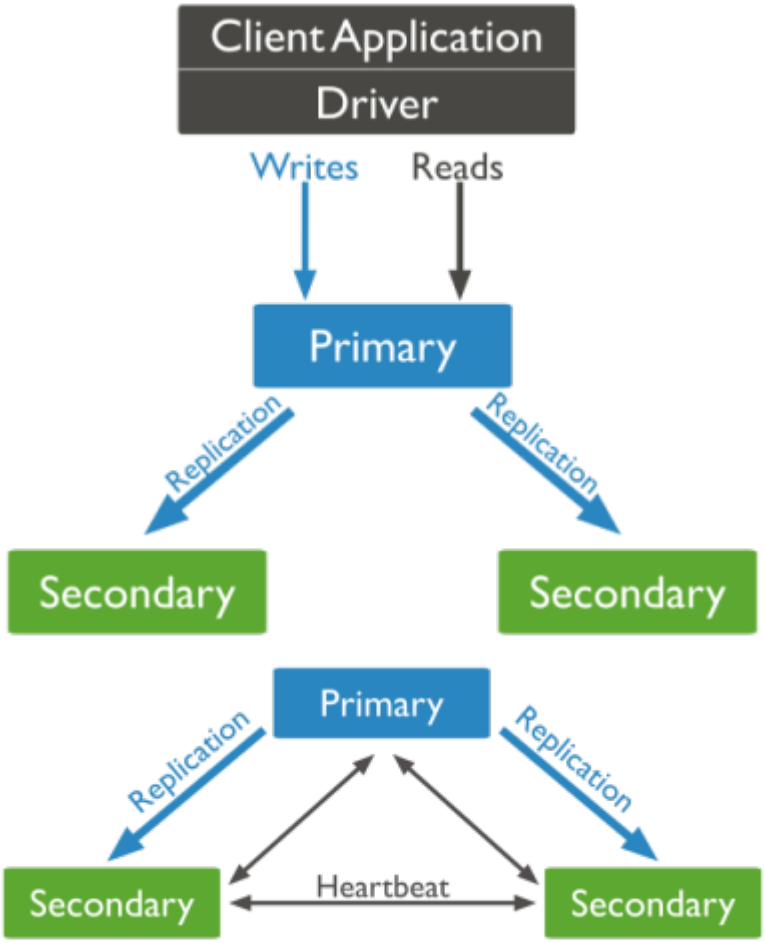
Installazione MongoDB e configurazione Cluster

Inseguito verrà descritta la procedura per installare MongoDB e realizzare un Cluster di replica come nel esempio seguente.



Ambiente e requisiti

- Sistema operativo Centos (testato in Centos 7)
- MongoDB 3.4

Installazione MongoDB

Pacchetti da installare

- mongodb-org
- mongodb-org-server
- mongodb-org-mongos
- mongodb-org-shell
- mongodb-org-tools

Configurare i pacchetti YUM

Crea repo file per installare direttamente da YUM

`/etc/yum.repos.d/mongodb-org-3.4.repo`

Il contenuto del file:

```
[mongodb-org-3.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc
```

Installazione Mongodb YUM

Eseguire il seguente commando:

```
sudo yum install -y mongodb-org
```

Disabilitare selinux

Modifica /etc/selinux/config

```
SELINUX=disabled
```

Riavvia la macchina!

Avvia MongoDB

```
service mongod start
```

Avvio nel starup

```
chkconfig mongod on
```

Configurazione Cluster

- Nel nostro caso servono tre macchine uguali. Una di loro farà soltanto il ruolo secondario e non diventerà mai primario. Il ruolo primario sarà deciso fra le altre due.

Nome macchine:

1. mongodb01 (primario)
2. mongodb02 (secondario)
3. mongodb03 (secondario)

- In tutte e tre le macchine configurare il nome della replica, uguale per tutte.

Modifica /etc/mongo.conf

```
replication:
     replSetName: db3d
```

- Nel server mongodb01 entra nel mongo:

```
mongo –host mongodb01
```

Eseguire:

```
db.createUser(
  {
    user: "admin",
    pwd: "3dinformatica",
    roles: [ { role: "root", db: "admin" } ]
  }
);
```

- Creare chiave per la comunicazione tra i server.

Nel server primario mongodb01 eseguire nel terminale:

```
[root@mongodb01 ~]# mkdir /var/lib/mongo/key
[root@mongodb01 ~]# openssl rand -base64 756  >   /var/lib/mongo/key/mongo_key
[root@mongodb01 ~]# chmod 400 /var/lib/mongo/key/mongo_key
[root@mongodb01 ~]# chown mongod:mongod /var/lib/mongo/key/mongo_key
```

Copiare la chiave anche nelle altre macchine nello stesso percorso /var/lib/mongo/key .

- Abilitiamo la security in tutte le macchine modificando /etc/mongo.conf

```
security:
  keyFile: /var/lib/mongo/key/mongo_key
```

```
  * Entriamo nel server mongodb01 e accediamo mogno con utente admin creato precedentemente.
```

```
mongo –host mongodb01 -u admin -p 3dinformatica –authenticationDatabase admin
```

- Eseguire in ordine i seguenti comandi. Priority 0 per il server che sarà esclusivamente secondario:

```
rs.initiate()
rs.add(“mongodb02:27017”)
```

```
rs.add( { host: "mongodb03:27017", priority: 0 } )
```

- Per dare priorità al nostro server primario mognodb01, lo definiamo con una priorità maggiore di tutte e tre. Di default hanno il valore 1:

`rs.conf()` visualizza la configurazione attuale della replica

```
cfg = rs.conf() ⇒ copia la configurazione della replica a cfg
cfg.members[0].priority = 3 ⇒ cambia il valore priority a 3 del server mongodb01 che ha id_0
rs.reconfig(cfg) ⇒ applica le modifiche
```

- Collegarsi al server mongodb01 con admin per vedere lo stato della replica con comando `rs.status()`

```
mongo –host mongodb01 -u admin -p 3dinformatica –authenticationDatabase admin
```