Guide: Hadoop Cluster on AWS

Ignacio M. Llorente, Simon Warchol

v3.0 - 14 March 2021

Abstract

This is a screenshot document of how to run EMR Hadoop cluster and run MapReduce jobs on AWS environment.

Requirements

- **First you should have followed the Guide "First Access to AWS"**. It is assumed you already have an AWS account and a key pair, and you are familiar with the AWS EC2 environment.
- We strongly recommend cluster instances with at least 4 vCPUs (**m4.xlarge)** to be able to evaluate parallel implementation within each node. m4 instances are optimized for Amazon EBS, the block storage for ec2 instances, and are best when distributing large files amongst your cluster. They are the preferred instance type for Hadoop clusters.
- The files needed to do the exercises are available for download from **Canvas**.

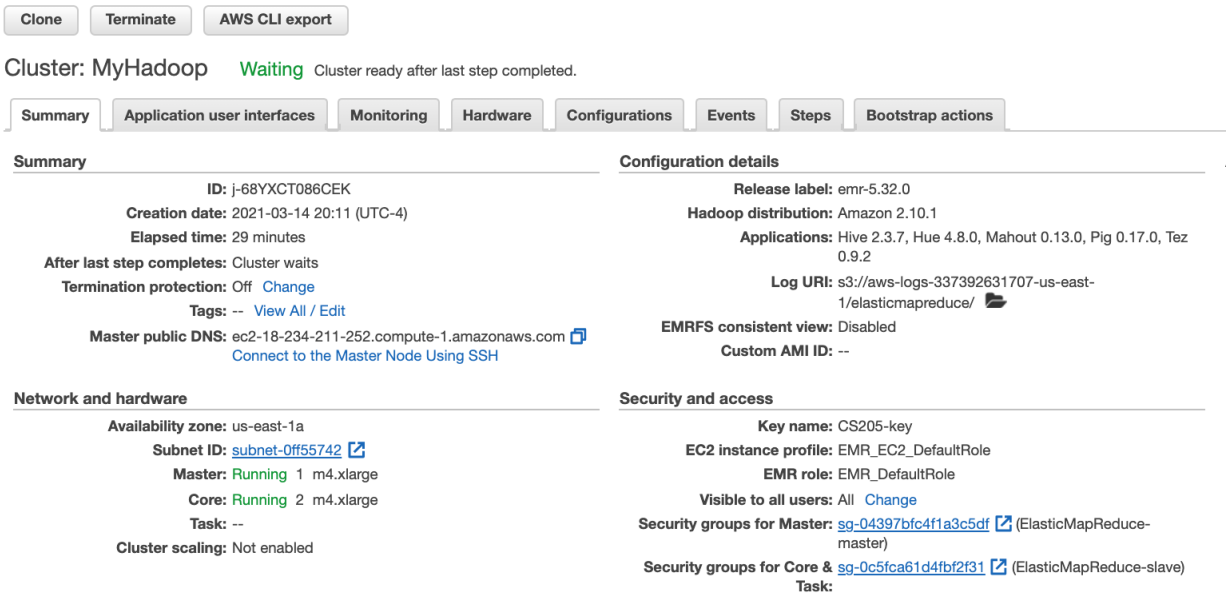
Acknowledgments

The author is grateful for constructive comments and suggestions from David Sondak, Charles Liu, Matthew Holman, Keshavamurthy Indireshkumar, Kar Tong Tan, Zudi Lin, Nick Stern, Dylan Randle, Hayoun Oh, Zhiying Xu and Zijie Zhao.

1. Launch Hadoop EMR cluster

- Go to the EMR dashboard (https://console.aws.amazon.com/elasticmapreduce/home) and click "Create cluster". We recommend the following configuration

