

# Configuration de l'environnement Cassandra

v1.1

# Table of Contents

Intro.....	1
Architecture.....	1
Mise en place de l'architecture .....	1
Installation de la machine virtuelle .....	1
Connexion en ssh sur la VM .....	2
Gestion du cluster via CCM .....	3
<b>DevCenter</b> .....	5
Preparation .....	6
Installation <b>DevCenter</b> : .....	7

# Intro

Pour le TP Cassandra vous allez utiliser une machine virtuelle qui contient un cluster Cassandra de 3 noeuds hebergees sur la meme VM. Vous allez vous connecter a cette machine via **ssh**. Optionnellement vous pouvez aussi installer **DataStax DevCenter** qui offre un client CQL plus abouti.



Pour ce tp vous avez besoin d'une poste (idealement linux) avec 4GB de RAM et 4GB d'espace disque disponible. **VirtualBox** et un client **SSH** doivent etre deja installes.

## Architecture

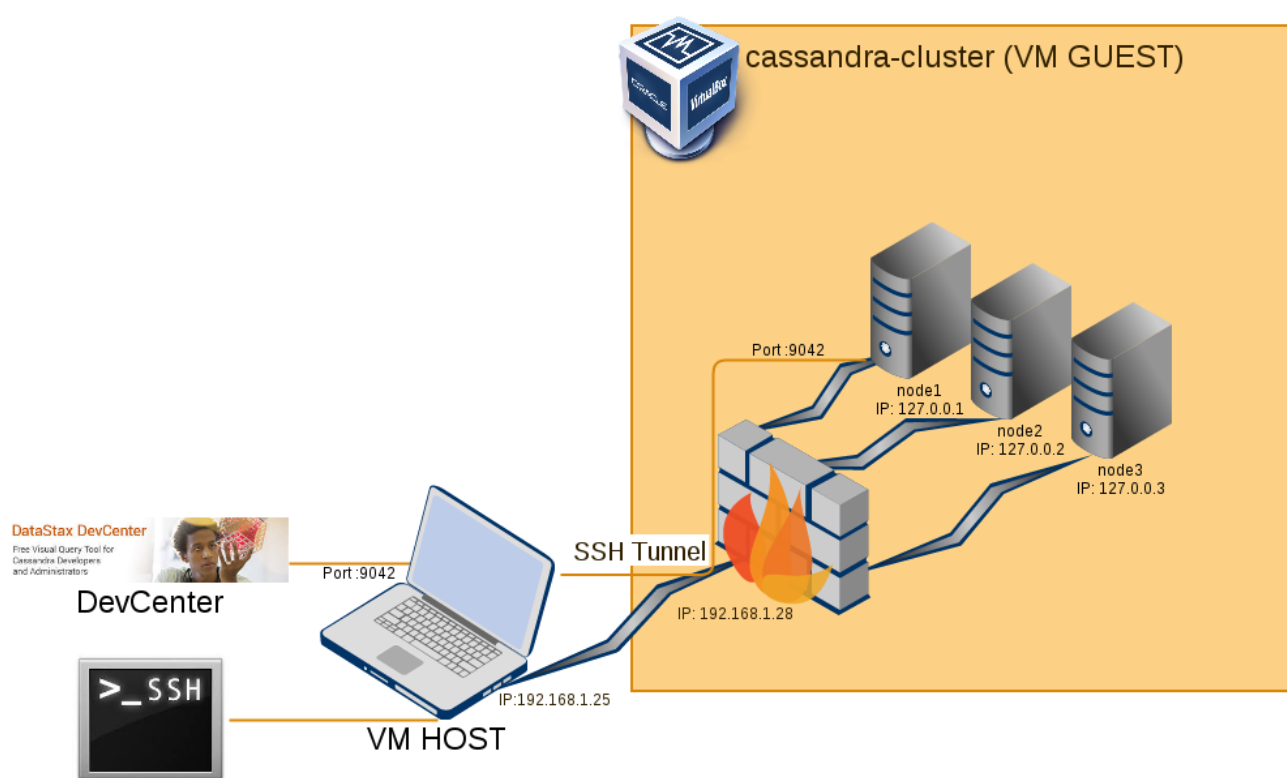


Figure 1. Architecture

## Mise en place de l'architecture

### Installation de la machine virtuelle

1. Trouver la VM dans l'archive (cassandra\_spark\_cluster.ova)
2. Importer cette VM dans votre VirtualBox
3. Lancer la VM
4. Noter l'adresse IP affichee lors du demarrage



Choisissez pour la VM le type de connexion Réseau privé hôte. Si aucune interface réseau n'est disponible, vous pouvez en créer une dans le menu Fichier > Paramètres > Réseau > Réseau hôte uniquement. Choisissez d'ajouter une nouvelle interface (premier pictogramme sur la droite de la fenêtre).

