



Cati Sicpa

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

# Installation d'un cluster ZooKeeper sous CentOS 7 (Paramétrage alternative)

28 août 2018

## 1 Objectifs

Dans le document nommé "**Install ZooKeeper Cluster**" Nous avons décrit la configuration de l'ensemble zookeeper que nous avons mis en place pour coordonner les services d'un cluster Apache Spark afin de rendre ce dernier hautement disponible et sans point unique de défaillance. Le contenu de ce document décrit l'ensemble zookeeper que nous avons mis en place pour coordonner les services d'un cluster Apache Solr afin de le rendre également hautement disponible et sans point unique de défaillance.

Un système distribué est un ensemble de plusieurs machines indépendantes interconnectées par un réseau et qui exploite convenablement les ressources de chaque machine pour fournir un service donné comme s'il s'agissait d'une seule machine géante. Un des problèmes des systèmes distribués est celui de la coordination des actions ou opérations entre les différentes machines de façon à toujours garantir une bonne qualité du service malgré les incidents qui peuvent subvenir durant son fonctionnement. Apache ZooKeeper est un logiciel open source de la fondation Apache qui fournit une solution aux problèmes variés de coordinations des tâches dans des larges systèmes distribués. En production, il est recommandé d'utiliser un nombre impair d'instances ZooKeepers installées dans des serveurs différents. Chacune de ces instances maintiendra les mêmes informations sur le système. L'une de ces instances sera élue comme leader et les autres instances seront des followers. En cas de défaillance du leader, l'un des followers est automatiquement élu comme leader pour assurer la continuité du service. Dans ce document, nous allons décrire la procédure d'installation et de configuration d'un cluster Zookeeper constitué de trois machines distinctes fonctionnant sous CentOS 7. Plus précisément, nous installerons zookeeper-3.4.11 qui est une version stable de Zookeeper.

## 2 Installation de Zookeeper dans chaque machine

### 2.1 Installation de Java

Zookeeper est implémenté en Java. Ainsi, nous devons avoir une version récente de Java installée dans le système. Pour installer la version récente OpenJDK de Java, nous pouvons exécuter la commande suivante :

```
# sudo yum install java -y
```

Nous pouvons vérifier la version installée de Java avec la commande suivante :

```
# sudo java -version
```

## 2.2 Ouverture des ports TCP

Si le pare-feu est activé dans les machines, il est nécessaire d'ouvrir les ports TCP 2181, 2888 et 3888. On peut le faire avec les commandes ci-dessous :

```
# firewall-cmd --permanent --add-port=2181/tcp
```

```
# firewall-cmd --permanent --add-port=2888/tcp
```

```
# firewall-cmd --permanent --add-port=3888/tcp
```

```
# firewall-cmd --reload
```

On peut également vérifier les ports ouverts avec la commande suivante :

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

## 2.3 Téléchargement de Zookeeper

Pour installer Zookeeper dans notre cluster, il faut télécharger et décompresser le fichier `zookeeper-3.4.11.tar.gz` dans chacune des trois machines. Pour cela, nous pouvons par exemple exécuter les commandes suivantes :

```
# yum install wget
```

```
# wget https://archive.apache.org/dist/zookeeper/  
zookeeper-3.4.11/zookeeper-3.4.11.tar.gz
```

```
# tar -zxf zookeeper-3.4.11.tar.gz
```

```
# cp -r zookeeper-3.4.11 /opt/zookeeper-3.4.11
```

```
# cd /opt/zookeeper-3.4.11
```

## 2.4 Définition des variables d'environnement

Nous allons définir les variables `ZK_HOME` et `PATH` de façon permanente et globale pour tous les utilisateurs. Pour cela, nous éditons le fichier `/etc/profile` de chaque machine avec la commande suivante :

```
# vi /etc/profile
```

Ensuite, nous insérons les deux lignes ci-dessous juste avant la ligne ayant contenant le texte `export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL`

```
export ZOOKEEPER_HOME=/opt/zookeeper-3.4.11
export PATH=$PATH:/opt/zookeeper-3.4.11/bin
```

A la fin de cette modification, nous enregistrons et fermons ce fichier. Ensuite, nous exécutons la commande ci-dessous pour que ces modifications soient immédiatement présent en compte par le système :

```
# source /etc/profile
```

### 3 Configuration d'un cluster de 3 machines Zookeeper

Nous avons suivi la procédure décrite dans la section précédente pour installer Zookeeper dans trois machines distinctes. Il s'agit de nos machines virtuelles nomées **node11**, **node12** et **node13**. Dans cette section, nous allons configurer ces trois machines en un ensemble Zookeeper. Pour configurer notre cluster, nous devons créer un fichier nommé `zoo.cfg` et un fichier nommé `zookeeper-env.sh` dans le répertoire de configuration de ZooKeeper. Nous devons également créer un fichier nommé `myid` dans un répertoire à spécifier pour stockage des données de ZooKeeper.

