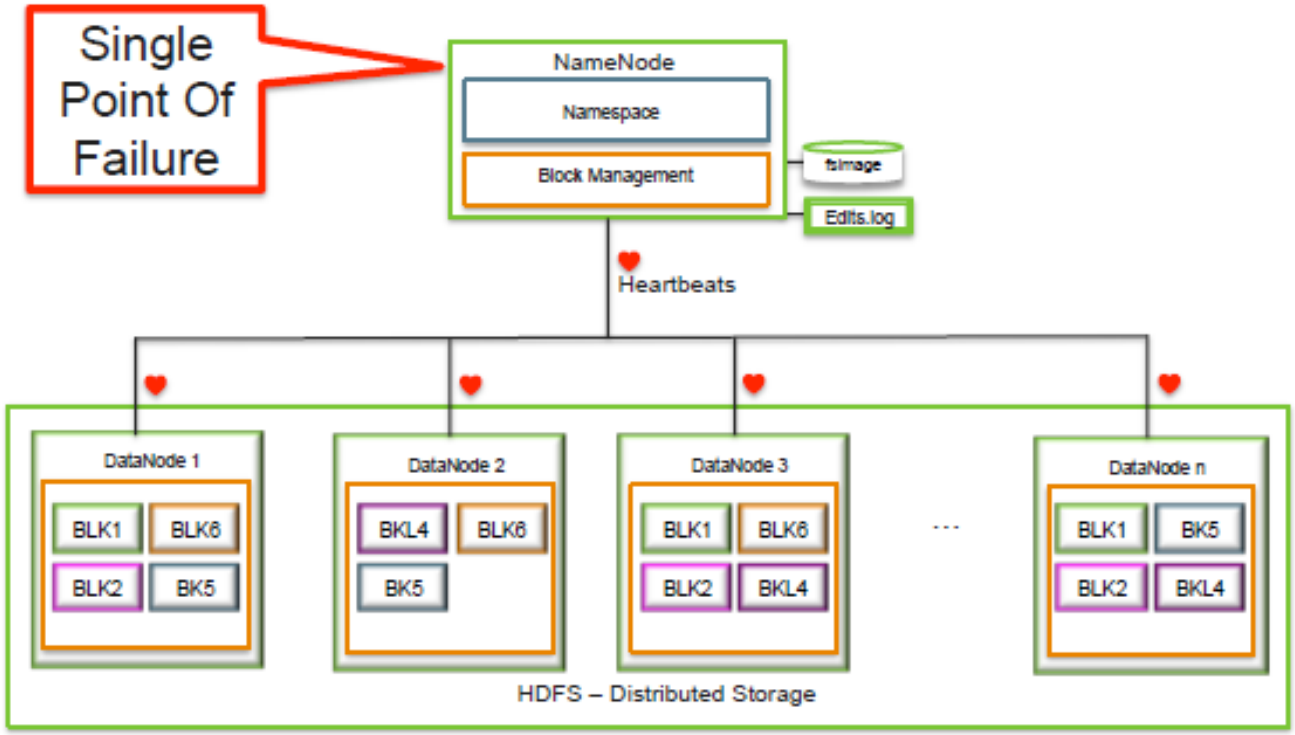
Namenode HA

NameNode Architecture



NameNode High Availability

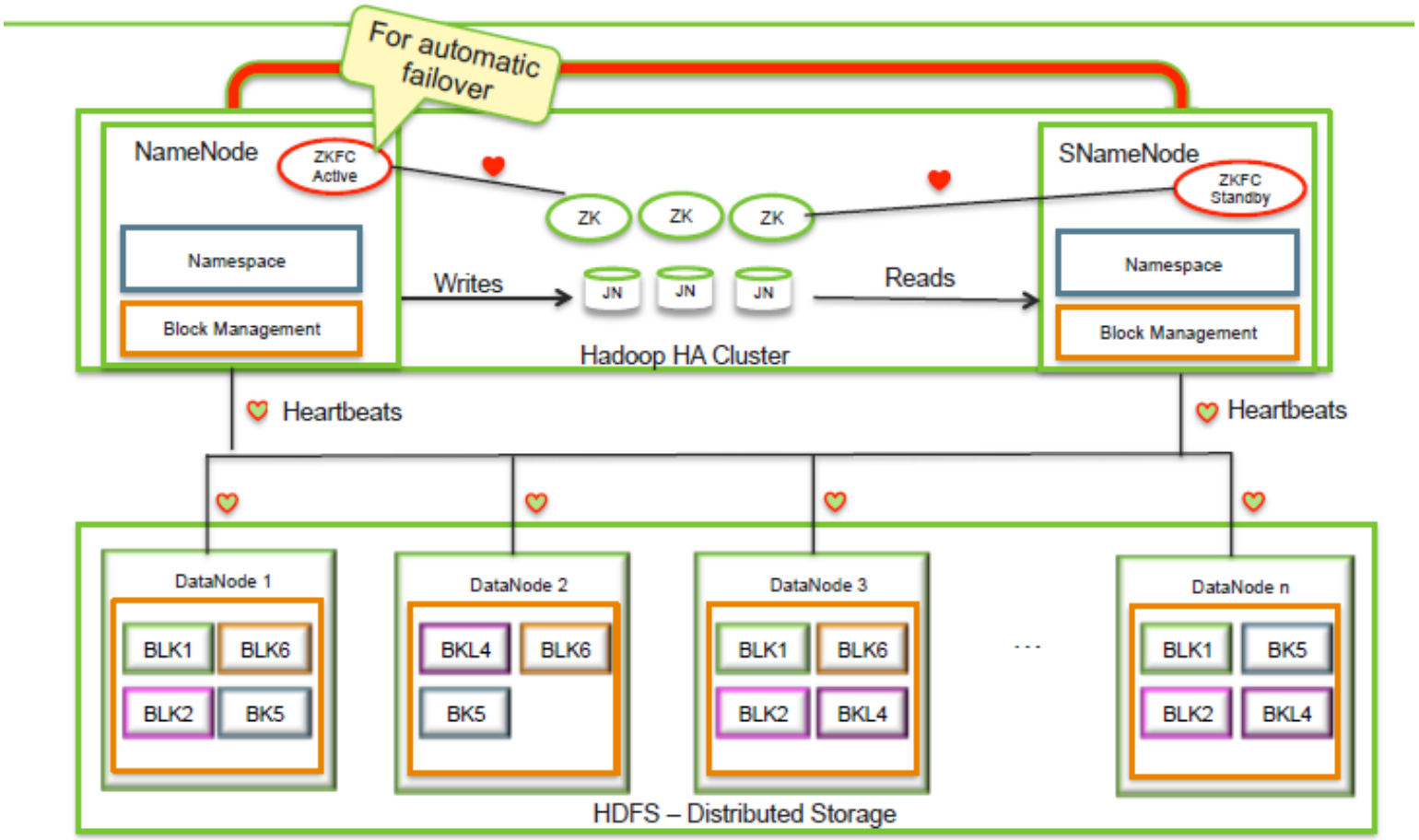
- **Supports manual and automatic failover**
- **Automatic failover with Failover Controller**
  - Active NN election and failure detection using Zookeeper
  - Period NN health check
  - Failover on NN failure
- **Removed shared storage dependency**
  - Quorum Journal Manager
  - 3 to 5 Journal Nodes for storing edit.log
  - Edits must be written to quorum number of Journal Nodes

