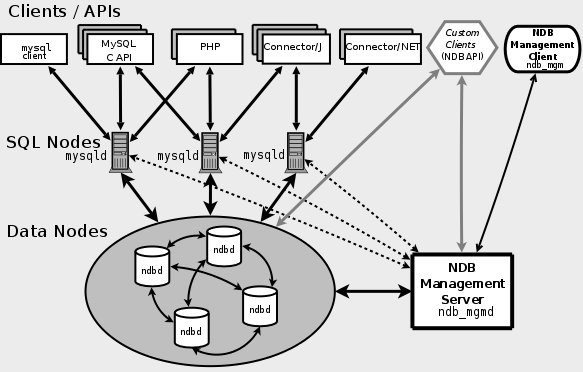
ami memilih melkukan proses replicasi pada mysql. Kami memilih paket yang sudah didukung oleh pengembang mysql sendiri, yaitu mysql-cluster dimana paket ini menggunakan engine NDB yang memang dikususkan untuk clustering dengan fault tolerance yang sangat tinggi. Kami menjalankan cluster tersebut diatas docker agar lebih hemat sumber daya. Cluster yang kami bangun memiliki topologi seperti berikut:



Proses - proses yang kami lakukan adalah sebagai berikut:

1. clone & modifikasi repo mysql-cluster untuk docker

modifikasi yang kami lakukan adalah mengkonfigurasi ulang IP yang digunakan oleh cluster dibeberapa file konfigurasi yaitu file my.cnf & file mysql-cluster.cnf link github ada di:

<https://github.com/mysql/mysql-docker/tree/mysql-cluster>

2. build image docker menggunakan perintah docker build -t wongko/mysqlcluster:latest . dimana wongko/mysqlcluster adalah repo hub.docker.com milik saya dan . menandakan

lokasi file docker file yang ingin saya build.

3. Setelah build selesai kami jalankan berintah seperti di bawah mengikuti panduan yang ada di laman https://hub.docker.com/r/mysql/mysql-cluster/ yang sudah kami modifikasi

speerti berikut :

# docker network create cluster --subnet=192.168.10.0/16

# docker run -d --net=cluster --name=management1 --ip=192.168.10.2 \

wongko/mysqlcluster ndb\_mgmd <untuk management cluster-node>

# docker run -d --net=cluster --name=ndb1 --ip=192.168.10.3 \

wongko/mysqlcluster ndbd <untuk datanode ke-1>

# docker run -d --net=cluster --name=ndb2 --ip=192.168.10.4 \

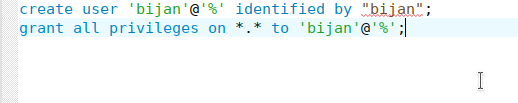
wongko/mysqlcluster ndbd <untuk datanode ke-2>

# docker run -d --net=cluster --name=mysql1 --ip=192.168.10.10 \

-e MYSQL\_ROOT\_PASSWORD=bijan wongko/mysqlcluster mysqld \

<untuk mysql-client>

4. Kemudian kami membuat user baru beserta privileges yang akan kami gunakan di aplikasi mysql-client seperti mysql workbench seperti berikut :



5. Kemudian kami buat database baru dan beberapa data untuk pengecekan Kemudian kami masuk ke dalam salah satu datanode dan menemukan data-data yang sudah di fragmentasi

bash-4.2# ls

S0.FragLog S10.FragLog S12.FragLog S14.FragLog

S2.FragLog S4.FragLog S6.FragLog S8.FragLog

S1.FragLog S11.FragLog S13.FragLog S15.FragLog

S3.FragLog S5.FragLog S7.FragLog S9.FragLog

bash-4.2# pwd

/var/lib/mysql/ndb\_2\_fs/D10/DBLQH

bash-4.2#>

6. Selesai

[Script – script lain yang kami jalankan]

