

## Session Pratique IBI-3

### Intégration d'outils bioinformatiques dans une appliance

contact : [support@france-bioinformatique.fr](mailto:support@france-bioinformatique.fr)

#### I) Description des appliances du cloud IFB à l'usage des développeurs d'appliances.

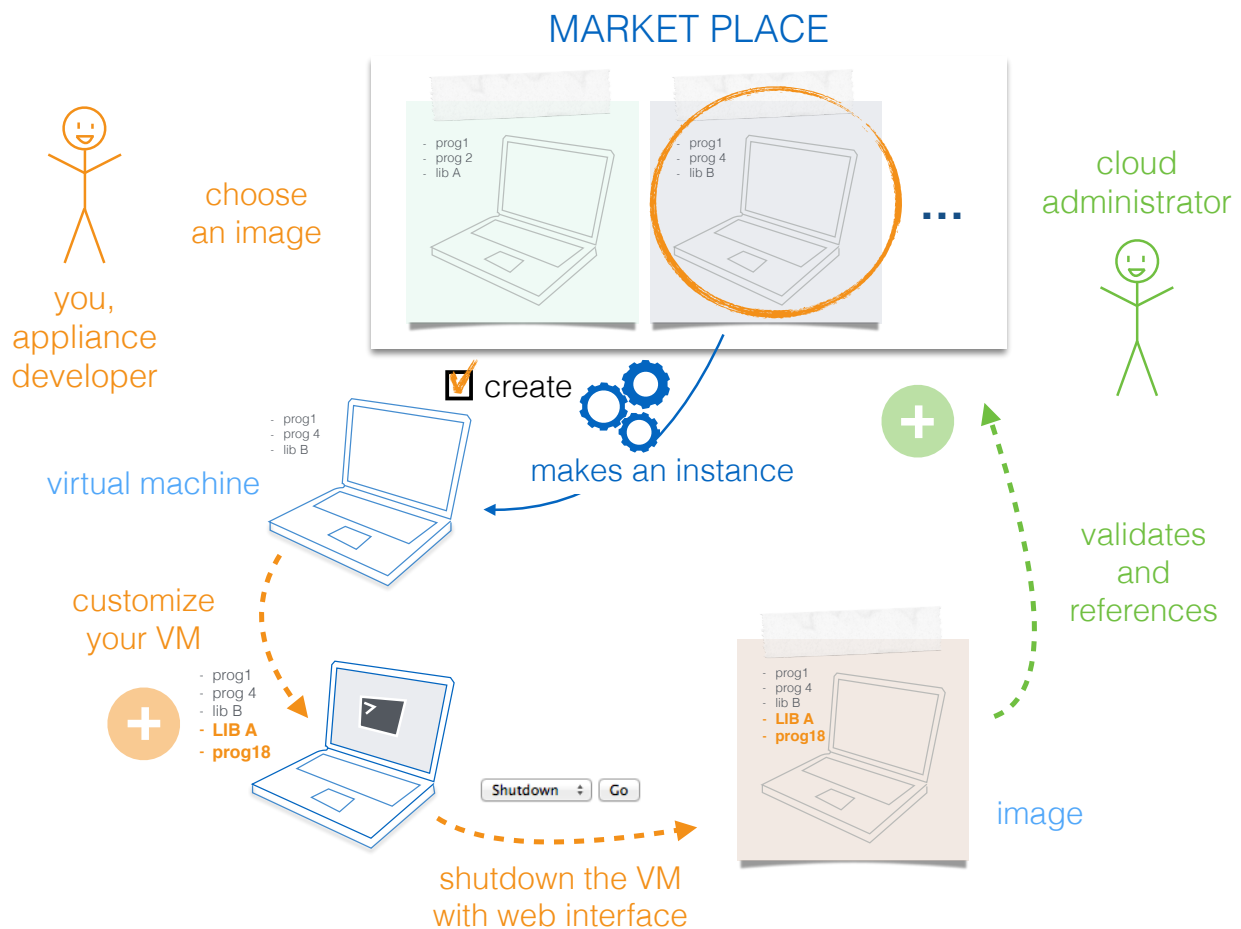
- Les **appliances** sont des modèles pré-définis de machines virtuelles. Elles contiennent un certain nombre d'outils pré-installés.
- La majorité des appliances du cloud IFB ont un **système d'exploitation CentOS**, mais il est également possible d'accéder à des appliances **Ubuntu**. Il n'y a pas d'autres systèmes d'exploitation actuellement disponibles sur le cloud.
- Par défaut, les utilisateurs peuvent se connecter aux instances d'appliances en tant que **super-utilisateur**. Lorsque vous créez une appliance, pensez à la possibilité de créer de nouveaux utilisateurs selon le public qui l'utilisera.
- Les programmes et bibliothèques bioinformatiques spécifiques aux appliances sont respectivement installés dans les répertoires **/ifb/bin**, **/ifb/lib** et **/ifb/include**. Utilisez ces répertoires pour installer vos propres programmes et dépendances. Vous pouvez également utiliser le répertoire **/ifb/src** si besoin.
- Plusieurs **banques de références** sont disponibles. Elles sont stockées sur un disque monté automatiquement dans le répertoire **/ifb/databases**. Pensez à la possibilité d'utiliser ces banques lors de la conception de votre appliance.
- Les utilisateurs peuvent monter un **disque virtuel** sur les instances de leurs appliances. Ce disque peut permettre d'augmenter la capacité de stockage d'une machine ou pour transférer des données entre plusieurs instances (comme un historique galaxy par exemple). Par défaut le montage de ce disque se fait dans le répertoire **/root/mydisk** mais il est possible de **redéfinir ce point de montage** en fonction des besoins de votre appliance.
- Si vous souhaitez déployer une application web, notez que tous les **ports** ne sont pas disponibles sur les machines du cloud, seulement les ports standard : 22, 80/443...
- Le cloud IFB a son propre système d'**installation logicielle automatisée** : **approver**. **Approver** est un script localisé dans le répertoire **/ifb/bin**. Ce script permet de lancer plusieurs scripts bash d'installation (de programmes) disponibles en ligne <http://cloud.france-bioinformatique.fr/approver/sh/> et de nettoyer ensuite les résidus de ces installations (zip, binaires...)
- **Docker** peut-être également employé pour déployer des logiciels sur le cloud.

