一、安装环境

```
1.操作系统：CentOS Linux release 7.5.1804
2.软件版本：
    zookeeper:zookeeper-3.4.14
    solr:solr-7.7.3
    tomcat:apache-tomcat-8.5.58
```

host	solr	zookeeper	tomcat
xmj(192.168.238.170)	√		√
xmjmaster(192.168.238.160)	√	√	√
xmjslave1(192.168.238.161)	√	√	√
xmjslave2(192.168.238.162)	√	√	√

二、安装配置

1.zookeeper安装配置

```
1)解压缩
tar -zxf zookeeper-3.4.14.tar.gz -C /usr/local
```

```
2）配置文件
cd /usr/local/zookeeper-3.4.14/conf
cp zoo_sample.cfg zoo.cfg
vim zoo.cfg

#设置dataDir和dataLogDir
dataDir=/usr/local/zookeeper-3.4.14/data
dataLogDir=/usr/local/zookeeper-3.4.14/data/logs

#添加集群配置
server.1=192.168.238.160:2881:3881
server.2=192.168.238.161:2881:3881
server.3=192.168.238.162:2881:3881
```

```
3）配置myid
cd /usr/local/zookeeper-3.4.14/data
vim myid
(xmjmaster:1,xmjslave1:2,xmjslave2:3)
```

```
4)配置环境变量
vim /etc/profile
export ZOOKEEPER_PREFIX=/usr/local/zookeeper-3.4.14
export PATH=$PATH:$ZOOKEEPER_PREFIX/bin
source /etc/profile
```

```
5)启动zookeeper集群
zkServer.sh start
##查看集群状态
zkServer.sh status
```

2.solr tomcat安装配置

```
1）上传tomcat和solr压缩包到/usr/local/solr-cloud,并解压缩(xmj)

mkdir -p /usr/local/solr-cloud
cd /usr/local/solr-cloud
tar -zxvf apache-tomcat-8.5.58.tar.gz
mv apache-tomcat-8.
5.58 tomcat
tar -xvf solr-7.7.3.tgz
```

```
2)tomcat中建立solr工程(xmj)
cp /usr/local/solr-cloud/solr-7.7.3/server/solr-webapp/webapp -rf
/usr/local/solr-cloud/tomcat/webapps/solr
```

```
3)拷贝jar包(xmj)
cd /usr/local/solr-cloud/solr-7.7.3/server/lib
cp ext/* /usr/local/solr-cloud/tomcat/webapps/solr/WEB-INF/lib/
cp metrics-* /usr/local/solr-cloud/tomcat/webapps/solr/WEB-INF/lib/
```

```
4)配置solrhome(xmj)
mkdir -p /usr/local/solr-cloud/solrhome
cd /usr/local/solr-cloud/solr-7.7.3/server/solr
cp -r * /usr/local/solr-cloud/solrhome/
```

```
5)修改solr的web.xml文件 把solrhome关联起来、去掉安全认证(xmj)
vim /usr/local/solr-cloud/tomcat/webapps/solr/WEB-INF/web.xml

<!-- 修改solrhome路径 -->
<env-entry>
    <env-entry-name>solr/home</env-entry-name>
    <env-entry-value>/usr/local/solr-cloud/solrhome</env-entry-value>
    <env-entry-type>java.lang.String</env-entry-type>
</env-entry>

<!-- 去掉安全认证 -->
<!--
    <security-constraint>
    ...
    </security-constraint>
-->
```

```
6）启动tomcat测试
cd /usr/local/solr-cloud
./tomcat/bin/startup.sh
```

