**INDEX**

[**Prerequisites**](#_1ftdw4xjgz91) **2**

[1. Port Opening for internal cluster communication](#_k9w746enc6ca) 2

[2. Before Ansible Deployment](#_kq4dheby2e0d) 2

[**Ansible Deployment Steps**](#_9u4xqv8xc1nq) **6**

[Prerequisites to run hadoop via ansible](#_ej22g2dpoj09) 6

[Setting up ansible configuration](#_7jfetd2y2v9q) 7

[Verify the hadoop services are running by accessing the WebUI](#_le63ge6c87l5) 13

[**Connecting Hadoop Service Remotely**](#_557lfq3mmhzt) **17**

[Submitting spark job remotely](#_b6sj9zkc0jcs) 19

[Connecting Hive services remotely](#_djq20qbu49mp) 20

[Run Hbase query remotely](#_r30lxvaw1htm) 21

Hadoop Ecosystem

# Prerequisites

## Port Opening for internal cluster communication

## Before Ansible Deployment

* Download the hadoop ecosystem binaries by using the below links:-  
  + Hadoop binary:- <https://archive.apache.org/dist/hadoop/common/hadoop-2.10.2/hadoop-2.10.2.tar.gz>
  + Zookeeper binary:-

<https://archive.apache.org/dist/zookeeper/zookeeper-3.4.6/zookeeper-3.4.6.tar.gz>

* + Hive binary:-

<https://downloads.apache.org/hive/hive-2.3.9/apache-hive-2.3.9-bin.tar.gz>

* + Hbase\_binary:-

<https://archive.apache.org/dist/hbase/2.4.15/hbase-2.4.15-bin.tar.gz>

* + Spark\_binary:-

<https://archive.apache.org/dist/spark/spark-3.2.2/spark-3.2.2-bin-hadoop2.7.tgz>

* + Flink\_binary:-

<https://archive.apache.org/dist/flink/flink-1.14.4/flink-1.14.4-bin-scala_2.11.tgz>

* + Presto\_binary:-

<https://repo1.maven.org/maven2/com/facebook/presto/presto-server/0.278.1/presto-server-0.278.1.tar.gz>

* Upload the above binaries onto ICS Bucket so that ansible will pull binaries from ICS bucket.
* Download s3cmd on server by using the below commands

| sudo wget <https://github.com/s3tools/s3cmd/releases/download/v2.3.0/s3cmd-2.3.0.tar.gz>  sudo wget<https://bootstrap.pypa.io/pip/2.7/get-pip.py> |
| --- |

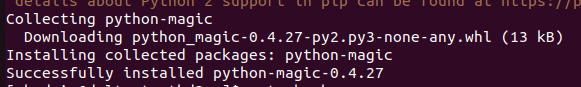
| python2 get-pip.py |
| --- |

* Check the pip version and the expected output should be the as below



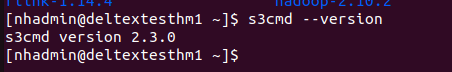
* Install python-magic with this command

| pip install --user python-magic |
| --- |



* Install s3cmd

| tar -xvf s3cmd-2.3.0.tar.gz  cd s3cmd-2.3.0  sudo python ./setup.py install  s3cmd --version |
| --- |



* Add configurations inside .bashrc

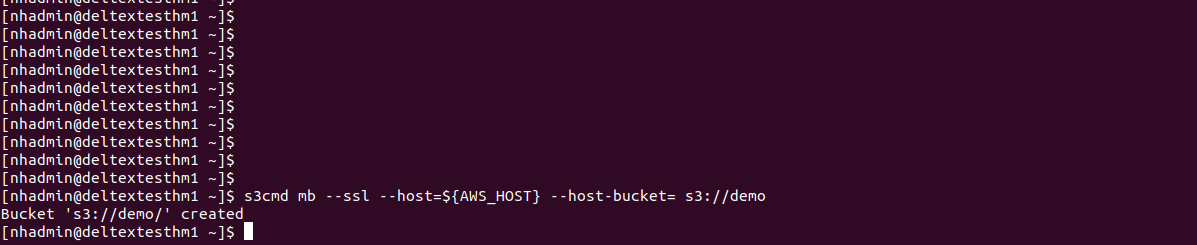
| export AWS\\_ACCESS\\_KEY\\_ID=access\_key  export AWS\\_SECRET\\_ACCESS\\_KEY=secret\_key  export AWS\\_HOST=host\_url  export AWS\\_ENDPOINT=endpoint\_url |
| --- |

* After this run the following command on the terminal

| source .bashrc |
| --- |

* In order to create bucket run the below command on the node you have installed s3cmd

| s3cmd mb --ssl --host=${AWS\_HOST} --host-bucket= s3://bucket-name |
| --- |

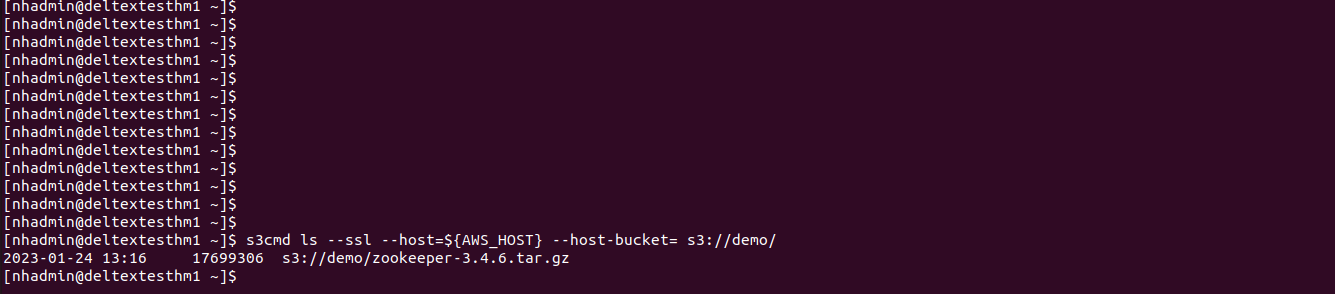


* In order to insert in the bucket run the below command

| s3cmd put filename --host=${AWS\_HOST} --host-bucket= ls s3://<bucket-name> |
| --- |

### 

### 

* In order to list the contents of the bucket run the below command
* 

# 

# 

# 

# 

# 

# Ansible Deployment Steps

## Prerequisites to run hadoop via ansible

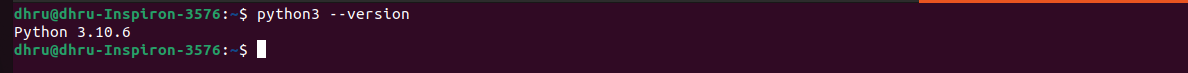
* We have deployed the hadoop ecosystem on RHEL version 7.9(maipo).

