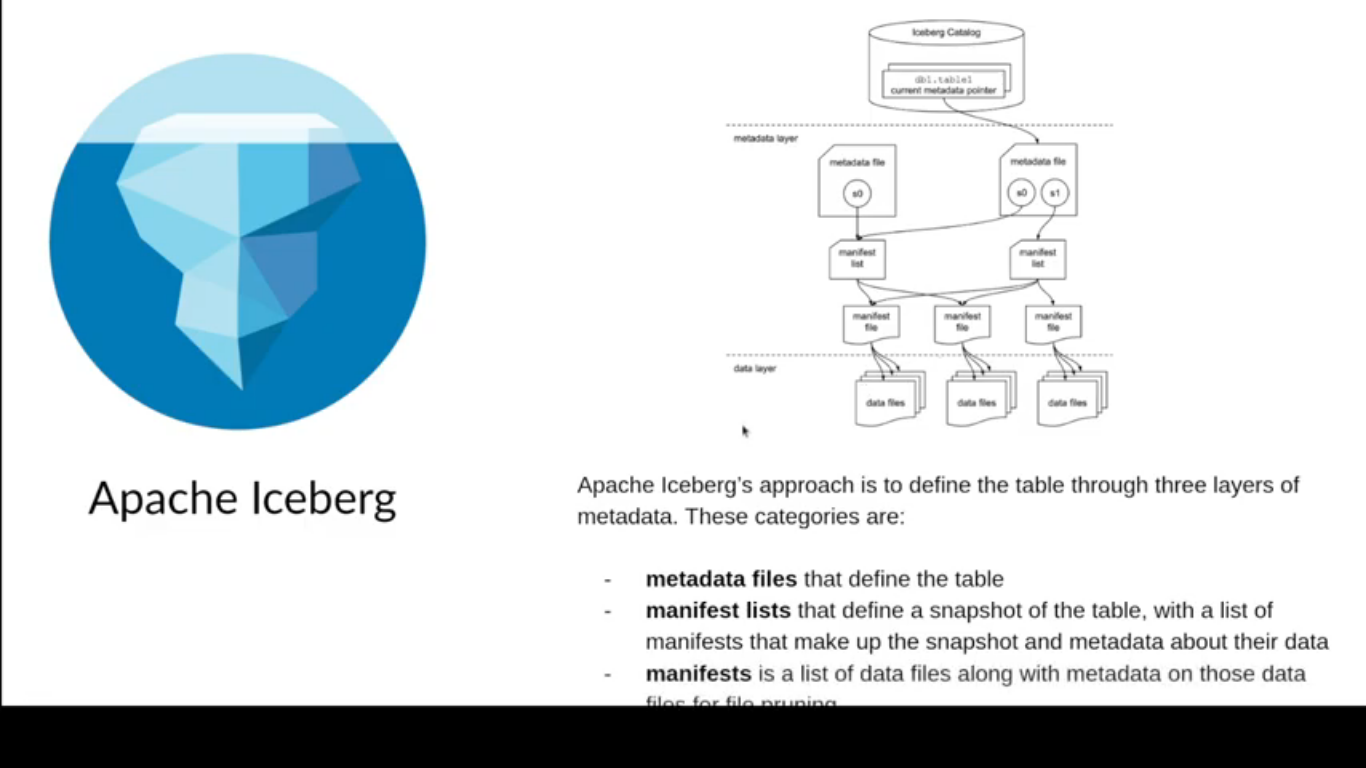
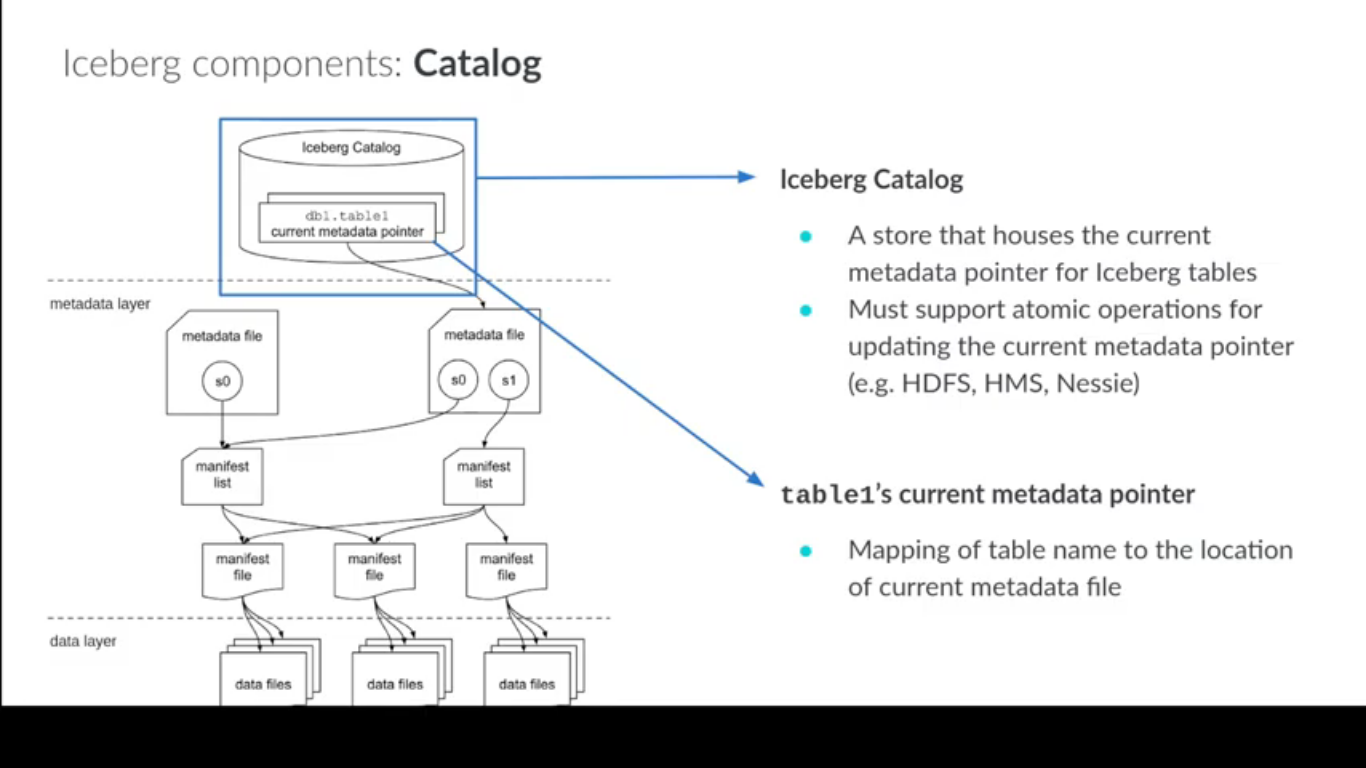
