

# CSE3BDC/CSE5BDC

## Lab 02: Hadoop MapReduce and EMR

Department of Computer Science and IT, La Trobe University

### Task A: Using the Cloudera VM

A company called Cloudera provides a free Linux distribution which allows you to easily run big data processing applications locally on a single computer. From this lab onwards, we will be using a modified Cloudera virtual machine (VM) as our work environment.

Please follow the steps in the “Cloudera Instructions” document from LMS to extract and start the VM. Once it’s booted, continue reading to learn about basic usage of the VM.

#### File browser

Browsing files using the GUI in the VM is very similar to Windows. There are two shortcuts on the desktop for launching the file browser:



**cloudera’s Home** Start the file browser in the home directory `/home/cloudera`. This is useful for accessing the Downloads folder, for example.

**CSE3BDC\_Files** Start the file browser in a shared folder mapped to your student drive. You can use this to move files to and from Windows, and to save work that you’ve done during the lab.

One thing to be aware of is that file access in **CSE3BDC\_Files** is slow and unstable, since it corresponds to a network mapped Unix drive. So I *strongly* recommend doing your work in another location (for example, the desktop), then copying the files to **CSE3BDC\_Files** afterwards. Please be aware that files you create which are not in your student drive **will be erased** after the lab (this includes files on the desktop), so you will want to copy your files across before leaving the lab.


**IMPORTANT: Make sure that all of your work is copied to your student drive (CSE3BDC\_Files) before leaving the lab!**

Now we will quickly check that the shared folder for your student drive is working correctly.

1. Double-click on the **CSE3BDC\_Files** desktop icon. You should be able to see the files already stored on your student drive—if not, please ask a demonstrator for assistance.
2. Create a folder using the file browser (Right-click > Create Folder). Name the folder “CSE3BDC” (or “CSE5BDC”). You can use this folder to store subject-related material between labs.

3. To check that everything worked as expected, open up your student drive in Windows and confirm that the new folder is there. You might need to press F5 to refresh the display.
4. Go back to the VM. We won't be using Windows directly anymore in this lab.

## Web browser

Firefox is installed in the VM. You can open it by clicking on the  Preferred Web Browser icon in the panel at the top of the screen.

We will now use Firefox to download this week's lab files inside the VM.

1. Open Firefox.
2. Log into LMS and download this week's lab to the desktop.
3. Extract the lab files with Right Click > Extract Here.

You should also follow these steps in future labs to get started.

## Terminal

In this subject there will be many times when commands will need to be run in a terminal window. You can open a terminal window from inside any folder using Right Click > Open in Terminal.

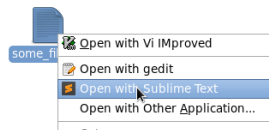
1. Open up the lab files folder.
2. Open up a terminal from that folder, and use the `ls` command to list files.

```
$ ls
```

The command output should list the same files that you can see in the file browser GUI.

## Text editor

There are a few text editors installed in the VM. If you don't have a preference, I recommend using Sublime Text. To edit a file in Sublime Text, just Right Click > Open with Sublime Text.



You can also start Sublime Text from the terminal using the `subl` command. This is especially useful for opening an entire directory. This command opens the current directory in Sublime:

```
$ subl ./
```


## Task B: Working with HDFS

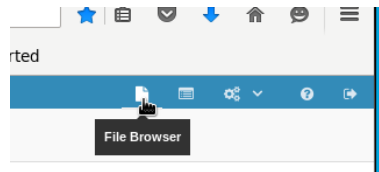
Now that you have started the VM and familiarised yourself with the basics, we can move on to some big data-related topics! We will begin by looking at how to store data using HDFS (Hadoop Distributed File System). HDFS is a special file system designed to make it easy to share large amounts of data across many computers in a cluster. In the big data world, it is very common for applications to read input from and/or write output to HDFS.

