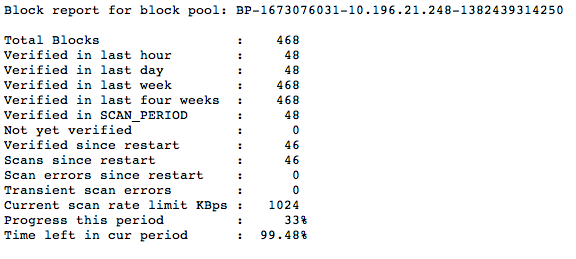
Lab 5.1: Verify Data with Block Scanner and fsck

|  |  |
| --- | --- |
| **Objective:** | View the various tools for performing block verification and the health of files in HDFS. |
| **Successful Outcome:** | You will see the result of the Block Scanner Report on node1, and the output of the fsck command. |
| **Before You Begin:** | SSH into node1. |

1. Configure the Scan Period
   1. Go to HDFS -> Configs, then expand the Custom hdfs-site.xml section and click the Add Property
   2. Add the following property and save the configs:

dfs.datanode.scan.period.hours=1

1. Restart the HDFS Service
2. View the Block Scanner Report
   1. Point your web browser to http://*node1*:50075/blockScannerReport. The report will look similar to the following:

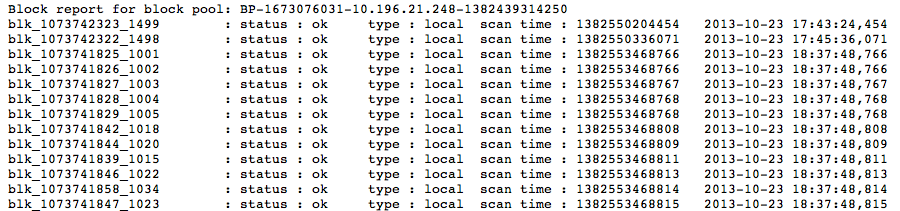


* 1. How many blocks are on your node1 DataNode? \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. View the Block Details
   1. Add the listblocks parameter to the blockScannerReport URL:

http://node1:50075/blockScannerReport?listblocks

You should see a list of all blocks on that DataNode and their status:



**NOTE**: If a block is corrupt, the NameNode is notified and attempts to fix the issue. The default time period for scanning blocks is every three weeks, so in a production environment you would not set this interval to 30 minutes like you did in this lab. Use the block scanner report as a quick way to verify the integrity of the blocks in your cluster.

1. Run the fsck Command on a File
   1. Put the file /root/data/test\_data into the /user/root folder of HDFS:

# hadoop fs -put ~/data/test\_data

* 1. Run the fsck command for the file /user/root/test\_data:

# hdfs fsck /user/root/test\_data -files

* 1. How many blocks did test\_data get split into? \_\_\_\_\_\_\_\_\_\_\_\_
  2. What is the average block replication of test\_data? \_\_\_\_\_\_\_\_\_\_\_

1. Using fsck Options
   1. Run the fsck command again, but this time add the -blocks option:

# hdfs fsck /user/root/test\_data -files -blocks

* 1. What did the blocks option add to the output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  2. Add the -locations option as well:

# hdfs fsck /user/root/test\_data -files -blocks -locations

* 1. What did the locations option add to the output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Run a File system Check
   1. You can run fsck on the entire file system. Enter the following command:

# hdfs fsck /

Notice this command fails, because root does not have permission to view all the files in HDFS.

* 1. Switch to the hdfs user:

# su - hdfs

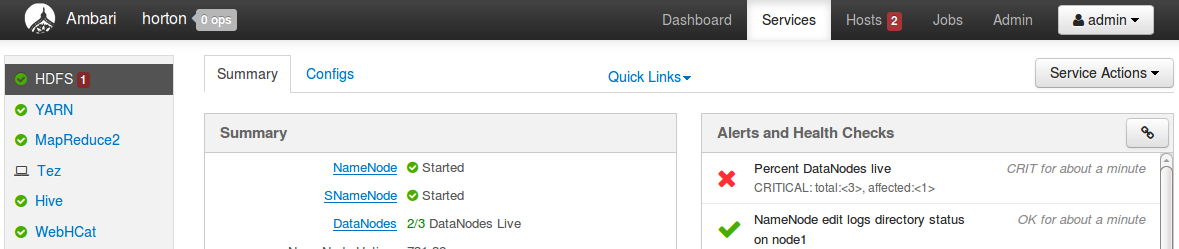
* 1. Run fsck on the entire file system:

$ hdfs fsck /

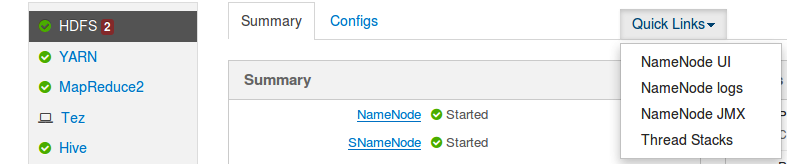
* 1. What is the total size of your HDFS? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  2. How many directories does your cluster have? \_\_\_\_\_\_\_\_\_\_\_\_
  3. How many files are on your cluster? \_\_\_\_\_\_\_\_\_\_\_\_\_
  4. How many total blocks are on your cluster? \_\_\_\_\_\_\_\_\_\_\_
  5. What is the average block replication of your cluster? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  6. Switch back to the root user:

[hdfs@node1 ~]$ exit

1. View the Health of your Cluster
   1. From the Hosts page for node1 in Ambari, stop the DataNode service on node1.
   2. Click on the Services page, then HDFS. You should see that one of your DataNodes is not live.



