

This guide contains instructions for running a (pre-installed) **standalone version of Hadoop on corn.stanford.edu**.

If you'd prefer to install Hadoop on your laptop, further down in the guide you will find instructions to download and install a **standalone non-distributed version of Hadoop** on your Mac or Windows. Such standalone versions run as a single Java process and are helpful for debugging a new Mapreduce. This will suffice for our purposes. It also contains instructions for testing out your installation.

Instructions for running Hadoop on corn.stanford.edu

- `ssh <sunset-id>@corn.stanford.edu`
- Download the skeleton code and data at <http://cs276.stanford.edu/examples.tar.gz>.
 - `mkdir ~/yarn`
 - `cd ~/yarn`
 - `wget http://cs276.stanford.edu/examples.tar.gz`
 - `tar -zxvf examples.tar.gz`

You should have the skeleton code directory **src/**, data directory (**doc/**) and the build file **build.xml**.

- Compile your code into a jar file by executing:
 - `bash`
 - `export JAVA_HOME=/usr/lib/jvm/default-java`
 - `export HADOOP=/afs/ir/class/cs276/hadoop/hadoop-2.7.2`
 - `ant`

You should have the file **HadoopExamples.jar** in your current directory.
(Hadoop was installed in `/afs/ir/class/cs276/hadoop/hadoop-2.7.2`)

- Run your code, suppose your program is `src/edu/stanford/cs276/WordCount.java`:
 - `$HADOOP/bin/hadoop jar HadoopExamples.jar edu.stanford.cs276.WordCount <input_dir> <output_dir>`where `<input_dir>` contains files you want to count words and `<output_dir>` is the directory where your results are stored.

For example, after running the below command, your results are in `output/part-00000`:

- i. `$HADOOP/bin/hadoop jar HadoopExamples.jar edu.stanford.cs276.WordCount docs output`

- Making modification to the code:
 - Edit the file `src/edu/stanford/cs276/WordCount.java`
 - Execute the below commands and you should now see 2 files `part-00000` `part-00001` in the `output/` directory:
 - i. `ant`
 - ii. `rm -rf output`
 - iii. `$HADOOP/bin/hadoop jar HadoopExamples.jar edu.stanford.cs276.WordCount docs output`
- All other codes (`edu.stanford.cs276.CreateLM`, `edu.stanford.cs276.CreateInvertedIndex`, `edu.stanford.cs276.ComputeDF`) expect similar sets of arguments: `<input_dir> <output_dir>`.

Installing Hadoop on Mac

These instructions assume that you already have a recent version of Java installed on your Mac (I have version 1.8 installed). Hadoop version 2.8.0 requires at least Java 1.7. Hadoop version 2.6.4 can also run with Java 1.6. Hadoop-specific installation instructions largely follow the instructions on:

<http://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-common/SingleCluster.html>

(Note: At the present moment, the “current” version of Hadoop is 3.0.0-alpha2. I have chosen not to upgrade to this version, and have stayed with version 2.8.0)

- Download the Hadoop tarball from
`http://hadoop.apache.org/releases.html`
 Get Hadoop version 2.8.0.
- Create a new directory for all your Hadoop work. I created it at `~/yarn`. The rest of the instructions assume that’s where you’ve put your files. Modify the instructions appropriately if you put it in some other location. From a Terminal window, issue the following command:


```
mkdir ~/yarn
```
- Untar the Hadoop tarball into the above directory with the following commands from the **Terminal** window. The following assumes you’ve downloaded the tarball to `~/Downloads/hadoop-2.8.0.tar.gz`

```
cd ~/yarn
tar xvf ~/Downloads/hadoop-2.8.0.tar.gz
```

This should place all the Hadoop files in `~/yarn/hadoop-2.8.0`. If your version of `tar` doesn't automatically unzip the file, use `gunzip` to do that first.

- Set up your environment variables by executing:

```
# Recommended way to set JAVA_HOME on a Mac.
export JAVA_HOME=$(/usr/libexec/java_home)

# Modify this if you installed Hadoop somewhere else
export HADOOP=${HOME}/yarn/hadoop-2.7.2
```

If you'd prefer not having to set these variables each time, you can place the above export commands in your `~/.bashrc`.

Source your `~/.bashrc` in your **Terminal** window to make sure these variables are set. You won't have to do this in the future since the **Terminal** window will automatically source `~/.bashrc` for each new window.

```
source ~/.bashrc
```

That concludes the installation of the standalone, non-distributed version of Hadoop. This will suffice for our purposes. If you are feeling adventurous, you can also try to set up Hadoop for pseudo-distributed operation using the instructions in the `hadoop.apache.org` URL referenced above.. In that setup you will run several daemons exactly as if you were running on a large Hadoop cluster, except that you'd have only one node in your cluster (i.e., just your computer). But, again, this is not needed for our purposes.