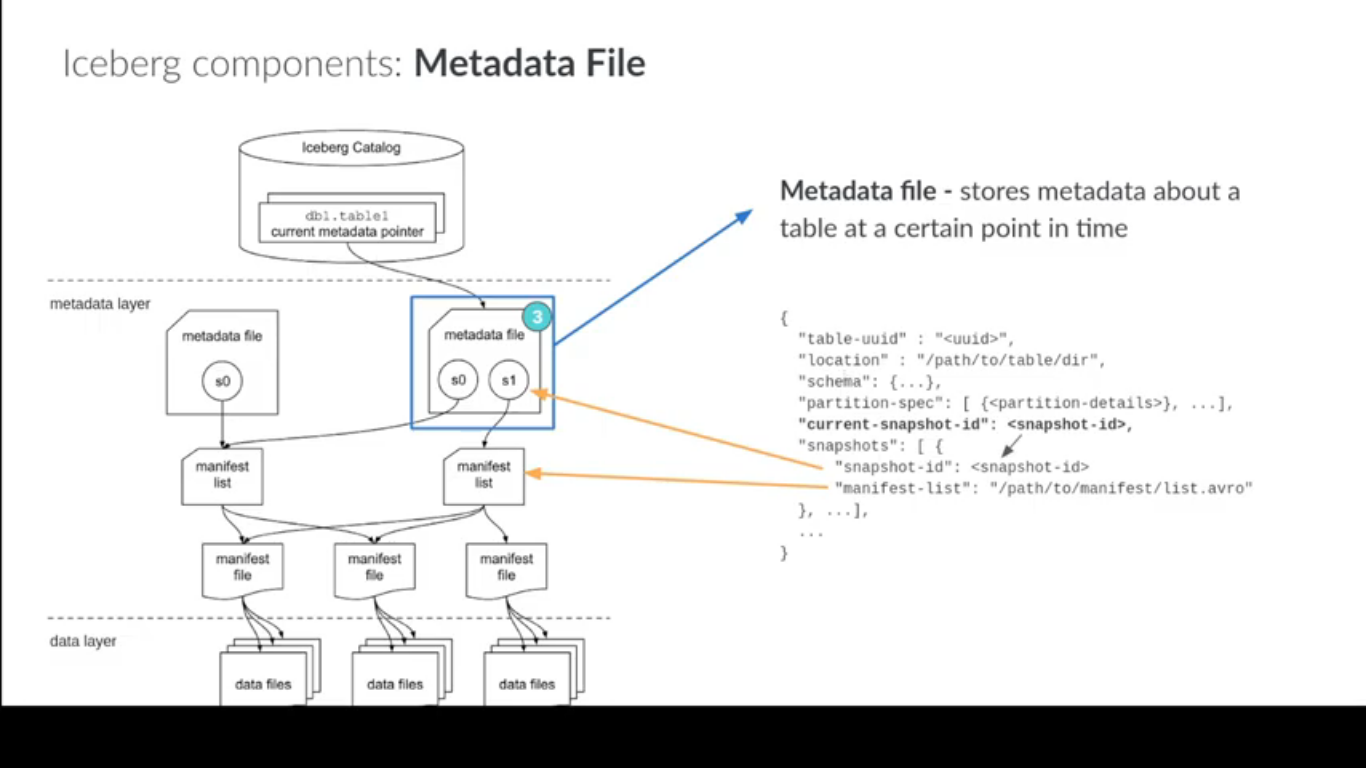
**Apache Iceberg Architecture for data lake**



