



Cati Sicpa

Ce document est mise à disposition selon les termes de la  
[Licence Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/)

# Configuration d'un cluster Apache Cassandra sous CentOS 7

23 avril 2018

## 1 Objectifs

Dans le premier document, nous avons décrit la procédure d'installation d'Apache Cassandra dans une machine fonctionnant sous CentOS 7. A ce stade, aucune configuration n'est nécessaire si l'on souhaite utiliser chaque machine comme un cluster à un seul noeud. En effet, après avoir démarré Cassandra sur une machine avec la commande `sudo service cassandra start`, l'exécution de la commande `sudo nodetool status` permet de constater qu'on est en présence d'un cluster nommé **Test Cluster** ayant un seul data center constitué d'un seul noeud qui est l'ordinateur sur lequel on est connecté. Cependant, lorsqu'on déploie un cluster Cassandra sur plusieurs machines distinctes, certains paramètres doivent être configurés convenablement sur chaque machine selon la structure de ce cluster. Ainsi, au démarrage, Cassandra détectera automatiquement les différents noeuds de ce cluster et fera fonctionner l'ensemble comme une seule et même entité de sorte que les utilisateurs ou clients puissent se connecter sur n'importe quel noeud de ce système pour accéder au cluster et y travailler. L'objectif de ce deuxième document est de mettre en place un cluster Cassandra constitué de 8 machines virtuelles CentOS 7. Ce cluster sera constitué de 2 data centers. Chaque data center regroupera 2 racks. Chaque rack contiendra 2 machines.

## 2 Pré-requis

Avant mettre en place un cluster, les éléments suivant sont à prendre en compte.

### 2.1 Configuration de l'accès au port du pare-feu

Dans un cluster à multiple noeuds, nous devons ouvrir un certains nombres de ports si les pare-feu sont activés sur les machines hébergeant le cluster Cassandra. Il y a trois ensembles de ports à ouvrir : les ports publics (port 22), les ports de communication entre les noeuds Cassandra (ports 7000, 7001 et 7199) et les ports de communication entre Cassandra et les clients (ports 9042 et 9160).

### 2.2 Définition ou sélection d'un nom pour le cluster

Avant de créer un cluster à plusieurs noeuds, il est important de sélectionner un nom pour ce cluster, pour chacun de ses data centers et pour chacun de ses racks. Notons qu'une

fois ces noms assignés, il est impossible de les modifier ! Nous utiliserons les noms suivants pour décrire la structure de notre cluster. 'DC1' et 'DC2' désigneront respectivement le data center 1 et le data center 2. 'RACK1' et 'RACK2' seront les noms attribués respectivement au rack 1 et au rack 2 dans chaque data center. Enfin, notre cluster sera appelé 'MyFirstCassandraCluster'.

## 2.3 Collection des adresses IP de toutes les machines du cluster

Il est indispensable de connaître les adresses IP de toutes les machines qui vont faire partir du cluster. Dans notre cas, les adresses IP de nos 8 machines virtuelles sont listées ci-dessous. Nous y avons indiqué leur nom ainsi que le data center et le rack dans lequel chacun d'eux appartiendra dans le cluster.

```
node2 10.10.10.2 (DC1, RACK1)
node3 10.10.10.3 (DC1, RACK1)
node4 10.10.10.4 (DC1, RACK2)
node5 10.10.10.5 (DC1, RACK2)
node6 10.10.10.6 (DC2, RACK1)
node7 10.10.10.7 (DC2, RACK1)
node8 10.10.10.8 (DC2, RACK2)
node9 10.10.10.9 (DC2, RACK2)
```

## 2.4 Détermination ou sélection des "seed nodes".

Les "seed nodes" sont les machines à utiliser comme points de départ pour les communications et les échanges d'informations entre les différentes machines du cluster. Il est obligatoire de les définir et il est recommandé d'en avoir au moins deux par data center. Les "seed nodes" permettent également de faciliter la reprise en cas de panne. Ils assurent aussi une transmission rapide des données vers un nouveau noeud joignant le cluster. Dans notre cas, nous choisissons les machines nommées node2, node4, node6 et node8 comme "seed nodes".