## II) Le développement d'appliances : théorie, conseils et bonnes pratiques

Les principales étapes du développement d'appliance sont les suivantes :

- Choisir une image de base (parmi les appliances déjà disponible sur le cloud)
- En faire une instance en cochant le mode create.
- Personnaliser votre appliance en y ajoutant les dépendances et les programmes souhaités.
- Eteindre votre appliance en utilisant l'interface web.

Ci-dessous, une illustration du cycle de création d'une appliance. **Notez qu'il est nécessaire qu'un administrateur cloud valide votre appliance pour l'enregistrer et la rendre visible dans l'interface web du cloud.**



Voici l'interface qui vous permettra de lancer une instance de votre image de base en mode create :

Une fois que votre instance est lancée, elle apparait en orange dans le tableau de bord du cloud :

Virtual machine

☐
2528
schema\_test
R statistical computing

0%

1

2

0

ssh

☐
2531
my\_app
CentOS 6.6 (base)

0%

1

0

0

ssh

Virtual machine in create mode

Pour faciliter le développement de votre appliance, merci de suivre quelques conseils :

- **Faire la liste de l'ensemble des outils et leurs versions** que vous souhaitez installer sur votre appliance devra impérativement être transmise à l'équipe d'administration du cloud : support@france-bioinformatique.fr
- Préparer le processus d'installation de vos outils sur une instance normale (c'est à dire pas en mode création)
- Une instance en mode création accumule des logs. Pour limiter cette accumulation, éviter de conserver trop longtemps vos instance en mode création.
- Utilisez l'interface web pour éteindre (shutdown) vos instances (pas le shell).
- Arrêtez (kill) les instances que vous ne souhaitez pas enregistrer.
- Faire une mise à jour du système avant d'installer vos outils ('yum update' et 'reboot' s'il a eut mise à jour du kernel)
- Ne changez pas les droits de fichiers par 777...

- N'ouvrez que les ports nécessaires.
- Utilisez les port standard (22, 80/443...)
- Installer les logiciels dans les répertoires standard /ifb ou /usr/local...
- Fournir une description de votre appliance (il s'agit d'un manifeste de quelque lignes) et de vos outils originaux (voir format utilisé par le registre Elixir : <https://elixir-registry.cbs.dtu.dk>)

### III) Pratique de l'installation logicielle

Il existe diverses méthodes pour installer des logiciels sur les appliances du cloud. Avant d'utiliser votre propre recette :

- utilisez de préférence des paquets officiels yum, apt...
- sinon, consultez la liste des programmes disponibles via approver.

Lors de cette session pratique, nous allons installer un logiciel de trois façons différentes:

- en ligne de commande
- en créant un script approver
- en créant un Dockerfile pour générer un container docker

Ici, on souhaite installer la version 2.0.14 de TopHat.