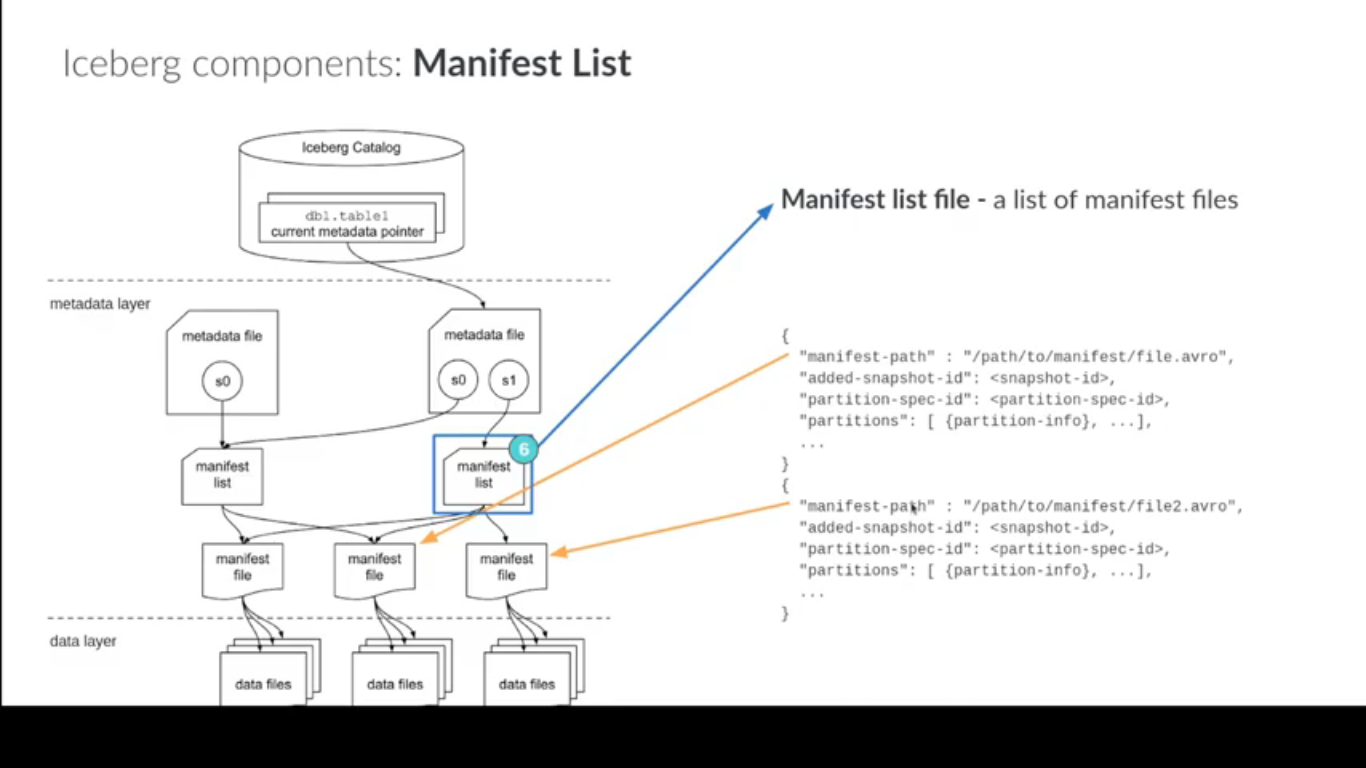
**Catalog (Project Nessie is open source catalog design specifically for iceberg)**

