Flipkart SRE Assignment : Hadoop Cluster Setup using Ansible

1. Introduction to Hadoop

Hadoop is an open-source framework that enables distributed storage and processing of large datasets using the MapReduce programming model. It is designed to scale horizontally across a cluster of machines and provides fault tolerance and high availability. Hadoop consists of several components, including:

- **HDFS (Hadoop Distributed File System)**: A distributed storage system.
- **YARN (Yet Another Resource Negotiator)**: Manages resources and job scheduling.
- **MapReduce**: A programming model for parallel data processing.

2. Why We Choose Ansible for Cluster Setup

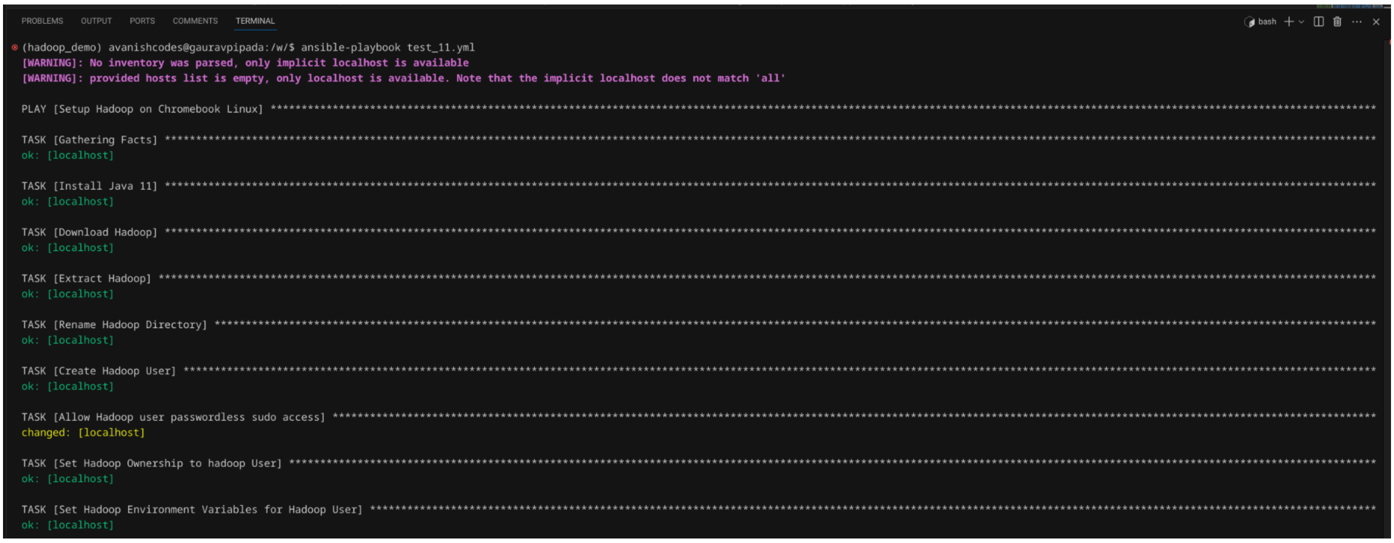
Ansible is an automation tool that allows us to configure and manage systems efficiently. We used Ansible for setting up the Hadoop cluster due to the following reasons:

- **Agentless Architecture**: No need to install agents on target nodes.
- **Ease of Use**: YAML-based playbooks simplify automation.
- **Scalability**: Can be used to manage multiple nodes easily.
- **Consistency**: Ensures all nodes are configured identically.
- **Idempotency**: Ensures tasks execute only when needed.

3. Demo of cluster-setup using Ansible (Single Node)

Why Single Node Setup?

A single-node setup allows Hadoop to run on a single machine without needing multiple VMs or additional hardware, making it ideal for my personal system.