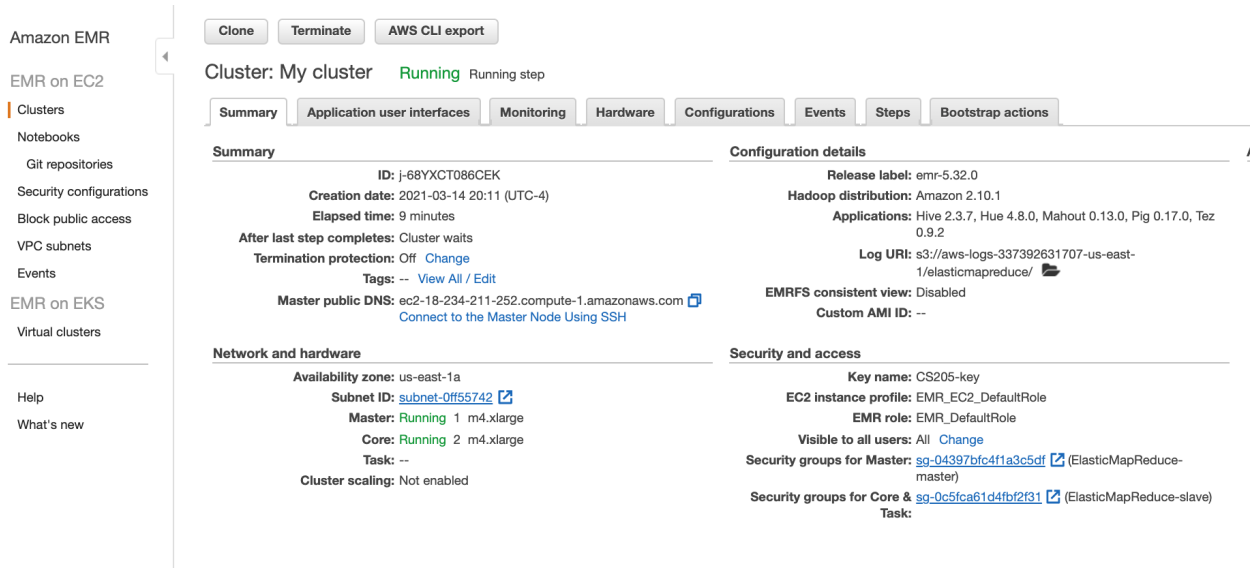
  - ClusterName: MyHadoop
  - Launch mode "Cluster"
  - Release: 5.32.0
  - Applications: Core Hadoop
  - Instance type: m4.xlarge
  - Number of Instances: 3
  - Key pair: course-key (or any other key you want to use, see Guide "First Access to AWS")



- Click on "Create Cluster"

- Wait for the cluster to be ready. This may take 5-10 minutes. This is a good opportunity to briefly call a loved one, take out the trash, or ask other questions of the esteemed TF leading this lab. The cluster is ready when its state is "Waiting" and the Master and Core under the **Networks and**

hardware section are both in “Running” state



2. Login to the cluster

This section is for illustrative purposes to show how EMR is a Hadoop cluster automatically installed and configured on-demand on EC2 instances. You can skip this section to complete this guide because, as it is described in Section 3, you can submit basic MapReduce jobs from the AWS web interface

- You can SSH into master with the Master public DNS address listed above. For instance

```
ssh -i your/course/ssh-key.pem hadoop@your-master-public-dns
```

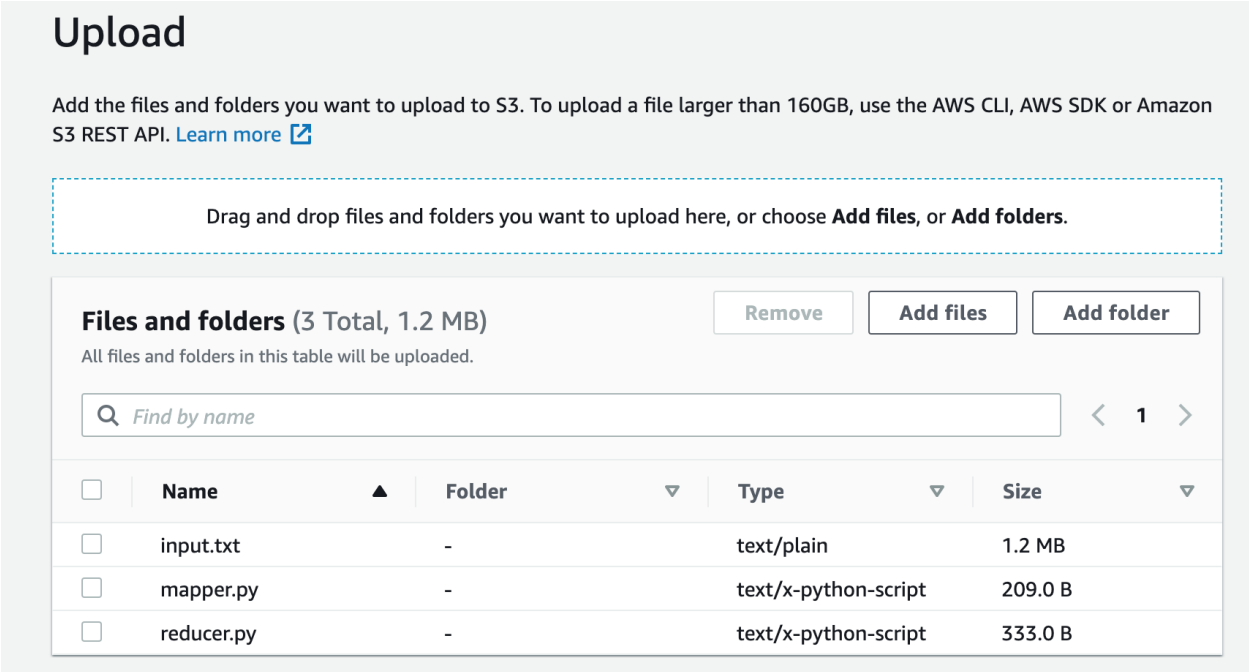
- If SSH fails, you may need to open port 22 on the master security group. Click the link to the security group next to **Security groups for Master**, click the Master security group and add an SSH rule with port 22 and source 0.0.0.0/0.
- SSH should now work if it didn’t already.

```
| => ssh -i ~/.ssh/CS205-key.pem hadoop@ec2-18-234-211-252.compute-1.amazonaws.com
Last login: Mon Mar 15 00:21:17 2021

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
38 package(s) needed for security, out of 76 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R::::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-17-242 ~]$
```

