**Importance of NameNode**

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these files itself.

Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file. The NameNode responds the successful requests by returning a list of relevant [DataNode](https://wiki.apache.org/hadoop/DataNode) servers where the data lives.

The NameNode is a [Single Point of Failure](https://wiki.apache.org/hadoop/Single%20Point%20of%20Failure) for the HDFS Cluster. HDFS is not currently a High Availability system. When the NameNode goes down, the file system goes offline. There is an optional [SecondaryNameNode](https://wiki.apache.org/hadoop/SecondaryNameNode) that can be hosted on a separate machine. It only creates checkpoints of the namespace by merging the edits file into the fsimage file and does not provide any real redundancy. Hadoop 0.21+ has a [BackupNameNode](https://wiki.apache.org/hadoop/BackupNameNode) that is part of a plan to have an HA name service, but it needs active contributions from the people who want it (i.e. you) to make it Highly Available.

It is essential to look after the NameNode. Here are some recommendations from production use

* Use a good server with lots of RAM. The more RAM you have, the bigger the file system, or the smaller the block size.
* Use ECC RAM.
* On Java6u15 or later, run the server VM with compressed pointers -XX:+UseCompressedOops to cut the JVM heap size down.
* List more than one name node directory in the configuration, so that multiple copies of the file system meta-data will be stored. As long as the directories are on separate disks, a single disk failure will not corrupt the meta-data.
* Configure the NameNode to store one set of transaction logs on a separate disk from the image.
* Configure the NameNode to store another set of transaction logs to a network mounted disk.
* Monitor the disk space available to the NameNode. If free space is getting low, add more storage.
* Do not host [DataNode](https://wiki.apache.org/hadoop/DataNode), [JobTracker](https://wiki.apache.org/hadoop/JobTracker) or [TaskTracker](https://wiki.apache.org/hadoop/TaskTracker) services on the same system.

\*NameNode is the center piece of HDFS.

\*NameNode is also known as the Master

\*NameNode only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster.

\*NameNode does not store the actual data or the dataset.

The data itself is actually stored in the DataNodes.

\*NameNode knows the list of the blocks and its location for any given file

in HDFS. With this information NameNode knows how to construct the file

from blocks.

\*NameNode is so critical to HDFS and when the NameNode is down,

HDFS/Hadoop cluster is inaccessible and considered down.

\*NameNode is a single point of failure in Hadoop cluster.

\*NameNode is usually configured with a lot of memory (RAM).Because the block locations are help in main memory.

• NameNode contains two important files on its hard disk:

1. fsimage (file system image)

It contains:

• All directory structure of HDFS, Replication level of file, Modification and access times of files, Access permissions of files and directories,Block size of files,The blocks constituting a file

2. Edits

• When any write operation takes place in HDFS, the directory structure

gets modified

• These modifications are stored in the memory as well as in edits files

• If existing fsimage file gets merged with edits, we’ll get an updated fsimage

file

• This process is called Checkpointing and is carried out by the Secondary

NameNode. It takes fsimage and edits files from NameNode and returns

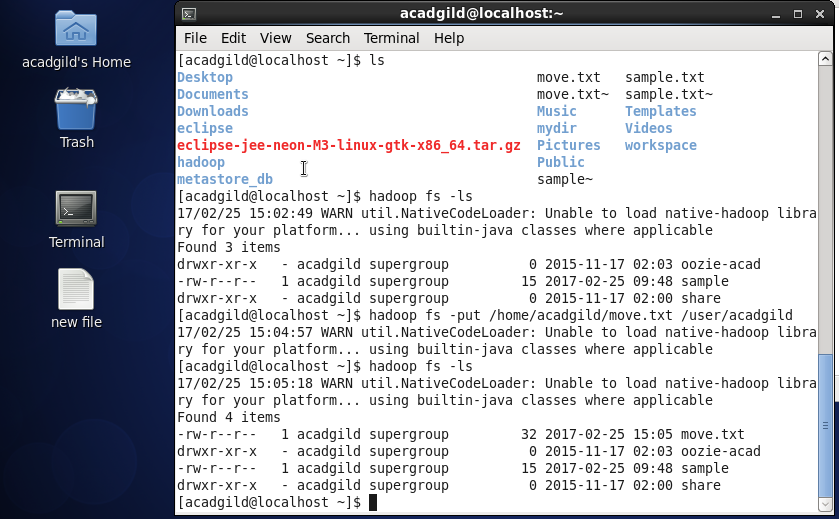
updated fsimage file after merging

In Hadoop 1.x, NameNode is the single point of failure. If it fails, we won’t be able to access the