HDFS HA Components

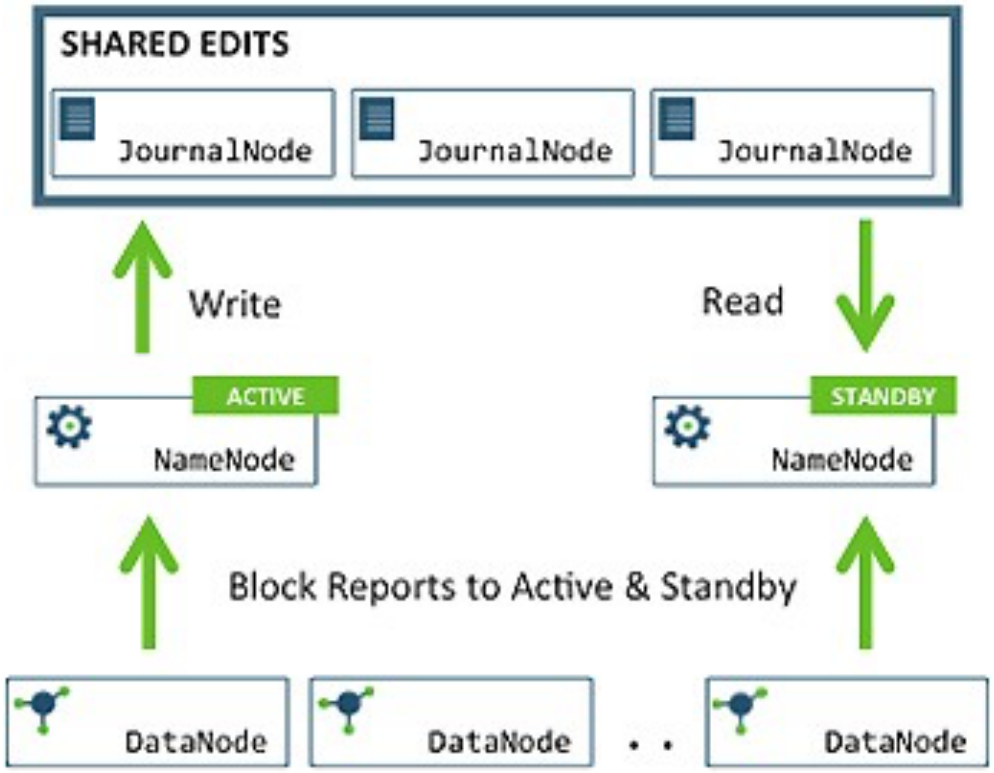
The HDFS HA architecture contains:

- Primary NameNode (active)
- Standby NameNode (passive)
- Journal Nodes: maintains HA state
  - Primary NameNode writes to Journal Nodes (JNs)
  - Standby NameNode reads from Journal Nodes to maintain state
- For Automatic Failover:
  - ZooKeeper Servers - maintain coordination service
  - ZooKeeper Failover Controllers

Understanding NameNode HA



HA cluster


SHARED EDITS
JournalNode
JournalNode
JournalNode
Write
Read
ACTIVE
NameNode
STANDBY
NameNode
Block Reports to Active & Standby
DataNode
DataNode
DataNode

NameNodes in HA

- **The active NameNode is responsible for all operations in its namespace**
- **The Standby NameNode maintains state so it can become active during a failover**
  - Performs checkpointing. A secondary NameNode is not required if running a Standby NameNode
- **DataNodes sent block reports to both the active and the standby NameNodes**

Failover Modes

NameNode HA supports two types of failover:
– Manual – Performed by an administrator
– Automatic – Occurs when there is an issue with the active NameNode

• A manual failover can occur when an administrator wants to test failover or perform maintenance on the active NameNode

• An automatic failover will occur when the HA infrastructure details a situation that requires a failover operation.
Automatic failover requires the ZooKeeper coordination process and a ZooKeeper FailOverController.

