# Cloudera Hadoop: Getting started with CDH Distribution

* [**Bookmark**](https://www.edureka.co/blog/cloudera-hadoop-tutorial/)

With the increasing demand for Big Data, and Apache Hadoop is at the heart of the revolution, it has changed the way we organize and compute the data. The need for organizations to align Hadoop with their business needs has fueled the emergence of the commercial distributions. Commercial Hadoop Distributions are usually packaged with features, designed to streamline the deployment of Hadoop. Cloudera Hadoop Distribution provides a scalable, flexible, integrated platform that makes it easy to manage rapidly increasing volumes and varieties of data in your enterprise.

In this blog on Cloudera Hadoop Distribution, we will be covering the following topics:

* + [Introduction to Hadoop](https://www.edureka.co/blog/cloudera-hadoop-tutorial/#Introduction)
  + [Hadoop Distributions](https://www.edureka.co/blog/cloudera-hadoop-tutorial/#HadoopDistributions)
  + [Cloudera vs MapR vs Hortonworks](https://www.edureka.co/blog/cloudera-hadoop-tutorial/#ClouderaVsMapRVsHortonworks)
  + [Cloudera Distribution](https://www.edureka.co/blog/cloudera-hadoop-tutorial/#CDHDistribution)
  + [Cloudera Manager](https://www.edureka.co/blog/cloudera-hadoop-tutorial/#ClouderaManager)
  + [Demonstration of the Cloudera Manager](https://www.edureka.co/blog/cloudera-hadoop-tutorial/#Cloudera_Manager_Demo)
  + [Creating an Oozie Workflow](https://www.edureka.co/blog/cloudera-hadoop-tutorial/#OozieWorkflow)

## ****Cloudera Hadoop: Introduction to Hadoop****

Hadoop is an Apache open-source framework that store and process Big Data in a distributed environment across the cluster using simple programming models. Hadoop provides parallel computation on top of distributed storage. To learn more about Hadoop in detail from [***Certified Experts***](https://www.edureka.co/big-data-and-hadoop) you can refer to this [**Hadoop tutorial blog.**](https://www.edureka.co/blog/hadoop-tutorial/)

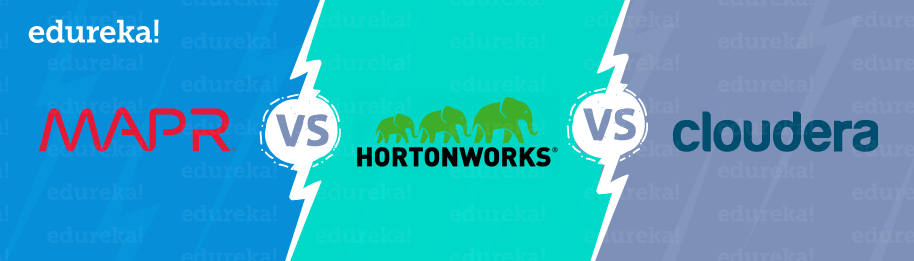
After this short introduction to Hadoop, let me now explain the different types of Hadoop Distribution.

## ****Cloudera Hadoop: Hadoop Distributions****

Since Apache Hadoop is open source, many companies have developed distributions that go beyond the original open source code. This is very akin to Linux distributions such as RedHat, Fedora, and Ubuntu. Each of the Linux distributions supports its own functionalities and features like user-friendly GUI in Ubuntu. Similarly, **Red Hat** is popular within enterprises because it offers support and also provides ideology to make changes to any part of the system at will. Red Hat relieves you from software compatibility problems. This is usually a big issue for users who are transitioning from Windows.

Likewise, there are 3 main types of Hadoop distributions which have its own set of functionalities and features and are built under the base HDFS.

## ****Cloudera vs MapR vs Hortonworks****

                                     Fig: MapR vs Hortonworks vs Cloudera

### ****Cloudera Hadoop Distribution****

Cloudera is the market trend in Hadoop space and is the first one to release commercial Hadoop distribution. It offers consulting services to bridge the gap between – “what does Apache Hadoop provides” and “what organizations need”.

Cloudera Distribution is:

* **Fast for business**: From analytics to data science and everything in between, Cloudera delivers the performance you need to unlock the potential of unlimited data.
* **Makes Hadoop easy to manage**: With Cloudera Manager, automated wizards let you quickly deploy your cluster, irrespective of the scale or deployment environment.
* **Secure without compromise:** Meets stringent data security and compliance needs without sacrificing business agility. Cloudera provides an integrated approach to data security and governance.

### Horton-Works****Distribution****

The Horton-Works Data Platform (HDP) is entirely an open source platform designed to maneuver data from many sources and formats. The platform includes various Hadoop tools such as the Hadoop Distributed File System (HDFS), MapReduce, Zookeeper, HBase, Pig, Hive, and additional components.

It also supports features like:

* HDP makes Hive faster through its new Stinger project.
* HDP avoids vendor lock-in by pledging to a forked version of Hadoop.
* HDP is focused on enhancing the usability of the Hadoop platform.

**MapR Distribution**

MapR is a platform-focused Hadoop solutions provider, just like HortonWorks and Cloudera. MapR integrates its own database system, known as MapR-DB while offering Hadoop distribution services. MapR-DB is claimed to be four to seven times faster than the stock Hadoop database, i.e. HBase, that is executed on other distributions.

It has its intriguing features like:

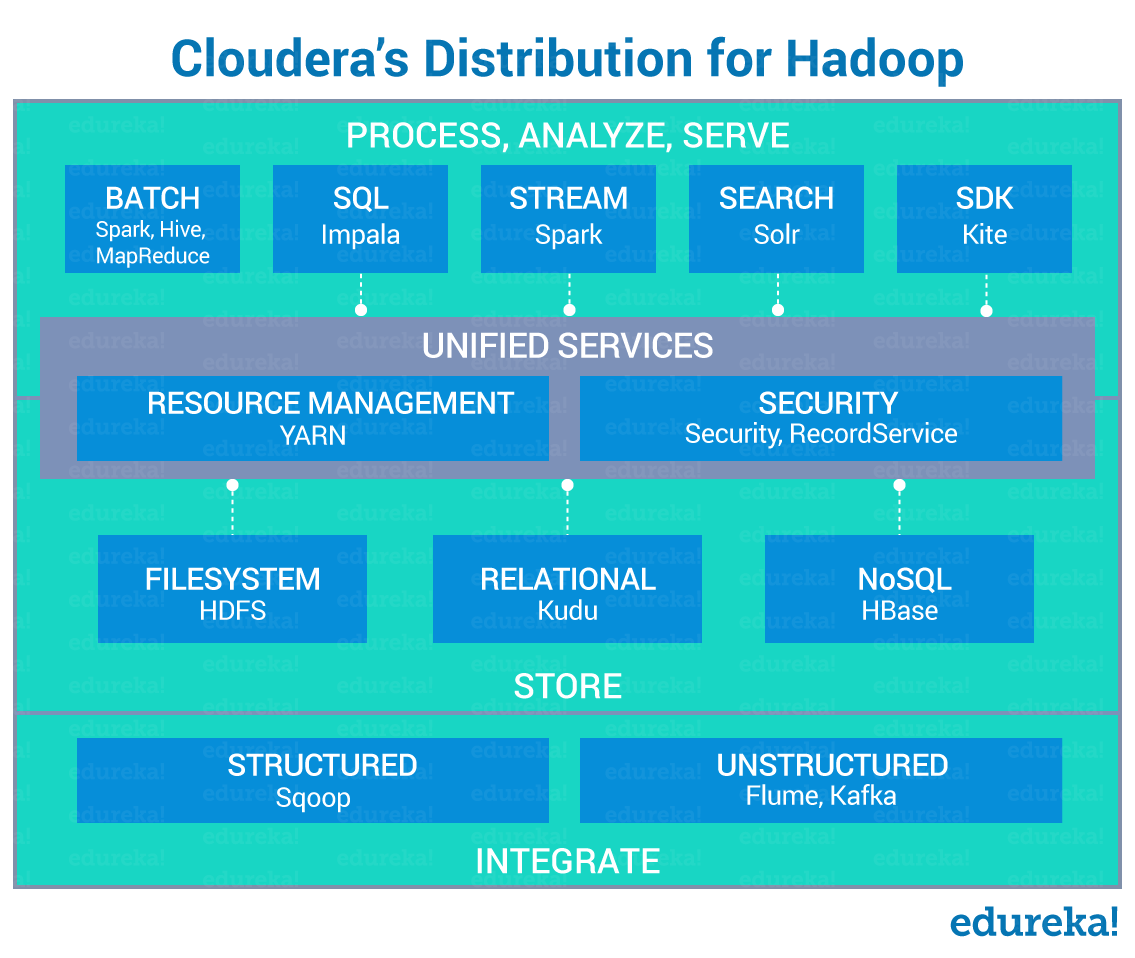
* It is the only Hadoop distribution that includes Pig, Hive, and Sqoop without any Java dependencies – since it relies on MapR-File System.
* MapR is the most production ready Hadoop distribution with many enhancements that make it more user-friendly, faster and dependable.

Now let’s discuss the Cloudera Hadoop Distribution in depth.

#### Subscribe to our YouTube channel to get new updates...

## ****Cloudera Hadoop: Cloudera Distribution****

Cloudera is the best-known player in the Hadoop space to release the first commercial Hadoop distribution.

                                           Fig: Cloudera Hadoop Distribution

Cloudera Hadoop Distribution supports the following set of features:

1. Cloudera’s CDH comprises all the open source components, targets enterprise-class deployments, and is one of the most popular commercial Hadoop distributions.
2. Known for its innovations, Cloudera was the first to offer **SQL-for-Hadoop** with its **Impala** query engine.
3. The management console – **Cloudera Manager**, is easy to use and implement with the rich user interface displaying all the cluster information in an organized and clean way.
4. In CDH you can add services to the up and running cluster without any disruption.
5. Other additions of Cloudera includes security, user interface, and interfaces for integration with third-party applications.
6. CDH provides **Node Templates** i.e. it allows the creation of a group of nodes in a Hadoop cluster with varying configuration. It eradicates the use of the same configuration throughout the Hadoop cluster.
7. It also supports:
   * **Reliability**Hadoop vendors promptly act in response whenever a bug is detected. With the intent to make commercial solutions more stable, patches and fixes are deployed immediately.
   * **Support**  
     Cloudera Hadoop vendors provide technical guidance and assistance that makes it easy for customers to adopt Hadoop for enterprise level tasks and mission-critical applications.
   * **Completeness**  
     Hadoop vendors couple their distributions with various other add-on tools which help customers customize the Hadoop application to address their specific tasks.

Cloudera distributions come up with 2 different types of editions.

