

Ecole informatique IN2P3 2014 : Maîtriser le Cloud
TP Dev : portage d'applications sur le Cloud
**TP2 : Utilisation d'une image disque
customisée**

Cécile Cavet
`cecile.cavet@apc.univ-paris7.fr`
Centre François Arago (FACe),
Laboratoire AstroParticule et Cosmologie (APC), LabEx UnivEarthS
*APC, Univ. Paris Diderot, CNRS/IN2P3,
CEA/lrfu, Obs. de Paris, Sorbonne Paris Cité, France*

3 Juillet 2014

Table des matières

0.1	Introduction	1
0.2	Customisation d'une image disque	2
0.2.1	Script de customisation	2
0.2.2	Customisation de l'image	3
0.3	Utilisation de l'image disque customisée	5
0.3.1	Application : code Python pour l'Astrophysique	5

0.1 Introduction

Le but de ce TP est d'automatiser le portage d'une application scientifique en customisant une image disque. Pour cela, vous allez utiliser une image disque qui a été créée selon la méthode utilisée précédemment (TP1). Cette image disque a déjà subi une première étape de customisation car elle possède plusieurs modules Python. Pour exécuter l'application astrophysique que vous allez porter sur une MV, il est nécessaire de réaliser une deuxième phase de customisation et c'est à ce niveau que vous intervenez. Comme vous le verrez, la customisation peut prendre plus ou moins de temps en fonction de l'application à porter mais l'automatisation de ce processus est au final un gain de temps important lors de l'utilisation quotidienne de l'image disque correspondante.

Note : vous n'allez pas utiliser l'image disque de base que vous avez créé au TP1 ou éventuellement celles équivalentes qui sont créées automatiquement par les administrateurs de StratusLab (référéncées sur le MarketPlace par le créateur (« endorser ») `images@stratuslab.eu`) et par les administrateurs d'OpenStack (`Glance_ID = official-SL-6x-x86_64`) car l'installation des différents modules Python prend un temps non négligeable.

0.2 Customisation d'une image disque

Afin de customiser l'image disque, un script (lancé dès la première initialisation de la MV) va être nécessaire. L'utilisateur générique `cloud-user` est défini par défaut sur toute MV et vous pourrez donc l'utiliser pour l'exécution de l'application. De plus, la partie publique de votre clé de connection SSH est aussi copiée dans le `~/.ssh/authorized_keys` de son compte et vous pouvez donc vous connecter directement en tant que `cloud-user` (sans passer par `root`) sur la MV. À la fin de l'étape de customisation, les images disques qui en résultent doivent être référencées sur les catalogues d'image comme dans la dernière partie du TP1.

Attention : spécificité de StratusLab.

Sur StratusLab, la clé publique n'est pas copiée automatiquement sur le compte `cloud-user`. Pour effectuer cette action de manière automatique, vous devez préparer un fichier `cloud-init.txt` grâce à la commande suivante :

```
$ stratus-prepare-context ssh,$HOME/.ssh/id_rsa.pub
```

Vous devez ensuite instancier les MV avec la commande `stratus-run-instance` et l'option `--context-file`. Cette remarque n'est valable que pour les exécutions d'applications. Vous conserverez le fichier `cloud-init.txt` pour les autres TP.

0.2.1 Script de customisation

Complétez les espaces [...] dans le script suivant `custom_image.sh` afin de porter le code de calcul `photoz` disposé sur un serveur Web sur une MV du Cloud. Pour cela, vous devez configurer l'environnement pour que le code puisse fonctionner (`bashrc` et permissions de `cloud-user`). Vous pouvez vous aider à déboguer le script en instanciant la MV selon la description de la Section 0.2.2.

Script `custom_image.sh` :

```
#!/bin/bash

## remove network remaining
rm -f /lib/udev/rules.d/*net-gen*
rm -f /etc/udev/rules.d/*net.rules

## import the scientific application and configure
git clone git://gitorious.org/photoz/photoz.git /home/cloud-user/photoz

cat >> [...] <<EOF
export PYTHONPATH=/home/cloud-user/photoz
EOF

## install application
mkdir /home/cloud-user/Test_dir
[...]

echo "::::::::::::::::::::::::::::::::::::::::"
echo "Virtual Machine is ready"
echo "::::::::::::::::::::::::::::::::::::::::"

exit 0
```

0.2.2 Customisation de l'image

Vous allez utiliser une image SL6.5 customisée avec des modules Python pour laquelle vous appliquerez le script de customisation de la section précédente. Pour se connecter aux MV, vous

trouverez des indications dans les Sections **0.4.X/Test** de StratusLab et OpenStack du TP1. Il est possible de vérifier la bonne exécution du script au démarrage de la MV avec le fichier `/var/log/boot.log`.

