COMP5434 - Tutorial #2

Hadoop

Outlines

In this tutorial, we will discover how to deploy Hadoop, one popular computation infrastructure for big data. We will also learn how to write a basic Hadoop program, which can count the words in a file.

This tutorial is based on Windows 11 (64-bit).

Table of Contents

WSL2

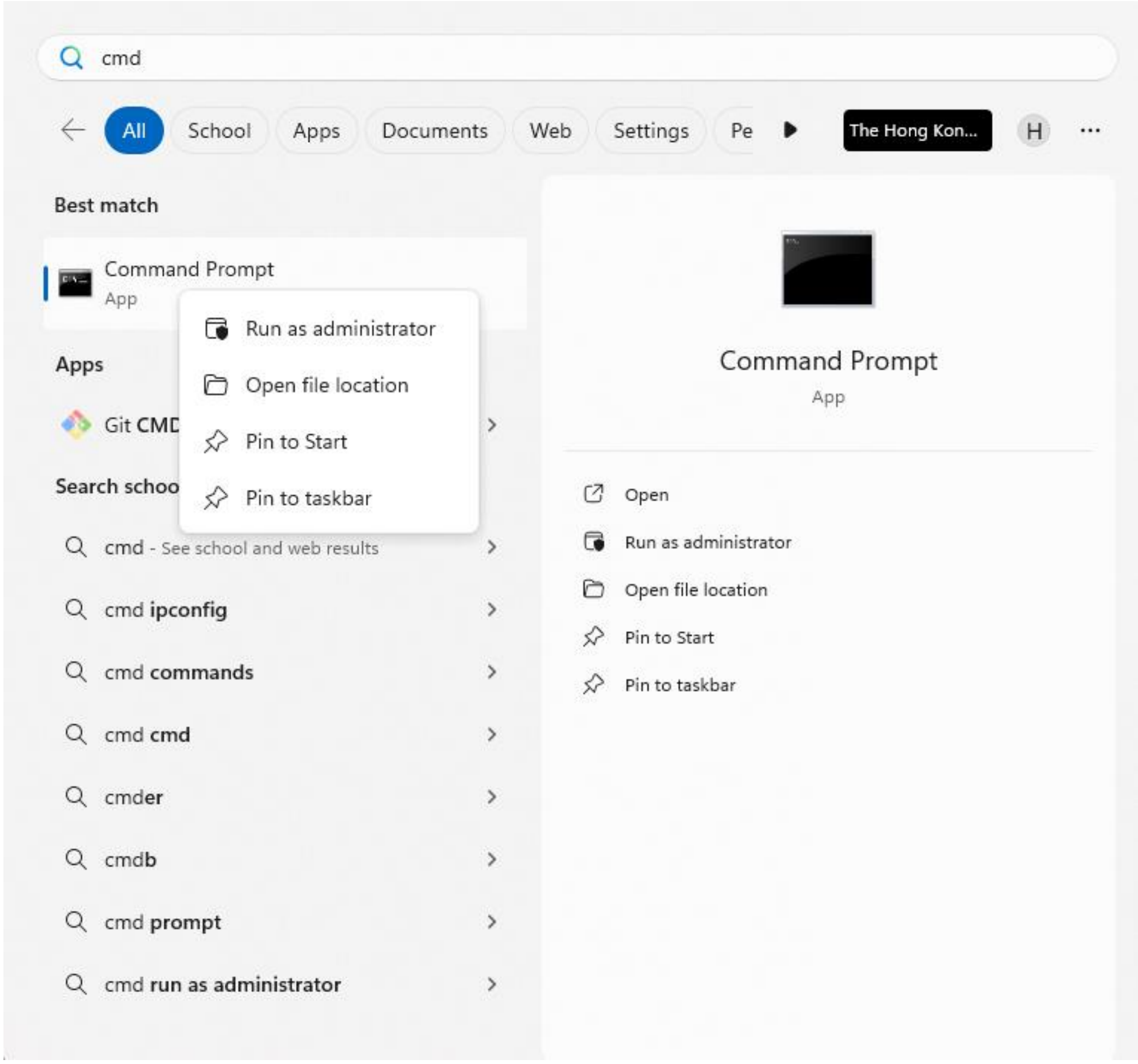
Outlines

Although Hadoop supports Windows and Linux OS, most servers in companies are using Linux actually. In this tutorial, we use *Windows Subsystem for Linux 2* (WSL2) to run a Linux subsystem in Windows, where Hadoop will run.

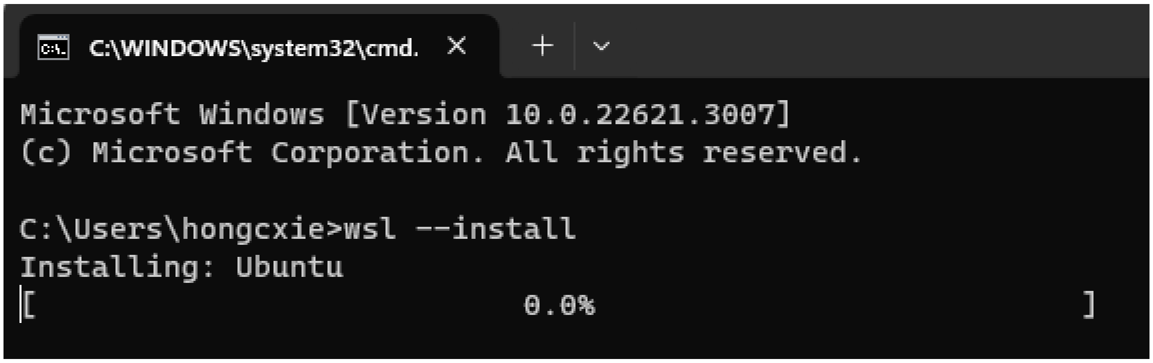
WSL2 Installation

You can skip this section if you are using Linux or macOS, or have already installed WSL2.

