INDEX

Hadoop Ecosystem

Prerequisites

1. Port Opening for internal cluster communication

?????

2. 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

```
[nhadmin@deltextesthm1 ~]$ pip -V
pip 20.3.4 from /usr/lib/python2.7/site-packages/pip (python 2.7)
```

- Install python-magic with this command

```
pip install --user python-magic
```

```
Collecting python-magic
  Downloading python_magic-0.4.27-py2.py3-none-any.whl (13 kB)
Installing collected packages: python-magic
Successfully installed python-magic-0.4.27
```

- Install s3cmd

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

```
[nhadmin@deltextesthm1 ~]$ s3cmd --version
s3cmd version 2.3.0
[nhadmin@deltextesthm1 ~]$
```

- Add configurations inside .bashrc

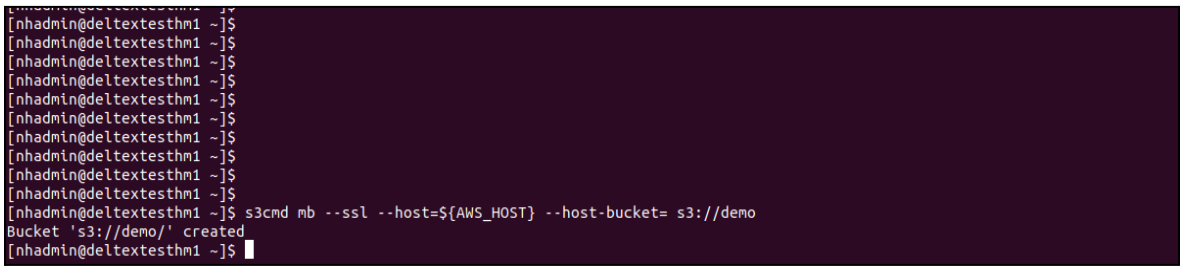
```
export AWS\_ACCESS\_KEY\_ID=access_key
export AWS\_SECRET\_ACCESS\_KEY=secret_key
export AWS\_HOST=host_url
export AWS\_ENDPOINT=endpoint_url
```

- 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>
```

```
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$ s3cmd put zookeeper-3.4.6.tar.gz --host=${AWS_HOST} --host-bucket= ls s3://demo/
upload: 'zookeeper-3.4.6.tar.gz' -> 's3://demo/zookeeper-3.4.6.tar.gz'  [part 1 of 2, 15MB] [1 of 1]
 15728640 of 15728640   100% in    2s     5.27 MB/s  done
upload: 'zookeeper-3.4.6.tar.gz' -> 's3://demo/zookeeper-3.4.6.tar.gz'  [part 2 of 2, 1924KB] [1 of 1]
 1970666 of 1970666   100% in    0s     2.06 MB/s  done
[nhadmin@deltextesthm1 ~]$
```

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

```
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$
[nhadmin@deltextesthm1 ~]$ s3cmd ls --ssl --host=${AWS_HOST} --host-bucket= s3://demo/
2023-01-24 13:16     17699306  s3://demo/zookeeper-3.4.6.tar.gz
[nhadmin@deltextesthm1 ~]$
```

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
```

```
[nhadmin@deltextesthm1 ~]$ cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.9 (Maipo)
```

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

```
python3 --version
```

```
dhru@dhru-Inspiron-3576:~$ python3 --version
Python 3.10.6
dhru@dhru-Inspiron-3576:~$
```

- To check the ansible version run the following command:

```
ansible --version
```

```
dhru@dhru-Inspiron-3576:~$ ansible --version
ansible [core 2.13.7]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/dhru/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/dhru/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.6 (main, Nov 14 2022, 16:10:14) [GCC 11.3.0]
  jinja version = 3.0.3
  libyaml = True
dhru@dhru-Inspiron-3576:~$
```

- 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 :

```
inventory

[masters]
primary zk_myid=1 ansible_host=100.96.20.168 ansible_user=nhadmin
secondary zk_myid=2 ansible_host=100.96.20.169 ansible_user=nhadmin
[datanode]
datanode1 zk_myid=3 ansible_host=100.96.20.170 ansible_user=nhadmin
[tasknode]
tasknode1 zk_myid=4 ansible_host=100.96.20.171 ansible_user=nhadmin
tasknode2 zk_myid=5 ansible_host=100.96.20.172 ansible_user=nhadmin
```

