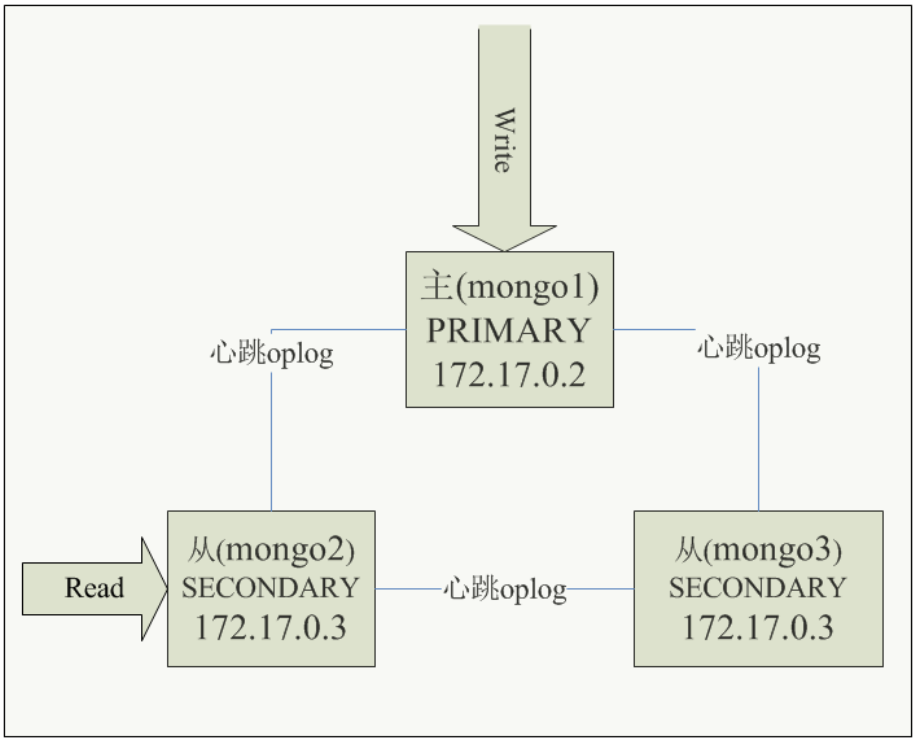
MongoDB 复制集机制及其应用

主要意义：服务高可用(24*7)　使用 Docker 部署 MongoDB 复制集



(1) 创建容器

docker run --name mongo1 -p 27017:27017 -d mongo --replSet "rs0" --bind_ip_all

docker run --name mongo2 -p 27018:27017 -d mongo --replSet "rs0" --bind_ip_all

docker run --name mongo3 -p 27019:27017 -d mongo --replSet "rs0" --bind_ip_all

```
[root@VM_0_13_centos ~]# docker run --name mongo1 -p 27017:27017 -d mongo  --replSet "rs0" --bind_ip_all
c1b886a239a475900ece306f9abb217ad0e6f9bb2b9e79239b47d7df6b72d825
[root@VM_0_13_centos ~]# docker run --name mongo2 -p 27018:27017 -d mongo  --replSet "rs0" --bind_ip_all
34f7fcce1483bec38d026ad20931cd96ac1e799948a7cf778c3e14897e4f22bb
[root@VM_0_13_centos ~]# docker run --name mongo3 -p 27019:27017 -d mongo  --replSet "rs0" --bind_ip_all
043f4cc304208715 6e3829e74b4ad1597336b9f56f1be6522f5d993bedad30ad
[root@VM_0_13_centos ~]# docker ps
CONTAINER ID        IMAGE                          COMMAND                  CREATED             STATUS              PORTS                     NAMES
043f4cc30420        mongo                          "docker-entrypoint.s…"   4 seconds ago       Up 4 seconds        0.0.0.0:27019->27017/tcp  mongo3
34f7fcce1483        mongo                          "docker-entrypoint.s…"   14 seconds ago      Up 13 seconds       0.0.0.0:27018->27017/tcp  mongo2
c1b886a239a4        mongo                          "docker-entrypoint.s…"   26 seconds ago      Up 25 seconds       0.0.0.0:27017->27017/tcp  mongo1
1fab20f8aad8        9228ee8bac7a                   "docker-entrypoint.s…"   2 months ago        Up 8 weeks          3306/tcp, 33060/tcp       home_mysql_1
1ecf895d7a91        edp963/davinci:v0.3.0-beta.9   "./bin/docker-entryp…"   2 months ago        Up 2 months         0.0.0.0:58080->8080/tcp   home_davinci_1
[root@VM_0_13_centos ~]#
```

