Solr cloud 5.0 部署

# 1. 环境准备

## 1.1. 服务器

3个centos 6.2，2\*4 核 2.2 cpu，16G内存，每个节点3块硬盘

## 1.2. 部署计划

|  |  |  |
| --- | --- | --- |
|  | solr | zookeeper |
| 172.16.5.1 | yes | no |
| 172.16.5.2 | yes | yes |
| 172.16.5.4 | yes | no |

### 1.2.1. Zookeeper

* 端口： 2181，也是默认端口

### 1.2.2. Solr

系统性能的优劣对比，大部分情况下比的是在完成同样的业务功能的情况下，谁能更饱和的利用所有硬件资源：比如在CPU利用率饱和的情况下，磁盘的读写是否达到了峰值。当然在不同的系统中，情况不一样，比如不能忽略系统的稳定性。

由于每个节点有三块硬盘，solr cloud部署有两个方案：

1. 一是每个节点部署三个solr服务，这样能保证多进程的读写磁盘，但是solr服务以及安装solr的容器会有额外的消耗，可能导致CPU和内存资源不够；
2. 二是使用单个solr服务，每个shard分布在不同磁盘，但是会导致集群的稳定性偏差：例如某个shard的崩溃导致j2ee容器异常退出，会误伤到其它两个shard；

这里因为是测试的目的，硬件资源有限，暂时采用方案二，生产系统在硬件资源充足的情况下建议使用方案一。

## 1.3. 软件环境

* solr cloud 5.0依赖jdk1.7 以上版本
* solr cloud 依赖zookeeper 来管理配置信息以及集群状态

下载jdk，选择1.7以上版本即可：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

下载zookper，选择release版本即可：

<http://zookeeper.apache.org/releases.html>

下载solr：

<http://www.apache.org/dyn/closer.cgi/lucene/solr/5.0.0>

设置3台节点的JAVA\_HOME 为1.7版本以上。

# 2. 安装

## 2.1. zookeeper

zookeeper 这里不做详细介绍，为了使安装流程尽量简化，这里只配置了一个zookeeper节点。就仅仅使用solr的角度来讲，zookeeper并不需要深入了解。

**解压：**

[root@localhost ~]# tar zxvf zookeeper-3.4.6.tar.gz

[root@localhost ~]# cd zookeeper-3.4.6

[root@localhost zookeeper-3.4.6]# cp conf/zoo\_sample.cfg conf/zoo.cfg

### 2.1.1. zoo.cfg

# 这里只列举了两个可能需要修改的配置

# zookeeper数据默认是存放在内存的，文件系统中会保留数据快照，这里就是快照的存放路径，不建议放在/tmp 目录下

dataDir=/tmp/zookeeper

# zookeeper服务端口

clientPort=2181

### 2.1.2. 服务状态维护

[root@localhost zookeeper-3.4.6]# ./bin/zkServer.sh

JMX enabled by default

Using config: /root/zookeeper-3.4.6/bin/../conf/zoo.cfg

Usage: ./bin/zkServer.sh {start|start-foreground|stop|restart|status|upgrade|print-cmd}

运行 zkServer.sh 看到相关的用法，使用 start 启动zookeeper。

[root@localhost zookeeper-3.4.6]# ./bin/zkServer.sh start

JMX enabled by default

Using config: /root/zookeeper-3.4.6/bin/../conf/zoo.cfg

Starting zookeeper ... STARTED

## 2.2. Solr

[root@localhost ~]# tar zxvf solr-5.0.0.tgz

[root@localhost ~]# cd solr-5.0.0

**查看solr命令使用介绍：**

[root@localhost solr-5.0.0]# ./bin/solr

**查看solr启动命令使用介绍：**

[root@localhost solr-5.0.0]# ./bin/solr start –help

**几个重要的启动参数：**

-c or –cloud : 以集群模式启动

-z : zookeeper的参数

**启动solr：**

[root@yoyo-demo-01 solr-5.0.0]# ./bin/solr start -c -z 172.16.5.2:2181

Waiting to see Solr listening on port 8983 [\]

Started Solr server on port 8983 (pid=3838). Happy searching!