Dans nos exemples l'adresse IP de la VM est **192.168.1.28**. *N'oubliez pas de remplacer partout dans les exemples cette adresse par l'adresse affichée dans la console VirtualBox:*

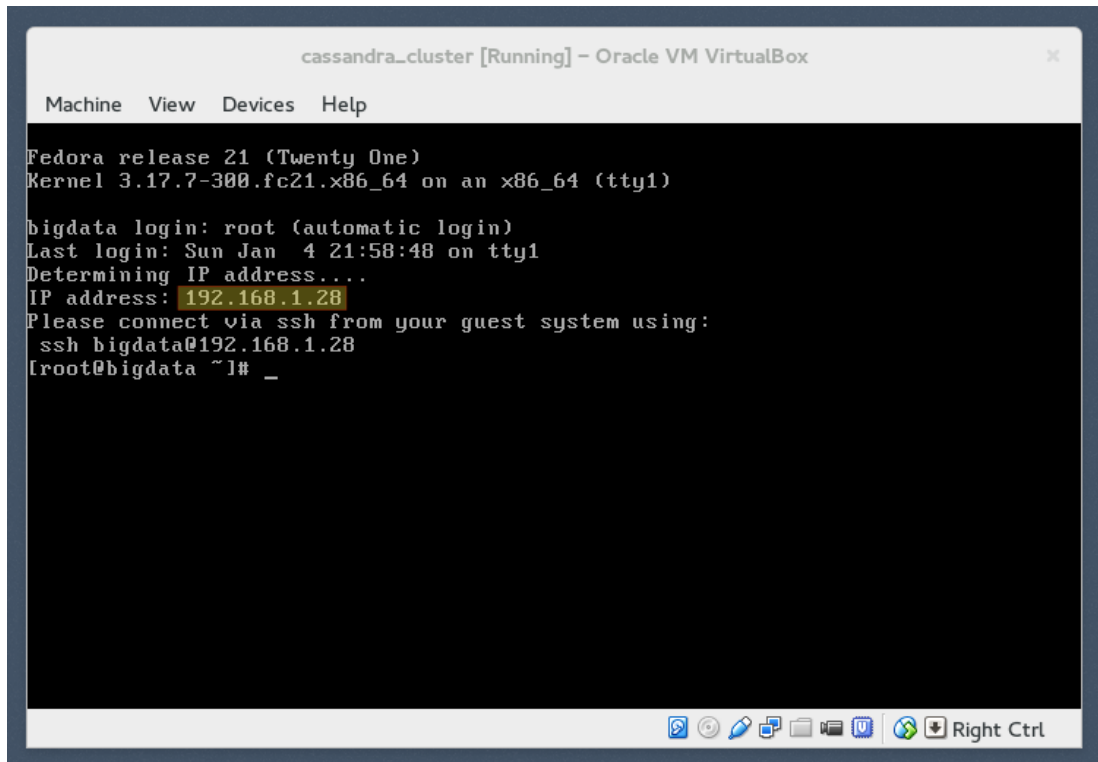


Figure 2. L'adresse IP que vous devez utiliser est affichée dans la fenêtre de la VM

## Connexion en ssh sur la VM

1. Connectez vous en ssh sur la VM depuis un terminal linux :

```
[andrei@desktop ~]$ ssh bigdata@192.168.1.28
bigdata@192.168.1.28's password:
Last login: Sun Jan 4 14:53:32 2015 from pc12.home
[bigdata@bigdata ~]$
```

Identifiants de connexion:



- utilisateur: **bigdata**
- password: **bigdatafuret**

Si vous êtes sur Windows vous pouvez utiliser [putty](#)

# Gestion du cluster via CCM

Dans la VM on a pre-configuré un cluster **test** de 3 noeuds via [CCM](#). Ces noeuds sont bindees sur les adresses **127.0.0.1**, **127.0.0.2** et **127.0.0.3**. Les fichiers de configuration pour ce clusteur sont dans le repertoire **/home/bigdata/.ccm/test**

