

CoreOS: Transpiler de configuration de conteneur Linux

Table of Contents

- [CoreOS: Transpiler de configuration de conteneur Linux](#)
- [Installation et configuration](#)
 - [Binaires pré-construits](#)
 - [Construire à partir de la source](#)
- [Utilisation](#)

Config Transpiler ("ct" en abrégé) est l'utilitaire chargé de transformer une configuration Container Linux en un fichier JSON, qui peut être fourni à une machine Container Linux lors de son premier démarrage pour provisionner la machine.

Installation et configuration

Binaires pré-construits

Le moyen le plus simple de commencer à utiliser ct consiste à télécharger l'un des fichiers binaires à partir de la page des versions sur GitHub.

On peut utiliser le script suivant pour télécharger et vérifier la signature de Config Transpiler:

```
# Spécifie la version de Config Transpiler
CT_VER=v0.6.1

# Spécifie l'architecture
# ARCH=aarch64 # Architecture 64 bits d'ARM
ARCH=x86_64

# Spécifie le système d'exploitation
# OS=apple-darwin # MacOS
# OS=pc-windows-gnu.exe # Windows
OS=unknown-linux-gnu # Linux

# Spécifie l'URL de téléchargement
DOWNLOAD_URL=https://github.com/coreos/container-linux-config-transpiler/releases/download

# Supprime les téléchargements précédents
rm -f /tmp/ct-${CT_VER}-${ARCH}-${OS} /tmp/ct-${CT_VER}-${ARCH}-${OS}.asc
```

```
/tmp/coreos-app-signing-pubkey.gpg

# Télécharge le binaire Config Transpiler
curl -L ${DOWNLOAD_URL}/${CT_VER}/ct-${CT_VER}-${ARCH}-${OS} -o /tmp/ct-
${CT_VER}-${ARCH}-${OS}
chmod u+x /tmp/ct-${CT_VER}-${ARCH}-${OS}

# Télécharge et importe la clé GPG de signature des applications CoreOS.
Curl https://coreos.com/dist/pubkeys/app-signing-pubkey.gpg -o /tmp/coreos-
app-signing-pubkey.gpg
Gpg2 --import --keyid-format LONG /tmp/coreos-app-signing-pubkey.gpg

# Télécharge et importe la clé GPG de signature d'une application CoreOS si
elle n'a pas déjà été importée.
curl https://coreos.com/dist/pubkeys/app-signing-pubkey.gpg -o /tmp/coreos-
app-signing-pubkey.gpg
gpg2 --import --keyid-format LONG /tmp/coreos-app-signing-pubkey.gpg
```

Construire à partir de la source

Pour compiler à partir des sources, le compilateur go doit être installé sur votre système.

```
git clone --branch v0.8.0
https://github.com/coreos/container-linux-config-transpiler
cd container-linux-config-transpiler
make
```

Le binaire ct sera placé dans ./bin/.

Utilisation

Ct est un outil qui consomme une configuration Container Linux et génère un fichier JSON pouvant être transmis à une machine Container Linux lors de son premier démarrage afin de configurer une nouvelle machine. Cette configuration permet à une machine de créer des utilisateurs, de formater le Système de fichiers racine, configurer le réseau, installer les unités systemd, etc.

Les configurations de Container Linux sont des fichiers YAML conformes au schéma ct.

À titre d'exemple simple, utilisons ct pour définir la clé SSH autorisée pour l'utilisateur principal sur une machine Container Linux.

```
passwd:
  users:
    - name: core
      ssh_authorized_keys:
```

```
- ssh-rsa AAAAB3NzaC1yc...
```

Dans ce fichier ci-dessus, la ligne `ssh-rsa AAAAB3NzaC1yc ...` contient notre propre clé publique `ssh` (qui est probablement le contenu de `~/.ssh/id_rsa.pub`).

Lorsqu'on donne ce fichier et le donnons à `ct`:

```
$ ./bin/ct --in-file example.yaml
{"ignition":{"version":"2.0.0","config":{},"storage":{},"systemd":{},"networkd":{},"passwd":{"users":[{"name":"core","sshAuthorizedKeys":["ssh-rsa AAAAB3NzaC1yc..."]}]]}}
```

On peut voir qu'il produit un fichier JSON, qui va certainement être pénible à lire/éditer (surtout quand on a des éléments multilignes comme des unités `systemd`). Il suffit simplement de fournir ce fichier à une machine `Container Linux`, l'utilitaire à l'intérieur de `Container Linux` qui reçoit ce fichier, saura quoi faire avec ce fichier.

La méthode selon laquelle ce fichier est fourni à une machine Linux conteneur dépend de l'environnement dans lequel la machine s'exécute:

- **Bare Metal** - Utiliser le paramètre de noyau `coreos.config.url` pour fournir une URL à la configuration, qui peut utiliser les schémas `http://`, `https://`, `tftp://` ou `s3://` pour spécifier `Config` ou le schéma `oem://` pour spécifier une configuration locale, enracinée dans `/usr/share/oem`.
- **PXE** - Utiliser les paramètres de noyau `coreos.config.url` et `coreos.first_boot=1` (dans le cas du tout premier démarrage PXE uniquement) pour fournir une URL à la configuration. L'URL peut utiliser les schémas `http://`, `https://`, `tftp://` ou `s3://` pour spécifier `Config` ou le schéma `oem://` pour spécifier une configuration locale, enracinée dans `/usr/share/oem`.
- **QEMU** - Ignition lit sa configuration à partir de `'opt/com.coreos/config'` sur le dispositif de configuration du micrologiciel QEMU.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=virtualisation:coreos-transpiler>

Last update: **2025/02/19 10:59**