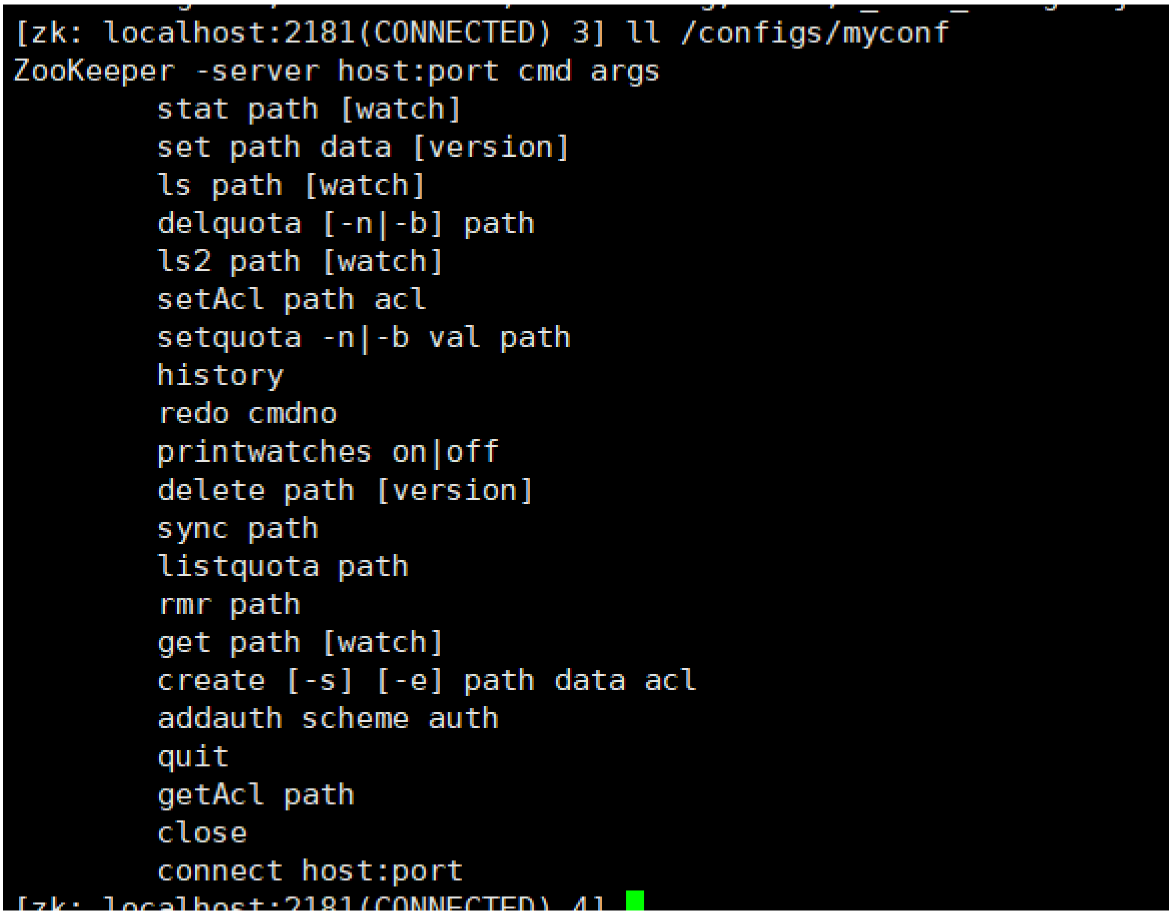
```
7）配置solrCloud相关的配置
vim /usr/local/solr-cloud/solrhome/solr.xml

<solrcloud>
    <str name="host">192.168.238.170</str>
    <int name="hostPort">8080</int>
    <str name="hostContext">${hostContext:solr}</str>
</solrcloud>
```

```
8)拷贝到xmjmaster、xmjslave1和xmjslave2， 并修改对应的solr.xml(参考7)
scp -r /usr/local/solr-cloud/ xmjmaster:/usr/local
scp -r /usr/local/solr-cloud/ xmjslave1:/usr/local
scp -r /usr/local/solr-cloud/ xmjslave2:/usr/local
```

```
9）让zookeeper统一管理配置文件
cd /usr/local/solr-cloud/solr-7.7.3/server/scripts/cloud-scripts/
##执行命令
./zkcli.sh -zkhost
192.168.238.160:2181,192.168.238.161:2181,192.168.238.162:2181 -cmd upconfig -
confdir /usr/local/solr-
cloud/solrhome/configsets/sample_techproducts_configs/conf -confname myconf
```

