

SolrCloud with Jboss

- [Introduction](#)
- [SolrCloud](#)
- [Getting Started](#)
- [Simple two shard cluster](#)
- [Installing and configuring Jboss](#)
- [Preparing Solr](#)
- [Installing and starting Zookeeper](#)
- [Upload the Solr configuration to the Zookeeper](#)
- [Modifying the solr.xml file](#)
 - [Pre 4.3.1](#)
 - [Solr 4.4 and above](#)
- [Starting JBoss](#)
- [Collection management using the Solr Collection Management API](#)
 - [Creating a new collection](#)
 - [Splitting a shard in two](#)
 - [Deleting documents from a Solr Collection](#)
- [Zookeeper management](#)
 - [Removing the clusterstate](#)

Introduction

In this section we will setup a [SolrCloud](#) using Jboss

SolrCloud

[SolrCloud](#) is the name of a set of new distributed capabilities in Solr. Passing parameters to enable these capabilities will enable you to set up a highly available, fault tolerant cluster of Solr servers. Use [SolrCloud](#) when you want high scale, fault tolerant, distributed indexing and search capabilities.

Getting Started

Download Solr 4-Beta or greater: <http://lucene.apache.org/solr/downloads.html>

If you haven't yet, go through the simple [Solr Tutorial](#) to familiarize yourself with Solr. Note: reset all configuration and remove documents from the tutorial before going through the cloud features. Copying the example directories with pre-existing Solr indexes will cause document counts to be off.

Download Jboss 7.x AS from <http://www.jboss.org/jbossas/>

Download Apache Zookeeper from <http://zookeeper.apache.org/>

Simple two shard cluster

<http://people.apache.org/~markrmiller/2shard2server.jpg>

In this example we will setup a 2 shard cluster using two running instances of Jboss. Both of the Jboss instances will be running on the same server and IP, but will be serving on different ports.

Installing and configuring Jboss

Unjar the downloaded Jboss jar files in two different directories. For this example, we will use /opt/jboss_1 and /opt/jboss_2

Default Jboss will serve the application on port 8080. We will keep that for jboss_1. For jboss_2 we will add a offset of the 100 in the socket definition for port, so that it starts on port 8180. Modify socket-binding-group section in the /opt/jboss_2/standalone/configuration/standalone-full.xml as follows. (Notice the port-offset:100)

```

    <socket-binding-group name="standard-sockets" default-interface="public" port-offset="{jboss.socket.
binding.port-offset:100}">
    <socket-binding name="management-native" interface="management" port="{jboss.management.native.port:
9999}" />
    <socket-binding name="management-http" interface="management" port="{jboss.management.http.port:9990}"
/>
    <socket-binding name="management-https" interface="management" port="{jboss.management.https.port:
9443}" />
    <socket-binding name="ajp" port="8009" />
    <socket-binding name="http" port="8080" />
    <socket-binding name="https" port="8443" />
    <socket-binding name="jacob" interface="unsecure" port="3528" />
    <socket-binding name="jacob-ssl" interface="unsecure" port="3529" />
    <socket-binding name="messaging" port="5445" />
    <socket-binding name="messaging-throughput" port="5455" />
    <socket-binding name="osgi-http" interface="management" port="8090" />
    <socket-binding name="remoting" port="4447" />
    <socket-binding name="txn-recovery-environment" port="4712" />
    <socket-binding name="txn-status-manager" port="4713" />
    <outbound-socket-binding name="mail-smtp">
        <remote-destination host="localhost" port="25" />
    </outbound-socket-binding>
</socket-binding-group>

```

Preparing Solr

For this example we will instantiate two instances of Solr, and place them in /opt/solr1 and /opt/solr2

Next copy the example/solr directory from the downloaded Solr tar file to the /opt/solr1 and /opt/solr2

```

cp example/solr /opt/solr1
cp example/solr /opt/solr2

```

Next modify the schema.xml in the /opt/solr1/collection1/conf/schema.xml and /opt/solr2/collection1/conf/schema.xml as needed

Installing and starting Zookeeper

Untar the Apache Zookeeper to a directory. For this example we will use /opt/zookeeper

Create a data for Zookeeper. For this example we will use /opt/zookeeper_data

copy the /opt/zookeeper/conf/zoo_sample.cfg to /opt/zookeeper/conf/zoo.cfg

```

cp /opt/zookeeper/conf/zoo_sample.cfg to /opt/zookeeper/conf/zoo.cfg

