Sécurisation d'un cluster Apache Cassandra

28 août 2018

1 Objectifs

Les deux éléments essentiels à configurer pour sécuriser un cluster Cassandra sont l'authentification et l'autorisation. L'authentification permet de définir les utilisateurs qui peuvent se connecter au cluster. L'autorisation permet de définir les droits d'accès aux ressources ou objets du cluster. Dans la plupart des bases de données, les termes utilisateurs et rôles sont utilisés pour désigner deux entités différentes. Les utilisateurs sont les comptes de connection à la base de données et les rôles sont les privilèges ou les droits que l'on accorde à chaque utilisateur sur les objets de la base de données. Dans Cassandra les comptes utilisateurs et les privilèges sont crées et manipulées de la même manière avec le mot clé "`ROLE`". La meilleure pratique consiste à créer séparément les groupes de privilèges et les comptes utilisateurs et d'assigner les comptes utilisateurs aux différents groupes de privilèges. Par défaut, Cassandra n'exige pas d'authentification ni d'autorisation pour se connecter et accéder aux ressources du cluster. Ce document présente la mise en place et la gestion de l'authentification et de l'autorisation dans un cluster Cassandra.

2 Configuration de l'authentification et de l'autorisation

2.1 Réplication multiple du keyspace "*system_auth*"

Dans Cassandra, les informations d'identification des rôles sont stockées dans les tables du keyspace "*system_auth*". Par défaut, ce keyspace est répliqué une seule fois dans le cluster Cassandra. La première étape pour sécuriser un cluster Cassandra consiste à se connecter sur un des nodes du cluster via le shell "`cqlsh`" de Cassandra. Puis, de changer la stratégie de réplication et le facteur de réplication de ce keyspace dans chaque data center du cluster de façon à augmenter sa disponibilité. Une meilleure pratique consiste à définir 3 à 5 copies de ce keyspace dans chaque data center. Pour notre cluster cassandra, nous avons réalisé cela de la manière suivante :

```
# cqlsh 10.10.10.2

ALTER KEYSPACE system_auth WITH
replication = {'class': 'NetworkTopologyStrategy', 'DC1': 3, 'DC2': 3};

exit;
```

Pour forcer le changement du facteur de réplication du keyspace "*system_auth*", nous éxécutons la commande suivante :

```
# nodetool repair system_auth
```

2.2 Changement des options d'authentification et d'autorisation

Dans Cassandra, les paramètres de sécurité dont les options doivent être modifiées sont "`authenticator`" pour l'authentification et "`authorizer`" pour l'autorisation. Ces deux paramètres se trouvent dans le fichier de configuration "`cassandra.yaml`".

"`AllowAuthenticator`" est la première option du paramètre "`authenticator`". Cette option est définie par défaut et indique à Cassandra de ne pas effectuer de vérification lors de la connection d'un client au cluster. Alternativement, "`PasswordAuthenticator`" est la seconde option du paramètre "`authenticator`". Cette option indique à Cassandra de ne permettre l'accès au cluster qu'aux utilisateurs ayant un droit d'accès préalablement créé par un super utilisateur.

"`AllowAuthorizer`" est la première option du paramètre "`authorizer`". Cette option est définie par défaut et indique à Cassandra de permettre à n'importe quel utilisateur connecté d'effectuer n'importe quelle action dans le cluster. Alternativement, "`CassandraAuthorizer`" est la seconde option du paramètre "`authorizer`". Cette option indique à Cassandra de restreindre à chaque utilisateur connecté au cluster uniquement l'accès aux ressources ou privilèges qui lui sont préalablement sont accordés par un utilisateur accrédidé à le faire.

La dexième étape pour configurer la sécurité dans un cluster Cassandra consiste à changer convenablement les options des deux paramètres "`authenticator`" et "`authorizer`" dans chaque node du cluster et de redémarrer le node pour que ces changement soient prises en compte. Nous réalisons cela en effectuant les actions suivantes dans chaque node de notre cluster. Si le cluster contient déjà des données, tous les nodes ne doivent pas être configurés simultanément. Dans ce cas, il faut réaliser toutes les opérations ci-dessous sur un seul node avant de passer à un autre node.

i) Nous arrêtons cassandra sur un node avec la commande suivante :
```
# sudo service cassandra stop
```

ii) Nous éditons le fichier "`cassandra.yaml`" avec la commande suivante :
```
# vi /etc/cassandra/conf/cassandra.yaml
```

iii) Nous recherchons la ligne suivante :
```
authenticator: AllowAllAuthenticator
```

iv) Nous remplaçons la ligne trouvée à l'étape iii) par la ligne ci-dessous :

```
authenticator: PasswordAuthenticator
```

v) De même, nous recherchons la ligne suivante :
```
authorizer: AllowAllAuthorizer
```

vi) Nous remplaçons la ligne trouvée à l'étape v) par la ligne ci-dessous :
```
authorizer: CassandraAuthorizer
```

vii) Nous enregistrons et fermons le fichier "`cassandra.yaml`".

viii) Nous redémarrons le node
```
# reboot
```

ix) Nous redémarrons Cassandra sur le node (si ce dernier n'est pas configuré pour se lancer autmatiquement au démarrage du node)
```
# sudo service cassandra start
```

