

一、环境

1. linux版本: CentOS release 6.8 (Final)

2. mongodb版本: mongodb-linux-x86_64-4.1.3.tgz

3. 架构

主机IP: 192.168.238.153

路由节点端口: 27017

配置节点集群: 17011, 17013, 17015

分片节点1: 37011, 37013, 37015, 37017

分片节点2: 47011, 47013, 47015, 47017

分片节点3: 57011, 57013, 57015, 57017

分片节点4: 58011, 58013, 58015, 58017

二、生成秘钥文件 并修改权限

```
openssl rand -base64 756 >
/usr/local/mongo/mongodb_shard_cluster/key/keyFile.file
chmod 600 key/keyFile.file
```

三、配置并启动config节点集群

1. 节点config-17011.conf

```
# 数据库文件位置
dbpath=config/config1
# 日志文件位置
logpath=config/logs/config1.log
# 以追加方式写入日志
logappend=true
# 是否以守护进程方式运行
fork = true
bind_ip=0.0.0.0
port = 17011
# 表示是一个配置服务器
configsvr=true
# 配置服务器副本集名称
replset=configsvr
auth=true
keyFile=key/keyFile.file
```

2. 节点config-17013.conf

```
# 数据库文件位置
dbpath=config/config2
# 日志文件位置
logpath=config/logs/config2.log
# 以追加方式写入日志
logappend=true
```

```
# 是否以守护进程方式运行
fork = true
bind_ip=0.0.0.0
port = 17013
# 表示是一个配置服务器
configsvr=true
#配置服务器副本集名称
replSet=configsvr
auth=true
keyFile=key/keyFile.file
```

3.节点config-17015.conf

```
# 数据库文件位置
dbpath=config/config3
#日志文件位置
logpath=config/logs/config3.log
# 以追加方式写入日志
logappend=true
# 是否以守护进程方式运行
fork = true
bind_ip=0.0.0.0
port = 17015
# 表示是一个配置服务器
configsvr=true
#配置服务器副本集名称
replSet=configsvr
auth=true
keyFile=key/keyFile.file
```

4.启动配置节点

```
./bin/mongod -f config/config-17011.conf
./bin/mongod -f config/config-17013.conf
./bin/mongod -f config/config-17015.conf
```

5.进入节点并配置集群

```
./bin/mongo --port 17011
use admin
var cfg = {"_id":"configsvr",
  "members":[
    {"_id":1,"host":"192.168.238.153:17011"},
    {"_id":2,"host":"192.168.238.153:17013"},
    {"_id":3,"host":"192.168.238.153:17015"}
  ]
};
rs.initiate(cfg)
```

```

"members" : [
  {
    "_id" : 1,
    "name" : "192.168.238.153:17011",
    "health" : 1,
    "state" : 1,
    "stateStr" : "PRIMARY",
    "uptime" : 66,
    "optime" : {
      "ts" : Timestamp(1593879106, 3),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2020-07-04T16:11:46Z"),
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "infoMessage" : "could not find member to sync from",
    "electionTime" : Timestamp(1593879103, 1),
    "electionDate" : ISODate("2020-07-04T16:11:43Z"),
    "configVersion" : 1,
    "self" : true,
    "lastHeartbeatMessage" : ""
  },
  {
    "_id" : 2,
    "name" : "192.168.238.153:17013",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 17,
    "optime" : {
      "ts" : Timestamp(1593879106, 3),
      "t" : NumberLong(1)
    },
  },

```

四、配置shard集群

1.shard1(37011,37013,37015,37017)

```

--shard1-37011.conf
dbpath=shard/shard1/shard1-37011
bind_ip=0.0.0.0
port=37011
fork=true
logpath=shard/shard1/logs/shard1-37011.log
replSet=shard1
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard1-37013.conf
dbpath=shard/shard1/shard1-37013
bind_ip=0.0.0.0
port=37013
fork=true
logpath=shard/shard1/logs/shard1-37013.log
replSet=shard1
shardsvr=true
auth=true
keyFile=key/keyFile.file

