Session Pratique IBI-2
Utilisation avancée du cloud IFB

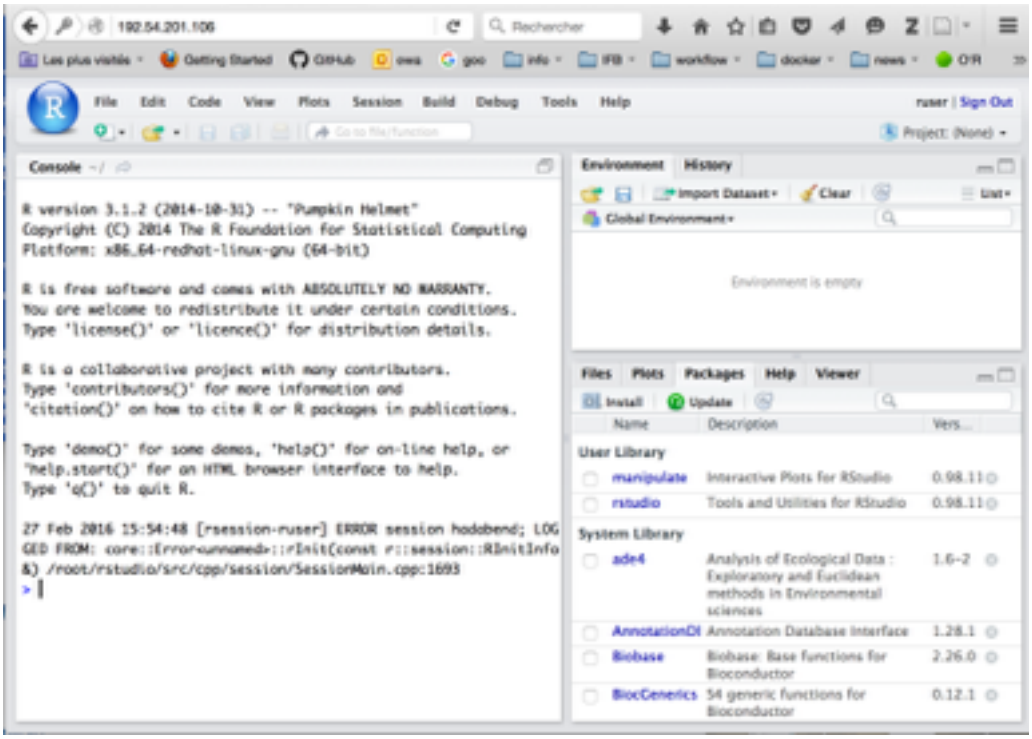
contact : support@france-bioinformatique.fr

SOMMAIRE

I) Installation manuelle

I.1) Installation d'un package dans RStudio

a) instancier une appliance **R statistical computing**;
b) accéder à RStudio avec le lien http;
c) se connecter comme utilisateur ruser / ruser;
d) dans la console:



e) installer le package dans le dossier utilisé par défaut : ~/R/x86_64-redhat-linux-gnu-library/3.1"

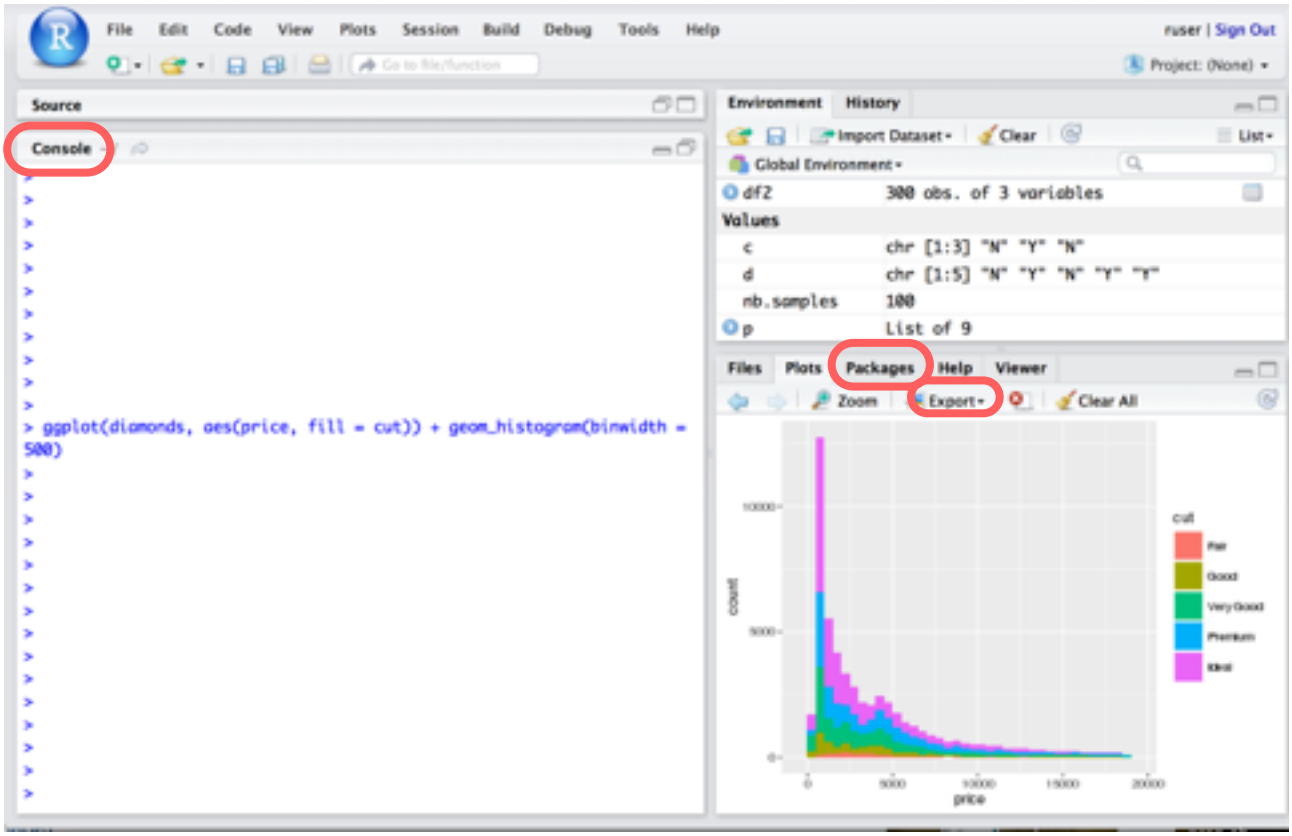
```
install.packages("ggplot2")
```

f) charger la librairie en ligne de commande ou dans la liste des packages disponibles, le chemin du package est optionnel s'il est enregistré dans le dossier utilisé par défaut par RStudio:

```
library("ggplot2", lib.loc="~/R/x86_64-redhat-linux-gnu-library/3.1")
```

