

CS4225/CS5425 Guide for Code Assignment

AY 2022/23 Semester 2

Prepared by:

Chen Jiqing, Chen Xihao

1 Overview	1
2 Using the SoC Compute Cluster	2
2.1 Create SOC Account	2
2.2 Login to SoC Cluster	2
2.2.1 SoC VPN (recommend)	2
2.2.2 Jump server	4
2.3 Modify and Transfer Files on Cluster	5
2.4 Configure Hadoop and Spark	5
2.5 Test Hadoop and Spark	6
2.5.1 Test Spark	6
2.5.2 Test Hadoop and Submission	7
3 VSCode Set-Up (Optional)	11
3.1 Install VSCode and SSH Client	11
3.2 Set Up Remote Development	12
4 Local Environment Setup (Optional)	14
4.1 Windows 10	14
4.2 Linux	20
4.3 MacOS	22

1 Overview

Getting your hands dirty is always an effective way of learning big data systems. It can be a tough and challenging process, but it will also be a fruitful experience. Let's start from here.

This document provides a guide on setting up and using your environment to use Hadoop and Spark. There are two methods: through SoC cluster, or on your local machine. **We recommend the former, as students have run into various issues setting up their own machines.**

- Test and build your programs in the SoC cluster. **We will grade your submission in this environment.**
 - As of AY 2022/23, SoC cluster nodes have been migrated to use Slurm, a job load manager. Users are no longer able to directly log into each machine to run their jobs. Instead, jobs will have to be submitted through Slurm. **Please do not run heavy computation work on the manager node. These nodes are used for submission).** To test your code you can simply submit your task to the task manager (see [Section 2.5.2](#))
 - The clusters have already been set-up with Java, Hadoop and Spark. Corresponding automation scripts have been written to aid the testing of your programmes. These will be released together with the assignments.
 - Refer to [section 2](#) on using the clusters.
- **[Optional]:** You may try to set up a local environment. Note this is not the easiest to do and you WILL face non-trivial issues. We recommend that you learn how to use the cluster first to submit your assignments before attempting a local set up.
 - Refer to section 4 on local setup. Note the guide is not exhaustive.

2 Using the SoC Compute Cluster

In this section we will introduce how to login to the SoC cluster and how to run code on the cluster.

2.1 Create SOC Account

All the students from SoC and students who take SoC modules can register a SoC account. Registration and enabling clusters are done on ‘**mySoC**’. You can register an account here: <https://mysoc.nus.edu.sg/app/newacct>.

Please make sure your Cluster Access is enabled using the link below:

<https://dochub.comp.nus.edu.sg/cf/guides/compute-cluster/enable-disable-access> .

2.2 Login to SoC Cluster

You have two ways to connect to the SoC cluster: through a jump server or through NUS VPN. For Windows and Mac users, using VPN is recommended. For Linux users, please connect via jump server.

2.2.1 SoC VPN (recommend)

For Windows and Mac users, please download FortiClient VPN from the following link: <https://webvpn.comp.nus.edu.sg/sslvpn/portal.html#/> . Then install and run FortiClient VPN on your laptop. After launching FortiClient, you should observe

FortiClient -- The Security Fabric Agent

File Help

FortiClient VPN

Upgrade to the full version to access additional features and receive technical support.

New VPN Connection

VPN: **SSL-VPN** | IPsec VPN

Connection Name: SOC

Description:

Remote Gateway: webvpn.comp.nus.edu.sg

+Add Remote Gateway

☐ Customize port: 443

Client Certificate: None

Authentication: ☒ Prompt on login ☐ Save login

☐ Do not Warn Invalid Server Certificate

Cancel Save

Fill in the information as above, then click "Save", you should see

FortiClient -- The Security Fabric Agent

File Help

FortiClient VPN

Upgrade to the full version to access additional features and receive technical support.

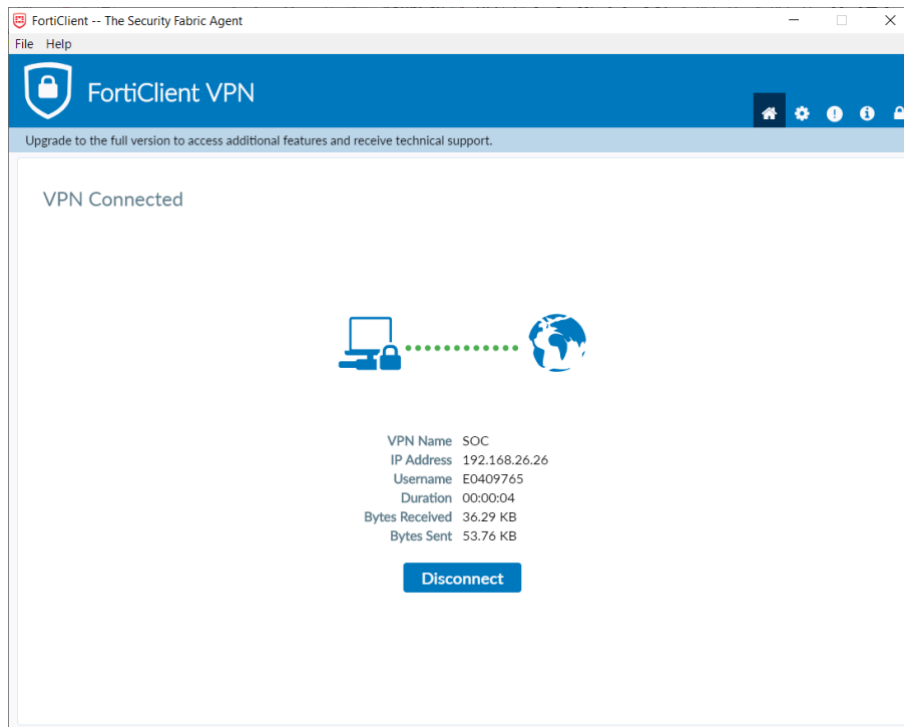
VPN Name: SOC

Username: E0409765

Password:

Cancel Connect

Choose "VPN Name" as "SOC", fill in your NUSNET ID and password, click "connect". After a few seconds, you should see the following information, which indicates successful connection.



