HCatalog

tutorialspoint
SIMPLYEASYLEARNING

www.tutorialspoint.com

About the Tutorial

HCatalog is a table storage management tool for Hadoop that exposes the tabular data of Hive metastore to other Hadoop applications. It enables users with different data processing tools (Pig, MapReduce) to easily write data onto a grid. HCatalog ensures that users don't have to worry about where or in what format their data is stored. This is a small tutorial that explains just the basics of HCatalog and how to use it.

Audience

This tutorial is meant for professionals aspiring to make a career in Big Data Analytics using Hadoop Framework. ETL developers and professionals who are into analytics in general may as well use this tutorial to good effect.

Prerequisites

Before proceeding with this tutorial, you need a basic knowledge of Core Java, Database concepts of SQL, Hadoop File system, and any of Linux operating system flavors.

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

Part 1: HCatalog Basics

1. HCATALOG – INTRODUCTION

What is HCatalog?

HCatalog is a table storage management tool for Hadoop. It exposes the tabular data of Hive metastore to other Hadoop applications. It enables users with different data processing tools (Pig, MapReduce) to easily write data onto a grid. It ensures that users don't have to worry about where or in what format their data is stored.

HCatalog works like a key component of Hive and it enables the users to store their data in any format and any structure.

Why HCatalog?

Enabling right tool for right Job

Hadoop ecosystem contains different tools for data processing such as Hive, Pig, and MapReduce. Although these tools do not require metadata, they can still benefit from it when it is present. Sharing a metadata store also enables users across tools to share data more easily. A workflow where data is loaded and normalized using MapReduce or Pig and then analyzed via Hive is very common. If all these tools share one metastore, then the users of each tool have immediate access to data created with another tool. No loading or transfer steps are required.

Capture processing states to enable sharing

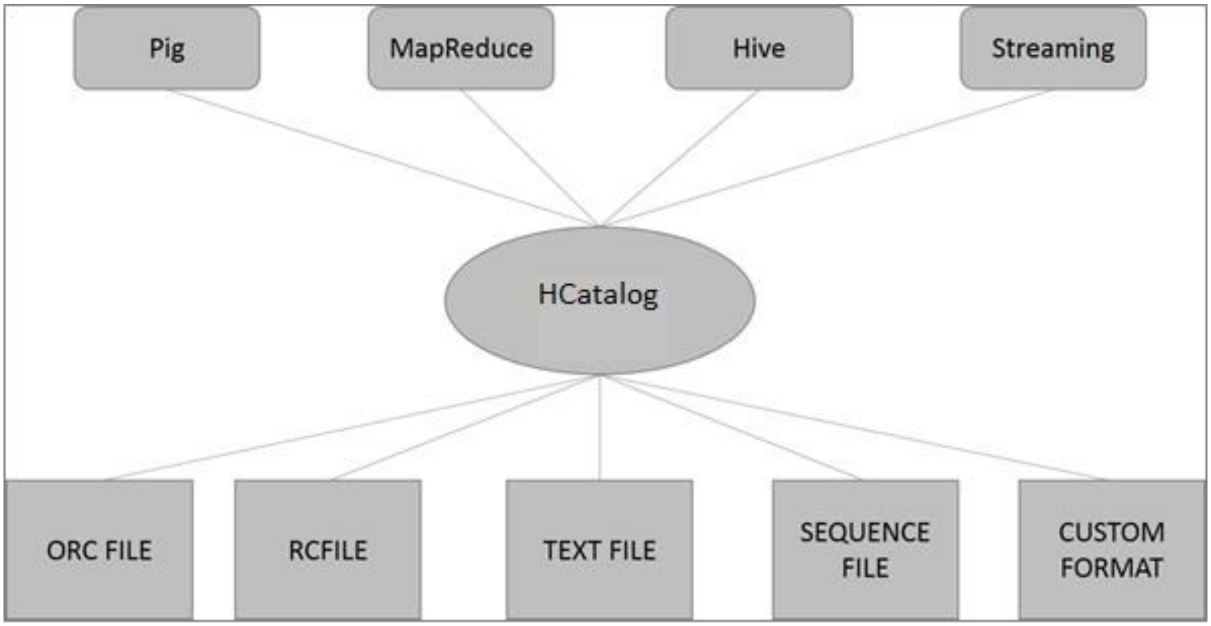
HCatalog can publish your analytics results. So the other programmer can access your analytics platform via "REST". The schemas which are published by you are also useful to other data scientists. The other data scientists use your discoveries as inputs into a subsequent discovery.