1. Click *Start Menu* and type *cmd*. Right-click on the icon and select *Run as Administrator*.

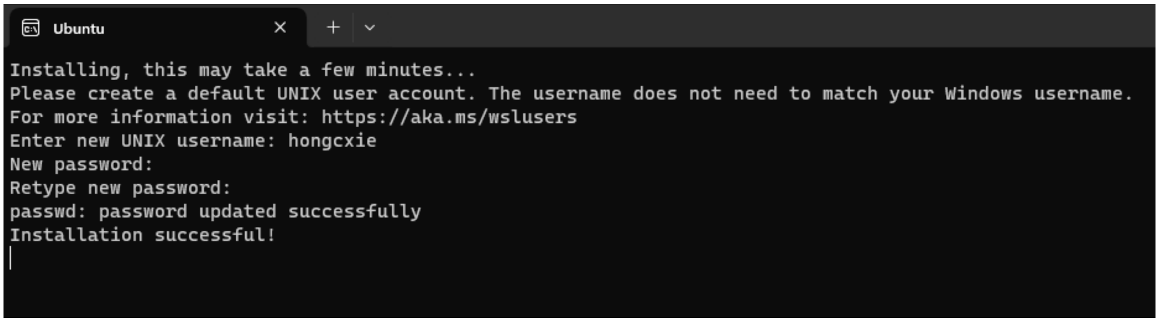


2. Type *wsl --install* to install WSL2.



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hongcxie>wsl --install
Installing: Ubuntu
[                           0.0%                           ]
```

3. Input username and password for Linux.



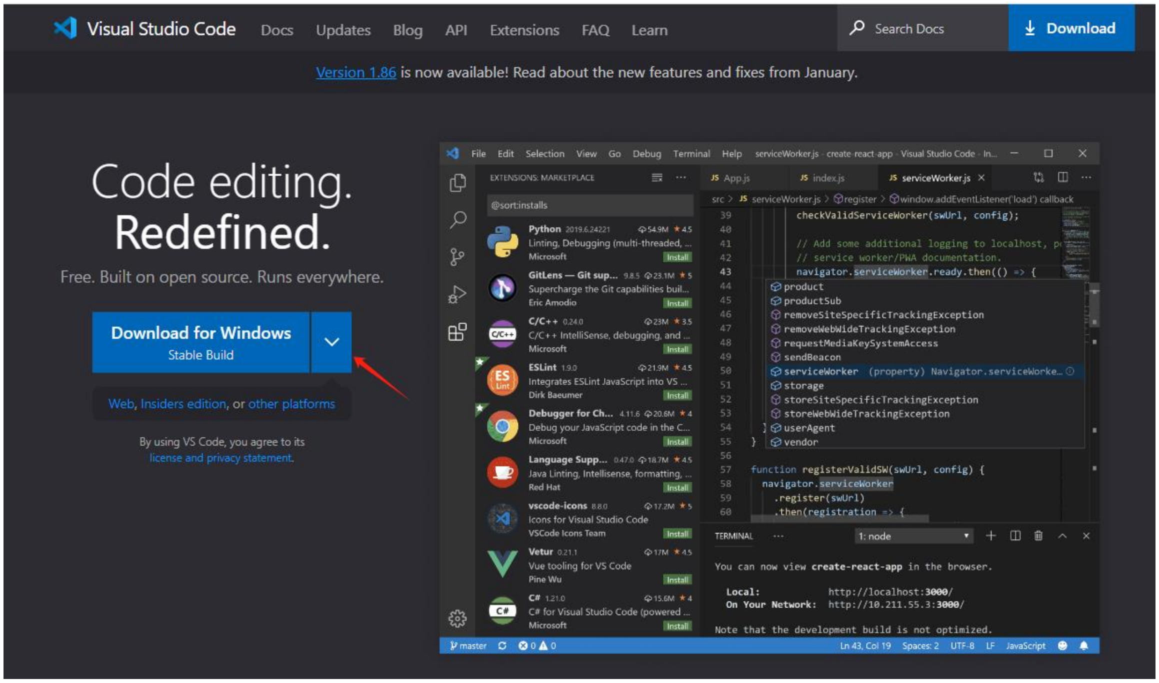
```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: hongcxie
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
```

4. Close the window.

Visual Studio Code (VSCode) Installation

Visual Studio Code (VSCode) is a source-code editor developed by Microsoft. It provides various Extensions which can help us write code efficiently.

1. Open your browser and visit https://code.visualstudio.com/.
2. Click *Download for Windows* to download the installer.



Version 1.86 is now available! Read about the new features and fixes from January.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

By using VS Code, you agree to its license and privacy statement.

```
        checkValidServiceWorker(swUrl, config);

        // Add some additional logging to localhost, p
        // service worker/PWA documentation.
        navigator.serviceWorker.ready.then(() => {

      }
  }

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

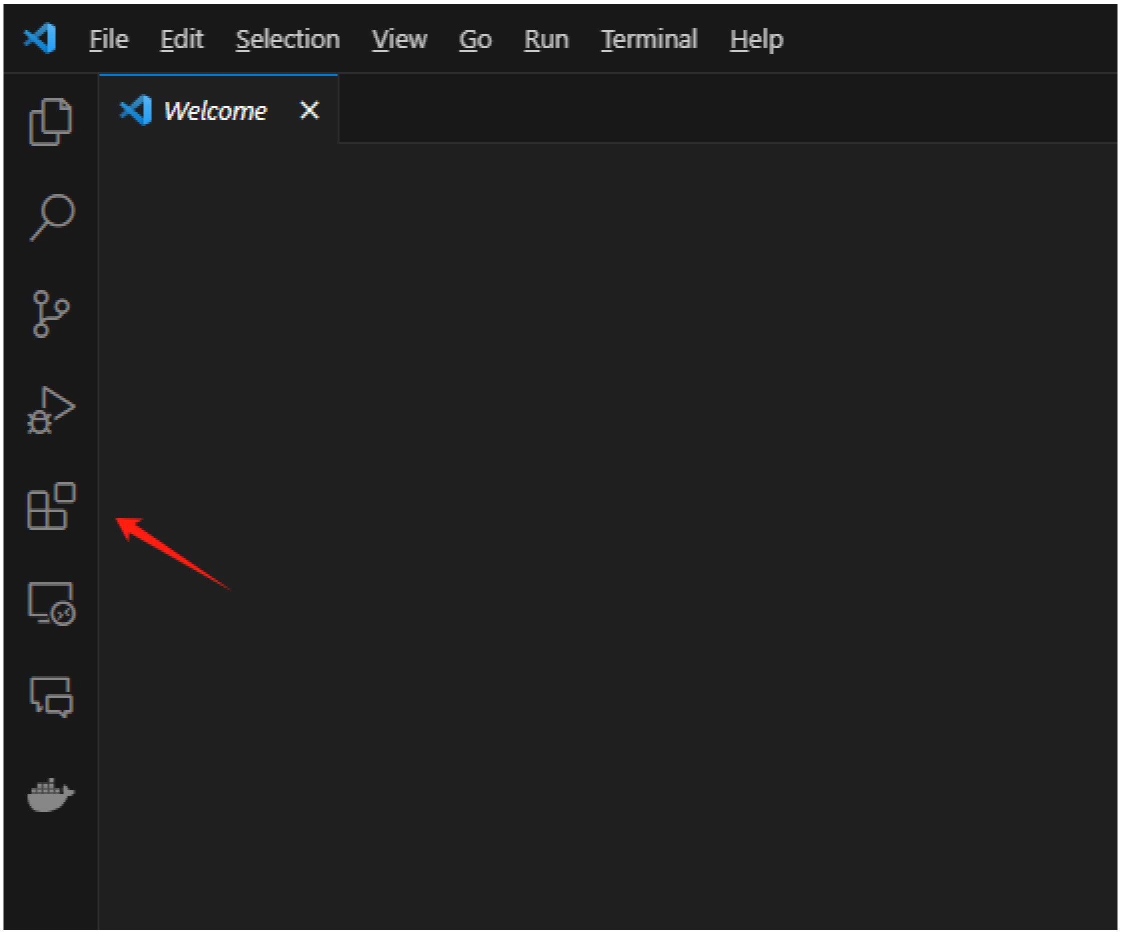
```
You can now view create-react-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://10.211.55.3:3000/

Note that the development build is not optimized.
```

3. Download the installer and install it.

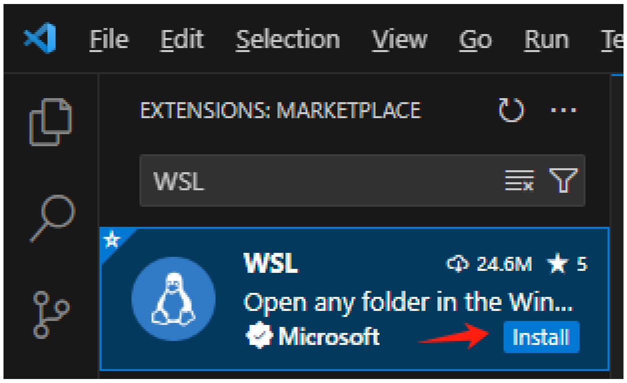
WSL Extension

WSL Extension is one extension in VSCode, which enables us to write code in WSL2 from Windows (host) directly. It can also open shells of WSL2 so that we can run Linux commands in WSL2.

1. Open VSCode and click *Extensions* button.



2. Type *WSL* in the textbox and download *WSL* extension.

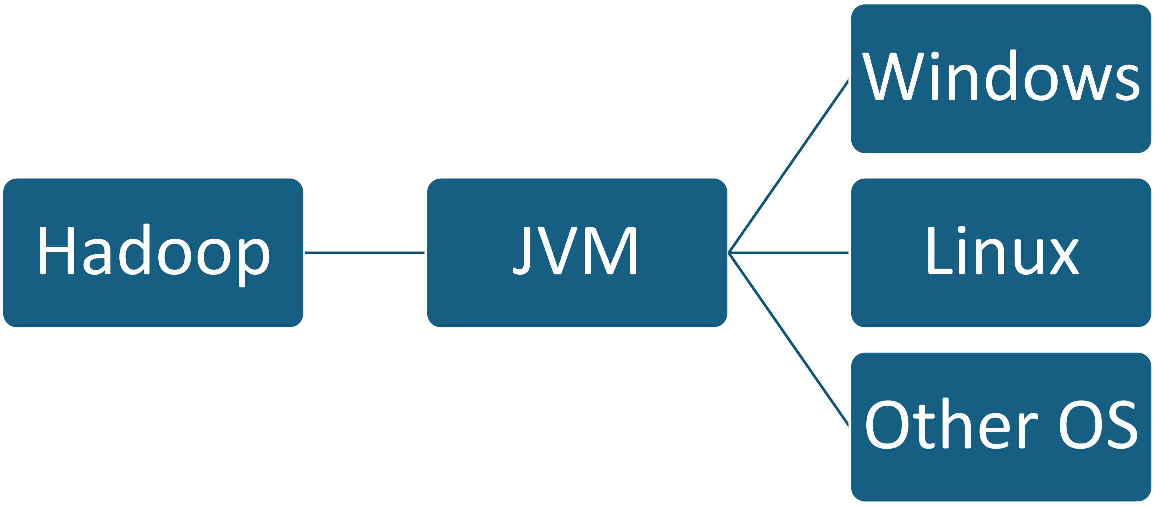


Hadoop Installation and Configuration

Outlines

Hadoop is written by Java and supports Windows and Linux OS.