**查看进程是否存在：**

[root@localhost solr-5.0.0]# ps aux | grep `pwd`

**查看端口是否已经启用：**

[root@localhost solr-5.0.0]# netstat –ntlp

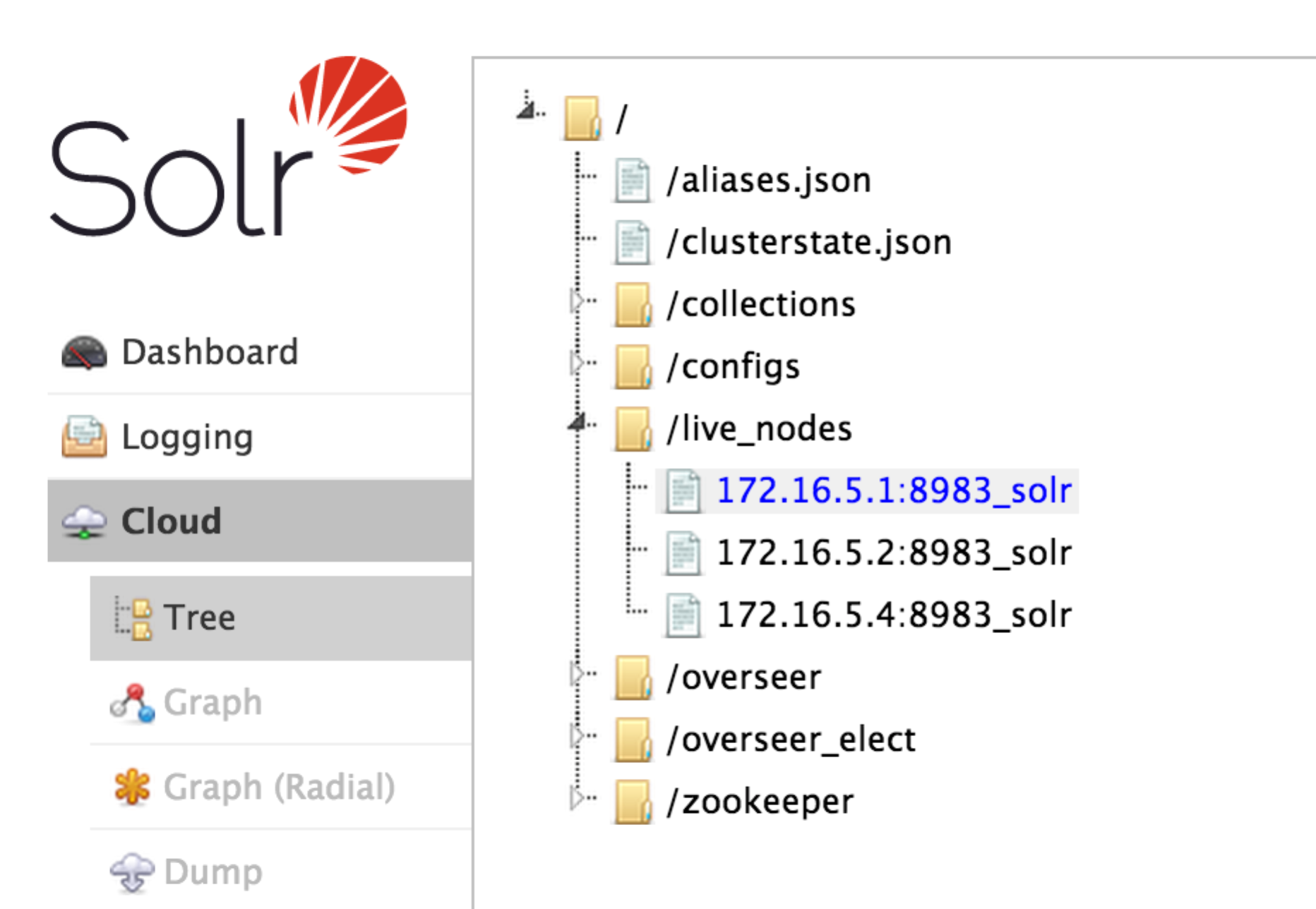
**依次启动集群中所有节点，之后登陆任意节点的solr管理控制台：**

<http://172.16.5.1:8983/solr/>

查看集群中所有节点是否都已启动成功：

<http://172.16.5.1:8983/solr/#/~cloud?view=tree>

每个节点启动之后，会在zookeeper中 /live\_nodes中写入一个空文件，如下图所示：



# 2.3. Solr使用测试

## 2.3.1. 相关概念

摘自： <http://tech.uc.cn/?p=2387>

* **Config Set**: Solr Core提供服务必须的一组配置文件。每个config set有一个名字。最小需要包括solrconfig.xml (SolrConfigXml)和schema.xml (SchemaXml)，除此之外，依据这两个文件的配置内容，可能还需要包含其它文件。它存储在Zookeeper中。Config sets可以重新上传或者使用upconfig命令更新，使用Solr的启动参数bootstrap\_confdir指定可以初始化或更新它。
* **Core**: 也就是Solr Core，一个Solr中包含一个或者多个Solr Core，每个Solr Core可以独立提供索引和查询功能，每个Solr Core对应一个索引或者Collection的Shard，Solr Core的提出是为了增加管理灵活性和共用资源。在SolrCloud中有个不同点是它使用的配置是在Zookeeper中的，传统的Solr core的配置文件是在磁盘上的配置目录中。
* **Leader**: 赢得选举的Shard replicas。每个Shard有多个Replicas，这几个Replicas需要选举来确定一个Leader。选举可以发生在任何时间，但是通常他们仅在某个Solr实例发生故障时才会触发。当索引documents时，SolrCloud会传递它们到此Shard对应的leader，leader再分发它们到全部Shard的replicas。
* **Replica**: Shard的一个拷贝。每个Replica存在于Solr的一个Core中。一个命名为“test”的collection以numShards=1创建，并且指定replicationFactor设置为2，这会产生2个replicas，也就是对应会有2个Core，每个在不同的机器或者Solr实例。一个会被命名为test\_shard1\_replica1，另一个命名为test\_shard1\_replica2。它们中的一个会被选举为Leader。