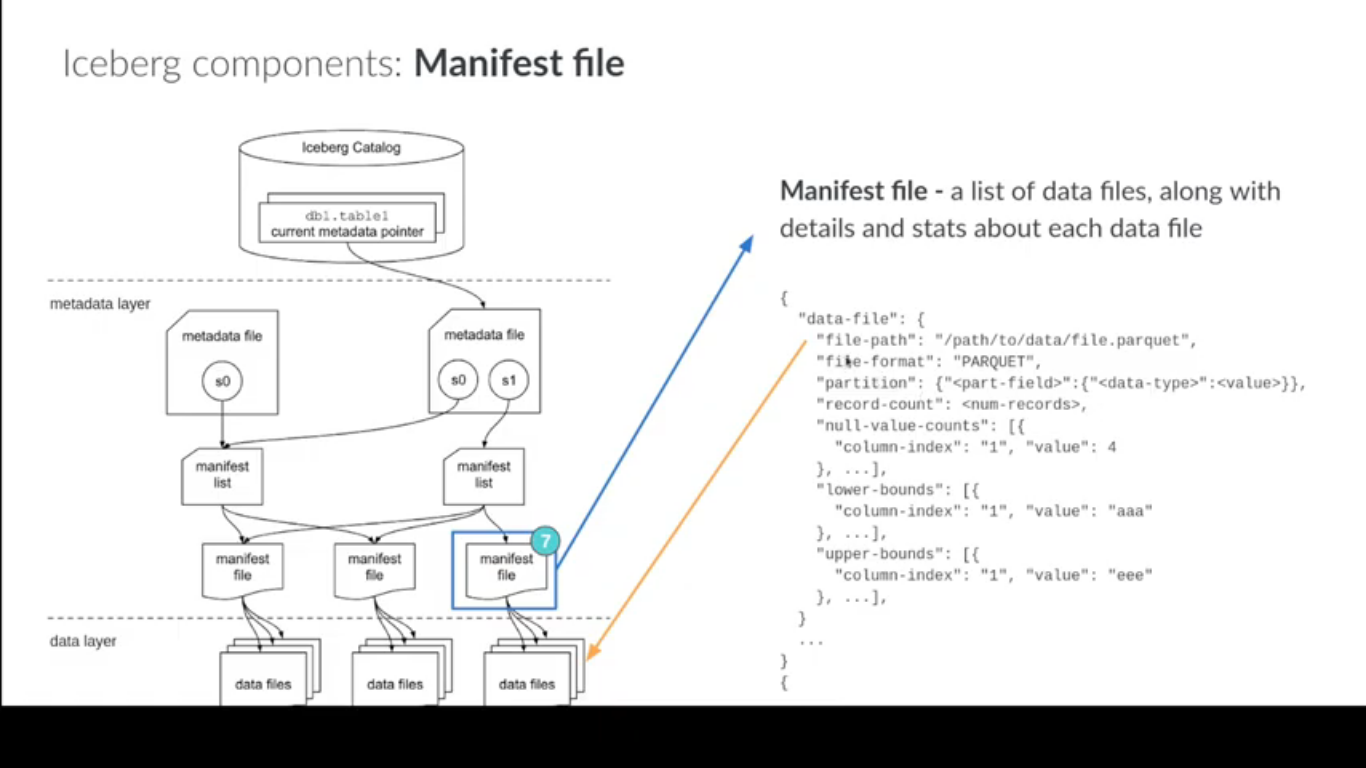


**Metadata File Format:**

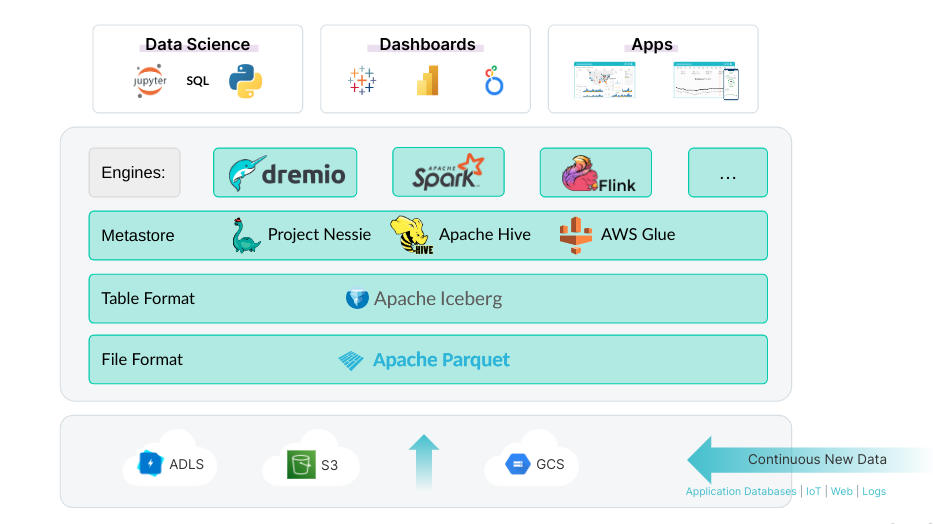


**Manifest List File :**





**Data Lake Architecture**

****

**For detail information about iceberg go through below link**

**https://www.dremio.com/resources/guides/apache-iceberg-an-architectural-look-under-the-covers/**

**Installation of Hadoop**

**Prerequisites:**

Java is the primary requirement for running Hadoop on any system, So make sure you have Java installed on your system using the following command. If you don’t have Java installed on your system, use one of the following links to install it first.

* [How to Install JAVA 8 on CentOS/RHEL 7/6/5](https://tecadmin.net/install-java-8-on-centos-rhel-and-fedora/)

**Create SSH key for user:**

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

chmod 0600 ~/.ssh/authorized\_keys

Let’s verify key based login. Below command should not ask for the password but the first time it will prompt for adding RSA to the list of known hosts.

ssh localhost

exit

**3. Download Hadoop 3.1 Archive**

In this step, download hadoop 3.1 source archive file using below command. You can also select alternate [download mirror](http://www.apache.org/dyn/closer.cgi/hadoop/common/) for increasing download speed.

cd ~

wget http://www-eu.apache.org/dist/hadoop/common/hadoop-3.1.0/hadoop-3.1.0.tar.gz

tar xzf hadoop-3.1.0.tar.gz

mv hadoop-3.1.0 hadoop

## 4. Setup Hadoop Pseudo-Distributed Mode

### 4.1. Setup Hadoop Environment Variables

First, we need to set environment variable uses by Hadoop. Edit **~/.bashrc** file and append following values at end of file.

export HADOOP\_HOME=/home/hadoop/hadoop

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

Now apply the changes in the current running environment

source ~/.bashrc

Now edit **$HADOOP\_HOME/etc/hadoop/hadoop-env.sh** file and set **JAVA\_HOME** environment variable. Change the JAVA path as per install on your system. This path may vary as per your operating system version and installation source. So make sure you are using correct path.

