**PR 1: Write python script to Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes.**

To write a Python script that installs and runs Hive, and then uses Hive to create, alter, and drop databases, tables, views, functions, and indexes, we need to follow a few steps:

1. **Install Hive and its dependencies**: This involves downloading and setting up Java, Hadoop, and Hive.
2. **Start the required services**: Hadoop services, Hive Metastore, and HiveServer2.
3. **Connect to Hive using Python**: Use the pyhive library to interact with Hive.
4. **Perform the desired operations**: Create, alter, and drop databases, tables, views, functions, and indexes.

### Step 1: Install Prerequisites

Below is the Python script to automate the installation process. This script assumes you are running on a Linux-based system with sudo privileges.

**import os**

**import subprocess**

**def run\_command(command):**

**subprocess.run(command, shell=True, check=True)**

**# Install Java**

**run\_command('sudo apt-get update')**

**run\_command('sudo apt-get install -y default-jdk')**

**# Download and Install Hadoop**

**run\_command('wget https://downloads.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz')**

**run\_command('tar -xzf hadoop-3.3.4.tar.gz')**

**run\_command('sudo mv hadoop-3.3.4 /usr/local/hadoop')**

**# Set Hadoop environment variables**

**os.environ['HADOOP\_HOME'] = '/usr/local/hadoop'**

**os.environ['PATH'] += ':/usr/local/hadoop/bin:/usr/local/hadoop/sbin'**

**# Download and Install Hive**

**run\_command('wget https://downloads.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz')**

**run\_command('tar -xzf apache-hive-3.1.3-bin.tar.gz')**

**run\_command('sudo mv apache-hive-3.1.3-bin /usr/local/hive')**

**# Set Hive environment variables**

**os.environ['HIVE\_HOME'] = '/usr/local/hive'**

**os.environ['PATH'] += ':/usr/local/hive/bin'**

**# Start Hadoop Services**

**run\_command('start-dfs.sh')**

**run\_command('start-yarn.sh')**

**# Initialize Hive Schema**

**run\_command('schematool -dbType derby -initSchema')**

**# Start Hive Metastore and HiveServer2**

**run\_command('nohup hive --service metastore &')**

**run\_command('nohup hive --service hiveserver2 &')**

### Step 2: Using Hive to Perform Operations

Next, we will use the pyhive library to connect to Hive and perform the desired operations.

First, install pyhive and pandas:

**pip install pyhive pandas**

Then, use the following Python script to create, alter, and drop databases, tables, views, functions, and indexes in Hive:

**from pyhive import hive**

**import pandas as pd**

**# Connect to Hive**

**conn = hive.Connection(host='localhost', port=10000, username='hadoop')**

**cursor = conn.cursor()**

**# Create Database**

**cursor.execute("CREATE DATABASE IF NOT EXISTS test\_db")**

**# Use Database**

**cursor.execute("USE test\_db")**

**# Create Table**

**cursor.execute("""**

**CREATE TABLE IF NOT EXISTS employees (**

**id INT,**

**name STRING,**

**age INT,**

**department STRING**

**)**

**ROW FORMAT DELIMITED**

**FIELDS TERMINATED BY ','**

**""")**

**# Insert Data**

**cursor.execute("INSERT INTO employees VALUES (1, 'John Doe', 30, 'HR')")**

**cursor.execute("INSERT INTO employees VALUES (2, 'Jane Doe', 25, 'Finance')")**

**# Alter Table**

**cursor.execute("ALTER TABLE employees ADD COLUMNS (salary DOUBLE)")**

**# Create View**

**cursor.execute("CREATE VIEW IF NOT EXISTS employee\_view AS SELECT name, department FROM employees")**

**# Create Function (UDF)**

**cursor.execute("""**

**CREATE FUNCTION my\_upper AS 'org.apache.hadoop.hive.ql.udf.generic.GenericUDFUpper'**

**USING JAR 'hdfs:///user/hive/lib/hive-contrib-\*.jar'**

**""")**

**# Create Index**

**cursor.execute("CREATE INDEX IF NOT EXISTS idx\_name ON TABLE employees (name) AS 'COMPACT' WITH DEFERRED REBUILD")**

**# Drop Index**

**cursor.execute("DROP INDEX IF EXISTS idx\_name ON employees")**

**# Drop View**

**cursor.execute("DROP VIEW IF EXISTS employee\_view")**

**# Drop Table**

**cursor.execute("DROP TABLE IF EXISTS employees")**

**# Drop Database**

**cursor.execute("DROP DATABASE IF EXISTS test\_db")**

**cursor.close()**

**conn.close()**

### Explanation

1. **Install Prerequisites**: The script installs Java, Hadoop, and Hive. It sets up the environment variables required for Hadoop and Hive.
2. **Start Services**: The script starts the Hadoop DFS and YARN services, initializes the Hive schema, and starts the Hive Metastore and HiveServer2 services.
3. **Connect to Hive**: Using the pyhive library, the script connects to the Hive server.
4. **Perform Operations**: The script creates, alters, and drops databases, tables, views, functions, and indexes as specified.

Make sure to adapt the paths and configurations according to your environment.