1. Cloudera Express Edition
2. Cloudera Enterprise Edition

Now let’s look at the differences between them.

|  |  |  |
| --- | --- | --- |
| **Features** | **Cloudera-Express** | **Cloudera-Enterprise** |
| **Cluster Management** | | |
| 1. Multi-Cluster Management | Yes | Yes |
| 2. Resource Management | Yes | Yes |
| **Deployment** | | |
| 1. Support for CDH 4 and 5 | Yes | Yes |
| 2. Rolling upgrade of CDH | No | Yes |
| **Service and Configuration Management** | | |
| 1. Manage HDFS, MapReduce, YARN, Impala, HBase, Hive, Hue, Oozie, Zookeeper, Solr, Spark, and Accumulo services | Yes | Yes |
| 2. Rolling restart of services | No | Yes |
| **Security** | | |
| 1. LDAP Authentication | No | Yes |
| 2. SAML Authentication | No | Yes |
| **Monitoring and Diagnostics** | | |
| 1. Health History | Yes | Yes |
| **Alert Management** | | |
| 1. Alert via email | Yes | Yes |
| 2. Alert via SNMP | No | Yes |
| **Advanced Management Features** | | |
| 1. Automated backup and recovery | No | Yes |
| 2. File browsing and searching | No | Yes |
| 3. MapReduce, Impala, HBase, Yarn      usage reports | No | Yes |

## ****Cloudera Hadoop: Cloudera Manager****

According to Cloudera, Cloudera Manager is the best way to **install**, **configure**, **manage**, and **monitor** the Hadoop stack.

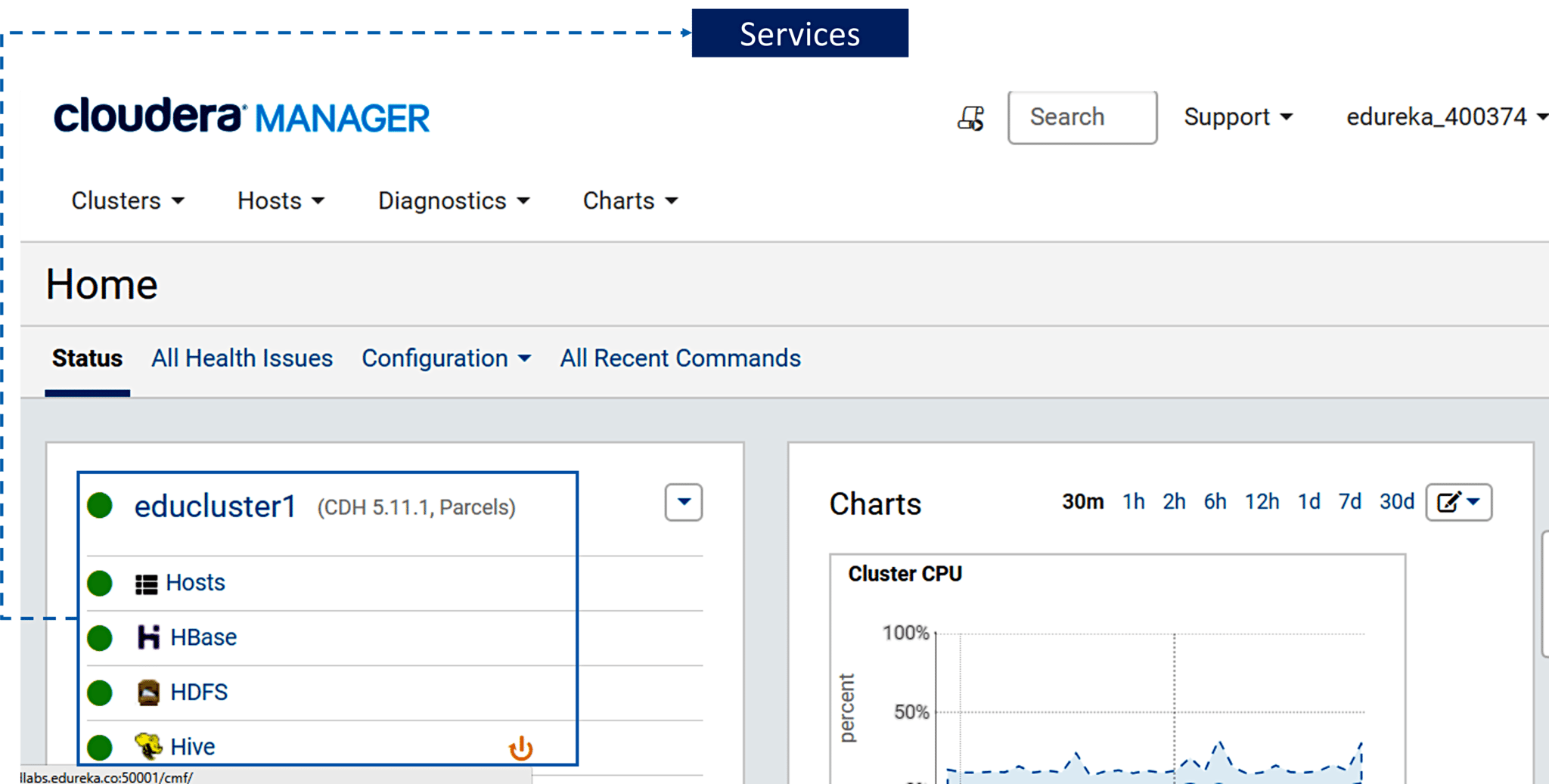
It provides:

1. Automated deployment and configuration
2. Customizable monitoring and reporting
3. Effortless robust troubleshooting
4. Zero – Downtime maintenance

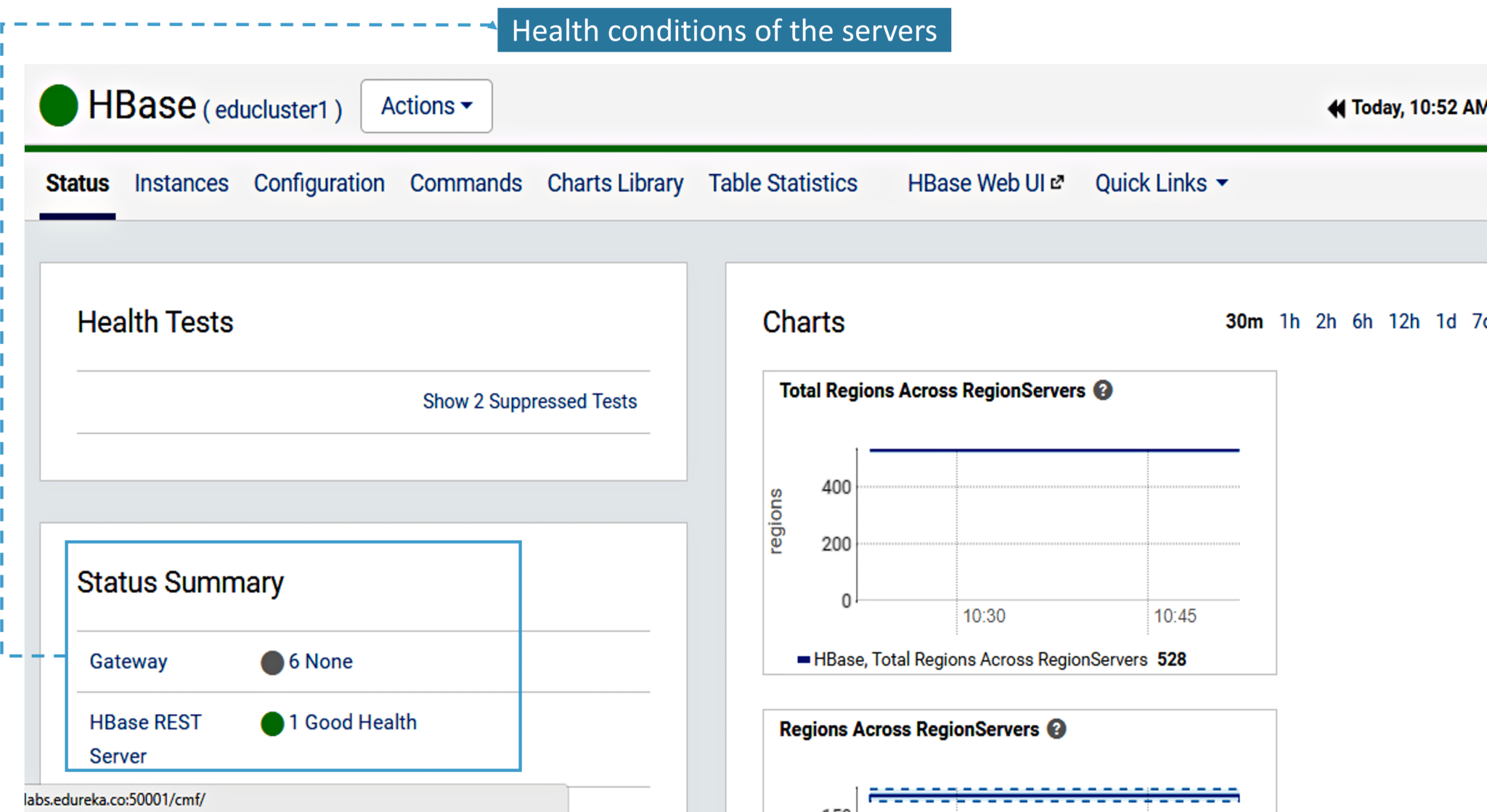
## ****Demonstration of Cloudera Manager****

Let’s explore the Cloudera Manager.

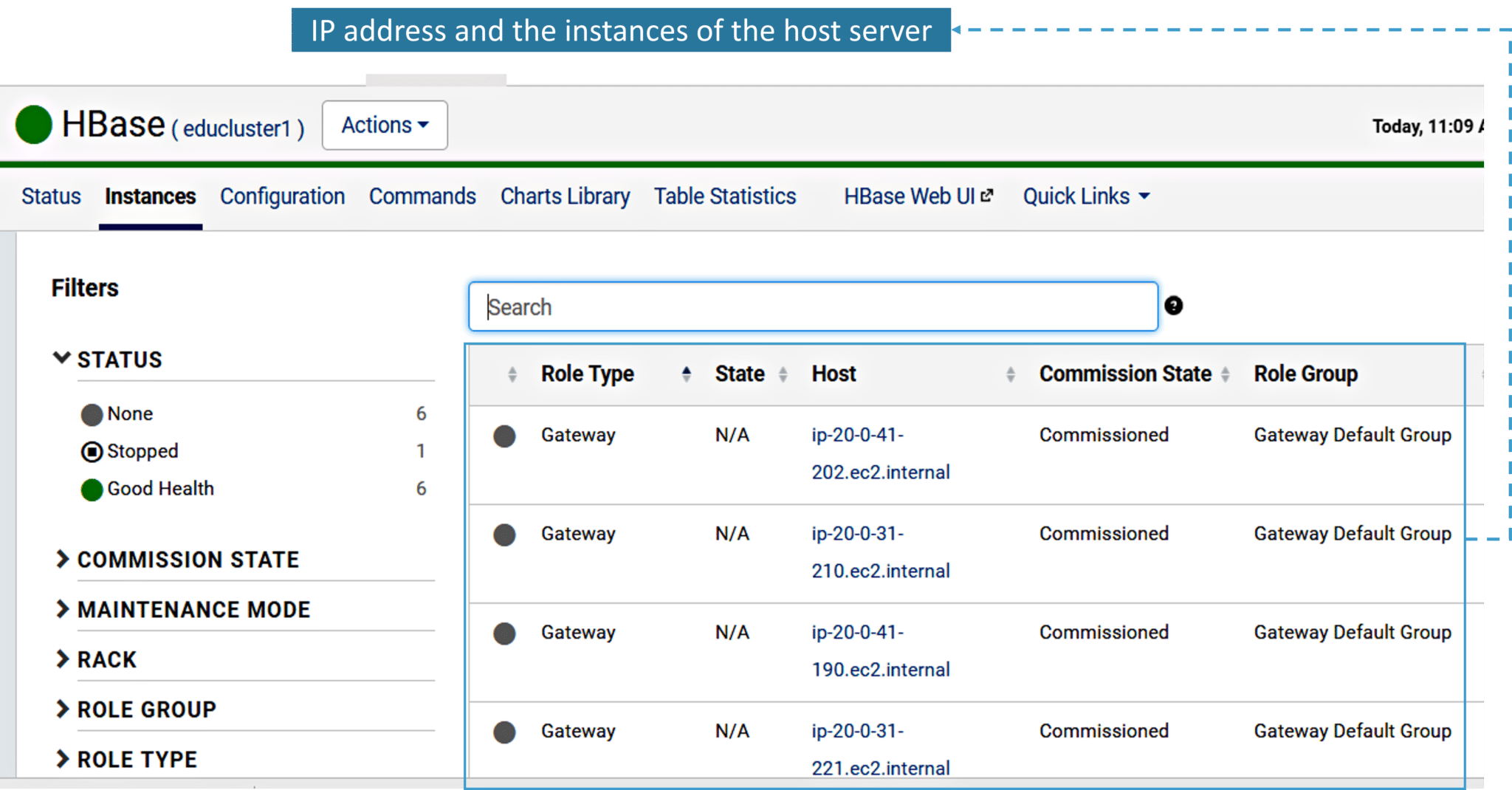
1. Below figure shows the number of services that are currently running in the Cloudera Manager. You can also view the charts about cluster CPU usage, Disk IO usage, etc.