g) créer un histogramme.

```
ggplot(diamonds, aes(price, fill = cut)) + geom_histogram(binwidth =
500)
```



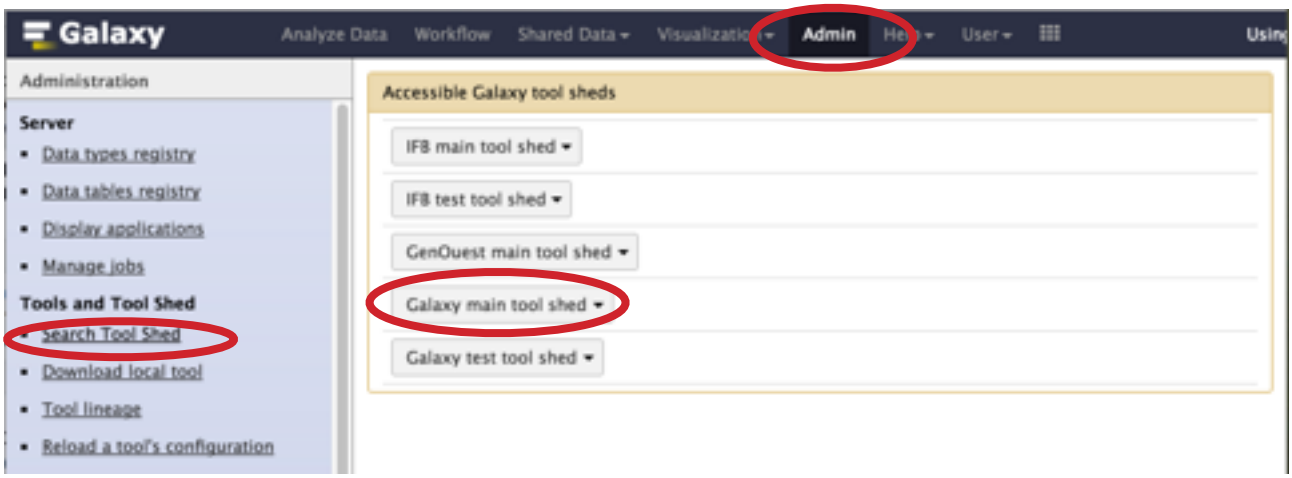
h) sauvegarder l'image dans le dossier /home/ruser/R.

```
ls /home/ruser/R/
ggplot2_histo.png                -> graphique enregistré
x86_64-redhat-linux-gnu-library  -> dossier d'installation du package
```

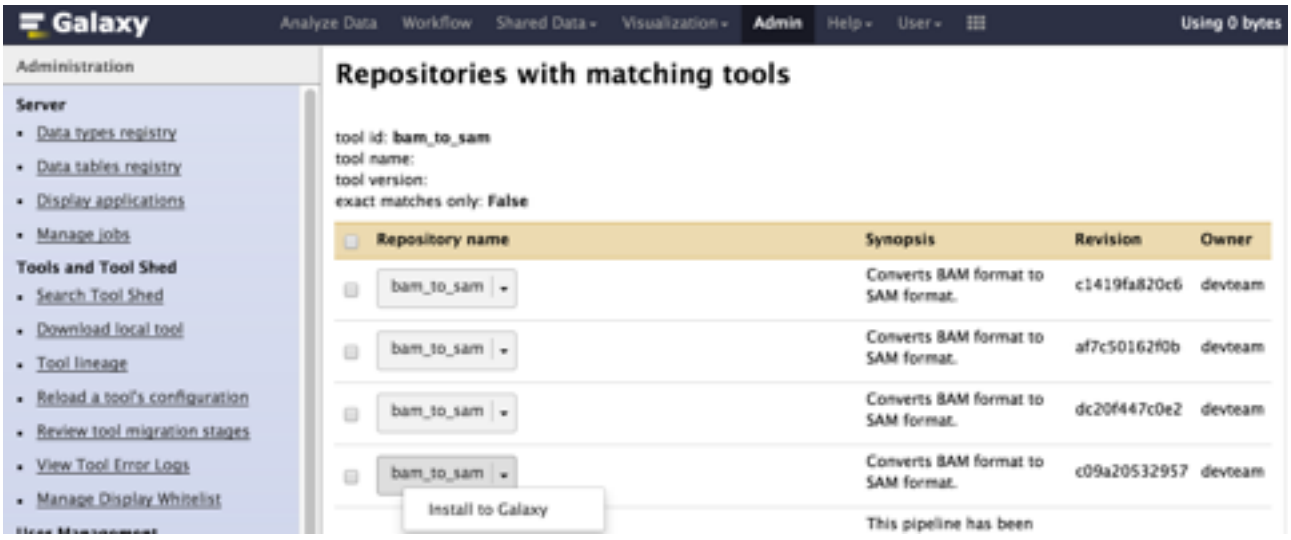
I.2) Installation d'un outil dans Galaxy