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

### 4.2. Setup Hadoop Configuration Files

Hadoop has many of configuration files, which need to configure as per requirements of your Hadoop infrastructure. Let’s start with the configuration with basic Hadoop single node cluster setup. first, navigate to below location

cd $HADOOP\_HOME/etc/hadoop

#### Edit core-site.xml

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

#### Edit hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.name.dir</name>

<value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>

</property>

<property>

<name>dfs.data.dir</name>

<value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>

</property>

</configuration>

#### Edit mapred-site.xml

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

#### Edit yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

### 4.3. Format Namenode

Now format the namenode using the following command, make sure that Storage directory is

hdfs namenode -format

Sample output:

WARNING: /home/hadoop/hadoop/logs does not exist. Creating.

2018-05-02 17:52:09,678 INFO namenode.NameNode: STARTUP\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STARTUP\_MSG: Starting NameNode

STARTUP\_MSG: host = tecadmin/127.0.1.1

STARTUP\_MSG: args = [-format]

STARTUP\_MSG: version = 3.1.0

...

...

...

2018-05-02 17:52:13,717 INFO common.Storage: Storage directory /home/hadoop/hadoopdata/hdfs/namenode has been successfully formatted.

2018-05-02 17:52:13,806 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hadoop/hadoopdata/hdfs/namenode/current/fsimage.ckpt\_0000000000000000000 using no compression

2018-05-02 17:52:14,161 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoop/hadoopdata/hdfs/namenode/current/fsimage.ckpt\_0000000000000000000 of size 391 bytes saved in 0 seconds .

