**ASSIGNMENT 8.6**

**Problem Statement:**

* How can you check the status of a job submitted in hortonworks cluster
* Open the HDFS web UI in hortonworks cluster and browse the file system
* How can you check the log files of a job in hortonworks cluster

**Solution:**

* **How can you check the status of a job submitted in hortonworks cluster:**

Similar to the information available from the Resource Manager UI about MapReduce Jobs, it is possible to obtain details on Running and Finished MapReduce Jobs on the command line. Information such as the number of maps and reduces, various counters and resource usage by the job can be collected using the "mapred" command line tool.

**Follow the steps given below for "FINISHED" MapReduce Jobs:**

Obtain the specific application ID and job ID by running the following command as "yarn" user:

# yarn application -list -appStates FINISHED

**An example output is as follows:**

application\_1448259330136\_0016 select count(\*) from hdfsa...dt>"2015-11-23"(Stage-1) MAPREDUCE ratish hiveq FINISHED SUCCEEDED 100% http://rm-testnode1.novalocal:19888/jobhistory/job/job\_1448259330136\_0016

**Use the job ID from the URL in the previous command output to check the job details:**

# mapred job -status <Job\_ID>

**And for Jobs that are "RUNNING", follow the below steps to find the job details:**

Find the application ID from the list of RUNNING applications:

Login as "yarn" user, and run the command:

# yarn application -list -appStates RUNNING

An example output is as follows:

application\_1450309877692\_0007 word count MAPREDUCE ambari-qa default RUNNING UNDEFINED 50% <http://rm-testnode1.novalocal:57521>

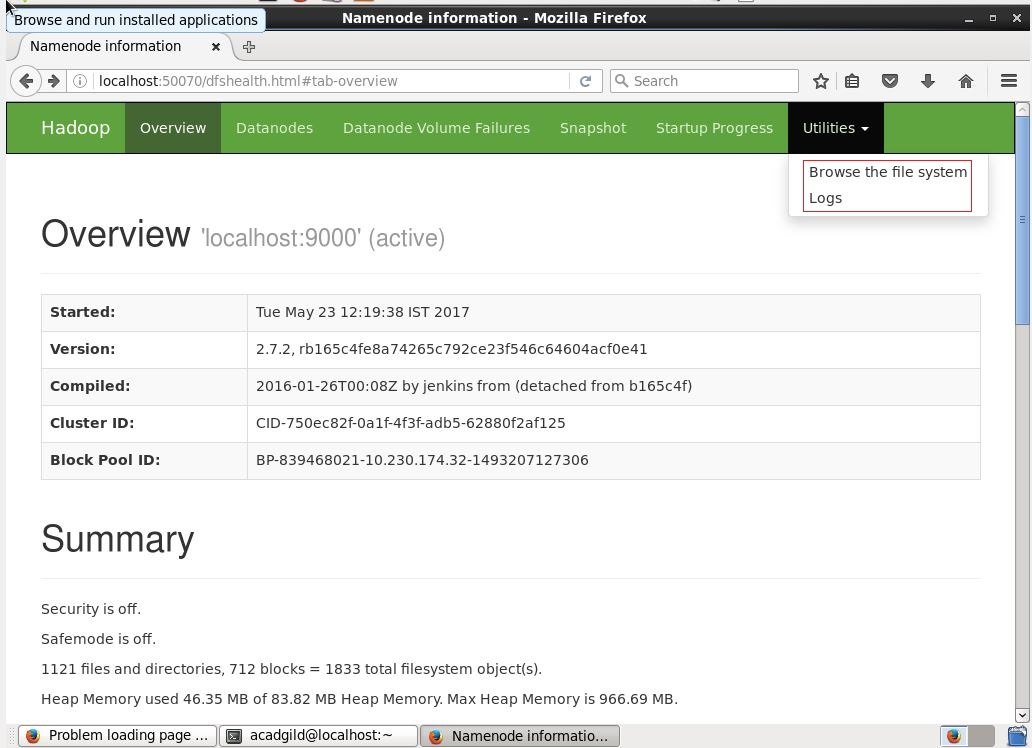
Get the corresponding job ID from mapred command:

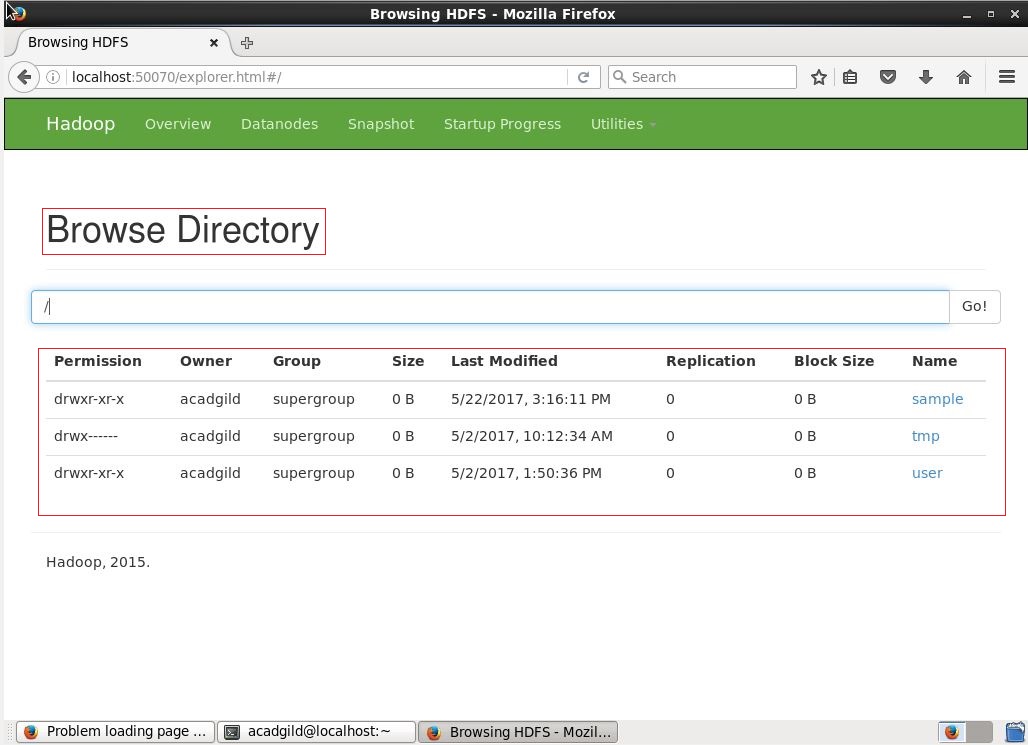
# mapred job -list all | grep <Application ID>