source: https://wiki.galaxyproject.org/Admin/Tools/AddToolFromToolShedTutorial

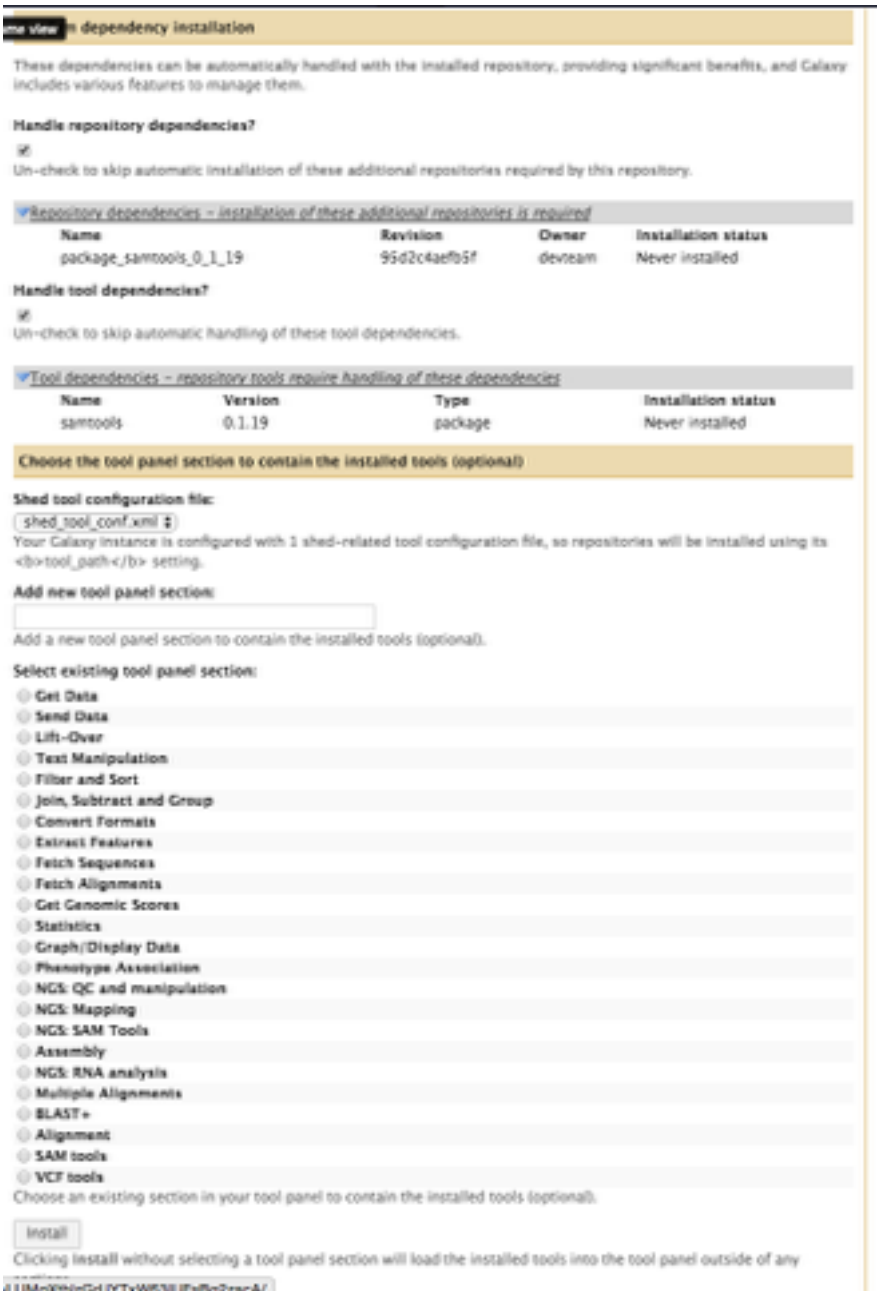
a. se connecter en tant qu'admin (admin@galaxy.ifb.fr / ifbadmin) ;
b. sélectionner le menu admin ;
c. dans le menu à gauche, sélectionner ***Search Tool sheds.***
Vous pouvez voir l'ensemble des toolshed mis à disposition dans l'instance Galaxy.



d. choisir le toolshed, dans le menu sélectionné ***Search for valid tools*** ;
e. renseigner le formulaire de recherche (ex: bam_to_sam) ;
f. sélectionner l'application recherchée dans la liste, dans le menu contextuel choisir ***Install*** ;



g. sélectionner le sous-menu où installer l'outil ou en créer un nouveau ;
h. lancer l'installer de l'application et des dépendances;



i. vérifier le succès de l'installation de l'outil en allant dans le menu ***Manage installed tools***

I.3) Installation d'une application graphique dans un bureau virtuel

a. créer une appliance avec un bureau virtuel :
   a. une machine disposant d'un bureau virtuel pré-configuré, ex ImageJ;
b. ouvrir X2Go sur l'hôte, créer une session pour accéder à la VM (name ego, bureau: gnome);
c. récupérer l'archive au format zip pour une installation sous Linux;
d. dézipper le fichier fastqc_v0.11.4.zip et le supprimer;
e. déplacer le dossier FastQC dans /home/ego/bin;
f. rendre le script exécutable (chmod +x /home/ego/bin/FastQC/fastqc);
g. créer un lien sur le bureau (ln -s //home/ego/bin/FastQC/fastqc /home/ego/Desktop);
h. depuis le bureau, lancer l'application puis sélectionner un fichier FastQ.

Ou utiliser le script d'exemple d'installation fournir dans le dataset.

De préférence utiliser des scripts d'installation et de configuration, les machines virtuelles ont par principe une courte durée de vie, il est plus simple de relancer la création d'une instance d'une machine et d'appliquer ses modifications.