Java programs, such as Hadoop, should be run in Java Virtual Machines (JVM) so that Java developers do not have to consider the differences between underlying operating systems, illustrated as follows. Developers can just use the interfaces provided by Java so that Java provides good portability for its programs.



In this tutorial, Hadoop will run in **Pseudo-Distributed Mode**. Although Hadoop is a distributed framework, all nodes will run in a single machine but in different processes.

Hadoop Installation

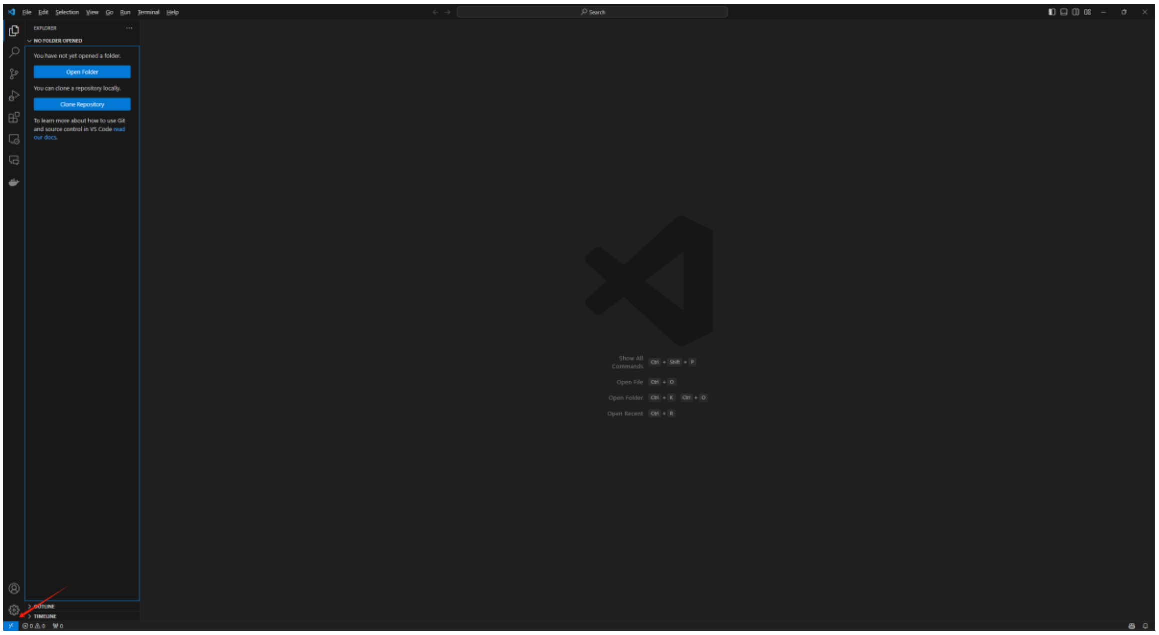
Java Development Kit (JDK) Installation

Since Hadoop is written in Java, we must install JVM to run Hadoop.

Java Development Kit (JDK) is a kit which provides JVM and other tools for Java programming.

JDK also has a subset called Java Runtime Environment (JRE) which only supports running Java programs (i.e., JVM and other utilities). JRE cannot be used to write Java.

1. Open VSCode and click the icon in the bottom-left corner.

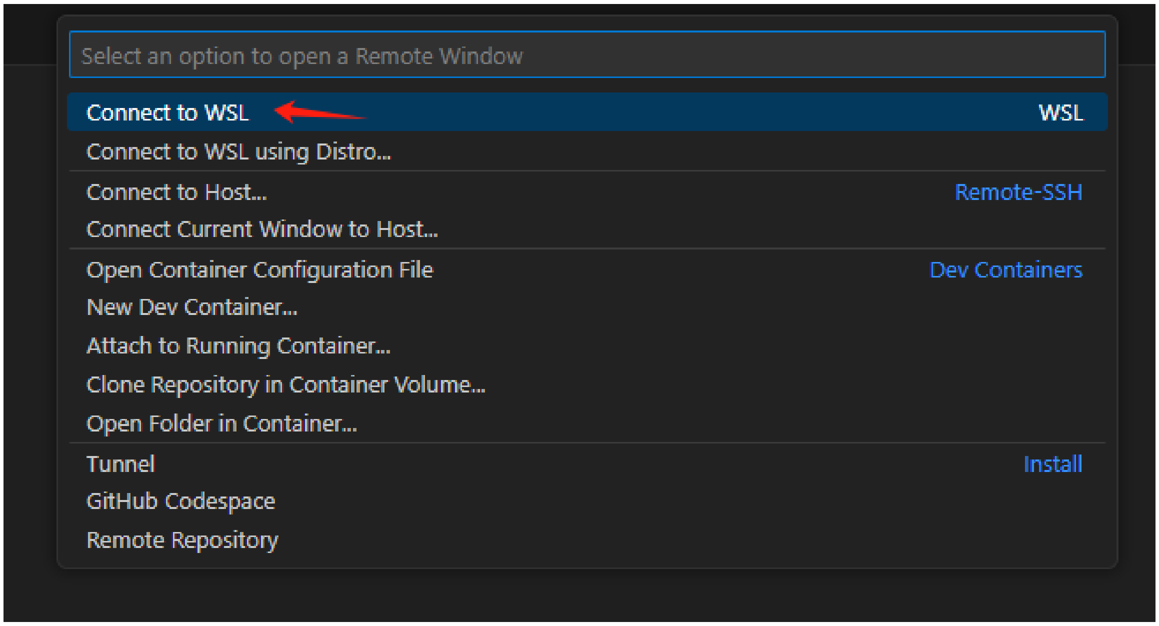


You have not yet opened a folder.

You can clone a repository locally.

To learn more about how to use Git and source control in VS Code read our docs.

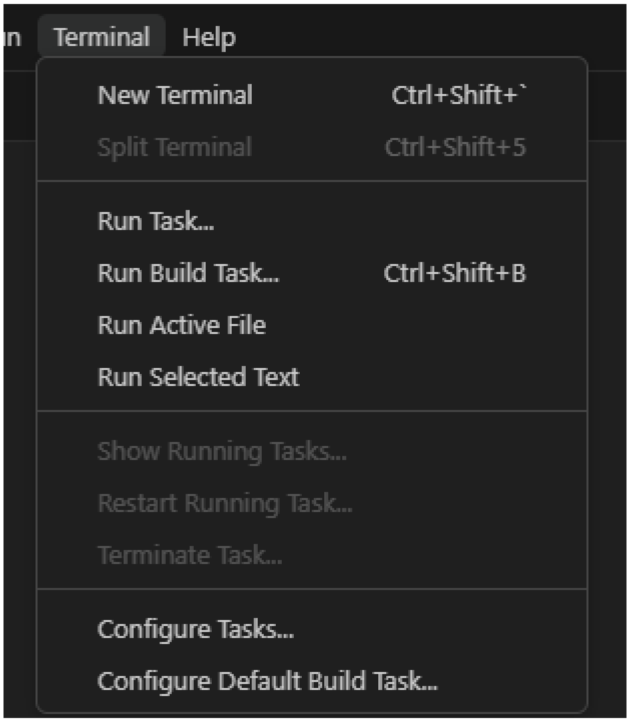
2. Choose *Connect to WSL*.



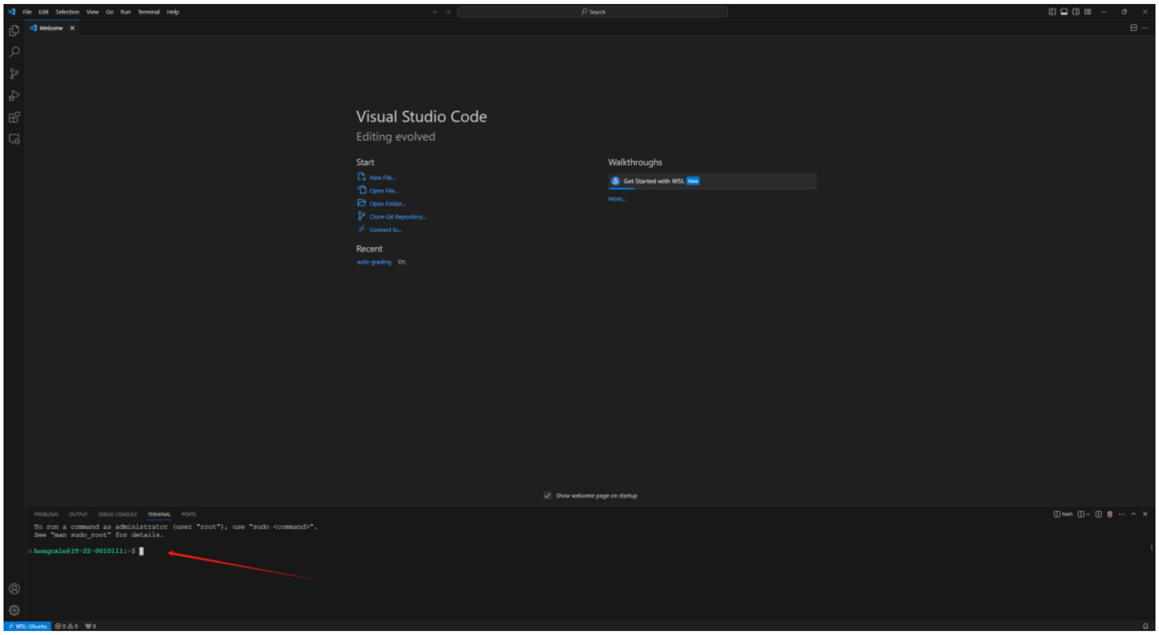
Select an option to open a Remote Window