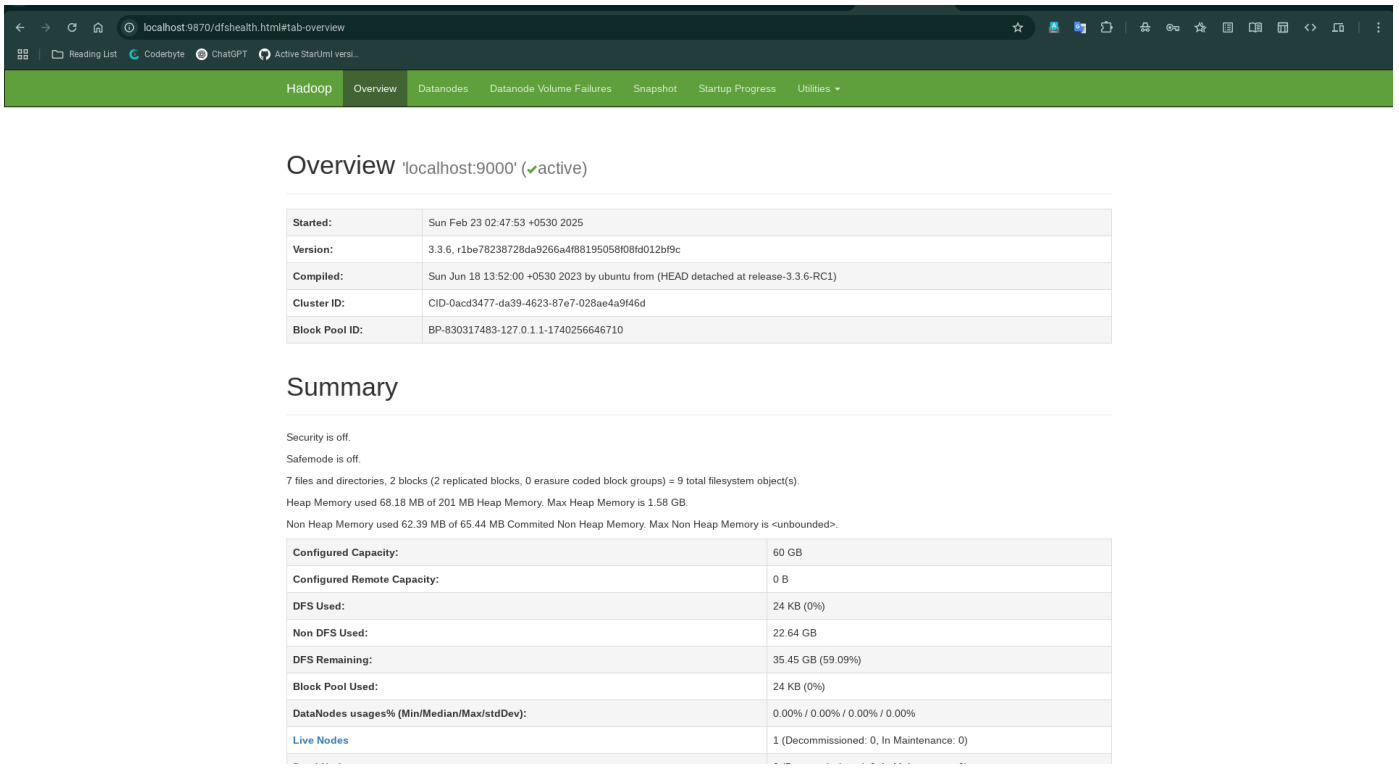
```
(hadoop_demo) avanishcodes@gauravpipada:/w/$ ansible-playbook test_11.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Setup Hadoop on Chromebook Linux] *************************************************************************************************************************************************************

TASK [Gathering Facts] ******************************************************************************************************************************************************************************
ok: [localhost]

TASK [Install Java 11] ******************************************************************************************************************************************************************************
ok: [localhost]

TASK [Download Hadoop] ******************************************************************************************************************************************************************************
ok: [localhost]

TASK [Extract Hadoop] *******************************************************************************************************************************************************************************
ok: [localhost]

TASK [Rename Hadoop Directory] **********************************************************************************************************************************************************************
ok: [localhost]

TASK [Create Hadoop User] ***************************************************************************************************************************************************************************
ok: [localhost]

TASK [Allow Hadoop user passwordless sudo access] ***************************************************************************************************************************************************
changed: [localhost]

TASK [Set Hadoop Ownership to hadoop User] **********************************************************************************************************************************************************
ok: [localhost]

TASK [Set Hadoop Environment Variables for Hadoop User] *********************************************************************************************************************************************
ok: [localhost]
```

```
TASK [Ensure JAVA_HOME is set in hadoop-env.sh] ********************************************************************************************************************
ok: [localhost]

TASK [Configure Hadoop (core-site.xml)] ****************************************************************************************************************************
ok: [localhost]

TASK [Configure Hadoop (hdfs-site.xml)] ****************************************************************************************************************************
ok: [localhost]

TASK [Format HDFS] *************************************************************************************************************************************************
changed: [localhost]

TASK [Start Hadoop Services as hadoop User] ************************************************************************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************************************************************************************************
localhost                  : ok=16   changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

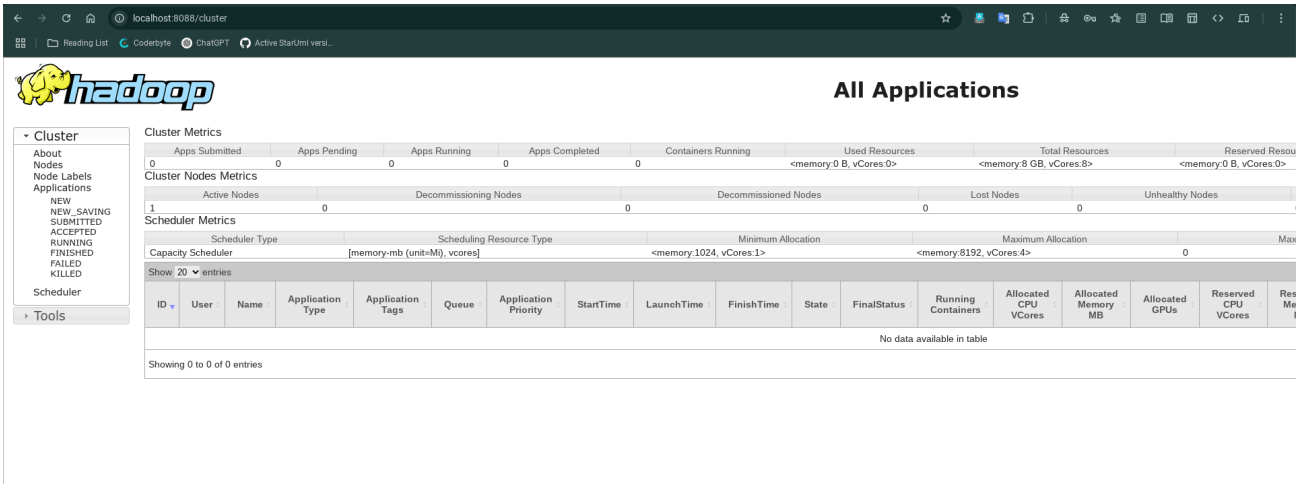
```
hadoop@penguin:/home/hadoop_demo$ jps
8405 Jps
7783 ResourceManager
7880 NodeManager
7037 NameNode
7582 SecondaryNameNode
hadoop@penguin:/home/hadoop_demo$ hdfs dfsadmin -report
Configured Capacity: 0 (0 B)
Present Capacity: 0 (0 B)
DFS Remaining: 0 (0 B)
DFS Used: 0 (0 B)
DFS Used%: 0.00%
Replicated Blocks:
        Under replicated blocks: 0
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
```

After the cluster setup, the following web consoles are accessible for monitoring:

- **HDFS NameNode UI** (`http://localhost:9870/dfshealth.html#tab-overview`) – Provides HDFS health and file system details.



- **YARN ResourceManager UI** (`http://localhost:8088/cluster`) – Displays cluster resource usage and running applications.



4. Ansible Code :

