## Building Batch Data Analytics Solutions on AWS:

## Lab 4 - Interactive Demos

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

# Lab overview

You use this lab environment to access the AWS Management Console and interact with the AWS services discussed in class. The environment is available to you throughout the day, and it is then reset for the following day's class.

Follow along with your instructor as they complete these interactive demos. These instructions will guide you through the process, but follow the instructor closely as some of the steps are left intentionally open-ended.

### AWS SERVICES NOT USED IN THIS LAB

AWS service capabilities used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

### ICON KEY

Various icons are used throughout this lab to call attention to certain aspects of the guide. The following list explains the purpose for each one:

- Specifies the command you must run.
- Verify the output of a command or edited file.
- Specifies important hints, tips, guidance, or advice.
- Calls attention to information of special interest or importance. Failure to read the note does not result in physical harm to the equipment or data, but it could result in the need to repeat certain steps.

# Start lab

1. To launch the lab, at the top of the page, choose Start lab.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

2. To open the lab, choose Open Console.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING: Do not change the Region unless instructed.**

### COMMON SIGN-IN ERRORS

**Error: You must first sign out**


If you see the message, **You must first log out before logging into a different AWS account:**

- Choose the **click here** link.
- Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
- Choose Open Console again.

### Error: Choosing Start Lab has no effect

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

- Add the lab domain name to your pop-up or script blocker's allow list or turn it off.
- Refresh the page and try again.

# Interactive Demo 1: 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-6.3.0**.
7. For **Application bundle**, select **Custom**.
8. In the **Customize your application bundle section**:
   - Remove **Pig**.
   - Add **Spark**.
9. In the **Instance groups** section for **Primary** choose **m4.large**.
10. For the **Core** instnace, choose **m4.large**.
11. For **Task 1 of 1** choose the **m4.large** instance type.
12. In the **Networking** section for **Virtual private (VPC)**, choose **Lab VPC**.
13. For **Subnet**, choose the **Lab VPC Private Subnet**.
14. For **Cluster termination**, choose **Manually terminate cluster**.
15. In the **Security configuration and EC2 key pair** section, for **Security configuration**, choose the value of **emrSecurityConfig** located to the left of these insructions.
16. For **Amazon EC2 key pair for SSH to the cluster**, choose **EMRKey**.

17. For **Amazon EMR service role**, choose **Choose an existing service role**.
18. For **Service role**, select *EMRDefaultRole*.
19. For **EC2 instance profile for Amazon EMR**, choose **EMR_EC2_DefaultRole**.
20. Choose **Create cluster**.

*Congratulations, you have successfully created an EMR cluster! This will help with batch processing tasks in the next demo.*

You can now log off the interactive demo session in the AWS Management Console, but ***do not end the lab***. You resume the next demo from here.

# Interactive Demo 2: 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.

21. 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:

22. Click on the **Cluster ID** created in previous demo.
23. Scroll down to **Network and security** section.
24. 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.
25. Click on **Edit inbound rules** and then on **Add rule**.
26. Near the bottom of the screen a new entry will appear, click on **Custom TCP** dropdown menu and choose **SSH**.
27. In the same line, click on a textbox and type

```
10.0.0.0/16
```

.

28. Choose **Save rules**.
29. To open the Session Manager terminal, paste the **CommandHostSessionManagementUrl** value from the left of these instructions to a new tab in your browser.
30. 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.

A message similar to the following indicates a successful connection to the leader node:

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

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
```

31. 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
```

A shell with `scala>` appears:

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1-amzn-0
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_282)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

32. 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()
```

Take a look at the resulting analysis. You will see a sample table from the stock data and a table showing the maxium close price of several companies.

33. To exit the Spark terminal, paste the following command:

```
sys.exit
```

Congratulations! You have successfully performed a batch analytics job.

You can now log off the interactive demo session in the AWS Management Console, but ***do not end the lab***. You resume the next demo from here.

# Interactive Demo 3: 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.

34. To open the Session Manager terminal, paste the **CommandHostSessionManagementUrl** value from the left of these instructions to a new tab in your browser.
35. 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.

36. 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.

37. 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.

38. 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
```

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

```
cat encryptedOutputFile.txt
```

40. 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
```

Do you recognize the text you wrote earlier?

You can read more about EMRFS encryption here.

*Congratulations, you have successfully explored client-side encryption in EMR using EMRFS.*

# End lab

Follow these steps to close the console and end your lab.

41. Return to the **AWS Management Console**.
42. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
43. Choose End lab and then confirm that you want to end your lab.