                                              Fig: Homepage of Cloudera Manager

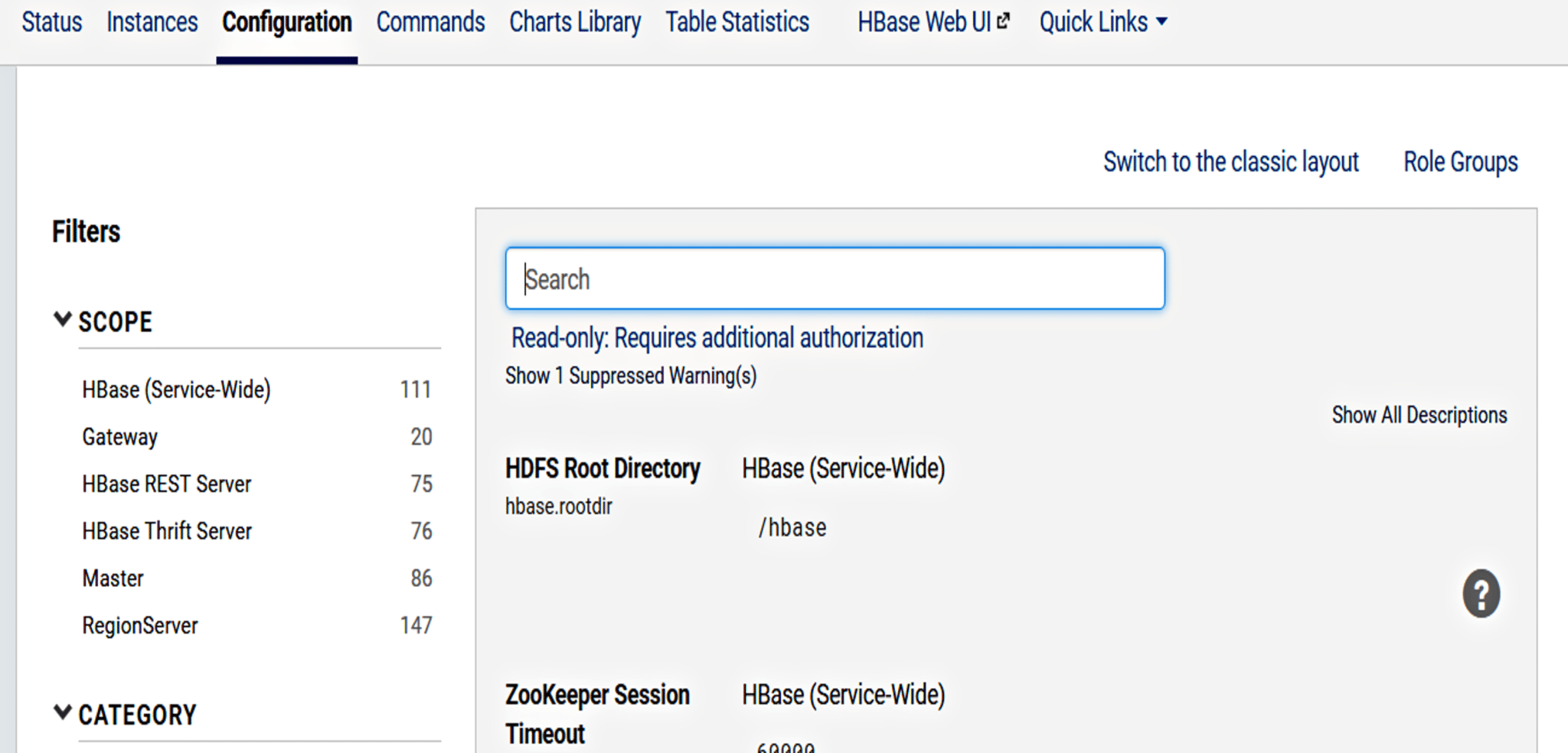
2. Below image demonstrates the HBase cluster. It gives you charts and graphs about the health conditions of the currently running HBase REST server.

                                       Fig: Health Conditions of the HBase server

3. Now, let’s have a look at the Instances tab of HBase cluster where you can check the status and the IP configuration.

                              Fig: Status and IP address of the Host Server of the HBase cluster

4. Next, you have Configuration tab. Here you can see all the configuration parameters and change their values.

                                               Fig: Configuration of the HBase cluster

Now, let’s understand what are Parcels in Cloudera.

## ****Cloudera Hadoop: Parcels****

A parcel is a binary distribution format containing the program files, along with additional metadata used by Cloudera Manager.

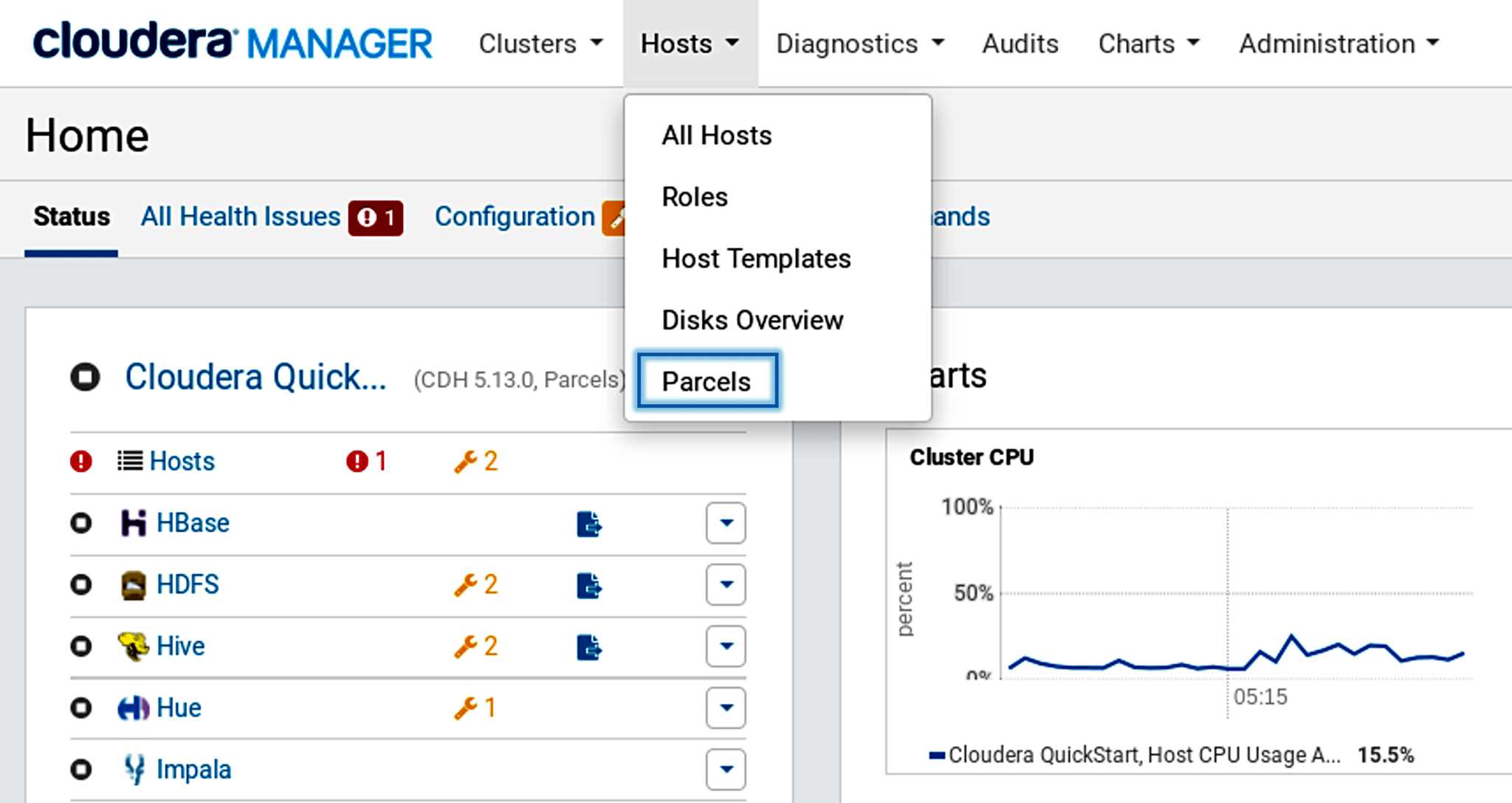
Parcels are self-contained and installed in a versioned directory, which means that multiple versions of a given service can be installed side-by-side.

Below are the benefits of using Parcel:

* It provides distribution of CDH as a single object i.e. instead of having a separate package for each part of CDH, parcels just have a single object to install.
* It offers internal consistency (as the complete CDH is distributed as a single parcel, all the CDH components are matched and there will be no risk of different parts coming from different versions of CDH).
* You can install, upgrade, downgrade, distribute, and activate the parcels in CDH using few clicks.