NameNode Architectures

supports different types of NameNode (NN) Architectures:

- Single NN with Secondary NN
- Single NN with Standby NN (non-Federated)
- In recent release:
  - NameNode Federation (without HA)
  - NameNode Federation (HA)
  - Individual Federated NN can have HA

hdfs haadmin Command

The ***hdfs haadmin*** command has options for managing a HDFS HA cluster

- **Usage: dfs ha**admin **[-ns <nameserviceId>]**

[-transitionToActive <serviceId>]
[-transitionToStandby <serviceId>]
[-failover [--forcefence] [--forceactive] <serviceId> <serviceId>]
[-checkHealth <serviceId>]
[-help <command>]

Configuring the NameNode HA Cluster

hdfs-site.xml

logical name : used for both configuration

```
<property>
      <name>dfs.nameservices</name>
      <value>mycluster</value>
      <description>Logical name for this new nameservice</description>
</property>
```

dfs.ha.namenodes.[$nameservice ID] –
to determine all the NameNodes in the cluster

```
<property>
      <name>dfs.ha.namenodes.mycluster</name>
      <value>nn1,nn2,nn3</value>
      <description>Unique identifiers for each NameNode in the
        nameservice</description>
</property>
```

Configuring the NameNode HA Cluster

dfs.namenode.rpc-address.[$nameservice ID].[$name node ID] - specify the fully-qualified RPC address for each NameNode to listen on.

```
<property>
 <name>dfs.namenode.rpc-address.mycluster.nn1</name>
 <value>machine1.example.com:8020</value>
</property>

<property>
 <name>dfs.namenode.rpc-address.mycluster.nn2</name>
 <value>machine2.example.com:8020</value>
</property>

<property>
  <name>dfs.namenode.rpc-address.mycluster.nn3</name>
  <value>machine3.example.com:9820</value>
</property>
```

Configuring the NameNode HA Cluster

Specify the fully-qualified HTTP address for each NameNode to listen on.

```
<property>
 <name>dfs.namenode.http-address.mycluster.nn1</name>
 <value>machine1.example.com:9870</value>
</property>

<property>
 <name>dfs.namenode.http-address.mycluster.nn2</name>
 <value>machine2.example.com:9870</value>
</property>

<property>
 <name>dfs.namenode.http-address.mycluster.nn3</name>
 <value>machine3.example.com:9870</value>
</property>
```

Configuring the NameNode HA Cluster

dfs.namenode.shared.edits.dir : to specify the URI that identifies a group of JournalNodes (JNs) where the NameNode will write/read edits.

```
<property>
 <name>dfs.namenode.shared.edits.dir</name>
 <value>qjournal://node1.example.com:8485;node2.example.com:
    8485;node3.example.com:8485/mycluster</value>
</property>
```

Configuring the NameNode HA Cluster

dfs.client.failover.proxy.provider.[$nameservice ID] : Java class to determine which NameNode is the current Active

```
<property>
 <name>dfs.client.failover.proxy.provider.mycluster</name>
 <value>org.apache.hadoop.hdfs.server.namenode.ha.
     ConfiguredFailoverProxyProvider</value>
</property>
```

Configuring the NameNode HA Cluster

dfs.ha.fencing.methods : to fence the Active NameNode during a failover.

```
<property>
 <name>dfs.ha.fencing.methods</name>
 <value>sshfence</value>
</property>

<property>
 <name>dfs.ha.fencing.ssh.private-key-files</name>
 <value>/home/exampleuser/.ssh/id_rsa</value>
</property>
```

Configuring the NameNode HA Cluster

fs.defaultFS :The default path prefix used by the Hadoop FS client

core-site.xml

```
<property>
 <name>fs.defaultFS</name>
 <value>hdfs://mycluster</value>
</property>
```

Configuring the NameNode HA Cluster

dfs.journalnode.edits.dir - absolute path on the JournalNode machines where the edits and other local state (used by the JNs) will be stored

```
<property>
 <name>dfs.journalnode.edits.dir</name>
 <value>/path/to/journal/node/local/data</value>
</property>
```

NameNode HA cluster

Steps:

- initialize JournalNodes
- run the required configurations for HA on the NameNodes
- validate the HA configuration.

