Utilisation basique de Singularity

Singularity est une technologie qui vous permet d'exécuter une application linux à l'intérieur d'un environnement isolé et reproductible, appelé conteneur, qui ne dépend que du noyau linux de la machine sur laquelle vous êtes. Un conteneur ressemble à une machine virtuelle, sauf qu'il n'embarque pas nécessairement un système d'exploitation au complet, ce qui lui permet de se lancer en quelque secondes et d'être plus léger.

Dans ce plongeon, nous allons nous familiariser avec le vocabulaire et apprendre à créer simplement quelques conteneurs à partir d'images déjà prêtes.

- prérequis : connaissance basique de Linux.
- plongeon préalable recommandé : Utilisation basique de Docker (Docker_01_Run/README.md)
- hauteur du plongeoir : 1m.
- maître(s) nageur(s) : Martin Souchal (https://annuaire.in2p3.fr/agents/Y249U291Y2hhbCBNYXJ0aW4sb3U9cGVvcGxlLGRjPWluMnAzLGRjPWZy/show), David Chamont (https://annuaire.in2p3.fr/agents/Y249Q2hhbW9udCBEYXZpZCxvdT1wZW9wbGUsZGM9aW4ycDMsZGM9ZnI=/show).

Introduction

Qu'est-ce qu'un conteneur ?

L'industrie utilise aujourd'hui massivement les machines virtuelles. Ces machines exécutent des applications au sein d'un système d'exploitation "invité", qui exploite un matériel "virtuel" émulé par le système d'exploitation "hôte" de la vraie machine.

L'isolation entre invité et hôte est excellente, mais se paie cher par l'émulation du matériel virtuel et l'exécution d'un système d'exploitation invité complet.

Les conteneurs peuvent être vus comme une variante plus légère : en exploitant plus directement les couches basses du système hôte (noyau), les conteneurs fournissent une isolation presque aussi forte que les machines virtuelles, à un cout en performance beaucoup moins élevé.

Qu'est-ce que Singularity ?

Il s'agit d'un ensemble d'outils qui facilitent l'écriture, le partage et le déploiement d'applications de calcul intensif au sein de conteneurs. Chaque conteneur contient l'environnement nécessaire à l'exécution d'une application, sur n'importe quelle machine exécutant un noyau linux. Singularity est compatible avec toutes les variantes de Linux et tous les planificateurs de tâches (job schedulers). Il intègre X11, MPI, et permet l'utilisation de l'infiniband et des GPGPUs. A la différence d'autres systèmes de conteneurs, Singularity ne demande aucun droit de type administrateur, aucun démon, ne virtualise pas le réseau et dialogue directement avec le système de fichiers de son hôte. Chaque conteneur est lancé et s'arrête en même temps que l'application qu'il encapsule.

Un peu de vocabulaire

Image

Comme dans le cadre des machines virtuelles, on appelle "image" une description statique de conteneur, une sorte de photographie de machine, que vous pouvez échanger avec vos collaborateurs, et à partir de laquelle on peut instancier et exécuter des conteneurs. Singularity possède son propre mécanisme d'images, mais peut aussi s'interfacer avec les images Docker.

Conteneur

Une machine virtuelle légère, chargée en mémoire, qui sert à lancer une application au sein d'un environnement isolé et reproductible. Un conteneur s'instancie à partir d'une image.

Registre

Entrepôt ou l'on stocke des images prêtes à l'emploi. Le registre central officiel de Singularity est consultable sur la toile à l'adresse https://singularity-hub.org/ (https://singularity-hub.org/).

Installation de singularity

```
git clone https://github.com/singularityware/singularity.git
cd singularity
git fetch --all
git checkout 2.6.0
./autogen.sh
./configure --prefix=/usr/local
make
sudo make install
```

L'aide de singularity est accessible via la commande suivante :

```
singularity help
USAGE: singularity [global options...] <command> [command options...] ...

GLOBAL OPTIONS:
    -d|--debug    Print debugging information
    -h|--help     Display usage summary
    -s|--silent   Only print errors
    -q|--quiet    Suppress all normal output
       --version  Show application version
    -v|--verbose  Increase verbosity +1
    -x|--sh-debug Print shell wrapper debugging information

GENERAL COMMANDS:
    help       Show additional help for a command or container
    selftest   Run some self tests for singularity install

CONTAINER USAGE COMMANDS:
    exec       Execute a command within container
    run        Launch a runscript within container
    shell      Run a Bourne shell within container
    test       Launch a testscript within container

CONTAINER MANAGEMENT COMMANDS:
    apps       List available apps within a container
    bootstrap  *Deprecated* use build instead
    build      Build a new Singularity container
    check      Perform container lint checks
    inspect    Display container's metadata
    mount      Mount a Singularity container image
    pull       Pull a Singularity/Docker container to $PWD

COMMAND GROUPS:
    image      Container image command group
    instance   Persistent instance command group


CONTAINER USAGE OPTIONS:
    see singularity help <command>

For any additional help or support visit the Singularity
website: https://www.sylabs.io/
```

Faire sa propre image

Dans le fichier ci-dessous, que nous appellerons `example.def`, nous partons d'une image Docker ubuntu, sur laquelle nous ajoutons les paquets `wget` et `build-essential`. Nous créons également un répertoire qui contiendra nos données. En fait, nous pouvons utiliser n'importe quelle commande bash dans ce fichier de bootstrap.

```
Bootstrap: docker
From: ubuntu:latest

%post
apt-get update && apt-get -y install wget build-essential
mkdir /data
```

Pour construire le conteneur correspondant à ce fichier de recette, utilisez la commande `build` :

```
sudo singularity build example.img example.def
```

Le fichier contenant le conteneur va s'appeller example.img. Pour partager ce conteneur, il suffit de copier ce fichier ou de le publier sur un registre.

Pour rentrer dans le conteneur, on lance un shell à l'intérieur :

```
singularity shell example.img
```

Pour aller plus loin

Si vous voulez rester dans l'eau, vous pouvez enchainer sur les deux TP Singularity suivants :

- Utilisation d'openMPI avec Singularity (https://gitlab.in2p3.fr/MaitresNageurs/EnBarque/Singularity_02_openmpi)
- Utilisation avancée de Singularity (https://gitlab.in2p3.fr/MaitresNageurs/EnBarque/Singularity_03_advanced)

Sortie de bain

Merci d'envoyer quelques commentaires aux maîtres nageurs :

- Environnement de travail opérationnel ? oui [ ], non [ ]
- Vous aviez les pre-requis ? oui [ ], non [ ]
- Comment jugez-vous la difficulté du plongeon ? trop simple [ ], adapté [ ], trop compliqué [ ]
- Durée du plongeon ? trop court [ ], 15-20 minutes [ ], trop long [ ]
- Pourquoi avez-vous choisi ce plongeon ? :
- Signalement de typos, erreurs, etc :
- Commentaires libres :

Sources

- Site Officiel (http://singularity.lbl.gov/).
- TP de Martin Souchal (https://gitlab.in2p3.fr/souchal/tp-singularity).
- Singularity & Docker (http://singularity.lbl.gov/docs-docker).
- TP de Loic Gouarin (https://github.com/gouarin/container_precis2017)
- TP Ecole IN2P3 Conteneurs en production (https://gitlab.in2p3.fr/alexandre.dehne-garcia/TP_singularity_EcoleConteneursProd)