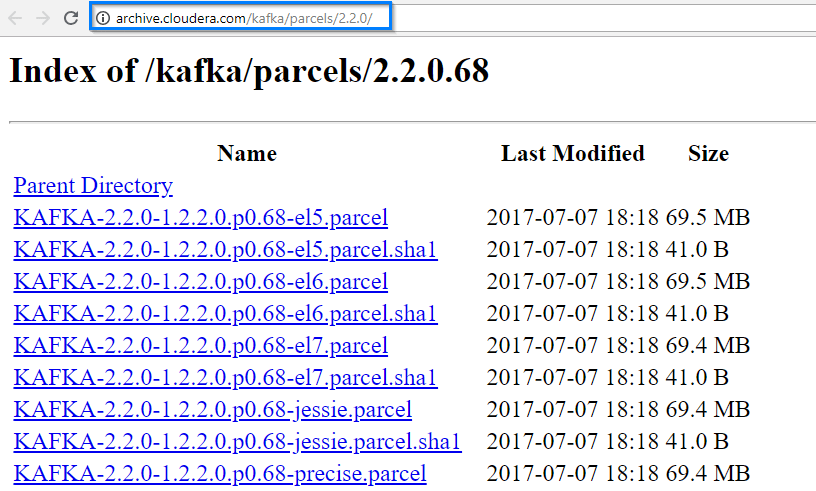
Now, let’s see how to install and activate Kafka service in CDH using Parcels.

1. Go to Cloudera manager homepage >> Hosts >>Parcels as shown below

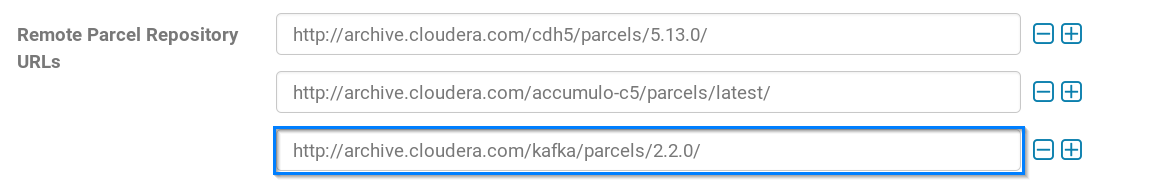
                                       Fig: Selecting parcels from the hosts

2. If you do not see Kafka in the list of parcels, you can add the parcel to the list.

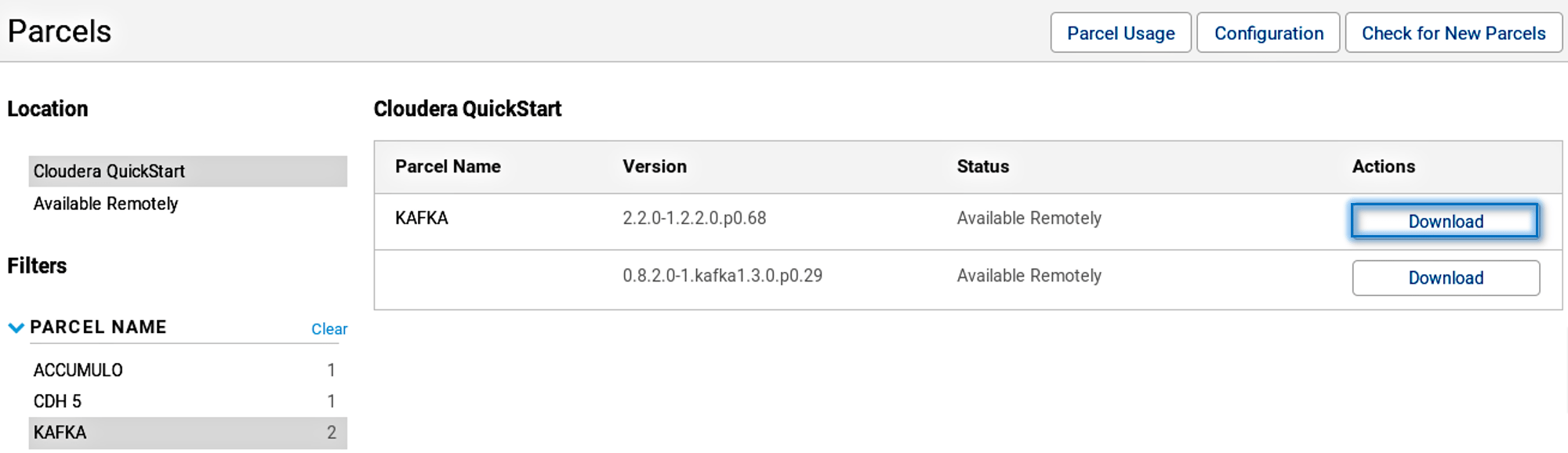
1. Find the parcel of the Kafka version you want to use. If you do not see it, you can add the parcel repository to the list.
2. Find the parcel for the version of Kafka you want to install – [Cloudera Distribution of Apache Kafka Versions](https://www.cloudera.com/documentation/kafka/latest/topics/kafka_packaging.html#concept_fzg_phl_br).  
   Below figure demonstrates the same.

                                           Fig: Repository path for the parcel.

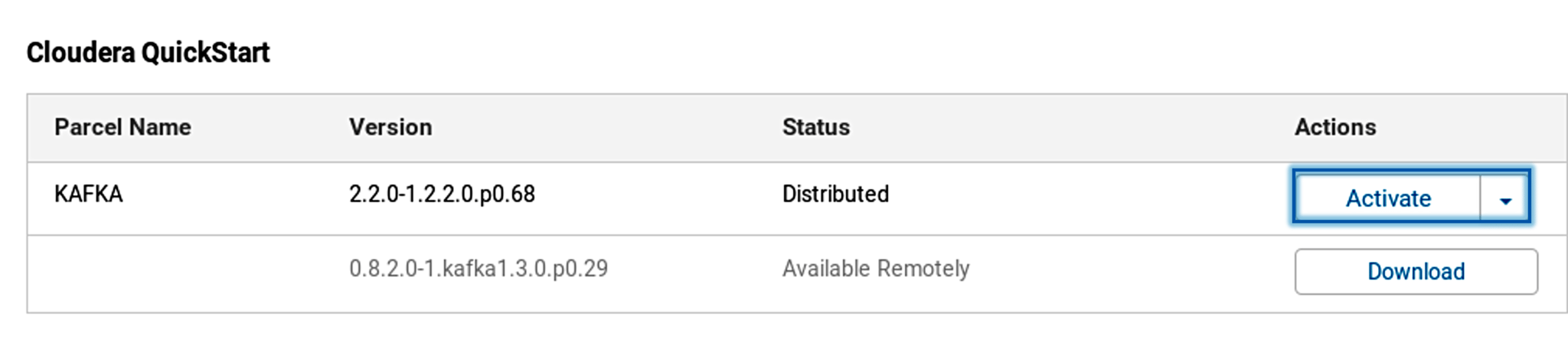
3. Copy the link as shown in the above figure and add it to the Remote Parcel Repository as shown below.

                                           Fig: Addition of the Kafka path from the repository

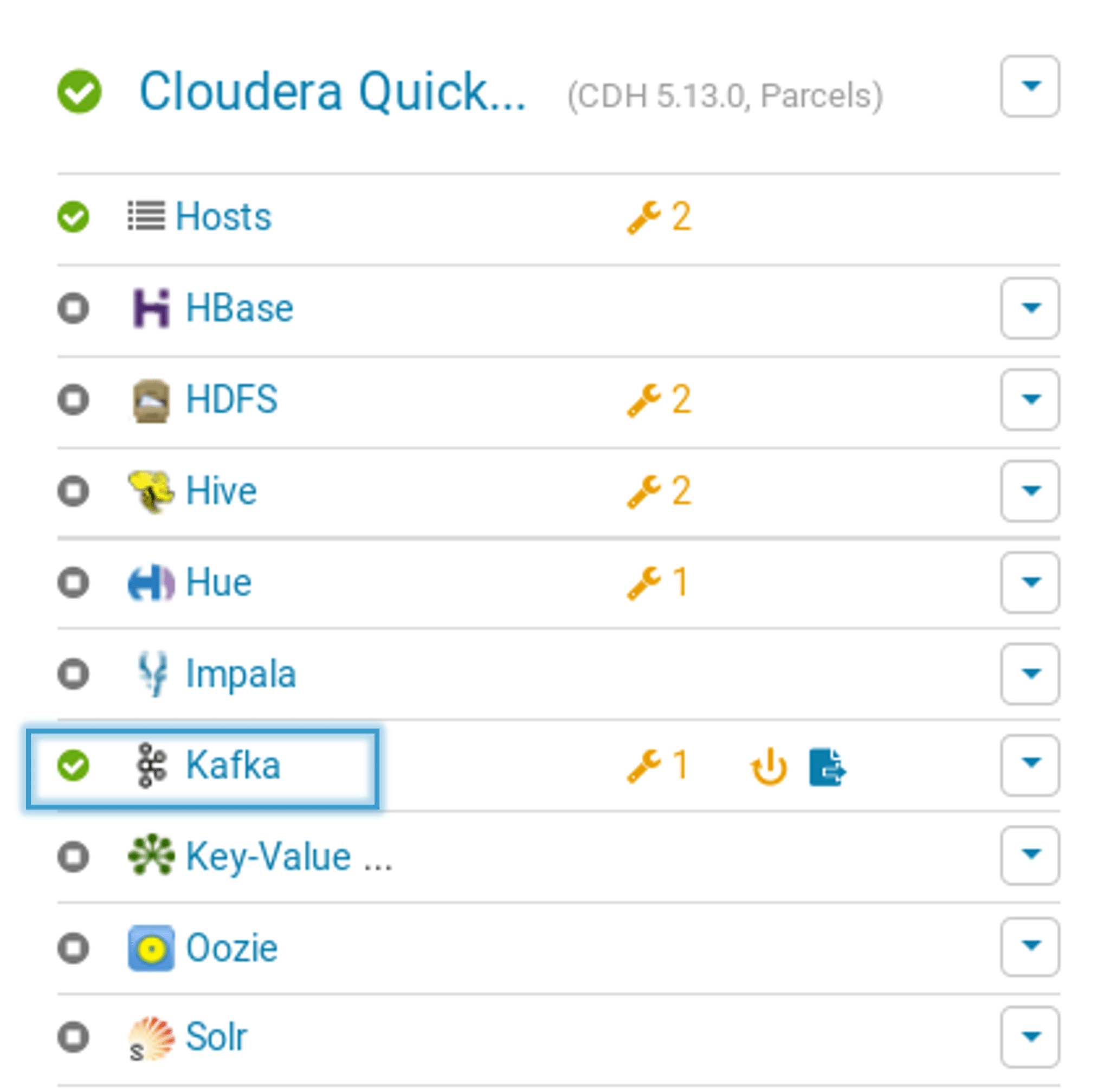
4. After adding the path, Kafka will be ready for download. You can just click on the download button and download the Kafka.

                                                         Fig: Downloading the Kafka

5. Once Kafka is downloaded, all you need to do is to distribute and activate it.

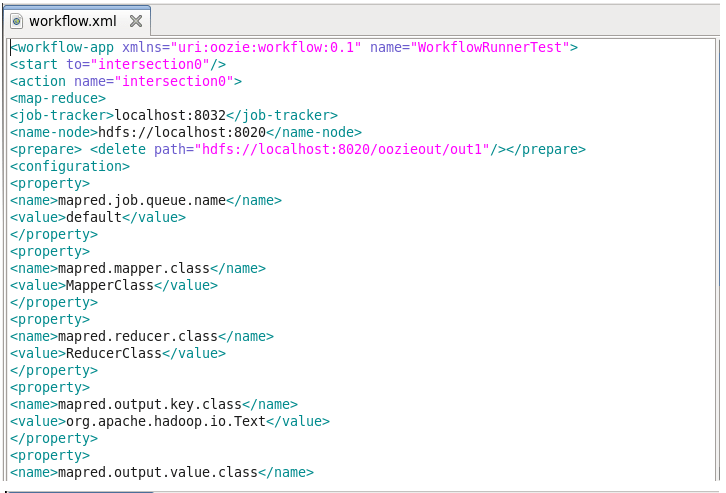
                                                    Fig: Activating the Kafka

Once it is activated, you can go ahead and view the Kafka in the services tab in Cloudera manager.

