Guide: Hadoop Cluster on AWS

Ignacio M. Llorente

v2.0 - 2 March 2020

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.
- The files needed to do the exercises are available for download from **Canvas**.

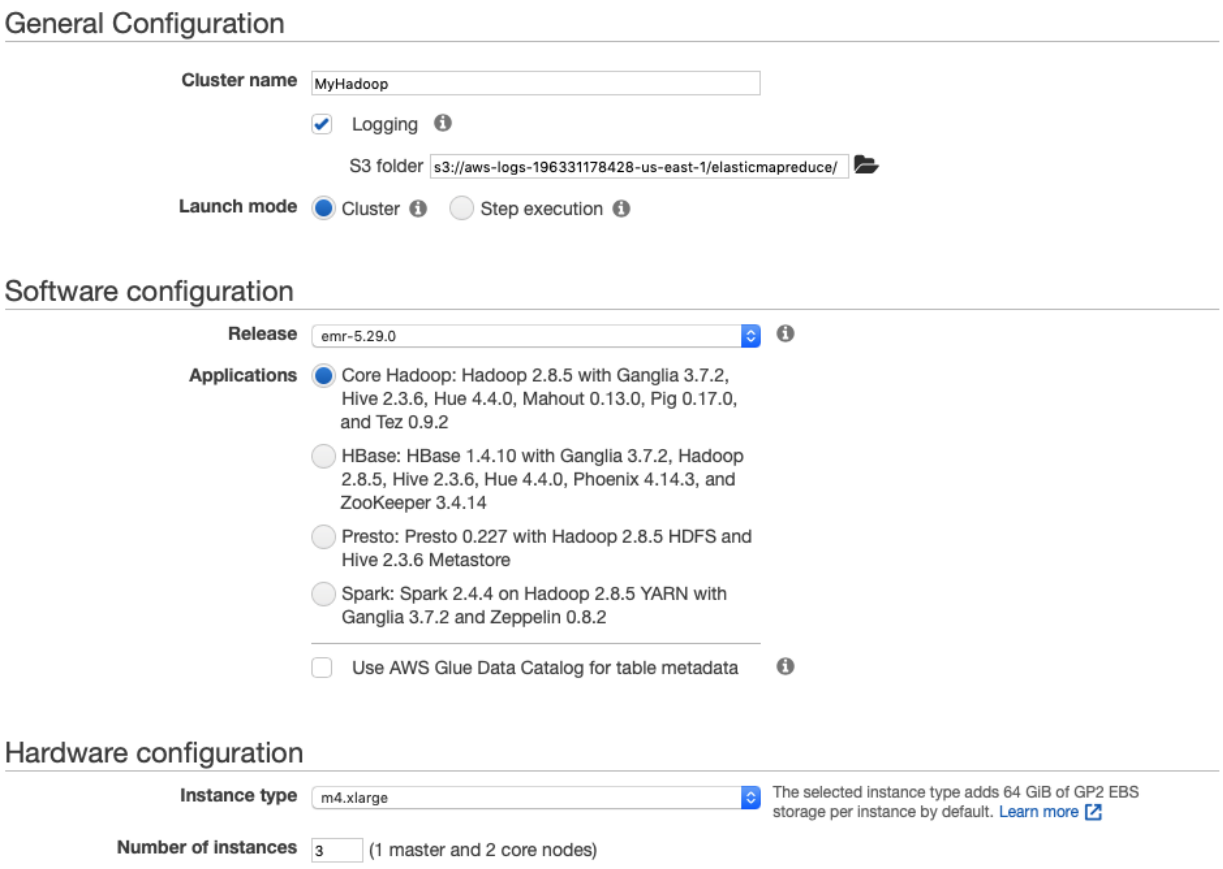
Acknowledgments

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.

