MapReduce on Amazon Web Service's Elastic MapReduce

Introduction

In this guide, we'll be going over how to use Amazon Web Service's Elastic MapReduce to run MapReduce jobs. Elastic MapReduce (a.k.a EMR) is Amazon's solution to data analytics based on Hadoop. Users can create clusters running Hadoop and run distributed MapReduce/Spark programs. This guide covers how to set up your own Hadoop cluster using AWS EMR, how to interact with the Hadoop Cluster, and how to run MapReduce Jobs.

Contents

1 Setting up AWS

1.1 Signing up for AWS

In order to use EMR, you need an AWS account to access AWS's services. If don't have an AWS account already, you can sign up at https://portal.aws.amazon.com/billing/signup . Note: You will need to a credit card to sign up for an AWS account.

1.2 Getting Educational Credit

Now that you have an AWS account, you need credit so you can use AWS services without being charged on out credit cards. Currently, Amazon will give $100 of free credit which should be more than enough to run a Hadoop Cluster for 100+ hours. To do this,

1. Go to https://aws.amazon.com/education/awseducate/
2. Click "Join AWS Educate" to start registering
3. Fill out the form accordingly. Ensure you use your school .edu email and leave the promo code field empty
4. You will notice there are two options. Select the first one to use the AWS account you created.
5. Enter your AWS Account ID. You can find this by logging into AWS, selecting your name at the top of the page, and clicking "My Account". Under the account setting section on your account page, your account id should be listed.
6. Click "Next"
7. Read the Terms and Conditions, select "I Agree", and click submit.
8. After AWS reviews your application, you should get an email with a credit code worth $100 dollars.
9. To add the credit to your account, follow the directions in the email.

Sign up time: AWS has review your educational credit application before you receive any credit. This could take from a few minutes to a few hours.

Free educational account with no credit card: It is possible to sign up for a free AWS educate account but it is limited and you can not keep running anything when the limits have been reached. It is recommended to sign up with a regular AWS account and apply for educational credit so you are not bound by any limits. The $100 of AWS credit you receive should be more than enough to last you the duration of your project around 100 hours of cluster uptime).

2 Cluster Management

2.1 Creating a Security Key-Pair

When you create your own Hadoop cluster, it will open to the internet and you will able to access it from anywhere. You should create a security key pair so when you do create any clusters, you can configure it to only accept incoming connections from those who have the key. To create a key-pair, do the following:

1. Go to https://us-west-2.console.aws.amazon.com/ec2/home

2. On the left hand panel, under "NETWORK & SECURITY", click "Key Pairs"
3. Start creating a key pair by clicking "Create Key Pair"
4. Name it the new key pair and click "Create".
5. The key pair will be created and your browser should prompt you to download the key (.pem file). Save this in a safe space, you will need this for when you SSH or SCP to the cluster.

Using AWS without keys: It is possible to configure clusters to not need a key to access. This is not recommended.

2.2 Starting your cluster

In order to start a new cluster, do the following:

1. Go to AWS EMR by going to Services, and under the "Analytics" section, click "EMR".
2. On the EMR home page, click "Create Cluster".
3. Within the cluster configuration page, give your cluster a name.
4. Under the Software Configuration Section, make sure Core Hadoop is selected under Applications
5. Under the Hardware Configuration Section, make sure "m3.xlarge" in the dropdown. This is the default size. This should be enough for typical usage.
6. Under Security and access, if you made a security key-pair, select the key-pair you made in the EC2 key pair dropdown.
7. Click "Create cluster".
8. Now to connect to remotely your cluster follow the next section to add security rules for you cluster.

2.3 Setting Security Rules

By default, you won't be able to connect to your cluster. In order to connect to the cluster you have to manually add security rules in order for you to connect via SSH, TCP, and ICMP protocols over the internet. To do this, do the following:

1. On the Amazon EMR dashboard, navigate to your cluster by clicking its name.
2. Within the summary tab for your cluster, under the "Security and Access" section, click the link following "Security groups for Master"
3. Select the entry for ElasticMapReduce-master.
4. Navigate to the "Inbound" tab below and click "Edit"
5. Add a rule for TCP by clicking "Add Rule", selecting "All TCP" in the type dropdown, and selecting "Anywhere" in the source dropdown.
6. Add a rule for ICMP by clicking "Add Rule", selecting "All ICMP - IPv4" in the type dropdown, and selecting "Anywhere" in the source dropdown.
7. Finally, add a final rule for SSH by clicking "Add Rule", selecting "SSH" in the type dropdown, and selecting "Anywhere" in the source dropdown.
8. Save the rules you just created by clicking "Save".

