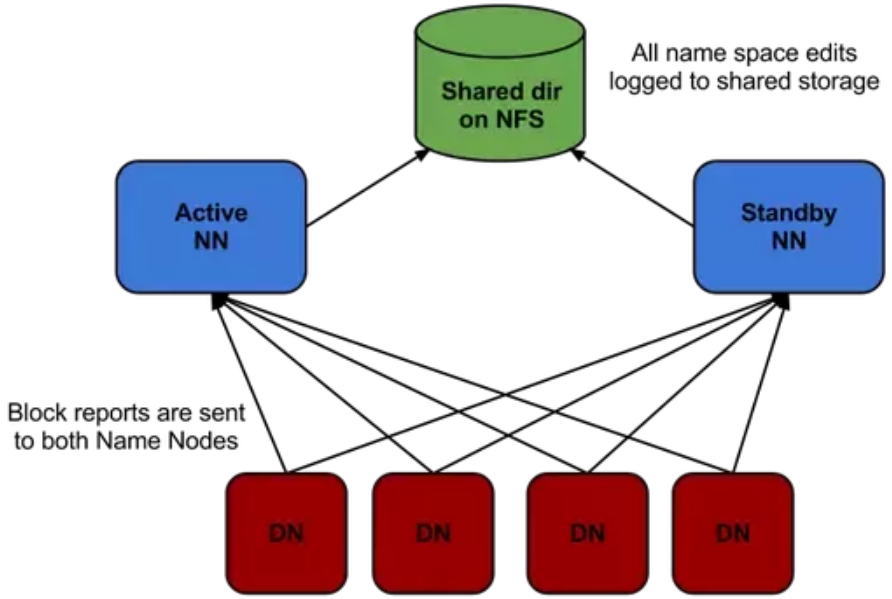

Data
Engineering
and Big Data
Masters
Program

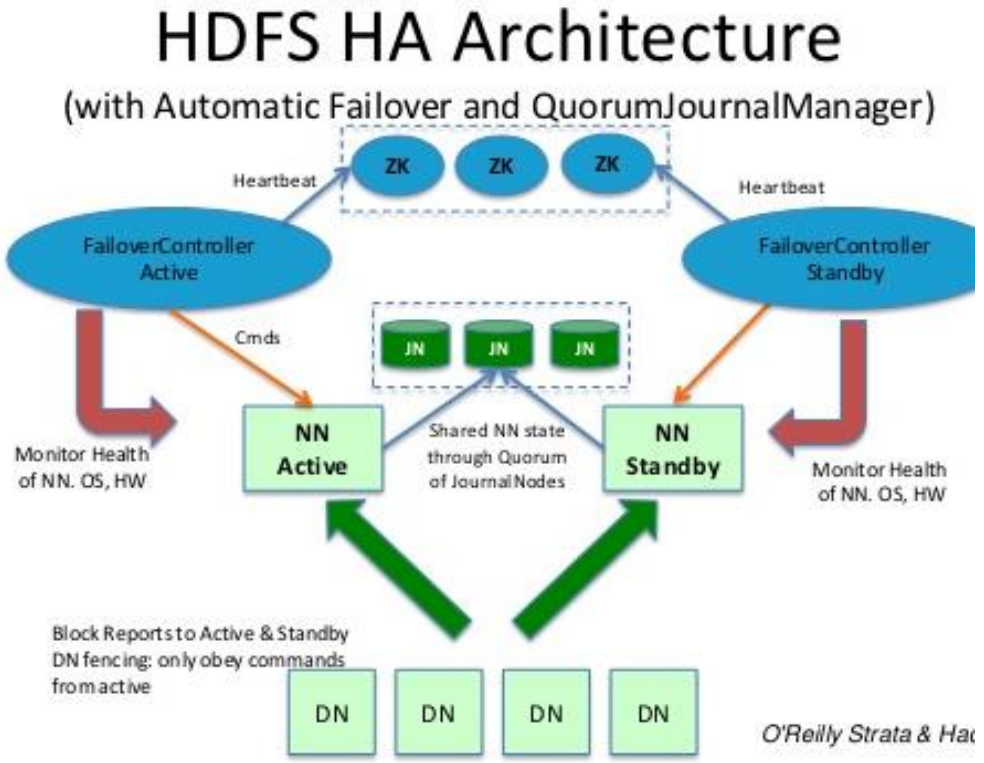
Standby Name node in Hadoop 2

- Secondary Namenode is optional now & Standby Namenode has been to used for failover process.
- Standby NameNode will stay up-to-date with all the file system changes the Active NameNode makes .
- Standby Namenode came into picture. The standby namenode is the node that removes the problem of SPOF (Single Point Of Failure) that was there in Hadoop 1.x. The standby namenode provides automatic failover in case Active Namenode (can be simply called 'Namenode' if HA is not enabled) fails.



HDFS High Availability

- HDFS High availability is possible with two options : NFS and Quorum Journal Manager but Quorum Journal Manager is preferred option.
- When any namespace modification is performed by the Active node, it durably logs a record of the modification to a majority of these JNs. The Standby node reads these edits from the JNs and apply to its own name space.
- In the event of a failover, the Standby will ensure that it has read all of the edits from the JournalNodes before promoting itself to the Active state. This ensures that the namespace state is fully synchronized before a failover occurs.
- Name Node is Daemon & Failover controller is a Daemon. If Name Node Daemon fails, Failover controller Daemon detects and takes corrective action. Even if entire machine crashes, ZooKeeper server detects it and lock will be expired and other Standby name node will be elected as Active Name node.



Difference between Hadoop 2 and Hadoop 3

Fault Tolerance

Hadoop 2.x- In this version, replication handles fault tolerance.

Hadoop 3.x- In this version, erasure coding handle fault tolerance.

Data Balancing

Hadoop 2.x- Uses HDFS Balancer for data balancing

Hadoop 3.x- Uses Intra-data node balancer, which is invoked via the HDFS disk balancer CLI.

YARN Timeline Service

Hadoop 2.x- Uses old timeline service which has scalability issues.

Hadoop 3.x- This version improves the timeline service v2. It also improves the scalability and reliability of timeline service.

Compatible File System

Hadoop 2.x- It supports HDFS (Default FS), FTP File system: This also stores all its data on remotely accessible FTP servers. It also supports Amazon S3 (Simple Storage Service) file system Windows Azure Storage Blobs (WASB) file system.

Hadoop 3.x- It supports all the previous one as well as Microsoft Azure Data Lake filesystem.

Running MR Job

Step 1 –Create a file named word_count_data.txt

Step 2 – hdfs dfs -put word_count_data.txt /user/root/

Step 3 – Create the mapper.py and reducer.py files

Step 4 – Run below command to start the job

```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar  \
-input /user/root/word_count_data.txt  \
-output /user/root/wcoutputnewsaurav  \
-mapper mapper.py \
-reducer reducer.py \
-file /root/code/mapper.py  \
-file /root/code/reducer.py
```

Step 5 – Monitor the job here

http://localhost:8088/cluster