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

```
ansible.cfg
[defaults]
roles=/Desktop/ansible
vars=/Desktop/ansible/vars
callbacks_enabled = timer, profile_tasks
#host_key_checking= False
ansible_log_path= ./
remote-user = nhadmin
log_path=./ansible.log
ANSIBLE_DEBUG=True
ansible_python_interperter= /usr/bin/python3
ansible_ssh_pass= TAta!@34
```

- 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

```yaml
vars > ! external_vars.yml
# Master and worker node IP's

#private ip for all nodes
primary: 100.96.20.168
secondary: 100.96.20.169
datanode1: 100.96.20.170
tasknode1: 100.96.20.171
tasknode2: 100.96.20.172

ha_zookeeper_quorum: primary:2181,secondary:2181

#For presto node.properties
node_id_master: ip-172-31-68-112-primary
node_id_slave1: ip-172-31-66-56-tasknode1
node_id_slave2: ip-172-31-77-110-tasknode2
node_environment: production

enable_debug: false
jps_debug: true
status_debug: true
yarn_nodelist_debug: false

s3_bucket_name: hadoop

#Linux
home_path: /home/nhadmin/
java_path: /usr
sleep_time: 10
user: nhadmin
```

- 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

```yaml
playbook.yml

---

- hosts: all
  vars_files:
    - ./vars/external_vars.yml
    - ./aws_creds.yml
  roles:
   - hadoop
   - zookeeper
   - hive
   - mysql
   - hbase
   - spark
   - flink
   - presto
   - activate-service
```

- 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

```yaml
playbook.yml

---

- hosts: all
  vars_files:
    - ./vars/external_vars.yml
    - ./aws_creds.yml
  roles:
   - hadoop
   - zookeeper
  #  - hive
  #  - mysql
   - hbase
  #  - spark
  #  - flink
  #  - presto
   - activate-service
```

- 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)

```
dhru@dhru-Inspiron-3576:~/Desktop/ansible$ ansible all -i inventory -m ping -v --ask-pass
Using /home/dhru/Desktop/ansible/ansible.cfg as config file
SSH password:
datanode1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
secondary | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
tasknode1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
tasknode2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
primary | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
dhru@dhru-Inspiron-3576:~/Desktop/ansible$
```

- Command to start playbook:

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

```
dhru@dhru-Inspiron-3576:~/Desktop/ansible$ ansible-playbook -i inventory playbook.yml --ask-vault-pass
Vault password:

PLAY [all] *********************************************************************************************

TASK [Gathering Facts] *********************************************************************************
Tuesday 24 January 2023  18:02:18 +0530 (0:00:00.131)       0:00:00.131 *******
ok: [datanode1]
ok: [tasknode1]
ok: [primary]
ok: [secondary]
ok: [tasknode2]

TASK [hadoop : Install OpenJDK Java] *******************************************************************
Tuesday 24 January 2023  18:02:21 +0530 (0:00:03.376)       0:00:03.508 *******
```

- Ensuring all the services are up and running in respective nodes by executing jps command.

```
[nhadmin@deltextesthm1 ~]$ jps
9393 DFSZKFailoverController
67107 QuorumPeerMain
123843 ResourceManager
99891 NameNode
8870 JournalNode
68699 HMaster
61069 StandaloneSessionClusterEntrypoint
30749 RunJar
10990 PrestoServer
11023 Jps
[nhadmin@deltextesthm1 ~]$
```

