Exploring Hadoop Using Hortonworks Sandbox

# Setup

* Make sure that the Sandbox is up and running
* Use [this manual](https://drive.google.com/file/d/143od6WEiasRzRZ72SPdoQmhgJItVLDYs/view?usp=sharing) to get login details and passwords

## Logging into Ambari (the web interface)

* Use the sandbox URL: <http://localhost:8080/> to log into Ambari
* For username, you can use any of the usernames provided in the manual. For example: *maria\_dev* and the password is also *maria\_dev.*

## Logging into shell through browser and the terminal

* Use <http://localhost:4200> to get to the shell. For username, use ‘root’ and password use ‘hadoop’
* You will be asked to change your root password, please do it and make sure you remember the new password.
* Identify the IP address for your sandbox. It's most likely: 127.0.0.1
* Follow the instructions in [the manual](https://drive.google.com/file/d/143od6WEiasRzRZ72SPdoQmhgJItVLDYs/view?usp=sharing) (pages 1-2) to configure terminal access. For instance, add the IP address to the lists of hosts so that when when you use ssh, you can use sandbox-hdp.hortonworks.com instead of the IP address like below



## Setup Ambari admin password manually

## Start your sandbox and open a terminal (mac or linux)

## SSH into the sandbox as root using ssh sandbox-hdp.hortonworks.com -p 2222 .

## Type the following commands:

# Updates password

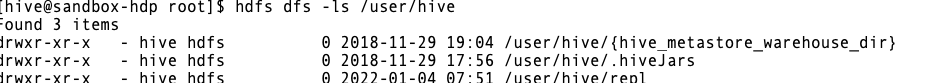
ambari-admin-password-reset

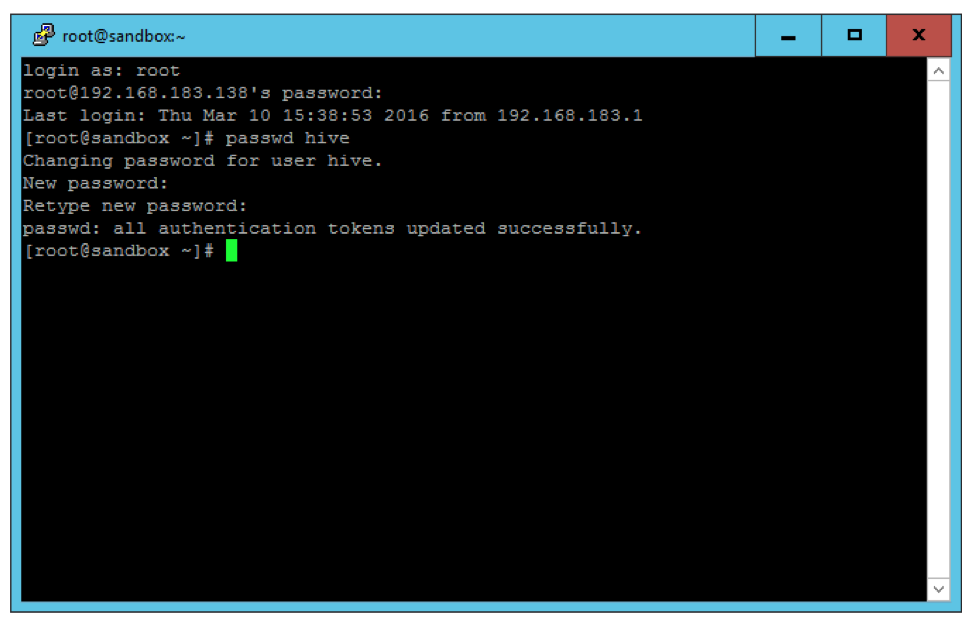
# If Ambari doesn't restart automatically, restart ambari service ambari-agent restart