cluster

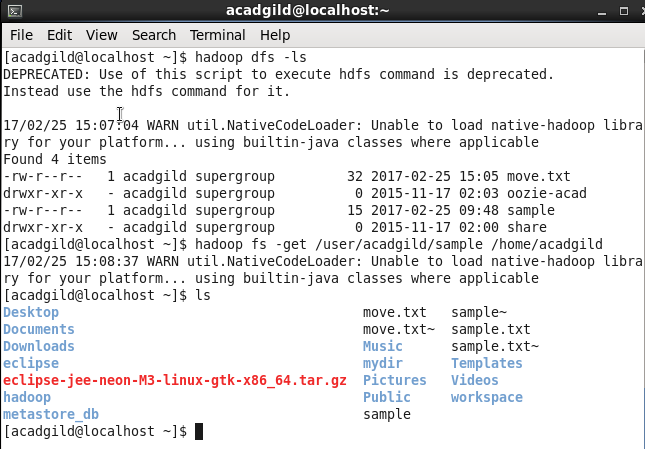
**basic Hadoop beginners commands**

**1.put command and ls command**



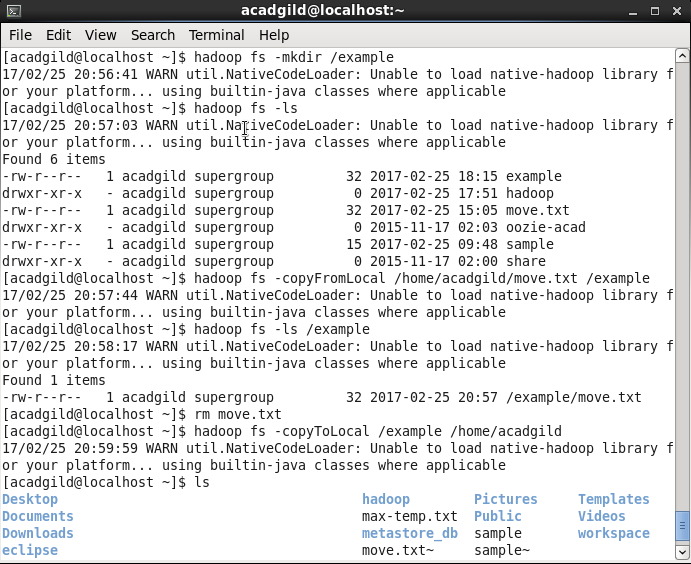
In home/acadgild the file move.txt is moved to Hadoop in location user/acadgild

Using ls command we can see the files before and after copying to Hadoop.

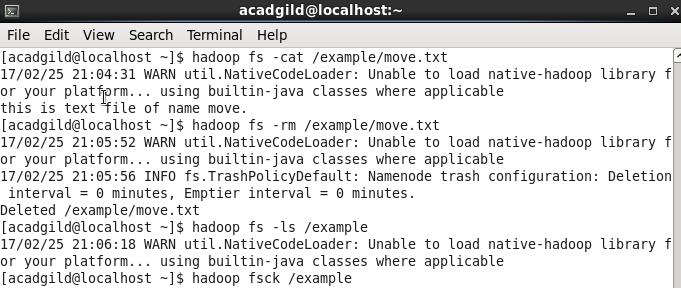
**2.get command**

**Here sample file from hadoop is moved to local path using get.**

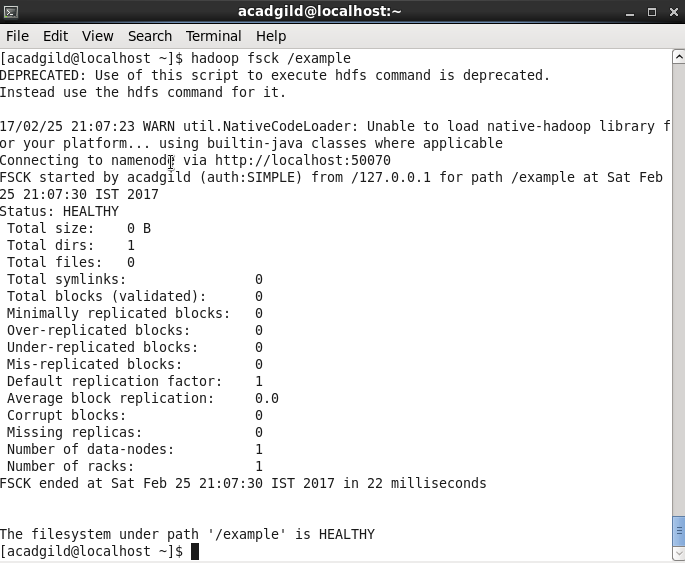
**3.mkdir ,copyfromlocal,copytolocal commands**