### Copying files into HDFS

1. Open a terminal from the lab files folder (the folder containing `small_data`).
2. Put the file `small_data` into HDFS:

```
$ hdfs dfs -put small_data
```

3. Open up Firefox and click on the [Hue](#) bookmark. Hue is Cloudera's web interface for a few different big data services, including HDFS.
4. If you are prompted to sign in, do so with username "cloudera" and password "cloudera".
5. Click the  [File Browser](#) link in the top-right of the page to bring up the HDFS file explorer.



6. You should be able to see the `small_data` file that we just put into HDFS. Nice! Notice that the file was placed in the `/user/cloudera` directory by default, so the full path to the file in HDFS is `/user/cloudera/small_data`.
7. Click on `small_data` to preview the text contents of the file. You should see lines like:

```
464    0
13     0
78     0
205    0
...
```

8. You can use the arrows near the top-right of the page to browse different parts of the file. Since it is common to store very large data files in HDFS, so it makes sense that the preview doesn't try to show the whole file at once.

## Task C: Running MapReduce jobs

The file we put into HDFS in the last task, `small_data`, contains two columns of data. The first column is the document ID, and the second column is a word. So the line “301 wish” indicates that the document with ID 301 contains the word “wish”. Duplicate entries indicate multiple occurrences of a word in a file.



We will now run a simple MapReduce application on the data which finds the longest word from each document. The complete source code for the application is included in the `Task_C` folder. The code is quite short and well documented, so feel free to take a moment to read it but don’t make any changes yet.

1. Open a terminal in the `Task_C` folder, then build the application into a Java `.jar` file.

```
$ javac -classpath `hadoop classpath` *.java
$ jar cvfe longest_word.jar Main *.class
```

2. Run the application.

```
$ hadoop jar longest_word.jar small_data longest_word_output
```

3. Open Hue in Firefox.
4. Click on the  Job Browser link. You should see a job called “Longest word”. Click on it. If the job is still running, wait until the status changes to “Succeeded” before moving on to the next step (this may take a minute).
5. Click on the “Counters” tab to list metrics gathered while the job was running. Use Ctrl+F to find the total for “Reduce Shuffle Bytes”. This indicates the amount of data that would need to be shuffled between machines in a cluster to complete the job. Make a note of this value (it should be around 15 megabytes).
6. Click on the  File Browser link. Notice that the output of the application has appeared, `longest_word_output`. After clicking on it you will notice that there are three output files, one for each reducer. The number of reducers was specified in `Main.java`. Take a brief look at their contents to see what the per-document longest words look like.

## Adding a combiner

We have verified that our simple MapReduce application works, but unfortunately it’s not very well optimised. In fact, the amount of data that it sends through the network during the data phase is roughly equal to the total size of the data. By using a combiner, we can reduce this number greatly.

1. Open `Main.java` in a text editor.
  - (a) Change the job name from “Longest word” to “Longest word (with combiner)”.
  - (b) Uncomment the line which sets the combiner (look for the `TODO` comment).

2. Execute the two commands to build the application into a Java `.jar` file again.
3. Run the application, but this time use `combiner_output` as the output directory instead of `longest_word_output`.
4. Once the “Longest word (with combiner)” job has finished running, check the “Reduce Shuffle Bytes”. If everything went well, it should indicate that less than 1/1000th of the data has been shuffled. Much better!

## Task D: MapReduce in the cloud

An EMR (Elastic Map Reduce) cluster is a group of computation resources provided by AWS, designed and configured to run any MapReduce tasks you can throw at it. Spinning up a cluster is like building your own computation-oriented server farm—configured for Hadoop MapReduce Tasks—in well under an hour. We will be using EMR to run the “Longest word” program from the previous task on a larger dataset.