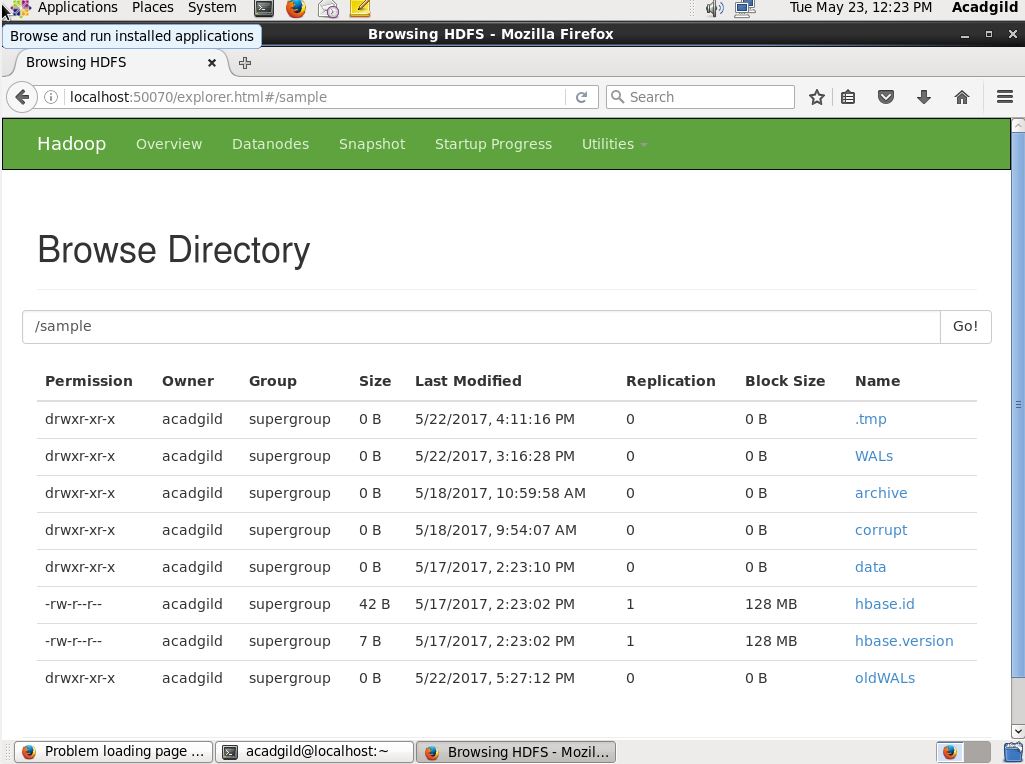
Use the Job ID from step #2 to get the job details:

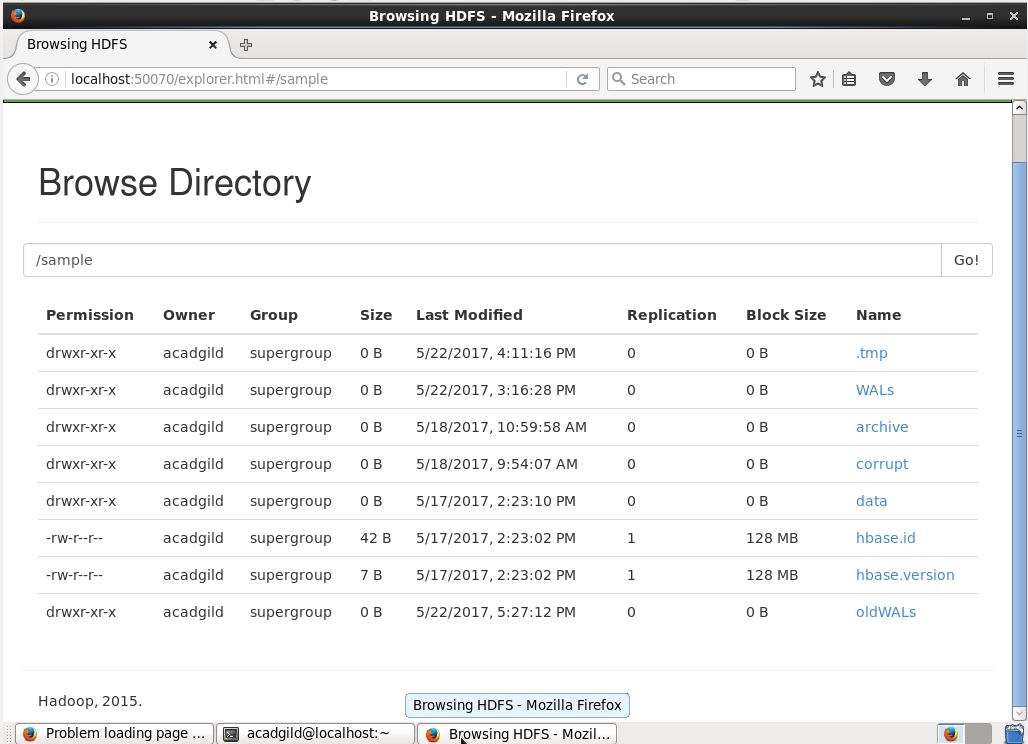
# mapred job -status <Job\_ID>

* **Open the HDFS web UI in hortonworks cluster and browse the file system**

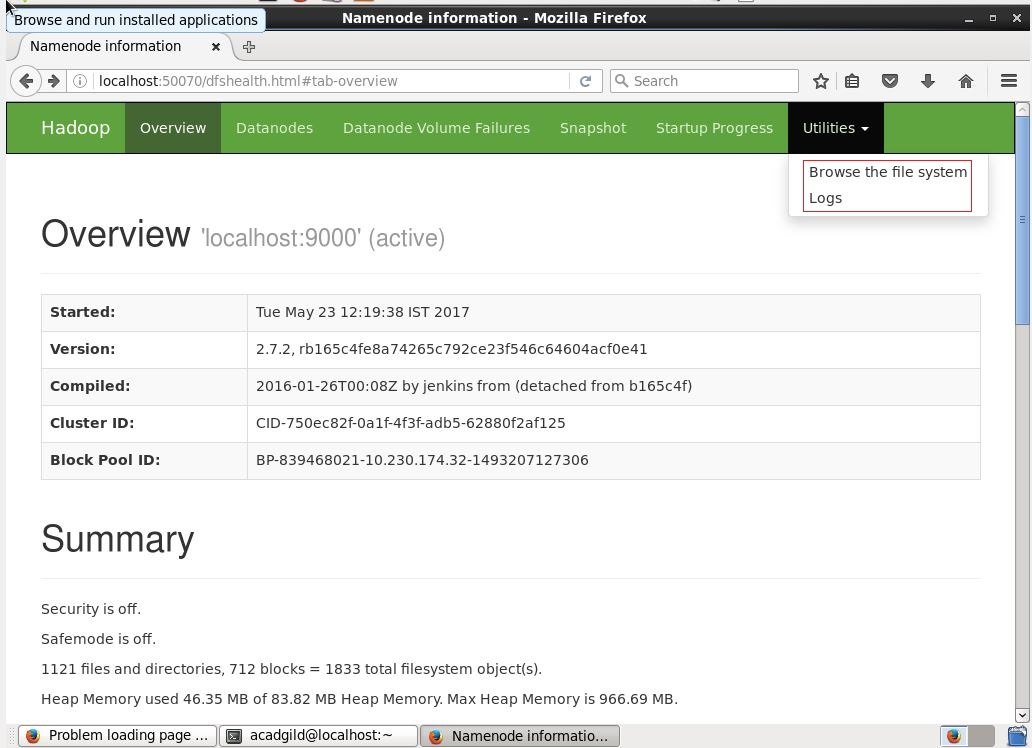


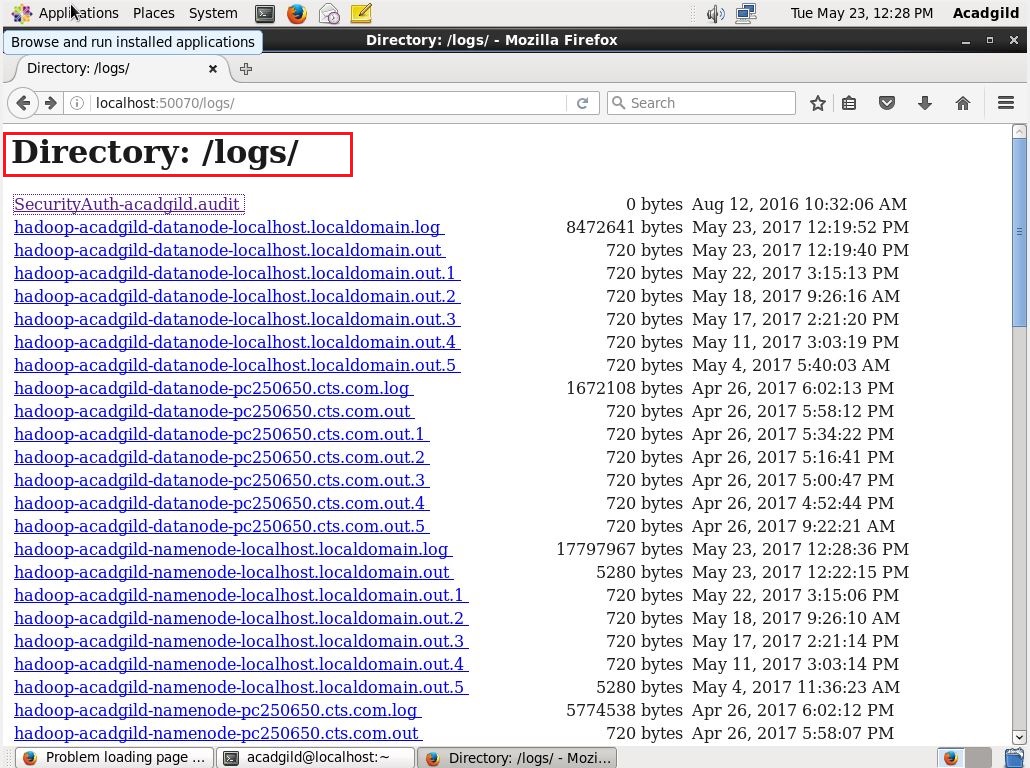






* **How can you check the log files of a job in hortonworks cluster:**





**Hive Logging:**

Hive uses log4j for logging. By default logs are not emitted to the console by the CLI. The default logging level is WARN for Hive releases prior to 0.13.0. Starting with Hive 0.13.0, the default logging level is INFO.

The logs are stored in the directory /tmp/<user.name>:

/tmp/<user.name>/hive.log Note: In local mode, prior to Hive 0.13.0 the log file name was ".log" instead of "hive.log". This bug was fixed in release 0.13.0 (see HIVE-5528 and HIVE-5676).

To configure a different log location, set hive.log.dir in $HIVE\_HOME/conf/hive-log4j.properties. Make sure the directory has the sticky bit set (chmod 1777 <dir>).

hive.log.dir=<other\_location>

If the user wishes, the logs can be emitted to the console by adding the arguments shown below:

bin/hive --hiveconf hive.root.logger=INFO,console //for HiveCLI (deprecated)

bin/hiveserver2 --hiveconf hive.root.logger=INFO,console

Alternatively, the user can change the logging level only by using:

bin/hive --hiveconf hive.root.logger=INFO,DRFA //for HiveCLI (deprecated)

bin/hiveserver2 --hiveconf hive.root.logger=INFO,DRFA