Startup Time: When starting a cluster, it typically takes some time for the hardware resources to be allocated and functionalities set up for your cluster. It could possibly take to up to 15-20 minutes for your cluster to start.

2.4 Terminating a Cluster

To terminate/shut down a cluster you started, do the following:

1. Go the EMR Home and go to Clusters
2. Select the cluster you want to terminate and click "Terminate"

You will lose everything: Note that when you terminate your cluster, you will lose everything you have stored on the cluster. Make sure to copy anything that you need back to your machine or use Amazon S3 buckets to persist data between cluster sessions. (We'll go over S3 bucket a little later in this document)

Shut Down Time: It could possibly take a few minutes to shut down a cluster.

2.5 Restarting a Cluster

Since AWS allocates resources for your cluster on demand, there no such thing as "restarting" a cluster. Once it's gone, it's gone. What you can do is clone a previous cluster to start a new cluster with the same cluster configuration. To do this,

1. Go to the EMR home and go to clusters.
2. Select the cluster you want to clone
3. Click "Clone"

2.6 Tips for Cluster Usage

2.6.1 Avoiding Unnecessary Charges

To avoid unnecessary charges, make sure you terminate your cluster whenever you are not going to use your cluster. It would be fine to leave the cluster running if you wont be using the cluster for 30 minutes or so.

2.6.2 Checking Cluster Usage

You can check your billing forecast on AWS by clicking your name at the top and clicking "My billing dashboard". It should bring you to your billing dashboard, showing the expected cost based on your current cluster usage.

2.6.3 Sharing the Cluster

It may be more efficient for you to share a single cluster if you are perhaps working a group. In order to do this you can just share the secret key with anyone else that you want to share the cluster with. You could also start the cluster with no security. But this is highly not recommended.

2.6.4 Cleaning up after use

When you are done using AWS EMR, make sure to terminate any running clusters and delete any S3 data you may have stored. This will ensure you won't be charged for any AWS usage in the future.

3 Running MapReduce

In order to run MapReduce you will have do three things. Place your jar somewhere the cluster can access it. Place the input data somewhere the cluster could access. Run MapReduce.

3.1 Connecting to the Cluster

You run commands on the EMR cluster, you can connect the the EMR cluster with SSH. To connect to your cluster you will use connect as Hadoop at master public DNS. You can find the master public DNS of your cluster under the summary section for your cluster.

```
$ ssh -i <key location> <remote>
```

example:

```
$ ssh -i mykey.pem hadoop@ec2-54-245-34-159.us-west-2.compute.amazonaws.com
```


Make sure you added security rules: If you can not connect to your cluster, ensure that you have added the proper security rules to allow you to connect. You can find out how to do this in section 2.3.

3.2 Ways to run MapReduce

In order to run MapReduce, you have to have files accessible to the cluster. You have two options when it comes to running MapReduce:

1. Store your files directly onto the cluster using scp and use hdfs commands to move files to hdfs. You will have to run thee Hadoop program by executing a command on the cluster via SSH.
2. Upload your files in Amazon S3 buckets and have Hadoop pull the data from there. You will to run a MapReduce Job by using Amazon EMR Steps.

3.3 Running MapReduce Manually using SSH, SCP, and Hadoop Commands

The following section the traditional way of running MapReduce, covering how you would typically interact with a Hadoop Cluster.

3.3.1 Manually Storing Files on Hadoop Cluster

In order to run MapReduce you'll need to put your runnable jar and input files onto the cluster. You can transfer files to your cluster by using the scp command. Typical usage is as follows:

```
$ scp <source> <destination>
```

Since you need a key to connect to your cluster you will need to pass the key to the command using the -i flag:

```
$ scp -i <key location> <source> <destination>
```

Example:

```
$ scp -i mykey.pem WordCount.jar
  hadoop@ec2-54-245-34-159.us-west-2.compute.amazonaws.com:~
```

Note: Notice the usage of the colon within the destination. The path after the colon specifies the file path the file should be placed on the destination system. In our example we simply used the tilde character, $\sim$, which represents the home directory.

You can see more about how to use scp here: https://haydenjames.io/linux-securely-copy-files-using-scp/

3.3.2 Getting Files Input Files into HDFS

Now for Hadoop to use your input, the input files must be on HDFS. By using scp you were able to get your files onto the machine, but not onto HDFS. Now you need to get it from the machine to HDFS. To get it on HDFS, use HDFS commands to move files to and from HDFS while SSH'd in the cluster.