                                      Fig: Kafka service

## ****Cloudera Hadoop: Creating an Oozie Workflow****

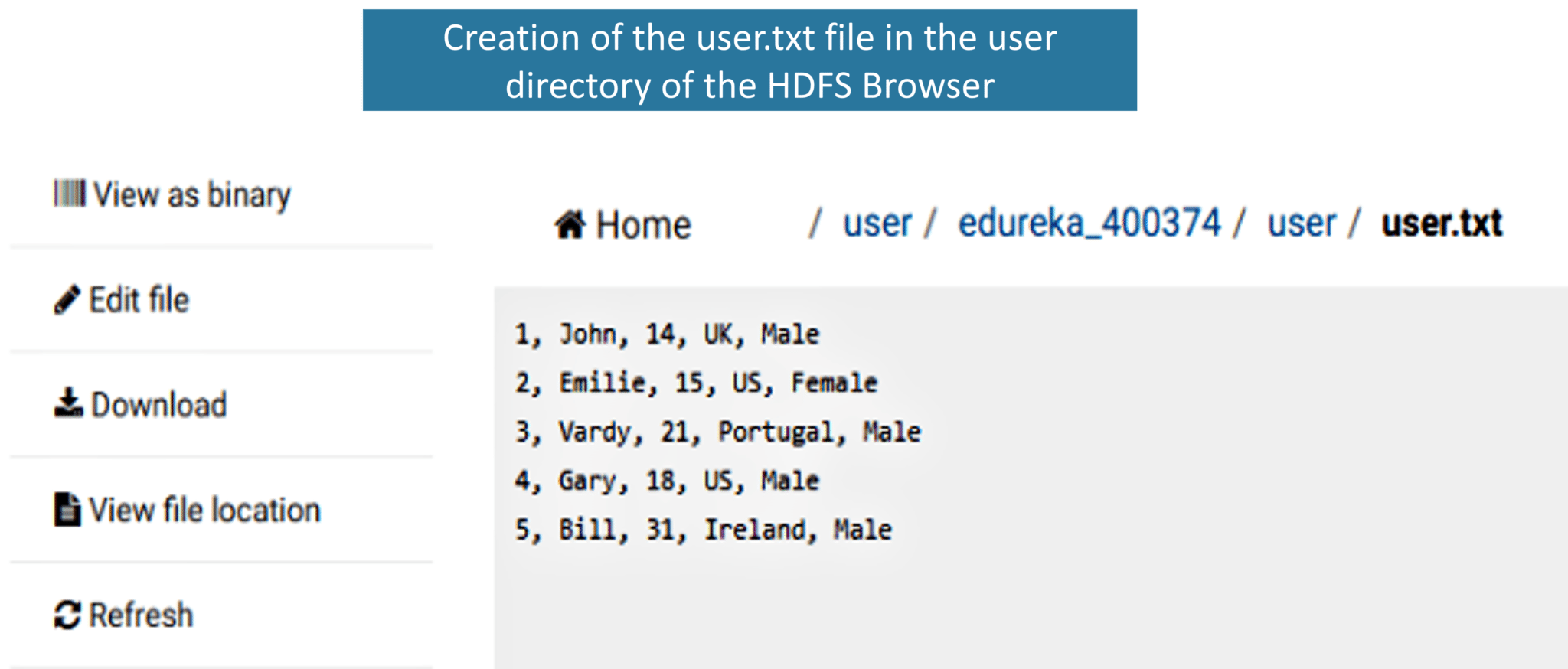
Creating a workflow by manually writing the XML code and then executing it, is complicated. You can refer this [Scheduling the Oozie job](https://www.edureka.co/blog/apache-oozie-tutorial/) blog, to know about the traditional approach.

You can see the below image, where we have written an XML file to create a simple Oozie workflow.                                                     Fig: Creating an Oozie workflow using a Traditional approach

As you can see even to create a simple Oozie scheduler we had to write huge XML code which is time-consuming, and debugging every single line becomes cumbersome. In order to overcome this, Cloudera Manager introduced a new feature called **Hue** which provides a GUI and a simple drag and drop features to create and execute Oozie workflows.

Now let’s see how Hue performs the same task in a simplified way.

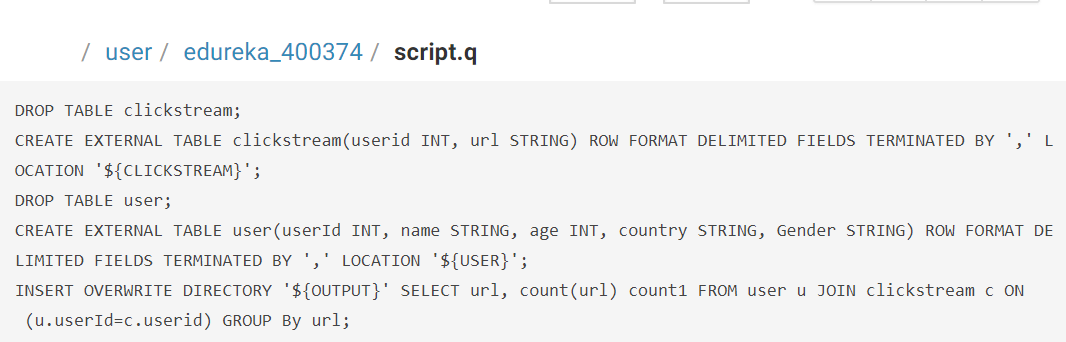
Before creating a workflow, let’s first create input files, i.e. clickstream.txt and user.txt.  
In the user.txt file, we have User Id, Name, Age, Country, Gender as shown below. We need this user file to know the user counts and clicks on the URL(mentioned in the clickstream file) based on the User Id.

                                                    Fig: Creating a text file

In order to know the number of clicks by the user on each URL, we have a clickstream containing the User Id and URL.                               

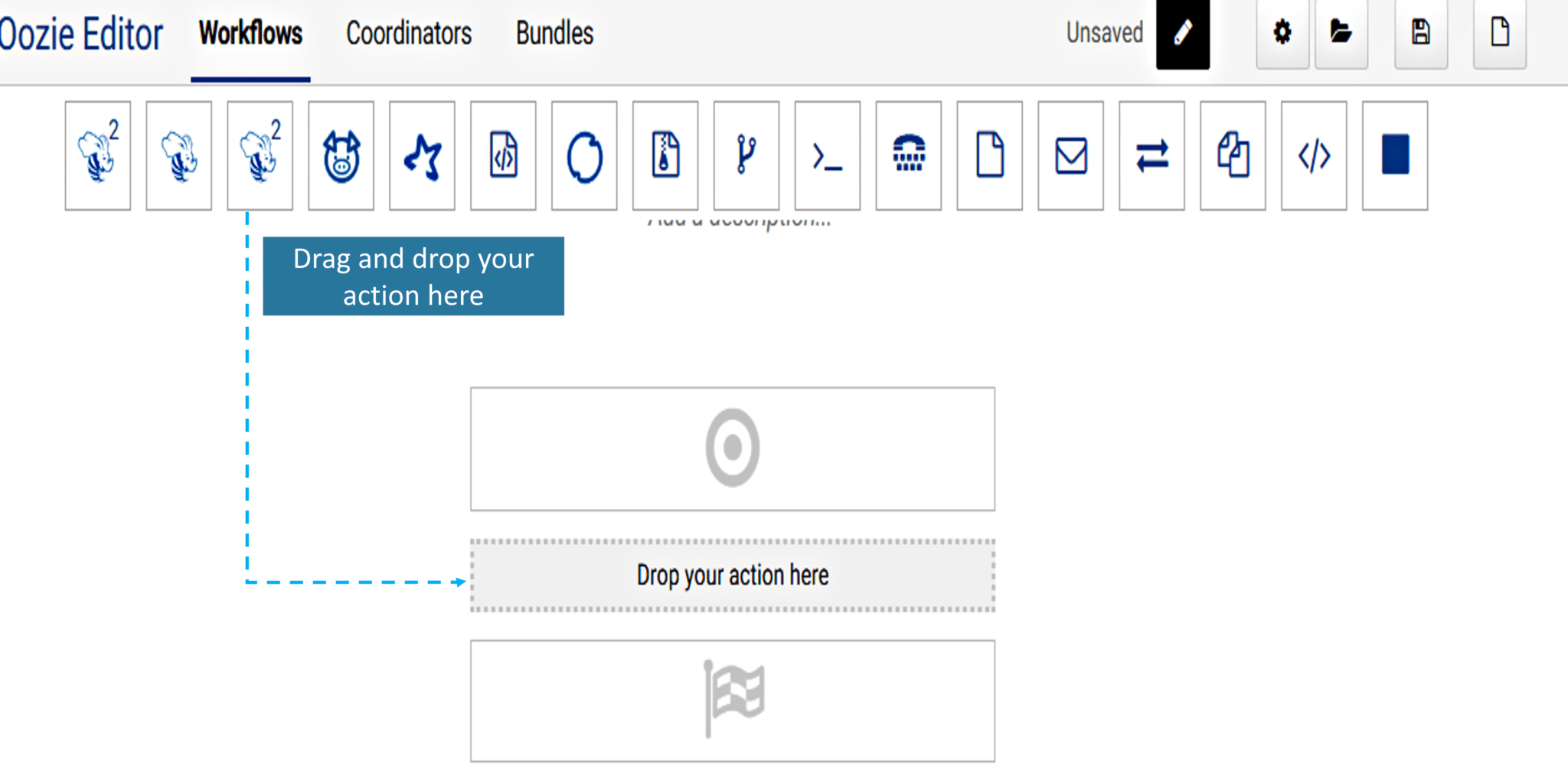
Fig: Clickstream file

Let’s write the queries in the script file.