Deploying a NameNode HA Cluster

Start the JournalNode daemons on those set of machines where the JNs are deployed

su –l hdfs –c "/usr/hdp/current/hadoop-hdfs-journalnode/../hadoop/sbin/hdfs-daemon.sh start journalnode"

Initialize JournalNodes.

At the NN1 host machine, execute the following command:

su –l hdfs –c "hdfs namenode -initializeSharedEdits -force"

Initialize HA state in ZooKeeper. Execute the following command on NN1:

hdfs zkfc -formatZK -force

creates a znode in ZooKeeper

Deploying a NameNode HA Cluster ..

start ZooKeeper - on the ZooKeeper host machine(s).

su - zookeeper -c "export ZOOCFGDIR=/usr/hdp/current/zookeeper-server/conf ; export ZOOCFG=zoo.cfg; source /usr/hdp/current/zookeeper-server/conf/zookeeper-env.sh ; /usr/hdp/current/zookeeper-server/bin/zkServer.sh start"

At the standby namenode host, execute the following command:

su -l hdfs -c "hdfs namenode -bootstrapStandby -force"

Start NN1. At the NN1 host machine, execute the following command:

```
su -l hdfs -c "/usr/hdp/current/hadoop-hdfs-namenode/../hadoop/sbin/hdfs-daemon.sh
    start namenode"
```

Deploying a NameNode HA Cluster ..

Format NN2 and copy the latest checkpoint (FSImage) from NN1 to NN2

```
su -l hdfs -c "hdfs namenode -bootstrapStandby -force"
```

Start NN2.

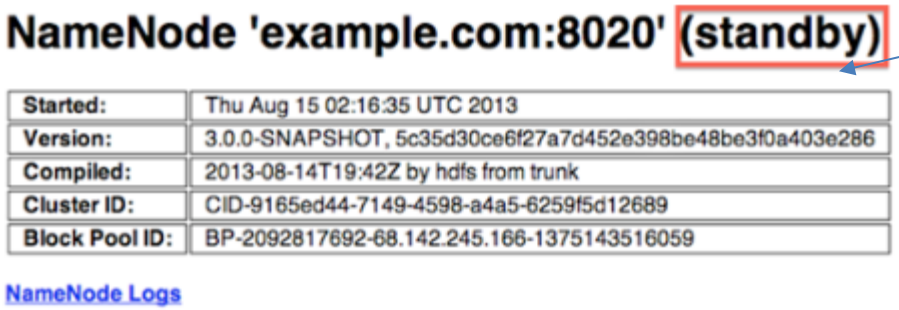
```
su -l hdfs -c "/usr/hdp/current/hadoop-hdfs-namenode/../hadoop/sbin/hadoop-daemon.sh
    start namenode"
```

Start DataNodes. su -l hdfs -c "/usr/hdp/current/hadoop-hdfs-datanode/..
/hadoop/sbin/hadoop-daemon.sh start datanode"

Deploying a NameNode HA Cluster ..

Validate the HA configuration.

The NameNode can be either in "standby" or "active" state.



NameNode 'example.com:8020' (standby)

Started:	Thu Aug 15 02:16:35 UTC 2013
Version:	3.0.0-SNAPSHOT, 5c35d30ce6f27a7d452e398be48be3f0a403e286
Compiled:	2013-08-14T19:42Z by hdfs from trunk
Cluster ID:	CID-9165ed44-7149-4598-a4a5-6259f5d12689
Block Pool ID:	BP-2092817692-68.142.245.166-1375143516059

NameNode Logs

Transition one of the HA NameNodes to the Active state.

hdfs haadmin -failover --forcefence --forceactive <serviceId> <namenodeId>

Execute the command on that NameNode host machine

HDFS High Availability Using the Quorum Journal Manager – 3 Hrs