



***cassandra***

Banco de dados NoSQL - Cassandra (Operação)

Prof. Gustavo Leitão

Criando um cluster local

# Cluster Cassandra

Vamos criar um cluster local do cassandra com ajuda do Docker.

Criando uma rede:

```
$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
2a4e341c6039	bridge	bridge	local
4fbef5286425	host	host	local
8062e4e7cdca	none	null	local

```
$ docker network create cassandra-net
```

# Cluster Cassandra

Inicie o primeiro nó:

```
$ docker run --name cassandra-node1 --network cassandra-net -d  
cassandra
```

Inicie o segundo nó:

```
$ docker run --name cassandra-node2 -d --network cassandra-net -e  
CASSANDRA_SEEDS=cassandra-node1 cassandra
```

Perceba que o segundo nó inclui o endereço do primeiro. O SEED são máquinas já presentes no cluster para que o novo nó possa encontrar os demais nós

Inicie o terceiro nó:

```
$ docker run --name cassandra-node3 -d --network cassandra-net -e  
CASSANDRA_SEEDS=cassandra-node1 cassandra
```

# Cluster Cassandra

Inicie o primeiro nó:

```
$ docker run --name cassandra-node1 --network cassandra-net -d  
cassandra
```

Inicie o segundo nó:

```
$ docker run --name cassandra-node2 -d --network cassandra-net -e  
CASSANDRA_SEEDS=cassandra-node1 cassandra
```

Perceba que o segundo nó inclui o endereço do primeiro. O SEED são máquinas já presentes no cluster para que o novo nó possa encontrar os demais nós

Inicie o terceiro nó:

```
$ docker run --name cassandra-node3 -d --network cassandra-net -e  
CASSANDRA_SEEDS=cassandra-node1 cassandra
```

# Gerenciando Cluster

Acesse o terminal de alguns dos nós:

```
$ docker exec -it cassandra-node1 bash
$ nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address            Load           Tokens          Owns (effective)  Host ID
Rack
UN   172.19.0.2          274.89 KiB     256             100.0%
d839a61b-bd26-41a3-aad0-caa914089e49  rack1
```

# Cluster Cassandra

Acesse o terminal chql

```
$ docker run -it --network cassandra-net --rm cassandra cqlsh  
cassandra-node1
```

Crie KEYSPACE

```
CREATE KEYSPACE timeserie  
WITH REPLICATION = {  
  'class' : 'SimpleStrategy',  
  'replication_factor' : 2  
};
```

# Cluster Cassandra

## Crie Column Family

```
CREATE TABLE timeserie.sensors (  
    sensor text,  
    day text,  
    ts timestamp,  
    value double,  
    primary key((sensor, day), ts)  
) WITH CLUSTERING ORDER BY (ts DESC);
```

Insira dados:

```
$ INSERT INTO timeserie.sensors (sensor, day, ts, value) VALUES  
( 'mysensor', '2017-01-01', dateof(now())  
, 10);
```



# Gerenciando Cluster

Acesse o terminal de alguns dos nós:

```
$ docker exec -it cassandra-node1 bash

$ nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address          Load          Tokens         Owns (effective)  Host ID
Rack
UN  172.19.0.2      274.89 KiB    256            100.0%
d839a61b-bd26-41a3-aad0-caa914089e49  rack1
```

Criando um cluster real

# Cluster Cassandra

Descubra o IP da sua máquina

```
$ ifconfig
```

```
$ hostname -I
```

# Cluster Cassandra

## Inicie primeiro nó

```
$ docker run -d -e CASSANDRA_BROADCAST_ADDRESS=[IP_LOCAL] -p 7000:7000 -p 9042:9042 -p 9160:9160 cassandra:tag
```

## Inicie os nós seguintes:

```
$ docker run -d -e CASSANDRA_BROADCAST_ADDRESS=[IP_LOCAL] -p 7000:7000 -e CASSANDRA_SEEDS=[IP_NÓ_1] cassandra:tag
```

# Cluster Cassandra

## Inicie primeiro nó

```
$ docker run -d -e CASSANDRA_BROADCAST_ADDRESS=[IP_LOCAL] -p  
7000:7000 -p 9042:9042 -p 9160:9160 cassandra:tag
```

## Inicie os nós seguintes:

```
$ docker run -d -e CASSANDRA_BROADCAST_ADDRESS=[IP_LOCAL] -p  
7000:7000 -e CASSANDRA_SEEDS=[IP_NÓ_1] cassandra:tag
```

# Gerenciando Cluster

## Verifique o estado do cluster

```
$ docker exec -it cassandra-node1 bash
$ nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address            Load           Tokens         Owns (effective)  Host ID
Rack
UN   172.19.0.2          274.89 KiB     256            100.0%
d839a61b-bd26-41a3-aad0-caa914089e49  rack1
```

# Gerenciando Cluster

E se eu precisar remover um nó do cluster?

```
$ nodetool decommission
```

Esse comando irá iniciar o processo de desligamento de um nó do cluster. Suas replicas serão enviadas para outros nós antes do desligamento.

E se uma máquina (nó) caiu e perdi acesso?

```
$ nodetool status  
$ nodetool removenode {d0844a21-3698-4883-ab66-9e2fd5150edd}  
$ nodetool removenode status
```

Nesse caso, mesmo sem acesso é possível remover um nó do cluster. Se houve réplicas, o dado será redistribuído a partir das réplicas.

# Gerenciando Cluster

## Backup

```
$ nodetool snapshot
```

Faz snapshot do banco de todos keyspaces e todas tabelas.

```
$ nodetool snapshot -t 2015.07.17 cycling
```

Cria arquivo chamado 2015.07.17 com backup do keyspace cycling



# Exercício

Crie com mais dois colegas um cluster do Cassandra com várias máquinas para armazenar dados de série temporal. Siga os seguintes passos:

1. Crie um keyspace com fator de replicação 2
2. Insira 100 mil dados
3. Remova um dos nós e acompanhe o status

<https://github.com/gustavoleitao/cassandra-loader>



***cassandra***

Banco de dados NoSQL - Cassandra (Operação)

Prof. Gustavo Leitão