After successfully connecting, you are now under SoC network. There are three nodes used for the assignment: `xlog[0-2]`. You can connect them directly via SSH. For example, if your SoC ID is “studentA”, then you can log in to `xlog0` as follows:

```
$ ssh <studentA>@xlog0.comp.nus.edu.sg
```

You can connect to `xlog1` by changing `xlog0.comp.nus.edu.sg` to `xlog1.comp.nus.edu.sg`, etc. Note that the three nodes are indifferent from each other, and you can use any of them as they each have access to your home directory.

2.2.2 Jump server

Access the SoC compute cluster by first ssh-ing into `stu.comp.nus.edu.sg` using your MySoC account and password, e.g.

```
ssh <studentA>@stu.comp.nus.edu.sg.
```

Then under your SoC account, access the Slurm servers by ssh-ing into `xlog[0-2]`, e.g. `ssh xlog0`.

2.3 Modify and Transfer Files on Cluster

You need to modify or upload your assignment files onto the cluster. You can either:

- We recommend you to use remote development on VSCode, please refer [Section 3](#) to set up VSCode, or
- Use `scp` (you can learn `scp` command [here](#)) to copy the files into `xlog[0-2]` and edit your code using `vim` (you can learn `vim` [here](#)) or some other command-line based editor, or
- Use GitHub to “transfer” files between your computer or the cluster, or
- Some other method that you are comfortable with.

Note that all clusters share the same home folders and every machine has access to your folders. You only need to `scp` your files into one of the compute nodes (other `stu.comp.nus.edu.sg`). You do not need to manually copy your files between nodes in the cluster.

2.4 Configure Hadoop and Spark

Hadoop and spark have been already installed on the clusters. All you need to do is configure the work path for Hadoop and Spark. All the following procedures are done on clusters.

You need to create a file named `.bash_profile` under your home directory.

You can make a copy of the file from the course directory using the following command (this will overwrite any existing file!):

```
$ cp /home/course/cs4225/.bash_profile ~/.bash_profile
```

Alternatively, you can paste the following code into your `.bash_profile`. Remember to save the file after editing.

```
BASE_DIR=/home/course/cs4225/cs4225_assign/lib
export JAVA_HOME=$BASE_DIR/jdk-11
export HADOOP_HOME=$BASE_DIR/hadoop-3.3.0
export SPARK_HOME=$BASE_DIR/spark-3.3.1-bin-hadoop3
export PATH=$PATH:$HADOOP_HOME/bin
```

```
export PATH=$PATH:$SPARK_HOME/bin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_CLASSPATH=$JAVA_HOME/lib/tools.jar
export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin:$PATH
```

If you are using VSCode to access the cluster, you can simply create this file from EXPLORER and then copy and paste the code into the file.

If you are using the command line to access the cluster, you may need to use `vim` to edit the file.

After creating `.bash_profile`, run the command:

```
$ source ~/.bash_profile
```

This will set up the environment using `.bash_profile`. When you log in, the system may source “`~/bashrc`” to reset the environment. Hence, you may need to repeat the above command each time you log in. You can add the above line to the `.bashrc` file if needed.

Run the following command to check your environment variables are set up correctly:

```
$ echo $HADOOP_HOME && echo $SPARK_HOME
```

```
(base) xchen@xlog1:~$ source ~/.bash_profile
(base) xchen@xlog1:~$ echo $HADOOP_HOME && echo $SPARK_HOME
/home/course/cs4225/cs4225_assign/lib/hadoop-3.3.0
/home/course/cs4225/cs4225_assign/lib/spark-3.3.1-bin-hadoop3
(base) xchen@xlog1:~$
```

2.5 Test Hadoop and Spark

2.5.1 Test Hadoop

To verify the availability of hadoop, you can check its version:

```
$ hadoop version
```

```
cs4225@stu2:~$ hadoop version
Hadoop 3.3.0
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r aa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled by brahma on 2020-07-06T18:44Z
Compiled with protoc 3.7.1
From source with checksum 5dc29b802d6ccd77b262ef9d04d19c4
This command was run using /user/course/cs4225/cs4225_assign/lib/hadoop-3.3.0/share/hadoop/common/hadoop-common-3.3.0.jar
```

Assignment 0 is designed as a more exhaustive verification for the installation of Hadoop. Please refer to the assignment guide.