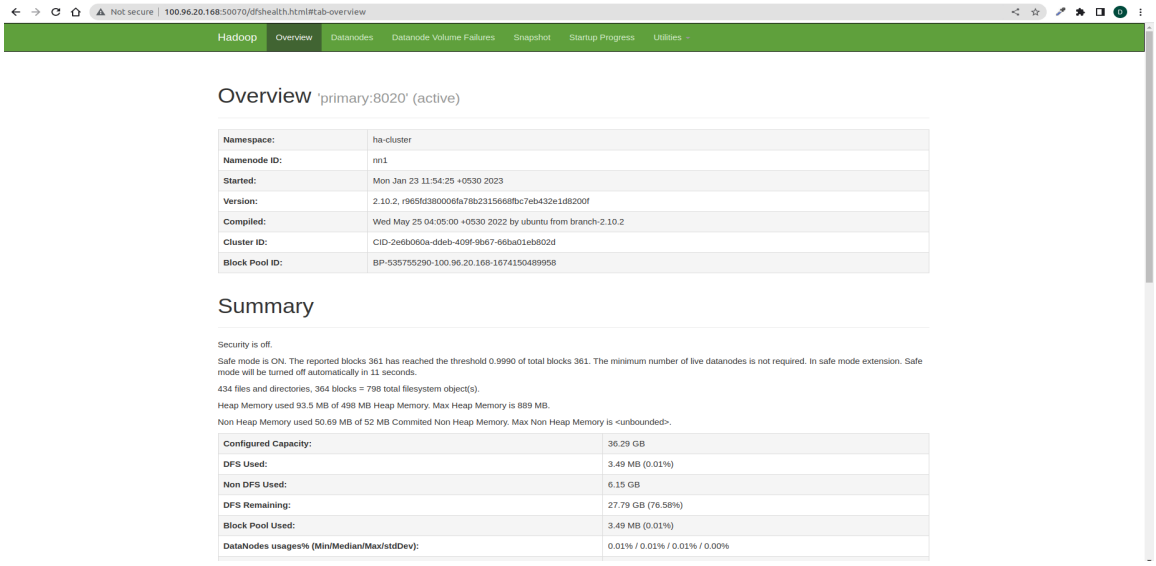
- After playbook run is successful check the below ips to get the respective service UI**:**

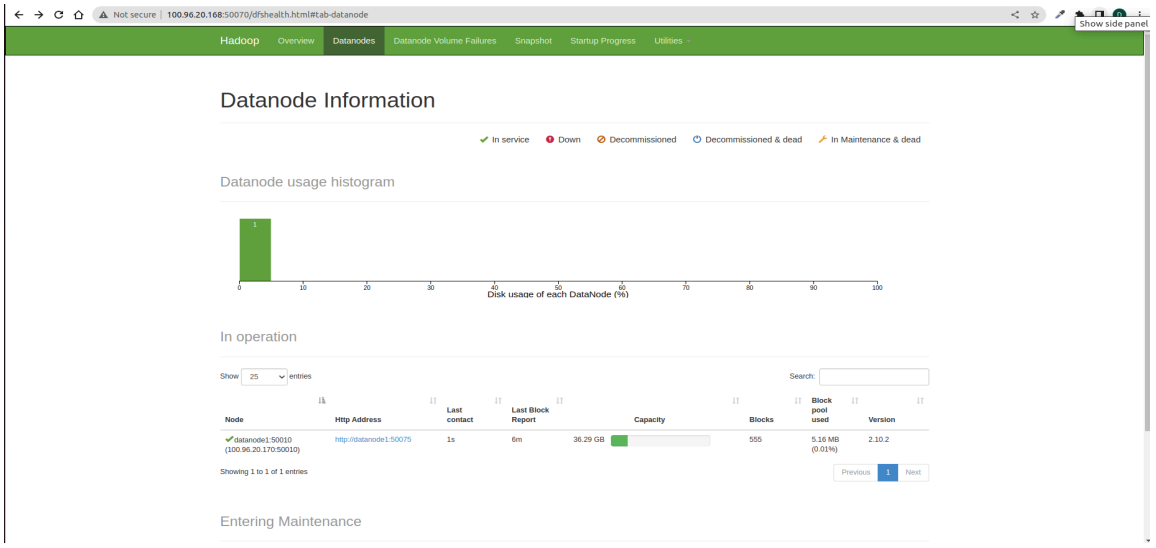
Service	**Hostname**	**Ports**
Namenode	primary	50070
Namenode	secondary	50070
Yarn	primary	8088
Yarn	secondary	8088
Flink	primary	8081
Presto	primary	8080

