1.2.1.10 Pacemaker, Corosync dan haproxy

i. Install paket pacemaker dengan perintah

# apt-get install pacemaker haveged corosync pcs haproxy

ii. Generate corosync key dengan perintah

$ sudo corosync-keygen

iii. Distribusikan key corosync ke masing masing controller dengan perintah

$ sudo cat /etc/corosync/authkey | ssh controller0{2..3} sudo tee /etc/corosync/authkey

iv. Di setiap node controller, ubah permission file authkey nya dengan perintah

$ sudo chmod 400 /etc/corosync/authkey

$ sudo chown root /etc/corosync/authkey

Edit file /etc/corosync/corosync.conf, dengan memasukan bagian node:

nodelist {

node {

name: controller01

nodeid: 1

ring0\_addr: 172.16.3.35

}

node {

name: controller02

nodeid: 2

ring0\_addr: 172.16.3.58

}

node {

name: controller03

nodeid: 3

ring0\_addr: 172.16.3.61

}

}

Kemudian daftarkan service pacemaker ke dalam corosync pada baris paling bawah dengan mengisikan

service {

name: pacemaker

ver: 1

}

v. Distribusikan konfigurasi corosync dengan perintah

$ sudo cat /etc/corosync/corosync.conf | ssh controller0{2..3} sudo tee /etc/corosync/corosync.conf

vi. Restart service corosync dan pacemaker dengan perintah

$ systemctl restart corosync pacemaker

vii. Buat service corosync dan pacemaker agar start ketika server boot dengan perintah

$ systemctl enable corosync pacemaker

viii. Cek status cluster dengan perintah

$ sudo crm status

Apabila sudah tergabung status clusternya, maka pacemaker sudah selesai diclusterkan. Pastikan statusnya seperti pada di bawah

Node List:

\* Online: [ controller01 controller02 controller03]

Selanjutnya set pacemaker dengan perintah, di satu node

$ sudo crm configure property stonith-enabled=false

$ sudo crm configure property no-quorum-policy=ignore

ix. Buat ip float pada pacemaker beserta parameternya dengan perintah

$ sudo crm configure primitive VIP\_CLUSTER\_OPENSTACK ocf:heartbeat:IPaddr2 params ip="172.16.3.80" cidr\_netmask="24" op monitor interval="10s" meta migration-threshold="2" failure-timeout="60s" resource-stickiness="100"

$ sudo crm configure primitive VIP\_EXT\_OPENSTACK ocf:heartbeat:IPaddr2 params ip="10.0.0.80" cidr\_netmask="24" op monitor interval="10s" meta migration-threshold="2" failure-timeout="60s" resource-stickiness="100"

IP 172.16.3.30 dan 10.0.0.80 adalah IP Float yang digunakan untuk HA service di masing-masing node.

x. Cek kembali dengan perintah sudo crm status dan pastikan statusnya seperti pada di bawah ini

Node List:

\* Online: [ controller01 controller02 controller03 ]

Full List of Resources:

\* VIP\_CLUSTER\_OPENSTACK (ocf::heartbeat:IPaddr2): Started controller01

\* VIP\_EXT\_OPENSTACK (ocf::heartbeat:IPaddr2): Started controller02

xi. Re-check IP Float yang sebelumnya di set di /etc/hosts

10.0.0.80 cloud.graha-infracom.com

172.16.3.80 controller controller.os.internal

xii. Pastikan kedua IP float tersebut sudah bisa di PING dari masing-masing controller.

xiii. Selanjutnya konfig haproxy untuk melakukan loadbalance pada service yang sudah diset sebelumnya, yaitu rabbitmq dan database mysql. Edit /etc/haproxy/haproxy.cfg lalu pada bagian paling bawah, tambahkan baris di bawah ini.

#Mariadb HA

frontend mysql\_openstack

bind controller:3306

mode tcp

default\_backend be\_mysql\_openstack

backend be\_mysql\_openstack

balance source

mode tcp

option tcpka

option mysql-check user haproxy

server controller01 controller01:3306 check inter 2000 rise 2 fall 5

server controller02 controller02:3306 check inter 2000 rise 2 fall 5

server controller03 controller03:3306 check inter 2000 rise 2 fall 5

#Rabbit HA

frontend messaging\_openstack

bind controller:5672

default\_backend be\_messaging\_openstack

mode tcp

ba ckend be\_messaging\_openstack

balance leastconn

mode tcp

option tcp-check

server controller01 controller01:5672 check inter 2000 rise 2 fall 5

server controller02 controller02:5672 check inter 2000 rise 2 fall 5

server controller03 controller03:5672 check inter 2000 rise 2 fall 5

Ini membuat MARIADB dan RABBIT up dan di akses secara cluster di masing-masing node. Namun ada tambahan untuk healthcheck mariadb dimana haproxy akan melakukan cek menggunakan user haproxy ke database. Sehingga dibutuhkan user haproxy di dalam maridb.

Masuk ke mariadb

$ sudo mariadb

Di mariadb console, create user haproxy tanpa password dan tanpa grant

> CREATE USER 'haproxy'@'%';

> FLUSH PRIVILEGES;

xiv. Restart haproxy dengan perintah

$ sudo systemctl restart haproxy

xv. Distribusikan konfig haproxy ke controller yang lain dengan perintah

$ sudo cat /etc/haproxy/haproxy.cfg | ssh controller0{2..3} sudo tee /etc/haproxy/haproxy.cfg

xvi. Masukkan baris di bawah pada file /etc/sysctl.conf. Hal ini bertujuan agar Float IP dapat di gunakan di masing-masing node.

net.ipv4.ip\_nonlocal\_bind = 1

Setelah itu eksekusi perintah sysctl -p untuk mengaktifkan konfigurasi di atas.

xvii. Restart service haproxy dengan perintah

$ systemctl restart haproxy