Integrate Hadoop with everything

Hadoop as a processing and storage environment opens up a lot of opportunity for the enterprise; however, to fuel adoption, it must work with and augment existing tools. Hadoop should serve as input into your analytics platform or integrate with your operational data stores and web applications. The organization should enjoy the value of Hadoop without having to learn an entirely new toolset. REST services opens up the platform to the enterprise with a familiar API and SQL-like language. Enterprise data management systems use HCatalog to more deeply integrate with the Hadoop platform.

HCatalog Architecture

The following illustration shows the overall architecture of HCatalog.



HCatalog supports reading and writing files in any format for which a **SerDe** (serializer-deserializer) can be written. By default, HCatalog supports RCFile, CSV, JSON, SequenceFile, and ORC file formats. To use a custom format, you must provide the InputFormat, OutputFormat, and SerDe.

HCatalog is built on top of the Hive metastore and incorporates Hive's DDL. HCatalog provides read and write interfaces for Pig and MapReduce and uses Hive's command line interface for issuing data definition and metadata exploration commands.

2. HCATALOG – INSTALLATION

All Hadoop sub-projects such as Hive, Pig, and HBase support Linux operating system. Therefore, you need to install a Linux flavor on your system. HCatalog is merged with Hive Installation on March 26, 2013. From the version Hive-0.11.0 onwards, HCatalog comes with Hive installation. Therefore, follow the steps given below to install Hive which in turn will automatically install HCatalog on your system.

Step 1: Verifying JAVA Installation

Java must be installed on your system before installing Hive. You can use the following command to check whether you have Java already installed on your system:

```
$ java –version
```

If Java is already installed on your system, you get to see the following response:

```
java version "1.7.0_71"
Java(TM) SE Runtime Environment (build 1.7.0_71-b13)
Java HotSpot(TM) Client VM (build 25.0-b02, mixed mode)
```

If you don’t have Java installed on your system, then you need to follow the steps given below.

Step 2: Installing Java

Download Java (JDK <latest version> - X64.tar.gz) by visiting the following link http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html.

Then **jdk-7u71-linux-x64.tar.gz** will be downloaded onto your system.

Generally you will find the downloaded Java file in the Downloads folder. Verify it and extract the **jdk-7u71-linux-x64.gz** file using the following commands.

```
$ cd Downloads/
$ ls
jdk-7u71-linux-x64.gz


$ tar zxf jdk-7u71-linux-x64.gz
$ ls
jdk1.7.0_71   jdk-7u71-linux-x64.gz
```

To make Java available to all the users, you have to move it to the location "/usr/local/". Open root, and type the following commands.

```
$ su
password:
# mv jdk1.7.0_71 /usr/local/
# exit
```

For setting up **PATH** and **JAVA_HOME** variables, add the following commands to **~/.bashrc** file.

```
export JAVA_HOME=/usr/local/jdk1.7.0_71
export PATH=PATH:$JAVA_HOME/bin
```

Now verify the installation using the command **java -version** from the terminal as explained above.

Step 3: Verifying Hadoop Installation

Hadoop must be installed on your system before installing Hive. Let us verify the Hadoop installation using the following command:

```
$ hadoop version
```

If Hadoop is already installed on your system, then you will get the following response:

```
Hadoop 2.4.1
Subversion https://svn.apache.org/repos/asf/hadoop/common -r 1529768
Compiled by hortonmu on 2013-10-07T06:28Z
Compiled with protoc 2.5.0
From source with checksum 79e53ce7994d1628b240f09af91e1af4
```

If Hadoop is not installed on your system, then proceed with the following steps:

Step 4: Downloading Hadoop

Download and extract Hadoop 2.4.1 from Apache Software Foundation using the following commands.

```
$ su
password:
# cd /usr/local
# wget http://apache.claz.org/hadoop/common/hadoop-2.4.1/
hadoop-2.4.1.tar.gz
# tar xzf hadoop-2.4.1.tar.gz
# mv hadoop-2.4.1/* to hadoop/
# exit
```

Step 5: Installing Hadoop in Pseudo Distributed Mode

The following steps are used to install **Hadoop 2.4.1** in pseudo distributed mode.

Setting up Hadoop

You can set Hadoop environment variables by appending the following commands to **~/.bashrc** file.

```
export HADOOP_HOME=/usr/local/hadoop
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 all the changes into the current running system.

```
$ source ~/.bashrc
```

Hadoop Configuration

You can find all the Hadoop configuration files in the location “$HADOOP_HOME/etc/hadoop”. You need to make suitable changes in those configuration files according to your Hadoop infrastructure.

```
$ cd $HADOOP_HOME/etc/hadoop
```

In order to develop Hadoop programs using Java, you have to reset the Java environment variables in **hadoop-env.sh** file by replacing **JAVA_HOME** value with the location of Java in your system.