# 3 Procédure de configuration

Nous listons ci-dessous les étapes à suivre pour configurer notre cluster Cassandra. On suppose que Apache Cassandra est déjà installé sur toutes les machines de ce cluster.

## 3.1 Ouverture des ports de communications

Il faut ouvrir sur chaque noeud certains ports de communication si le pare-feu est activé. Les commandes suivantes permettent d'ouvrir les ports appropriés et de persister cette configuration :

```
# firewall-cmd --permanent --add-port=22/tcp
# firewall-cmd --permanent --add-port=7000/tcp
```

```
# firewall-cmd --permanent --add-port=7001/tcp
# firewall-cmd --permanent --add-port=7199/tcp
# firewall-cmd --permanent --add-port=9042/tcp
# firewall-cmd --permanent --add-port=9160/tcp
# firewall-cmd --reload
```

On peut par ailleurs vérifier les ports ouverts avec la commande suivante :

```
# firewall-cmd --list-ports
```

### 3.2 Suppression des données

Il faut supprimer les données existantes si Cassandra a déjà été lancé sur une machine. Pour cela nous pouvons exécuter les commandes suivantes :

```
# sudo service cassandra stop
# sudo rm -rf /var/lib/cassandra/data/system/*
```

Le but est de supprimer le nom de cluster par défaut '**Test Cluster**' pour que tous les noeuds aient par la suite le même nom '**MyFirstCassandraCluster**' que nous attribuerons à notre cluster. Cette suppression doit être faite sur toutes les machines du cluster.

### 3.3 Modification du fichier "cassandra.yaml"

Les propriétés du cluster sont spécifiées dans le fichier "**cassandra.yaml**" de chaque machine. Il existe plusieurs dizaines de paramètres à configurer. Nous laisserons certains paramètres avec leurs valeurs par défaut que nous pouvons par ailleurs modifier plus tard en fonction des besoins. Certaines de ces propriétés sont communes à toutes les machines tandis que d'autres sont propres à chaque noeud.

Les propriétés communes que nous modifieront sont les suivantes.

- **cluster\_name** : Le nom du cluster est indiqué comme valeur pour ce paramètre. Le nom permet d'éviter qu'une même machine appartienne simultanément à deux clusters Cassandra différents.
- **seeds** : Les adresses IP des machines utilisées comme "seed nodes" sont indiquées comme valeurs pour ce paramètre.

- **endpoint\_snitch** : Ce paramètre indique à Cassandra le mode d'orientation des requêtes ainsi que le mode de répartition des répliquions à travers le cluster. C'est tout simplement le mode souhaité pour le fonctionnement du cluster.

Les propriétés spécifiques à chaque nœud que nous modifierons sont les suivantes.

- **listen\_address** : Ce paramètre reçoit l'adresse IP du nœud considéré. C'est celle qu'utilisent les autres nœuds Cassandra pour s'y connecter. Si cette valeur n'est pas spécifiée, elle est inférée de la machine machine hébergeant l'instance Cassandra.
- **rpc\_address** : Ce paramètre reçoit l'adresse IP permettant la connexion d'un client à cette machine. Si cette valeur n'est pas spécifiée, elle est inférée de la machine machine hébergeant l'instance Cassandra.