Connect to WSL	WSL
Connect to WSL using Distro...	
Connect to Host...	Remote-SSH
Connect Current Window to Host...	
Open Container Configuration File	Dev Containers
New Dev Container...	
Attach to Running Container...	
Clone Repository in Container Volume...	
Open Folder in Container...	
Tunnel	Install
GitHub Codespace	
Remote Repository	

3. Open *Terminal* menu and choose *New Terminal*.

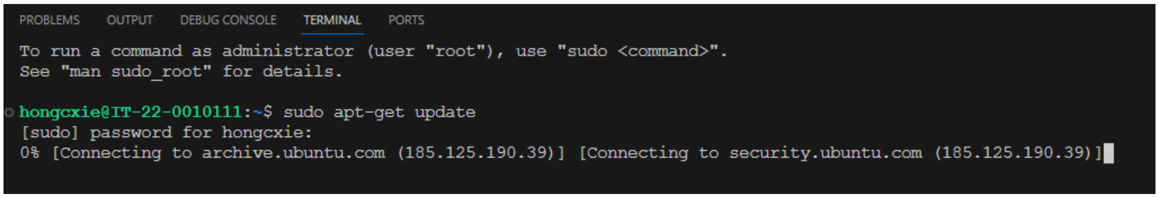


4. You can see the shell of WSL2 here.



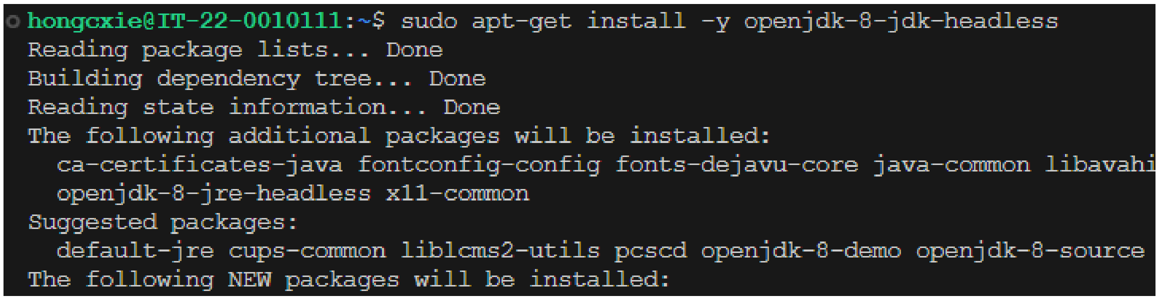
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hongxie@IT-22-0010111:~$ 
```

5. Type *sudo apt-get update* to update the index of package manager. (You may need to input your password)



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hongcxie@IT-22-0010111:~$ sudo apt-get update
[sudo] password for hongcxie:
0% [Connecting to archive.ubuntu.com (185.125.190.39)] [Connecting to security.ubuntu.com (185.125.190.39)]
```

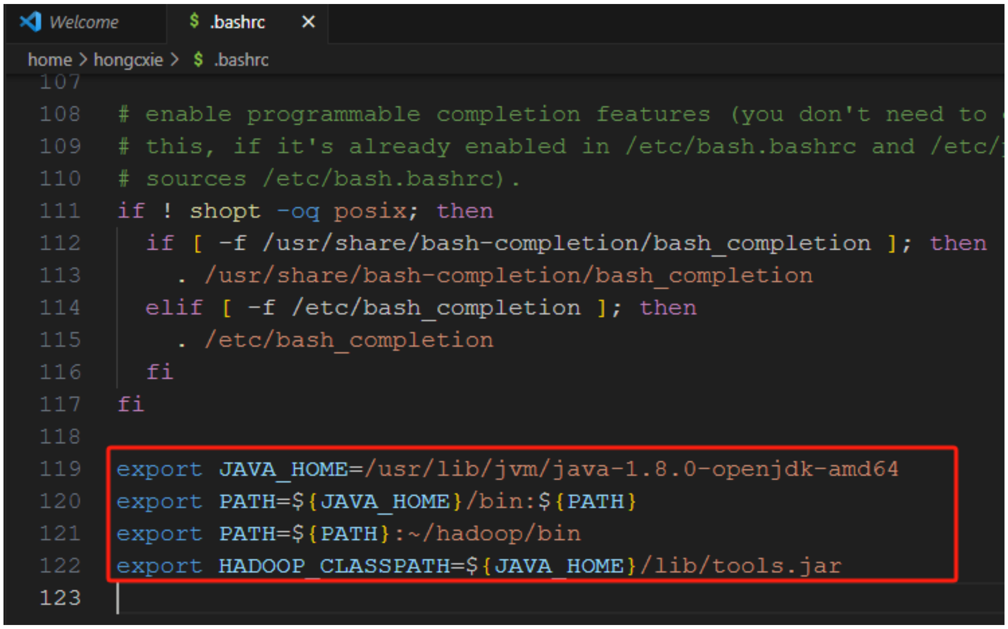
6. Type *sudo apt-get install -y openjdk-8-jdk-headless* to install JDK in Linux.



```
hongcxie@IT-22-0010111:~$ sudo apt-get install -y openjdk-8-jdk-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common libavahi
  openjdk-8-jre-headless x11-common
Suggested packages:
  default-jre cups-common liblcms2-utils pcscd openjdk-8-demo openjdk-8-source
The following NEW packages will be installed:
```

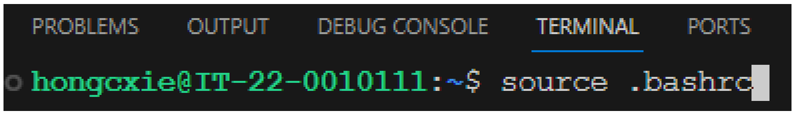
7. Open *File* menu and choose *Open Files*. Choose *.bashrc* to setup environment variables for Java and Hadoop program. Add lines as follow.
    a. JAVA_HOME may be different in different OS (WSL2 uses Ubuntu by default). Please use search engine to check your JDK path.

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

export PATH=${JAVA_HOME}/bin:${PATH}

export PATH=${PATH}:~/hadoop/bin

export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```



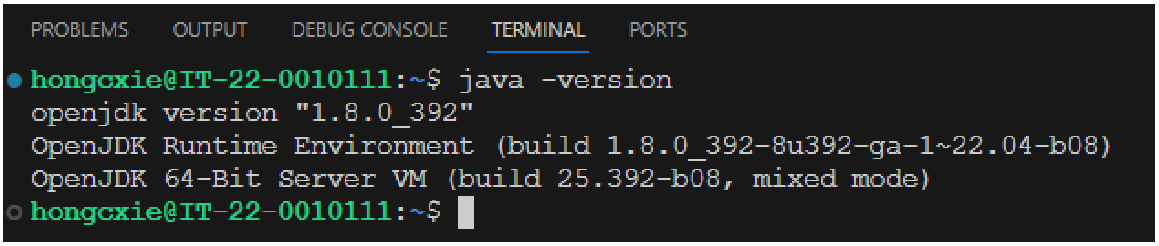
```
# enable programmable completion features (you don't need to
# this, if it's already enabled in /etc/bash.bashrc and /etc/
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
export PATH=${JAVA_HOME}/bin:${PATH}
export PATH=${PATH}:~/hadoop/bin
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

8. Save and Exit.
9. Type *source .bashrc* and press *Enter* in the shell to update our modification.



```
hongcxie@IT-22-0010111:~$ source .bashrc
```

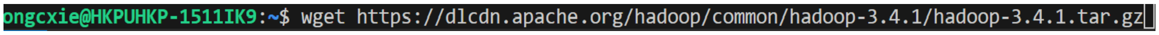
10. Type *java -version* to check whether the installation is successful. Your result should be similar with this.