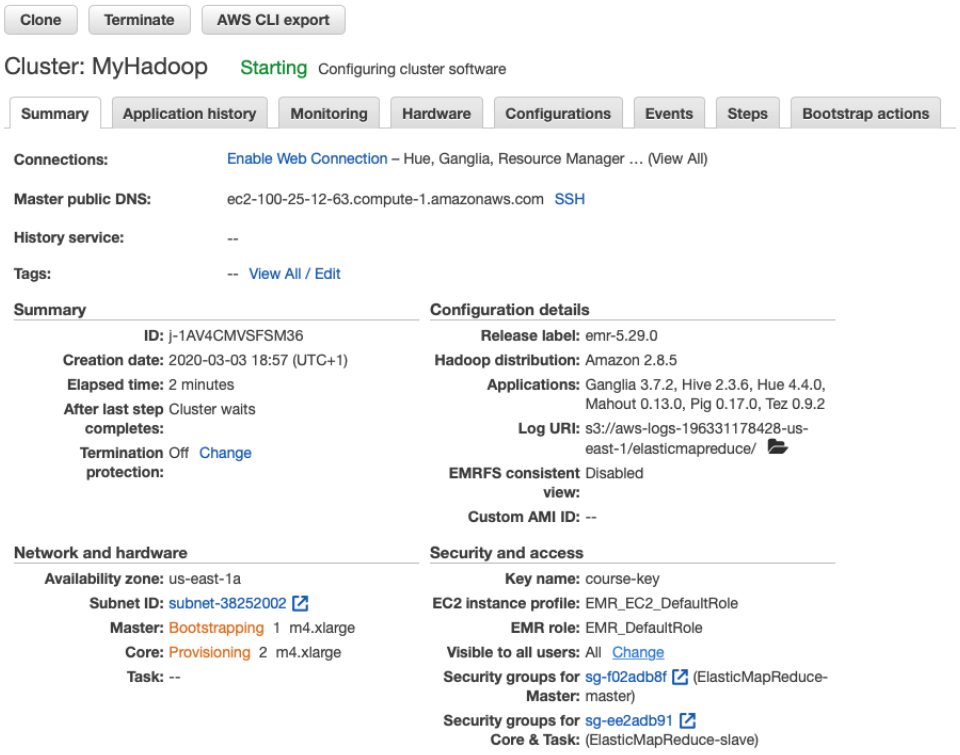
1. Launch Hadoop EMR cluster

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

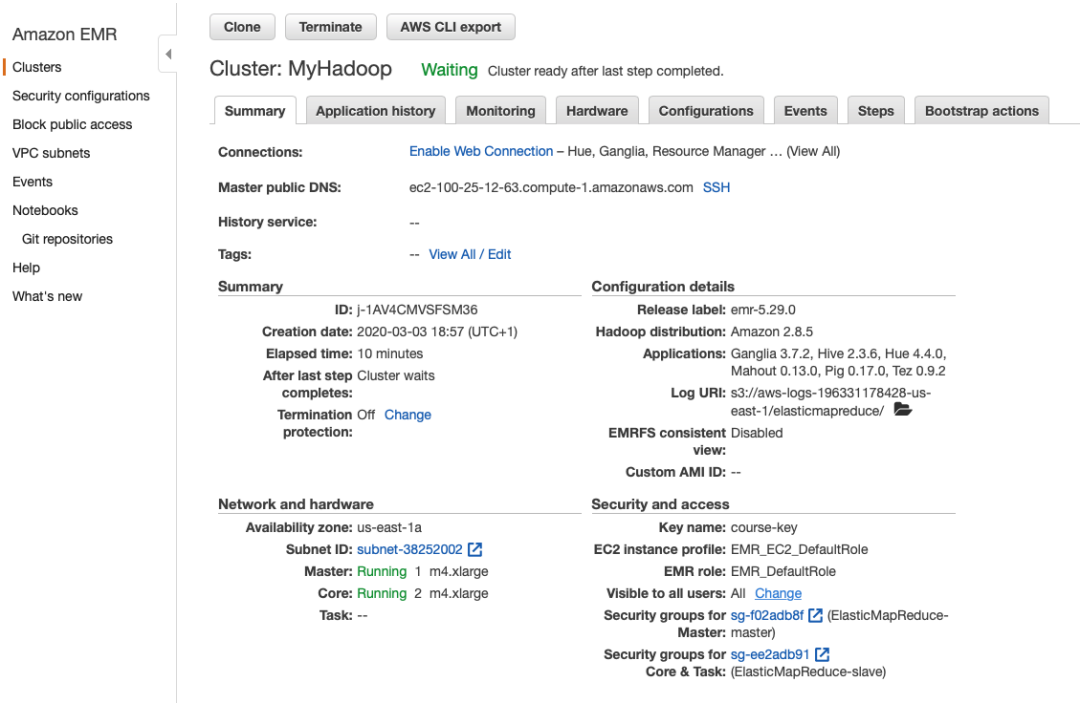
  - ClusterName: MyHadoop
  - Launch mode "Cluster"
  - Release: 5.29.0
  - Applications: Core Hadoop
  - Instance type: m4.xlarge
  - Number of Instances: 3
  - Key pair: course-key (or any other key you want to use, see Guide "First Access to AWS")



- Click on “Create Cluster”