* 1. Using the Quick Links menu on the HDFS Services page (shown in the screenshot below), open the NameNode UI:



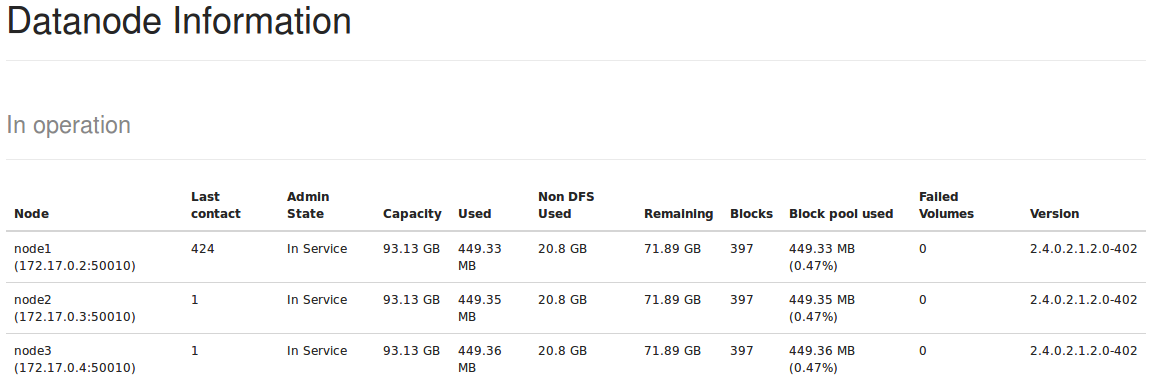
The NameNode UI opens the dfshealth.jsp page by default.

* 1. Notice you still have 3 Live Nodes and 0 Dead Nodes:

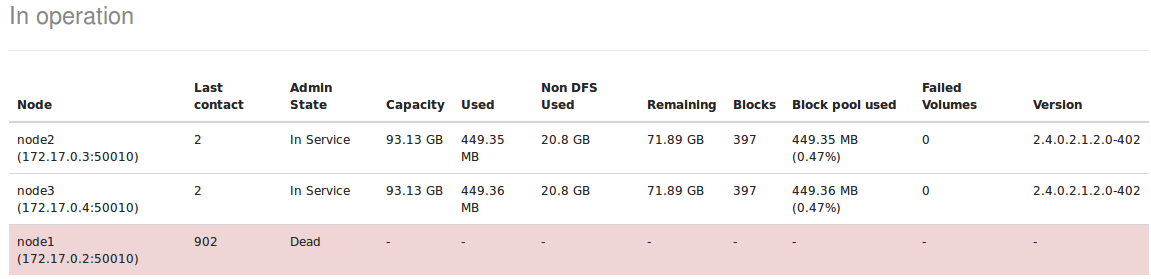


**NOTE**: It takes 10.5 minutes for a DataNode to be marked as dead in a cluster.

* 1. Click on the Live Nodes link to view the Live DataNodes in your cluster:



1. Take a Break
   1. ...and wait for your DataNode to be marked as Dead in your cluster!
   2. Refresh the Live DataNodes page, and you should only see two live DataNodes:



* 1. Go back to the dfshealth.jsp page and refresh it. Notice you now have 1 Dead Node and a large number of under-replicated blocks:



* 1. Why does your cluster have so many under-replicated blocks? \_\_\_\_\_\_\_\_\_\_\_  
       
     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Run fsck Again
   1. Switch to the hdfs user and run fsck on the entire file system:

# su - hdfs

[hdfs@node1 ~]$ hdfs fsck /

Notice you get a long list of every file that contains under-replicated blocks.

* 1. What is the average block replication now on your cluster? \_\_\_\_\_\_\_\_\_\_\_\_\_
  2. Compare the value of Missing replicas in the output of fsck with the value of Number of Under-Replication Blocks in the NameNode UI.

1. Start the DataNode Again
   1. Using Ambari, start the DataNode process on node1.
   2. Refresh the dfshealth.jsp page in the NameNode UI frequently, and you can watch as the number of under-replicated blocks gradually decreases to 0:





* 1. Run fsck again on your entire file system, and notice everything is back to normal again.

**RESULT**: The Block Scanner Report is a quick way to view the status of the blocks on the DataNodes of your cluster. The fsck tool is a great way to view the health of your file system and block replication, as is using the NameNode UI.