Hadoop Installation

Chronological Installation of Hadoop, Hive, and Spark

Step 0: Pre-Installation Checks

Check OS

```
uname -a    # Linux / macOS check
```

1. Make sure you have at least **8 GB RAM** and **20 GB disk**.

Install Java (mandatory for Hadoop, Hive, Spark)

```
java -version
```

2.
   - ○ Hadoop 3.x, Hive 4.x, Spark 3.x require **Java 8 or 11** (not Java 17+).

If missing, install OpenJDK:

```
sudo apt-get update
sudo apt-get install openjdk-11-jdk -y
```

   - ○

Set `JAVA_HOME`:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH
```

   - ○

Check SSH (for Hadoop pseudo-distributed mode)

```
ssh localhost
```

If password prompt appears → configure passwordless SSH:

```
ssh-keygen -t rsa -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

3.

Step 1: Install Hadoop

Download & Extract

```
wget
https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
tar -xvzf hadoop-3.3.6.tar.gz
mv hadoop-3.3.6 ~/hadoop
```

1.

Set Environment Variables (add to `~/.bashrc`)

```
export HADOOP_HOME=~/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

2.
3. **Edit Hadoop Configs**
   - `hadoop-env.sh` → set `JAVA_HOME`

```
core-site.xml
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

○

```
hdfs-site.xml
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>

<value>file:///home/youruser/hadoopdata/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>

<value>file:///home/youruser/hadoopdata/datanode</value>
  </property>
</configuration>
```

○

```
mapred-site.xml
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

○

```
yarn-site.xml
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

○

Format Namenode

```
hdfs namenode -format
```

Start Hadoop

```
start-dfs.sh
start-yarn.sh
jps    # should see NameNode, DataNode, ResourceManager, NodeManager
```

Step 2: Install Hive

Download & Extract

```
wget https://downloads.apache.org/hive/hive-4.0.0/apache-hive-4.0.0-bin.tar.gz
tar -xvzf apache-hive-4.0.0-bin.tar.gz
mv apache-hive-4.0.0-bin ~/hive
```

1.

Set Environment Variables

```
export HIVE_HOME=~/hive
export PATH=$PATH:$HIVE_HOME/bin
```

2.
3. **Configure Metastore (PostgreSQL recommended over Derby)**
    - Install PostgreSQL / MySQL, create DB `hive_metastore`.
    - Add JDBC driver to `$HIVE_HOME/lib`.

   Update `hive-site.xml`
   Example for PostgreSQL:

```
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:postgresql://localhost:5432/hive_metastore</value>
  </property>
  <property>

<name>javax.jdo.option.ConnectionDriverName</name>
    <value>org.postgresql.Driver</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hive</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>hivepassword</value>
  </property>
  <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/user/hive/warehouse</value>
  </property>
</configuration>
```

Initialize Schema

```
schematool -initSchema -dbType postgres
```

Start Hive

```
hive
```

Step 3: Install Spark

Download & Extract

```
wget https://downloads.apache.org/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz
tar -xvzf spark-3.5.1-bin-hadoop3.tgz
mv spark-3.5.1-bin-hadoop3 ~/spark
```

1.

Set Environment Variables

```
export SPARK_HOME=~/spark
export PATH=$PATH:$SPARK_HOME/bin
```

2.
3. **Configure Spark with Hadoop & Hive**
   - Add `hive-site.xml` to `$SPARK_HOME/conf/`.

Make sure Hadoop config dirs are available in Spark's environment:

```
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

   -

Start Spark Shell

```
spark-shell
```

or for PySpark:

```
pyspark
```

4. Spark should now integrate with Hive Metastore and Hadoop FS.

Step 4: Post-Installation Validation

HDFS test

```
hdfs dfs -mkdir /test
hdfs dfs -ls /
```

1.

Hive test

```
CREATE DATABASE testdb;
USE testdb;
CREATE TABLE emp(id INT, name STRING);
SHOW TABLES;
```

2.

Spark test

```
val df =
spark.read.json("examples/src/main/resources/people.
json")
df.show()
```

3.