## 

## Change hive password

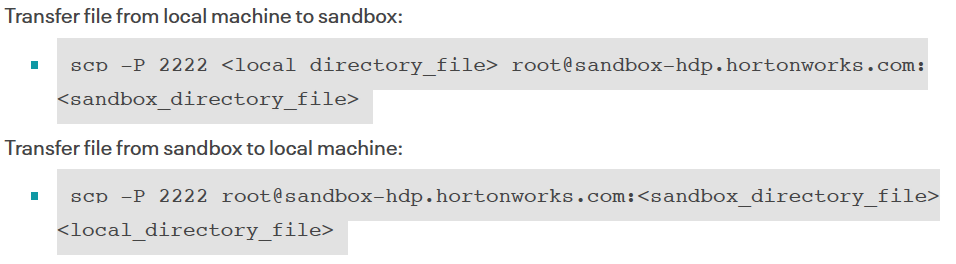
* Type this command: passwd hive to change ‘hive’ user password. As shown in the screen below
* When you have successfully change the password, try switching to hive user like this: sudo su hive. Once you changed to hive, run this HDFS command to see which files the user hive has. We will use this hive user later.





## Transferring Data between your local machine and the sand box

Open your terminal (linux or mac) or git bash (windows). To send data from your local machine to the sandbox, you would input the following command.



# Interacting with the cluster using Ambari

Navigate to Ambari and check out the different components of the cluster.

## Explore availability functionality in Ambari

* What can you do as a cluster admin person in Ambari?

## Explore the core components of Hadoop

### HDFS, MapReduce and YARN

* What configurations are available for each of these components?

# Running your first MapReduce program

In Hadoop, the hello-world program is the word count program. In the word count, as explained in the slides, we use MapReduce to count occurence of unique words in a set of input text files. Through this tutorial, you will learn the following:

* How to interact with the HDFS system; how to create directories, transfer files and more
* How to run your mapreduce jobs
* Understand structure of MapReduce programs

The tutorial we are using is [this one](https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html), you can refer to it for more details.

## Required files

Please download the following files and save to your local disk:

* **word\_count\_file.txt** from [here](https://drive.google.com/file/d/1jttz88nEa5J1RfNXPGcVhEioGWzko33l/view?usp=sharing)
* **WordCount.java** from [here](https://drive.google.com/file/d/10KmIk2I4EJsY2pWiSZIoCjCMCkgNDgJY/view?usp=sharing). Please use your favorite text editor to inspect this Java file. Make sure you can identify the map and reduce components of the program.

### Transfer these files to your sandbox

Use the instructions provided above to the two files above to your sandbox

## Connect to sandbox through terminal

I suggest you open two terminal sessions. In one, use ssh as explained above to connect to the sandbox. In the other terminal, you will use it to transfer files from your local computer to the sandbox.

## Interacting with HDFS

In order to interact with HDFS, we need to use specific commands as provided in [this guide](https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html).

### Structure of HDFS commands

* All HDFS commands follows this syntax: hdfs dfs [-command] directory details
* For example to list all folders in HDFS do the following: hdfs dfs -ls /user/
* If you run into permission issues, please add HADOOP\_USER\_NAME=hdfs, so the command above becomes: HADOOP\_USER\_NAME=hdfs hdfs dfs -ls /user/

### Create folders in the HDFS

* Create a folder in your name like this: hdfs dfs -mkdir /user/[your name]. If you run into permission issues, please use the hack/workaround explained above. Add HADOOP\_USER\_NAME=hdfs at the beginning of your command.
* Create a wordcount folder within the folder you created above: hdfs dfs -mkdir /user/[your name]/wordcount/. Next create an input folder within the wordcount folder: hdfs dfs -mkdir /user/[your name]/wordcount/input/

## Put files into HDFS

The files you moved from your local computer to your sandbox are actually on the sandbox “local” file system. In order for us to use them, we have to move them into the HDFS system. Use the -put command to achieve that. hdfs dfs -put [file name] /user/[your name]/wordcount/input/. For example, this is how I moved my word\_count\_file.txt in to the wordcount input folder: HADOOP\_USER\_NAME=hdfs hdfs dfs -put word\_count\_file.txt /user/dunstan/wordcount/input/

## Prepare JAVA environment variables

In the terminal where you are connected to the sandbox, double check that you have Java by running: java. If you get help information about the command, that means you have java. Now run the following 3 commands:

export JAVA\_HOME=/usr/java/default

export PATH=${JAVA\_HOME}/bin:${PATH}

export HADOOP\_CLASSPATH=${JAVA\_HOME}/lib/tools.jar

## Compile WordCount.java and create a jar

For Java programs, we have to compile them before we use them. While you are in the sandbox through ssh, please run the following command to compile the java WordCount file.