| cat /etc/redhat-release |
| --- |

****

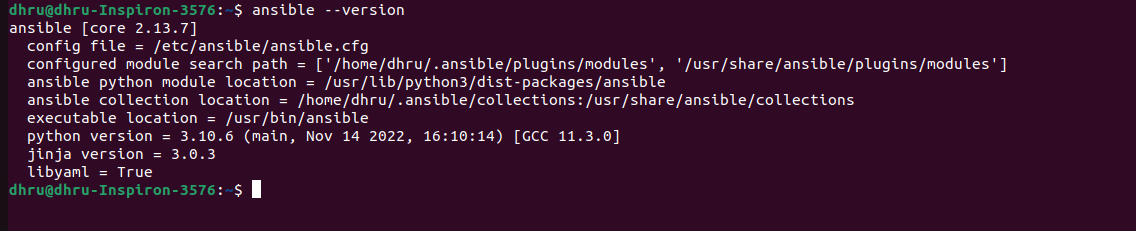
* Ensure that you have python3 and ansible installed on your local system

| python3 --version |
| --- |



* To check the ansible version run the following command:

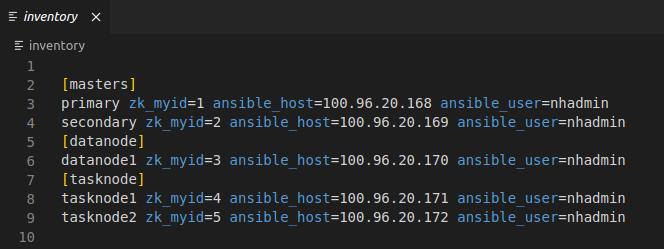
| ansible --version |
| --- |



* If you have not downloaded ansible and python, follow the below link to install
  + For python3:<https://docs.python-guide.org/starting/install3/linux>
  + For ansible:<https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html>

## Setting up ansible configuration

* Inventory file in ansible will look like this :



As you can see from the above example, you can change **ansible\_host** according to the ip address you have available. Similarly if the user is different change the **ansible\_user**

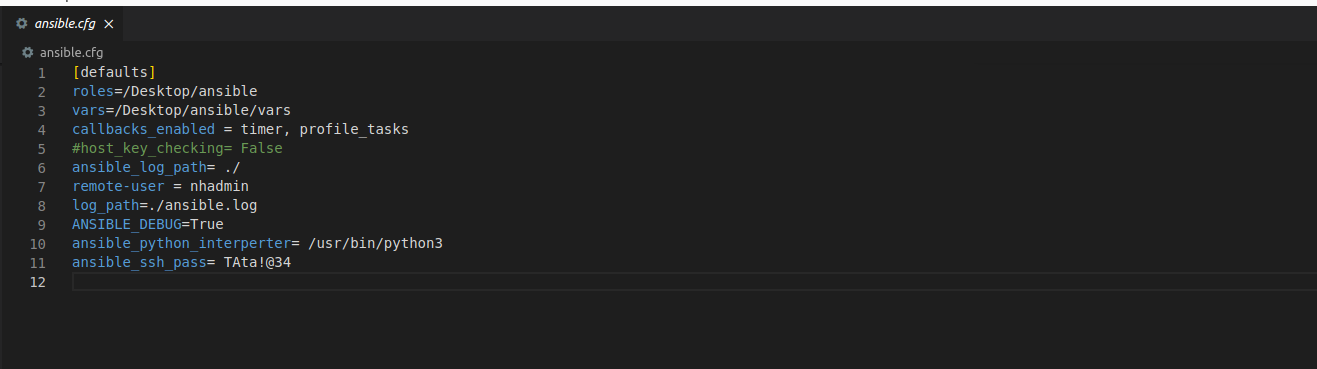
* Setting up Ansible.cfg file:

Give the roles path under "roles" section of the code line

Give path for the global variable under vars section folder

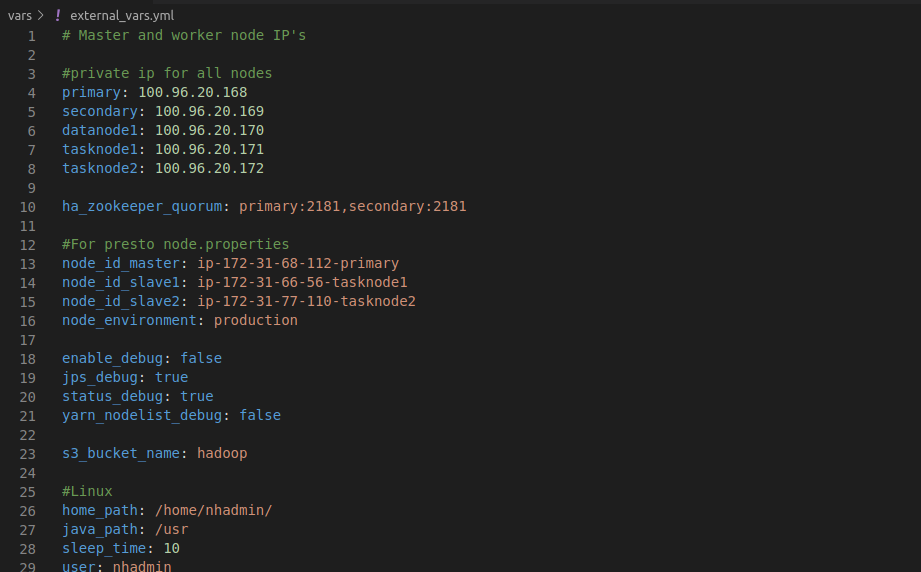
Give ansible\_python\_interperter path present in your system

Give the path for the vault file under global vars section in playbook.yml

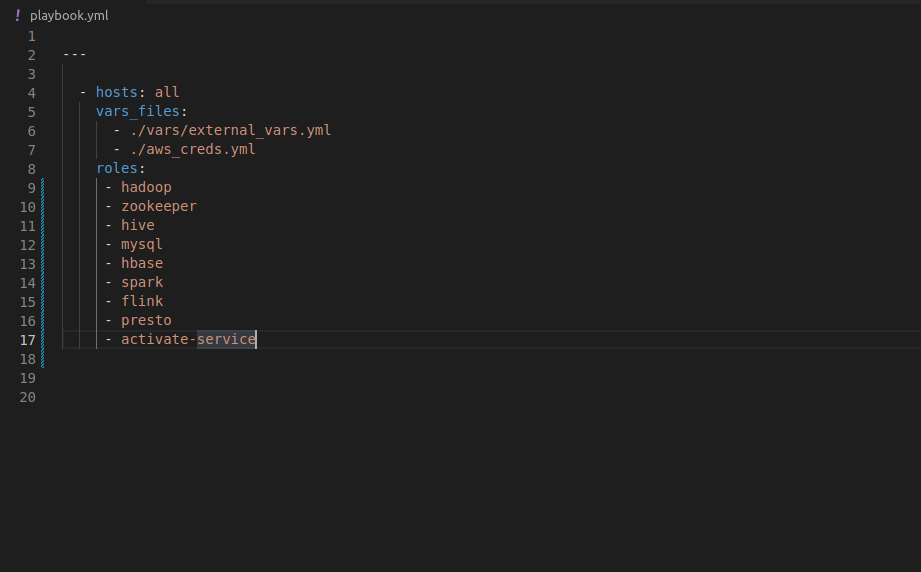


* For global variables file stored under vars folder

Copy the ip of all the instances and paste it in the "vars/external\_vars.yml" present ithe  
 For eg:-  
 *primary: hadoop\_primary\_ip  
 secondary: hadoop\_secondary\_ip  
 datanode1: hadoop\_datanode1\_ip  
 tasknode1: hadoop\_tasknode1\_ip  
 tasknode2: hadoop\_tasknode2\_ip*