(2) 查看容器 IP

docker inspect mongo1 | grep -i ipaddress

docker inspect mongo2 | grep -i ipaddress

docker inspect mongo3 | grep -i ipaddress

```
[root@VM_0_13_centos ~]# docker inspect mongo1 | grep -i ipaddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.2",
                    "IPAddress": "172.17.0.2",
[root@VM_0_13_centos ~]# docker inspect mongo2 | grep -i ipaddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.3",
                    "IPAddress": "172.17.0.3",
[root@VM_0_13_centos ~]# docker inspect mongo3 | grep -i ipaddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.4",
                    "IPAddress": "172.17.0.4",
```

(3)查看容器端口

docker port mongo1

docker port mongo2

docker port mongo3

```
[root@VM_0_13_centos ~]# docker port mongo1
27017/tcp -> 0.0.0.0:27017
[root@VM_0_13_centos ~]# docker port mongo2
27017/tcp -> 0.0.0.0:27018
[root@VM_0_13_centos ~]# docker port mongo3
27017/tcp -> 0.0.0.0:27019
[root@VM_0_13_centos ~]#
```

(3) 初始化复制集

```
rs.initiate({
    _id: 'rs0',
    members: [{
        _id: 0,
        host: '172.17.0.2:27017'
    },
    {
        _id: 1,
        host: '172.17.0.3:27017'
    },
    {
        _id: 2,
        host: '172.17.0.4:27017'
    }]
})
```

```
> rs.initiate({
...     _id: 'rs0',
...     members: [{
...         _id: 0,
...         host: '172.17.0.2:27017'
...     },
...     {
...         _id: 1,
...         host: '172.17.0.3:27017'
...     },
...     {
...         _id: 2,
...         host: '172.17.0.4:27017'
...     }]
... })
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1592903251, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1592903251, 1)
}
rs0:SECONDARY>
```

根据 replica-set-configuration-document ，

- `_id` ：复制集的名称。必须与启动 mongod 的 `--replSet` 一致
- `members` ：成员配置文件的列表。
  - `members[n]._id` ：用来识别成员的 id
  - `members[n].host` ：成员对应的 `host:port`

From：

https://docs.mongodb.com/manual/reference/replica-configuration/#replica-set-configuration-document

(4) 连接字符串为

mongo mongodb://172.17.0.2:27017,172.17.0.3:27017,172.17.0.4:27017/test?replicaSet=rs0

```
root@c1b886a239a4:/# mongo mongodb://172.17.0.2:27017,172.17.0.3:27017,172.17.0.4:27017/test?replicaSet=rs0
MongoDB shell version v4.2.7
connecting to: mongodb://172.17.0.2:27017,172.17.0.3:27017,172.17.0.4:27017/test?compressors=disabled&gssapiServiceName=mongodb&replicaSet=rs0
2020-06-23T09:10:13.620+0000 I  NETWORK  [js] Starting new replica set monitor for rs0/172.17.0.2:27017,172.17.0.3:27017,172.17.0.4:27017
2020-06-23T09:10:13.620+0000 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to 172.17.0.3:27017
2020-06-23T09:10:13.620+0000 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to 172.17.0.2:27017
2020-06-23T09:10:13.621+0000 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to 172.17.0.4:27017
2020-06-23T09:10:13.623+0000 I  NETWORK  [ReplicaSetMonitor-TaskExecutor] Confirmed replica set for rs0 is rs0/172.17.0.2:27017,172.17.0.3:27017,172.17.0.4:27017
Implicit session: session { "id" : UUID("fec22676-5b0d-4fbb-bce2-31ccec07da57") }
MongoDB server version: 4.2.7
Server has startup warnings:
2020-06-23T08:33:06.320+0000 I  STORAGE  [initandlisten]
2020-06-23T08:33:06.320+0000 I  STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-06-23T08:33:06.320+0000 I  STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-06-23T08:33:07.046+0000 I  CONTROL  [initandlisten]
2020-06-23T08:33:07.046+0000 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-06-23T08:33:07.046+0000 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-06-23T08:33:07.046+0000 I  CONTROL  [initandlisten]
2020-06-23T08:33:07.046+0000 I  CONTROL  [initandlisten]
2020-06-23T08:33:07.046+0000 I  CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-06-23T08:33:07.046+0000 I  CONTROL  [initandlisten] **        We suggest setting it to 'never'
2020-06-23T08:33:07.047+0000 I  CONTROL  [initandlisten]
2020-06-23T08:33:07.047+0000 I  CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2020-06-23T08:33:07.047+0000 I  CONTROL  [initandlisten] **        We suggest setting it to 'never'
2020-06-23T08:33:07.047+0000 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

rs0:PRIMARY>
```

(5) 复制特性

查看节点信息

```
rs0:SECONDARY> rs.status()
```

在 mongo1 DB 上面插入数据

db.order.insert({price: 1})

db.order.insert({price: 2})

db.order.insert({price: 3})

```
rs0:PRIMARY> db
test
rs0:PRIMARY>  db.order.insert({price: 1})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.order.insert({price: 2})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.order.insert({price: 3})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY>
```