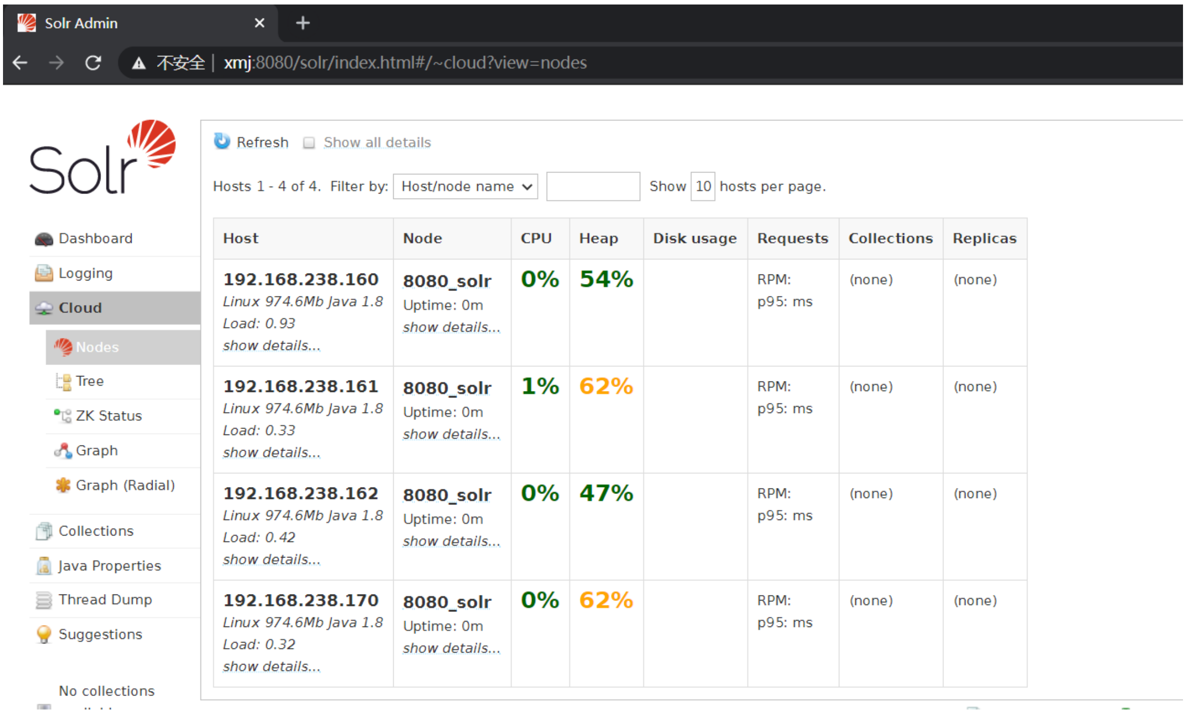
```
10）查看zookeeper上的配置文件
zkCli.sh
```



```
[zk: localhost:2181(CONNECTED) 3] ll /configs/myconf
ZooKeeper -server host:port cmd args
        stat path [watch]
        set path data [version]
        ls path [watch]
        delquota [-n|-b] path
        ls2 path [watch]
        setAcl path acl
        setquota -n|-b val path
        history
        redo cmdno
        printwatches on|off
        delete path [version]
        sync path
        listquota path
        rmr path
        get path [watch]
        create [-s] [-e] path data acl
        addauth scheme auth
        quit
        getAcl path
        close
        connect host:port
[zk: localhost:2181(CONNECTED) 4]
```

