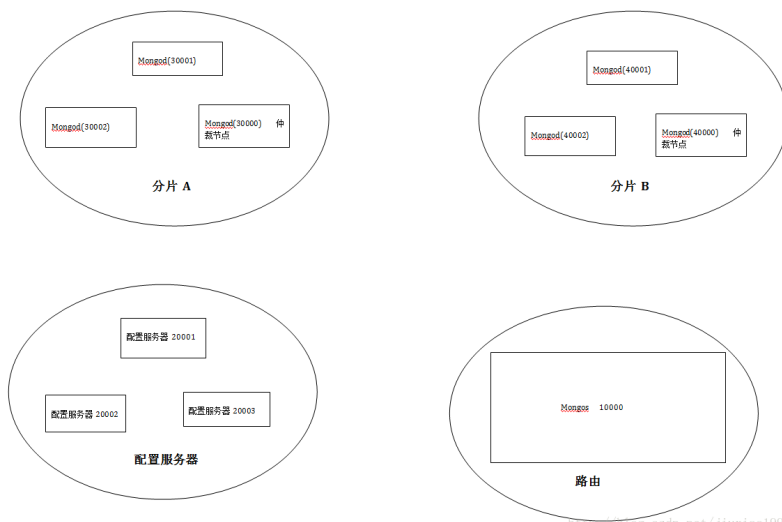


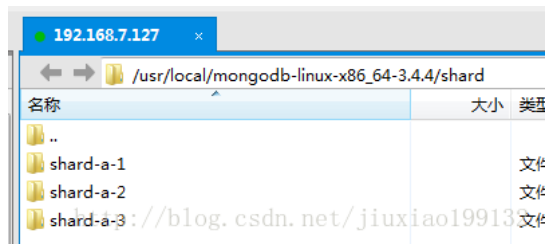
## 结构图



<http://blog.csdn.net/jiuxiao199132>

## 配置分片A

1 新建如下目录



2 分别启动分片A的三个节点

```
1 [root@localhost bin]# ./mongod --port 30001 --shardsvr --replSet shard-a --dbpath /usr/local/mongod
2
3 [root@localhost bin]# ./mongod --port 30002 --shardsvr --replSet shard-a --dbpath /usr/local/mongod
4
5 [root@localhost bin]# ./mongod --port 30003 --shardsvr --replSet shard-a --dbpath /usr/local/mongod
```

3 配置分片A这个副本集

连接mongo 30001这个节点

```
1 [root@localhost bin]# ./mongo --port 30001
```

初始化副本集

```
1 rs.initiate(
2   {
3     "_id": "shard-a",
4     "members": [
5       {
6         "_id": 0,
7         "host": "192.168.7.127:30001"
```

```

8
9         }
10    ]
11    }
12 );

```

添加从节点

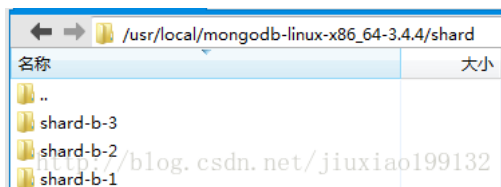
```
1 rs.add("192.168.7.127:30002")
```

添加仲裁节点

```
1 rs.add("192.168.7.127:30003",{arbiterOnly:true})
```

## 配置分片B

1 新建如下目录



2 分别启动分片B的三个节点

```

1 [root@localhost bin]# ./mongod --port 40001 --shardsvr --replSet shard-b --dbpath /usr/local/mongodl
2
3 [root@localhost bin]# ./mongod --port 40002 --shardsvr --replSet shard-b --dbpath /usr/local/mongodl
4
5 [root@localhost bin]# ./mongod --port 40003 --shardsvr --replSet shard-b --dbpath /usr/local/mongodl

```

3 配置分片A这个副本集

连接mongo 40001这个节点

```
1 [root@localhost bin]# ./mongo --port 40001
```

初始化副本集

```

1 rs.initiate(
2     {
3         "_id":"shard-b",
4         "members":[
5             {
6                 "_id":0,
7                 "host":"192.168.7.127:40001"
8             }
9         ]
10    }
11 )
12 );