#### 3.1 Création et modification du fichier "zoo.cfg"

Dans chaque machine du cluster, nous pouvons éditer le fichier `zoo.cfg` avec la commande suivante :

```
# mkdir -p /var/lib/zookeeper/data

# cat > /opt/zookeeper-3.4.11/conf/zoo.cfg << "EOF"
tickTime=2000
dataDir=/var/lib/zookeeper/data
clientPort=2181

initLimit=5
syncLimit=2
server.1=10.10.10.11:2888:3888
server.2=10.10.10.12:2888:3888
server.3=10.10.10.13:2888:3888

autopurge.snapRetainCount=3
autopurge.purgeInterval=1
EOF
```

Les termes `10.10.10.x` sont ne sont rien d'autres que les adresses IP des trois machines que nous avons choisies pour notre cluster. Il s'agit de nos machines virtuelles nommées `node11`, `node12` et `node13`.

### 3.2 Création et modification du fichier "zookeeper-env.sh"

Dans chaque machine du cluster, nous pouvons éditer le fichier `zookeeper-env.sh` avec la commande suivante :

```
# mkdir -p /var/lib/zookeeper/log

# cat > /opt/zookeeper-3.4.11/conf/zookeeper-env.sh << "EOF"
ZOO_LOG_DIR="/var/lib/zookeeper/log"
ZOO_LOG4J_PROP="INFO,ROLLINGFILE"

SERVER_JVMFLAGS="-Xms2048m -Xmx2048m -verbose:gc -XX:+PrintHeapAtGC -XX:+PrintGCDetails
-XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+PrintTenuringDistribution
-XX:+PrintGCApplicationStoppedTime -Xloggc:$ZOO_LOG_DIR/zookeeper_gc.log
-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=9 -XX:GCLogFileSize=20M"
EOF
```

### 3.3 Création et modification du fichier myid dans chaque machine

On peut éditer le fichier `myid` dans la machine nommée **node11** avec les commandes suivantes

```
cat > /var/lib/zookeeper/data/myid << "EOF"
1
EOF
```

On peut éditer le fichier `myid` dans la machine nommée **node12** avec les commandes suivantes

```
cat > /var/lib/zookeeper/data/myid << "EOF"
2
EOF
```

On peut éditer le fichier `myid` dans la machine nommée **node13** avec les commandes suivantes

```
cat > /var/lib/zookeeper/data/myid << "EOF"
3
EOF
```

Les numéros 1, 2 ou 3 permettent tout simplement d'identifier et de distinguer chaque serveur Zookeeper faisant partir du cluster.

## 4 Démarrage et arrêt du cluster

Pour démarrer notre cluster Zookeeper, nous devons exécuter sur chaque machine la commande suivante :

```
# zkServer.sh start
```

Après démarrage du cluster, nous pouvons exécuter la commande suivante dans chaque machine pour savoir laquelle est élu leader et lesquelles sont les followers :

```
# zkServer.sh status
```

Le shell de Zookeeper pour notre cluster se lance avec la commande ci-dessous dans laquelle les adresses IP des noeuds utilisés ainsi que le port d'écoute de Zookeeper sont indiqués :

```
# zkCli.sh -server 10.10.10.11:2181,10.10.10.12:2181,10.10.10.13:2181
```

On peut quitter ce shell en tapant la commande

```
>close
```

Pour arrêter le cluster, il faut par exemple exécuter dans chaque machine la commande ci-dessous :

```
# zkServer.sh stop
```

## 5 Conclusion

Nous avons montré dans ce document les étapes d'installation et de configuration d'un cluster ZooKeeper constitué des trois serveurs distincts. Cela nous sera absolument utile dans la suite pour la mise en place d'un cluster Apache Solr qui soit hautement disponible et n'ayant pas de point unique de défaillance.

## Références