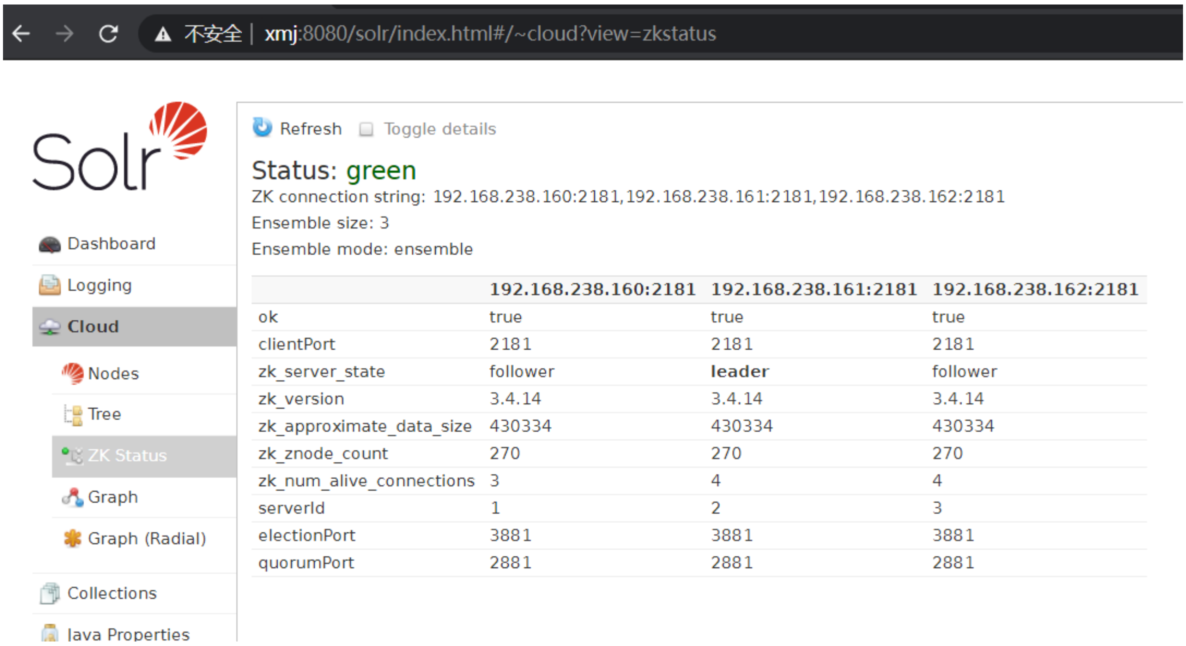
```
11）修改tomcat/bin目录下的catalina.sh 文件，关联solr和zookeeper
vim /usr/local/solr-cloud/tomcat/bin/catalina.sh
## 搜索 umask
JAVA_OPTS="-
DzkHost=192.168.238.160:2181,192.168.238.161:2181,192.168.238.162:2181"
```

```
12）启动所有tomcat,浏览器访问集群
./tomcat/bin/startup.sh
http://xmj:8080/solr/index.html
```



Hosts 1 - 4 of 4.

Host	Node	CPU	Heap	Disk usage	Requests	Collections	Replicas
192.168.238.160 *Linux 974.6Mb Java 1.8* *Load: 0.93* *show details...*	8080_solr Uptime: 0m *show details...*	0%	54%		RPM: p95: ms	(none)	(none)
192.168.238.161 *Linux 974.6Mb Java 1.8* *Load: 0.33* *show details...*	8080_solr Uptime: 0m *show details...*	1%	62%		RPM: p95: ms	(none)	(none)
192.168.238.162 *Linux 974.6Mb Java 1.8* *Load: 0.42* *show details...*	8080_solr Uptime: 0m *show details...*	0%	47%		RPM: p95: ms	(none)	(none)
192.168.238.170 *Linux 974.6Mb Java 1.8* *Load: 0.32* *show details...*	8080_solr Uptime: 0m *show details...*	0%	62%		RPM: p95: ms	(none)	(none)