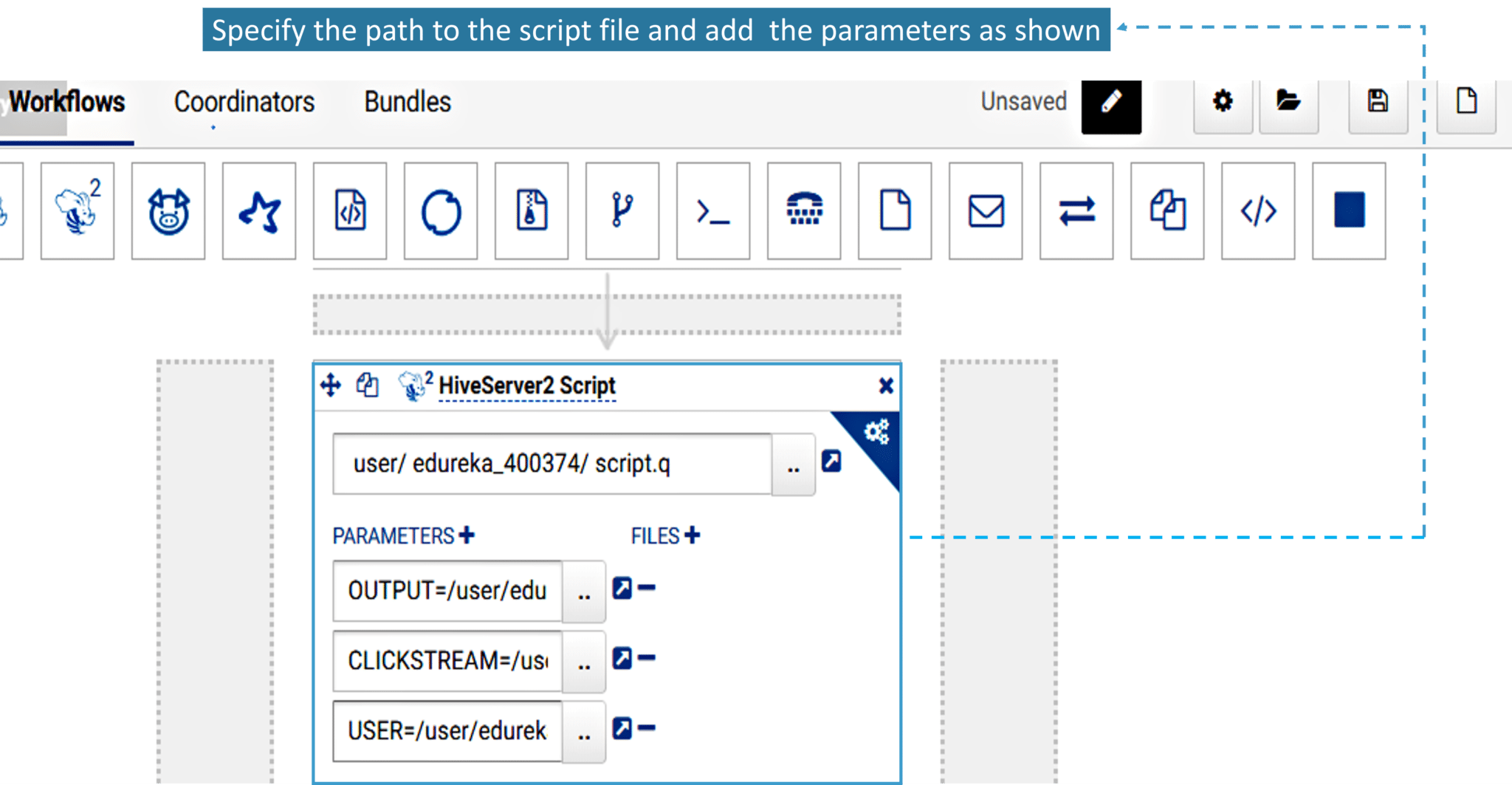
                                                           Fig: Script file

After creating the user file, clickstream file, and script file next, we can go ahead and create the Oozie workflow.

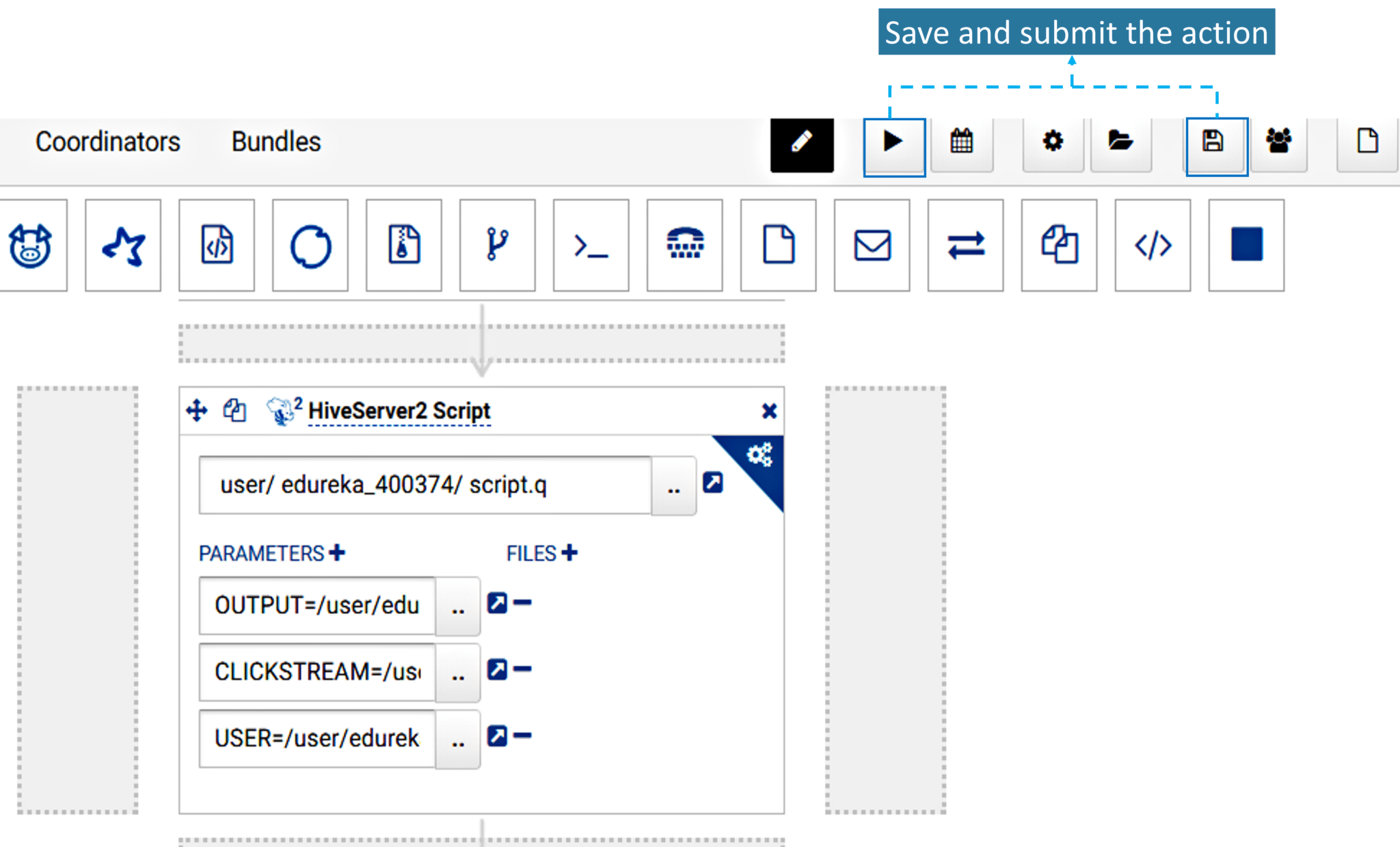
1. You can simply drag and drop the Oozie workflow as shown in the image.

                                 Fig: Drag and drop feature of creating the Oozie workflow

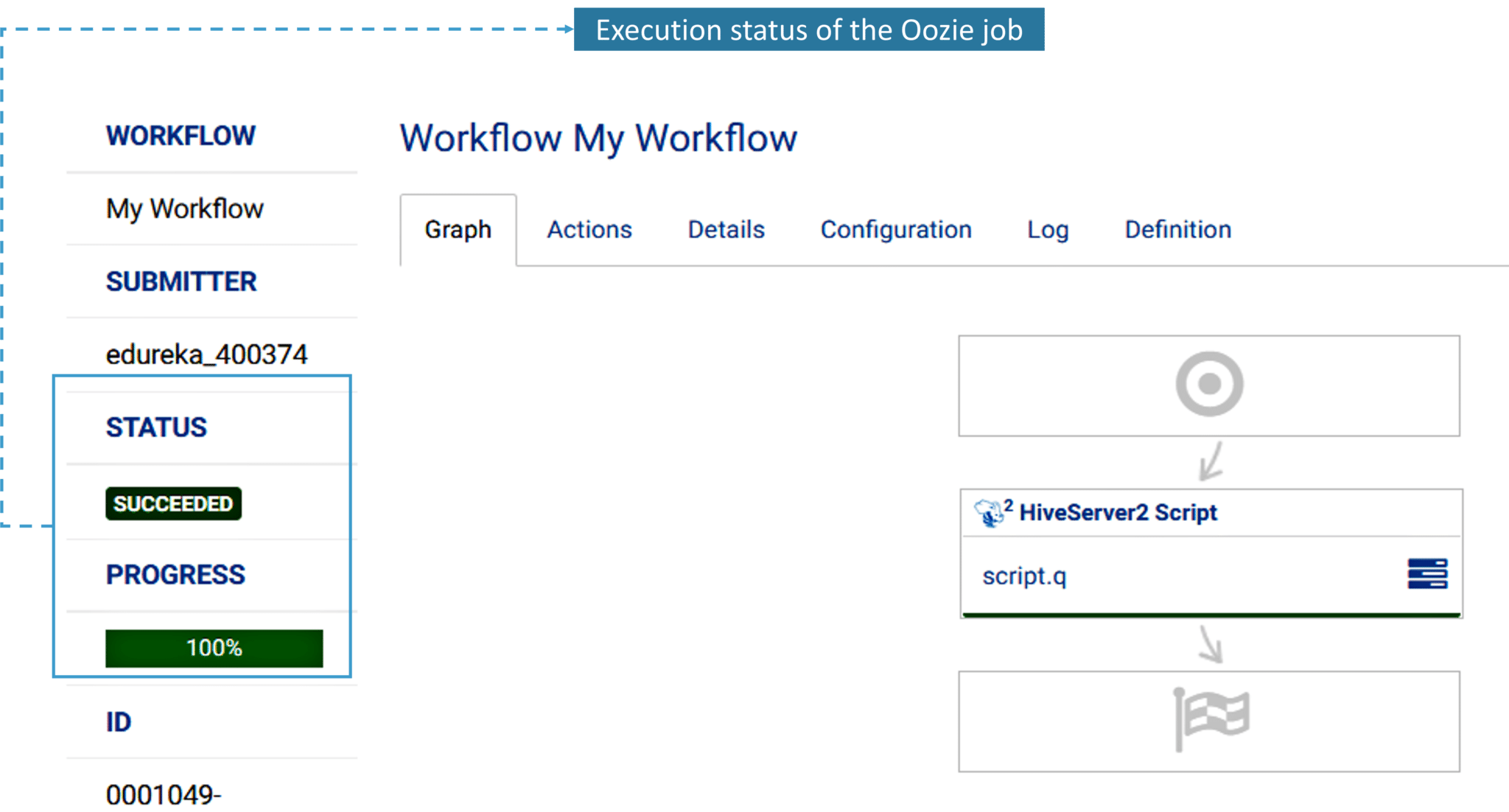
2. Soon after dropping your action you have to specify the paths to the script file and add the parameters mentioned in the script file. Here you need to add OUTPUT, CLICKSTREAM, and USER parameters and specify the path to each of the parameters.