在 mongo2 DB 上查看同步的数据

A docker exec -it mongo2 /bin/bash

B mongo

C db.getMongo().setSlaveOk()

D db.order.find()

```
rs0:SECONDARY> db.getMongo().setSlaveOk()
rs0:SECONDARY> db.order.find()
{ "_id" : ObjectId("5ef1c77805b7714e3e1b1704"), "price" : 1 }
{ "_id" : ObjectId("5ef1c77c05b7714e3e1b1705"), "price" : 2 }
{ "_id" : ObjectId("5ef1c77e05b7714e3e1b1706"), "price" : 3 }
rs0:SECONDARY> db.order.find()
{ "_id" : ObjectId("5ef1c77805b7714e3e1b1704"), "price" : 1 }
{ "_id" : ObjectId("5ef1c77c05b7714e3e1b1705"), "price" : 2 }
{ "_id" : ObjectId("5ef1c77e05b7714e3e1b1706"), "price" : 3 }
{ "_id" : ObjectId("5ef1c7fe05b7714e3e1b1707"), "price" : 4 }
rs0:SECONDARY>
```

(6) 在主节点 mongo1 创建用户

db.createUser({user:'root',pwd:'root',roles:[{role:'root',db:'admin'}]})

在 admin 数据库下面创建集合

db.order.insert({price: 100})

db.order.insert({price: 101})

db.order.insert({price: 102})

db.order.insert({price: 103})

```
rs0:PRIMARY> use admin
switched to db admin
rs0:PRIMARY> db.createUser({user:'root',pwd:'root',roles:[{role:'root',db:'admin'}]})
Successfully added user: {
        "user" : "root",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ]
}
rs0:PRIMARY> db.order.insert({price: 100})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.order.insert({price: 101})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.order.insert({price: 102})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.order.insert({price: 103})
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY>
```

(7) 在 mongo3 DB 上查看同步的数据

```
rs0:SECONDARY> db.getMongo().setSlaveOk()
rs0:SECONDARY> use admin
switched to db admin
rs0:SECONDARY> db.order.find().pretty
function() {
    this._prettyShell = true;
    return this;
}
rs0:SECONDARY> db.order.find().pretty()
{ "_id" : ObjectId("5ef1ca9905b7714e3e1b1708"), "price" : 100 }
{ "_id" : ObjectId("5ef1ca9b05b7714e3e1b1709"), "price" : 101 }
{ "_id" : ObjectId("5ef1ca9d05b7714e3e1b170a"), "price" : 102 }
{ "_id" : ObjectId("5ef1ca9f05b7714e3e1b170b"), "price" : 103 }
rs0:SECONDARY>
```

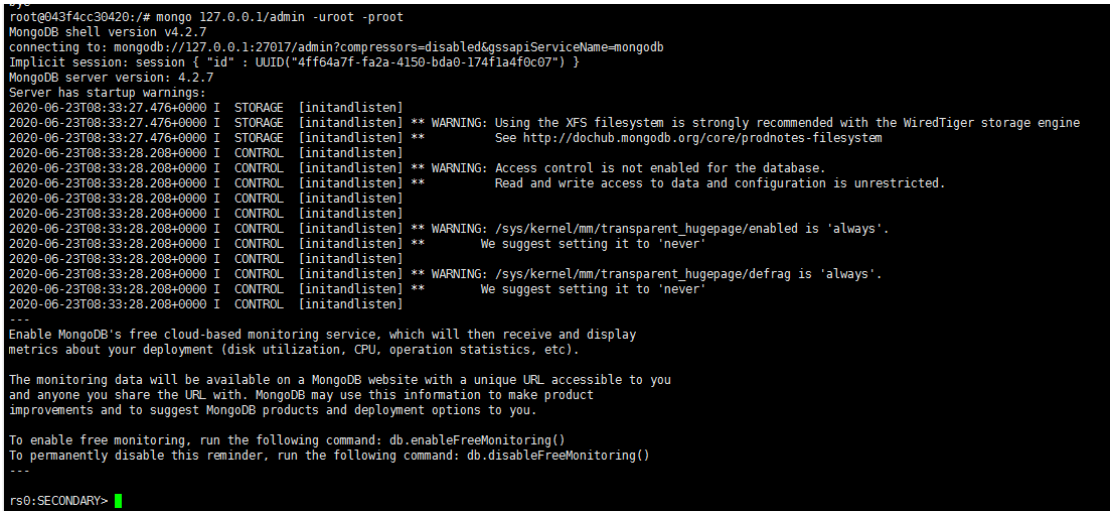
db.getMongo().setSlaveOk()

use admin

db.order.find().pretty()

同时主节点 mongo1DB 的用户的信息也已经同步到 mongo3DB

mongo 127.0.0.1/admin -uroot –proot



本文参考：

https://www.cnblogs.com/jay54520/p/8433515.html