2018-05-02 17:52:14,224 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0

2018-05-02 17:52:14,282 INFO namenode.NameNode: SHUTDOWN\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SHUTDOWN\_MSG: Shutting down NameNode at tecadmin/127.0.1.1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 5. Start Hadoop Cluster

Let’s start your Hadoop cluster using the scripts provides by Hadoop. Just navigate to your $HADOOP\_HOME/sbin directory and execute scripts one by one.

cd $HADOOP\_HOME/sbin/

Now run **start-dfs.sh** script.

./start-dfs.sh

Sample output:

Starting namenodes on [localhost]

Starting datanodes

Starting secondary namenodes [tecadmin]

2018-05-02 18:00:32,565 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Now run **start-yarn.sh** script.

./start-yarn.sh

Sample output:

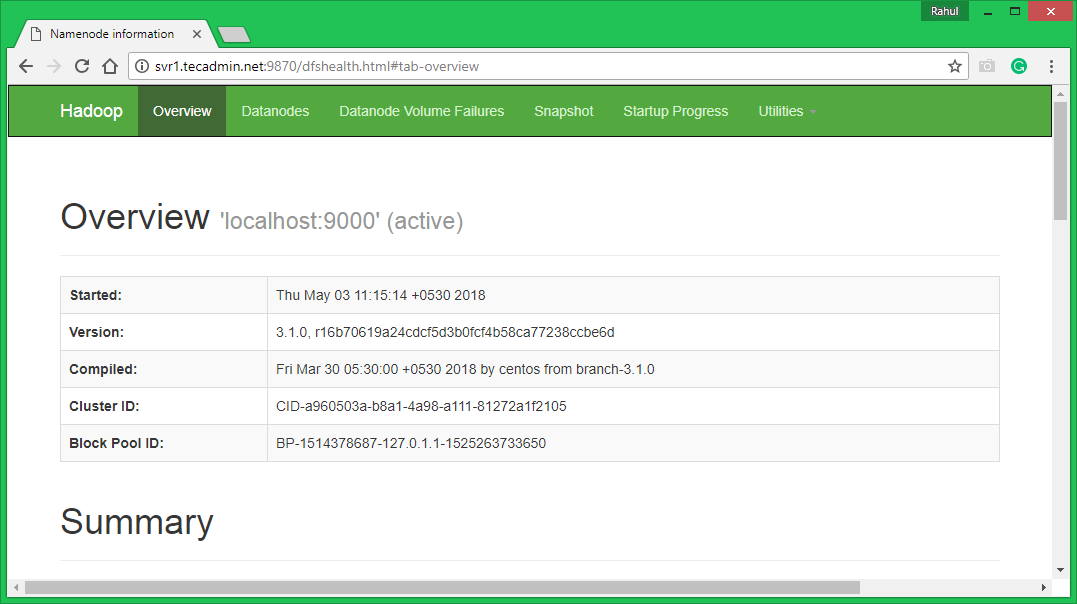
Starting resourcemanager

Starting nodemanagers

## 6. Access Hadoop Services in Browser

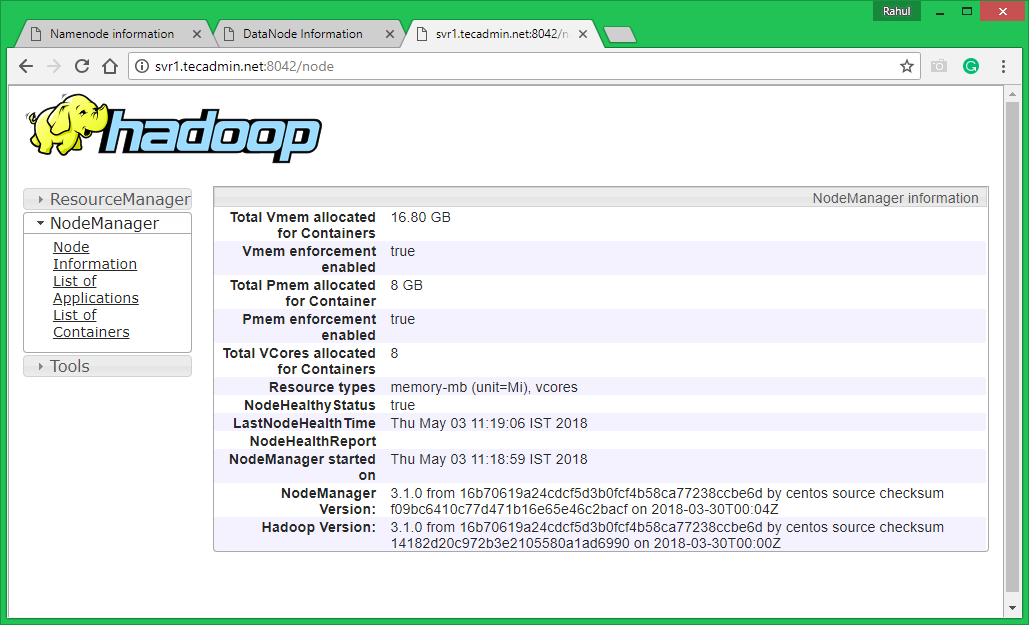
Hadoop NameNode started on port 9870 default. Access your server on port 9870 in your favorite web browser.