root@budosen-pc:/home/budosen# docker run -it --net=cluster \

mysql/mysql-cluster ndb\_mgm

[Entrypoint] MySQL Docker Image 7.5.11-1.1.7-cluster

[Entrypoint] Starting ndb\_mgm

-- NDB Cluster -- Management Client --

ndb\_mgm> show

Connected to Management Server at: 192.168.10.2:1186

Cluster Configuration

---------------------

[ndbd(NDB)] 2 node(s)

id=2 @192.168.10.3 (mysql-5.7.23 ndb-7.5.11, Nodegroup: 0, \*)

id=3 @192.168.10.4 (mysql-5.7.23 ndb-7.5.11, Nodegroup: 0)

[ndb\_mgmd(MGM)] 1 node(s)

id=1 @192.168.10.2 (mysql-5.7.23 ndb-7.5.11)

[mysqld(API)] 1 node(s)

id=4 @192.168.10.10 (mysql-5.7.23 ndb-7.5.11)

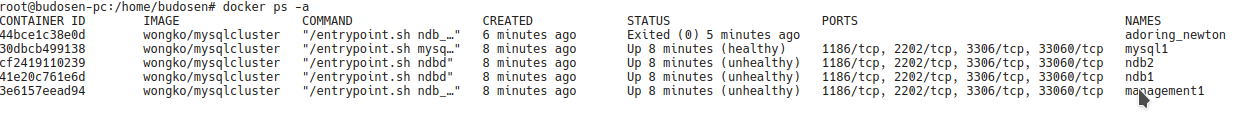
ndb\_mgm> 2 status

Node 2: started (mysql-5.7.23 ndb-7.5.11)

ndb\_mgm> 3 status

Node 3: started (mysql-5.7.23 ndb-7.5.11)

ndb\_mgm>



root@budosen-pc:/home/budosen# docker network inspect cluster

[

{

"Name": "cluster",

"Id": "bf1fe8a0b632a0f4e9e641759b631adf8231dab4475d0308c2f849f90ff33cd2",

"Created": "2018-11-14T19:28:04.96183234+07:00",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": {},

"Config": [

{

"Subnet": "192.168.10.0/16",

"Gateway": "192.168.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"30dbcb499138bb595f2d23428413c5e0abbf8722bcd08b07ed1262950a39b130": {

"Name": "mysql1",

"EndpointID": "4d3d0022d342189a45331b9d3f97168f1d5d296bd\

3e9e8720a4a5cdb7528504b",

"MacAddress": "02:42:c0:a8:0a:0a",

"IPv4Address": "192.168.10.10/16",

"IPv6Address": ""

},

"3e6157eead94d50678dcc6c9630ea2d3783cd5f6d58d2c08d7e7fe805f9998e3": {

"Name": "management1",

"EndpointID": "0b32f1b774663c2a3f291b42850ef9e1a243fd9e7\

2b16dac40e7ef4d57f7a1dc",

"MacAddress": "02:42:c0:a8:0a:02",

"IPv4Address": "192.168.10.2/16",

"IPv6Address": ""

},

"41e20c761e6de609826f7e28c9de1d180a483d6e1f1955c87f2c5034c277c664": {

"Name": "ndb1",

"EndpointID": "8439d9d92db2191ec91b6e7e4f74082c596a667be4c\

d377133ecd4e41ca8dff7",

"MacAddress": "02:42:c0:a8:0a:03",

"IPv4Address": "192.168.10.3/16",

"IPv6Address": ""

},

"cf24191102395ab23bbb602213daa814fbb69a33f2cce17f3c07b540257f5021": {

"Name": "ndb2",

"EndpointID": "2e1386085773bb648863a745c09750f0231ecf6861da\

725edec1e671ac339d04",

"MacAddress": "02:42:c0:a8:0a:04",

"IPv4Address": "192.168.10.4/16",

"IPv6Address": ""

}

},

"Options": {},

"Labels": {}

}

]