Sodebo - Cluster RabbitMQ

Mehdi EL KOUHEN mehdi.elkouhen@softeam.fr

L'objectif de cette étude est de mettre en place un cluster de serveurs RabbitMQ sur deux sites Sodebo : un site en France et un site au Brésil.

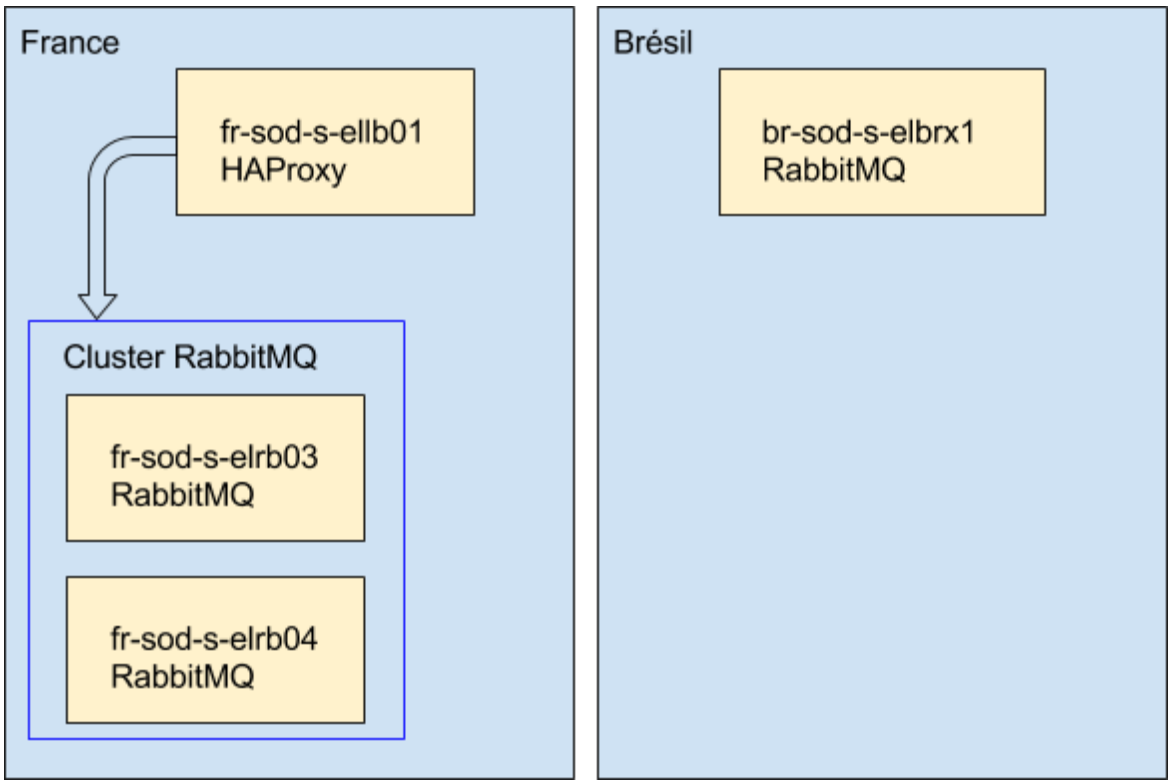
Comme RabbitMQ recommande de ne pas mettre en place un cluster sur des sites distants géographiquement, nous avons mis en place deux clusters (un cluster sur le site Français, un cluster sur le site Brésilien) ainsi qu'un mécanisme de synchronisation entre les deux clusters. La synchronisation est réalisée en utilisant le plugin shovel de RabbitMQ.

Tables des matières

Architecture

Dans notre architecture, nous avons mis en place un cluster de serveurs RabbitMQ pour la France et (pour simplifier) une instance unique RabbitMQ pour le Brésil.

Comme le dimensionnement du cluster peut varier, nous avons mis en place un load balancer tcp (HAProxy) en frontal du cluster. De cette manière les clients RabbitMQ n'ont besoin de "connaître" que le load balancer. Nous pouvons donc ajouter ou supprimer des instances RabbitMQ au cluster sans impacter les applications clientes.



Pour rendre l'architecture haute dispo, nous avons mis en place une VIP (keepalived) sur les deux load balancers (fr-sod-s-ellb01 et fr-sod-s-ellb02).

Le nom de domaine de la VIP est rmq.sodebo.fr

Les clients RabbitMQ envoient les messages au serveur rmq.sodebo.fr sur le port 5672.