Verify the hadoop services are running by accessing the WebUI

- Check the UI of namenode primary by typing below URLin browser

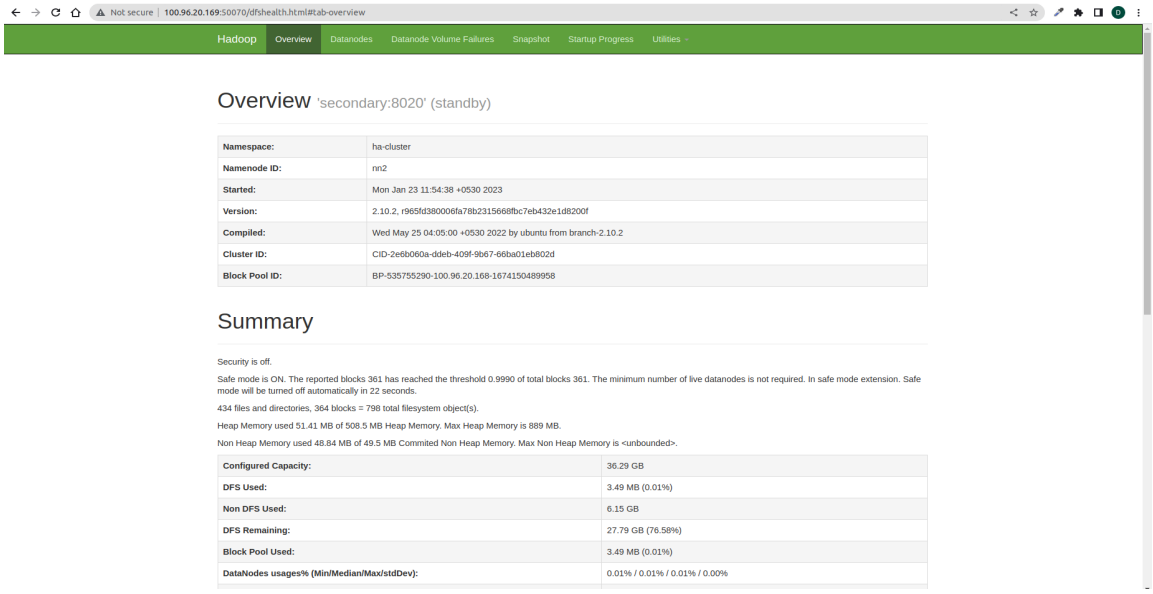
```
hadoop_primary_ip:50070
```

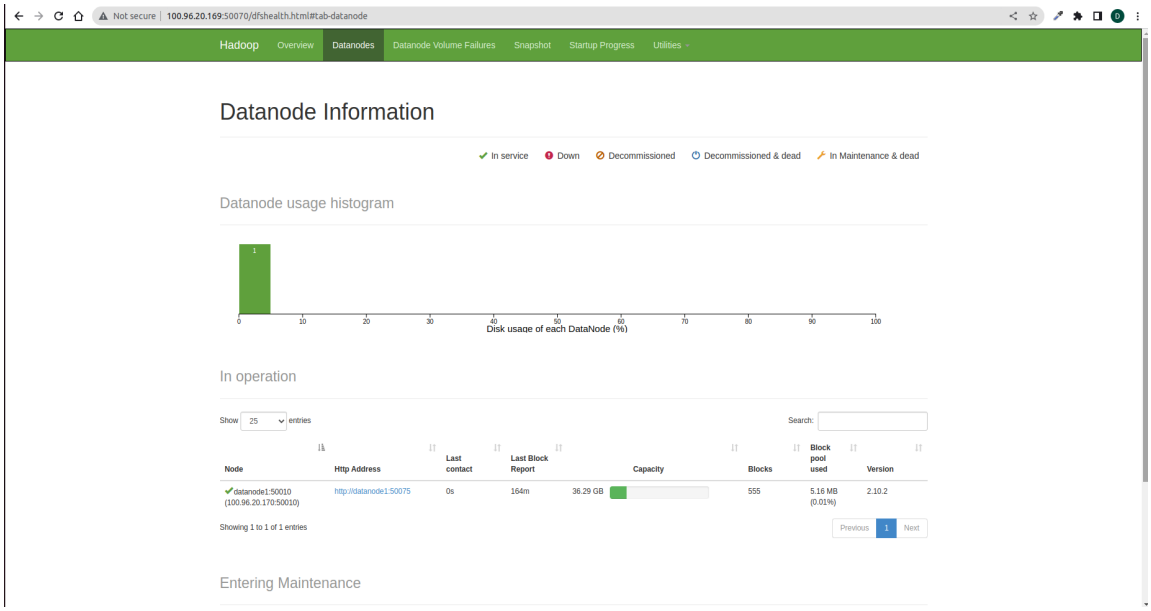




- Check the UI of namenode secondary by typing below URLin browser