* Change the “**home path”** according to the environment you are working on
* Change the “**user”** according to the environment you are working on
* For rhel linux:-
  + Change the “**s3 bucket name”** by the bucket name you created under "s3\_bucket\_name" section
  + Change the node\_id properties for all the three with the private ip except user "-" instead of "."  
    (Note: Ensure that you know which is the master node and which are the slave nodes)
* Playbook.yml



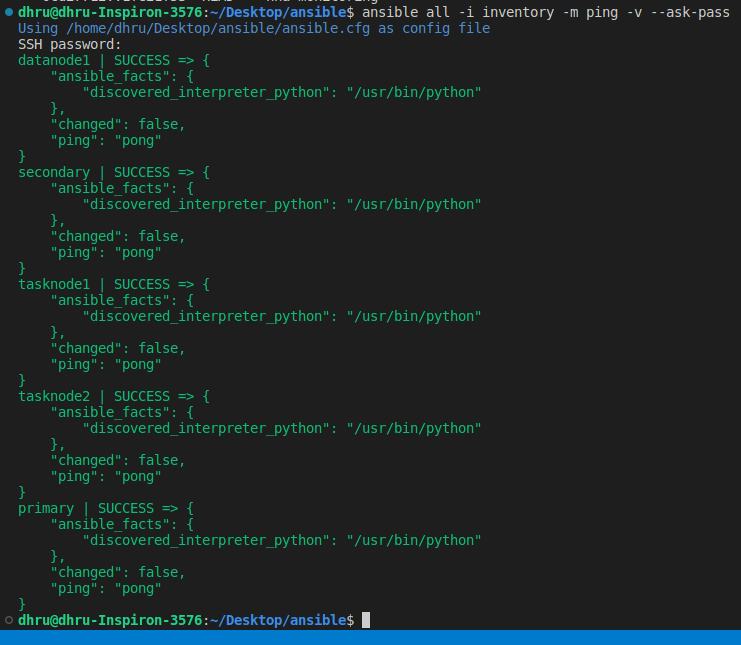
* We have given "hosts: all" as there are 3 groups present in the inventory
* Provided global variable path under "vars\_files:" section
* We provide all the roles we have used in the script under "roles" section
* Now according to your requirement you can choose what components you want in your hadoop ecosystem.
  + For example:-If you want to hadoop, hbase in your system run the following roles in the playbook.yml



* Execute ansible ping command to ensure all the servers are reachable through ansible

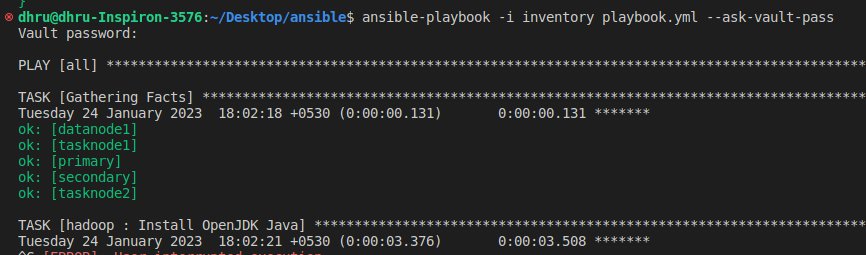
| ansible all -i inventory -m ping -v --ask-pass |
| --- |

(In password prompt you need to enter ssh password for nodes)

****

* Command to start playbook:

| ansible-playbook -i inventory playbook.yml --ask-vault-pass |
| --- |



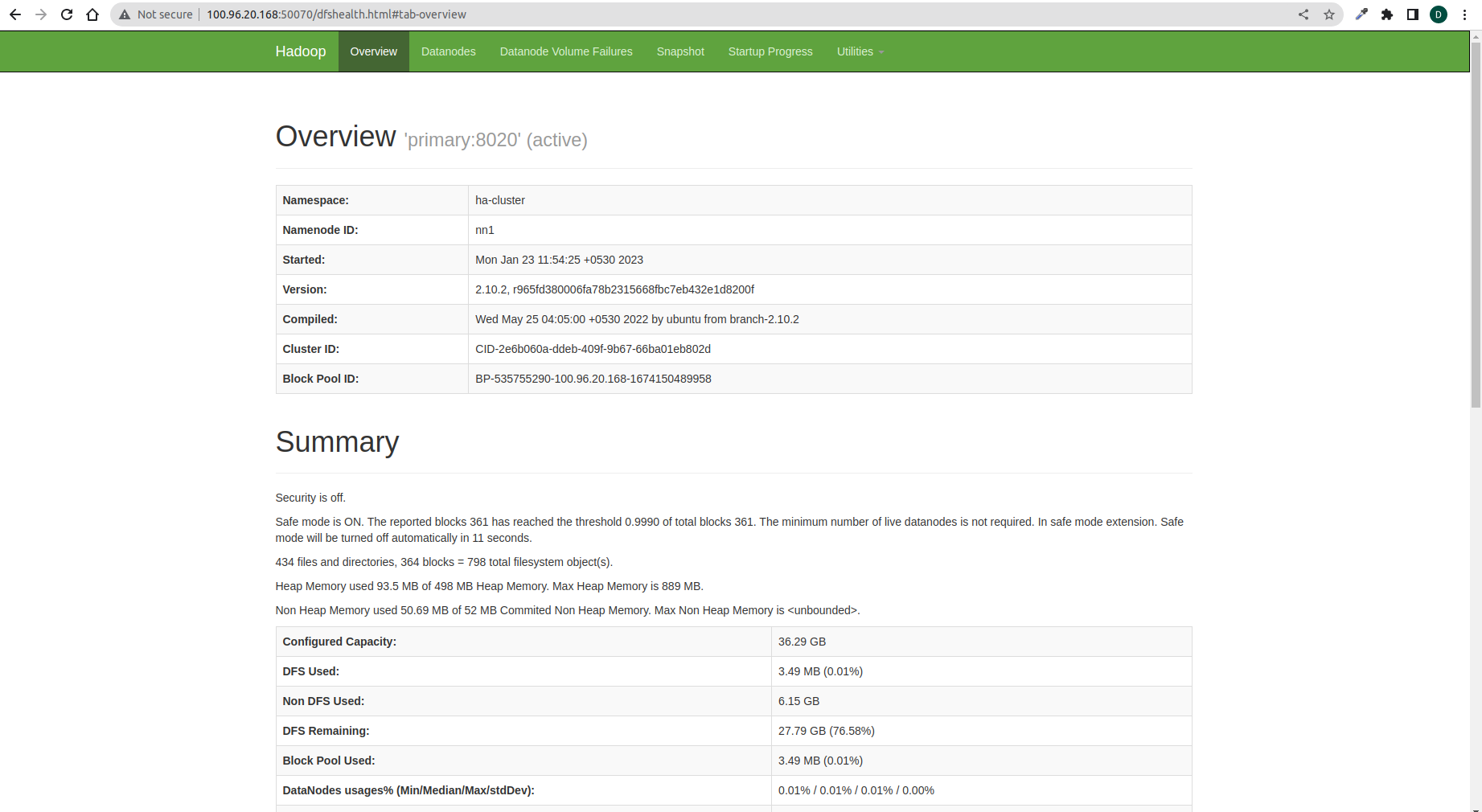
* Ensuring all the services are up and running in respective nodes by executing jps command.  
    