                    Fig: Adding a script file and the required Parameters to execute the action

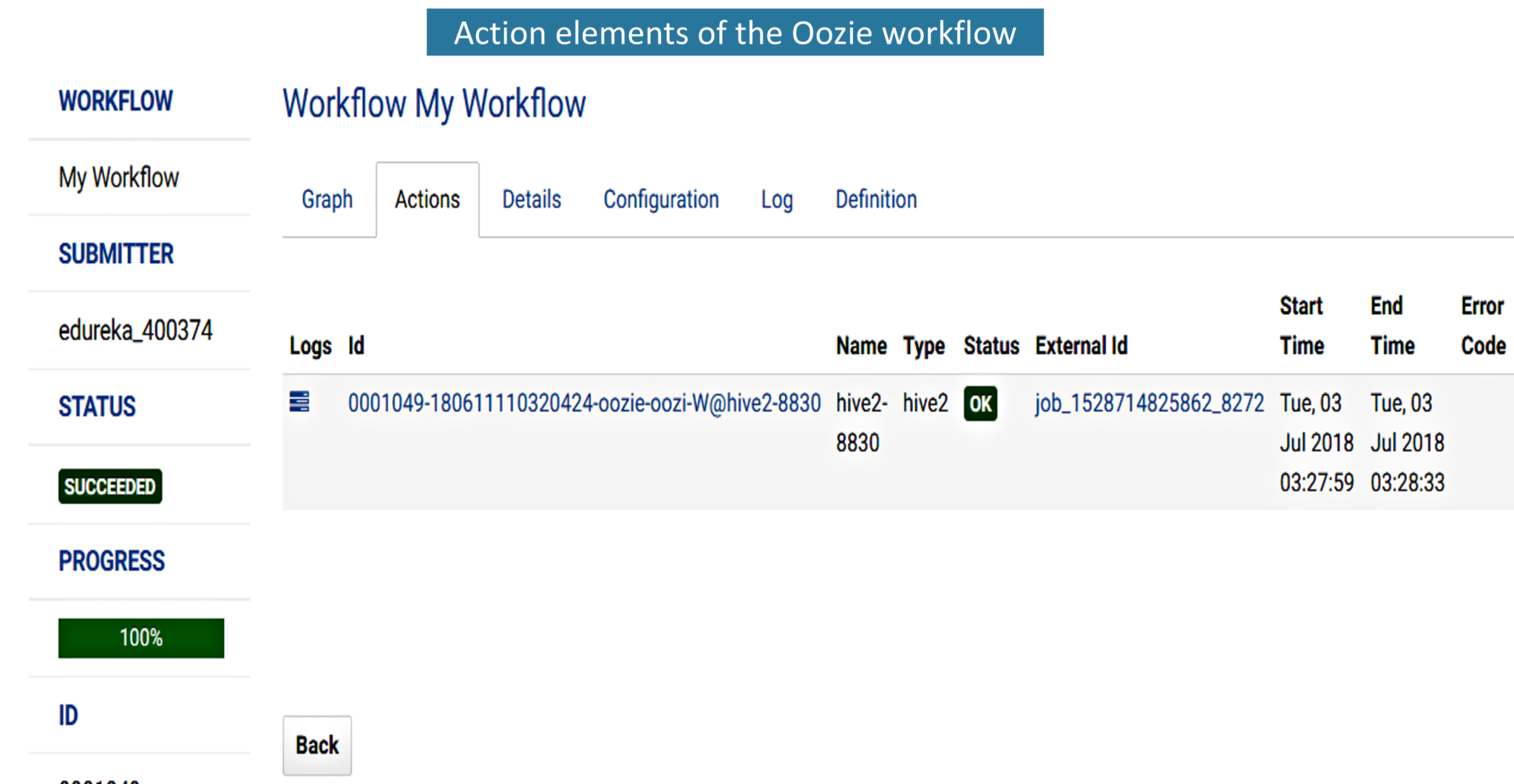
3. Once you have specified the paths and added the parameters, now simply save and submit the workflow as shown in the below image.

                                    Fig: Saving and submitting the Oozie action

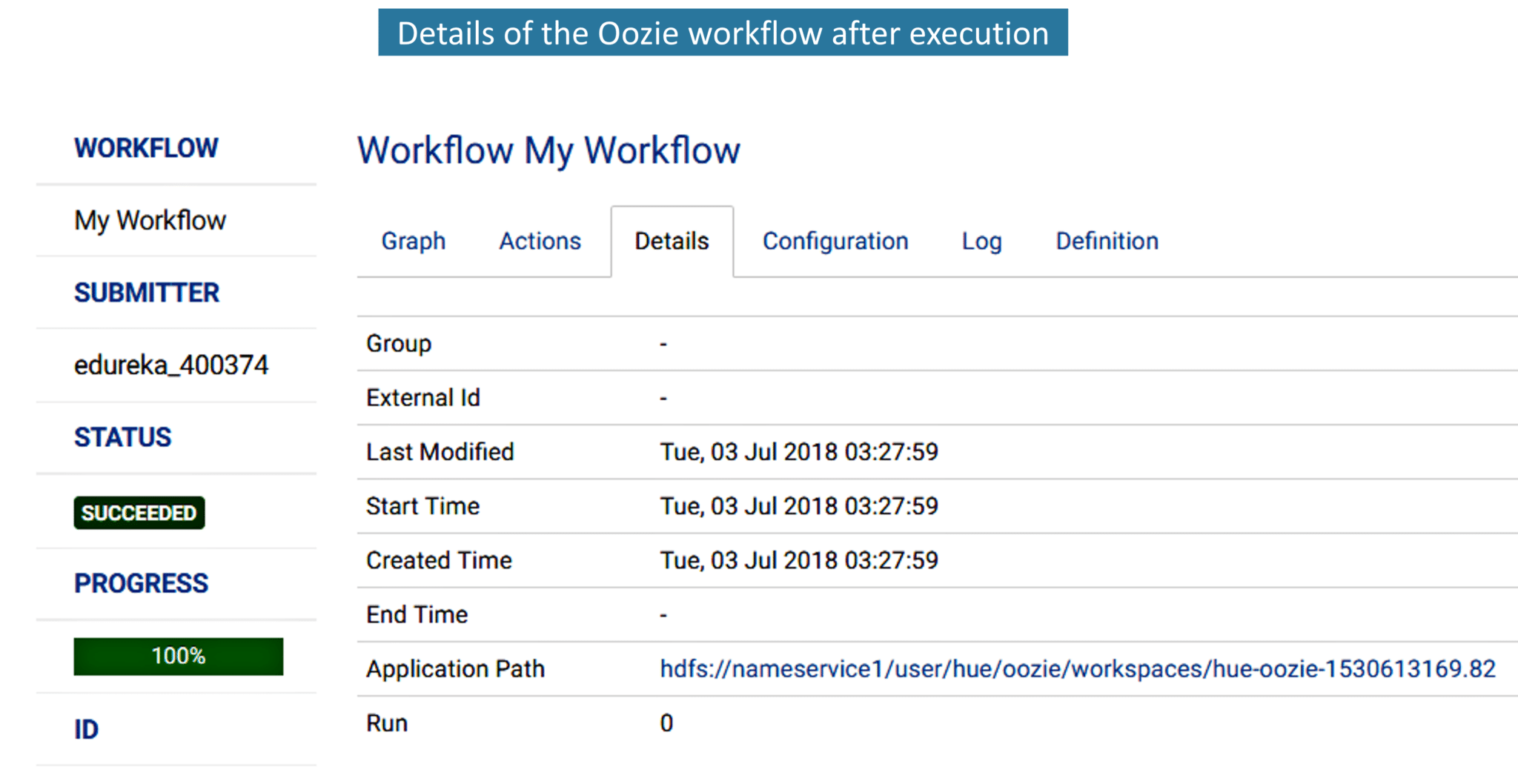
4. Once you submit the task, your job is completed. Execution and the other steps are taken care by Hue.

                                              Fig: Execution status of the Oozie job

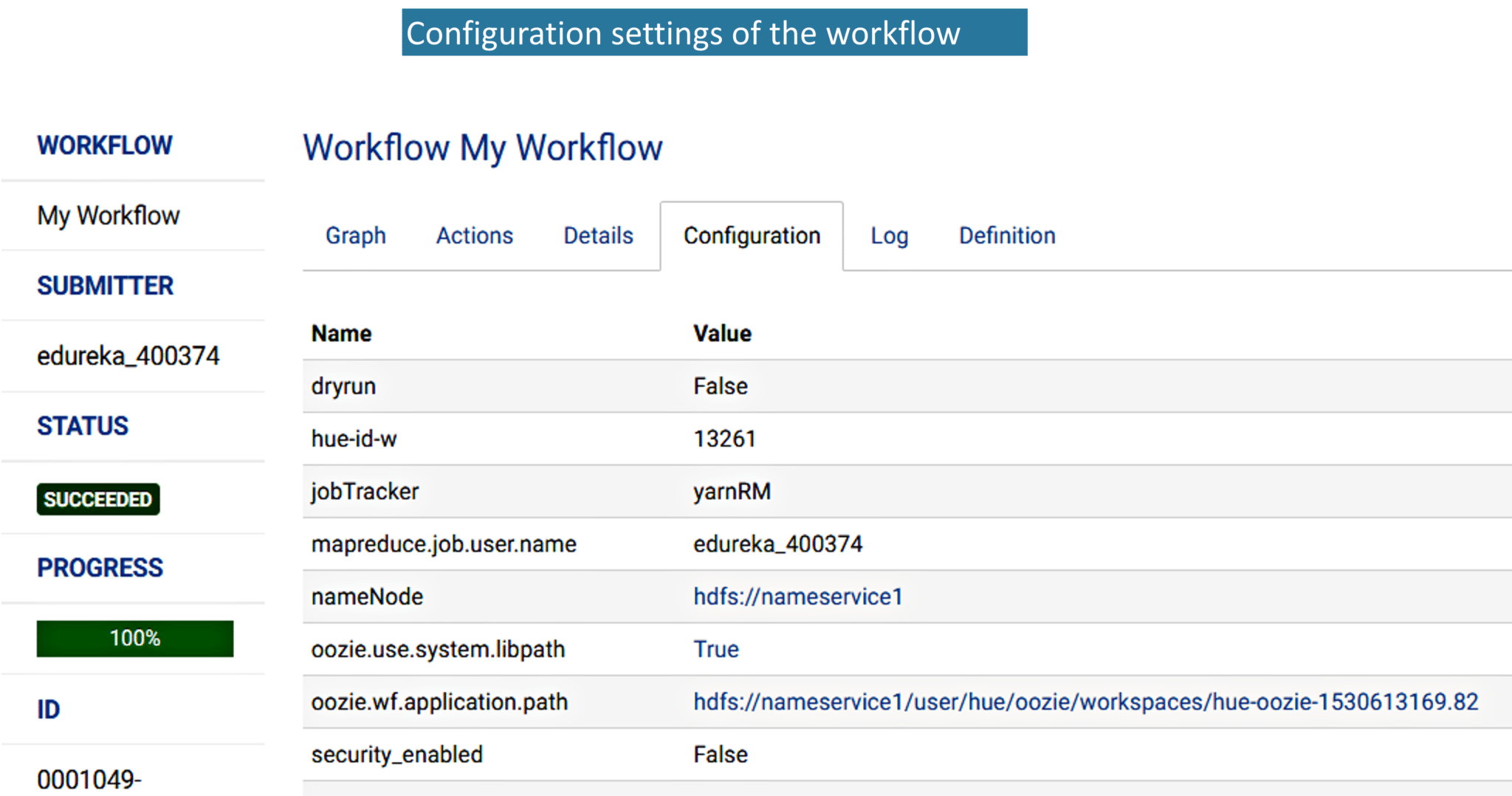
5. Now that we have executed the Oozie job, let’s take a look at the action tab. It contains the user ID and the status of the workflow. It also shows error codes if they’re any, the start and end time of the action item.