```
hongcxie@IT-22-0010111:~$ java -version
openjdk version "1.8.0_392"
OpenJDK Runtime Environment (build 1.8.0_392-8u392-ga-1~22.04-b08)
OpenJDK 64-Bit Server VM (build 25.392-b08, mixed mode)
hongcxie@IT-22-0010111:~$
```

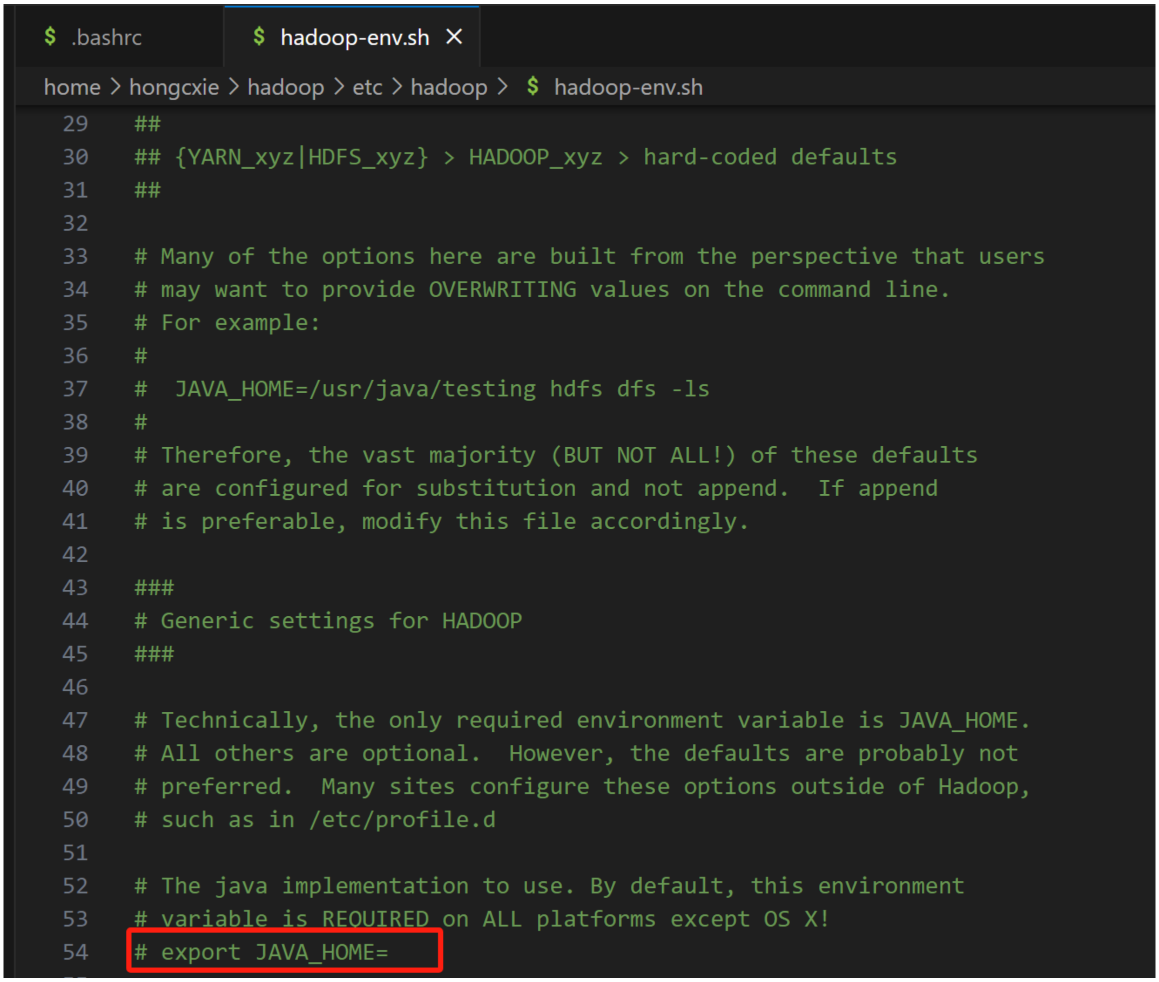
Hadoop Installation

1. Type *wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.1/hadoop-3.4.1.tar.gz* and press *Enter* to download the compressed Hadoop.



```
ongcxie@HKPUHKP-1511IK9:~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.1/hadoop-3.4.1.tar.gz
```

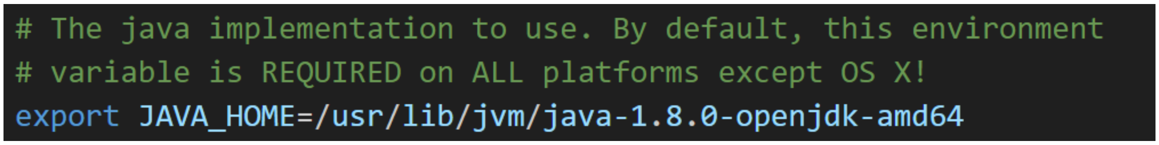
2. Type *tar -zxvf hadoop-3.4.1.tar.gz* and press *Enter* to decompress it in a folder.
3. Type *mv hadoop-3.4.1 hadoop* and press *Enter* to rename the folder.
4. Use VSCode to open file hadoop/etc/hadoop/hadoop-env.sh



```
##
## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
##

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
#  JAVA_HOME=/usr/java/testing hdfs dfs -ls
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append.  If append
# is preferable, modify this file accordingly.

###
# Generic settings for HADOOP
###

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
# export JAVA_HOME=
```

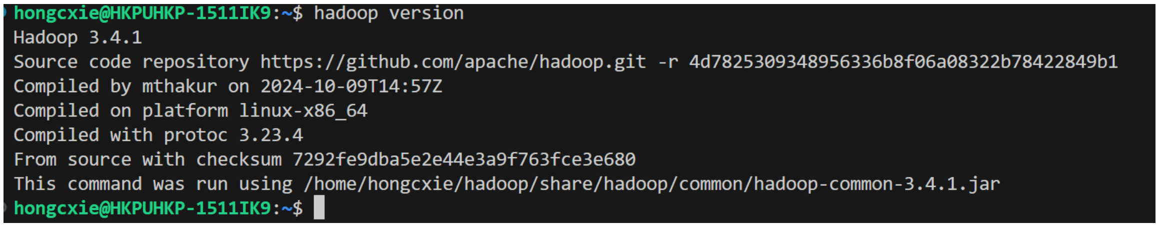
5. Change the above line to

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```



```
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

6. Save and exit.
7. Type *hadoop version* and press *Enter* to check whether Hadoop is installed properly. Your result should be like this.



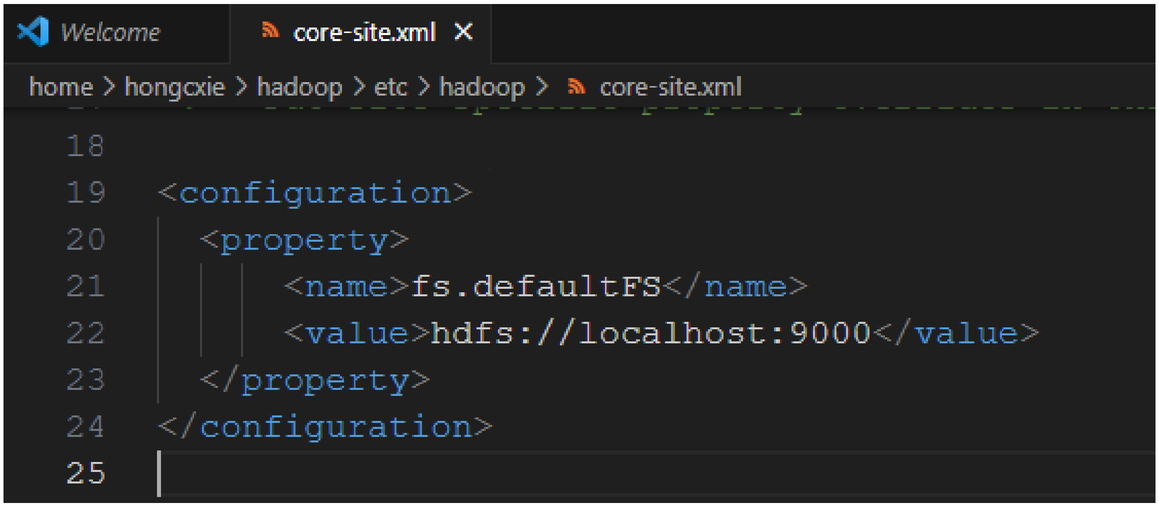
```
hongcxie@HKPUHKP-1511IK9:~$ hadoop version
Hadoop 3.4.1
Source code repository https://github.com/apache/hadoop.git -r 4d7825309348956336b8f06a08322b78422849b1
Compiled by mthakur on 2024-10-09T14:57Z
Compiled on platform linux-x86_64
Compiled with protoc 3.23.4
From source with checksum 7292fe9dba5e2e44e3a9f763fce3e680
This command was run using /home/hongcxie/hadoop/share/hadoop/common/hadoop-common-3.4.1.jar
hongcxie@HKPUHKP-1511IK9:~$ 
```

Hadoop Configuration

In this section, we will configure Hadoop to enable Pseudo-Distributed Mode.