2.5.2 Test Spark

To test the availability of spark, simply run an example program of spark. Copy, paste and run the following code in the terminal:

```
$ spark-submit --deploy-mode client --class
org.apache.spark.examples.SparkPi
$SPARK_HOME/examples/jars/spark-examples_2.12-3.3.1.jar
```

This programme estimates the value of Pi. It should output something similar to the following. (Note: lines above and below have been omitted from the screenshot.)

```
r zombie tasks for this job
23/01/27 19:56:16 INFO TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
23/01/27 19:56:16 INFO DAGScheduler: Job 0 finished: reduce at SparkPi.scala:38, took 0.535600 s
Pi is roughly 3.137835689178446
23/01/27 19:56:16 INFO SparkUI: Stopped Spark web UI at http://xlog1.comp.nus.edu.sg:4040
23/01/27 19:56:17 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/01/27 19:56:17 INFO MemoryStore: MemoryStore cleared
23/01/27 19:56:17 INFO BlockManager: BlockManager stopped
```

PS: The program will output a different estimated value each time. You can check at <https://github.com/apache/spark/blob/master/examples/src/main/python/pi.py> for reference.

2.5 Obtaining the Assignment Package

Assignments will be released through the cluster. Here, we run through assignment 0 as an example. All commands are performed on the cluster.

a) Download Assignment Files

Assignment releases will be placed in the directory `/home/course/cs4225/cs4225_assign/release` on SoC clusters. You can download these files by navigating to where you want to copy them (e.g. your home

directory), then using the following command:

```
$ cp -r  
/home/course/cs4225/cs4225_assign/release/CodeAssignment_1 .
```

Then you should find a new folder `CodeAssignment_1` in your home directory. In it, you will find two folders `assign0_hadoop_test` and `assign1_common_words`.

```
$ ls CodeAssignment_1
```

```
xchen@xlog1:~/CS4225_Lab$ cp -r /home/course/cs4225/cs4225_assign/release/CodeAssignment_1  
.  
xchen@xlog1:~/CS4225_Lab$ ls CodeAssignment_1/  
assign0_hadoop_test  assign1_common_words  
xchen@xlog1:~/CS4225_Lab$
```

Alternatively, if you want to first write your codes locally, you can also use `scp` to copy the folder to your own laptop.

The upload can also be done by `scp`. With your (e.g. stuA's) NUS VPN connected, enter the folder which contains the directory you want to upload and run this command **on your own device**:

```
$ scp -r assign0_hadoop_test  
stuA@xlog0.comp.nus.edu.sg:./CodeAssignment_1
```

b) Write your code

For assignment 0, the code in `WordCount.java` is already written for you. There is no need to edit.

c) Compile and Run Your Codes

To run your code, you need to submit them as jobs to the Slurm queue, where they will await their turn to be executed. To get started with submitting jobs on Slurm, here are two useful commands:

- `sbatch <job script>` to submit a job request. The job scripts have been included in the assignment folders, titled `slurm_run.sh`. Simply enter `sbatch slurm_run.sh` to test your job. You should see “*Submitted batch job <job ID>*” printed on the screen.
- `squeue -u <username>` to see all of your job requests. Or, you can simply

use `squeue` to see every job.

When the job disappears from the queue, your job has been completed. You can see your output in the same directory as your job script. For the assignments, the outputs are titled “`ASSIGN_<X>.out`” where X is the assignment number.

To check that your answer is correct, verify that the last line of the output file states “*Test passed.*” by displaying the file using the `cat` command, e.g. `cat ASSIGN_0.out`.

More information on Slurm can be found on [SoC’s Slurm quick start guide](#) and the [official Slurm documentation](#).

Note that the computational resources are NOT exclusive to CS4225/5425 students. Expect heavy loads and long wait times nearing submissions. Hence, please start your assignments early.

2.6 Assignment Submission

After completing the assignment, **you need to submit your code on both Canvas and the cluster. (Files uploaded to Canvas will be used as a backup.)**

To submit to the cluster, you can run the following command:

```
$ ./submit
```

Note that `./submit` is only valid when running on clusters. Do NOT run on your own device. Follow the prompt from script to input your **matriculation number** (starting with A). If your submission was successful, you should see the following:

```
zhaomin@xcnd0:~/cs4225$ ./submit
Please input your student matriculation number: A[REDACTED]0N
You have successfully submitted.
```

You are allowed to submit multiple times before the due date; only your latest submission will be graded. After the due date, the submission folder will be locked and **no more submissions will be accepted**. Please **do NOT modify any filename in your work directory (especially the .java file)**, or the scripts may fail. If the submission script reports any error, that means your submission is unsuccessful. Enquire your TA if needed.

Finally, once you have successfully tested using assignment 0, you can start on assignment 1.

3 VSCode Set-Up (Optional)

This part is optional but it can help to make it more convenient for you to write and test code directly in the remote development environment. In this section, we will briefly demonstrate how to set up VSCode to access and work on the SoC cluster.

3.1 Install VSCode and SSH Client

[Visual Studio Code](#) (VSCode) is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. The Remote Development of VSCode allows you to write and test code directly in the remote development environment in a more convenient way.

To begin with, please go to the official website to download and install VSCode following the SETUP guides:


- a) **Windows:** <https://code.visualstudio.com/docs/setup/windows>
- b) **Linux:** <https://code.visualstudio.com/docs/setup/linux>
- c) **MacOS:** <https://code.visualstudio.com/docs/setup/mac>

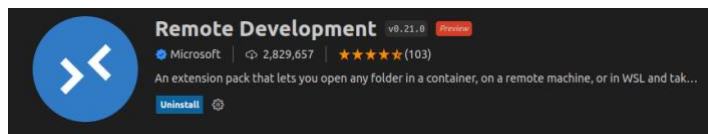
Then install an [OpenSSH compatible SSH client](#) (PuTTY is not supported) as follows:



OS	Instructions
Windows 10 1803+ / Server 2016/2019 1803+	Install the Windows OpenSSH Client .
Earlier Windows	Install Git for Windows .
macOS	Comes pre-installed.
Debian/Ubuntu	Run <code>sudo apt-get install openssh-client</code>
RHEL / Fedora / CentOS	Run <code>sudo yum install openssh-clients</code>

VSCode will look for the ssh command in the `PATH`. Failing that, on Windows it will attempt to find `ssh.exe` in the default Git for Windows install path. You can also specifically tell VS Code where to find the SSH client by adding the `remote.SSH.path` property to `settings.json`.

