

Installation et mise en production de Ray

Creation et propagation de clés SSH

Connexion serveur et aller dans son dossier personnel :

```
ssh dominguesw@lsis-dimag-cluster-01
cd ~
```

Une fois connectée création de la clé :
(Stockez la dans le répertoire par défaut de votre compte sans passphrase)

```
ssh-keygen -b 2048
```

Voici un retour :

```
dominguesw@lsis-dimag-cluster-01:~$ ssh-keygen -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dominguesw/.ssh/id_rsa):
Created directory '/home/dominguesw/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dominguesw/.ssh/id_rsa.
Your public key has been saved in /home/dominguesw/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:T01TlvT1CanIRLMAuMyxG8EUv28K0wnq1tVuMHLXvNw dominguesw@lsis-dimag-cluster-01
The key's randomart image is:
+---[RSA 2048]---+
|  ooo....o  o=.. |
|   =.   ..o  +o.+ |
|  o =.   o..  +.o |
|   *    .o  +   |
|   .o.. S   .   |
|   .oo=oo  =    |
|   ..o+o=o.  +   |
|   .. .o oo.o E  |
|   ..   ..      |
+---[SHA256]-----+
dominguesw@lsis-dimag-cluster-01:~$
```

Figure 1

Copiez ensuite cette clé dans vos autres compte du cluster (attention pour que cela soit symétrique il faut propager la clé publique et la clé privée.

```
dominguesw@lsis-dimag-cluster-01:~$ cd .ssh/  
dominguesw@lsis-dimag-cluster-01:~/.ssh$ ls  
id_rsa id_rsa.pub
```

Figure 2

Pour être sur de ne pas avoir de problème en amont vous devez créer un dossier caché .ssh sur chaque noeud :

```
ssh dominguesw@lsis-dimag-cluster-02 "mkdir ~/.ssh"
```

Répondez oui à la question s'il vous demande d'importer la signature du serveur (voir).

```
dominguesw@lsis-dimag-cluster-01:~$ ssh dominguesw@lsis-dimag-cluster-02 "mkdir ~/.ssh"  
The authenticity of host 'lsis-dimag-cluster-02 (194.167.251.147)' can't be established.  
ECDSA key fingerprint is SHA256:SPGZA96dBoyIEjql5NMa181wgECcfNz4q9+t/rRfr/0.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'lsis-dimag-cluster-02,194.167.251.147' (ECDSA) to the list of known  
hosts.  
dominguesw@lsis-dimag-cluster-02's password:
```

Figure 3

Copiez votre clé publique dans un fichier nommé authorized_keys et copiez tout (3 fichiers) d'un serveur à l'autre via la commande scp :

```
cp .ssh/id_rsa.pub .ssh/authorized_keys  
scp .ssh/id_rsa .ssh/id_rsa.pub .ssh/authorized_keys dominguesw@lsis-  
dimag-cluster-02:~/.ssh/
```

Voilà ce qui doit se passer si le dossier de destination existe :

```
dominguesw@lsis-dimag-cluster-01:~$ cp .ssh/id_rsa.pub .ssh/authorized_keys  
dominguesw@lsis-dimag-cluster-01:~$ scp .ssh/id_rsa .ssh/id_rsa.pub .ssh/authorized_keys doming  
uesw@lsis-dimag-cluster-02:~/.ssh/  
id_rsa  
id_rsa.pub  
authorized_keys  
dominguesw@lsis-dimag-cluster-01:~$
```

Figure 4

Si cela fonctionne maintenant vous pouvez faire un ssh vers un autre noeud du cluster sans saisir votre mot de passe.

```
ssh lsis-dimag-cluster-02
```

Une fois le test effectué pensez à revenir sur le noeud 1 pour continuer le déploiement de l'outil.

```
dominguesw@lsis-dimag-cluster-01:~$ ssh lsis-dimag-cluster-02
Linux lsis-dimag-cluster-02 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3 (2019-09-02) x86_64
Répondre → Transférer Archiver Indésiré
The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
Last login: Wed May 6 09:54:50 2020 from 194.167.251.146
dominguesw@lsis-dimag-cluster-02:~$
```

Installation de Ray

La version stable à l'heure de la rédaction de ce document est 0.8.4. Cette partie a été prise en charge par l'administrateur système du cluster pour déployer les paquets pour tous les utilisateurs.

La commande est simple pour python 3.x

```
pip3 install -U ray
```

Pour plus d'information si vous avez besoin de modifier quelques choses reportez-vous à l'URL suivante : <https://docs.ray.io/en/latest/installation.html>

Pour un fonctionnement mode cluster ray utilise Redis. Mais après des tests il semble qu'il ne faut pas que redis-server soit installé sur le noeud du cluster. Donc si besoin pour le supprimer de notre cluster sous Debian 9 toujours en tant que « root » (administrateur du système) pour l'installation et le lancement du service :

```
/etc/init.d/redis-server stop
apt remove redis-server
```

Test de fonctionnement manuel

Nous allons faire un premier test pour voir si deux noeuds peuvent communiquer. Lançons sur le premier ray en mode écoute (génération de la base Redis) et un second qui vient se connecter au premier.