1. Use VSCode to open file hadoop/etc/hadoop/core-site.xml and add the content like this.

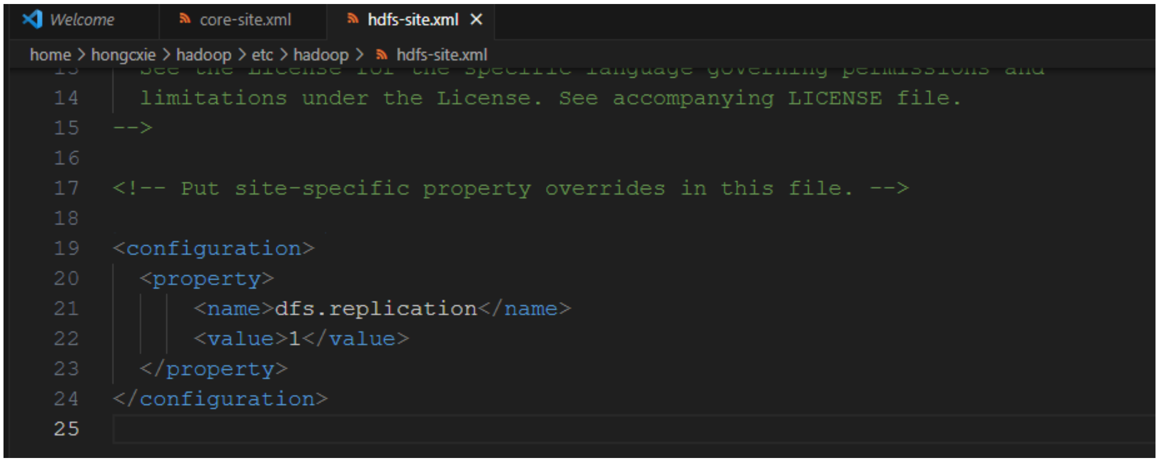
```
<configuration>
  <property>
      <name>fs.defaultFS</name>
      <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```



```

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

```

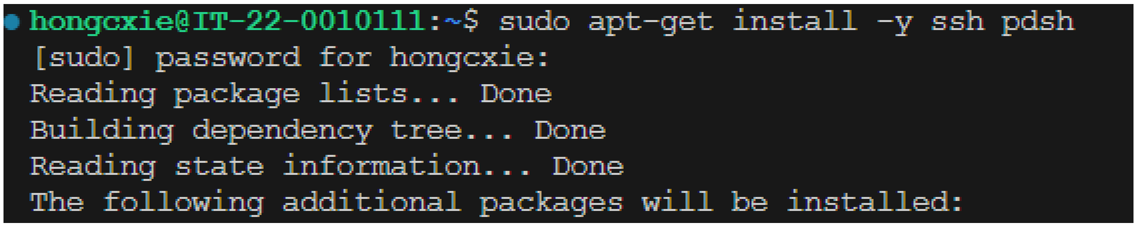
2. Save and exit.
3. Use VSCode to open file hadoop/etc/hadoop/hdfs-site.xml and add the content like this.

```
<configuration>

  <property>

      <name>dfs.replication</name>

      <value>1</value>

  </property>

</configuration>
```



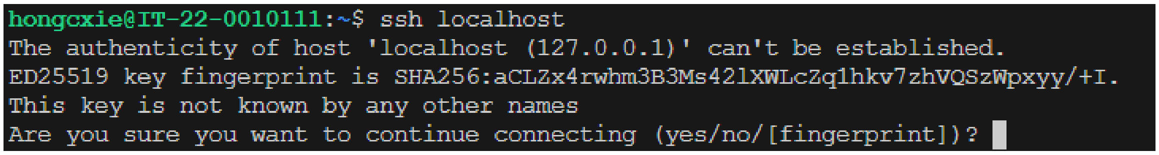
```
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
      <name>dfs.replication</name>
      <value>1</value>
  </property>
</configuration>

```

4. Save and exit.
5. Type *sudo apt-get install -y ssh pdsh* and press *Enter* to install ssh.



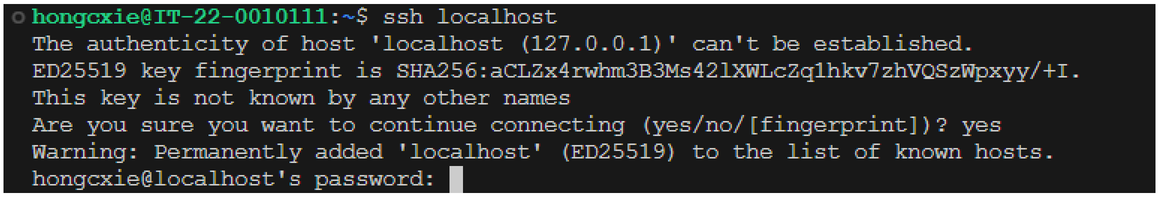
```
hongcxie@IT-22-0010111:~$ sudo apt-get install -y ssh pdsh
[sudo] password for hongcxie:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

6. Type *ssh localhost* and press *Enter* to check whether you need a passphrase to login. (You may need to type *yes* if the shell displays this message.)



```
hongcxie@IT-22-0010111:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:aCLZx4rwhm3B3Ms42lXWLcZq1hkv7zhVQSzWpxyy/+I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

   a. If the shell display this, you need to press *Ctrl+C* to exit.

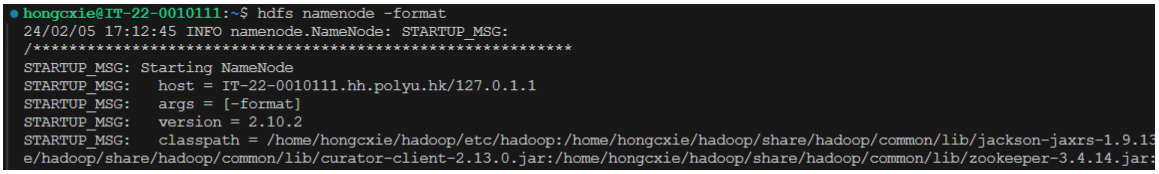


```
hongcxie@IT-22-0010111:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:aCLZx4rwhm3B3Ms421XWLcZq1hkv7zhVQSzWpxyy/+I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
hongcxie@localhost's password:
```

i. Run these commands one by one to enable signing without a passphrase.

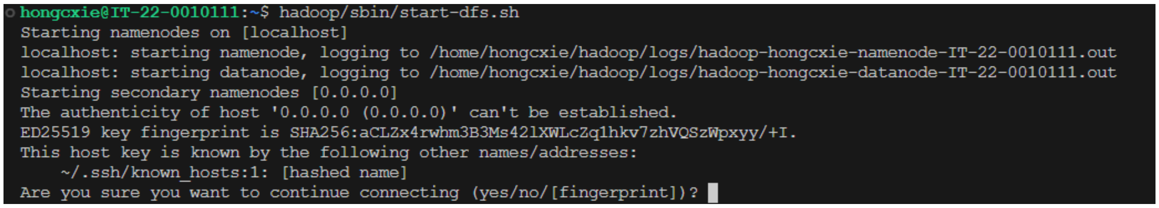
```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

chmod 0600 ~/.ssh/authorized_keys
```

7. Run *hdfs namenode -format* to format the distributed file system.