  
* After playbook run is successful check the below ips to get the respective service UI**:**

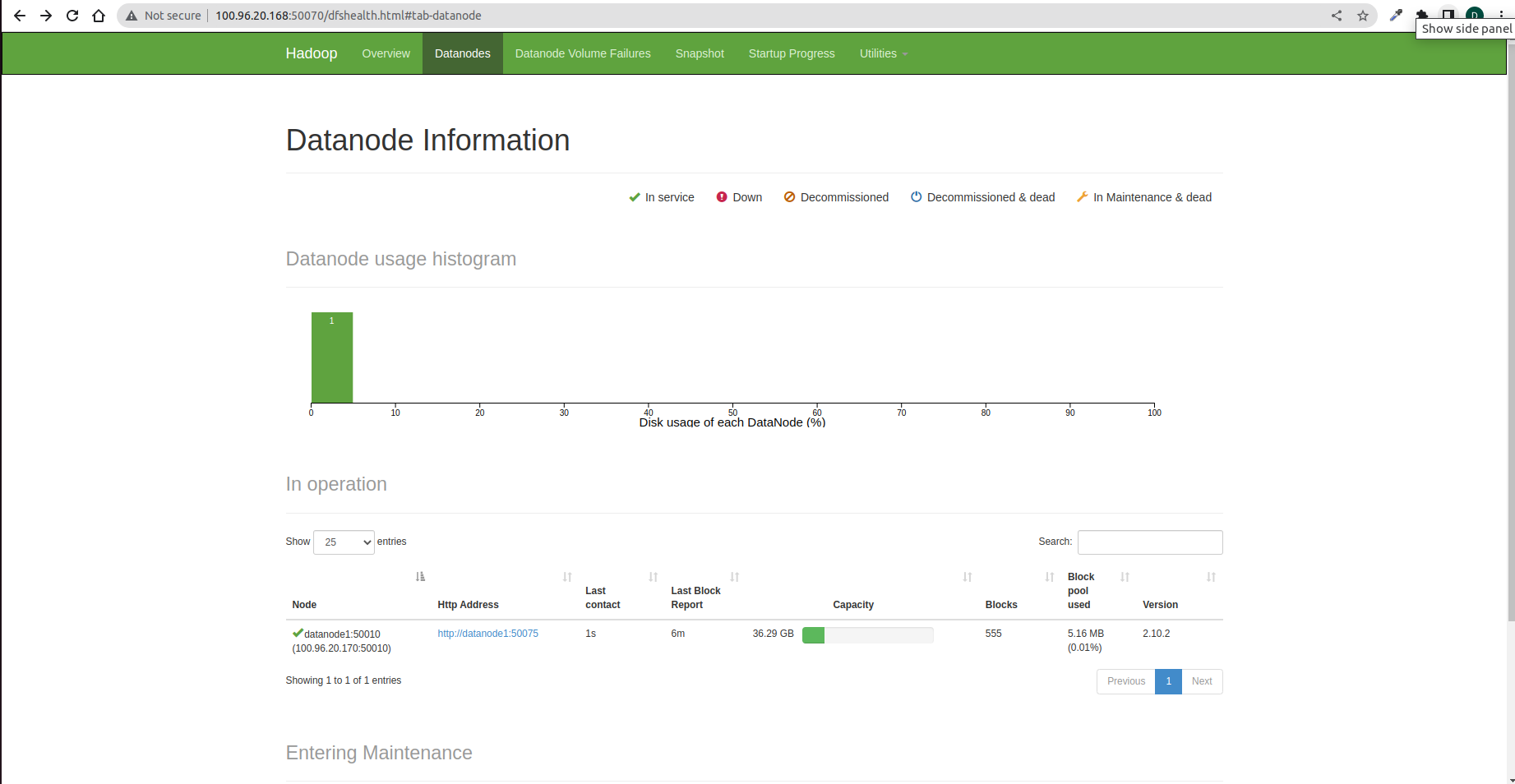
| **Service** | **Hostname** | **Ports** |
| --- | --- | --- |
| Namenode | primary | 50070 |
| Namenode | secondary | 50070 |
| Yarn | primary | 8088 |
| Yarn | secondary | 8088 |
| Flink | primary | 8081 |
| Presto | primary | 8080 |