IMPORTANT:

Les bureaux virtuelles proposent deux comptes de connexion :

Appliance	compte	bureau	commentaire
ImageJ	ego root	gnome gnome	/home/ego identifier comme utilisateur ego
BioDataCloud IGV	root	kde	/home/ego identifier comme utilisateur root, accès à tous les dossiers
Cytoscape	root	kde	/home/ego identifier comme utilisateur root, accès à tous les dossiers

II) Installation d'un outil en ligne de commande

II.1) Utilisation de scripts 'approver' sur une CentOS

L'IFB a mis en place un système d'installation de logiciels basé sur des scripts shell : 'approver'.

a) Instancier une appliance **Biocompute** ;
b) Afficher la liste des applications disponibles dans approver ;

```
ls -l /ifb/bin/
```

c) rechercher le mapper bwa, installer la version la plus récente dans le dossier /ifb (option -i). Le nom 'bwa' seul renvoie vers la dernière version de l'application disponible, soit ici la version 0.7.12 :

```
approver -l | grep bwa
bwa-0.5.1
bwa-0.7.10
bwa-0.7.12
bwa

approver -i /ifb -t bwa

bwa
Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.12-r1039
```

d) copier le fichier yeast.fasta dans l'appliance depuis votre ordinateur :

```
scp -P 22 yeast.fasta root@192.54.201.xxx:/root
```

e) lancer la création d'un index

```
bwa index -p yeastbwaidx -a bwtsw yeast.fasta
[bwa_index] Pack FASTA... 0.00 sec
[bwa_index] Construct BWT for the packed sequence...
[BWTIncCreate] textLength=1626368, availableWord=2066506
[bwt_gen] Finished constructing BWT in 5 iterations.
[bwa_index] 0.21 seconds elapse.
[bwa_index] Update BWT... 0.00 sec
[bwa_index] Pack forward-only FASTA... 0.01 sec
[bwa_index] Construct SA from BWT and Occ... 0.06 sec
[main] Version: 0.7.12-r1039
[main] CMD: bwa index -p yeastbwaidx -a bwtsw yeast.fasta
[main] Real time: 0.400 sec; CPU: 0.291 sec

ls -l
-rw-r--r--  1 root root      11 27 févr. 19:59 yeastbwaidx.amb
-rw-r--r--  1 root root      80 27 févr. 19:59 yeastbwaidx.ann
-rw-r--r--  1 root root  813256 27 févr. 19:59 yeastbwaidx.bwt
-rw-r--r--  1 root root  203298 27 févr. 19:59 yeastbwaidx.pac
-rw-r--r--  1 root root  406648 27 févr. 19:59 yeastbwaidx.sa
-rw-r--r--  1 root root  826794 27 févr. 19:57 yeast.fasta
```

II.2) Installation avec Conda/Bioconda

Source : https://github.com/arose/nglview

Sujet : utilisation d'une instance de Jupyter (http://jupyter.org/), un notebook qui permet d'associer du texte, du code, de la visualisation et installation de la bibliothèque nglview, un widget IPython de vue interactive de structure et de trajectoires moléculaires.

Instancier une **appliance Docker** (mais le TP n'a pas de pré-requis, mais les commandes sont présentées pour un OS Ubuntu).

a) faire la mise de l'appliance

```
apt-get update
```

b) installer anaconda2, par défaut dans $HOME/anaconda2, c'est un script interactif :

```
cd /tmp
wget http://repo.continuum.io/archive/Anaconda2-4.1.0-Linux-x86_64.sh
chmod 755 Anaconda2-4.1.0-Linux-x86_64.sh
./Anaconda2-4.1.0-Linux-x86_64.sh
```

Mise à jour du PATH, le script inclut le chemin vers conda, si on répond 'yes' à la question

```
source /root/.bashrc
```

Sinon mettre à jour manuelle le fichier bashrc, en ajoutant une ligne :
export PATH="<path>/anaconda2/bin:$PATH"
pour avoir accès directement au commande conda.

```
conda info
```

c) installation de nglview

```
conda install nglview -c bioconda
```

d) Par défaut l'application est accessible à localhost:8888. Or le port 8888 n'est pas accessible sur le cloud IFB, seul le port 80/443 est ouvert sur le cloud de l'IFB :

La solution simple, modifier le fichier de configuration pour obtenir l'URL : IP_machine:80
- génération du fichier de configuration (attention au copier/coller : 2 tirets devant l'option):

```
jupyter notebook --generate-config
Writing default config to: /root/.jupyter/jupyter_notebook_config.py
```

- modification de deux valeurs dans le fichier créé. Décommenter la ligne et changer la valeur

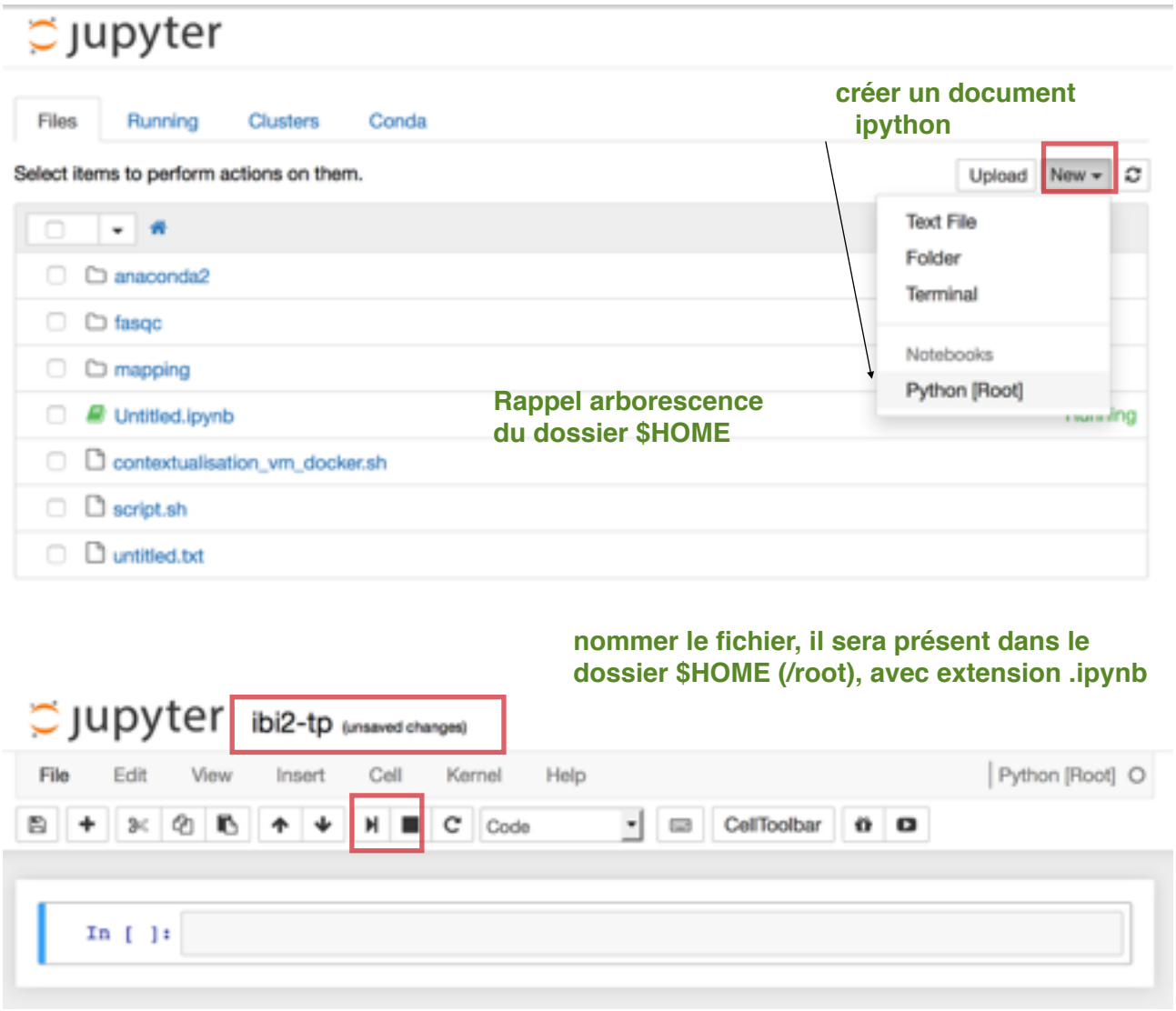
```
c.NotebookApp.port = 80
c.NotebookApp.ip = '<IP_machine>' # retaper les simples quotes
```