* **Shard**: Collection的逻辑分片。每个Shard被化成一个或者多个replicas，通过选举确定哪个是Leader。
* **Zookeeper**: Zookeeper提供分布式锁功能，对SolrCloud是必须的。它处理Leader选举。Solr可以以内嵌的Zookeeper运行，但是建议用独立的，并且最好有3个以上的主机。

## 2.3.2. 创建collection

solr安装包默认提供了一个实例配置： example/example-DIH/solr/solr/conf

创建命令：

[root@localhost solr-5.0.0]# ./bin/solr create\_collection

重要参数：

-c : collection名称

-d : 配置文件的路径，可以使用上面提供的实例配置

-n : 配置名称可以和collection名称不同，默认这个参数不填的话，会使用collection名称作为config名称

-shards : 创建的shard个数，建议和集群节点数量一致。

-replicationFactor : 每个shard的副本数，综合考虑为了保证集群的稳定性，建议配置为 最少2个，最多集群节点数量/shard数量 \* 2

[root@localhost solr-5.0.0]# ./bin/solr create\_collection -c example -d example/example-DIH/solr/solr/conf/ -shards 3 -replicationFactor 2

Connecting to ZooKeeper at 172.16.5.2:2181

Uploading /root/solr-5.0.0/example/example-DIH/solr/solr/conf for config example to ZooKeeper at 172.16.5.2:2181

Creating new collection 'example' using command:

http://172.16.5.1:8983/solr/admin/collections?action=CREATE&name=example&numShards=3&replicationFactor=2&maxShardsPerNode=2&collection.configName=example

