

Hadoop advanced.

Task1.

Here are the 2 of my namenodes:

The first screenshot shows the 'Overview' page for the active namenode '192.168.56.101:9000' (active). The page displays the following information:

Namespace:	ha-cluster
Namenode ID:	nn1
Started:	Fri Dec 14 12:54:19 -0500 2018
Version:	3.1.1, r2b9a8c1d3a2caf1e733d57f346af3ff0d5ba529c
Compiled:	Thu Aug 02 00:26:00 -0400 2018 by leftnoteasy from branch-3.1.1
Cluster ID:	CID-14f2aa7f-3319-43c2-94ac-5981efada2bf
Block Pool ID:	BP-1149363737-127.0.0.1-1544809170885

The second screenshot shows the 'Overview' page for the standby namenode '192.168.56.102:9000' (standby). The page displays the following information:

Namespace:	ha-cluster
Namenode ID:	nn2
Started:	Fri Dec 14 12:41:53 -0500 2018
Version:	3.1.1, r2b9a8c1d3a2caf1e733d57f346af3ff0d5ba529c
Compiled:	Thu Aug 02 00:26:00 -0400 2018 by leftnoteasy from branch-3.1.1
Cluster ID:	CID-d8891a1b-2d31-4029-a3e5-769127e6c432
Block Pool ID:	BP-1522273790-127.0.0.1-1544806605064

Here I uploaded test.csv into hdfs and executed "ls" to see that it's there.

```
[root@localhost ~]# hdfs dfs -copyFromLocal test.csv /user
[root@localhost ~]# hdfs dfs -ls /user
Found 1 items
-rw-r--r--    1 root supergroup  276554476 2018-12-14 12:55 /user/test.csv
[root@localhost ~]#
```

Next, in the screenshot below, I first check the running services with “jps”, as you can see NameNode is there. Then I stop it by executing “hdfs –daemon stop namenode” and voila, I can still access hdfs and my standby node become the active one.

```

centos2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 13:28

root@localhost:~/hadoop-3.1.1/etc/hadoop

[root@localhost hadoop]# jps
12384 QuorumPeerMain
18454 ResourceManager
31256 NameNode
31354 JournalNode
31499 DFSZKFailoverController
11406 DataNode
31599 Jps
[root@localhost hadoop]# hdfs haadmin -getServiceState nn1
standby
[root@localhost hadoop]# hdfs haadmin -getServiceState nn2
active
[root@localhost hadoop]# jps
12384 QuorumPeerMain
18454 ResourceManager
31256 NameNode
31720 Jps
31354 JournalNode
31499 DFSZKFailoverController
11406 DataNode
[root@localhost hadoop]# hdfs dfs -ls /user
Found 1 items
-rw-r--r-- 1 root supergroup 276554476 2018-12-14 12:55 /user/test.csv
[root@localhost hadoop]# hdfs --daemon stop namenode
[root@localhost hadoop]# hdfs dfs -ls /user
Found 1 items
-rw-r--r-- 1 root supergroup 276554476 2018-12-14 12:55 /user/test.csv
[root@localhost hadoop]# hdfs haadmin -getServiceState nn1
active
[root@localhost hadoop]# hdfs haadmin -getServiceState nn2
2018-12-14 13:27:24,406 INFO ipc.Client: Retrying connect to server: 192.168.56.102/192.168.56.102:9000. Already tried 0 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=1, sleepTime=1000 MILLISECONDS)
Operation failed: Call From localhost/127.0.0.1 to 192.168.56.102:9000 failed on connection exception: java.net

```

Task2.

I followed the tutorial on spring’s website, everything in my code is the same as there, except the Pojo implementation, which I modified to suit task 2.

Here’s how I start my application:

```
[pavel_orekhov@ecsc00a02339 HadoopAdvanced]$ java -jar gs-yarn-basic-dist/target/gs-yarn-basic-dist/gs-yarn-basic-client-0.1.0.jar --my.client.filePath=/user/train.csv
```

Here’s the result for train.csv:

```

[pavel_orekhov@ecsc00a02339 HadoopAdvanced]$ yarn logs -applicationId application_1545290316073_0014 | grep JobPojo
18/12/20 09:27:10 INFO client.RMPProxy: Connecting to ResourceManager at ecsc00a022c6.epam.com/10.6.218.24:8032
[2018-12-20 09:26:39.618] boot - 26751 INFO [main] --- JobPojo: Answer: 1277716
[2018-12-20 09:26:39.618] boot - 26751 INFO [main] --- JobPojo: Answer: 275737
[2018-12-20 09:26:39.618] boot - 26751 INFO [main] --- JobPojo: Answer: 141535
[pavel_orekhov@ecsc00a02339 HadoopAdvanced]$

```