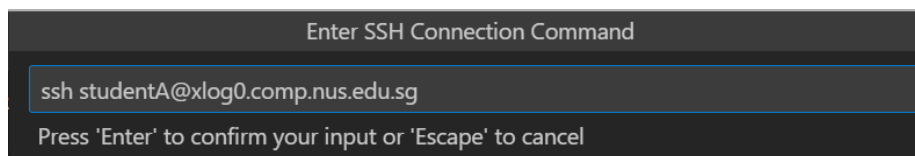
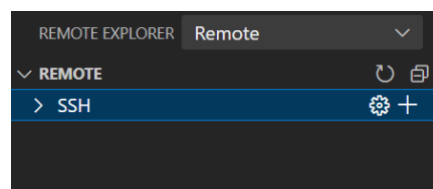
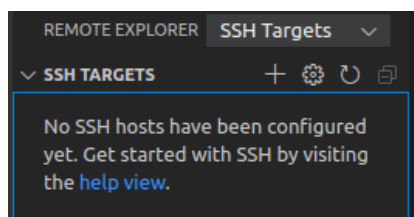
3.2 Set Up Remote Development

After installation, click the **Extensions**  in the column on the left of the VSCode window. Then search and install the extension **Remote Development** as follows.



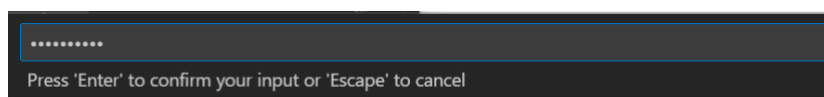
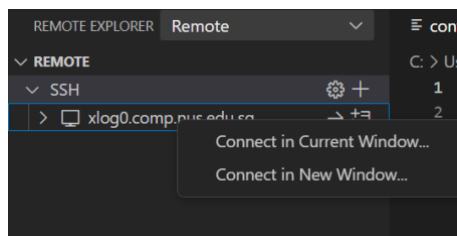
Now there will be an additional item **Remote Explorer**  in the column on the left, just below the **Extensions** .

Click the item and choose **SSH Targets**. If there are no **SSH Targets**, then select **Remote**, **SSH** will be under **Remote**. Then click the plus and type in the ssh command to connect to the SoC cluster xlog[0-2](also remember to connect to SoC VPN in advance).



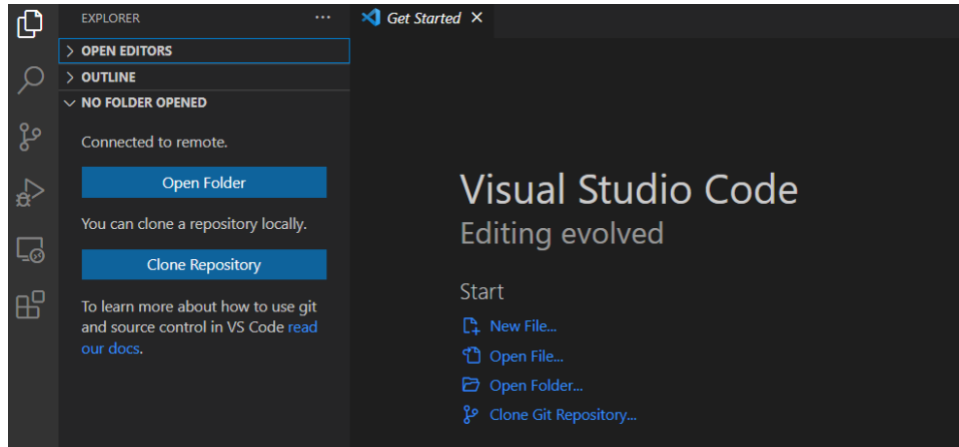
After typing in and selecting the ssh configuration file, the ssh command will be added to the list of **SSH targets**.

To connect, click the **Connect to Host in New Window** item to the right of the hostname, and enter your password.



Afterwards, VSCode will open a new window which prompts you to connect to a remote server.

To edit the code files, you can click the **Explorer** item in the right column and **Open Folder** to the path of the file(s). Please notice that you'll need to enter the password again here but the folder path will also be added to the list of **SSH targets**.



4 Local Environment Setup (Optional)

We strongly advise you to use SoC Cluster for this assignment. Setting local environment may encounter unexpected problems and TAs may not be able to help you.

Therefore, please try to use the SoC Cluster first.

If you are really interested in this area and want to save your time for future projects, Here are some guides to help you set up the debug environments locally. We recommend IntelliJ IDEA 2020.1 as IDE, on which this section is based. For this assignment, this section is optional and only for reference, as you can also choose to debug and test solely on the cluster. You can also choose other IDEs based on your preference.

To do this, follow the guides in subsections 4.1-4.3 based on your operating system.

