc0*). Les interfaces sont reliées par un câble virtuel.

Sur le système hôte, l'interface correspondant au conteneur sera associée à une interface *bridge* qu'il est nécessaire de configurer.

2.4.1 Sur la machine hôte

La configuration réseau sur la machine hôte dépend de la distribution considérée. Pour Debian, il est nécessaire d'installer le paquet `bridge-utils`, avant d'éditer le fichier */etc/network/interfaces* et y ajouter les lignes :

```
auto br0
iface br0 inet static
    bridge_ports none
    address 172.16.0.254
    netmask 24
```

N'oubliez pas de redémarrer votre réseau (`service networking restart` sous Debian).

1. On peut également décompresser l'archive ailleurs et faire un symlink à la place.
2. Fichier de configuration du conteneur : `/var/lib/lxc/clip-sdk-stable/config`
3. Modèle de configuration à jour sur le dépôt Subversion *clip-dev* : `clip-devutils/share/lxc/`
4. Modèle de configuration disponible dans un SDK : `/usr/share/clip-devutils/lxc/`

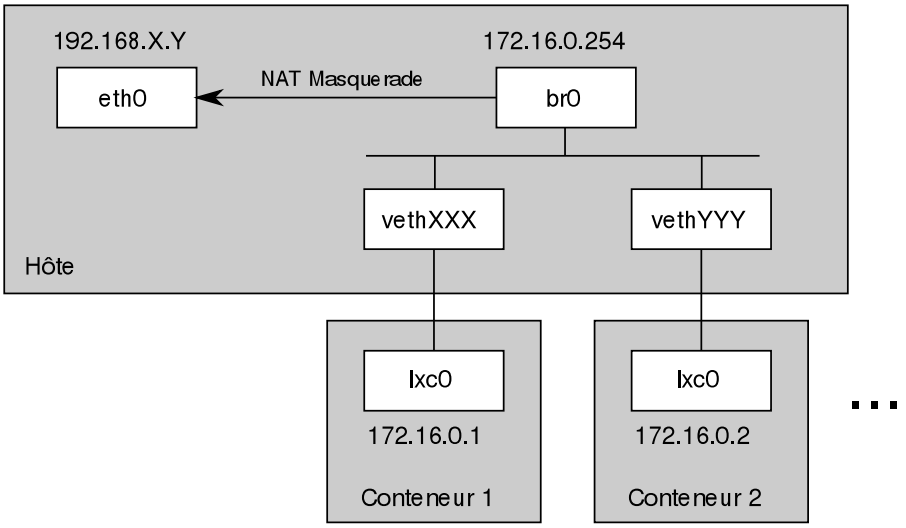


FIGURE 1 – Modèle du réseau entre l'hôte et les conteneurs

2.4.2 Sur le conteneur

La configuration de l'interface (IP, masque, passerelle) se fait automatiquement au démarrage en fonction des valeurs du fichier de configuration LXC pour le conteneur. Il faut cependant penser à attribuer une adresse IP différente à chaque conteneur (ligne *lxc.network.ipv4* dans le fichier de configuration) quand il est prévu d'en utiliser plusieurs simultanément.

La configuration de la résolution de noms de domaine consiste à recopier le fichier `/etc/resolv.conf` de l'hôte.

```
# cp /etc/resolv.conf /var/lib/lxc/clip-sdk-stable/rootfs/etc/
```

2.4.3 Routage

La configuration des interfaces telle que décrite suffit pour que l'hôte et le(s) conteneur(s) communiquent. La commande `iptables` suivante configure une règle NAT afin que le conteneur puisse communiquer avec le réseau de l'hôte :

```
# iptables -t nat -A POSTROUTING -s 172.16.0.0/24 -o eth0 -j MASQUERADE
```

Il est en outre nécessaire d'activer le *forwarding* :

```
# sysctl net.ipv4.ip_forward=1
```

Afin que ces règles persistent après un redémarrage, il est préférable (sous Debian) d'utiliser `iptables-persistent` (cf. `http://wiki.debian.org/iptables`) pour les règles `iptables`, et de décommenter la ligne idoine du fichier */etc/sysctl.conf* pour le *forwarding* :

```
net.ipv4.ip_forward=1
```

Une alternative est d'utiliser le fichier */etc/rc.local* pour stocker des commandes qui seront lancées à chaque démarrage.

