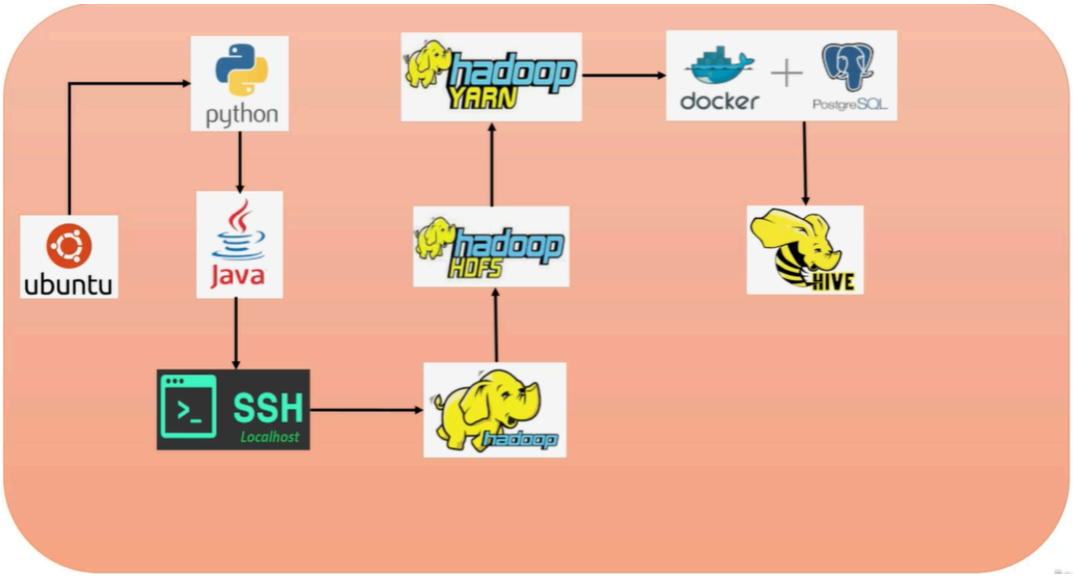
Setup Hive and Spark on Single Node Cluster



https://www.youtube.com/watch?v=xzpF8DcjMbk

Here is a step-by-step guide to set up Apache Hive and Apache Spark on a single node cluster, typically on a Linux machine where you already have Hadoop running:

Prerequisites

- Hadoop single node cluster installed and running
- Java JDK 8+ installed (same as Hadoop)
- SSH configured for localhost

Part 1: Install and Configure Apache Hive

Step 1: Download Hive

Download Apache Hive from the official mirror, e.g.,

```
Wget
https://downloads.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin
.tar.gz
```

Extract Hive:

```
tar -xzf apache-hive-3.1.3-bin.tar.gz
sudo mv apache-hive-3.1.3-bin /usr/local/hive
```

Step 2: Setup Environment Variables

Add the following to your `.bashrc` or `.profile`:

```
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
```

Reload:

```
source ~/.bashrc
```

Step 3: Configure Hive

Hive uses a metastore (by default a Derby embedded database). For production, you would set up MySQL or Postgres, but for single node use Derby:
Create directory for warehouse:

```
mkdir -p /user/hive/warehouse
```

Edit `$HIVE_HOME/conf/hive-site.xml` (copy template first if not present):

```
cp $HIVE_HOME/conf/hive-default.xml.template
$HIVE_HOME/conf/hive-site.xml
```

Add the following configuration block inside `<configuration>` tags:

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

<value>jdbc:derby:;databaseName=metastore_db;create=true<
/value>
  <description>JDBC connect string for a JDBC
metastore</description>
</property>

<property>
  <name>hive.metastore.warehouse.dir</name>
  <value>/user/hive/warehouse</value>
</property>
```

Step 4: Initialize the Hive Metastore

Run:

```
schematool -initSchema -dbType derby
```

Step 5: Test Hive

Start Hive CLI:

```
hive
```

Run sample commands:

```
CREATE TABLE test (id INT, name STRING);
SHOW TABLES;
```

Part 2: Install and Configure Apache Spark

Step 1: Download Spark

Download a Spark pre-built distribution compatible with your Hadoop version:

```
wget
https://downloads.apache.org/spark/spark-3.3.2/spark-3.3.
2-bin-hadoop3.tgz
```

Extract and move it:

```
tar -xzf spark-3.3.2-bin-hadoop3.tgz
sudo mv spark-3.3.2-bin-hadoop3 /usr/local/spark
```

Step 2: Setup Environment Variables

Add to `.bashrc`:

```
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
```

Reload environment:

```
source ~/.bashrc
```

Step 3: Configure Spark to Use Hadoop (optional)

In `$SPARK_HOME/conf/spark-env.sh` (copy template first):

```
cp $SPARK_HOME/conf/spark-env.sh.template
$SPARK_HOME/conf/spark-env.sh
```

Add:

```
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
```

Step 4: Start Spark Standalone Cluster

Start the Spark master:

```
start-master.sh
```

You should see a URL like `spark://<hostname>:7077`
Start a worker connected to the master (replace `<master-url>` with above URL):

```
start-worker.sh spark://localhost:7077
```

Check web UI at http://localhost:8080 for master and http://localhost:8081 for worker.

Step 5: Run Spark with Hive support

Spark can use Hive as its metastore for SQL queries.

- Copy the Hive configuration to Spark conf:

```
cp $HIVE_HOME/conf/hive-site.xml $SPARK_HOME/conf/
```

- When launching Spark shell:

```
spark-shell --conf
spark.sql.warehouse.dir=/user/hive/warehouse
```

Test Spark SQL with Hive:

```
spark.sql("SHOW TABLES").show()
```

Summary Architecture

Component	**Role**
Hadoop HDFS	Distributed storage (single node here)
YARN	Resource manager for cluster
Hive Metastore	Stores metadata about Hive tables
Hive CLI	Query interface to Hive
Spark Standalone	Compute engine with master and worker nodes

Spark SQL	Runs SQL queries using Hive metastore