Sur chaque nœud, nous pouvons éditer le fichier "**cassandra.yaml**" avec la commande suivante :

```
# vi /etc/cassandra/conf/cassandra.yaml
```

Ensuite, on recherche les paramètres concernés et on leur attribue les valeurs indiquées ci-dessous :

```
cluster_name: 'MyFirstCassandraCluster'

seed_provider:
  - class_name: org.apache.cassandra.locator.SimpleSeedProvider
    - seeds: 10.10.10.2,10.10.10.4,10.10.10.6,10.10.10.8
      (ce sont les adresses IP de tous les "seed nodes")

listen_address: (mettre ici l'adresse IP de la machine concernée)

rpc_address: (mettre ici l'adresse IP de la machine concernée)

endpoint_snitch: GossipingPropertyFileSnitch
                 (C'est la valeur recommandée en production)
```

Enfin, on enregistre et on ferme le fichier "**cassandra.yaml**" après ces modifications.

### 3.4 Modification du fichier "**cassandra-rackdc.properties**"

Le data center et le rack dans lequel appartient un nœud sont spécifiés dans le fichier "**cassandra-rackdc.properties**" de chaque machine. Sur chaque nœud, on peut éditer le fichier "**cassandra-rackdc.properties**" avec la commande suivante :

```
# vi /etc/cassandra/conf/cassandra-rackdc.properties
```

Ensuite, on recherche les paramètres **"dc"** et **"rack"** pour les attribuer les noms appropriés pour la localisation du noeud considéré :

```
dc= (mettre ici le nom du data center dans lequel se trouve ce noeud)
```

```
rack= (mettre ici le nom du rack dans lequel se trouve ce noeud)
```

Enfin, on enregistre et on ferme le fichier **"cassandra-rackdc.properties"** après ces modifications.

### 3.5 Suppression du fichier **"cassandra-topology.properties"**

La propriété **"GossipingPropertyFileSnitch"** que nous avons attribuée au paramètre **endpoint\_snitch**, n'empêche pas Cassandra de charger le fichier **"cassandra-topology.properties"** si ce dernier existe. Il faut donc le supprimer de chaque machine du cluster car il utilise un mode de communication et de transfert des données qui est différent de celui que nous avons choisi. On peut supprimer ce fichier avec la commande suivante :

```
# rm /etc/cassandra/conf/cassandra-topology.properties
```

### 3.6 Redémarrage des noeuds

Nous devons redémarrer chaque noeud du cluster pour que les modifications effectuées ci-dessus dans les fichiers **"cassandra.yaml"** et **"cassandra-rackdc.properties"** soient prises en compte. Pour cela, on peut utiliser la commande suivante :

```
# reboot
```

## 4 Démarrage et arrêt du cluster

Après installation et configuration de toutes les machines du cluster, il faut lancer Cassandra sur toutes les machines de ce cluster en démarrant premièrement les machines définies comme **"seed nodes"**. Pour notre cluster, nous commencerons par démarrer les node2, node4, node6 et node8 avant de démarrer les node3, node5, node7 et node9. Pour démarrer Cassandra, on peut utiliser la commande suivante :

```
# sudo service cassandra start
```

On peut vérifier si le cluster fonctionne correctement en exécutant sur l'un de ses noeuds la commande suivante :

```
# nodetool status
```

On peut se connecter au cluster à l'aide de la commande **"cqlsh"** suivie de l'adresse IP de l'un des noeuds du cluster. Par exemple, si l'on souhaite se connecter au cluster par le "node2" on peut utiliser la commande suivante :

```
# cqlsh 10.10.10.2
```

On peut exécuter la commande ci-dessous pour quitter le shell `cqlsh` de Cassandra :

```
cqlsh> exit;
```

Pour arrêter le cluster, il faut stopper les services sur chacun de ses noeuds. On peut exécuter la commande ci-dessous sur chaque machine du cluster pour l'arrêter :

```
# sudo service cassandra stop
```

## 5 Conclusion

Dans ce document, nous avons utilisé 8 machines virtuelles fonctionnant sous CentOS 7 pour illustrer la procédure d'installation d'un cluster Cassandra constitué de deux data centers. Dans les prochains documents, nous allons installer et configurer un cluster Apache Spark sur ces mêmes machines virtuelles. Puis, nous allons connecter ce cluster Spark au cluster Cassandra.

## Références