{

"responseHeader":{

"status":0,

"QTime":5482},

"success":{"":{

"responseHeader":{

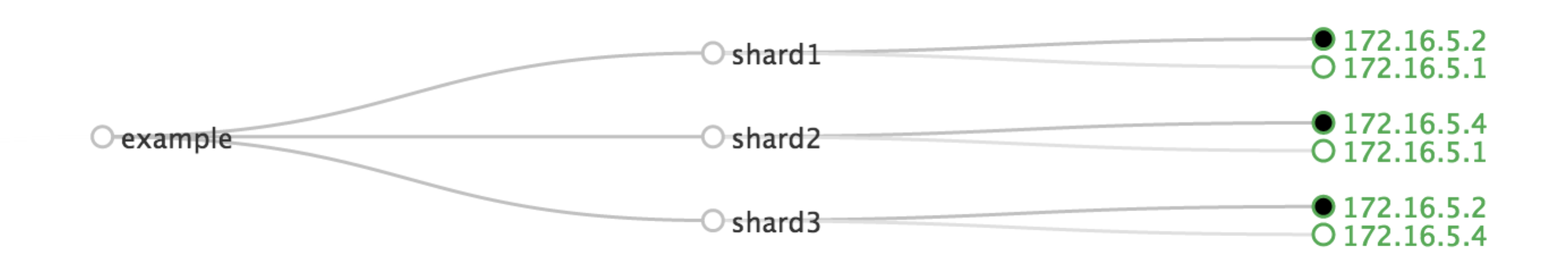
"status":0,

"QTime":5043},

"core":"example\_shard2\_replica2"}}}

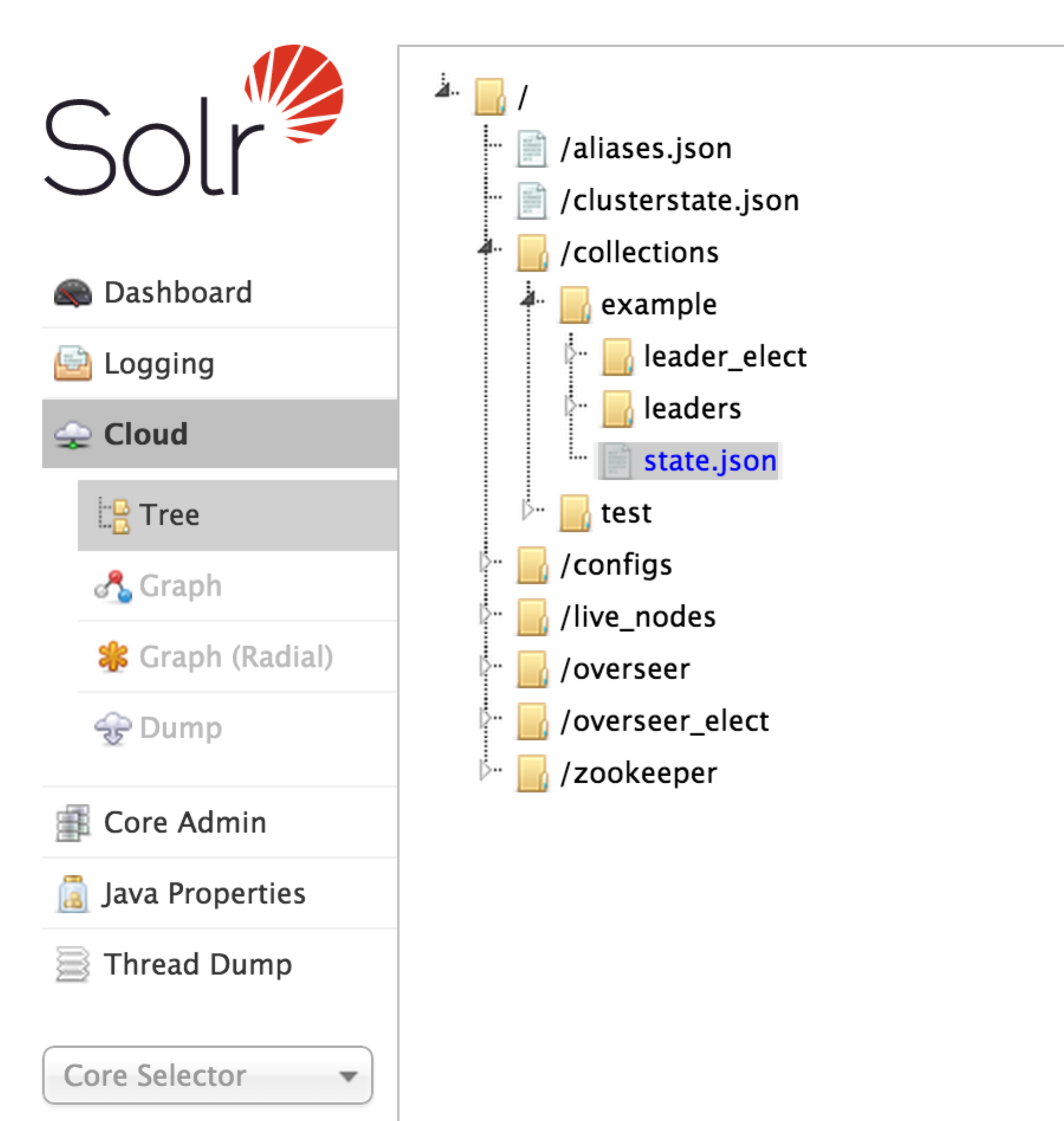
**查看collection创建结果：**

<http://172.16.5.1:8983/solr/#/~cloud>



**还可查看zookeeper中的状态：**

<http://172.16.5.1:8983/solr/#/~cloud?view=tree>



导入测试数据，可以使用example/exampledocs/ 中的数据：

<http://172.16.5.1:8983/solr/#/example_shard1_replica1/documents>

查询测试：

<http://172.16.5.1:8983/solr/#/example_shard1_replica1/query>

# 3. 结束语

至此solrcloud部署已经完成。Solr cloud维护比较麻烦，上文提到的多硬盘的部署以及集群扩容、数据迁移等等都可以单独拿出来长篇大论。Solr的数据是可以存放到hadoop fs中的，这样可以大大降低维护成本，尤其是使用例如cdh发行的hadoop版本，更能降低一个层次的维护成本。

Solr默认的官方版本不支持中文分词，需要自行下载分词器，并在schema.xml中做配置。

Solr支持的功能比较丰富，详细的功能介绍可以参考官方文档《apache-solr-ref-guide-5.0.pdf》的Searching章节，后续我也会补充部分使用心得。