```

edit the /opt/zookeeper/conf/zoo.cfg file to modify the Zookeeper dataDir

```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sake.
dataDir=/opt/zookeeper_data
# the port at which the clients will connect
clientPort=2181
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
```

Start Zookeeper by using the zkServer.sh

```
/opt/zookeeper/bin/zkServer.sh start
```

This will start Zookeeper on port 2181

Upload the Solr configuration to the Zookeeper

Use the zkCli.sh bundled with Solr distribution for this. It is available in example/cloud-scripts/ as of Solr 4.3.1

Upload config to zookeeper using solr zookeeper cli

```
cloud-scripts/zkcli.sh -cmd upconfig -zkhost 127.0.0.1:2181 -confdir /opt/solr1/collection1/conf/ -confname
myconf
```

Link uploaded config with target collection. For this example, we will use collection1 as the collection name

```
cloud-scripts/zkcli.sh -cmd linkconfig -zkhost 127.0.0.1:2181 -collection collection1 -confname myconf -
solrhome solr
```

Modifying the solr.xml file

Pre 4.3.1

Next modify the /opt/solr1/solr.xml and /opt/solr2/solr.xml files to add definitions for the Zookeeper and the Shards

/opt/solr1/solr.xml should look as follows (notice the zkHost and shard values):

```
<?xml version="1.0" encoding="UTF-8" ?>
<solr persistent="true" zkHost="127.0.0.1:2181">
  <cores defaultCoreName="collection1" host="{host:}" adminPath="/admin/cores"
zkClientTimeout="{zkClientTimeout:15000}" hostPort="8180" hostContext="{hostContext:solr}">
    <core loadOnStartup="true" shard="shard1" instanceDir="collection1/" transient="false" name="collection1"
collection="collection1"/>
  </cores>
</solr>
```

/opt/solr2/solr.xml should look as follows (notice the zkHost and shard values):

```
<?xml version="1.0" encoding="UTF-8" ?>
<solr persistent="true" zkHost="127.0.0.1:2181">
  <cores defaultCoreName="collection1" host="{host:}" adminPath="/admin/cores"
zkClientTimeout="{zkClientTimeout:15000}" hostPort="8180" hostContext="{hostContext:solr}">
    <core loadOnStartup="true" shard="shard2" instanceDir="collection1/" transient="false" name="collection1"
collection="collection1"/>
  </cores>
</solr>
```

Solr 4.4 and above

Note zkHost attribute:

```
<solr>

  <solrcloud>
    <str name="host">{host:}</str>
    <int name="hostPort">8180</int>
    <str name="hostContext">{hostContext:solr}</str>
    <int name="zkClientTimeout">{zkClientTimeout:15000}</int>
    <bool name="genericCoreNodeNames">{genericCoreNodeNames:true}</bool>

    <str name="zkHost">127.0.0.1:2181</str>

  </solrcloud>

  <shardHandlerFactory name="shardHandlerFactory"
    class="HttpShardHandlerFactory">
    <int name="socketTimeout">{socketTimeout:0}</int>
    <int name="connTimeout">{connTimeout:0}</int>
  </shardHandlerFactory>

</solr>
```

Starting JBoss

Start the first instance of Jboss

```
/opt/jboss_1/jboss-as-7.1.1.Final-2/bin/standalone.sh -c standalone-full.xml
```

Start the second instance of Jboss

```
/opt/jboss_2/jboss-as-7.1.1.Final-2/bin/standalone.sh -c standalone-full.xml
```

You should now have a [SolrCloud](#) running in Jboss.

SolrCloudShardsSmall.png!

Collection management using the Solr Collection Managment API

Creating a new collection

The following will create a new collection called collection2 with two shards

```
{solrserver_base_url}/solr/admin/collections?  
action=CREATE&name=collection2&numShards=2&replicationFactor=1&maxShardsPerNode=2
```

Splitting a shard in two

The following will split a shard into two equal shards. The parent shard is not removed.

```
{solrserver_base_url}/solr/admin/collections?action=SPLITSHARD&collection=collection1&shard=shard1
```

Deleting documents from a Solr Collection

```
{solrserver_base_url}/solr/collection1_shard1_replica1/update?stream.body=<delete><query>*:*/</query><  
/delete>&commit=true
```

Zookeeper management

Removing the clusterstate

If you want to start over, you can remove the clusterstate.json from zookeeper as follows:

```
bin/zkCli.sh -server 127.0.0.1:2181  
[zk: localhost:2181] rmr /clusterstate.json
```