## Verify the hadoop services are running by accessing the WebUI

* + Check the UI of namenode primary by typing below URLin browser

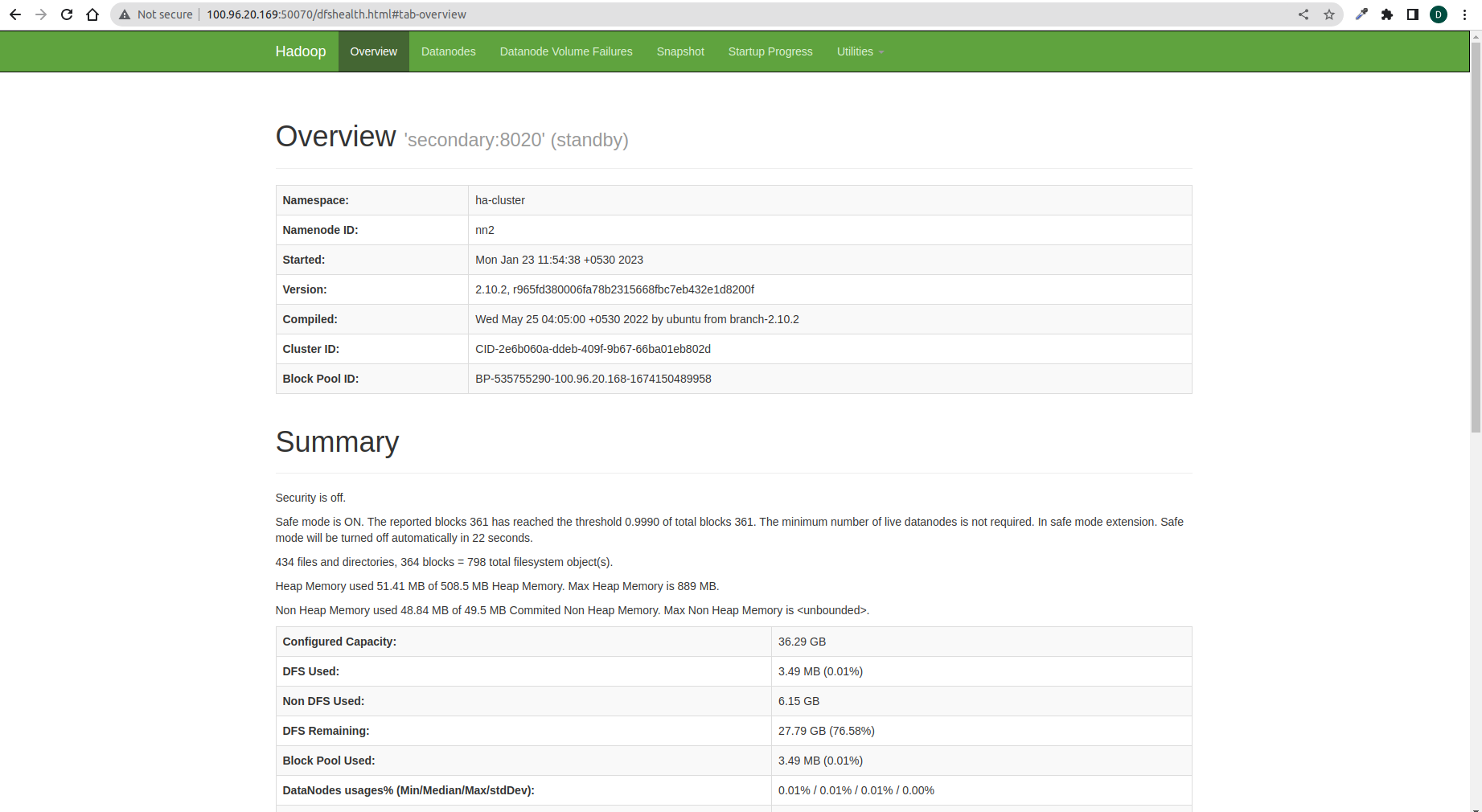
| hadoop\_primary\_ip:50070 |
| --- |

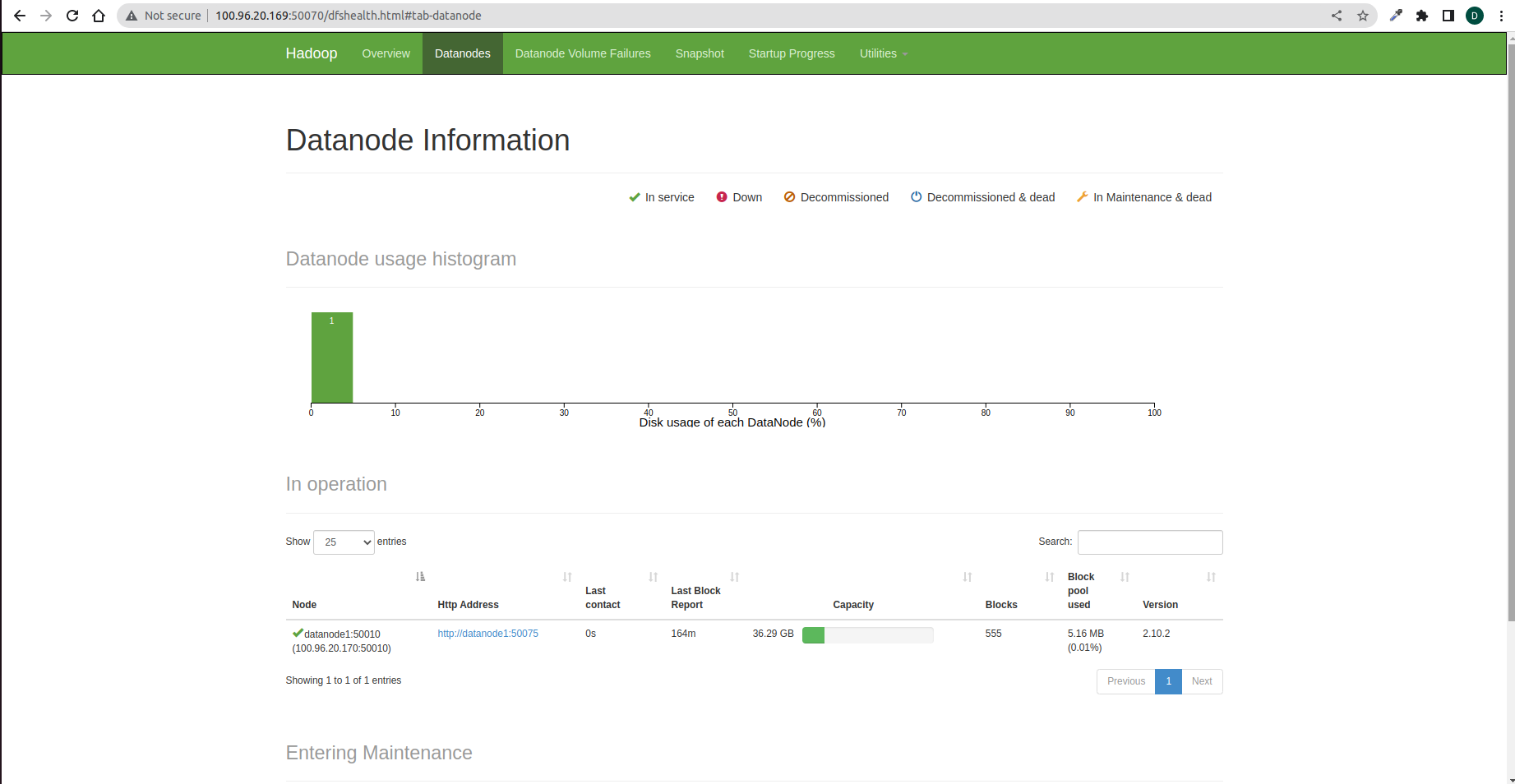




