The **ZKFailoverController** (ZKFC) is a new component which is a ZooKeeper client which also monitors and manages the state of the NameNode. Each of the machines which runs a **NameNode** also runs a ZKFC, and that ZKFC is responsible for:

**Health monitoring** - the ZKFC pings its local NameNode on a periodic basis with a health-check command. So long as the NameNode responds in a timely fashion with a healthy status, the ZKFC considers the node healthy. If the node has crashed, frozen, or otherwise entered an unhealthy state, the health monitor will mark it as unhealthy.

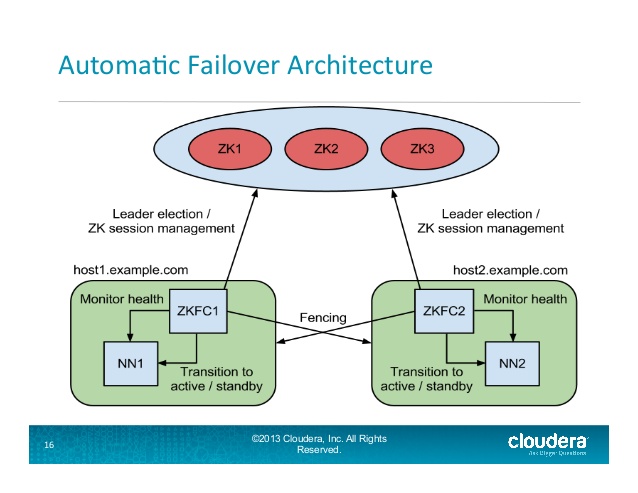
**ZooKeeper session management** - when the local NameNode is healthy, the ZKFC holds a session open in ZooKeeper. If the local NameNode is active, it also holds a special "**lock**" znode. This lock uses ZooKeeper's support for "**ephemeral**" nodes; if the session expires, the lock node will be automatically deleted.

**ZooKeeper-based election** - if the local NameNode is healthy, and the ZKFC sees that no other node currently holds the lock znode, it will itself try to acquire the lock. If it succeeds, then it has "**won the election**", and is responsible for running a failover to make its local NameNode active.

Have a look at this [Apache PDF](https://issues.apache.org/jira/secure/attachment/12519914/zkfc-design.pdf) which is part of **HDFS-2185** JIRA issue

Slide 16 from

<http://www.slideshare.net/cloudera/hdfs-update-lipcon-federal-big-data-apache-hadoop-forum>

[](https://i.stack.imgur.com/geLQh.jpg):

**Automatic Namenode failover process in Hadoop:**

In a typical HA cluster, two separate machines are configured as NameNodes. At any point in time, exactly one of the NameNodes is in an Active state, and the other is in a Standby state. The Active NameNode is responsible for all client operations in the cluster, while the Standby is simply acting as a slave, maintaining enough state to provide a fast failover if necessary.

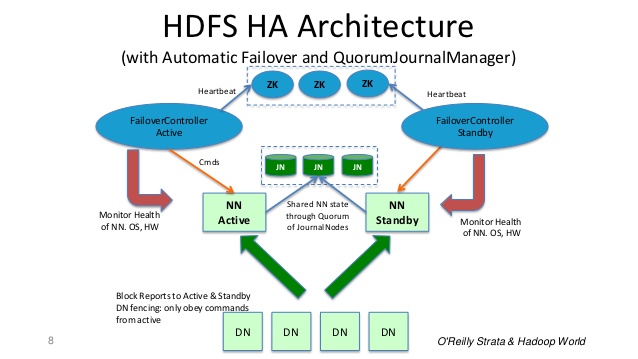
In order for the Standby Namenode to keep its state synchronized with the Active Namenode, both nodes communicate with a group of separate daemons called **JournalNodes** (JNs).

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 is 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 JounalNodes before promoting itself to the Active state. This ensures that the namespace state is fully synchronized before a failover occurs.

It is vital for an HA cluster that only one of the NameNodes is Active at a time. ZooKeeper has been used to avoid split brain scenario so that name node state is not getting diverged due to failover.

Slide 8 from : <http://www.slideshare.net/cloudera/hdfs-futures-world2012-widescreen>

[](https://i.stack.imgur.com/k46RU.jpg):

**In Summary:** 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.

1. [Zookeeper](https://zookeeper.apache.org/)

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications

Role of Zookeeper in Hadoop ecosystem:

During the Hadoop Namenode failover process, ZooKeeper has been used to avoid split brain scenario so that name node state is not getting diverged due to failover.

Refer to this post for more details:

[How does Hadoop Namenode failover process works?](https://stackoverflow.com/questions/33311585/how-does-hadoop-namenode-failover-process-works/33313804#33313804)

[2. JournalNode](https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HDFSHighAvailabilityWithQJM.html) ( Used in Namenode failover process)

In order for the Standby node to keep its state synchronized with the Active node, both nodes communicate with a group of separate daemons called “JournalNodes” (JNs).

**JournalNode machines** - the machines on which you run the JournalNodes. The JournalNode daemon is relatively lightweight, so these daemons may reasonably be collocated on machines with other Hadoop daemons, for example **NameNodes**, the **JobTracker**, or the **YARN ResourceManager**.

**Note:** There must be at least 3 **JournalNode** daemons, since edit log modifications must be written to a majority of JNs. This will allow the system to tolerate the failure of a single machine

3.**Quorum Journal Manager (QJM)** allows to share edit logs between the Active and Standby NameNodes

Importantly, when using the **Quorum Journal Manager**, only one **NameNode** will ever be allowed to write to the **JournalNodes**, so there is no potential for corrupting the file system metadata from a split-brain scenario

Hadoop NameNode: Active and StandBy namenode co-ordination As we know that in Hadoop cluster to avoid single point of failure of NameNode. We must have at least two NamNode should be configured. One node as an active NameNode and another will be in StandBy mode. Zookeeper leader selection mechanism will help, which node will be master node and which node will remain in standby mode. As soon as active name node goes down. Zookeeper failovercontroller will be notified and within less than a second stand by NameNode will become active NameNode and as an administrator you can handle failed NameNode to bring it up.

Znode

Znode is just a file that contain information which needs to be shared across the nodes in a group (e.g group of NameNodes, group of message processing nodes). Any node from the group can watch this znode to monitor the change in information stored in it.

Quorum

Quorum means, there is certainly majority required.

Journal Node: It is a separate daemon process (this is very lightweight process, hence can be setup on master nodes like NameNode, ResourceManager etc

Quorum Journal Manager make sure that any changes needs to be written on majority or the Journal Nodes e.g. (2 out of 3) before information success commit. It will also make sure that only once of the NameNode will serve request to avoid any split-brain scenario.