```
- name: Setup Hadoop on Chromebook Linux
  hosts: localhost
  connection: local
  become: yes
  tasks:

    - name: Install Java 11
      apt:
        name: openjdk-11-jdk
        state: present

    - name: Download Hadoop
      get_url:
        url:
"https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz"
        dest: "/tmp/hadoop-3.3.6.tar.gz"

    - name: Extract Hadoop
      unarchive:
        src: "/tmp/hadoop-3.3.6.tar.gz"
        dest: "/usr/local/"
        remote_src: yes

    - name: Rename Hadoop Directory
      command: mv /usr/local/hadoop-3.3.6 /usr/local/hadoop
      args:
        creates: /usr/local/hadoop

    - name: Create Hadoop User
      user:
        name: hadoop
        shell: /bin/bash
        createhome: yes
```

```
- name: Allow Hadoop user passwordless sudo access
  copy:
    dest: /etc/sudoers.d/hadoop
    content: "hadoop ALL=(ALL) NOPASSWD: ALL"
    mode: '0440'

- name: Set Hadoop Ownership to hadoop User
  file:
    path: /usr/local/hadoop
    owner: hadoop
    group: hadoop
    recurse: yes

- name: Set Hadoop Environment Variables for Hadoop User
  blockinfile:
    path: /home/hadoop/.bashrc
    block: |
      export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
      export HADOOP_HOME=/usr/local/hadoop
      export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
      export HDFS_NAMENODE_USER="hadoop"
      export HDFS_DATANODE_USER="hadoop"
      export HDFS_SECONDARYNAMENODE_USER="hadoop"
      export YARN_RESOURCEMANAGER_USER="hadoop"
      export YARN_NODEMANAGER_USER="hadoop"

- name: Ensure JAVA_HOME is set in hadoop-env.sh
  lineinfile:
    path: /usr/local/hadoop/etc/hadoop/hadoop-env.sh
    regexp: '^export JAVA_HOME='
    line: 'export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64'