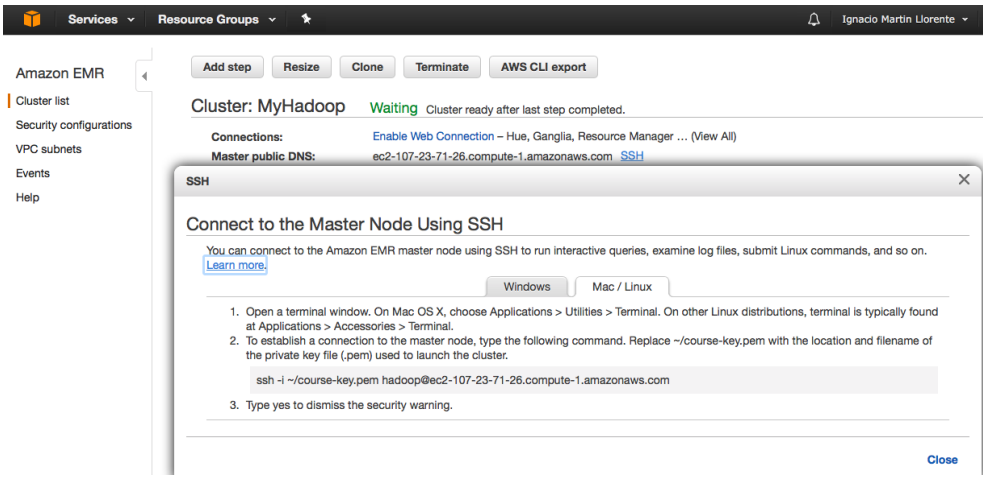
- Wait for the cluster to be ready. The cluster is ready when its state is “Waiting” and the Master and Core under the **Networks and hardware** section are both in “Running” state



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

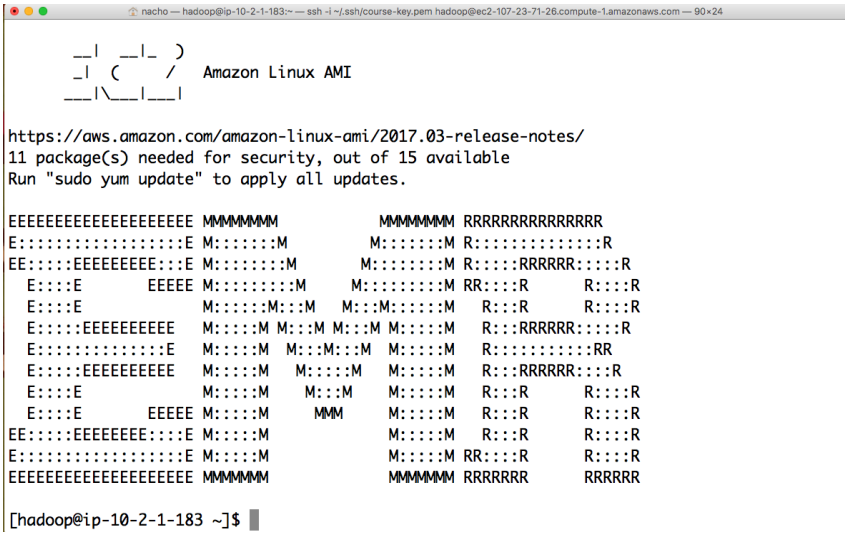
- Write down the "Master public DNS" and click on the SSH link next to it. The SSH link gives you the commands you might use to login to your cluster



- Most likely you will need to open port 22 to be able to login. Make sure that the security groups (firewalls) of the EMR cluster master node opens the port 22 to the outside world (see Guide "First Access to AWS"). 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 to the machine using the private key