Sur StratusLab

Utilisez l'image `Marketplace_ID = LUatm7ZCp4dof05XauGpLWRf4Xr` (SL6.5 + modules Python), le script `custom_image.sh` et la commande `stratus-create-image` avec entre autres l'option `--scripts` afin de customiser l'image disque.

Afin de débiter les problèmes éventuels, vous pouvez utiliser l'option `--no-shutdown` qui permet de vous connecter à la MV avant sa sauvegarde. Dans ce cas, vous devrez utiliser la commande `stratus-shutdown-instance` afin de finaliser l'image disque.

Après l'étape de customisation, la MV passera par les états shutdown et epilog avant que l'image disque soit accessible. L'image customisée restera 48h sur le MarketPlace sous le nom de créateur `jane.test@example.org` si elle n'est pas signée avec un certificat de grille. Vous recevrez un email avec les méta-données à signer et avec les commandes `stratus-sign-metadata` et `stratus-upload-metadata` vous pourrez réaliser cette action si vous possédez un certificat. Il est toutefois possible d'utiliser l'image customisée sans signature en récupérant son identifiant `Marketplace_ID` soit lors de l'exécution de la dernière commande, soit par email.

Sur OpenStack

Utilisez l'image `Glance_ID = SL6.5_cloud_school_TP2` (SL6.5 + modules Python), le script `custom_image.sh` et la commande `nova boot` avec entre autres l'option `--user-data` afin de customiser l'image disque. Sur OpenStack, étant donné que la MV n'a pas d'adresse IP publique lors de son initialisation, il est nécessaire d'exécuter la partie du script qui télécharge l'application (commande `git clone ...`) lors de la première connexion à la MV en tant que `cloud-user`.

Vous devez ensuite prendre un instantané de l'image disque présente sur la MV avec la commande `nova image-create` afin de sauvegarder la customisation que vous venez d'effectuer¹.

1. Cette commande peut générer l'erreur `ERROR : <attribute 'message' of 'exceptions.BaseException' objects> (HTTP 500)` en fonctionnant quand même.

0.3 Utilisation de l'image disque customisée

0.3.1 Application : code Python pour l'Astrophysique

Le code Pyraeus développé en Python permet de calculer les décalages spectraux photométriques de galaxies à partir d'un catalogue observationnel comme visible sur la Figure 1. Les décalages spectraux de galaxies vont ensuite servir à mesurer l'accélération de l'expansion de l'Univers qui est le grand défi de la cosmologie moderne.

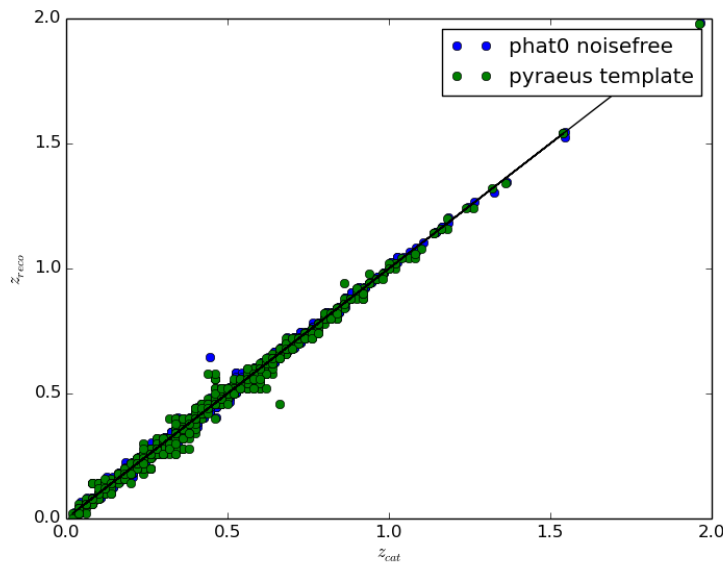


FIGURE 1 – Pyraeus : reconstruction des décalages spectraux photométriques de galaxies.

Lancez une MV avec comme ressources 1 CPU, 2-3 GB de RAM et l'image disque précédemment customisée. Connectez-vous à la MV en tant qu'utilisateur et avec le portage de X11. Exécutez le code Pyraeus à l'aide de la commande Python :

```
$ python photoz/test/phat0/phat_mag.py --testz -testdir ~/Test_dir
```

Vous devez voir apparaître la Figure 1. Le portage de l'application est à présent automatique. L'application est donc accessible instantanément sur les MV et elle peut être utilisée par ce biais par un grand nombre d'utilisateurs.