**Q1. Explain what is High availability of Namenode.**

Answer:

1. Hadoop 1.0 NameNode has single point of failure (SPOF) problem- which means that if the NameNode fails, then that Hadoop Cluster will become out-of-the-way.

2. Hadoop 2.0 overcomes this SPOF shortcoming by providing support for multiple NameNodes. It introduces Hadoop 2.0 High Availability feature that brings in an extra NameNode (Passive Standby NameNode) to the Hadoop Architecture which is configured for automatic failover.

3. The main motive of the Hadoop 2.0 High Availability project is to render availability to big data applications 24/7 by deploying 2 Hadoop NameNodes –One in active configuration and the other is the Standby Node in passive configuration.

4. Earlier there was one Hadoop NameNode for maintaining the tree hierarchy of the HDFS files and tracking the data storage in the cluster. Hadoop 2.0 High Availability allows users to configure Hadoop clusters with uncalled- for NameNodes so as to eliminate the probability of SPOF in a given Hadoop cluster.

5. The Hadoop Configuration capability allows users to build clusters horizontally with several NameNodes which can operate autonomously through a common data storage pool, thereby, offering better computing scalability when compared to Hadoop 1.0

6. With Hadoop 2.0, Hadoop architecture is now configured in a manner that it supports automated failover with complete stack resiliency and a hot Standby NameNode.

**2. Explain what is check pointing and how it is useful.**

Answer:

1. A typical edit ranges from 10s to 100s of bytes, but over time enough edits can accumulate to become unwieldy.

2. A couple of problems can arise from these large edit logs. In extreme cases, it can fill up all the available disk capacity on a node, but more subtly, a large edit log can substantially delay NameNode startup as the NameNode reapplies all the edits. This is where checkpointing comes in.

3. Checkpointing is a process that takes an fsimage and edit log and compacts them into a new fsimage.

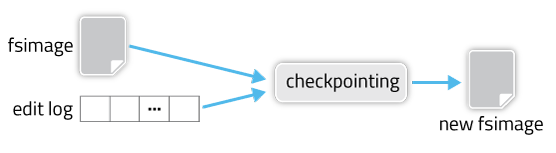
4. This way, instead of replaying a potentially unbounded edit log, the NameNode can load the final in-memory state directly from the fsimage.

5. This is a far more efficient operation and reduces NameNode startup time.

6. However, creating a new fsimage is an I/O- and CPU-intensive operation, sometimes taking minutes to perform.

7. During a checkpoint, the namesystem also needs to restrict concurrent access from other users. So, rather than pausing the active NameNode to perform a checkpoint, HDFS defers it to either the SecondaryNameNode or Standby NameNode, depending on whether NameNode high-availability is configured.

8. The mechanics of checkpointing differs depending on if NameNode high-availability is configured.



**Q3. Explain what is HDFS federation.**

Answer:

1. Hadoop federation allows scaling the name service horizontally.

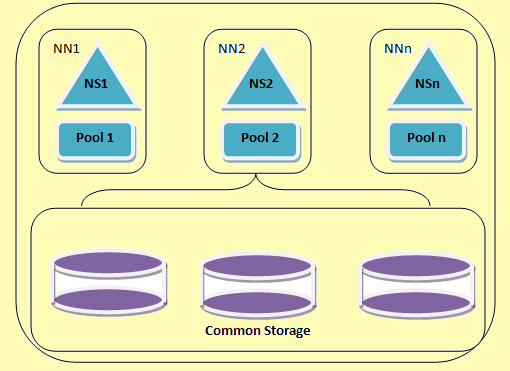
2. It uses several namenodes or namespaces which are independent of each other. These independent namenodes are ***federated*** i.e. they don’t require ***inter coordination***.

3. The datanodes are used as common storage by all the namenodes. Each datanode is registered with all the namenodes in the cluster. These datanodes send periodic reports and responds to the commands from the name nodes.

4. We have a block pool which is a set of blocks that belong to a single namespace.

5. In a cluster, the datanodes stores blocks for all the block pools. Each block pool is managed independently. This enables the name space to generate block ids for new blocks without informing other namespaces. If one namenode fails for any reason, the datanode keeps on serving from other namenodes.

6. One namespace and its block are collectively called ***Namespace Volume***. When a namespace or a namenode is deleted the corresponding block pool at the datanode is deleted automatically. In the process of cluster up-gradation, each namespace volume is upgraded as a unit.



**Q4. What are the configuration files that are to be edited for sure while installing a hadoop cluster.**

Answer:

The four files that need to be configured explicitly while setting up a single node hadoop cluster are:

* Core-site.xml
* HDFS-site.xml
* YARN-site.xml
* Mapred-site.xml