```

添加从节点

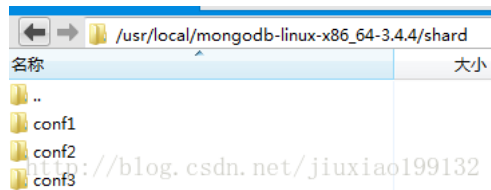
```
1 rs.add("192.168.7.127:40002")
```

添加仲裁节点

```
1 rs.add("192.168.7.127:40003",{arbiterOnly:true})
```

## 启动配置服务器

1 先建立如下文件夹



2 分别启动配置服务器

```
1 [root@localhost bin]# ./mongod --configsvr --port 20001 --replSet cfgReplSet --dbpath /usr/local/mo
2
3 [root@localhost bin]# ./mongod --configsvr --port 20002 --replSet cfgReplSet --dbpath /usr/local/mo
4
5 [root@localhost bin]# ./mongod --configsvr --port 20003 --replSet cfgReplSet --dbpath /usr/local/mo
```

3 配置 配置服务器这个副本集

连接mongo 20001这个节点

```
1 [root@localhost bin]# ./mongo --port 20001
```

初始化副本集

```
1 rs.initiate(
2   {
3     "_id":"cfgReplSet",
4     "members":[
5       {
6         "_id":0,
7         "host":"192.168.7.127:20001"
8       }
9     ]
10  }
11 );
12 );
```

添加从节点

```
1 rs.add("192.168.7.127:20002")
```

添加从节点 (注意这里没有仲裁节点)

```
1 rs.add("192.168.7.127:20003")
```

## 启动路由

1 先建立如下文件夹

```
1 /usr/local/mongodb-linux-x86_64-3.4.4/shard/route
```

2 启动路由

```
1 [root@localhost bin]# ./mongos --port 10000 --configdb cfgReplSet/192.168.7.127:20001,192.168.7.127
```

## 配置集群

1 连接路由节点

```
1 [root@localhost bin]# ./mongo --port 10000
```

2 添加分片(这里没有添加仲裁节点)

添加分片A

```
1 mongos> sh.addShard("shard-a/192.168.7.127:30001,192.168.7.127:30002")
```

添加分片B

```
1 mongos> sh.addShard("shard-b/192.168.7.127:40001,192.168.7.127:40002")
```

3 查看数据库

输入命令

```
1 mongos> show dbs
```

输出结果

```
1 admin 0.000GB
2 config 0.000GB
```

会发现有二个数据库 查看一下config数据库中的表

```
1 mongos> use config
2 mongos> show collections
```

输出结果

```
1 actionlog
2 changelog
3 chunks
4 databases
5 lockpings
6 locks
7 mongos
8 settings
9 shards
10 system.indexes
11 version
```

可以发现config数据库就存放了分片的信息

## 测试分片

目标：通过Java 驱动的方式写入到 数据库 cloud-users的表 user一千条数据

1 开启 cloud-uses 数据库分片，不管这个数据库是否存在

```
1 mongos> sh.enableSharding("cloud-users")
```

2 开启表 user的 分片，不管表是否存在直接执行命令即可， 分片键 由 age 和 id组合而成

```
1 sh.shardCollection("cloud-users.user",{ "age":1,"_id":1})
```

3 Java驱动执行代码

```
1 coll = new MongoClient( "192.168.7.127", 10000).getDatabase("cloud-users").getCollection("user");
2 for(int i = 0 ; i < 1000 ; i ++){
3     Document doc = new Document();
4     User u = User.initUser();
5     doc.put("address", u.getAddress());
6     doc.put("age", u.getAge());
7     doc.put("email", u.getEmail());
8     doc.put("height", u.getHeight());
9     doc.put("job", u.getJob());
10    doc.put("nickname", u.getNickname());
11    doc.put("phone", u.getPhone());
12    doc.put("school", u.getSchool());
13    doc.put("sex", u.getSex());
14    doc.put("hoby", u.getHoby());
15    Document dog = new Document();
16    dog.put("name", u.getDog().getName());
17    dog.put("age", u.getDog().getAge());
18    doc.put("dog", dog);
19    coll.insertOne(doc);
20 }
```