```
hongcxie@IT-22-0010111:~$ hdfs namenode -format
24/02/05 17:12:45 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = IT-22-0010111.hh.polyu.hk/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.10.2
STARTUP_MSG:   classpath = /home/hongcxie/hadoop/etc/hadoop:/home/hongcxie/hadoop/share/hadoop/common/lib/jackson-jaxrs-1.9.13
e/hadoop/share/hadoop/common/lib/curator-client-2.13.0.jar:/home/hongcxie/hadoop/share/hadoop/common/lib/zookeeper-3.4.14.jar:
```

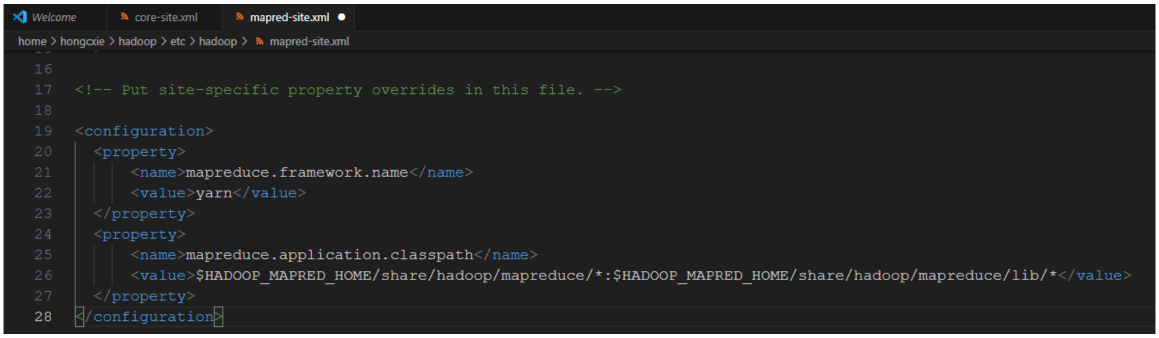
8. Run *hadoop/sbin/start-dfs.sh* to start distributed file system. (You may need to type *yes* and press Enter)



```
hongcxie@IT-22-0010111:~$ hadoop/sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hongcxie/hadoop/logs/hadoop-hongcxie-namenode-IT-22-0010111.out
localhost: starting datanode, logging to /home/hongcxie/hadoop/logs/hadoop-hongcxie-datanode-IT-22-0010111.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ED25519 key fingerprint is SHA256:aCLZx4rwhm3B3Ms421XWLcZq1hkv7zhVQSzWpxyy/+I.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

9. Open *hadoop/etc/hadoop/mapred-site.xml* in VSCode and add content like this.

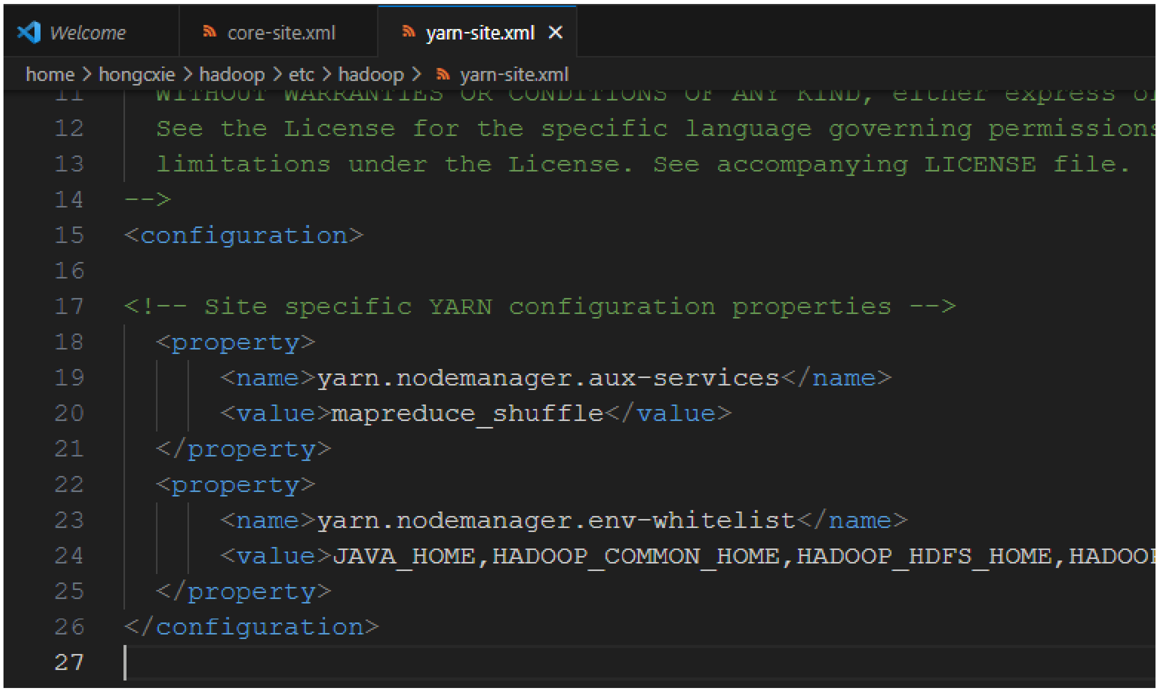
```
<configuration>
  <property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
  </property>
  <property>
      <name>mapreduce.application.classpath</name>

<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```



```

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
  </property>
  <property>
      <name>mapreduce.application.classpath</name>
      <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```

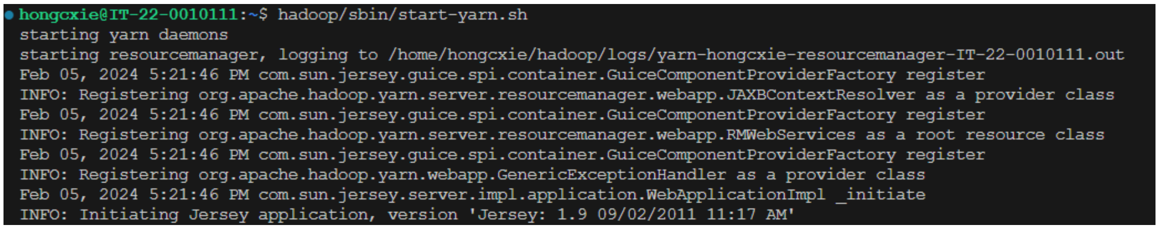
10. Save and exit.
11. Open *hadoop/etc/hadoop/yarn-site.xml* in VSCode and add content like this.

```
<configuration>

  <property>

      <name>yarn.nodemanager.aux-services</name>

      <value>mapreduce_shuffle</value>

  </property>

  <property>

      <name>yarn.nodemanager.env-whitelist</name>


<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_HOME,PATH,LANG,TZ,HADOOP_MAPRED_HOME</value>

  </property>

</configuration>
```



```
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
  See the License for the specific language governing permissions
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
  <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
  </property>
  <property>
      <name>yarn.nodemanager.env-whitelist</name>
      <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP
  </property>
</configuration>
```

12. Save and exit.
13. Run *hadoop/sbin/start-yarn.sh*.



```
● hongcxie@IT-22-0010111:~$ hadoop/sbin/start-yarn.sh
  starting yarn daemons
  starting resourcemanager, logging to /home/hongcxie/hadoop/logs/yarn-hongcxie-resourcemanager-IT-22-0010111.out
  Feb 05, 2024 5:21:46 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
  INFO: Registering org.apache.hadoop.yarn.server.resourcemanager.webapp.JAXBContextResolver as a provider class
  Feb 05, 2024 5:21:46 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
  INFO: Registering org.apache.hadoop.yarn.server.resourcemanager.webapp.RMWebServices as a root resource class
  Feb 05, 2024 5:21:46 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
  INFO: Registering org.apache.hadoop.yarn.webapp.GenericExceptionHandler as a provider class
  Feb 05, 2024 5:21:46 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
  INFO: Initiating Jersey application, version 'Jersey: 1.9 09/02/2011 11:17 AM'
```

Tutorial: Word Count

Outlines

In this tutorial, we will write a program which leverages MapReduce in Hadoop to perform Word Count (WC).

Definition of WC

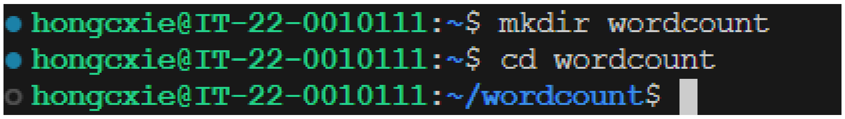
Input: Text

Output: the count of occurrences of each term in the text

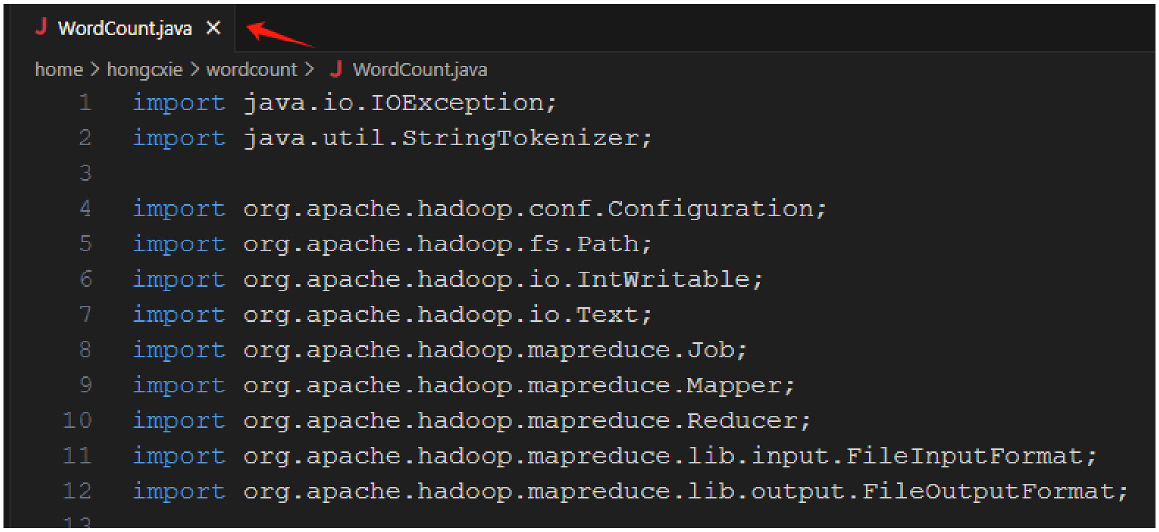
Source Code

Please refer to the attached file (WordCount.java).