****

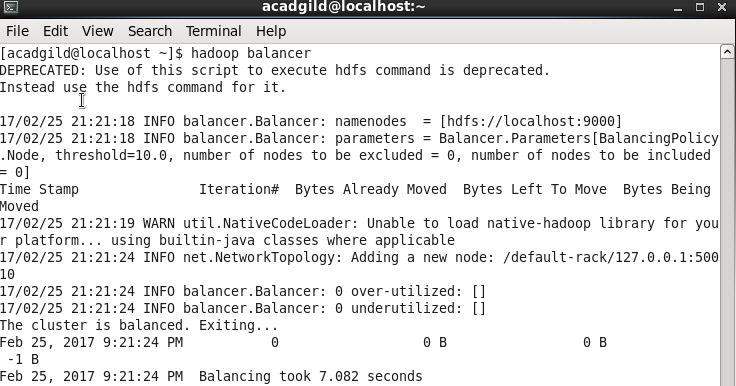
### 4.cat and rm command

****

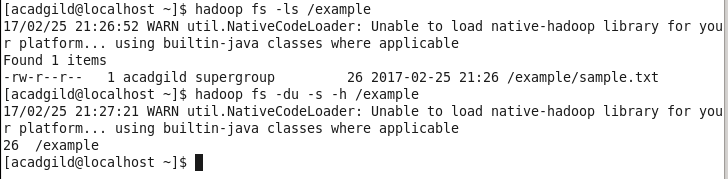
**Run a DFS Filesystem to Check Utility**

****

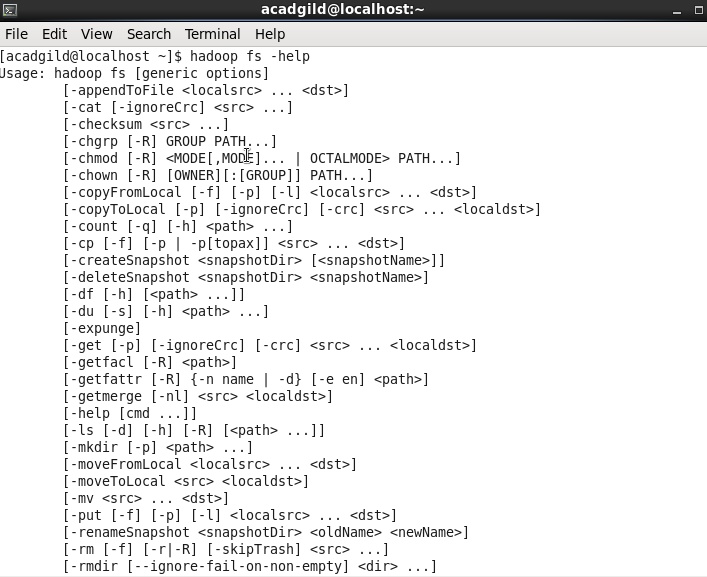
**Hadoop balancer**

****

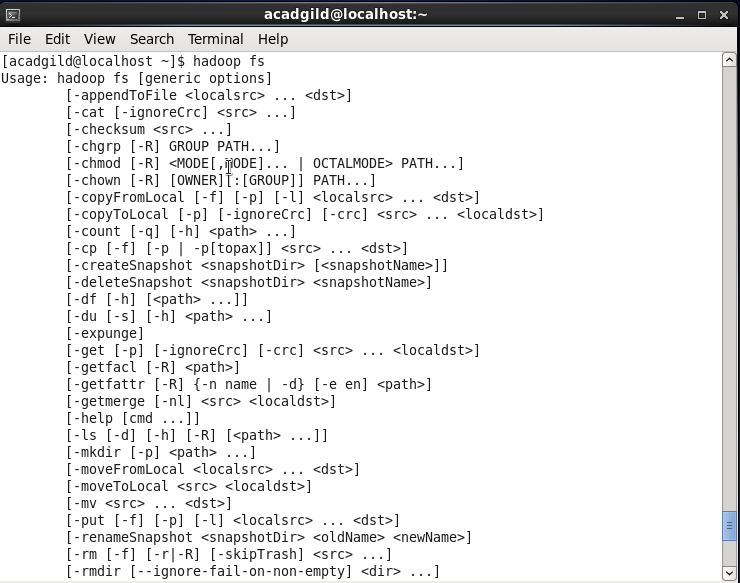
**Size of file in Hadoop**

****

**Helpin Hadoop**

****

**Shell commands in Hadoop**

****