```
export JAVA_HOME=/usr/local/jdk1.7.0_71
```

Given below are the list of files that you have to edit to configure Hadoop.

core-site.xml

The **core-site.xml** file contains information such as the port number used for Hadoop instance, memory allocated for the file system, memory limit for storing the data, and the size of Read/Write buffers.

Open the core-site.xml and add the following properties in between the <configuration> and </configuration> tags.

```
<configuration>

  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>

</configuration>
```

hdfs-site.xml

The **hdfs-site.xml** file contains information such as the value of replication data, the namenode path, and the datanode path of your local file systems. It means the place where you want to store the Hadoop infrastructure.

Let us assume the following data.

```
dfs.replication (data replication value) = 1


(In the following path /hadoop/ is the user name.
hadoopinfra/hdfs/namenode is the directory created by hdfs file system.)


namenode path = //home/hadoop/hadoopinfra/hdfs/namenode
```

```
(hadoopinfra/hdfs/datanode is the directory created by hdfs file system.)
```

```
datanode path = //home/hadoop/hadoopinfra/hdfs/datanode
```

Open this file and add the following properties in between the <configuration>, </configuration> tags in this file.

```
<configuration>

  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/hadoop/hadoopinfra/hdfs/namenode</value>
  </property>



  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/hadoop/hadoopinfra/hdfs/datanode</value>
  </property>

</configuration>
```

Note: In the above file, all the property values are user-defined and you can make changes according to your Hadoop infrastructure.

yarn-site.xml

This file is used to configure yarn into Hadoop. Open the yarn-site.xml file and add the following properties in between the <configuration>, </configuration> tags in this file.

```
<configuration>

  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

</configuration>
```

mapred-site.xml

This file is used to specify which MapReduce framework we are using. By default, Hadoop contains a template of yarn-site.xml. First of all, you need to copy the file from **mapred-site,xml.template** to **mapred-site.xml** file using the following command.

```
$ cp mapred-site.xml.template mapred-site.xml
```

Open **mapred-site.xml** file and add the following properties in between the <configuration>, </configuration> tags in this file.

```
<configuration>

  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>

</configuration>
```

Step 6: Verifying Hadoop Installation

The following steps are used to verify the Hadoop installation.

Namenode Setup

Set up the namenode using the command “hdfs namenode -format” as follows:

```
$ cd ~
$ hdfs namenode -format
```

The expected result is as follows:

```
10/24/14 21:30:55 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = localhost/192.168.1.11
STARTUP_MSG:   args = [-format]
```

```
STARTUP_MSG:   version = 2.4.1
...
...
10/24/14 21:30:56 INFO common.Storage: Storage directory
/home/hadoop/hadoopinfra/hdfs/namenode has been successfully formatted.
10/24/14 21:30:56 INFO namenode.NNStorageRetentionManager: Going to retain 1
images with txid >= 0
10/24/14 21:30:56 INFO util.ExitUtil: Exiting with status 0

10/24/14 21:30:56 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at localhost/192.168.1.11
************************************************************/
```

Verifying Hadoop DFS

The following command is used to start the DFS. Executing this command will start your Hadoop file system.

```
$ start-dfs.sh
```

The expected output is as follows:

```
10/24/14 21:37:56
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hadoop/hadoop-2.4.1/logs/hadoop-
hadoop-namenode-localhost.out
localhost: starting datanode, logging to /home/hadoop/hadoop-2.4.1/logs/hadoop-
hadoop-datanode-localhost.out
Starting secondary namenodes [0.0.0.0]
```

Verifying Yarn Script

The following command is used to start the Yarn script. Executing this command will start your Yarn daemons.