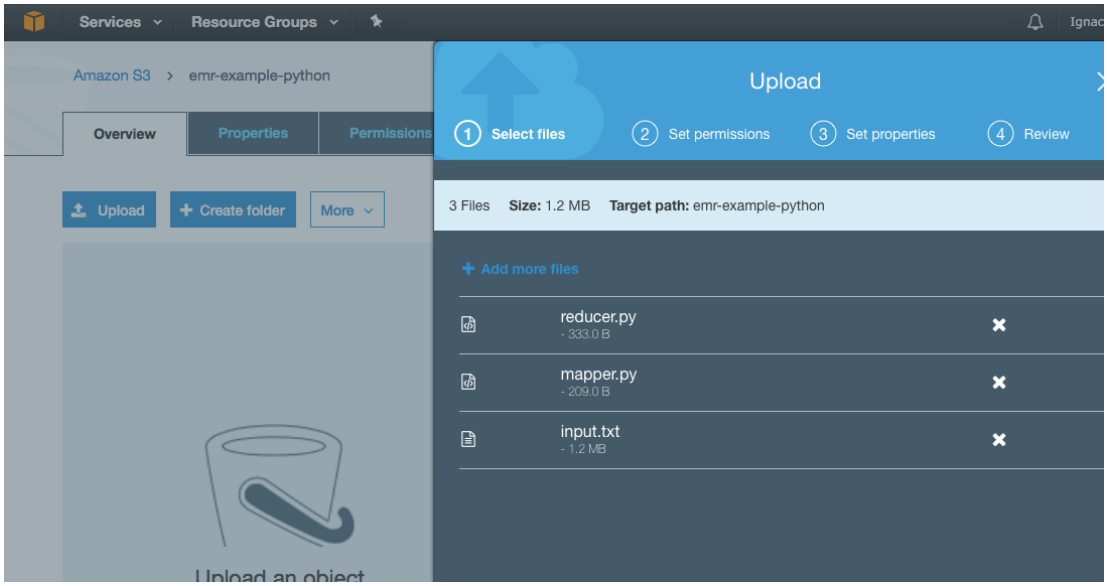
```
$ ssh -i your/course/ssh-key.pem hadoop@your-master-public-dns
```



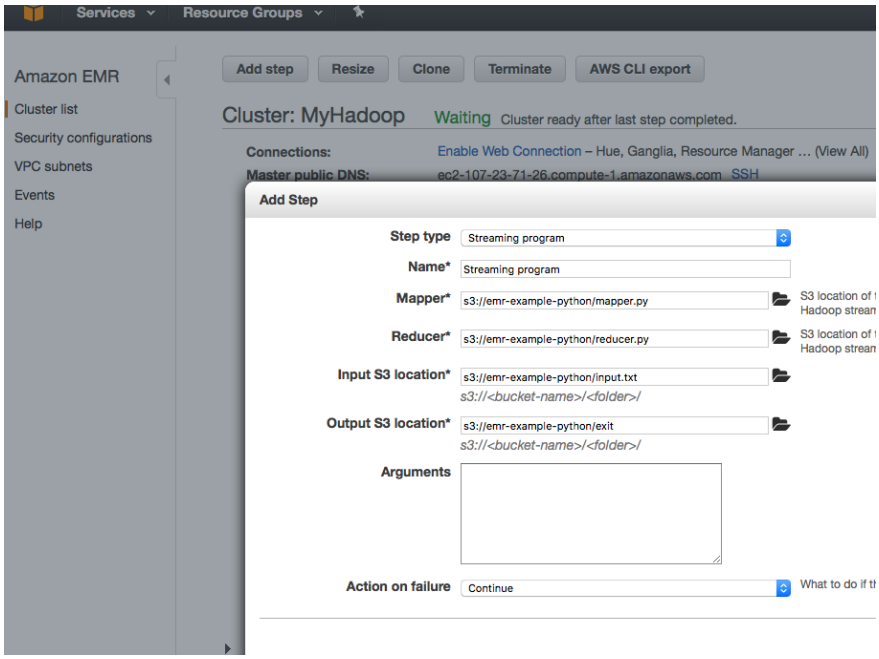
3. Submit a MapReduce job

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. 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
    - 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.



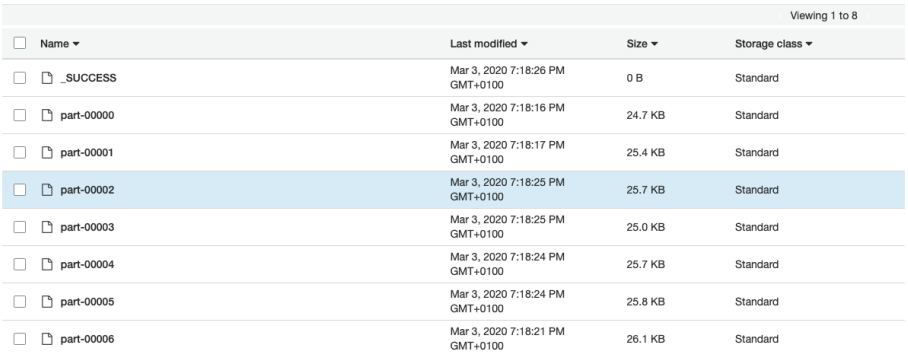
- Go to the Hadoop cluster dashboard's **Steps** tab and click on "Add Step" with the following configuration
    - Step type: Streaming program
    - Name: MyHadoopJob
    - Mapper: Complete path to uploaded mapper
    - Reducer: Complete path to uploaded reducer
    - Input: Complete path to uploaded 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: 72 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



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