4.1 Windows 10

a) Install Java 11

Please follow this tutorial to install Java 11 on Windows

<https://java.tutorials24x7.com/blog/how-to-install-java-11-on-windows>. You should also ensure all the environmental variables of Java are set properly (as the tutorial).

b) Install Hadoop

1. Download Hadoop 3.3.0 from <https://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/> Unzip to a directory, e.g. `C:\\Program Files\\hadoop-3.3.0`. You do not need to run



Parent Directory	-
CHANGELOG.md	2020-07-15 17:05 376K
RELEASENOTES.md	2020-07-15 17:05 26K
hadoop-3.3.0-aarch64.tar.gz	2020-07-15 17:19 478M
hadoop-3.3.0-rat.txt	2020-07-15 17:05 2.0M
hadoop-3.3.0-site.tar.gz	2020-07-15 17:33 40M
hadoop-3.3.0-src.tar.gz	2020-07-15 17:05 32M
hadoop-3.3.0.tar.gz	2020-07-15 17:30 478M

the installer.

2. However, this package does not contain some windows native required components. Download these components from :

<https://github.com/kontext-tech/winutils>,

then unzip and copy the whole directory `hadoop-3.3.0/bin` to your installation path of Hadoop, e.g. `C:\\Program Files\\hadoop-3.3.0`.

When conflict happens, choose to replace all conflict files.

3. Also, copy `hadoop-3.3.0/bin/hadoop.dll` to `C:\\Windows\\System32`.

c) Configure environment variables for Hadoop

Open file explorer (by Press `Ctrl+E`). Right-click "`This PC`" and choose "`properties`". In the popup window, click "`Advanced System Settings`", then click "`Environment Variables`". In "`system variables`", create 2 new system variables by "`New...`":

Add `HADOOP_HOME`:

<code>HADOOP_HOME</code>	<code><Your Path of Hadoop></code>
--------------------------	--

Add `HADOOP_BIN_PATH`:

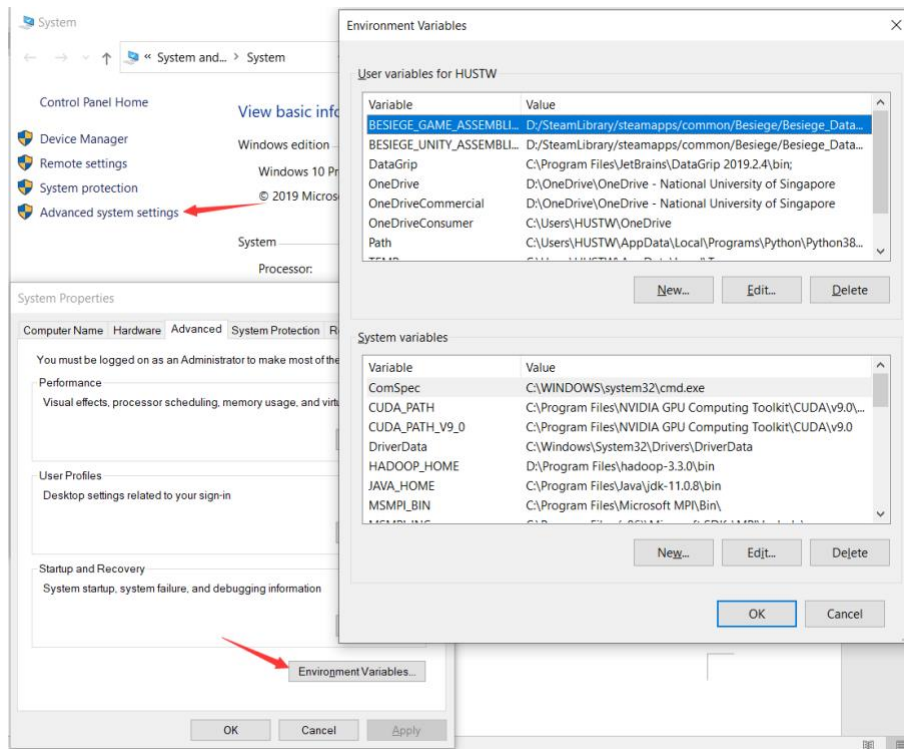
<code>HADOOP_BIN_PATH</code>	<code>%HADOOP_HOME%\bin</code>
------------------------------	--------------------------------

Here is an example of created system variables:

<code>HADOOP_BIN_PATH</code>	<code>%HADOOP_HOME%\bin</code>
<code>HADOOP_HOME</code>	<code>D:\\Program Files\\hadoop-3.3.0</code>

In `User variables`, edit the "`PATH`" variable by adding `%HADOOP_HOME%\bin` and `%HADOOP_HOME%\sbin`.

Click "`OK`", "`OK`", "`OK`" to save the changes. It should take effect immediately.

