[edureka@localhost ~]$ hadoop

Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]

CLASSNAME run the class named CLASSNAME

or

where COMMAND is one of:

fs run a generic filesystem user client

version print the version

jar <jar> run a jar file

note: please use "yarn jar" to launch

YARN applications, not this command.

checknative [-a|-h] check native hadoop and compression libraries availability

distcp <srcurl> <desturl> copy file or directories recursively

archive -archiveName NAME -p <parent path> <src>\* <dest> create a hadoop archive

classpath prints the class path needed to get the

Hadoop jar and the required libraries

credential interact with credential providers

daemonlog get/set the log level for each daemon

trace view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.

1. hadoop fs

the hadoop utility first load the hadoop-config then identify the command. Here it is fs command and then assign org.apache.hadoop.fs.FsShell to CLASS parameter. It loads classpath then execute java program as below with this class -

$ java org.apache.hadoop.fs.FsShell -put WordCount.txt /user/edureka

-> it calls main() method here -> from there it calls ToolRunner.run() method. In run method for sub command -put, it calls copyFromLocal() method.

You can see code for FsShell class at grepcode website -

http://grepcode.com/file/repo1.maven.org/maven2/com.ning/metrics.action/0.2.7/org/apache/hadoop/fs/FsShell.java

2. hadoop version

It prints installed hadoop version as per HADOOP\_HOME environment variable as below-

[edureka@localhost bin]$ hadoop version

Hadoop 2.8.1

Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r 20fe5304904fc2f5a18053c389e43cd26f7a70fe

Compiled by vinodkv on 2017-06-02T06:14Z

Compiled with protoc 2.5.0

From source with checksum 60125541c2b3e266cbf3becc5bda666

This command was run using /usr/lib/hadoop-2.8.1/share/hadoop/common/hadoop-common-2.8.1.jar

[edureka@localhost bin]$ echo $HADOOP\_HOME

/usr/lib/hadoop-2.8.1

Here as per version command, it assign org.apache.hadoop.util.VersionInfo to CLASS variable

$java org.apache.hadoop.util.VersionInfo

it calls main() method -> from there it calls getVersion() method -> from there it calls HadoopVersionAnnotation.version().

You can see this class code at below -

http://grepcode.com/file/repo1.maven.org/maven2/com.ning/metrics.action/0.2.7/org/apache/hadoop/util/VersionInfo.java?av=f

3. hadoop jar

This jar command uses to the run the jar as below -

$ hadoop jar hadoop-mapreduce-examples-2.8.1.jar pi 2 5

Here as per jar command, it assign org.apache.hadoop.util.RunJar to CLASS variable in hadoop utility then execute java command as below -

$ org.apache.hadoop.util.RunJar hadoop-mapreduce-examples-2.8.1.jar pi 2 5

it calls main() method-> from there it gets jar file name which asses as argument and check the manifest for jar main class to execute with its argument via runtime class loader

You can see the class code at below -

http://grepcode.com/file/repo1.maven.org/maven2/com.ning/metrics.action/0.2.7/org/apache/hadoop/util/RunJar.java#RunJar.main%28java.lang.String%5B%5D%29

4. hadoop classpath

prints the class path needed to get the Hadoop jar and the required libraries.

[edureka@localhost bin]$ hadoop classpath

/usr/lib/hadoop-2.8.1/etc/hadoop:/usr/lib/hadoop-2.8.1/share/hadoop/common/lib/\*:/usr/lib/hadoop-2.8.1/share/hadoop/common/\*:/usr/lib/hadoop-2.8.1/share/hadoop/hdfs:/usr/lib/hadoop-2.8.1/share/hadoop/hdfs/lib/\*:/usr/lib/hadoop-2.8.1/share/hadoop/hdfs/\*:/usr/lib/hadoop-2.8.1/share/hadoop/yarn/lib/\*:/usr/lib/hadoop-2.8.1/share/hadoop/yarn/\*:/usr/lib/hadoop-2.8.1/share/hadoop/mapreduce/lib/\*:/usr/lib/hadoop-2.8.1/share/hadoop/mapreduce/\*:/usr/lib/hadoop-2.8.1/contrib/capacity-scheduler/\*.jar

As per the command classpath, it assign org.apache.hadoop.util.Classpath to CLASS variable then execute java programme as below -

$ java org.apache.hadoop.util.Classpath

It runs main() method -> it checks for argument , then print all jars from class path as per system property java.class.path.

You can see the code as below -

http://grepcode.com/file/repo1.maven.org/maven2/org.apache.hadoop/hadoop-common/2.7.1/org/apache/hadoop/util/Classpath.java#Classpath

5. hadoop CLASSNAME

Runs the class named CLASSNAME. The class must be part of a package.

$ java Main

6. hadoop archive

It archive the hdfs

archive -archiveName NAME -p <parent path> <src>\* <dest> create a hadoop archive

$ hadoop archive -archiveName archiveTest.har -p /user/edureka -r 3 dir1 dir2 /user/archive

The above example is creating an archive using /user/edureka as the relative archive directory. The directories /user/hadoop/dir1 and /user/hadoop/dir2 will be archived in the following file system directory – /user/archive/archiveTest.har. Archiving does not delete the input files. If you want to delete the input files after creating the archives (to reduce namespace), you will have to do it on your own. In this example, because -r 3 is specified, a replication factor of 3 will be used.

This command assign org.apache.hadoop.tools.HadoopArchives to CLASS variable.

$ java org.apache.hadoop.tools.HadoopArchives -archiveName archiveTest.har -p /user/edureka -r 3 dir1 dir2 /user/archive

it calls main() method from it calls ToolRunner.run() method

http://grepcode.com/file/repo1.maven.org/maven2/org.apache.hadoop/hadoop-archives/2.7.1/org/apache/hadoop/tools/HadoopArchives.java#HadoopArchives

7. hadoop distcp

This command uses for copy file or directories recursively. DistCp (distributed copy) is a tool used for large inter/intra-cluster copying.

hadoop distcp <srcurl> <desturl>

$ hadoop distcp /user/edureka/WordCount.txt hdfs://otherip:9000/user/hadoop/

This command assign org.apache.hadoop.tools.DistCp to CLASS variable.

$ java org.apache.hadoop.tools.DistCp /edureka/WordCount.txt hdfs://otherip:9000/user/hadoop/

it calls main() method from it calls ToolRunner.run() method -> run()->copy()

http://grepcode.com/file/repository.cloudera.com/content/repositories/releases/com.cloudera.hadoop/hadoop-tools/0.20.2-737/org/apache/hadoop/tools/DistCp.java#DistCp.copy%28org.apache.hadoop.conf.Configuration%2Corg.apache.hadoop.tools.DistCp.Arguments%29

8. trace

This command is for view and modify Hadoop tracing settings.

$ hadoop trace

This command assign org.apache.hadoop.tracing.TraceAdmin to CLASS variable.

$ java org.apache.hadoop.tracing.TraceAdmin

it calls main() method, from there it sets the configuration and call run method.

http://grepcode.com/file/repo1.maven.org/maven2/org.apache.hadoop/hadoop-common/2.7.0/org/apache/hadoop/tracing/TraceAdmin.java#TraceAdmin.run%28java.lang.String%5B%5D%29