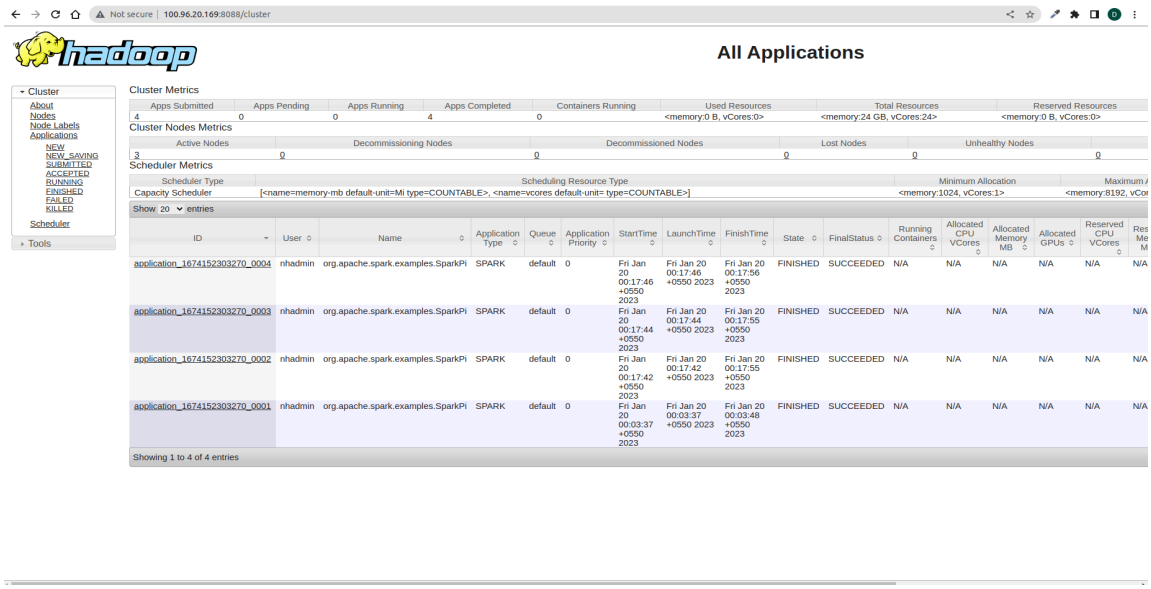
```
hadoop_secondary_ip:50070
```





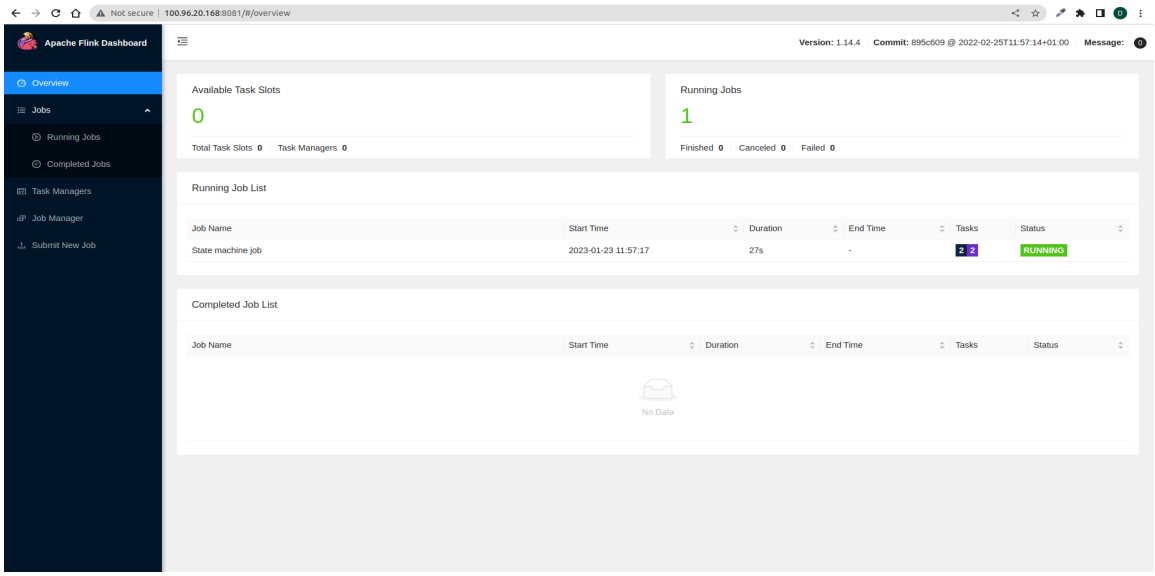
- Check the UI of yarn resource manager either on primary or secondary by typing below URL in browser

hadoop_primary_ip:8088 or hadoop_secondary_ip:8088



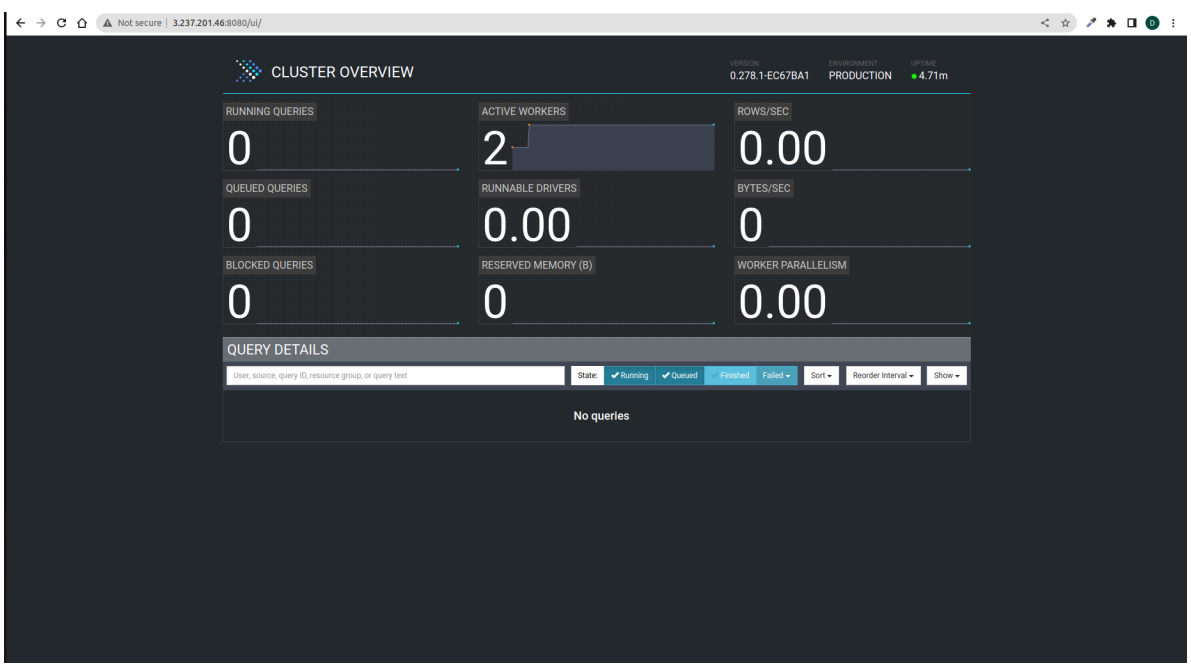
- Check the UI of flink on primary by typing below URL in browser

```
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

```
dhru@dhru-Inspiron-3576:~$ java -version
openjdk version "11.0.17" 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-1ubuntu222.04, mixed mode, sharing)
```

- 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