e) lancement de jupyter

```
jupyter notebook
```

Pour quitter la fenêtre ouverte, taper sur 'q' et confirmer. On accéder aux sorties du fichier log de logiciel, **en quittant le fichier, vous arrêtez Jupyter.**

```
Pour accéder à l'interface WEB de jupyter, dans un navigateur saisir
l'IP de la machine :
192.54.201.X
```



NB: **une autre solution possible** pour se connecter au jupyter notebook est d'utiliser un proxy pour rediriger les sorties de jupyter sur le port 8888 vers le port 80.
Une solution est présentée avec le serveur NGINX :

```
apt-get install nginx
```

- remplacer le fichier de **/etc/nginx/sites-enabled/default**, par celui fourni dans le dataset, bien **mettre la valeur de l'IP** de la machine virtuelle courante. Redémarrer le service :

```
service nginx restart
service nginx status
```

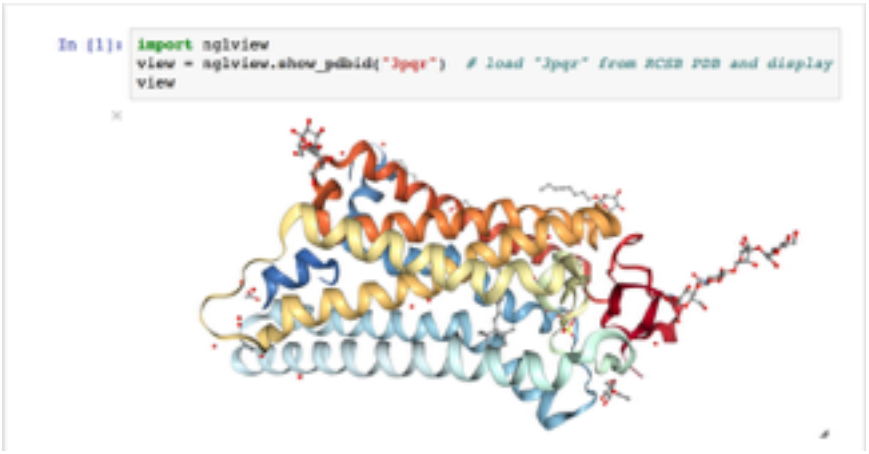
f) exercice1

Copier / coller le code dans une cellule de la nouvelle page de iPython et lancer l'exécution du code avec le run 'RUN'

```
import nglview
view = nglview.show_pdbid("3pqr")

view
```

- copier le code dans la cellule
- lancer l'exécution
In [*]
- attendre la fin de l'exécution (l'étoile est remplacée par le numéro du code, le graphique s'affiche
In [1]



Arrêt de Jupyter
dans la console, faire Control + C et confirmer

II.3) Installation d'un outil depuis un dépôt docker

Le cloud IFB propose une VM Docker. L'IFB utilise le dépôt d'images docker **BioShaDock** (http://docker-ui.genouest.org/app/#/) mis en place par la plateforme GenOuest. Le nom des images faites par l'IFB est préfixées par ifb.

a) instancier une appliance docker

b) récupérer l'image docker de l'outil cutadapt version 1.9.1

```
docker pull docker-registry.genouest.org/ifb/cutadapt:1.9.1
1.9.1: Pulling from ifb/cutadapt
5e7f975cbeeb: Pull complete
a3ed95caeb02: Pull complete
c90c3e7bc69b: Pull complete
951e77862ac9: Pull complete
b7b8d77a1a33: Pull complete
Digest:
sha256:38786feb00f37eace6d02d8770a4dec76c2ddf4fa2902a105b157ac1bbd02675
Status: Downloaded newer image for docker-registry.genouest.org/ifb/
cutadapt:1.9.1
```

c) vérifier la présence de l'image

```
docker images
REPOSITORY                                      TAG                 IMAGE ID
CREATED             SIZE
docker-registry.genouest.org/ifb/cutadapt   1.9.1
ee8d4a3067fa        4 weeks ago         320.6 MB
```