Please use the web browser inside the Cloudera VM to complete this task. **Do not use Windows like we did in Lab 01.**

### Creating a cluster

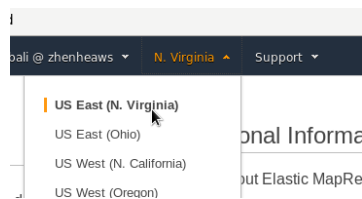
Spinning up an EMR cluster is a similar process to launching an EC2 Instance from Lab 01.

1. Login to AWS by using the following URL: <https://zhenheaws.signin.aws.amazon.com/console>.
2. You should have already created a folder for yourself on S3 during Lab 01 at the following path:

```
s3://latrobe-bdc/<student number>
```

If not, please create it now:

- (a) Navigate to the S3 console (Services > S3).
  - (b) Click on “latrobe-bdc”, since this is the bucket that we want to create the directory in.
  - (c) Click on the “Create folder” button, and enter your student number as the folder name. Leave the encryption settings on the default option, “None (Use bucket settings)”.
3. Navigate to the EMR console (Services > EMR, under the “Analytics” heading). From the next screen, you can view all the clusters that have been or are running under the AWS IAM account you are using. For this lab, you will be creating your own cluster and reusing it for each task.
  4. First make sure that you select the US East (N. Virginia) region from the top-right of the page.



5. Click on Create Cluster to begin the process of launching a new cluster.
  - (a) In the next screen, set the “Cluster name” to a value that includes your student number, like “<student\_number>-cluster”.

- (b) Ensure that the “Logging option” is **enabled**. For the “S3 folder” location specify your S3 user sub directory, with the subdirectory “/emr\_logs/” appended to the end. For example (do not copy and paste, please type this in):

```
s3://latrobe-bdc/<student number>/emr_logs/
```

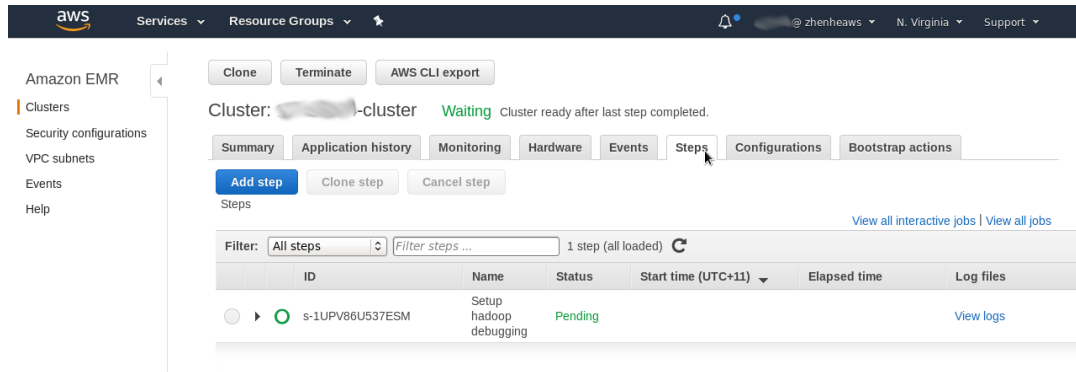
- (c) Keep the “Launch mode” selected as **Cluster**.
  - (d) Under the “Software configuration” section, ensure that the “Core Hadoop” application is selected (this should be the default).
  - (e) Under the “Hardware configuration” section, ensure that that the “Instance type” is **m3.xlarge**, and that the “Number of instances” is **3**.
  - (f) Under the “Security and access” section, set the EC2 key pair to the one that you generated in Lab 01. If you have lost your key pair .pem file, you will need to create a new key pair and use it instead. Leave the other options with their default values.
6. Click “Create Cluster”. AWS will start provisioning for the requested hardware, then configure and boot up the instances. This may take a few minutes, so you should move on with the remainder of this lab while you wait. Be sure to check back occasionally on the status of the cluster by clicking the refresh icon on the page, until the cluster is shown as Running or Waiting.