#### 1) Installation manuelle

Voici, à quoi peut ressembler une **installation manuelle** :

```
wget ccb.jhu.edu/software/tophat/downloads/
tophat-2.0.14.Linux_x86_64.tar.gz
# or upload of the archive from your local computer if needed with a scp
command

tar -xzf tophat-2.0.14.Linux_x86_64.tar.gz
cd ophat-2.0.14.Linux_x86_64

# you can remove files that you don't need for instance
rm -f AUTHORS COPYING README

# then put the bin in the IFB directory
mv * /ifb/bin/

# clean the remains
cd ..
rm -r -f tophat-2.0.14.Linux_x86_64*
```

#### 2) Script Approver

Afin d'automatiser cette installation sous la forme d'un **script approver** utilisez des variables pour gérer les noms de programmes et paquets, leur numéro de version et leur URL. De plus il n'est pas nécessaire de passer par la phase de nettoyage.

Exemple :

```
tool_id="tophat"
tool_bin="tophat"
tool_version="2.0.14"
tool_url="ccb.jhu.edu/software/tophat/downloads"
```

```

tool_pkg="${tool_id}-${tool_version}.Linux_x86_64.tar.gz"

# install the tool
wget "${tool_url}/${tool_pkg}"
tar -xzf ${tool_pkg}

rm -f -r "${tool_pkg%.tar.gz}/AUTHORS ${tool_pkg%.tar.gz}/COPYING $
{tool_pkg%.tar.gz}/README"
mv "${tool_pkg%.tar.gz}"/* ${tools_dir}/bin/

```

Pour ensuite lancer l'installation de TopHat 2.0.14, il est ensuite nécessaire d'enregistrer ce script sous le nom nom-version.sh (tophat-2.0.14.sh) puis de le mettre en ligne (l'équipe IFB peut s'en charger). Enfin, la commande à lancer depuis votre instance est :

```
approve -i /ifb -t tophat-2.0.14
```

### 3) Image Docker

Afin d'obtenir un conteneur Docker qui inclue TopHat 2.0.14, il est possible de s'inspirer d'un script approve pour générer un **Dockerfile**. Celui-ci peut avoir le contenu suivant :

```

# use centos as base image
FROM centos
MAINTAINER Your Name <your@adress.fr>

# set variables
ENV tool_id tophat
ENV tool_version 2.0.14
ENV tool_url ccb.jhu.edu/software/tophat/downloads
ENV tool_pkg ${tool_id}-${tool_version}.Linux_x86_64.tar.gz

RUN yum -y install wget
RUN yum -y install tar

# Download blast
RUN wget "${tool_url}/${tool_pkg}"

# extract files
RUN tar -xzf ${tool_pkg}
RUN rm -f -r "${tool_pkg%.tar.gz}/AUTHORS ${tool_pkg%.tar.gz}/COPYING $
{tool_pkg%.tar.gz}/README" && mv "${tool_pkg%.tar.gz}"/* ${tools_dir}/
bin/

# here you can define or not tools_dir in advance but if you do remember
to export the path

# ENTRYPOINT ["tophat2"]

```

Pour construire une image et lancer un conteneur à partir de ce Dockerfile, enregistrez ce fichier sous le nom Dockerfile, le placer dans un répertoire. Placez-vous dans ce répertoire et lancer la commande suivante pour construire votre image :

```
docker build -t youdev/tophat:v2014 .
```

Puis lancer le conteneur :

```
docker run -t -i youdev/tophat:v2014 /bin/bash
```

## IV) Contextualisation des applications

La contextualisation est l'ensemble des moyen mis en place pour configurer automatiquement certaines caractéristiques des machines virtuelles, telles que :

- l'utilisateur par défaut de la machine (root?)
- le point de montage de disques virtuels
- les montages automatiques
- portails web

Actuellement, la contextualisation sur le cloud IFB est effectuée au moyen de scripts et de variables environnementales mais bientôt, le système **cloudinit** sera également disponible.

Les scripts activant les différents types de contextualisation sont situés dans le dossier `/etc/stratuslab/conf.d`.

Ces scripts permettent de lancer les scripts de contextualisation à proprement parler situés dans le dossier `/usr/bin` avec le préfixe `stratus-`. Il contiennent des variables environnementale en entête permettant de moduler l'action du script.

NB : ces variables se retrouvent dans l'interface web des administrateurs du cloud.

Pour observer les différents types de contextualisation mis en place, vous pouvez regarder le contenu de ces dossiers et fichiers dans les appliances BioCompute et Galaxy.