```

```

--shard1-37015.conf
dbpath=shard/shard1/shard1-37015
bind_ip=0.0.0.0
port=37015
fork=true
logpath=shard/shard1/logs/shard1-37015.log
replSet=shard1
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard1-37017.conf
dbpath=shard/shard1/shard1-37017
bind_ip=0.0.0.0
port=37017
fork=true
logpath=shard/shard1/logs/shard1-37017.log
replSet=shard1
shardsvr=true
auth=true
keyFile=key/keyFile.file

--启动节点
./bin/mongod -f shard/shard1/shard1-37011.conf
./bin/mongod -f shard/shard1/shard1-37013.conf
./bin/mongod -f shard/shard1/shard1-37015.conf
./bin/mongod -f shard/shard1/shard1-37017.conf

--配置节点集群
./bin/mongo --port 37011
var cfg = {"_id":"shard1",
  "protocolVersion" : 1,
  "members":[
    {"_id":1,"host":"192.168.238.153:37011"},
    {"_id":2,"host":"192.168.238.153:37013"},
    {"_id":3,"host":"192.168.238.153:37015"},
    {"_id":4,"host":"192.168.238.153:37017","arbiterOnly":true}
  ]
};
rs.initiate(cfg)

```

```

    },
    "optimeDate" : ISODate("2020-07-04T16:34:30Z"),
    "optimeDurableDate" : ISODate("2020-07-04T16:34:30Z"),
    "lastHeartbeat" : ISODate("2020-07-04T16:34:37.070Z"),
    "lastHeartbeatRecv" : ISODate("2020-07-04T16:34:37.273Z"),
    "pingMs" : NumberLong(0),
    "lastHeartbeatMessage" : "",
    "syncingTo" : "192.168.238.153:37011",
    "syncSourceHost" : "192.168.238.153:37011",
    "syncSourceId" : 1,
    "infoMessage" : "",
    "configVersion" : 1
  },
  {
    "_id" : 4,
    "name" : "192.168.238.153:37017",
    "health" : 1,
    "state" : 7,
    "stateStr" : "ARBITER",
    "uptime" : 19,
    "lastHeartbeat" : ISODate("2020-07-04T16:34:37.070Z"),
    "lastHeartbeatRecv" : ISODate("2020-07-04T16:34:37.101Z"),
    "pingMs" : NumberLong(0),
    "lastHeartbeatMessage" : "",
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "infoMessage" : "",
    "configVersion" : 1
  }
],
"ok" : 1
}

```

ard1:PRIMARY> █

2.shard2(47011,47013,47015,47017)

```

--shard2-47011.conf
dbpath=shard/shard2/shard2-47011
bind_ip=0.0.0.0
port=47011
fork=true
logpath=shard/shard2/logs/shard2-47011.log
replSet=shard2
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard2-47013.conf
dbpath=shard/shard2/shard2-47013
bind_ip=0.0.0.0
port=47013
fork=true
logpath=shard/shard2/logs/shard2-47013.log
replSet=shard2
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard2-47015.conf
dbpath=shard/shard2/shard2-47015
bind_ip=0.0.0.0
port=47015