```
export JAVA_HOME="/usr"
export HADOOP_HOME="/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/hadoop-jan19/hadoop/hadoop-2.10.2"
export SPARK_HOME="/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/spark-jan19/spark/spark-3.2.2"
export HIVE_HOME="/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/hive-jan19/hive"
export PRESTO_HOME="/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/presto-server-0.278.1-jan19/presto-server-0.278.1"
export HBASE_HOME="/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/hbase-jan19/hbase/hbase-2.4.15"
export FLINK_HOME="/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/flink-jan19/flink/flink-1.14.4"
export ZOOKEEPER_HOME="/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/zookeeper-jan19/zookeeper/zookeeper-3.4.6"
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin:$SPARK_HOME/bin:$HIVE_HOME/bin:$HBASE_HOME/bin:$FLINK_HOME/bin:$PRESTO_HOME/bin:$JAVA_HOME/bin:$ZOOKEEPER_HOME/bin
```

- Run the following command after updating .bashrc

```
source .bashrc
```

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

```
sudo vi /etc/hosts
```

```
dhru@dhru-Inspiron-3576:~$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       dhru-Inspiron-3576
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters



100.96.20.168 primary
100.96.20.169 secondary
100.96.20.170 datanode1
100.96.20.171 tasknode1
100.96.20.172 tasknode2
```

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

```
# Override text processing tools for use by these launch scripts.
# export GREP="${GREP-grep}"
# export SED="${SED-sed}"
export JAVA_HOME=/usr
export HBASE_PID_DIR=/home/nhadmin/HA/hbase/pids
export HBASE_MANAGES_ZK=false
```

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
```

```
dhru@dhru-Inspiron-3576:~/Desktop/hadoop_configs/OneDrive_1_25-01-2023/spark-jan19/spark/spark-3.2.2$ ./bin/spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode cluster  --c
onf spark.yarn.am.nodeLabelExpression=TASK    --conf spark.yarn.executor.nodeLabelExpression=TASK ./examples/jars/spark-examples_2.12-3.2.3.jar
23/01/25 13:00:15 WARN Utils: Your hostname, dhru-Inspiron-3576 resolves to a loopback address: 127.0.1.1; using 192.168.29.109 instead (on interface wlp3s0)
23/01/25 13:00:15 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/spark-jan19/spark/spark-3.2.2/jars/spark-unsafe_2.12-3.2.3.jar) to con
structor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
23/01/25 13:00:15 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
23/01/25 13:00:16 INFO ConfiguredRMFailoverProxyProvider: Failing over to rm2
23/01/25 13:00:16 INFO Client: Requesting a new application from cluster with 3 NodeManagers
23/01/25 13:00:16 INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (8192 MB per container)
23/01/25 13:00:16 INFO Client: Will allocate AM container, with 896 MB memory including 384 MB overhead
23/01/25 13:00:16 INFO Client: Setting up container launch context for our AM
23/01/25 13:00:16 INFO Client: Setting up the launch environment for our AM container
23/01/25 13:00:16 INFO Client: Preparing resources for our AM container
23/01/25 13:00:16 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
23/01/25 13:00:19 INFO Client: Uploading resource file:/tmp/spark-23749c4c-7a24-4f35-b77d-e10dc0eb6ea1/__spark_libs__1464848794144907593.zip -> hdfs://ha-cluster/user/dhru/.sparkStaging/application_1674
152303270_0006/__spark_libs__1464848794144907593.zip
23/01/25 13:01:48 INFO Client: Uploading resource file:/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/spark-jan19/spark/spark-3.2.2/examples/jars/spark-examples_2.12-3.2.3.jar -> hdfs://ha-clust
er/user/dhru/.sparkStaging/application_1674152303270_0006/spark-examples_2.12-3.2.3.jar
23/01/25 13:01:49 INFO Client: Uploading resource file:/tmp/spark-23749c4c-7a24-4f35-b77d-e10dc0eb6ea1/__spark_conf__2049723486518325432.zip -> hdfs://ha-cluster/user/dhru/.sparkStaging/application_16741
52303270_0006/__spark_conf__.zip
```

```
23/01/25 13:02:10 INFO Client: Application report for application_1674152303270_0006 (state: RUNNING)
23/01/25 13:02:11 INFO Client: Application report for application_1674152303270_0006 (state: RUNNING)
23/01/25 13:02:12 INFO Client: Application report for application_1674152303270_0006 (state: RUNNING)
23/01/25 13:02:13 INFO Client: Application report for application_1674152303270_0006 (state: RUNNING)
23/01/25 13:02:14 INFO Client: Application report for application_1674152303270_0006 (state: FINISHED)
23/01/25 13:02:14 INFO Client:
         client token: N/A
         diagnostics: N/A
         ApplicationMaster host: tasknode2
         ApplicationMaster RPC port: 39429
         queue: default
         start time: 1674631923131
         final status: SUCCEEDED
         tracking URL: http://secondary:8088/proxy/application_1674152303270_0006/
         user: dhru
23/01/25 13:02:14 INFO Client: Deleted staging directory hdfs://ha-cluster/user/dhru/.sparkStaging/application_1674152303270_0006
23/01/25 13:02:14 INFO ShutdownHookManager: Shutdown hook called
23/01/25 13:02:14 INFO ShutdownHookManager: Deleting directory /tmp/spark-8dce3829-18d1-4446-9474-519773753878
23/01/25 13:02:14 INFO ShutdownHookManager: Deleting directory /tmp/spark-23749c4c-7a24-4f35-b77d-e10dc0eb6ea1
```

