Complete, step-by-step Hadoop setup guide for Apple M-series Macs.
This covers **Java installation, Hadoop installation, SSH config, and environment variables** so you can get a working single-node Hadoop cluster.

Step 0: Prerequisites

- macOS (Apple Silicon M1/M2/M4)
- Homebrew installed
- Terminal knowledge
- Internet connection

Step 1: Install Java (Temurin 17)

1. Update Homebrew and tap versions:

```
brew update
brew tap homebrew/cask-versions
```

2. Install Java 17:

```
brew install --cask temurin@17
```

3. Verify installation:

```
ls /Library/Java/JavaVirtualMachines/
```

You should see:

```
temurin-17.jdk
```

4. Set JAVA_HOME in your shell (~/.zshrc):

```
echo 'export JAVA_HOME=/Library/Java/JavaVirtualMachines/temurin-17.jdk/Contents/Home'
>> ~/.zshrc
echo 'export PATH=$JAVA_HOME/bin:$PATH' >> ~/.zshrc
source ~/.zshrc
```

5. Verify Java:

```
java -version
echo $JAVA_HOME
```

Step 2: Install Hadoop (3.4.2 recommended)

1. Install Hadoop via Homebrew:

```
brew install hadoop
```

2. Add Hadoop environment variables to ~/.zshrc:

```
echo 'export HADOOP_HOME=/opt/homebrew/Cellar/hadoop/3.4.2/libexec' >> ~/.zshrc
echo 'export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH' >> ~/.zshrc
echo 'export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop' >> ~/.zshrc
source ~/.zshrc
```

3. Verify Hadoop:

```
hadoop version
```

Step 3: Configure SSH for Hadoop

Hadoop needs passwordless SSH to localhost for start-dfs.sh and start-yarn.sh.

1. Generate SSH key (if not exists):

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
```

2. Add public key to authorized keys:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
chmod 700 ~/.ssh
```

3. Test SSH:

```
ssh localhost
```

- First login may show: Are you sure you want to continue connecting (yes/no)? → type yes.
- Ignore compdef warnings; they are zsh autocomplete issues.

Step 4: Configure Hadoop

1. Edit **Hadoop environment**:

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

Set:

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/temurin-17.jdk/Contents/Home
```

2. Configure core files for single-node:

- **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:///usr/local/hadoop_data/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///usr/local/hadoop_data/hdfs/datanode</value>
  </property>
</configuration>
```

- **mapred-site.xml** (create by copying template):

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

Step 5: Format HDFS

Create Hadoop directories:

```
sudo mkdir -p /usr/local/hadoop_data/hdfs/namenode
sudo mkdir -p /usr/local/hadoop_data/hdfs/datanode
sudo chown -R $(whoami) /usr/local/hadoop_data
```

Format HDFS:

hdfs namenode -format

Step 6: Start Hadoop Services

1. Start HDFS:

start-dfs.sh

2. Start YARN:

start-yarn.sh

3. Check running Hadoop processes:

jps

You should see:

NameNode
DataNode
SecondaryNameNode
ResourceManager
NodeManager

Step 7: Test HDFS

hdfs dfs -mkdir /user
hdfs dfs -mkdir /user/$(whoami)
hdfs dfs -ls /

You should see the directories you just created.

Step 8: Notes on Warnings

- WARN util.NativeCodeLoader: normal on ARM64; Hadoop will use Java classes.

- logs does not exist: Hadoop creates it automatically.