```

```
fork=true
logpath=shard/shard2/logs/shard2-47015.log
replSet=shard2
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard2-47017.conf
dbpath=shard/shard2/shard2-47017
bind_ip=0.0.0.0
port=47017
fork=true
logpath=shard/shard2/logs/shard2-47017.log
replSet=shard2
shardsvr=true
auth=true
keyFile=key/keyFile.file

--启动节点
./bin/mongod -f shard/shard2/shard2-47011.conf
./bin/mongod -f shard/shard2/shard2-47013.conf
./bin/mongod -f shard/shard2/shard2-47015.conf
./bin/mongod -f shard/shard2/shard2-47017.conf

--配置节点集群
./bin/mongo --port 47011
var cfg = {"_id":"shard2",
  "protocolVersion" : 1,
  "members":[
    {"_id":1,"host":"192.168.238.153:47011"},
    {"_id":2,"host":"192.168.238.153:47013"},
    {"_id":3,"host":"192.168.238.153:47015"},
    {"_id":4,"host":"192.168.238.153:47017","arbiterOnly":true}
  ]
};
rs.initiate(cfg)
```

```

        "optimeDurable" : {
            "ts" : Timestamp(1593881124, 1),
            "t" : NumberLong(-1)
        },
        "optimeDate" : ISODate("2020-07-04T16:45:24Z"),
        "optimeDurableDate" : ISODate("2020-07-04T16:45:24Z"),
        "lastHeartbeat" : ISODate("2020-07-04T16:45:34.467Z"),
        "lastHeartbeatRecv" : ISODate("2020-07-04T16:45:36.085Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 1
    },
    {
        "_id" : 4,
        "name" : "192.168.238.153:47017",
        "health" : 1,
        "state" : 7,
        "stateStr" : "ARBITER",
        "uptime" : 12,
        "lastHeartbeat" : ISODate("2020-07-04T16:45:34.467Z"),
        "lastHeartbeatRecv" : ISODate("2020-07-04T16:45:34.805Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 1
    }
],
    "ok" : 1
}
shard2:PRIMARY>

```

3.shard3(57011,57013,57015,57017)

```

--shard3-57011.conf
dbpath=shard/shard3/shard3-57011
bind_ip=0.0.0.0
port=57011
fork=true
logpath=shard/shard3/logs/shard3-57011.log
replSet=shard3
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard3-57013.conf
dbpath=shard/shard3/shard3-57013
bind_ip=0.0.0.0
port=57013
fork=true
logpath=shard/shard3/logs/shard3-57013.log
replSet=shard3
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard3-57015.conf
dbpath=shard/shard3/shard3-57015

```

```

bind_ip=0.0.0.0
port=57015
fork=true
logpath=shard/shard3/logs/shard3-57015.log
replSet=shard3
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard3-57017.conf
dbpath=shard/shard3/shard3-57017
bind_ip=0.0.0.0
port=57017
fork=true
logpath=shard/shard3/logs/shard3-57017.log
replSet=shard3
shardsvr=true
auth=true
keyFile=key/keyFile.file

--启动节点
./bin/mongod -f shard/shard3/shard3-57011.conf
./bin/mongod -f shard/shard3/shard3-57013.conf
./bin/mongod -f shard/shard3/shard3-57015.conf
./bin/mongod -f shard/shard3/shard3-57017.conf

--配置节点集群
./bin/mongo --port 57011
var cfg = {"_id":"shard3",
  "protocolVersion" : 1,
  "members":[
    {"_id":1,"host":"192.168.238.153:57011"},
    {"_id":2,"host":"192.168.238.153:57013"},
    {"_id":3,"host":"192.168.238.153:57015"},
    {"_id":4,"host":"192.168.238.153:57017","arbiterOnly":true}
  ]
};
rs.initiate(cfg)

```



```

    },
    "optimeDurable" : {
      "ts" : Timestamp(1593881173, 1),
      "t" : NumberLong(-1)
    },
    "optimeDate" : ISODate("2020-07-04T16:46:13Z"),
    "optimeDurableDate" : ISODate("2020-07-04T16:46:13Z"),
    "lastHeartbeat" : ISODate("2020-07-04T16:46:25.976Z"),
    "lastHeartbeatRecv" : ISODate("2020-07-04T16:46:25.857Z"),
    "pingMs" : NumberLong(0),
    "lastHeartbeatMessage" : "",
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "infoMessage" : "",
    "configVersion" : 1
  },
  {
    "_id" : 4,
    "name" : "192.168.238.153:57017",
    "health" : 1,
    "state" : 7,
    "stateStr" : "ARBITER",
    "uptime" : 12,
    "lastHeartbeat" : ISODate("2020-07-04T16:46:25.977Z"),
    "lastHeartbeatRecv" : ISODate("2020-07-04T16:46:24.158Z"),
    "pingMs" : NumberLong(0),
    "lastHeartbeatMessage" : "",
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "infoMessage" : "",
    "configVersion" : 1
  }
],
"ok" : 1
}
shard3:PRIMARY>