d) construire un dossier de test et copier un fichier fastq d'input.

```
mkdir test_docker && cd $_
scp -P 22 input.fastq root@192.54.201.xxx:/root
```

e) lancer cutadapt pour affichier l'help:

```
docker run docker-registry.genouest.org/ifb/cutadapt:1.9.1 cutadapt
cutadapt version 1.9.1
Copyright (C) 2010-2015 Marcel Martin <marcel.martin@scilifelab.se>
cutadapt removes adapter sequences from high-throughput sequencing
reads.
[...]
```

f) lancer une commande en mode non interactif

```
docker run -v $(pwd):/tmp docker-registry.genouest.org/ifb/cutadapt:
1.9.1 cutadapt -a AACCGGTT -o /tmp/output.fastq /tmp/input.fastq
This is cutadapt 1.9.1 with Python 2.7.3
Command line parameters: -a AACCGGTT -o /tmp/output.fastq /tmp/
input.fastq
Trimming 1 adapter with at most 10.0% errors in single-end mode ...
Finished in 0.01 s (303 us/read; 0.20 M reads/minute).
[…]

ls -l
-rw-r--r-- 1 root root 5345 Feb 27 18:17 input.fastq
-rw-r--r-- 1 root root 5346 Feb 27 18:18 output.fastq
```

g) lancer une commande en mode interactif

```
root@vm0151:~/test_docker# docker run -it --rm -v $(pwd):/tmp docker-
registry.genouest.org/ifb/cutadapt:1.9.1 bash
root@9be0d6bf7b48:/# cd /tmp
root@9be0d6bf7b48:/tmp# ls -l
-rw-r--r-- 1 root root 5345 Feb 27 18:17 input.fastq

root@9be0d6bf7b48:/tmp# cutadapt -a AACCGGTT -o output2.fastq
input.fastq
This is cutadapt 1.9.1 with Python 2.7.3
Command line parameters: -a AACCGGTT -o output2.fastq input.fastq
Trimming 1 adapter with at most 10.0% errors in single-end mode ...
Finished in 0.01 s (303 us/read; 0.20 M reads/minute).
[…]

root@9be0d6bf7b48:/tmp# ls -l total 28
-rw-r--r-- 1 root root 5345 Feb 27 18:17 input.fastq
-rw-r--r-- 1 root root 5346 Feb 27 18:18 output.fastq
-rw-r--r-- 1 root root 5346 Feb 27 18:20 output2.fastq
```

NB : quelques mots sur le montage NFS

Le montage NFS permet de partager un volume entre plusieurs machines virtuelles. Pour réaliser un montage permanent, veuillez suivre les indications suivantes :
1) Lancer une instance de l'appliance NFS server (elle servira de répertoire).
- Désactiver les iptables des machines client (temporairement pour les besoins du TP).
- Choisir de préférence le point de montage par défaut sur les machines client (`/root/mydisk`).

Exemple

Reprendre l'exemple précédent mais les fichiers sont enregistrés sur un disque monté sur un serveur NFS, chaque machine cliente : l'instance Biocompute avec approver et l'instance docker vont monter le disque.

a. créer un disque de 5 Go, nom 'disk_nfs' ;
b. créer une instance de NFS server (2016-02) en sélectionnant ce disque;
c. créer un fichier sur le disque du serveur NFS ;

```
ssh -A -p 22 root@192.54.201.98 touch /root/mydisk/hello
```

d. sur chaque machine devant utiliser le disque :
   a. ajouter une ligne dans le fichier /etc/fstab, en mettant l'IP du serveur :

```
192.54.201.XXX:/root/mydisk  /root/sharedisk  nfs defaults 0 0
```

   b. créer le dossier servant le point de montage
   c. lancer le montage

```
mkdir /root/sharedisk
mount -a
mount
ls -l /root/sharedisk
```

A ce moment, il n'est pas possible d'écrire sur le disque depuis cette machine.
e. sur le **serveur NFS**, le disque est monté dans 'mydisk' :
   a. donner accès au disque aux deux machines qui vont faire les traitements, du fichier **/etc/exports,** effacer le contenu du fichier.

   b. définir une règle par machine dans le fichier /etc/exports, le dossier mydisk du serveur sera accessible uniquement aux machines clientes renseignées dans le fichier :

```
/root/mydisk  IP_client1(rw,no_root_squash)
/root/mydisk  IP_client2(rw,no_root_squash)
```

   c. relancer le serveur NFS

```
service nfs reload
exportfs -av
```

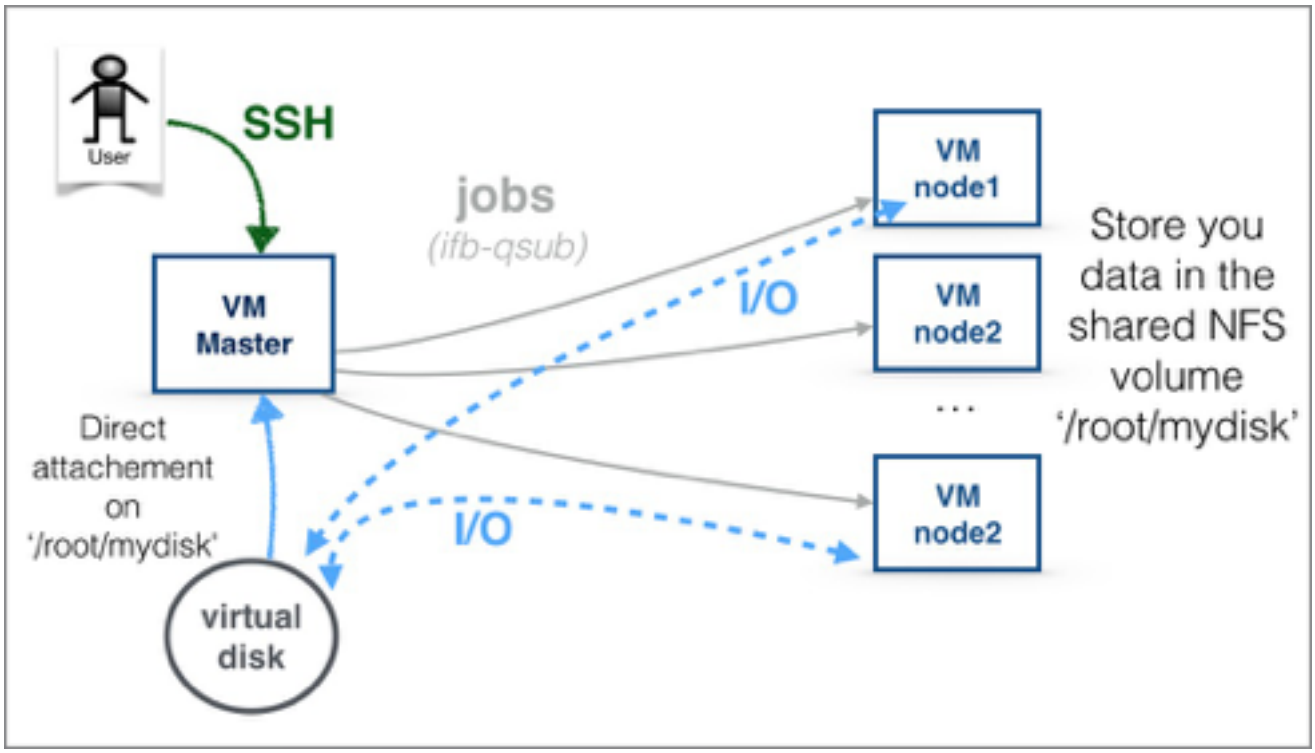
Vérifier que vous pouvez maintenant écrire sur le disque.