- Verifier l'etat du cluster **test**:

```
[bigdata@bigdata ~]$ ccm status
```

```
Cluster: 'test'
```

```
node1: DOWN  
node3: DOWN  
node2: DOWN
```

- Demarrer le cluster

```
[bigdata@bigdata ~]$ ccm start
```

- Verifier que le cluster a bien demare et que les noeuds sont UP

```
[bigdata@bigdata ~]$ ccm status
```

```
Cluster: 'test'
```

```
node1: UP  
node3: UP  
node2: UP
```

## Autres commandes utiles:

- afficher des infos sur la configuration d'un noeud

```
[bigdata@bigdata ~]$ ccm node3 show
```

```
node3: UP
  cluster=test
  auto_bootstrap=False
  thrift=('127.0.0.3', 9160)
  binary=('127.0.0.3', 9042)
  storage=('127.0.0.3', 7000)
  jmx_port=7300
  remote_debug_port=0
  initial_token=3074457345618258602
  pid=973
```

- se connecter en cqlsh sur un noeud

```
[bigdata@bigdata ~]$ ccm node1 cqlsh
```

```
Connected to test at 127.0.0.1:9160.
[cqlsh 4.1.1 | Cassandra 2.0.5-SNAPSHOT | CQL spec 3.1.1 | Thrift protocol 19.39.0]
Use HELP for help.
cqlsh>
```

- executer une commande (ex [nodetool](#)) sur un noeud

```
[bigdata@bigdata ~]$ ccm node2 nodetool status
```

```
Datacenter: datacenter1
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Owns (effective)  Host ID
Token
UN  127.0.0.1    50.48 KB      66.7%            e7a2acdd-5ff3-4832-b21a-f75be75c3ba3
-9223372036854775808      rack1
UN  127.0.0.2    50.45 KB      66.7%            f9ce2279-9b4c-42ff-8296-83dc8e374a20
-3074457345618258603      rack1
UN  127.0.0.3    50.45 KB      66.7%            27cb947e-02d2-4318-9c62-a4a835174d24
3074457345618258602      rack1
```

```
[bigdata@bigdata ~]$
```

- arreter/demarrer des noeuds:

```
[bigdata@bigdata ~]$ ccm status

Cluster: 'test'
node1: UP
node3: UP
node2: UP

[bigdata@bigdata ~]$ ccm node1 stop

[bigdata@bigdata ~]$ ccm status

Cluster: 'test'
-----
node1: DOWN
node3: UP
node2: UP

[bigdata@bigdata ~]$ ccm node1 start

[bigdata@bigdata ~]$ ccm status

Cluster: 'test'
node1: UP
node3: UP
node2: UP
```



Pour plus de details sur CCM vous pouvez utiliser [la documentation](#) ainsi que [l'article sur Planet Cassandra](#)

## DevCenter

**DataStax DevCenter** est un client CQL base sur Eclipse. Il offre des fonctionalitees interessantes pour apprendre CQL:

- visualisation des metadonnees (keyspaces, configuration des column families, partition keys, clustering columns etc.)
- auto-completion et auto-correction des requetes
- query trace (on voit qui est le coordonateur, tous les echanges entre les noeuds...)