## Deploying to EMR

Deploying a Map Reduce task on EMR is a relatively simple process. All you need is an internet connection, an EMR cluster (for computation), an S3 bucket (for data storage and logging), and a MapReduce .jar program to deploy. The good news is that you should have all of these things already! Isn’t it funny how these things tend to work out?

1. Open the AWS S3 console in a separate tab in your browser (so that you can check back on the EMR cluster more easily).
2. Navigate to the following S3 bucket: **s3://inputdatafiles**. You will notice there is a file called **Lab02Data.txt** (171.0 MB). This file is a much larger version of the **small\_data** file which we were working with earlier. Its contents follow exactly the same two-column format (document ID, word). You don’t need to download the file or anything, I just wanted to show you where the input data is stored.
3. Now we are ready to upload our MapReduce application .jar file.
  - (a) Go back to the AWS S3 console, and navigate to your folder in the **s3://latrobe-bdc** bucket.
  - (b) Create a sub-directory called **lab02jars**.
  - (c) Upload the file **longest\_word.jar** from Task C into the **lab02jars** directory. Leave all of the options with their default values.

4. Now that we've filled in a bit of time, check to see if your EMR Cluster is running yet by clicking the refresh icon within the EMR details page, and looking to see if the status label displays Running or waiting. If it is, continue ever onwards!
5. Expand the Steps field. Here you can see all the tasks that the EMR cluster has been given (including its base start-up tasks), and the status of each.

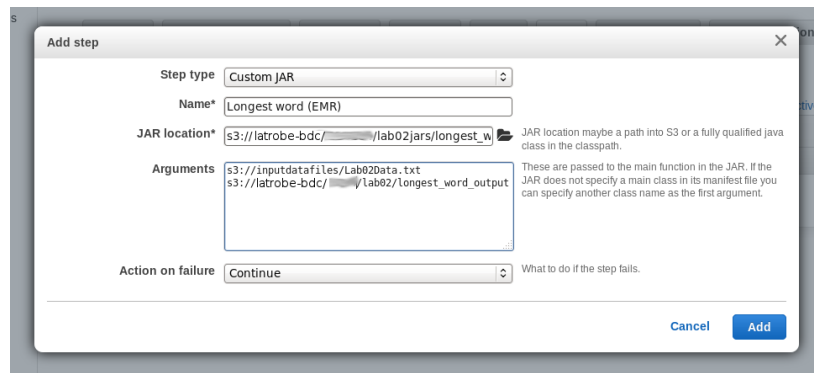


6. Click “Add Step” to specify our own job.
  - (a) For “Step type”, select **Custom JAR**. This allows you to specify a custom compiled and packaged MapReduce application.
  - (b) Name the task something simple and easy to identify, such as “Longest word (EMR)”.
  - (c) For “JAR location”, click the folder icon and navigate to the `longest_word.jar` file you uploaded to S3 earlier. Click “Select” to confirm your selection and close the file selection pop-up.
  - (d) The “Arguments” field allows you to specify arguments that are passed to your application when it’s run. Recall that the `longest_word.jar` application takes two arguments: the input file, and the location to write the output to. It is really really, really important that you type this information in correctly, otherwise your EMR job will not work. To make this easier, we have included a text file called `EMRparameters.txt` with the lab materials. You can copy and paste the parameters from the text file into the “Arguments” field (or type it in carefully), and fill in your student number. **Please do not copy and paste from this PDF.** Bad things will happen!

