Cluster Postgres (Patroni + ETCD + HAProxy) – RHEL8

Description

O que é Patroni ?

O **Patroni** é um conjunto de ferramentas de software livre, que é escrito em Python e que garante a configuração de ponta a ponta dos clusters do **PostgreSQL** HA, incluindo replicação de streaming.

O que é ETCD ?
O etcd é um **Distributed Consensus Store (DCS)** e é necessário para fornecer um mecanismo de tomada de decisão distribuído para permitir que o Patroni determine qual **instância deve ser a líder e qual deve ser a réplica**.

O que é HAProxy ?

O **HAProxy** é um software de código aberto e gratuito que fornece um balanceador de carga e servidor proxy de alta disponibilidade para aplicativos baseados em TCP/IP e HTTP que espalha solicitações por vários servidores.

Instalar tudo nos 2 hosts – Proceumento homologado em RHEL 8

Instalar Postgres

```
dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86
dnf -qy module disable postgresql
dnf install -y postgresql14-server
```

Não subir o postgres depois da instalação

Instalar etcd
Instale pacotes basicos

```
dnf -y install curl wget vim
```

Download do ETCD

```
ETCD_RELEASE=$(curl -s https://api.github.com/repos/etcd-io/etcd/releases/late
echo $ETCD_RELEASE
wget https://github.com/etcd-io/etcd/releases/download/${ETCD_RELEASE}/etcd-${
```

Extrair o pacote

```
tar xvf etcd-${ETCD_RELEASE}-linux-amd64.tar.gz
```
Mover os binarios para o local correto
```
cd etcd-${ETCD_RELEASE}-linux-amd64 ; mv etcd* /usr/local/bin
```

Verificar se os 3 binarios foram copiados (etcd etcdctl etcdutl)

```
ls /usr/local/bin
```

Configurar o systemd

```
mkdir -p /var/lib/etcd/ mkdir /etc/etcd groupadd --system etcd
useradd -s /sbin/nologin --system -g etcd etcd
chown -R etcd:etcd /var/lib/etcd/
```

Criar o arquivo etcd.service com o conteudo abaixo

```
vim /etc/systemd/system/etcd.service
```

Conteudo :

```
[Unit]
Description=etcd
Documentation=https://github.com/etcd-io/etcd
After=network.target
[Service]
User=etcd
Type=notify
ExecStart=/usr/local/bin/etcd --config-file /etc/etcd/etcd.yml
StandartOutput=/var/log/etcd/etcd.log
StandartError=/var/log/etcd/etcd_error.log [Install]
WantedBy=multi-user.target
```


Criar pasta de logs

```
mkdir -p /var/log/etcd chown -R etcd:etcd /var/log/etcd
```

Criar arquivo de configuração
O name é o hostname do host que está sendo instalado

```
vim /etc/etcd/etcd.yml
```

Conteudo:

```
name: 'hostname'
data-dir: '/var/lib/etcd'
listen-peer-urls: 'http://10.30.50.37:2380'
listen-client-urls: 'http://10.30.50.37:2379,http://127.0.0.1:2379'
initial-advertise-peer-urls: 'http://10.30.50.37:2380'
advertise-client-urls: 'http://10.30.50.37:2379'
initial-cluster: 'hostname=http://10.30.50.37:2380,hostname-do-cluster-2=http:
initial-cluster-state: 'new'
initial-cluster-token: 'etcd-cluster-1'
```

Atenção com a pasta “data-dir” quando subir a primeira vez **essa pasta `/var/lib/etcd` deve estar vazia**, caso contrário as configurações que estiverem na pasta irão sobrepor as do arquivo de configuração !

ATENÇÃO ALTERAR HOSTNAME PELO HOSTNAME DO HOST E HOSTNAME-DO-CLUSTER-2

PELO HOSTNAME DO OUTRO SERVIDOR.

Alterar o IP de acordo com o host

Ativar e subir o serviço somente depois de configurar nos dois lados para que sejam ativados juntos

```
systemctl daemon-reload systemctl enable --now etcd.service
```

Verificar saude do cluster

```
etcdctl endpoint health
```

Verificar membros do cluster

```
etcdctl member list
```

O serviço roda na porta 2379 para verificar podemos usar o comando abaixo

```
ss -tunelp
```