* Check the UI of namenode secondary by typing below URLin browser

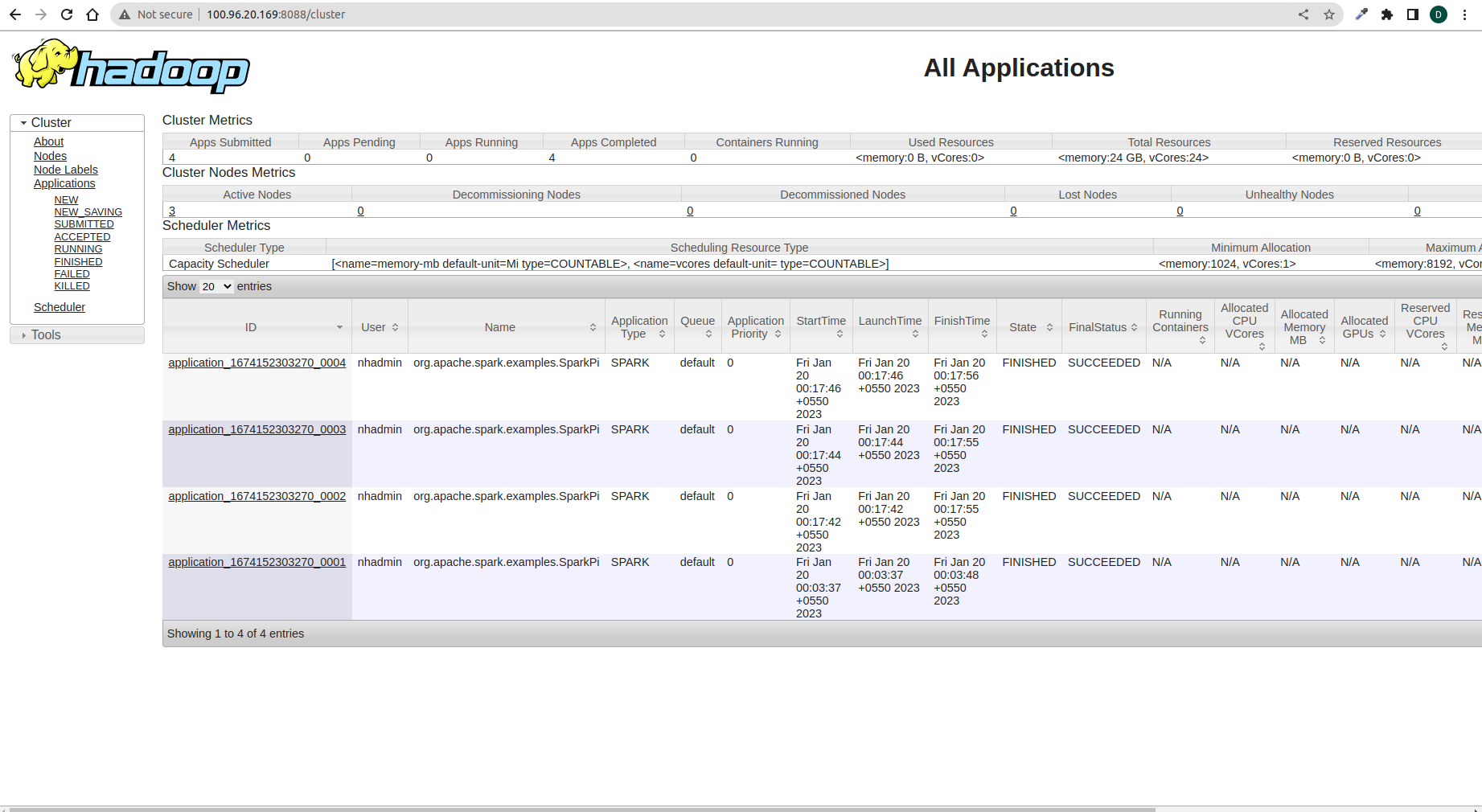
| hadoop\_secondary\_ip:50070 |
| --- |





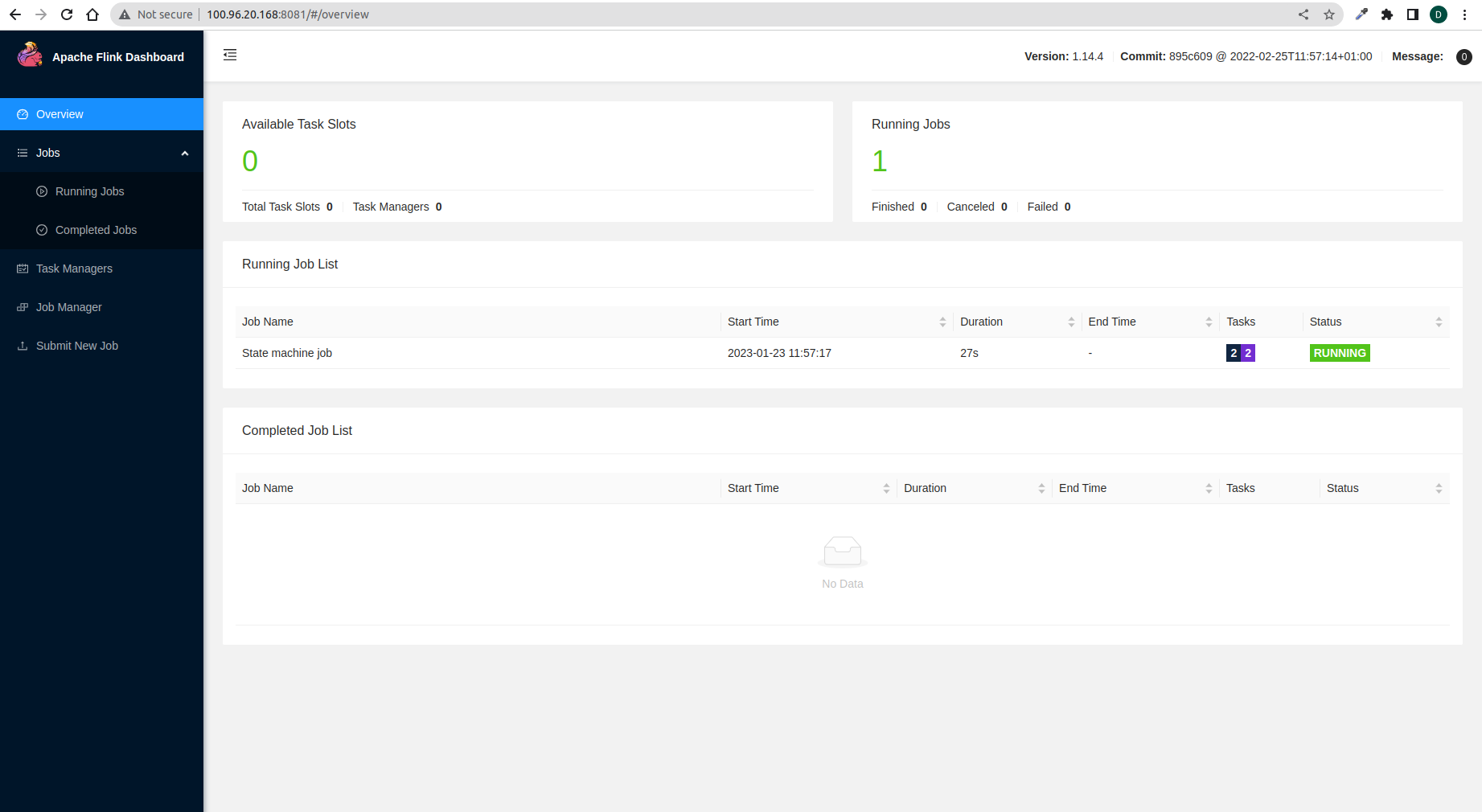
* Check the UI of yarn resource manager either on primary or secondary by typing below URL in browser