- name: Configure Hadoop (core-site.xml)
  copy:
    dest: /usr/local/hadoop/etc/hadoop/core-site.xml
    content: |
      <configuration>
          <property>
              <name>fs.defaultFS</name>
              <value>hdfs://localhost:9000</value>
          </property>
      </configuration>

- name: Configure Hadoop (hdfs-site.xml)
  copy:
    dest: /usr/local/hadoop/etc/hadoop/hdfs-site.xml
    content: |
      <configuration>
          <property>
              <name>dfs.replication</name>
              <value>1</value>
```

```
            </property>
        </configuration>

    - name: Format HDFS
      shell: |
        echo Y | sudo -u hadoop JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
/usr/local/hadoop/bin/hdfs namenode -format

    - name: Start Hadoop Services as hadoop User
      shell: |
        sudo -u hadoop bash -lc "/usr/local/hadoop/sbin/start-dfs.sh &&
/usr/local/hadoop/sbin/start-yarn.sh"
```

5. Operations performed on cluster (Create, Upload, Modify and Delete)

5.1 Create a Directory in HDFS

```
PROBLEMS    OUTPUT    PORTS    COMMENTS    TERMINAL

(hadoop_demo) avanishcodes@gauravpipada:/w/$ sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -mkdir -p /user/hadoop/test_dir
(hadoop_demo) avanishcodes@gauravpipada:/w/$ sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -ls /user/hadoop
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2025-02-23 02:59 /user/hadoop/test_dir
(hadoop_demo) avanishcodes@gauravpipada:/w/$
```

5.2 Upload ,Modify and Delete Operations :

```
PROBLEMS    OUTPUT    PORTS    COMMENTS    TERMINAL

(hadoop_demo) avanishcodes@gauravpipada:/w/$ echo "This is a test file. for Flipkart SRE Assignment" | sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -put - /user/hadoop/flipkart_test/flipkart_test_file.
txt
(hadoop_demo) avanishcodes@gauravpipada:/w/$ sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -cat /user/hadoop/flipkart_test/flipkart_test_file.txt
This is a test file. for Flipkart SRE Assignment
(hadoop_demo) avanishcodes@gauravpipada:/w/$ echo "Additional content for Flipkart SRE Assignment." | sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -appendToFile - /user/hadoop/flipkart_test/flipkart_te
st_file.txt
(hadoop_demo) avanishcodes@gauravpipada:/w/$ sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -cat /user/hadoop/flipkart_test/flipkart_test_file.txt
This is a test file. for Flipkart SRE Assignment
Additional content for Flipkart SRE Assignment.
(hadoop_demo) avanishcodes@gauravpipada:/w/$ sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -rm /user/hadoop/flipkart_test/flipkart_test_file.txt
Deleted /user/hadoop/flipkart_test/flipkart_test_file.txt
(hadoop_demo) avanishcodes@gauravpipada:/w/$ sudo -u hadoop /usr/local/hadoop/bin/hdfs dfs -ls /user/hadoop/flipkart_test
(hadoop_demo) avanishcodes@gauravpipada:/w/$
```

6. Challenges Faced During Setup

While setting up the Hadoop cluster using Ansible, we encountered several challenges:

- **Cluster ID Mismatch**: The DataNode was repeatedly getting down due to a mismatch in the Cluster ID. We had to reinitialize the HDFS and ensure all nodes had the correct configuration.
- **Hadoop Compatibility with Java 17**: Initially, we attempted to run Hadoop with Java 17, but it was not compatible. We had to switch to JDK 11 to ensure proper functionality.
- **User Privilege Issues in Ansible**: Running Hadoop-related commands as the `hadoop` user required proper privilege escalation, which we resolved using `become` in Ansible.

7. Learnings from the Project

This project provided valuable insights into:

- **Setting Up a Hadoop Cluster**: Understanding Hadoop architecture and configuring different components.
- **Basic Hadoop Operations**: Running essential commands for creating, uploading and deleting the files and directories ,checking node status, and managing services.
- **Ansible Playbook Development**: Writing Ansible scripts for automation and troubleshooting issues.
- **Debugging Configuration Issues**: Identifying and resolving errors related to Cluster ID mismatches and Java version compatibility.
- **User Management in Linux**: Managing users and permissions to run Hadoop services correctly.

GitHub Project : [GauravPipada/Flipkart-SRE-Assesment]

Conclusion

This project helped in understanding Hadoop and automating its setup using Ansible. Overcoming challenges like Cluster ID mismatches, Java version issues, and privilege escalation provided hands-on experience in troubleshooting distributed systems. Future work can focus on optimizing performance and implementing a fully HA-enabled Hadoop cluster.