Saida do comando


```
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.1:323 0.0.0.0:* users:(("chronyd",pid=1126,fd=6)) ino:1
udp UNCONN 0 0 [::1]:323 [::]:* users:(("chronyd",pid=1126,fd=7)) ino:16273 sk
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* users:(("sshd",pid=1147,fd=4)) ino:29601
tcp LISTEN 0 128 127.0.0.1:2379 0.0.0.0:* users:(("etcd",pid=19327,fd=9)) uid:

tcp LISTEN 0 128 10.30.50.37:2379 0.0.0.0:* users:(("etcd",pid=19327,fd=8)) ui

tcp LISTEN 0 128 10.30.50.37:2380 0.0.0.0:* users:(("etcd",pid=19327,fd=7)) ui

tcp LISTEN 0 128 [::]:22 [::]:* users:(("sshd",pid=1147,fd=6)) ino:29603 sk:4
```

FONTE: https://computingforgeeks.com/how-to-install-etcd-on-rhel-centos-rocky-almalinux/

Criar arquivo de config e pastas do Patroni em cada host

```
mkdir -p /etc/patroni
mkdir -p /var/patroni/data/
mkdir -p /var/log/patroni
chown -R postgres:postgres /etc/patroni
chown -R postgres:postgres /var/patroni/data
chown -R postgres:postgres /var/log/patroni
chmod -R 700 /etc/patroni
chmod -R 700 /var/patroni/data
chmod -R 700 /var/log/patroni
vim /etc/patroni/patroni.yml
```

Conteudo do arquivo:

```
scope: postgres
namespace: /db/
name: node1



restapi:
listen: 10.30.50.37:8008
connect_address: 10.30.50.37:8008



etcd3:
hosts: 10.30.50.37:2379,10.30.50.38:2379



bootstrap:
dcs:
ttl: 30
loop_wait: 10
retry_timeout: 10
maximum_lag_on_failover: 1048576
postgresql:
use_pg_rewind: true
use_slots: true
parameters:



initdb:
- encoding: UTF8
- data-checksums



pg_hba:
- host replication replicator 127.0.0.1/32 md5
- host replication replicator 10.30.50.37/0 md5
- host replication replicator 10.30.50.38/0 md5
- host all all 0.0.0.0/0 md5



users:
admin:
password: admin
options:
- createrole
- createdb



postgresql:
listen: 10.30.50.37:5432
```

```
connect_address: 10.30.50.37:5432
data_dir: /var/patroni/data/
bin_dir: /usr/pgsql-14/bin/
pgpass: /tmp/pgpass
authentication:
replication:
username: replicator
password: replicator
superuser:
username: postgres
password: postgres
parameters:
unix_socket_directories: '.'



tags:
nofailover: false
noloadbalance: false
clonefrom: false
nosync: false
```

Criar serviço Patroni

```
vim /etc/systemd/system/patroni.service
```


Conteudo do arquivo:

```
[Unit]
Description=High availability PostgreSQL Cluster
After=syslog.target network.target

[Service]
Type=simple
User=postgres
Group=postgres
ExecStart=patroni -c /etc/patroni/patroni.yml
KillMode=process
StandartOutput=/var/log/patroni/patroni.log
StandartError=/var/log/patroni/patroni_error.log
TimeoutSec=30
Restart=no
[Install]
WantedBy=multi-user.target
```

Iniciar e ativar o serviço

```
systemctl daemon-reload
systemctl enable --now patroni
```

Já deve ser possivel logar no postgres normalmente

Podemos verificar quem é o “lider” com o patroni

```
patronictl -c /etc/patroni/patroni.yml list
```

FONTE : https://community.pivotal.io/s/article/How-to-setup-a-3-node-Patroni-cluster-using-etcd?language=en_US

~~Instalar o HAProxy (num terceiro servidor !)~~

~~dnf install net-tools haproxy~~

~~sudo vi /etc/haproxy/haproxy.cfg~~

~~Replace its context with this:~~

```
global

        maxconn 100
        log     127.0.0.1 local2

defaults
        log global
        mode tcp
        retries 2
        timeout client 30m
        timeout connect 4s
        timeout server 30m
        timeout check 5s

listen stats
    mode http
    bind *:7000
    stats enable
    stats uri /

listen postgres
    bind *:5000
    option httpchk
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server node1 10.30.50.37:5432 maxconn 100 check port 8008

    server node2 10.30.50.38:5432 maxconn 100 check port 8008



sudo systemctl restart haproxy

sudo systemctl status haproxy

Para verificar qual host está ativo

http://<haproxynode_ip>:7000/>
```

~~FONTE :~~
~~https://jfrog.com/community/devops/highly-available-postgresql-cluster-using-p~~

Category

1. Banco

Date Created
fevereiro 2023
Author
09789446748