```

4.shard4(58011,58013,58015,58017)

```

--shard4-58011.conf
dbpath=shard/shard4/shard4-58011
bind_ip=0.0.0.0
port=58011
fork=true
logpath=shard/shard4/logs/shard4-58011.log
replSet=shard4
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard4-58013.conf
dbpath=shard/shard4/shard4-58013
bind_ip=0.0.0.0
port=58013
fork=true
logpath=shard/shard4/logs/shard4-58013.log
replSet=shard4
shardsvr=true
auth=true
keyFile=key/keyFile.file

--shard4-58015.conf

```

```
dbpath=shard/shard4/shard4-58015
bind_ip=0.0.0.0
port=58015
fork=true
logpath=shard/shard4/logs/shard4-58015.log
replSet=shard4
shardsvr=true
auth=true
keyFile=key/keyFile.file
```

```
--shard4-58017.conf
dbpath=shard/shard4/shard4-58017
bind_ip=0.0.0.0
port=58017
fork=true
logpath=shard/shard4/logs/shard4-58017.log
replSet=shard4
shardsvr=true
auth=true
keyFile=key/keyFile.file
```

```
--启动节点
./bin/mongod -f shard/shard4/shard4-58011.conf
./bin/mongod -f shard/shard4/shard4-58013.conf
./bin/mongod -f shard/shard4/shard4-58015.conf
./bin/mongod -f shard/shard4/shard4-58017.conf
```

```
--配置节点集群
./bin/mongo --port 58011
var cfg = {"_id":"shard4",
  "protocolVersion" : 1,
  "members":[
    {"_id":1,"host":"192.168.238.153:58011"},
    {"_id":2,"host":"192.168.238.153:58013"},
    {"_id":3,"host":"192.168.238.153:58015"},
    {"_id":4,"host":"192.168.238.153:58017","arbiterOnly":true}
  ]
};
rs.initiate(cfg)
```

```

        "optimeDurable" : {
            "ts" : Timestamp(1593881227, 2),
            "t" : NumberLong(1)
        },
        "optimeDate" : ISODate("2020-07-04T16:47:07Z"),
        "optimeDurableDate" : ISODate("2020-07-04T16:47:07Z"),
        "lastHeartbeat" : ISODate("2020-07-04T16:47:11.781Z"),
        "lastHeartbeatRecv" : ISODate("2020-07-04T16:47:12.924Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "192.168.238.153:58011",
        "syncSourceHost" : "192.168.238.153:58011",
        "syncSourceId" : 1,
        "infoMessage" : "",
        "configVersion" : 1
    },
    {
        "_id" : 4,
        "name" : "192.168.238.153:58017",
        "health" : 1,
        "state" : 7,
        "stateStr" : "ARBITER",
        "uptime" : 18,
        "lastHeartbeat" : ISODate("2020-07-04T16:47:11.781Z"),
        "lastHeartbeatRecv" : ISODate("2020-07-04T16:47:12.797Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 1
    }
],
"ok" : 1

```

ard4:PRIMARY> █

五、配置路由route节点

```

--route-27017.conf
port=27017
bind_ip=0.0.0.0
fork=true
logpath=route/logs/route.log
configdb=configsvr/192.168.238.153:17011,192.168.238.153:17013,192.168.238.153:17015
keyFile=key/keyFile.file

--启动节点
./bin/mongos -f route/route-27017.conf

./bin/mongo --port 27017

--创建root用户
use admin
db.createUser({
    user:"root",
    pwd:"abc321",
    roles:[{role:"root",db:"admin"}]}
)
db.auth("root","abc321")