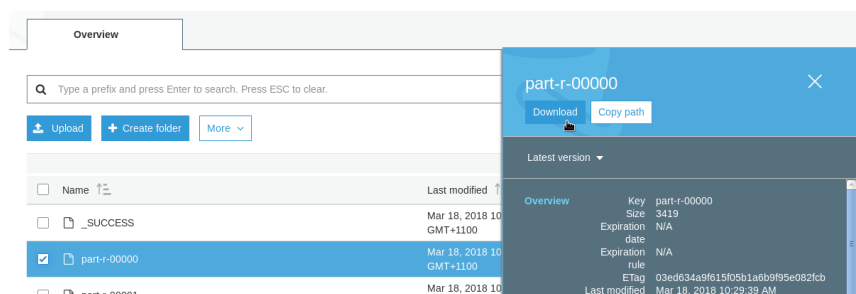
```
s3://inputdatafiles/Lab02Data.txt
s3://latrobe-bdc/<student number>/lab02/longest_word_output
```

- (a) Ensure that “Action on Failure” is set to “continue”, so that you can still execute other tasks on the EMR cluster if one of your tasks fails.
- (b) Double check that your settings are correct by comparing to the following screenshot:





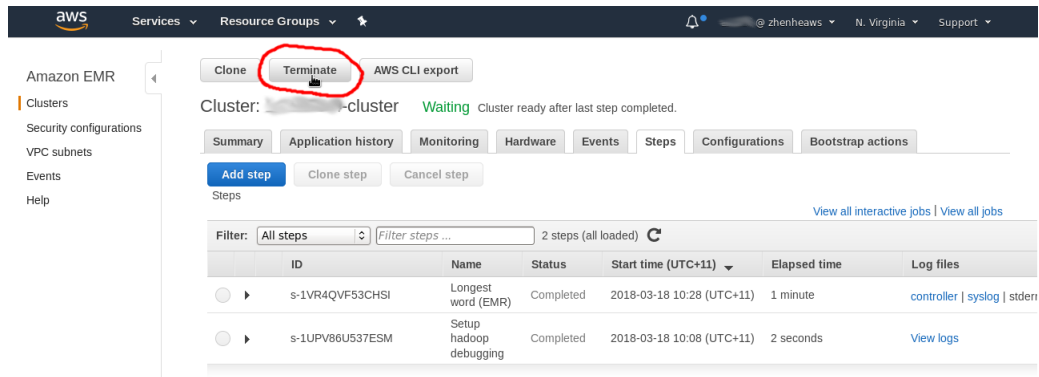
- (c) Click “Add” and the job should start. Be sure to click the refresh icon periodically to check on the status of the task. This will take around 2 minutes to complete.
7. Once the step has been completed, click “View logs”, then click “syslog”. Tell Firefox to allow the pop-up.
8. You should now be taken to a page displaying the log data, which includes execution details and metrics. This is nothing new from what you’ve seen running MapReduce jobs on the Cloudera VM, but now we have figures from a totally different cluster of machines. Of interest are the details on shuffled bytes, map output bytes, and CPU time.
9. Save the log (Right Click > Save Page As...) so that you can show your demonstrator for marking.
10. Using a different browser tab navigate to the output directory in S3, `s3://latrobe-bdc/<student number>/lab02/longest_word_output` (you may need to refresh the S3 directory to see the newly added output directory). There should be three parts to the output, just like when we ran the job in the VM.
11. Download `part-r-00000` and open it in a text editor. If everything went well, you should see a list of document IDs and really long words.



You have just deployed a MapReduce task on an international computation cluster, “in the cloud”. Congratulations! In case that doesn’t mean a lot to you, you should note that for large-scale purposes, you could provision additional, larger worker machines in order to process terabytes worth of data, in almost the same amount of time as this example has taken.

## Shutting down the cluster

Now that you've finished with your EMR cluster, you can terminate it so that it stops costing money. To do this, click the "Terminate" button, and then click "Terminate" again in the confirmation dialog that pops up.



## Leaving the lab

Once you have been marked by a demonstrator, you can pack up.

1. Make sure that you've terminated the EMR cluster.
2. Copy all of the files that you want to keep into your student drive using the CSE3BDC\_Files shared folder.
3. Shut down the VM (System > Shut Down... > Shut Down).
4. In Windows, delete the extracted VM folder (**but not the .zip file**).