```
$ start-yarn.sh
```

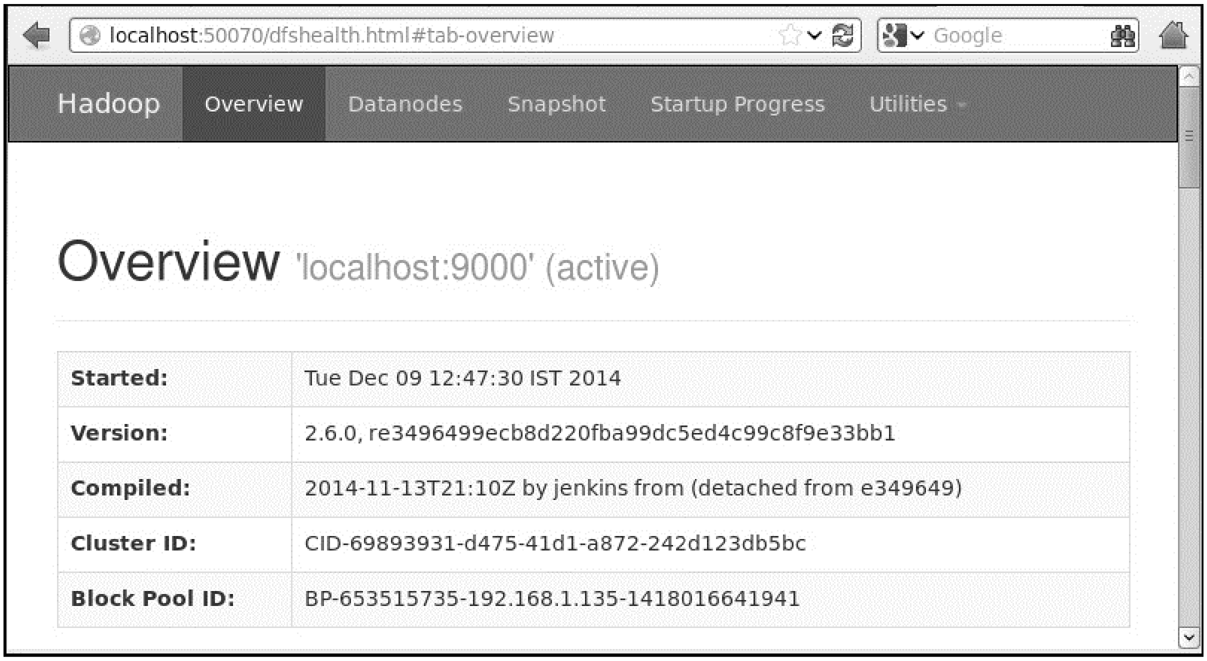
The expected output is as follows:

```
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop-2.4.1/logs/yarn-hadoop-
resourcemanager-localhost.out
localhost: starting nodemanager, logging to /home/hadoop/hadoop-2.4.1/logs/yarn-
hadoop-nodemanager-localhost.out
```

Accessing Hadoop on Browser

The default port number to access Hadoop is 50070. Use the following URL to get Hadoop services on your browser.

```
http://localhost:50070/
```



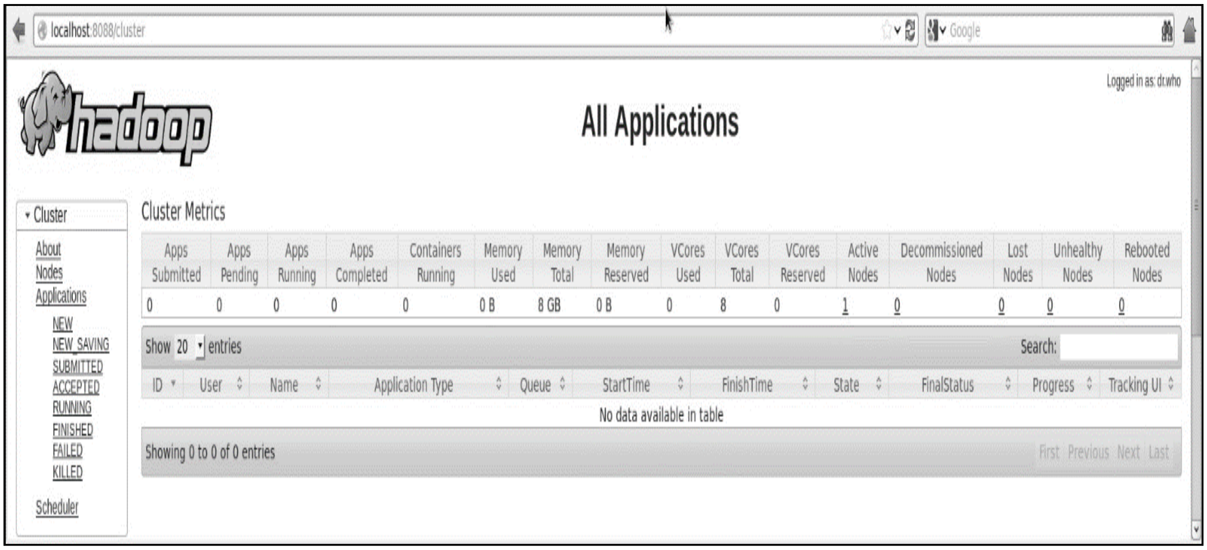
Overview 'localhost:9000' (active)

Started:	Tue Dec 09 12:47:30 IST 2014
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-69893931-d475-41d1-a872-242d123db5bc
Block Pool ID:	BP-653515735-192.168.1.135-1418016641941

Verify all applications for cluster

The default port number to access all applications of cluster is 8088. Use the following url to visit this service.

```
http://localhost:8088/
```



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
No data available in table										

Showing 0 to 0 of 0 entries

Once you are done with the installation of Hadoop, proceed to the next step and install Hive on your system.