http://svr1.tecadmin.net:9870/

[](https://tecadmin.net/wp-content/uploads/2015/11/hadoop-31-dfs.png)

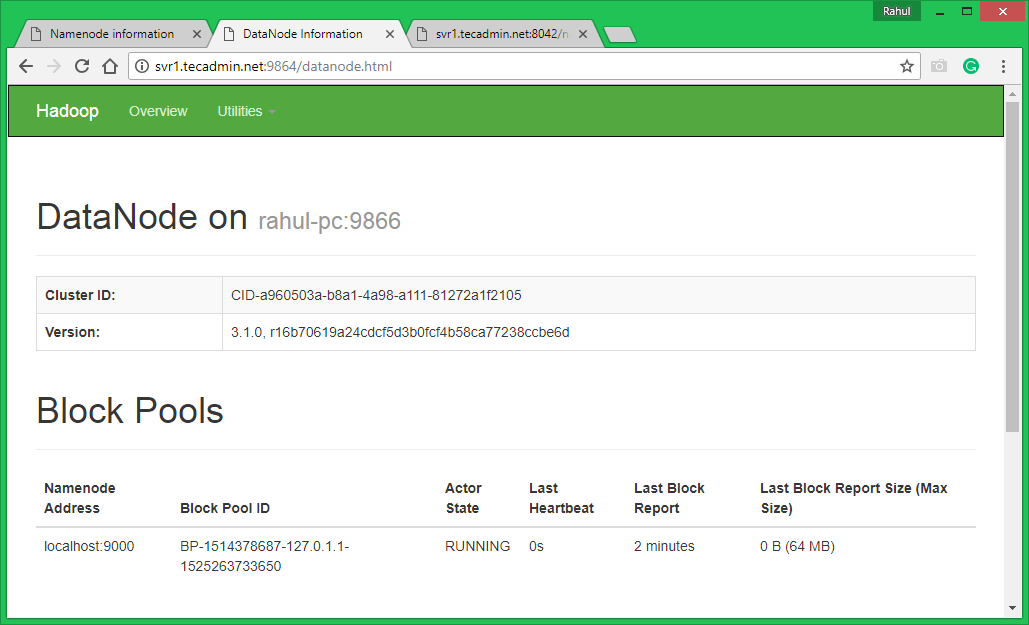
Now access port 8042 for getting the information about the cluster and all applications

http://svr1.tecadmin.net:8042/

[](https://tecadmin.net/wp-content/uploads/2015/11/hadoop-31-nodemanager.png)

Access port 9864 to get details about your Hadoop node.

http://svr1.tecadmin.net:9864/

[](https://tecadmin.net/wp-content/uploads/2015/11/hadoop-31-datanode.png)