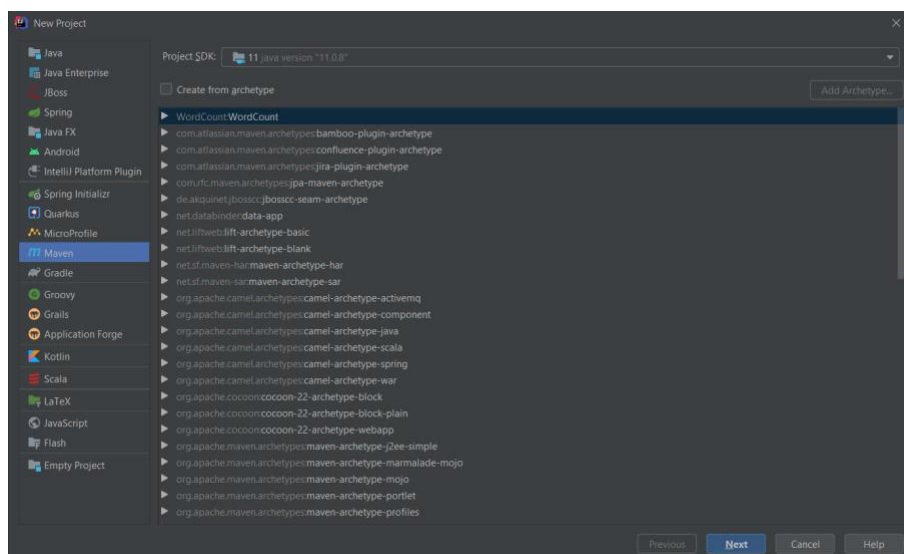


d) Install IntelliJ IDEA

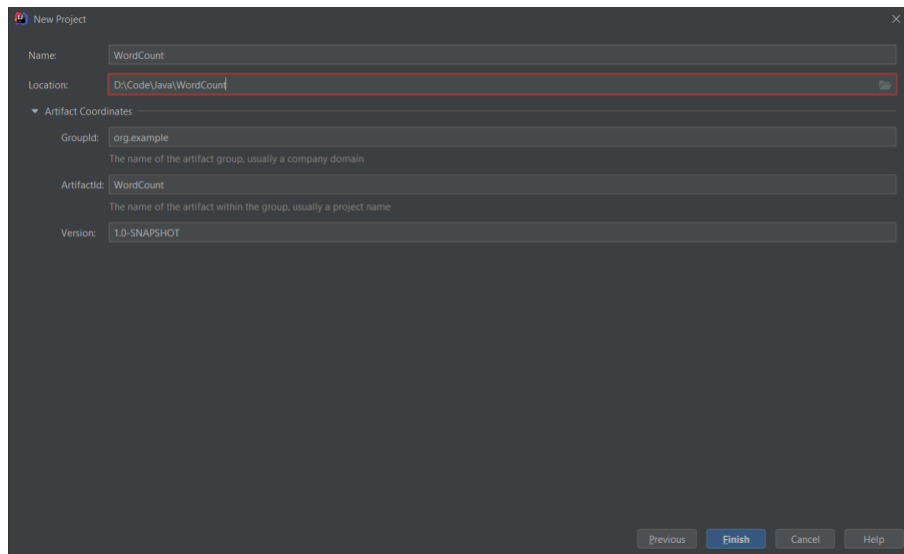
Download the latest IntelliJ IDEA from <https://www.jetbrains.com/idea/download/#section=windows> and install it.

e) Configure IDEA with Hadoop

Create Maven project by "File ? New ? Project ? Maven".

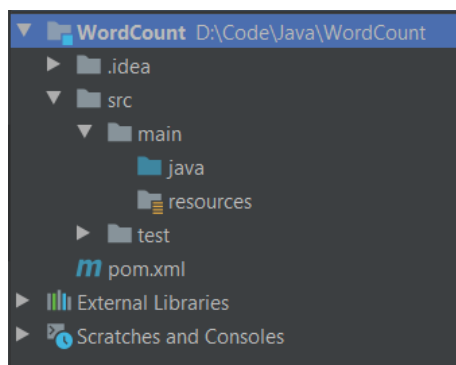


Then, click "Next", enter project `WordCount`. Click "Artifact Coordinates", enter information like the below figure.



The screenshot shows the 'New Project' dialog in IntelliJ IDEA. The 'Name' field is filled with 'WordCount'. The 'Location' field is filled with 'D:\Code\Java\WordCount'. Under the 'Artifact Coordinates' section, the 'GroupId' is 'org.example', the 'ArtifactId' is 'WordCount', and the 'Version' is '1.0-SNAPSHOT'. The 'Finish' button is highlighted in blue.

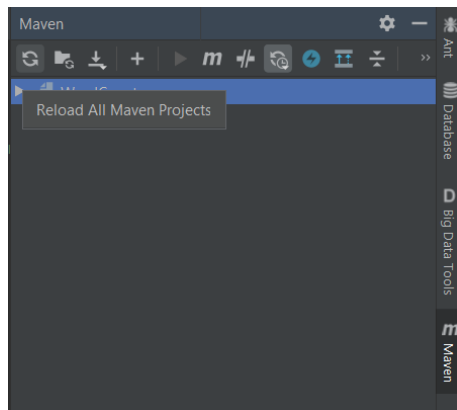
Click **Finish** to create the project. Your project structure should look like this.