                                Fig: Elements present in the action tab of the Oozie workflow

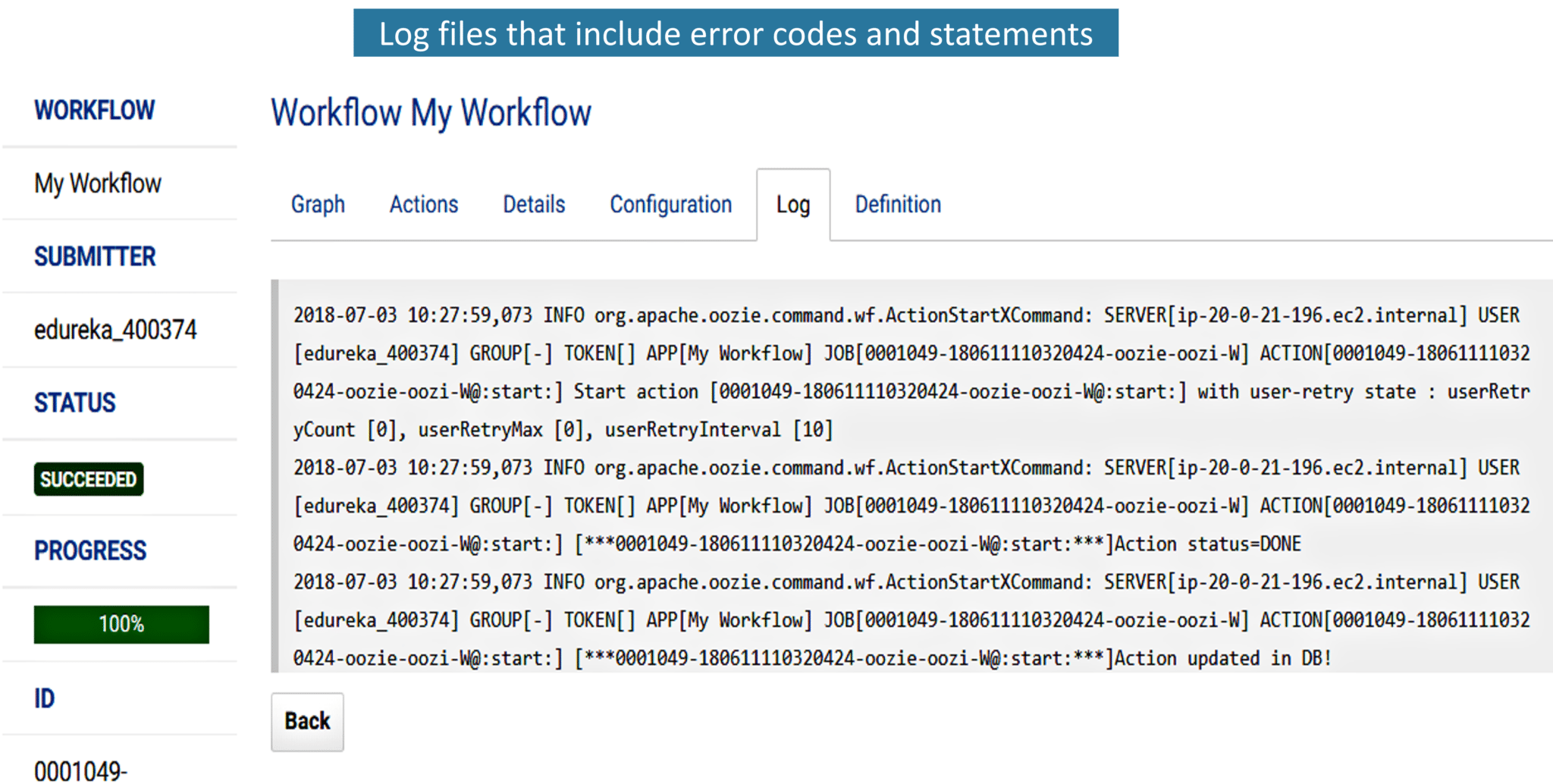
6. Next to the action tab is the details tab. In this, we can see the start time and the last modified time of the job.

                                                  Fig: Details of the Oozie workflow.

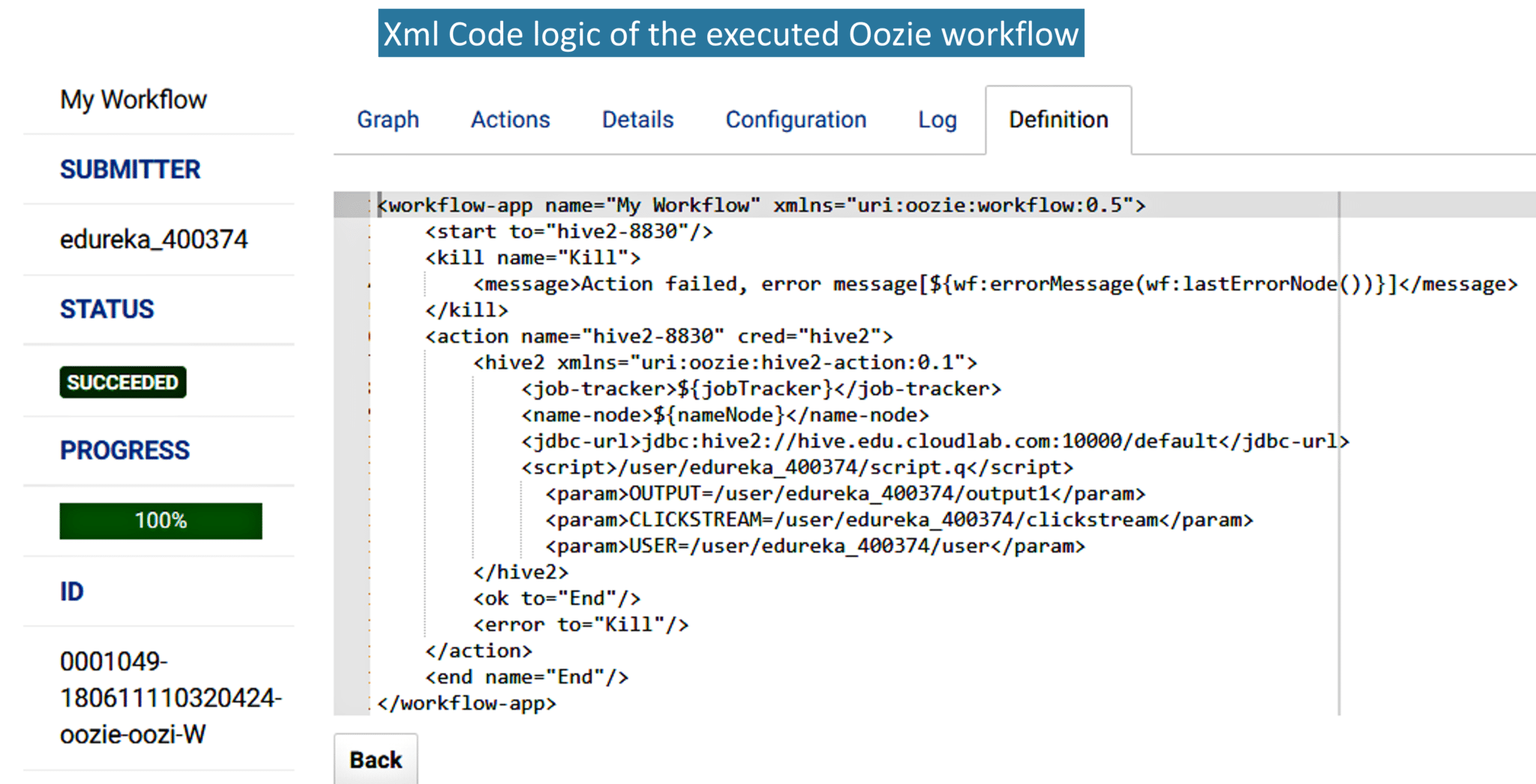
7. Next to Details tab, we have the Configuration tab of the workflow.

                                    Fig: Configuration settings of the Oozie workflow

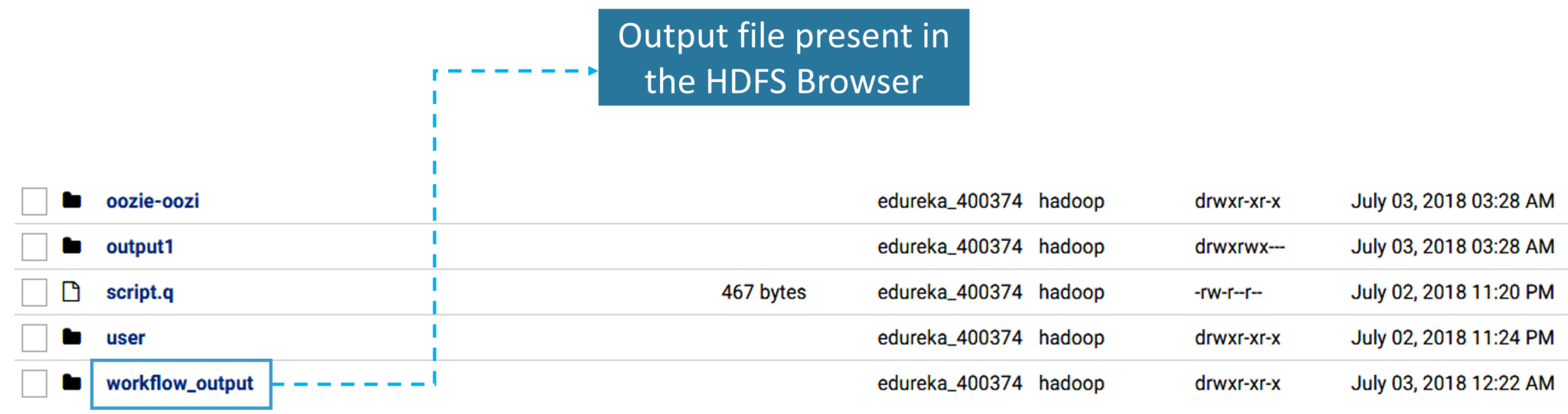
7. While executing the action item, if there are any errors, it will be listed in the Log tab. You can refer to the error statements and debug it accordingly.

                                 Fig: Log file that contains error codes and error statements

8. Here is the XML code of the workflow that is automatically generated by Hue.

[Wee](https://www.edureka.co/big-data-hadoop-training-certification" \t "_blank)                                            Fig: XML code of the Oozie workflow

9.1. As you have already specified the path for the output directory in step 2, here you have the output directory in the HDFS Browser as shown below.

                                       Fig: Output directory of the HDFS Browser

9.2 Once you click on the output directory, you will find a text file named as output.txt and that text file contains the actual output as shown in the below figure.

                    Fig: Final output text

This is how Hue makes our work simple by providing the drag and drop options to create an Oozie workflow.