To install Hadoop 3.4.0 on Ubuntu, follow these steps:

Step 1: Update the system

Open a terminal and run:

sudo apt-get update

sudo apt-get upgrade

Step 2: Install Java

Hadoop requires Java to be installed. You can install OpenJDK 11 (recommended) with:

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

Verify the installation:

java -version

You should see output indicating the version of Java installed. For Java 11, it should be something like:

openjdk version "11.0.x" 202x-xx-xx

OpenJDK Runtime Environment (build 11.0.x+xx-Ubuntu-xx.x)

OpenJDK 64-Bit Server VM (build 11.0.x+xx-Ubuntu-xx.x, mixed mode)

Step 3: Download Hadoop 3.4.0

Visit the Hadoop releases page and download Hadoop 3.4.0. Alternatively, use wget to download directly:

wget https://downloads.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz

Step 4: Extract Hadoop

Extract the downloaded file:

tar -xzvf hadoop-3.4.0.tar.gz

Move the extracted folder to /usr/local/:

sudo mv hadoop-3.4.0 /usr/local/Hadoop

Step 5: Configure Environment Variables

Edit the .bashrc file to include Hadoop environment variables:

nano ~/.bashrc

Add the following lines at the end of the file:

Hadoop Environment Variables

export HADOOP_HOME=/usr/local/hadoop

export HADOOP_INSTALL=$HADOOP_HOME

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_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 PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin

Apply the changes:

source ~/.bashrc

Locate Java Installation:

Find out where Java is installed:

which java

This will show the path to the java executable, typically something like /usr/bin/java.

Find Java Home Directory:

To find the Java home directory, use:

readlink -f $(which java) | sed "s:bin/java::"

This command will print the directory path where Java is installed, which should match /usr/lib/jvm/java-11-openjdk-amd64 or similar.

Verify JAVA_HOME

Make sure the JAVA_HOME environment variable is pointing to the correct Java installation directory.

Check JAVA_HOME:

If you haven't already, you can print the value of JAVA_HOME to ensure it's set correctly:

echo $JAVA_HOME

It should match the path found in the previous step.

Update JAVA_HOME if Necessary:

If JAVA_HOME is not set or is incorrect, update it in the .bashrc file as described earlier:

nano ~/.bashrc

Add or modify the JAVA_HOME line:

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

export PATH=$PATH:$JAVA_HOME/bin

Save and exit, then reload the .bashrc:

echo $JAVA_HOME

java -version

Verify Java Setup

Confirm that Java is correctly set up by running:

echo $JAVA_HOME

java -version

Ensure JAVA_HOME in Hadoop Configuration

Edit hadoop-env.sh:

Open the hadoop-env.sh file again:

nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh

Check or Add JAVA_HOME:

Ensure that the JAVA_HOME environment variable is correctly set in this file. Add or modify the line:

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

Save and exit the editor.

Reload Environment:

Reload the Hadoop environment settings:

source $HADOOP_HOME/etc/hadoop/hadoop-env.sh

Step 6: Configure Hadoop

You need to configure core-site.xml, hdfs-site.xml, and mapred-site.xml located in the $HADOOP_HOME/etc/hadoop/ directory.

Configure core-site.xml:

Edit the file:

nano $HADOOP_HOME/etc/hadoop/core-site.xml

Add the following configuration:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

Configure hdfs-site.xml:

Edit the file:

nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
```

```
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/usr/local/hadoop/tmp/dfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/usr/local/hadoop/tmp/dfs/data</value>
  </property>
</configuration>
```

Configure mapred-site.xml:

This file might not exist by default, so create it by copying the template:

```
cp $HADOOP_HOME/etc/hadoop/mapred-site.xml.template
$HADOOP_HOME/etc/hadoop/mapred-site.xml
```

Edit the file:

```
nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

Add the following configuration:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
  </property>
  <property>
```

```
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
  </property>
</configuration>
```

Configure yarn-site.xml

Open yarn-site.xml for Editing

```
nano /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

```
<configuration>
  <!-- ResourceManager settings -->
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>localhost</value>
  </property>
  <property>
    <name>yarn.resourcemanager.rpc-address</name>
    <value>localhost:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>localhost:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>localhost:8025</value>
  </property>
```

```
  <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>localhost:8141</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>/tmp/yarn/local</value>
  </property>
  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/tmp/yarn/log</value>
  </property>
  <!-- Additional NodeManager settings -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

Create and Set Permissions for Directories:

```
mkdir -p /usr/local/hadoop/tmp/dfs/name
mkdir -p /usr/local/hadoop/tmp/dfs/data
mkdir -p /tmp/yarn/local
mkdir -p /tmp/yarn/log
chown -R sai:sai /usr/local/hadoop/tmp
chown -R sai:sai /tmp/yarn
chmod -R 755 /usr/local/hadoop/tmp
```

chmod -R 755 /tmp/yarn

3. Format HDFS Filesystem

hdfs namenode -format

4. Start Hadoop Daemons

start-dfs.sh

start-yarn.sh

5. Verify Hadoop Setup

Check Hadoop Processes:

jps

Ensure that NameNode, DataNode, ResourceManager, and NodeManager are running

Access Web Interfaces:

HDFS Namenode: http://localhost:9870

YARN ResourceManager: http://localhost:8088

6. Run a Test Job

hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar pi 2 5

7. Stop Hadoop Daemons

stop-yarn.sh

stop-dfs.sh