## 7. Test Hadoop Single Node Setup

**7.1.** Make the HDFS directories required using following commands.

bin/hdfs dfs -mkdir /user

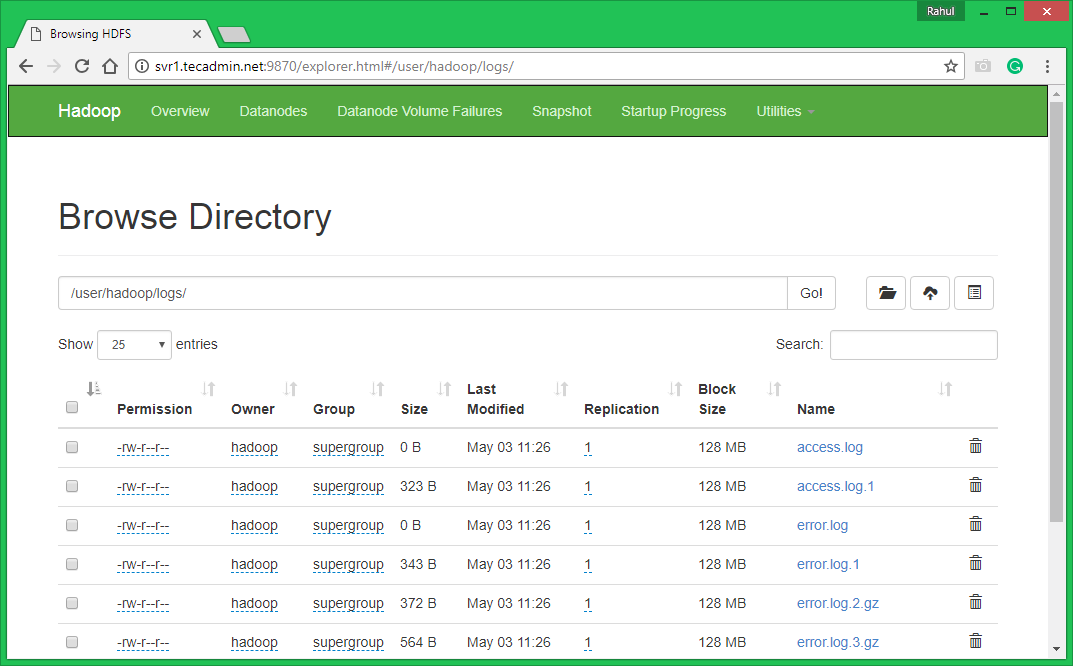
bin/hdfs dfs -mkdir /user/hadoop

**7.2.** Copy all files from local file system /var/log/httpd to hadoop distributed file system using below command

bin/hdfs dfs -put /var/log/apache2 logs

**7.3.** Browse Hadoop distributed file system by opening below URL in the browser. You will see an apache2 folder in the list. Click on the folder name to open and you will find all log files there.

http://svr1.tecadmin.net:9870/explorer.html#/user/hadoop/logs/

[](https://tecadmin.net/wp-content/uploads/2015/11/hadoop-31-list-data.png)

**7.4 –** Now copy logs directory for hadoop distributed file system to local file system.

bin/hdfs dfs -get logs /tmp/logs

ls -l /tmp/logs/

You can also check [this tutorial](https://tecadmin.net/hadoop-running-a-wordcount-mapreduce-example/) to run wordcount mapreduce job example using command line.

**Installation of Hive**

We use hive-2.3.9 in this tutorial. You can download it by visiting the following link <http://apache.petsads.us/hive/hive-2.3.9/.> Let us assume it gets downloaded onto the /Downloads directory. Here, we download Hive archive named “apache-hive-2.3.9-bin.tar.gz” for this tutorial. The following command is used to verify the download:

$ cd Downloads

$ ls

On successful download, you get to see the following response:

apache-hive-2.3.9-bin.tar.gz

## Installing Hive

The following steps are required for installing Hive on your system. Let us assume the Hive archive is downloaded onto the /Downloads directory.

### **Extracting and verifying Hive Archive**

The following command is used to verify the download and extract the hive archive:

$ tar zxvf apache-hive-2.3.9-bin.tar.gz

$ ls

### **Setting up environment for Hive**

You can set up the Hive environment by appending the following lines to **~/.bashrc** file:

export HIVE\_HOME=/usr/local/hive

export PATH=$PATH:$HIVE\_HOME/bin

export CLASSPATH=$CLASSPATH:/usr/local/Hadoop/lib/\*:.

export CLASSPATH=$CLASSPATH:/usr/local/hive/lib/\*:.

The following command is used to execute ~/.bashrc file.

$ source ~/.bashrc

## Step 5: Configuring Hive

To configure Hive with Hadoop, you need to edit the **hive-env.sh** file, which is placed in the **$HIVE\_HOME/conf** directory. The following commands redirect to Hive **config** folder and copy the template file:

$ cd $HIVE\_HOME/conf

$ cp hive-env.sh.template hive-env.sh

Edit the **hive-env.sh** file by appending the following line:

export HADOOP\_HOME=/usr/local/hadoop

Hive installation is completed successfully. Now you require an external database server to configure Metastore. We use Apache Derby database.

## Configuring Metastore of Hive

Configuring Metastore means specifying to Hive where the database is stored. You can do this by editing the hive-site.xml file, which is in the $HIVE\_HOME/conf directory. First of all, copy the template file using the following command:

$ cd $HIVE\_HOME/conf

$ cp hive-default.xml.template hive-site.xml

Edit **hive-site.xml** and append the following lines between the <configuration> and </configuration> tags:

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:derby://localhost:1527/metastore\_db;create=true </value>(your database url)

<description>JDBC connect string for a JDBC metastore </description>

</property>

**Follow below link for iceberg with hive catalog**

**https://iceberg.apache.org/docs/1.1.0/hive/**