3. Submit a MapReduce job

Hadoop Streaming is a utility that comes with Hadoop that enables you to develop MapReduce executables in languages other than Java. A Streaming application reads input from standard input and then runs a script or executable (called a mapper) against each input. The result from each of the inputs is saved locally on a Hadoop Distributed File System (HDFS) partition. After all the input is processed by the mapper, a second script or executable (called a reducer) processes the mapper results. The results from the reducer are sent to standard output.

- Upload `mapper`, `reducer` and `input` files to a new S3 bucket via the AWS interface. Create a S3 bucket, I named it `emr-example-python`. Remember this name should be unique. Moreover, because of Hadoop requirements, S3 bucket names used with Amazon EMR have the following constraints: must contain only lowercase letters, numbers, periods (.), and hyphens (-); and cannot end in numbers. This is a great opportunity to express your creativity!
    - Both mapper and reducer assume that lines are fed in through sys.stdin. Good sources of available text to play with are in Project Gutenberg.

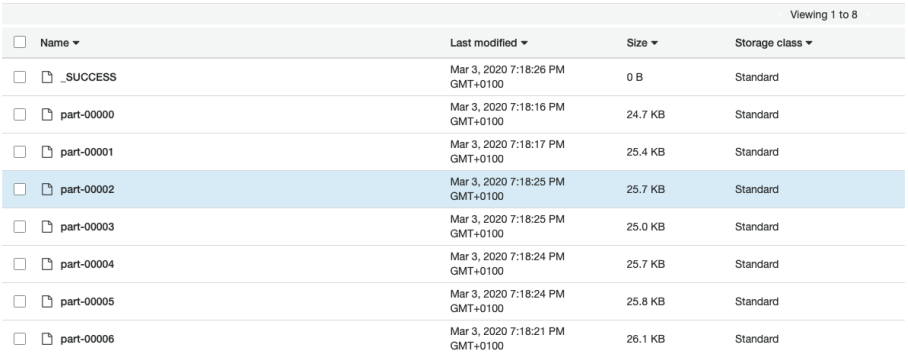


- Go to the Hadoop cluster dashboard's **Steps** tab and click on "Add Step" with the following configuration
  - Step type: Streaming program
  - Name: MyHadoopJob
  - Mapper: Complete path to uploaded mapper
  - Reducer: Complete path to uploaded reducer
  - Input: Complete path to uploaded input
  - Output: Complete path to new folder to be created with the output (**it should not exist**)

- Wait for the "step" to be "completed"
- After "completed" you can check the execution time in the `controller` log file


```
INFO total process run time: 78 seconds
2021-03-15T00:46:01.995Z INFO Step created jobs: job_1615767477849_0001
2021-03-15T00:46:01.996Z INFO Step succeeded with exitCode 0 and took 78 seconds
```

- If the job is not successfully "completed", you can check the logging files for further information
- Finally, check the results in the bucket, Hadoop creates one output file for each executed reducer task



Viewing 1 to 8

Name	Last modified	Size	Storage class
_SUCCESS	Mar 3, 2020 7:18:26 PM GMT+0100	0 B	Standard
part-00000	Mar 3, 2020 7:18:16 PM GMT+0100	24.7 KB	Standard
part-00001	Mar 3, 2020 7:18:17 PM GMT+0100	25.4 KB	Standard
part-00002	Mar 3, 2020 7:18:25 PM GMT+0100	25.7 KB	Standard
part-00003	Mar 3, 2020 7:18:25 PM GMT+0100	25.0 KB	Standard
part-00004	Mar 3, 2020 7:18:24 PM GMT+0100	25.7 KB	Standard
part-00005	Mar 3, 2020 7:18:24 PM GMT+0100	25.8 KB	Standard
part-00006	Mar 3, 2020 7:18:21 PM GMT+0100	26.1 KB	Standard

Terminate the cluster when you are sure you are done for the day to avoid incurring charges