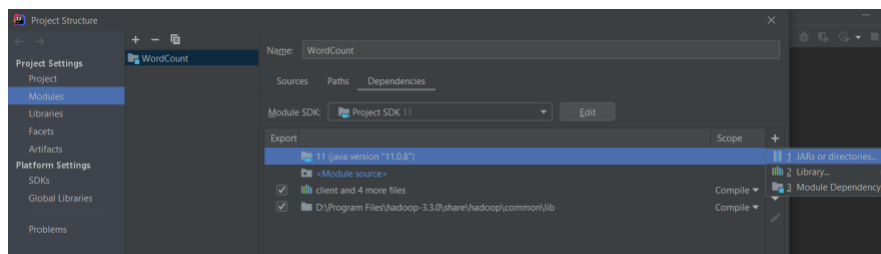
To prevent an error, add the following lines to `pom.xml`.

```
<properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

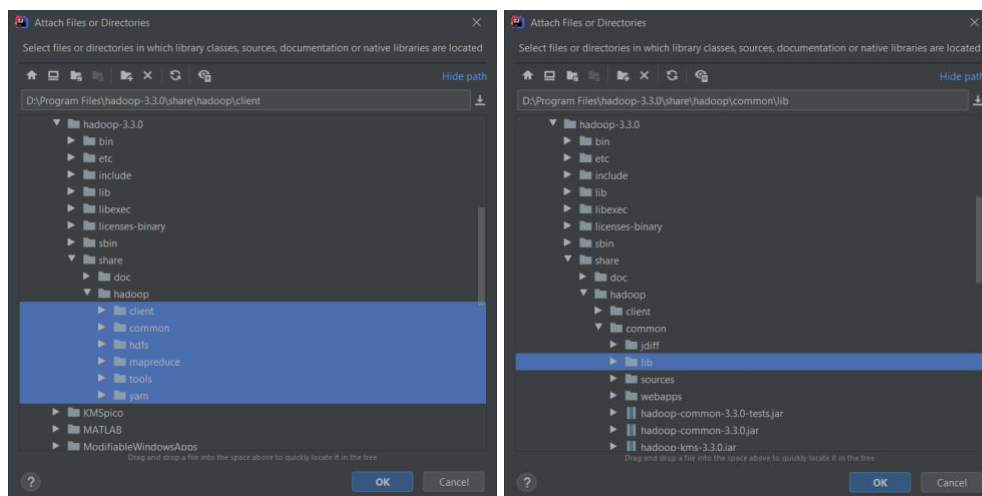
Then reload all maven projects



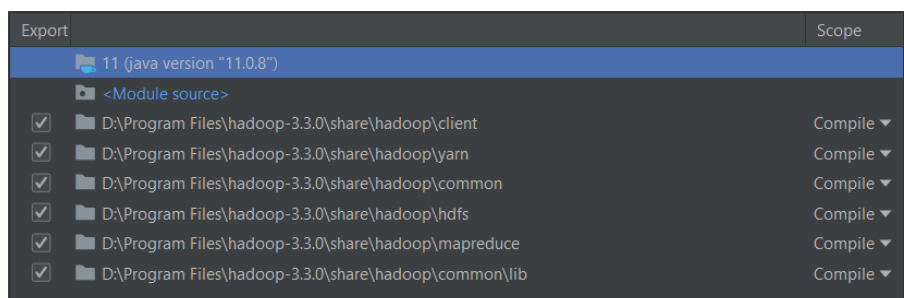
Then add Hadoop dependencies by **File ? Project Structure ? Modules ? Dependencies**. Click **" + ? JARS or directories "**



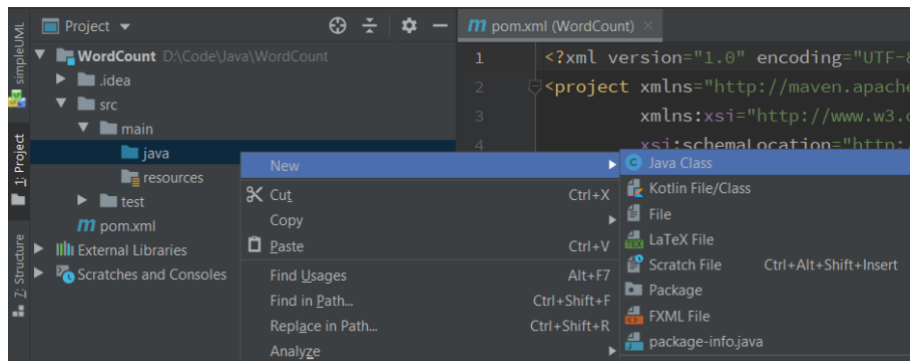
Add the following directories as dependencies.



After that, the dependencies should look like this. Then click **"OK"**.



Create a java class file `WordCount.java` like below



Download `assign0_hadoop_test` from Luminus or from the cluster (see subsection 5.2). Find example codes `WordCount.java` in the package. Copy the content of `WordCount.java` in that file. Then create a directory `input` and two text files in the directory `file0.txt` and `file1.txt`. Their contents are as below.

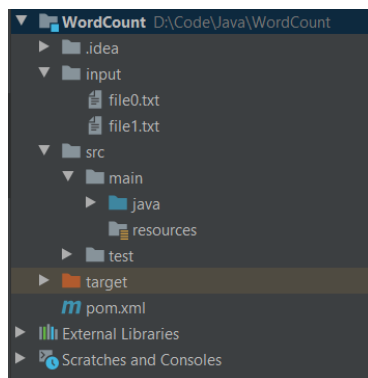
(input/file0.txt)

Hello World Bye World

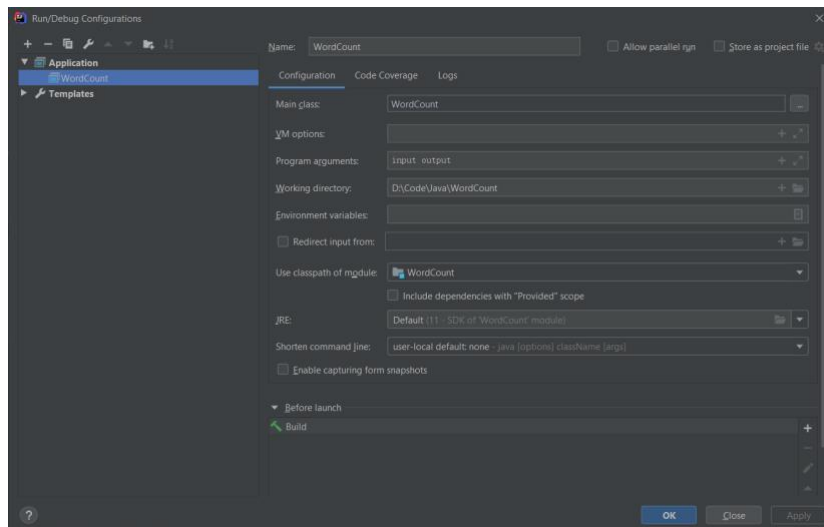
(input/file1.txt)

Hello Hadoop Goodbye Hadoop

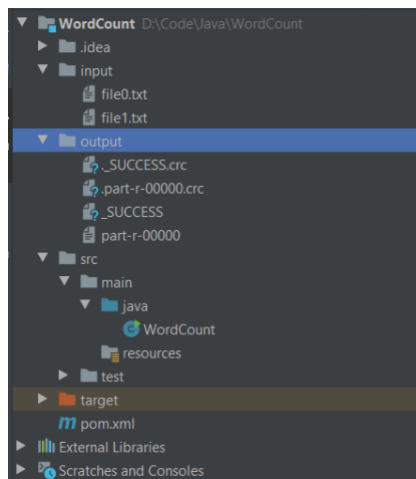
The directory structure should look like this now.



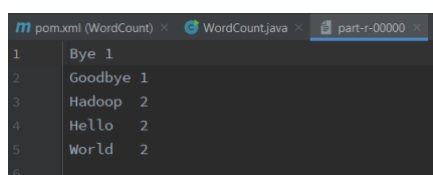
Then click "Add configurations" on right top. In the popup window, click "+ Application". Configure as below.



Then click **OK**. Click the green triangular button "**run**" to run the program. After it finishes, the project structure will contain a new folder **output**.



output/part-r-00000 contains the result of word count.



You can also click to debug button to debug your program. IntelliJ IDEA is a very powerful IDE. Enjoy exploring.

4.2 Linux

a) Install Java 11

There are a lot of tutorials about installing Java 11, you can choose one based on your Linux distribution.

- Ubuntu: <http://ubuntuhandbook.org/index.php/2018/11/how-to-install-oracle-java-11-in-ubuntu-18-04-18-10/>
- CentOS: https://linuxhint.com/install_oracle_jdk11_centos7/
- Arch Linux: <https://wiki.archlinux.org/index.php/Java>
- Fedora: <https://www.tecmint.com/install-java-in-fedora/>

After installation, check your java version by following command