NB: après suppression d'une machine virtuelle, pensez à mettre à jour la liste des machines avec un droit accès au disque sur le serveur NFS.

III) Mise en place d'un cluster virtuel

III.1) Principe avec RabbitMQ

Le **principe** du cluster sur le cloud IFB est simple : vous vous connectez au noeud maître sur lequel est attaché un disque virtuel, lors de la configuration du cluster ce disque est monté sur les noeuds esclaves (ce qui permet d'avoir accès aux mêmes données), ensuite vous lancez les jobs qui seront répartis parmi les noeuds esclaves depuis le maître (voir schéma ci-dessous).



Voici les étapes pour mettre en place le mode cluster:

a) Activer un agent ssh dans un terminal sur votre ordinateur à l'aide des commandes :

```
ssh-agent
ssh-add ~/.ssh/id_rsa (or id_dsa)
```

b) Lancer une instance en y attachant un disque virtuel (il s'agira du noeud maître)
c) Lancer plusieurs instances qui serviront de noeuds esclaves.
d) Se connecter au noeud maître en ssh avec l'option -A, puis éditer le fichier `~/cluster/nodes.l` en y ajoutant les adresses IP des noeuds esclaves.
e) Lancer le script de configuration à l'aide de la commande :

```
ifb-cluster cluster/nodes.l
```

ou

```
ifb-cluster cluster/nodes.l x
```

(avec `x` correspondant au nombre de CPUs disponibles par noeud esclave)

f) Soumettre les jobs avec la commande :

```
ifb-qsub $job
```

g) Les jobs sont exécutés par le script ifb-worker sur chaque noeud

Exemple pratique : BLAST

- Mettre en place l'agent ssh sur votre ordinateur
  ssh-add ~/.ssh/id_dsa
- Créer 1 disque virtuel
- Créer 1 instance de Biocompute avec 2 CPU en attachant ce disque (maître)
- Créer 2 instances de Biocompute avec 1 CPU (esclaves)
- Mettre le fichier test sur le noeud maître :
  `scp fasta1 root@<ipmasternode>:/root/mmydisk/`
- Ajouter les adresses IP au fichier nodes.l
- Modifier le fichier `~/cluster/job-proc` en y ajoutant une ligne contenant `$1` (cela permettra aux esclaves de récupérer et d'exécuter les commandes lancées à partir du maître).
- Lancer la configuration :
  `ifb-cluster cluster/nodes.l`
- Lancer plusieurs jobs avec la commande :

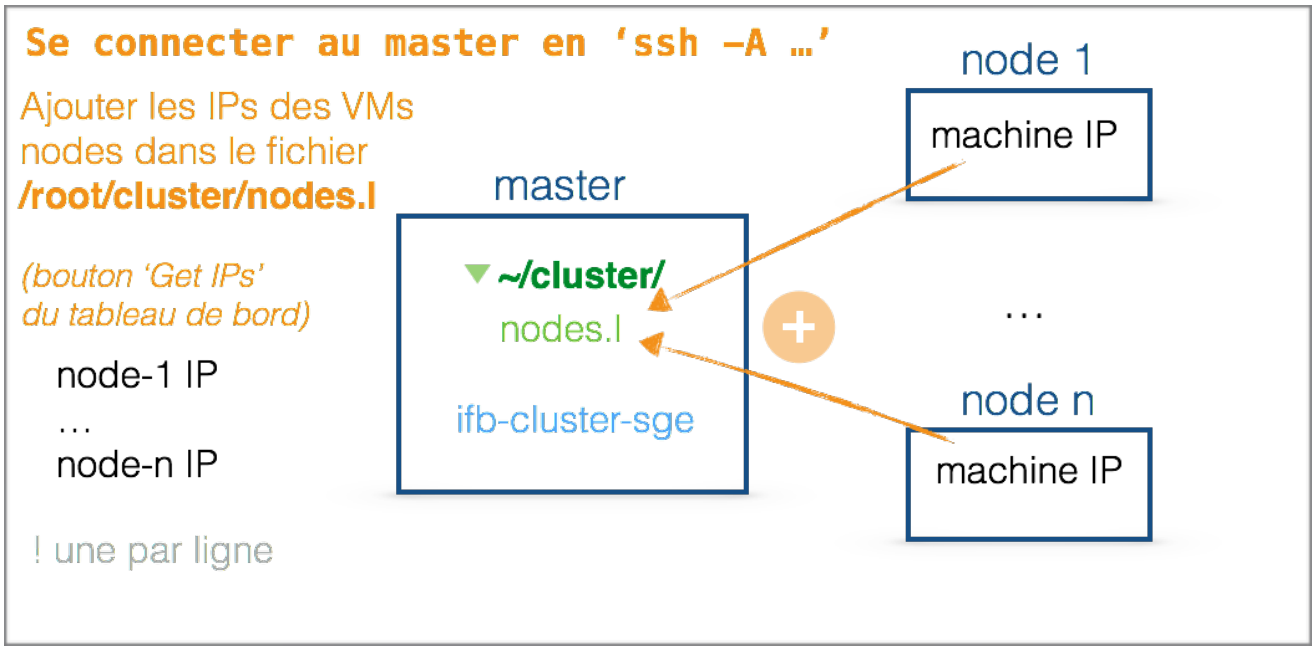
```
DB=/ifb/databases/est/est-2015-12-09/flat/est_mouse
for job in {1..2}; do ifb-qsub blastn -query ~/mydisk/blastn.fasta -db
$DB -out ~/mydisk/res${job} -outfmt 7 ; done
```