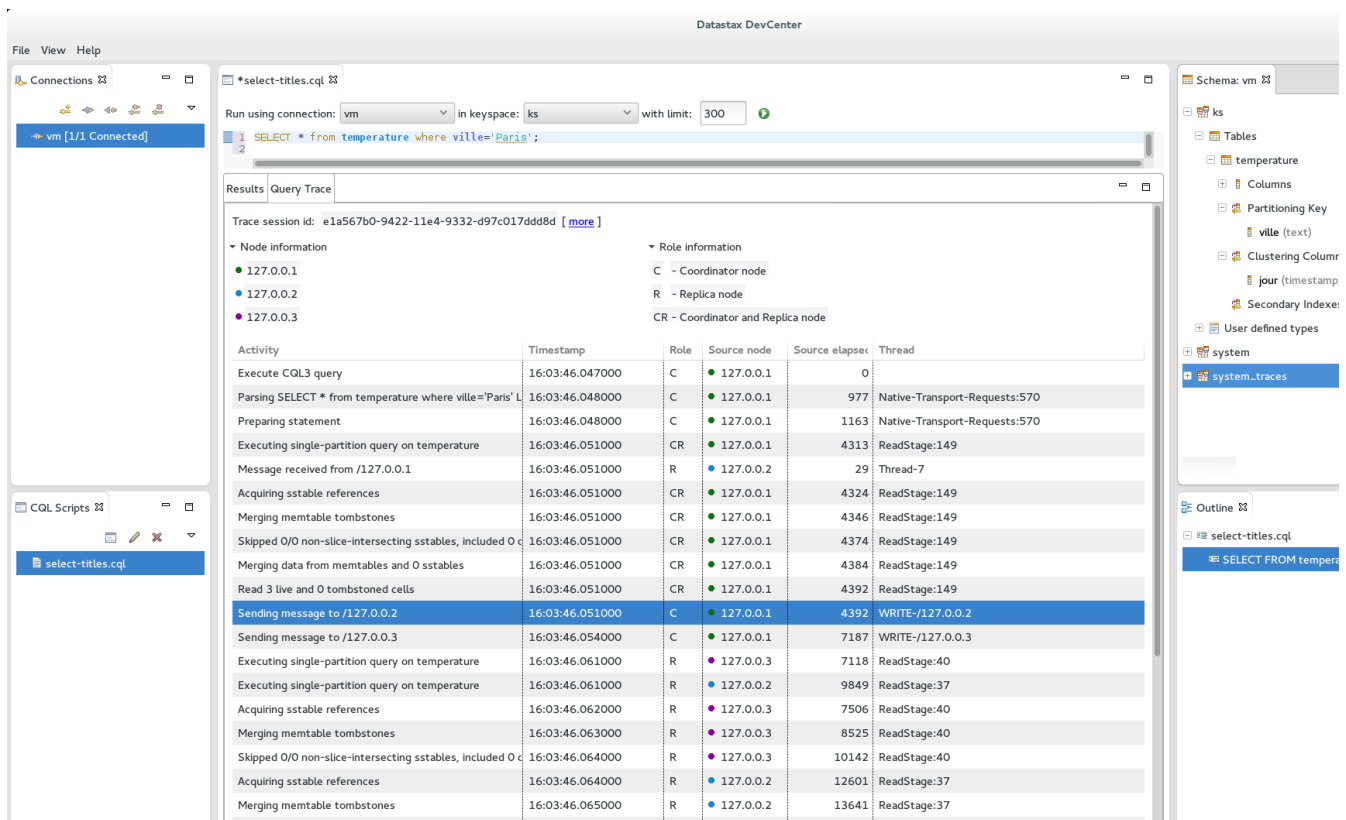


Figure 3. DevCenter features

## Preparation

Chaque noeud du cluster expose plusieurs ports TCP

```
[bigdata@bigdata ~]$ ccm node1 show
```

```
node1: UP
cluster=test
auto_bootstrap=False
thrift=('127.0.0.1', 9160) ①
binary=('127.0.0.1', 9042) ②
storage=('127.0.0.1', 7000)
jmx_port=7300
remote_debug_port=0
initial_token=3074457345618258602
pid=973
```

Pour notre TP on va utiliser des clients qui vont se connecter sur les ports suivants:

- ① Port de connexion via le protocole *thrift*, utilise par un *driver Java*
- ② Port utilise pour une connexion en mode *binary*, utilise par **DevCenter**





Pour accéder à ces ports depuis la machine hôte, on doit configurer une redirection des ports via un tunnel SSH. Par exemple, pour rediriger le port local **9042** vers le port **9042** du **noeud 1** du cluster (qui tourne sur l'ip **127.0.0.1**) lancer la commande suivante:

```
[andrei@desktop ~]$ ssh -nNT -L 9042:127.0.0.1:9042  
bigdata@192.168.1.28
```

bigdata@192.168.1.28's password: ①

① et saisir le mot de passe ssh ( **bigdatafuret** )

Pour **Windows** suivre le tuto suivant: [Configuration Tunnel SSH avec putty](#)

## Installation DevCenter:

1. [Telecharger DevCenter](#)
2. Décompresser cette archive et lancer DevCenter
3. Créer une connexion vers **localhost** sur le port **9042** (qui a été précédemment redirigé via un tunnel ssh vers l'un des nœuds du cluster)
4. Démarrer le TP (créer des keyspaces, column families, lancer des requêtes). [Documentation CQL 3.1](#)

**New Connection**

---

**Basic Settings**

Enter information about the cluster to connect to.

Connection name:

Contact hosts:

localhost

Native Protocol port:

Use compression: ☒ None ☐ Snappy ☐ LZ4

[Try to establish a connection](#) using these settings.

Figure 4. Parameres de configuration de la connexion au cluster