```

```
show dbs
```

```
--mongos（路由）中添加分片节点
```

```
sh.status()
```

```
sh.addShard("shard1/192.168.238.153:37011,192.168.238.153:37013,192.168.238.153:37015,192.168.238.153:37017");
```

```
sh.addShard("shard2/192.168.238.153:47011,192.168.238.153:47013,192.168.238.153:47015,192.168.238.153:47017");
```

```
sh.addShard("shard3/192.168.238.153:57011,192.168.238.153:57013,192.168.238.153:57015,192.168.238.153:57017");
```

```
sh.addShard("shard4/192.168.238.153:58011,192.168.238.153:58013,192.168.238.153:58015,192.168.238.153:58017");
```

```
sh.status()
```

```
--为数据库开启分片功能
```

```
sh.enableSharding("lg_resume")
```

```
--为指定集合开启分片功能
```

```
sh.shardCollection("lg_resume.lagou_resume_datas",{"name":"hashed"})
```

```
--创建用户lagou_gx
```

```
use lg_resume
```

```
db.createUser({
  user:"lagou_gx",
  pwd:"abc321",
  roles:[{role:"readWrite",db:"lg_resume"}]
})
```

```
db.auth("lagou_gx","abc321")
```

```
show dbs
```

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("5f00aa41be0034a559048736")
  }
  shards:
    { "_id" : "shard1", "host" : "shard1/192.168.238.153:37011,192.168.238.153:37013,192.168.238.153:37015", "state" : 1 }
    { "_id" : "shard2", "host" : "shard2/192.168.238.153:47011,192.168.238.153:47013,192.168.238.153:47015", "state" : 1 }
    { "_id" : "shard3", "host" : "shard3/192.168.238.153:57011,192.168.238.153:57013,192.168.238.153:57015", "state" : 1 }
    { "_id" : "shard4", "host" : "shard4/192.168.238.153:58011,192.168.238.153:58013,192.168.238.153:58015", "state" : 1 }
  active mongoses:
    "4.1.3" : 1
  autosplit:
    Currently enabled: yes
  balancer:
    Currently enabled: yes
    Currently running: no
    Failed balancer rounds in last 5 attempts: 0
    Migration Results for the last 24 hours:
      No recent migrations
  databases:
    { "_id" : "config", "primary" : "config", "partitioned" : true }
      config.system.sessions
        shard key: { "_id" : 1 }
        unique: false
        balancing: true
        chunks:
          shard1 1
          { "_id" : { "$minKey" : 1 } } --> { "_id" : { "$maxKey" : 1 } } on : shard1 Timestamp(1, 0)
```

```

mongos> use admin
switched to db admin
mongos> db.createUser({
...     user:"lagou_gx",
...     pwd:"abc321",
...     roles:[{role:"readWrite",db:"lg_resume"}]
... })
Successfully added user: {
  "user" : "lagou_gx",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "lg_resume"
    }
  ]
}
mongos> db.auth("lagou_gx","abc321")
1
mongos> show dbs
lg_resume  0.000GB

```

六、springboot项目

1.实体类Resume

```

public class Resume {

    private String id;
    private String name;
    private String city;
    private Date birthday;
    private double expectsSalary;

}

```

2.repository ResumeDao及实现类ResumeDaoImpl

```

public interface ResumeDao {

    void insertResume(Resume resume);

    Resume findByName(String name);

    List<Resume> findListByName(String name);

    List<Resume> findListByNameAndExpectSalary(String name, double expectsSalary);
}

@Repository
public class ResumeDaoImpl implements ResumeDao {

    @Autowired
    private MongoTemplate mongoTemplate;
}

```

```

@Override
public void insertResume(Resume resume) {
//    mongoTemplate.insert(resume);
    mongoTemplate.insert(resume, "lg_resume_datas");
}

@Override
public Resume findByName(String name) {
    List<Resume> datas = findListByName(name);
    return datas.isEmpty() ? null : datas.get(0);
}

@Override
public List<Resume> findListByName(String name) {
    Query query = new Query();
    query.addCriteria(Criteria.where("name").is(name));
    return mongoTemplate.find(query, Resume.class, "lg_resume_datas");
}

@Override
public List<Resume> findListByNameAndExpectSalary(String name, double
expectSalary) {
    Query query = new Query();

    query.addCriteria(Criteria.where("name").is(name).andOperator(Criteria.where("ex
pectSalary").is(expectSalary)));
    return mongoTemplate.find(query, Resume.class, "lg_resume_datas");
}
}

```

3.测试类