You can move a file to hdfs by using the -put command:

```
$ hdfs dfs -put <source> <dest>
```

You can list your files on hdfs by using the -ls command:

```
$ hdfs dfs -ls
```

Example:

```
$ ls
input.txt
$ hdfs dfs -put input.txt
$ hdfs dfs -ls
Found 1 item
-rw-r--r--   1 hadoop hadoop          0 2018-11-20 00:17 input.txt
```

You can read more on HDFS commands here:
https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html

3.3.3 Manually Running MapReduce Using Hadoop Jar

To run the MapReduce program, while SSH'd into the cluster, invoke Hadoop to run MapReduce by using the Hadoop jar command:

```
$ hadoop jar <jar location> <Main Class> <input> <output>
```

Example with the runnable jar on the cluster, the input data named input.txt in HDFS, and specifying Hadoop to create a folder named "output" to put the output files in:

```
$ hadoop jar WordCount-0.0.1-SNAPSHOT.jar WordCount input.txt output
```

Output must be Unique: When specifying the output folder, make sure the folder name is unique and unused. If there already exists a folder with the same name, MapReduce will not run successfully as Hadoop won't be able to create a folder with a name that already exists.

You will lose everything: Remember every time you terminate your cluster, everything stored on the cluster will be gone. Make sure to copy back any files you will need.

3.4 Running MapReduce using S3 Buckets and EMR steps

In order to make it easier for users, Amazon provides seamless integration with AWS S3 and the your EMR Hadoop Cluster as a means to run MapReduce without having to SSH into the cluster and run commands to move files around.

3.4.1 Storing files in S3 Buckets

Instead of storing putting data on the cluster you can instead upload our data to an AWS S3 bucket. AWS S3 is AWS's storage solution. You may find it easier and more helpful to store data in the S3 buckets in order to persist any of our data. To do this

1. Go to "Services" at the top of the page and under the "Storage" section, click "S3".
2. Create a new bucket to store files in.
3. Use the web interface to upload your files.

3.4.2 Adding A Step to the Cluster

You can invoke MapReduce without having to SSH into the cluster by utilizing AWS EMR's step interface. To do this do the following:

1. Go to the EMR home.
2. Go to clusters in the sidebar.
3. Click the cluster you are running.
4. Go to the steps tab.
5. Click "Add Step"
6. Select "Custom JAR" within the Step type dropdown.
7. For the location box, navigate to the jar file you stores on S3, or if you stored it on HDFS manually, enter the file location on HDFS.
8. Enter the jar arguments like you would if you did it manually, specifying the java Class to run, the input, and output.

Example:

Note: Notice how files within your S3 bucket are referenced. In this case we're invoking the Word Count Runnable jar passing in arguments, giving it the location of our input file "input.txt" located within our "cs-417" S3 bucket. We then specify Hadoop to place the results in a folder called "output" in our S3 bucket.

Output must be Unique: Even on Amazon S3, when specifying the output folder, make the output folder name unique. If there already exists a folder with the same name, MapReduce will not run successfully as Hadoop won't be able to create a folder with a name that already exists.

3.5 Analyzing MapReduce Output

To view the results of the MapReduce program, check out the resulting files Hadoop created in the output folder you specified.

3.5.1 Checking out Results in Secure Shell

If you ran the MapReduce manually, invoking Hadoop within an SSH session. You can checkout the output by using the hdfs ls and cat commands. For example if you specified Hadoop to store the results in a folder called "output" on HDFS, you can list the files within the output folder using the ls command

```
$ hdfs dfs -ls output
-rw-r--r--   1 hadoop hadoop          0 2018-11-20 00:06 output4/_SUCCESS
-rw-r--r--   1 hadoop hadoop      49831 2018-11-20 00:06 output/part-r-00000
-rw-r--r--   1 hadoop hadoop      49924 2018-11-20 00:06 output/part-r-00001
```

You can then print out the contents of one of the result files by using the cat command

```
$ hdfs dfs -cat output/part-r-00000
```

If you want to copy the results from HDFS back to your cluster, you can use the hdfs get command:

```
$ hdfs dfs -get output/part-r-00000
```

If you want to transfer the results back to your local computer (not the cluster machine), you can use a combination of the scp command like when you transferred files to the cluster, but instead switching the source and destination locations.

3.5.2 Checking out Results Stored on S3

If you specified Hadoop to store the results in your S3 bucket, you can checkout the results by navigating to your S3 buckets and downloading the resulting files.

4 Useful links and Resources

Hadoop Cluster Interaction

- *HDFS Overview and Commands*
  https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html

MapReduce Guides

- *Apache MapReduce Tutorial*
  https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

Last Updated: November 28, 2018