Installing Hadoop on Windows

The official `hadoop-2.6.0` does not have Windows binaries and requires building from source (which is complicated). Luckily, somebody has already done that.

- Get **hadoop-2.6.0.tar.gz** from <http://www.barik.net/archive/2015/01/19/172716/> and extract to a folder (e.g., `C:\yarn\hadoop-2.6.0`). These prebuilt binaries have been verified to work on Windows 8.1 x64 but should work on other recent versions as well.
- Download and install the JDK for Java 7 or 8 (**NOT JRE**).
 - If you already have Eclipse running, it is likely that you already have a JDK
 - Unlike what it says in the hadoop download page, Java 8 should work.

- Set your environment variables (through Control Panel → System → Advanced System Settings → Advanced → Environment Variables..., or press Win+Break → Advanced System Settings → Environment Variables..., or by using <http://www.rapidee.com/>)
 - **JAVA_HOME** needs to where your JDK lives (e.g., `C:\Java\jdk1.7.0_79`). If your path contains space, you need to use the [8.3 filename notation](#) like `C:\Progra~1\Java\jdk1.7.0_79`
 - **HADOOP** needs to point to where you installed Hadoop (e.g., `C:\yarn\hadoop-2.6.0`)

This will give you a basic working version of Hadoop. You can find more detailed information at:

<http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SingleCluster.html>

Testing your Hadoop installation on Mac (or Windows)

You can download sample data and test your Hadoop installation as follows.

- Download the file `examples.tar.gz` from <http://cs276.stanford.edu/examples.tar.gz>.
- Untar this file into a convenient place. I put these files in `~/yarn`

```
cd ~/yarn
tar xvf ~/Downloads/examples.tar.gz
```

(Windows: Just use a GUI such as [7-zip](#). Extract the files to `C:\yarn`)

This should create

- One directory, `docs/`, with sample data
 - A jar file, `HadoopExamples.jar`, that you can use to test your installation
 - Java source files in `src/edu/stanford/cs276` that you will modify during the lecture.
- Run the `WordCount` example. The following assume that `$HADOOP` (set above) is a directory in `~/yarn` (where the sample data and jar files have been placed).

```
cd $HADOOP
./bin/hadoop jar ../HadoopExamples.jar \
    edu.stanford.cs276.WordCount \
```

```
../docs ../output
```

Windows: (Warning: use \ instead of /)

```
cd %HADOOP%
.\bin\hadoop jar ../HadoopExamples.jar ^
edu.stanford.cs276.WordCount ^
../docs ../output
```

The Mapreduce should run successfully and end with a listing of various counters. You can check out the output with:

```
less ../output/part-r-00000
```

Windows:

```
more ../output\part-r-00000
```

To rerun the Mapreduce, make sure to remove the output directory:

```
rm -rf ../output
./bin/hadoop jar ../HadoopExamples.jar \
edu.stanford.cs276.WordCount \
../docs ../output
```

Windows:

```
rmdir ../output /S
.\bin\hadoop jar ../HadoopExamples.jar ^
edu.stanford.cs276.WordCount ^
../docs ../output
```

Building HadoopExamples.jar from Java sources

Here are different ways you can build `HadoopExamples.jar` from the Java source files in `~/yarn/src/`.

- Building in **Eclipse**.
 - Create a new project in Eclipse called `HadoopExamples` using:
File -> New -> Java Project

and filling in **HadoopExamples** in the **Project name** and clicking on

Finish.

- Create the **edu.stanford.cs276** package in **HadoopExamples** by right clicking (control-click on a Mac) on **HadoopExamples** and following:

New -> Package

entering **edu.stanford.cs276** for the **Name** of the package and clicking on **Finish**.

- import the Java source files by right clicking on the (newly created) **edu.stanford.cs276** package and following:

Import -> General -> File System

and browse to the **~/yarn/src/edu/stanford/cs276** directory, click on **Select All** to import all the Java sources in that directory, and click on **Finish**.

- Set up the include files. You'll need to include the following three jar files:

```
hadoop-common-2.8.0.jar in  
    ${HADOOP}/share/hadoop/common/  
hadoop-mapreduce-client-core-2.8.0.jar in  
    ${HADOOP}/share/hadoop/mapreduce/  
commons-logging-1.1.3.jar in  
    ${HADOOP}/share/hadoop/common/lib/
```

You can include these files by right clicking on the **HadoopExamples** project and following:

Build Path -> Add External Archives ...

browsing to each of the above two folders and selecting the corresponding files.

- Build and export the **HadoopExamples.jar** file by right clicking on the **HadoopExamples** project and following:

Export -> Java -> JAR file

click on **Next >** and select **HadoopExamples** in resources to export, **Browse** to **~/yarn/HadoopExamples.jar** and click on **Finish**.

- Building with **ant**

If you have **ant** installed, you can use the **build.xml** file in **~/yarn** as follows:

```
ant
```

This should build a new version of **HadoopExamples.jar** from the source files.