```

@SpringBootApplication
public class MongoRootMain {

    public static void main(String[] args) {
        ApplicationContext applicationContext =
SpringApplication.run(MongoRootMain.class, args);
        ResumeDao resumeDao = applicationContext.getBean(ResumeDao.class);
        Resume resume = new Resume();

        resume.setName("lisi");
        resume.setCity("北京");
        Date date = null;
        try {
            date = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").parse("2000-01-01
12:12:13");
        } catch (ParseException e) {
            e.printStackTrace();
        }
        resume.setBirthday(date);
        resume.setExpectSalary(23000);
        resumeDao.insertResume(resume);
        System.out.println("resume---->" + resume);

        Resume lisi = resumeDao.findByName("lisi");
    }
}

```

```

        System.out.println("lisi:" + lisi);
    }
}

```

4.application.properties

```

spring.data.mongodb.host=192.168.238.153
spring.data.mongodb.port=27017
spring.data.mongodb.database=lg_resume
spring.data.mongodb.username=lagou_gx
spring.data.mongodb.password=abc321

```

5.测试

The screenshot shows an IDE with a project named 'MongoRootMain'. The code in the editor is as follows:

```

// resumeDao.resume() resume);
38 System.out.println("resume---->" + resume);
39
40 Resume lisi = resumeDao.findByName("lisi");
41 System.out.println("lisi:" + lisi);
42
43 }
44

```

The console output shows the following logs:

```

2020-07-06 07:54:24.344 INFO 28508 --- [main] cn.xmj.MongoRootMain : Starting MongoRootMain on THINKPAD-LH with PID 28508 (E:\git\lagou\mongodb\mongo-code\target
2020-07-06 07:54:24.346 INFO 28508 --- [main] cn.xmj.MongoRootMain : No active profile set, falling back to default profiles: default
2020-07-06 07:54:24.675 INFO 28508 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
2020-07-06 07:54:24.689 INFO 28508 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 11ms. Found 0 MongoDB repository interfaces.
2020-07-06 07:54:24.942 INFO 28508 --- [main] org.mongodb.driver.cluster : Cluster created with settings {hosts=[192.168.238.153:27017], mode=MULTIPLE, requiredCluster
2020-07-06 07:54:24.942 INFO 28508 --- [main] org.mongodb.driver.cluster : Adding discovered server 192.168.238.153:27017 to client view of cluster
2020-07-06 07:54:24.990 INFO 28508 --- [8.238.153:27017] org.mongodb.driver.connection : Opened connection [connectionId{localValue:1, serverValue:46}] to 192.168.238.153:27017
2020-07-06 07:54:24.996 INFO 28508 --- [8.238.153:27017] org.mongodb.driver.cluster : Monitor thread successfully connected to server with description ServerDescription(address=
2020-07-06 07:54:24.998 INFO 28508 --- [8.238.153:27017] org.mongodb.driver.cluster : Discovered cluster type of SHARDED
2020-07-06 07:54:25.225 INFO 28508 --- [main] cn.xmj.MongoRootMain : Started MongoRootMain in 1.22 seconds (JVM running for 3.351)
2020-07-06 07:54:25.462 INFO 28508 --- [main] org.mongodb.driver.connection : Opened connection [connectionId{localValue:2, serverValue:47}] to 192.168.238.153:27017
resume---->Resume{id='5f026831e5b72843b761b7b0', name='lisi', city='北京', birthday=Sat Jan 01 00:12:13 CST 2000, expectSalary=23000.0}
lisi:Resume{id='5f026831e5b72843b761b7b0', name='lisi', city='北京', birthday=Sat Jan 01 00:12:13 CST 2000, expectSalary=23000.0}
2020-07-06 07:54:25.847 INFO 28508 --- [extShutdownHook] org.mongodb.driver.connection : Closed connection [connectionId{localValue:2, serverValue:47}] to 192.168.238.153:27017 beca

```