Connecting Hive services remotely

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

```
beeline
```

```
dhru@dhru-Inspiron-3576:~$ beeline
Beeline version 2.3.9 by Apache Hive
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

```
dhru@dhru-Inspiron-3576:~/Desktop/hadoop_configs/OneDrive_1_25-01-2023/spark-jan19/spark/spark-3.2.2/bin$ beeline
Beeline version 2.3.9 by Apache Hive
beeline> !connect jdbc:hive2://100.96.20.168:10000 root
Connecting to jdbc:hive2://100.96.20.168:10000
Enter password for jdbc:hive2://100.96.20.168:10000: ***********
23/01/25 13:17:15 INFO Utils: Supplied authorities: 100.96.20.168:10000
23/01/25 13:17:15 INFO Utils: Resolved authority: 100.96.20.168:10000
Connected to: Apache Hive (version 2.3.9)
Driver: Hive JDBC (version 2.3.9)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://100.96.20.168:10000> show databases;
+----------------+
| database_name  |
+----------------+
| default        |
+----------------+
1 row selected (1.233 seconds)
0: jdbc:hive2://100.96.20.168:10000>
```

Run Hbase query remotely

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

```
hbase shell
```

```
dhru@dhru-Inspiron-3576: $ hbase shell
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.hbase.unsafe.HBasePlatformDependent (file:/home/dhru/Desktop/hadoop_configs/OneDrive_1_25-01-2023/hbase-jan19/hbase/hbase-2.4.15/lib/hbase-unsafe-4
.1.2.jar) to method java.nio.Bits.unaligned()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.hbase.unsafe.HBasePlatformDependent
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2023-01-25 15:11:57,495 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.15, r35310fcd6b11a1d04d75eb7db2e592dd34e4d5b6, Thu Oct 13 11:42:20 PDT 2022
Took 0.0018 seconds
hbase:001:0>
```

Run the following command to create a table :

```
create 'dhrup', 'personal data', 'professional data'
```

```
hbase:001:0> create 'dhrup', 'personal data', 'professional data'
Created table dhrup
Took 1.1376 seconds
=> Hbase::Table - dhrup
hbase:002:0>
```

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

```
put 'dhrup','1','personal data:name','raju'
```

```
hbase:002:0> put 'dhrup','1','personal data:name','raju'
Took 0.4712 seconds
hbase:003:0>
```

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

```
scan 'dhrup'
```

```
hbase:003:0> scan 'dhrup'
ROW                                                    COLUMN+CELL
 1                                                     column=personal data:name, timestamp=2023-01-25T15:16:37.688, value=raju
1 row(s)
Took 0.0993 seconds
hbase:004:0>
```