2.5 Utilisation du conteneur

Pour lancer le conteneur :

```
# lxc-start -n clip-sdk-stable
```

Pour obtenir une console :

```
# lxc-console -n clip-sdk-stable
```

Ctrl-a q permet de se déconnecter de la console.
La Section 2.9 décrit la connexion au conteneur via SSH.

2.6 Récupération de *clip-int*

Vérifier dans un premier temps la bonne configuration du compte développeur en se connectant au Bugzilla [5].

Récupérer dans un second temps les arbres portage et les distfiles associés (ici dans */mnt/clip-src/clip-int/branches*) :

```
$ svn co https ://clip.ssi.gouv.fr/clip-int/branches/clip4/ /mnt/clip-src/clip-int/branches
```

Il faut ici veiller à remplacer *clip4* par la version de l'environnement qui vous intéresse. Il est également possible de stocker les sources dans un répertoire autre que */mnt/clip-src*. Dans ce cas, il faudra modifier les chemins dans le fichier de configuration lxc (`/var/lib/lxc/clip-sdk-stable/config`).

Les sources de CLIP (arbres *portage* et *distfiles*) peuvent être exposés par l'hôte dans le conteneur par le biais d'un montage *bind*. La directive suivante du fichier de configuration du conteneur crée un partage entre le répertoire `/mnt/clip-src/` coté hôte et conteneur (à supposer que le conteneur soit présent dans le répertoire `/var/lib/lxc/clip-sdk-stable/rootfs/`) :

```
lxc.mount.entry = /mnt/clip-src \
                  /var/lib/lxc/clip-sdk-stable/rootfs/mnt/clip-src \
                  none defaults,bind 0 0
```

Il est préférable de télécharger les sources de CLIP depuis le conteneur.

Au sein du conteneur, créer le lien symbolique pointant vers la version de *clip-int* utilisée par *clip-sdk-stable* :

```
# ln -s /mnt/clip-src/clip-int/branches/clip4 /opt/clip-int
```

2.7 Récupération des sources d'un projet

De la même façon, les source d'un projet interne CLIP peuvent être accessibles dans les différents environnements de développement.

Le téléchargement des sources d'un projet interne se fait de la façon suivante :

```
$ svn co https ://clip.ssi.gouv.fr/clip-dev/<project>/branches/clip4 /mnt/clip-src/clip-dev/<project>
```

2.8 Mise à jour de l'environnement de développement

Il est fortement recommandé de garder l'environnement de développement à jour. Pour se faire, il suffit de lancer les mises à jour via les commandes contenu dans le script dédié :

```
# gentoo-upkeep.sh
```

5. https://clip.ssi.gouv.fr/

2.9 Utiliser OpenSSH pour se connecter au conteneur

Il est également possible d'utiliser un serveur SSH pour ouvrir des sessions avec un conteneur. Exécutée dans le conteneur, la commande suivante active l'exécution automatique du serveur SSH au lancement du conteneur (il peut être nécessaire d'installer *net-misc/openssh*) :

```
# rc-update add sshd default
```

Il est recommandé d'utiliser des bi-clés SSH pour se connecter en tant que simple utilisateur aux différents comptes : utilisateur non privilégié pour développer et utilisateur *root* pour générer les paquetages et mettre à jour le SDK.

Références

[CLIP-DCS-120] *Règles et procédures de développement CLIP, CLIP_MAP-12000-007-DCS.*