| hadoop\_primary\_ip:8088 or hadoop\_secondary\_ip:8088 |
| --- |



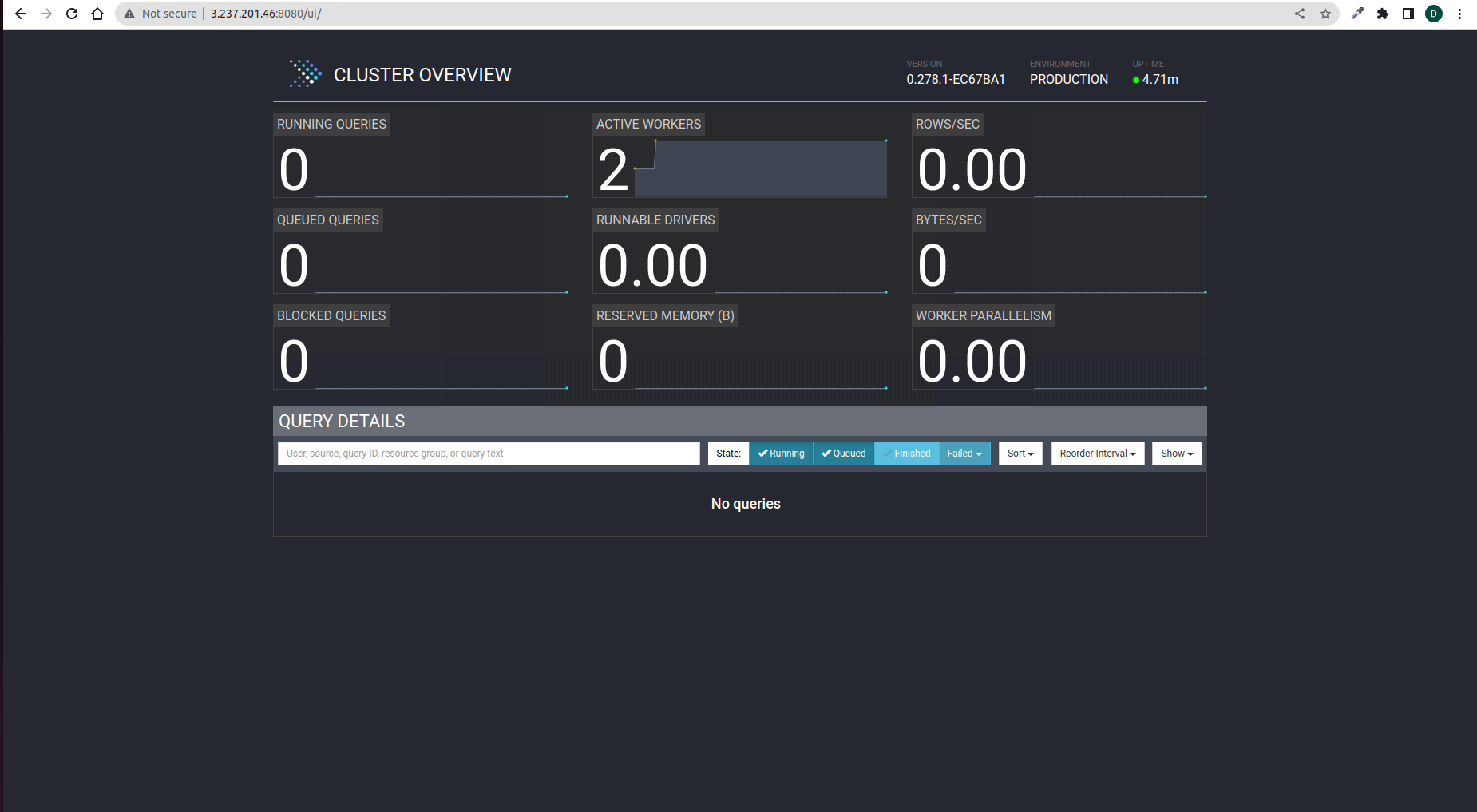
* Check the UI of flink on primary by typing below URL in browser

| hadoop\_primary\_ip:8081 |
| --- |



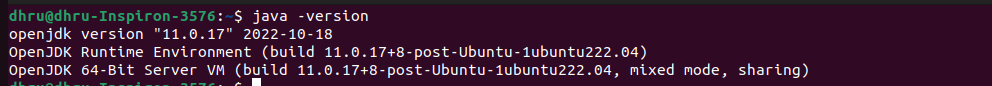
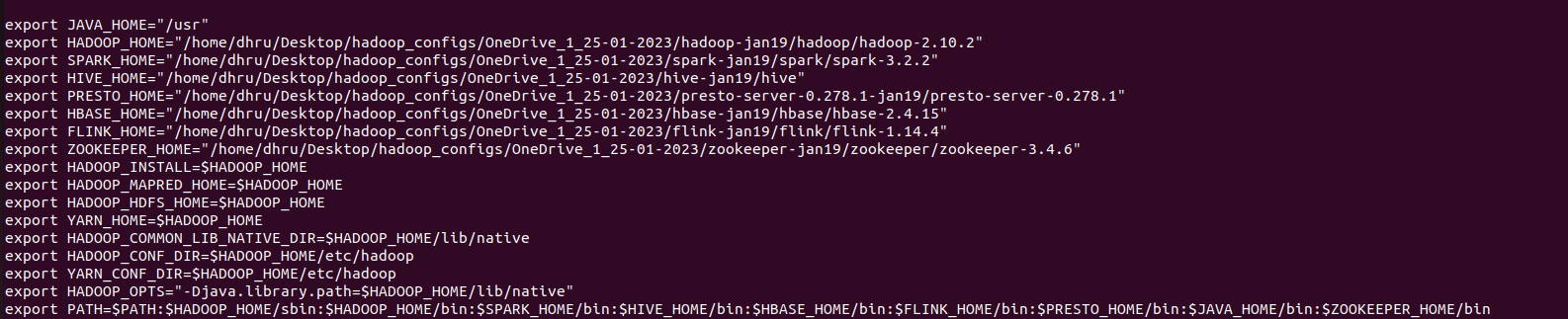
* Check the UI of Presto on primary by typing below URL in browser

| hadoop\_primary\_ip:8080 |
| --- |



# Connecting Hadoop Service Remotely

**Prerequisites**:

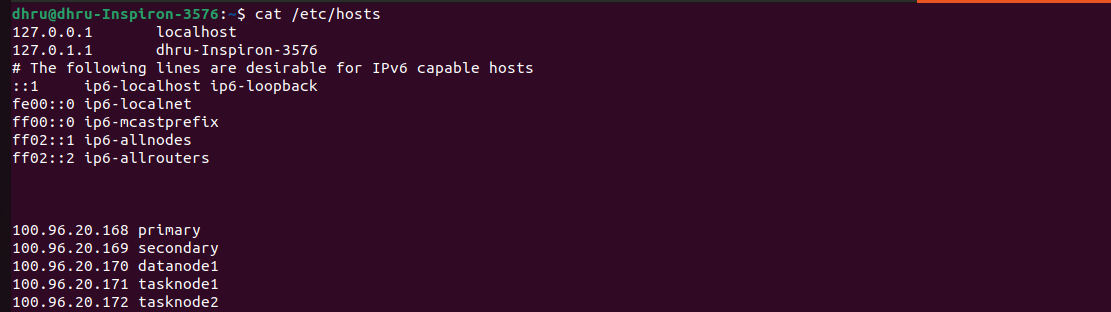
* Ensure that you have java installed in your local system  
    
  
* Download the configured binary files for all components and unzip the binary files
* Copy the configured binary files of all hadoop components on to your local system
* Give the environment path of all the components binaries in .bashrc  
  

