Launching an Amazon EMR cluster in minutes

In this demo, you use advanced options to create an EMR cluster in 10 minutes. With advanced options, you can select a wide range of applications and implement custom security options.

3. Navigate to EMR and create a new cluster.
4. If there is a message that displays The new EMR console is now the default console, choose Switch to the new console. This lab uses the new EMR console.
5. For Name: enter
   `mycluster`
6. For Amazon EMR release, select emr-6.15.0.
7. For Application bundle, select Custom.
8. De-select Pig.
9. Choose Spark.
10. Scroll down to the Cluster configuration section.
11. For Primary, choose m4.large.
12. For Core, choose m4.large.
13. For Task 1 of 1 choose m4.large.
14. In the Networking section, choose **Browse**, then:
- Select Lab VPC.
- Select **Choose**.
15. For Subnet, choose **Browse**, then:
- Select Lab VPC Private Subnet.
- Select **Choose**.
16. Expand Cluster termination and node replacement.
17. For Termination option, choose Manually terminate cluster.
18. Expand Security configuration and EC2 key pair.
19. For Security configuration, choose the emr-cfg file.
20. For Amazon EC2 key pair for SSH to the cluster, choose EMRKey.
21. Expand Identity and Access Management (IAM) roles section.
22. For Service role, choose EMRDefaultRole.
23. For Instance profile, choose EMR_EC2_DefaultRole.
24. Choose **Create cluster**.

Connect to an EMR cluster and perform Scala commands using the Spark shell

In this demo, you connect to your EMR cluster and use Spark to perform batch analytics on stock market data. Then, you find the maximum closing price for a selection of companies in the sample data and analyze the results.

25. In the EMR cluster you created in Interactive Demo 1, add a rule in the security group for the primary node with the settings outlined below. This grants SSH access from the CommandHost.

- Type: *SSH*
- Source: `10.0.0.0/16`

To add the security group, you may follow the below steps:

26. Click on the Cluster ID created in previous demo.
27. Scroll down to Network and security section.
28. Click on EC2 security groups (firewall). Click on the security group link starting with sg- mentioned below EMR-managed security group . This will open a new tab to edit the security group.
29. Click on Edit inbound rules and then on Add rule.
30. Near the bottom of the screen a new entry will appear, click on Custom TCP dropdown menu and choose SSH.
31. In the same line, click on a textbox and type `10.0.0.0/16`.
32. Choose Save rules.
33. To open the Session Manager terminal, paste the CommandHostSessionManagementUrl value fhttps://us-west-2.console.aws.amazon.com/systems-manager/session-manager/i-0a52c2a16fe43304d?region=us-west-2

```
# Get EMR cluster ID and export to the Environment.
export ID=$(aws emr list-clusters | jq '.Clusters[0].Id' | tr -d "")

# Use the ID to get the PublicDNS name of the EMR cluster
# and export to the Environment.
export HOST=$(aws emr describe-cluster --cluster-id $ID | jq '.Cluster.MasterPublicDnsName' | tr -d "")

# SSH to the EMR cluster
ssh -i ~/EMRKey.pem hadoop@$HOST
```

To export the Amazon Simple Storage Service (Amazon S3) bucket name as a `bucket` environment variable and open the Spark shell in the EMR leader node terminal, paste the following command:

```
export bucket=$(aws s3api list-buckets --query "Buckets[].Name" | grep databucket | tr -d ' ' | tr -d "" | tr -d ',')

echo $bucket

Spark-shell
```

```
' | tr -d '"' | tr -d ',')
[hadoop@ip-10-0-20-250 ~]$ echo $bucket
labstack-502883f7-b4ce-49fe-bc70-c4f0be-databucket-r4cic4osqkk3
[hadoop@ip-10-0-20-250 ~]$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/05/19 10:27:50 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Spark context Web UI available at http://ip-10-0-20-250.us-west-2.compute.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1747648673142_0001).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.4.1-amzn-2
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_442)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

36. To create a variable for the Amazon S3 location path, create a DataFrame, describe the schema of the loaded data, view the table content, and view the max close stock price, paste the following command:

```
val bucket = System.getenv("bucket")

val s3_loc = "s3://"+bucket+"/data/stock_prices.csv"



val df = spark.read.option("header","true").option("inferSchema","true").csv(s3_loc)

df.printSchema()

df.show()

  df.groupBy("Ticker").agg(max("Close")).sort("Ticker").show()
```

Client-side encryption with EMRFS

In this demo, you create a client-side encrypted file for your batch analytics jobs using EMR File System (EMRFS) by creating, encrypting, and decrypting a file.

38. To open the Session Manager terminal, paste the CommandHostSessionManagementUrl value from the left of these instructions to a new tab in your browser.
39. To connect to your EMR leader node, paste the following commands in the Session Manager terminal:

```
# Get EMR cluster ID and export to the Environment.

export ID=$(aws emr list-clusters | jq  '.Clusters[0].Id' | tr -d '"')



# Use the ID to get the PublicDNS name of the EMR cluster

# and export to the Environment.

export HOST=$(aws emr describe-cluster --cluster-id  $ID | jq '.Cluster.MasterPublicDnsName'  | tr -d '"')



# SSH to the EMR cluster

    ssh -i ~/EMRKey.pem hadoop@ $HOST
```

- When prompted to allow a first connection to this remote server, type `yes` and press ENTER.

40. To export the Amazon S3 bucket name as a bucket environment variable, paste the following command:

```
export bucket=$(aws s3api list-buckets --query  "Buckets[].Name" | grep databucket |  tr -d ' ' | tr -d '"' | tr -d ',')

    echo $bucket
```

Next, you will write an encrypted object to Amazon S3 from your EMR cluster.

41. Paste the following command to create a text file that contains the sentence: This is a practice lab!:

```
echo 'This is a practice lab!' > outputFile.txt
```

- Paste the following command to write this file to your Amazon S3 bucket using EMRFS:

```
hadoop fs -put outputFile.txt s3://${bucket}/
```

This command writes the file as a client-side encrypted object at rest using EMRFS to the Amazon S3 bucket that you created as part of the lab.

Now that you have encrypted an object, follow the next steps to see what the encryption did to the object in Amazon S3 and decrypt it using EMRFS.

42. Paste the following command to download the encrypted object directly from your Amazon S3 bucket into a encryptedOutputFile.txt file:

```
aws s3 cp s3://${bucket}/outputFile.txt encryptedOutputFile.txt
```

43. Paste the following command to view your encrypted object:

```
cat encryptedOutputFile.txt
```

44. Paste the following command to decrypt and read the object from Amazon S3 into the EMR cluster using EMRFS:

```
hadoop fs -cat s3://${bucket}/outputFile.txt
```