III.2) Utilisation du mode cluster avec SGE

Un mode cluster utilisant SGE est maintenant disponible pour les appliances CentOS 6.8 et 7.2. La configuration se fait via un script appelé `ifb-cluster-sge` et utilise le fichier `nodes.l` vu précédemment.

Pour **mettre en place le mode cluster utilisant SGE**, effectuer les étapes suivantes :
a) Activer un agent ssh dans un terminal sur votre ordinateur à l'aide des commandes :

```
ssh-agent
ssh-add ~/.ssh/id_rsa (or id_dsa)
```



b) Lancer une instance CentOS 6.7 en y attachant un disque virtuel (il s'agira du noeud maître)

c) Lancer plusieurs instances d'appliance qui serviront de noeuds esclaves.

d) Se connecter au noeud maître en ssh avec l'option -A, puis éditer le fichier `~/cluster/nodes.l` en y ajoutant les adresses IP des noeuds esclaves.

e) Lancer le script de configuration à l'aide de la commande :

```
ifb-cluster-sge cluster/nodes.l 2
```

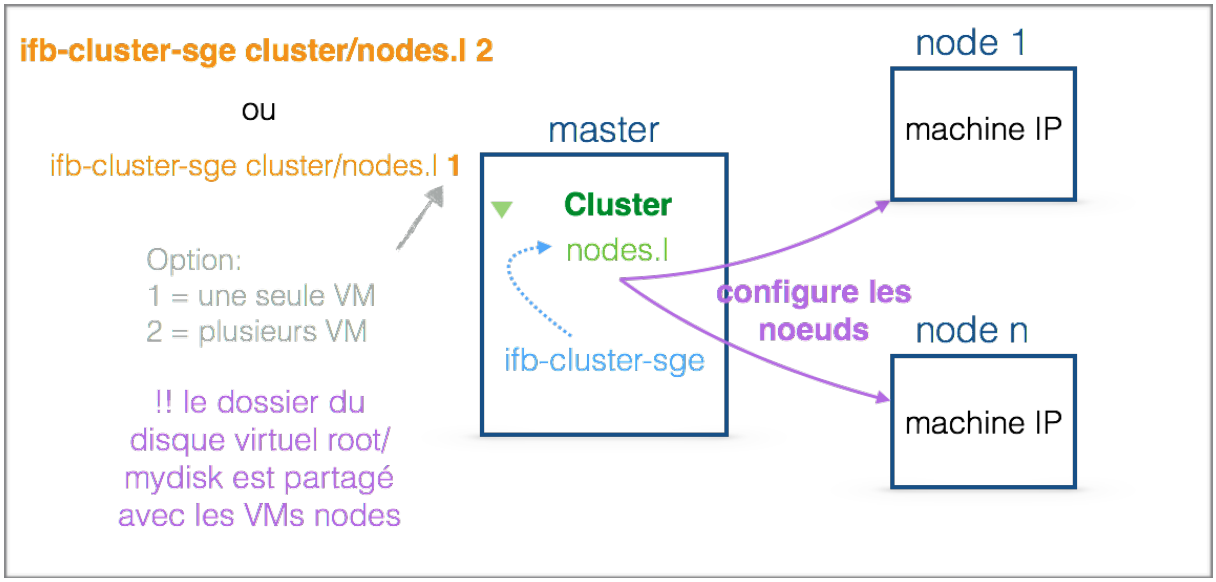
ou

```
ifb-cluster-sge cluster/nodes.l 1
```

L'option « 1 » crée un cluster sur une seule instance (noeud maitre et noeud esclave sur la même instance), l'option 2 avec des machines esclaves.

f) Soumettre les jobs grâce à la commande suivante, en tant que sge-admin:

```
qsub $job
```



Exemple pratique : BLAST

- Mettre en place l'agent ssh sur votre ordinateur
  `ssh-add ~/.ssh/id_dsa`
- Créer 1 disque virtuel
- Créer 1 instance de Biocompute avec 1 CPU en attachant ce disque (maître)
- Créer 2 instances de Biocompute avec 1 CPU (esclaves)
- Mettre les fichiers tests sur le noeud maître :

```
scp fasta1 root@<ipmasternode>:/root/mmydisk/
scp fasta2 root@<ipmasternode>:/root/mmydisk/
```

- Ajouter les adresses IP au fichier nodes.l

- Lancer la configuration :

```
ifb-cluster-sge cluster/nodes.l 2
```

- Lancer plusieurs jobs avec un « job array » .
- créer un fichier blastn.sh et copier le code suivant:

```
#!/bin/sh
#$ -t 1-2

DB=/ifb/databases/est/est-2015-12-09/flat/est_mouse

blastn -query /root/mydisk/fasta$SGE_TASK_ID -db $DB -out /root/
mydisk/res$SGE_TASK_ID -outfmt 7
```

- Lancer le « job array » :

```
qsub blastn.sh
```

- suivi de l'avancement

```
qstat -f ou qstat
```