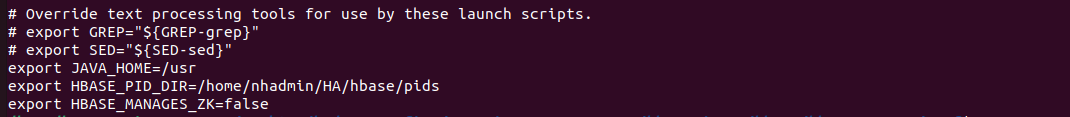
* Run the following command after updating .bashrc

| source .bashrc |
| --- |

* Make a host entry for Hadoop servers inside /etc/hosts

| sudo vi /etc/hosts |
| --- |



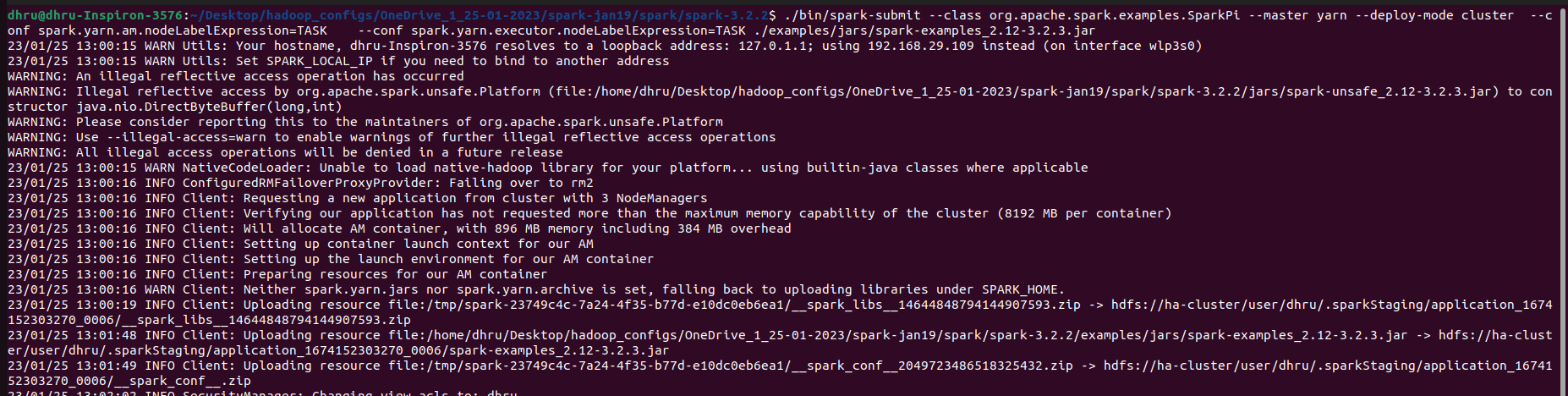
* Change the java path in hbase-env.sh file inside /conf folder in Hbase directory  
  

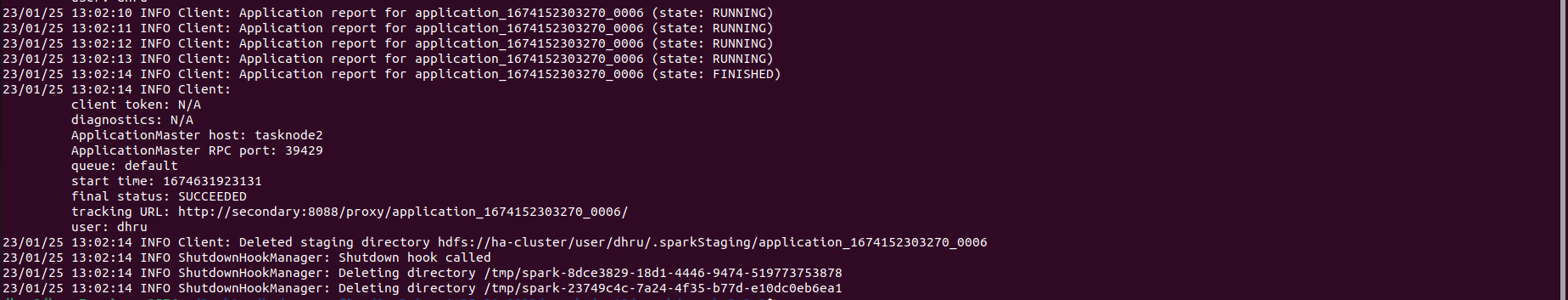
## 

## **Submitting spark job remotely**

* Navigate to spark binary folder present on local machine and execute below command

| ./bin/spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode cluster --conf spark.yarn.am.nodeLabelExpression=TASK --conf spark.yarn.executor.nodeLabelExpression=TASK ./examples/jars/spark-examples\_2.12-3.2.2.jar |
| --- |

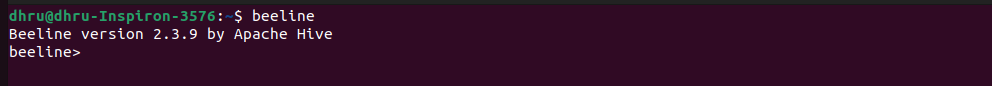




## **Connecting Hive services remotely**

* Run the following command either on home terminal or spark binary folder:-

| beeline |
| --- |



* Run the following command on the beeline terminal:-

| !connect jdbc:hive2://hadoop\_primary\_ip:10000 root |
| --- |

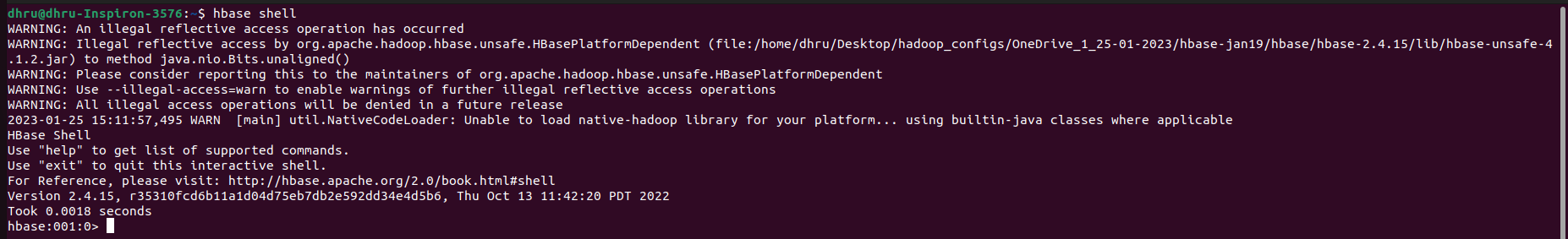
* Here “root” is the username that you have given during mysql setup
* It will prompt for a password,type the password that you have given during mysql setup



## **Run Hbase query remotely**

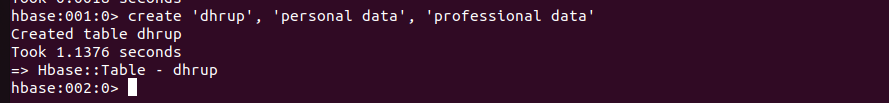
Run the following command in home terminal to start hbase shell:-

| hbase shell |
| --- |



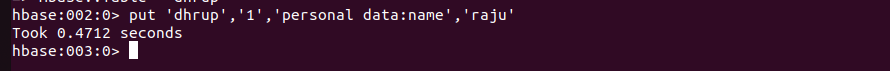
Run the following command to create a table :

| create 'dhrup', 'personal data', 'professional data' |
| --- |



Run the following command to put the contents in the table:

| put 'dhrup','1','personal data:name','raju' |
| --- |



Run the following command to list the contents of the table:

| scan 'dhrup' |
| --- |