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

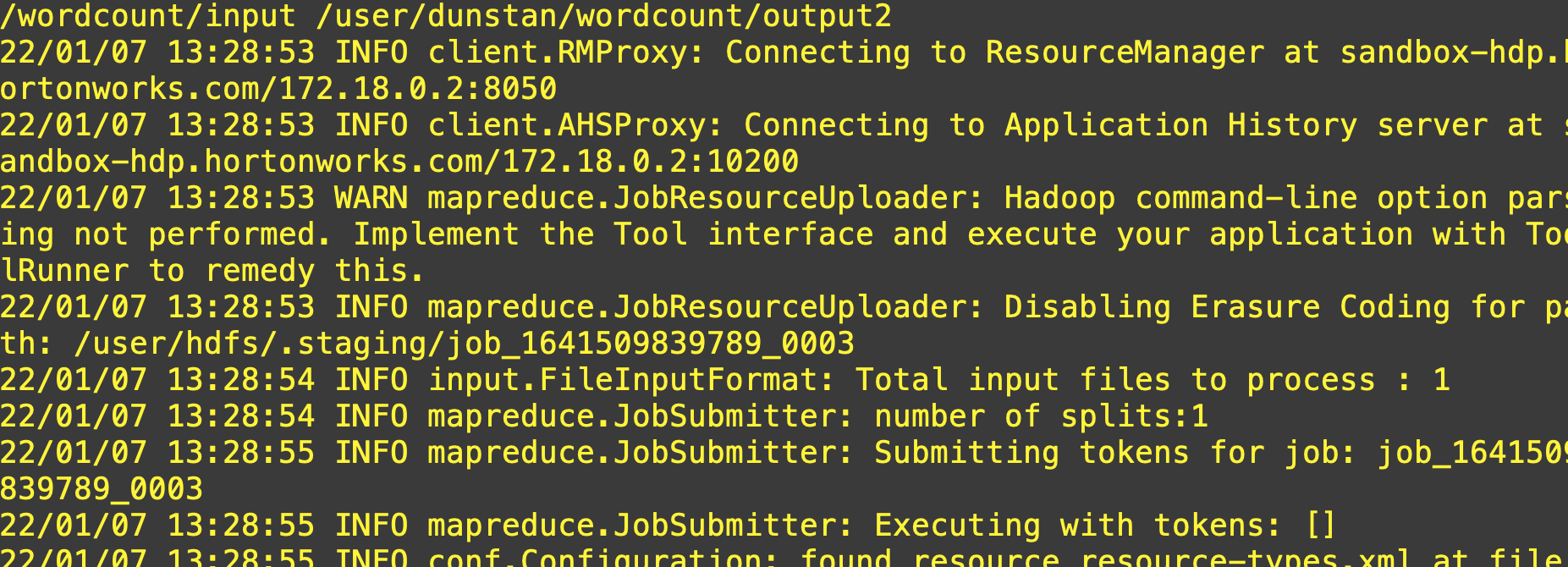
jar cf wc.jar WordCount\*.class

## Run the application

When running the application, we use the hadoop command, specify the jar files, program name (in this WordCount), full path to input folder in the HDFS (e.g., /user/dunstan/wordcount/input/) and full path to output

hadoop jar wc.jar WordCount /user/dunstan/wordcount/input /user/dunstan/wordcount/output

If the application run successfully, you will see output like this:



## Inspect the results

Once your application has finished running, you can inspect the output folder by using this command. First, use ls to check if there is something there. Next, use -cat to view and save the results to sandbox local file system. For example, in order to save the outputs to local sandbox file system as a text file, do it like this:

hdfs dfs -cat /user/dunstan/wordcount/output2/part-r-00000 > wordcount-results.txt