You can run *mkdir wordcount* in the shell to create a folder called *wordcount*, and save the source code in this folder with the name *WordCount.java* (note: follow the letter case strictly).



```
hongcxie@IT-22-0010111:~$ mkdir wordcount
hongcxie@IT-22-0010111:~$ cd wordcount
hongcxie@IT-22-0010111:~/wordcount$
```



```java
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

Compile Source Code

Run the following commands one by one.

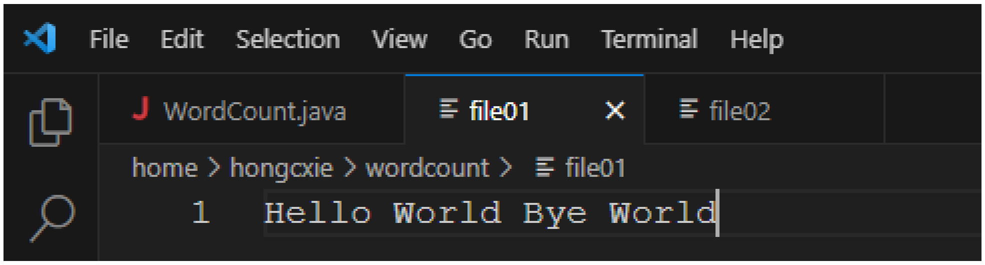
```
hadoop com.sun.tools.javac.Main WordCount.java

jar cf wc.jar WordCount*.class
```

The above commands will compile the java file and create a .jar file (the file package which aggregates multiple Java class files).

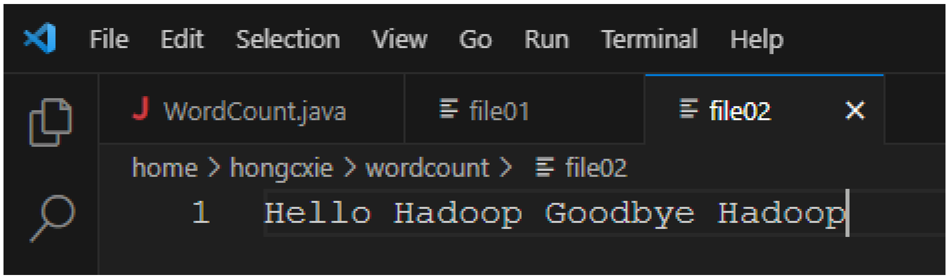
Create two files for experiment

Use VSCode to create two files with names *file01* and *file02* in *wordcount* folder.

The contents of each file are shown as follows.

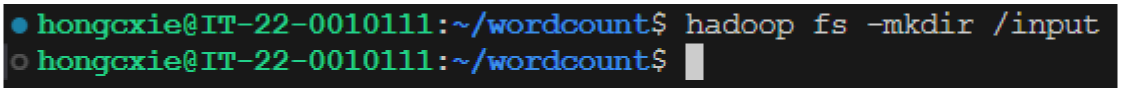


```
Hello World Bye World
```



```
Hello Hadoop Goodbye Hadoop
```

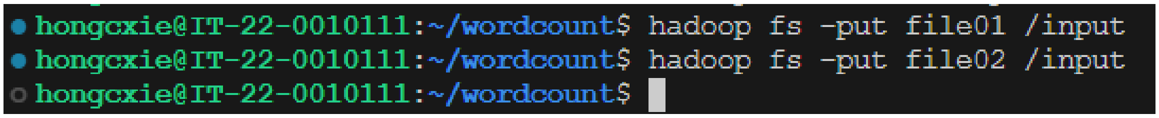
Run *hadoop fs -mkdir /input* to create a folder (/input) in distributed file system.



```
hongcxie@IT-22-0010111:~/wordcount$ hadoop fs -mkdir /input
hongcxie@IT-22-0010111:~/wordcount$ 
```

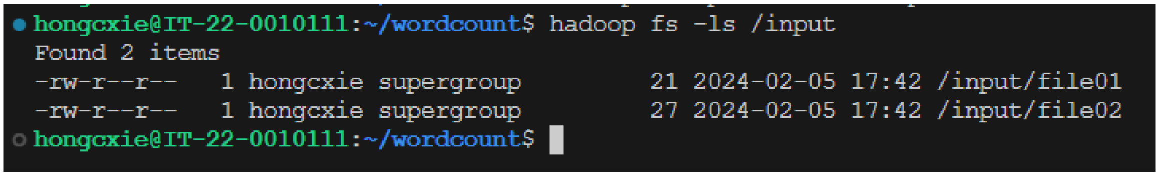
Run the following commands to copy *file01* and *file02* to */input* in the distributed file system.

```
hadoop fs -put file01 /input

hadoop fs -put file02 /input
```



```
hongcxie@IT-22-0010111:~/wordcount$ hadoop fs -put file01 /input
hongcxie@IT-22-0010111:~/wordcount$ hadoop fs -put file02 /input
hongcxie@IT-22-0010111:~/wordcount$ 
```

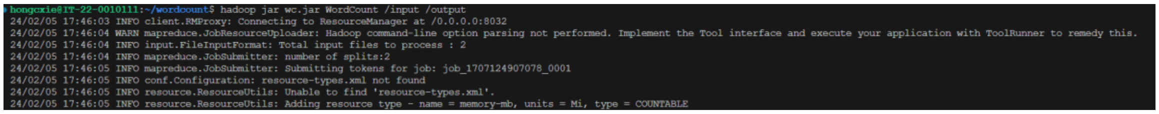
Run *hadoop fs -ls /input* to see the content of this folder.



```
hongcxie@IT-22-0010111:~/wordcount$ hadoop fs -ls /input
Found 2 items
-rw-r--r--   1 hongcxie supergroup         21 2024-02-05 17:42 /input/file01
-rw-r--r--   1 hongcxie supergroup         27 2024-02-05 17:42 /input/file02
hongcxie@IT-22-0010111:~/wordcount$ 
```

Run MapReduce application WC

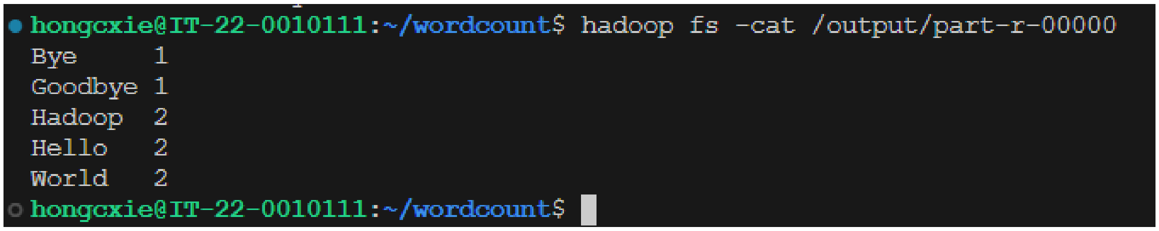
Run *hadoop jar wc.jar WordCount /input /output* to launch our application in Hadoop.



```
hongcxie@IT-22-0010111:~/wordcount$ hadoop jar wc.jar WordCount /input /output
24/02/05 17:46:03 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
24/02/05 17:46:04 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
24/02/05 17:46:04 INFO input.FileInputFormat: Total input files to process : 2
24/02/05 17:46:04 INFO mapreduce.JobSubmitter: number of splits:2
24/02/05 17:46:05 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1707124907078_0001
24/02/05 17:46:05 INFO conf.Configuration: resource-types.xml not found
24/02/05 17:46:05 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
24/02/05 17:46:05 INFO resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
```

The output is written in the folder /output in the distributed file system.

Run *hadoop fs -cat /output/part-r-00000* to see the result.



```
hongcxie@IT-22-0010111:~/wordcount$ hadoop fs -cat /output/part-r-00000
Bye     1
Goodbye 1
Hadoop  2
Hello   2
World   2
hongcxie@IT-22-0010111:~/wordcount$ 
```