Pour cela nous allons nous appuyer sur : <https://docs.ray.io/en/latest/using-ray-on-a-cluster.html>

Sur le premier nous lançons :

```
ssh lsis-dimag-cluster-01 -l dominguesw
cd ~
ray start --head --redis-port=6379
```

Voici le retour :

```

dominguesw@lsis-dimag-cluster-01:/tmp/ray$ ray start --head --redis-port=6379
2020-05-06 11:59:28,826 INFO scripts.py:357 -- Using IP address 194.167.251.146 for this node.
2020-05-06 11:59:28,829 INFO resource_spec.py:212 -- Starting Ray with 77.39 GiB memory available for workers and up to 37.17 GiB for objects. You
can adjust these settings with ray.init(memory=<bytes>, object_store_memory=<bytes>).
2020-05-06 11:59:29,272 INFO services.py:1148 -- View the Ray dashboard at localhost:8265
2020-05-06 11:59:29,296 INFO scripts.py:387 --
Started Ray on this node. You can add additional nodes to the cluster by calling

    ray start --address='194.167.251.146:6379' --redis-password='5241590000000000'

from the node you wish to add. You can connect a driver to the cluster from Python by running

    import ray
    ray.init(address='auto', redis_password='5241590000000000')

If you have trouble connecting from a different machine, check that your firewall is configured properly. If you wish to terminate the processes
that have been started, run

    ray stop
dominguesw@lsis-dimag-cluster-01:/tmp/ray$

```

et la connexion du second sur le premier :

```

ssh lsis-dimag-cluster-02 -l dominguesw
cd ~
ray start --address=194.167.251.146:6379

```

Ce qui doit donner :

```

dominguesw@lsis-dimag-cluster-02:~$ ray start --address=194.167.251.146:6379
2020-05-06 12:02:15,187 INFO scripts.py:429 -- Using IP address 194.167.251.147 for this node.
2020-05-06 12:02:15,195 INFO resource_spec.py:212 -- Starting Ray with 86.62 GiB memory available for workers and up to 37.13 GiB for objects. You
can adjust these settings with ray.init(memory=<bytes>, object_store_memory=<bytes>).
2020-05-06 12:02:15,224 INFO scripts.py:438 --
Started Ray on this node. If you wish to terminate the processes that have been started, run

    ray stop
dominguesw@lsis-dimag-cluster-02:~$

```

Figure 7

Pour tout arrêter sur chaque noeud :

```

ray stop

```

Test de ray en mode automatique

Nous allons créer un fichier de commande pour que ray puisse se lancer tout seul sur chaque noeud du cluster.

Attention il ne faut pas qu'il y ai de serveur redis lancé sur les noeuds impactés. Il faut avoir suivi la procédure donnée au debut du document pour le distribution de clé SSH.

En « root » vérifier que l'application pssh est bien installer sur tous les nœuds (au moins sur celui qui servira de maître (header dans le jargon de Ray)).

```

apt install pssh

```

Le fichier de configuration contient plusieurs informations afin de permettre le lancement automatique des noeuds du cluster et les adresses des noeuds.

Voici à quoi il ressemble il faut le placer dans votre repertoire maison :

```
cluster_name: RayTest

min_workers: 0
initial_workers: 7
max_workers: 7

autoscaling_mode: default
target_utilization_fraction: 0.8
idle_timeout_minutes: 5

docker:
  image: ""
  container_name: ""
  pull_before_run: True
  run_options: []

provider:
  type: local
  head_ip: 194.167.251.146
  worker_ips: [194.167.251.142, 194.167.251.143, 194.167.251.144,
194.167.251.145, 194.167.251.147, 194.167.251.148]

auth:
  ssh_user: dominguesw
  ssh_private_key: ~/.ssh/id_rsa

head_node: {}

worker_nodes: {}

file_mounts: {
#   "/path1/on/remote/machine": "/path1/on/local/machine",
#   "/path2/on/remote/machine": "/path2/on/local/machine",
}

initialization_commands: []

setup_commands:
- pip3 install -U ray

head_setup_commands: []

worker_setup_commands: []

head_start_ray_commands:
- ray stop
- ray start -head -redis-port=6379 --autoscaling-\\
config=/home/dominguesw/test.yaml --webui-host 0.0.0.0

worker_start_ray_commands:
- ray stop
- ray start --address=194.167.251.146:6379
```

Vous remarquerez que le chemin du fichier est mis en dur dans le block "head_start_py_commands".

De plus pour le démarrage des autres “workers” noeuds on précise en dur l’adresse du header. C’est la dernière ligne du fichier de configuration.

Pour Executer et lancer le cluster on lancer les commandes suivantes :

```
ssh lsis-dimag-cluster-01 -l dominguesw  
cd ~  
ray up ./test.yaml -y
```

Si tout c’est bien passé vous pouvez vous rendre via un client web à l’adresse <http://194.167.251.146:8265> pour admirer les noeuds se connecter les uns après les autres. Attention ce n’est pas instantané et chaque noeud prend un certain avant de ce connecter.

Voici à quoi cela ressemble :

Figure 8

Couplage avec Deepnlpf

Vous devez récupérer les deux fichiers qui vont permettre de faire les tests qui sont :

- custom_pipeline.yaml : pour la configuration du cluster
- custom_pipeline_yaml_unittest.py : Test unitaire pour valider le fonctionnement

Copiez les dans un espace commun comme votre espaces dans le NAS à la racine de chaque noeud du cluster et allez-y. Voici les commandes en partant du principe que les fichiers sont déjà dans votre répertoire “home” :

```
cd /NAS
mkdir ray_test
cd ray_test
cp ~/custom* .
ls
```

Et voici le retour écran :

```
dominguesw@lsis-dimag-cluster-01:/NAS$ cd /NAS
dominguesw@lsis-dimag-cluster-01:/NAS$ mkdir ray_test
dominguesw@lsis-dimag-cluster-01:/NAS$ cd ray_test/
dominguesw@lsis-dimag-cluster-01:/NAS/ray_test$ cp ~/custom* .
dominguesw@lsis-dimag-cluster-01:/NAS/ray_test$ ls
custom_pipeline.yaml  custom_pipeline_yaml_unittest.py
dominguesw@lsis-dimag-cluster-01:/NAS/ray_test$
```

Figure 9