Another option for logging is TimeBasedRollingPolicy (applicable for Hive 0.15.0 and above, HIVE-9001) by providing DAILY option as shown below:

bin/hive --hiveconf hive.root.logger=INFO,DAILY //for HiveCLI (deprecated)

bin/hiveserver2 --hiveconf hive.root.logger=INFO,DAILY

Note that setting hive.root.logger via the 'set' command does not change logging properties since they are determined at initialization time.

Hive also stores query logs on a per Hive session basis in /tmp/<user.name>/, but can be configured in hive-site.xml with the hive.querylog.location property.

Logging during Hive execution on a Hadoop cluster is controlled by Hadoop configuration. Usually Hadoop will produce one log file per map and reduce task stored on the cluster machine(s) where the task was executed. The log files can be obtained by clicking through to the Task Details page from the Hadoop JobTracker Web UI.

When using local mode (using mapreduce.framework.name=local), Hadoop/Hive execution logs are produced on the client machine itself. Starting with release 0.6 – Hive uses the hive-exec-log4j.properties (falling back to hive-log4j.properties only if it's missing) to determine where these logs are delivered by default. The default configuration file produces one log file per query executed in local mode and stores it under /tmp/<user.name>.

The intent of providing a separate configuration file is to enable administrators to centralize execution log capture if desired (on a NFS file server for example). Execution logs are invaluable for debugging run-time errors