```
$ java -version
openjdk version "11" 2018-09-25
OpenJDK Runtime Environment 18.9 (build 11+28)
OpenJDK 64-Bit Server VM 18.9 (build 11+28, mixed mode)
```

You should ensure the JDK version is 11. The implementation could be either OpenJDK or Oracle JDK. Meanwhile, remember the path where you install java, e.g. `/usr/lib/java`. Add an environmental variable `JAVA_HOME` by

```
$ echo 'export JAVA_HOME=/usr/lib/java' >> ~/.bash_profile
$ source ~/.bash_profile
```

You can check if successful by

```
$ echo $JAVA_HOME
/usr/lib/java
```

Note: Do not simply copy the commands. You need to check your installation path first.

b) Install Hadoop

Download Hadoop 3.3.0

```
$ wget https://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
$ tar xzvf hadoop-3.3.0.tar.gz
```

Configure java path for Hadoop: recall the path where you install java in a), e.g. `/usr/lib/java`. Edit `hadoop-3.3.0/etc/hadoop/hadoop-env.sh`. Find `export JAVA_HOME=`, and change this line to `export`

```
JAVA_HOME=/usr/bin/java.
```

Note: Do not simply copy the commands. You need to check your installation path first.

c) Install IntelliJ IDEA

Download latest IntelliJ IDEA from (Ultimate is free for NUS students, Community is enough for this module)

<https://www.jetbrains.com/idea/download/#section=linux>

Unzip the file

```
$ tar xzvf ideaIC-2020.2.tar.gz
```

run IDEA by

```
$ cd ideaIC-2020.2
```

```
$ bin/idea.sh
```

d) Configure IDEA with Hadoop

This part is exactly the same as that on Windows 10 in [Section 4.1 e\)](#) Configure IDEA with Hadoop. Please refer to that subsection.

4.3 MacOS

a) Install Java 11

Follow the guide of Linux in section 4.2 a).

b) Install Hadoop

You can install via brew. Simply run

```
brew install Hadoop
```

Make sure your installed version is **3.3.0**. You can also install as the Linux guide in [section 4.2 b\)](#).

Configure the Java path for Hadoop (recall the path where you install java, e.g. `/usr/lib/java`).

Edit `<Your-hadoop-path>/etc/hadoop/hadoop-env.sh`. Find `export`

`JAVA_HOME=`, and change this line to

`export JAVA_HOME=<Your-Java-path>`.

Note: Do not simply copy the commands. You need to check your installation path first.

c) Install IntelliJ IDEA

Download the latest IntelliJ IDEA from (Ultimate is free for NUS students, Community is enough for this module)

<https://www.jetbrains.com/idea/download/#section=mac>

d) Configure IDEA with Hadoop

This part is exactly the same as that on Windows 10 in [Section 4.1 e\)](#) Configure IDEA with Hadoop. Please refer to that subsection.