2.3 Désactivation du compte utilisateur par défaut "*cassandra*"

A la fin des deux étapes ci-dessus, l'authentification est désormais requise pour se connecter au cluster Cassandra. A ce stade de la configuration, seul le compte utilisateur "*cassandra*" ayant pour mot de passe "*cassandra*" est diponible et est le compte super utilisateur par défaut. Ce compte super utilisateur par défaut est bien connu par toutes les personnes qui ont une bonne connaissance du fonctionnement de Cassandra. Ainsi, pour renforcer la sécurité du cluster Cassandra, il est recommandé de créer un ou deux nouveaux comptex supers utilisateurs, de leur attribuer tous les droits et de désactiver le compte super utilisateur "*cassandra*" par défaut. Nous réalisons cela dans cette troisième étape de sécurisation d'un cluster Cassandra en effectuant les actions qui suivent dans un des node de notre cluster.

i) On se connecte au shell "`cqlsh`" du "`node2`" avec le compte "*cassandra*"
```
# cqlsh  10.10.10.2  -u cassandra  -p  cassandra
```

ii) On crée deux nouveaux supers utilisateurs qui auront tous les droits d'administration dans le cluster
```
CREATE ROLE 'dba_1' WITH SUPERUSER = true AND LOGIN = true
AND PASSWORD = 'dba_1*pw!';
CREATE ROLE 'dba_2' WITH SUPERUSER = true AND LOGIN = true
AND PASSWORD = 'dba_2*pw!';
```
iii) On autorise l'accès à tous les objets ou ressources du cluster à ces nouveaux administrateurs
```
GRANT ALL PERMISSIONS ON ALL KEYSPACES TO dba_1;
GRANT ALL PERMISSIONS ON ALL KEYSPACES TO dba_2;
GRANT ALL PERMISSIONS ON ALL ROLES TO dba_1;
GRANT ALL PERMISSIONS ON ALL ROLES TO dba_2;
```

```
GRANT ALL PERMISSIONS ON ALL FUNCTIONS TO dba_1;
GRANT ALL PERMISSIONS ON ALL FUNCTIONS TO dba_2;
GRANT ALL PERMISSIONS ON ALL MBEANS TO dba_1;
GRANT ALL PERMISSIONS ON ALL MBEANS TO dba_2;
```

iv) On se déconnecte et on se reconnecte au cluster avec l'un des comptes supers utilisateurs nouvellement créés

```
exit;
# cqlsh  10.10.10.2  -u 'dba_1'  -p  'dba_1*pw!'
```

v) On désactive le super utilisateur par défaut "*cassandra*"

```
ALTER ROLE cassandra WITH SUPERUSER = false AND LOGIN = false;
```

3 Création et gestion des utilisateurs et des privilèges

Dans cette section, nous allons créer d'autres utilisateurs et les assigner des rôles ou des privilèges. Dans un cluster Cassandra, on réalise cette opération en quatre étapes.

i) On crée un ou plusieurs utilisateurs.

ii) On crée un ou plusieurs rôles (groupes d'utilisateurs).

iii) On accorde les droits ou privilèges souhaités aux rôles créés.

iv) On accorde à chaque utilisateur les droits d'un ou de plusieurs rôles.

Le langage des requêtes CQL (Cassandra Query Language) de Cassandra utilise le mot clé "`ROLE`" pour représenter à la fois les utilisateurs et les groupes d'utilisateurs. Un groupe d'utilisateur est défini pour spécifier un ensemble de privilèges ou de droits.

On peut autoriser les permissions sur les objets suivants de la base de données : les keyspace, les tables, les fonctions, les comptes utilisateurs (roles). La liste des permissions que l'on peut accorder sur les objets de la base de données sont : `CREATE`, `ALTER`, `DROP`, `SELECT`, `MODIFY`, `DESCRIBE`, `EXECUTE`, `AUTHORIZE`.

Le fichier "*Cassandra_user_roles.cql*" contient quelques scripts qui permettent de créer des comptes utilisateurs et de les accorder certains droits. Pour la suite, nous supposons que ce fichier est placé dans le répertoire "`/usr/share/data/`" du "`node2`" de notre cluster Cassandra. Nous éxécuterons ces scripts dans le shell "`cqlsh`" de Cassandra. Tel que spécifiés dans ce fichier de scripts, les éventuels résultats seront automatiquement stockés dans un fichier texte situé dans le même répertoire. Pour commencer, on se connecte sur un des nodes du cluster Cassandra avec le compte d'accès de l'un des administrateurs. Nous choisissons de nous connecter au "node2" de notre cluster Cassandra.

```
# cqlsh  10.10.10.2  -u  'dba_1'  -p  'dba_1*pw!'
```

Une fois connecté au shell "`cqlsh`" de Cassandra, on peut lister les keyspaces dans le cluster avec la commande suivante :

```
describe keyspaces;
```

Pour ce qui est de la création et de la gestion de quelques comptes utilisateurs, nous pouvons éxécuter manuellement dans ce shell "`cqlsh`" actif de Cassandra les lignes de scripts contenus dans le fichier "*Cassandra_user_roles.cql*". On peut alternativement lancer l'éxécution séquentielle de tous les scripts contenus dans ce fichier avec la commande suivante :

```
SOURCE '/usr/share/data/Cassandra_user_roles.cql'
```

Nous avons spécifié à Cassandra dans le fichier de scripts de ne pas afficher les éventuels résultats directement dans la console, mais de les enregistrer dans un fichier texte que nous pouvons consulter avec la commande suivante :

```
#vi /usr/share/data/user_account_manager_output.txt
```

4 Conclusion

Dans ce document nous avons décrit une procédure permettant de configurer la sécurité dans un cluster Cassandra. Ensuite, nous avons présenté et illustré avec quelques exemples la démarche à suivre pour créer des utilisateurs et pour leurs accorder certains privilèges sur les ressources ou objets de notre cluster Cassandra. Dans le prochain document, nous allons décrire la procédure d'ajout, de suppression et de remplacement d'un node dans un cluster Cassandra.

Références