Installation de RabbitMQ

Installation des packages

```
sudo rpm --import https://www.rabbitmq.com/rabbitmq-release-signing-key.asc
sudo yum install erlang
sudo yum install
https://github.com/rabbitmq/rabbitmq-server/releases/download/rabbitmq_v3_6_5/rabbitmq-server-3.6.5-1.noarch.rpm
```

Démarrage

Démarrer RabbitMQ sur la VM master et les VMs esclave.

```
sudo systemctl start rabbitmq-server
```

Configuration de la console web

RabbitMQ s'administre via une console web qu'il faut installer (via la commande rabbitmq-plugins).

```
sudo rabbitmq-plugins list
sudo rabbitmq-plugins enable rabbitmq_management
```

Configuration réseau

Nous avons dû ouvrir les ports suivants

- 4369
- 5667-5999

Configuration du cluster

Vous trouverez ci-dessous la configuration du clustering RabbitMQ

- http://www.rabbitmq.com/clustering.html

Il existe différentes solutions pour configurer un cluster RabbitMQ. Soit en utilisant rabbitmqctl, soit en manipulant directement les fichiers de config RabbitMQ, soit en utilisant un plugin (rabbitmq-autocluster, rabbitmq-clusterer).
Nous avons choisi d'utiliser rabbitmq

- Manipuler directement les fichiers de config peut s'avérer risqué
- Le plugin rabbitmq-autocluster nécessite d'utiliser un "service discovery" (exemple: consul)

- L'utilisation de plugin rabbitmq-clusterer n'est pas toujours compatible avec l'utilisation de rabbitmqctl

Pré-condition : le master et les slaves sont éteints

Démarrer le master (exemple : fr-sod-s-elrb03)
```
sudo systemctl start rabbitmq-server
```

Copier le fichier /var/lib/rabbitmq/.erlang.cookie du master vers les slaves

Configurer les slaves pour rejoindre le cluster (exemple : fr-sod-s-elrb04)

```
sudo rabbitmq-server -detached
sudo rabbitmqctl stop_app
sudo rabbitmqctl join_cluster rabbit@fr-sod-s-elrb03
sudo rabbitmqctl start_app
```

Eteindre les slaves
```
sudo rabbitmqctl stop
```

Redémarrer les services RabbitMQ sur les slaves
```
sudo systemctl start rabbitmq-server
```

Vérifier l'état du cluster sur le master et les slaves
```
sudo rabbitmqctl cluster_status
```

```
Le résultat doit bien référencer tous les noeuds attendus (fr-sod-s-elrb03,
fr-sod-s-elrb04).

[{nodes,[{disc,['rabbit@fr-sod-s-elrb03','rabbit@fr-sod-s-elrb04']}]},
 {running_nodes,['rabbit@fr-sod-s-elrb04','rabbit@fr-sod-s-elrb03']},
 {cluster_name,<<"rabbit@fr-sod-s-elrb03.sodebo.fr">>},
 {partitions,[]},
 {alarms,[{'rabbit@fr-sod-s-elrb04',[]},{'rabbit@fr-sod-s-elrb03',[]}]}]
```

Le fichier de config est ici : /var/lib/rabbitmq/mnesia/rabbit@XXX/cluster_nodes.config

Gestion des utilisateurs

Vous trouverez ci-dessous la documentation RabbitMQ de gestion des droits des utilisateurs
- http://www.rabbitmq.com/access-control.html

Voici un exemple de création d'utilisateurs via rabbitmqctl. La création peut être réalisée via la console web.

```
sudo rabbitmqctl add_user svc_bresil password
```

```
sudo rabbitmqctl set_user_tags svc_bresil administrator
sudo rabbitmqctl set_permissions -p / svc_bresil ".*" ".*" ".*"
```

L'utilisateur guest créé par défaut n'est utilisable qu'en localhost.

Installation HAProxy

Installation des packages

```
sudo yum install haproxy
```

Configuration HAProxy

Extrait du fichier /etc/haproxy/haproxy.cfg

```
defaults
        mode                    tcp

listen rabbitmq 172.16.17.17:5672

server rabbitmaster 172.16.17.20:5672 check inter 5s rise 2 fall 3
server rabbitslave  172.16.17.19:5672 check inter 5s rise 2 fall 3
```

172.16.17.17 est l'adresse IP du serveur HAProxy

Redémarrage du serveur

```
sudo systemctl restart haproxy
```

Configuration RabbitMQ

Avant de mettre en place

sudo rabbitmqctl set_policy ha-all "" '{"ha-mode":"all","ha-sync-mode":"automatic"}'

Script de Test

```
#!/usr/bin/env python
import pika
import sys

try:

        parameters = pika.ConnectionParameters(
                '172.16.17.17', 5672, '/', pika.PlainCredentials('svc_france', 'password'))

        connection = pika.BlockingConnection(parameters)

        channel = connection.channel()
```

```
        channel.basic_publish(exchange='',
                              routing_key='queue1',
                              body='Hello World !')

        print(" [x] Sent 'Hello World!'")

        connection.close()

except:
        print("Unexpected error:", sys.exc_info())
```

Mise en place de la synchro

Nous avons mis en place le plugin shovel pour gérer la synchronisation site à site de files : i.e. redispatcher les messages reçus par une file du cluster vers l'instance RabbitMQ Brésilien.

Installation des plugins

```
sudo rabbitmq-plugins enable rabbitmq_shovel
sudo rabbitmq-plugins enable rabbitmq_shovel_management
```

Configuration du shovel

La configuration peut se faire via la ligne de commande (cf. ci-dessous) ou via la console d'admin.

```
sudo rabbitmqctl set_parameter shovel my-shovel '{"src-uri": "amqp://", "src-queue":
"queue1",
"dest-uri": "amqp://svc_bresil:password@br-sod-s-elbrx1", "dest-queue": "queue1"}'
```

Test

Envoyer un message vers la queue queue1 du cluster et vérifier que le message arrive bien dans la queue queue1 de br-sod-s-elbrx1