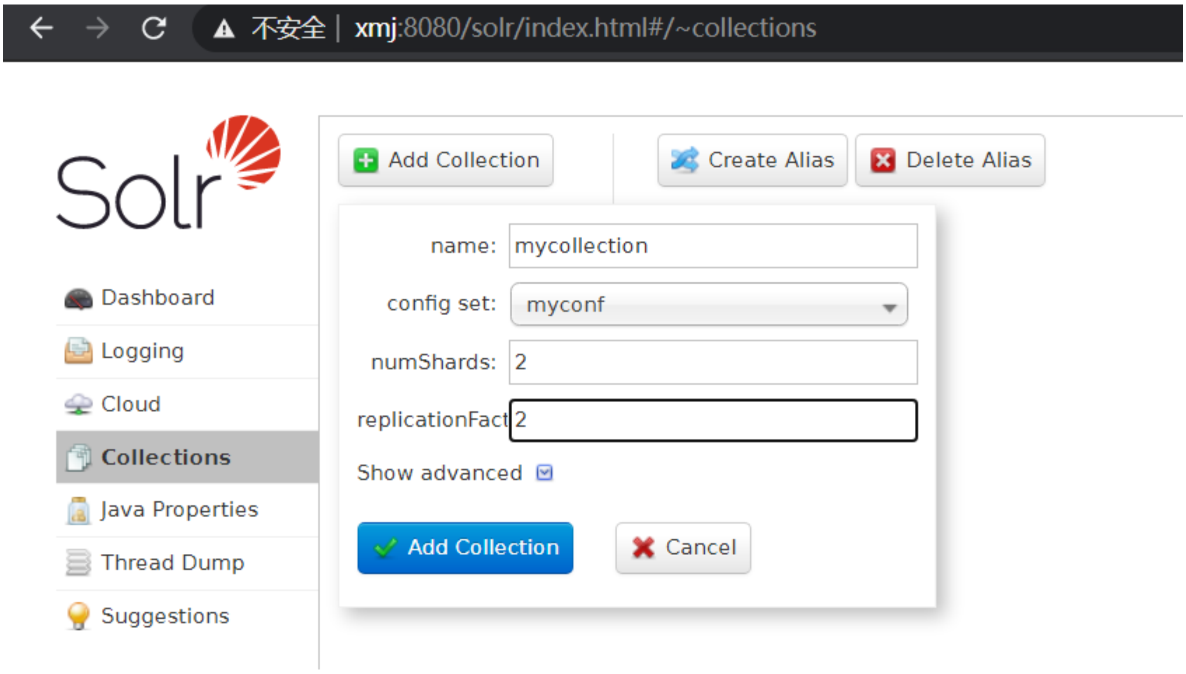
Status: green

ZK connection string: 192.168.238.160:2181,192.168.238.161:2181,192.168.238.162:2181

Ensemble size: 3

Ensemble mode: ensemble

	192.168.238.160:2181	192.168.238.161:2181	192.168.238.162:2181
ok	true	true	true
clientPort	2181	2181	2181
zk_server_state	follower	**leader**	follower
zk_version	3.4.14	3.4.14	3.4.14
zk_approximate_data_size	430334	430334	430334
zk_znode_count	270	270	270
zk_num_alive_connections	3	4	4
serverId	1	2	3
electionPort	3881	3881	3881
quorumPort	2881	2881	2881



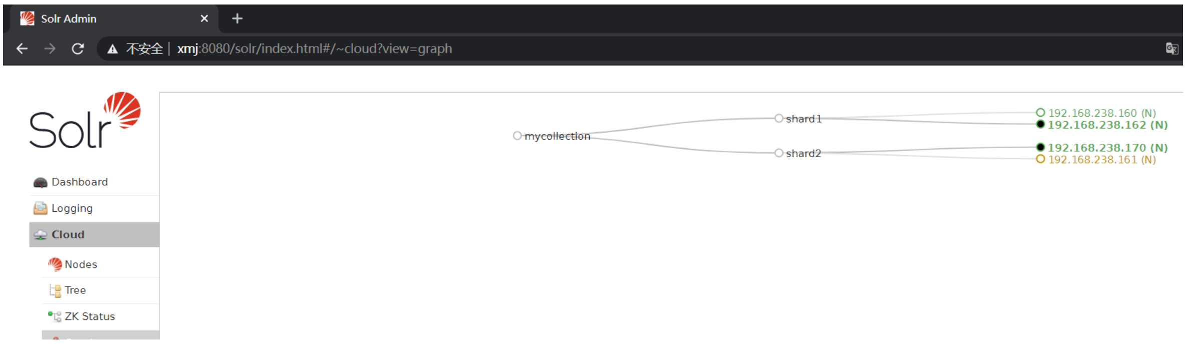
name: mycollection

config set: myconf

numShards: 2

replicationFact 2

Show advanced ☑



Solr Admin
不安全 | xmj:8080/solr/index.html#/~cloud?view=graph
Solr
Dashboard
Logging
Cloud
Nodes
Tree
ZK Status
mycollection
shard1
shard2
192.168.238.160 (N)
192.168.238.162 (N)
192.168.238.170 (N)
192.168.238.161 (N)