Analyze Big Data with Hadoop

**SPL-166 - Version 1.0.23**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

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

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

In this lab, you deploy a fully functional Hadoop cluster, ready to analyze log data in just a few minutes. You start by launching an Amazon EMR cluster and then use a HiveQL script to process sample log data stored in an Amazon Simple Storage Service (Amazon S3) bucket. **HiveQL** is a SQL-like scripting language for data warehousing and analysis. You can then use a similar setup to analyze your own log files.

This lab is based on the [Analyze Big Data with Hadoop](https://aws.amazon.com/getting-started/projects/analyze-big-data/) project.

OBJECTIVES

By the end of this lab, you will be able to do the following:

* Launch a fully functional Hadoop cluster using Amazon EMR.
* Define the schema and create a table for sample log data stored in Amazon S3.
* Analyze the data using a *HiveQL* script and write the results back to Amazon S3.
* Download and view the results on your computer.
* Connect to the Hive CLI and run *HiveQL* query script to view the results.

TECHNICAL KNOWLEDGE PREREQUISITES

Familiarity with Hadoop is recommended but not required for this lab. Basic familiarity with Amazon S3 and Amazon EC2 key pairs is also recommended but not required for this lab.

DURATION

This lab requires approximately *60* minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Learn more:** Where to find more information.
* **Caution:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **File contents:** A code block that displays the contents of a script or file you need to run that has been pre-created for you.
* **Knowledge check:** An opportunity to check your knowledge and test what you have learned.
* **Answer:** An answer to a question or challenge.
* **Refresh:** A time when you might need to refresh a web browser page or list to show new information.
* **Copy edit:** A time when copying a command, script, or other text to a text editor (to edit specific variables within it) might be easier than editing directly in the command line or terminal.
* **Task complete:** A conclusion or summary point in the lab.

**Start lab**

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

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

1. To open the lab, choose **Open Console**.

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

**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.

SERVICES USED IN THIS LAB

**Amazon EMR**

Amazon EMR is a managed service that makes it fast, easy, and cost-effective to run Apache Hadoop and Spark to process vast amounts of data. Amazon EMR also supports powerful and proven Hadoop tools such as Presto, Hive, Pig, HBase, and more.

**Learn more:** Refer to *What is Amazon EMR?* in the **Additional resources** section for more information.

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.

**Task 1: Create an Amazon S3 bucket**

In this task, you create an S3 bucket to store your log files and output data.

1. At the top of the AWS Management Console, in the search bar, search for and choose

S3

.

1. Choose **Create bucket**.

On the **Create bucket** page, configure the following:

* For **Bucket name**, enter

hadoopNUMBER

.

**Note:** Replace **NUMBER** with a random number.

* Choose **Create bucket**.

**Task complete:** You now have a bucket you can use to store your log files and output data.

**Task 2: Launch an Amazon EMR cluster**

In this task, you launch a Hadoop cluster, and then use it to process data.

1. At the top of the AWS Management Console, in the search bar, search for and choose

EMR

.

1. Ensure the region located at the top of your screen matches the value of **Region** located to the left of these instructions. If your region does not match, change your region to the value of **Region**.
2. Choose **Create cluster**.
3. On the **Create cluster** page, in the **Name and applications** section, configure the following:

* For **Name**, enter

My cluster

.

* For **Amazon EMR release**, select **emr-5.36.1** from the dropdown menu.
* For **Application bundle**, choose **Custom**.
* In the **Customize your application bundle** selection menu, select the following applications if not already selected:
  + Hue
  + Hadoop
  + Hive
  + Pig

1. In the **Cluster configuration** section, configure the following:

* Choose **Instance groups**.
* For **Primary**, **Core** and **Task 1 of 1**, select **m4.large** from the **Choose EC2 instance type** dropdown menu.

1. In the **Networking** section, configure the following:

* For **Virtual private cloud (VPC)** choose **Browse**.
* On the **Choose VPC** pop-up window, select **Lab VPC**.
* Choose **Choose**.
* Expand the **EC2 security groups (firewall)** section, and select the security group containing the name **xxxx-EmrSecurityGroup-xxxx** for both *Primary* and *Core and task* nodes.

1. In the **Cluster termination** section, de-select  **Use termination protection**.
2. Expand the **Cluster logs - *optional*** section, and configure the following:

* Select  **Publish cluster-specific logs to Amazon S3**.
* For **Amazon S3 location**, choose **Browse S3**.
* On the **Choose Amazon S3 location** pop-up window, select the Hadoop bucket name that was created earlier, and choose **Choose**.

1. In the **Security configuration and EC2 key pair - *optional*** section, configure the following:

* For **Amazon EC2 key pair for SSH to the cluster**, choose **Browse**.
* On the **Choose Amazon EC2 key pair for SSH to cluster** pop-up window, select the key pair named **EMRKey-lab** and choose **Choose**.

1. In the **Identity and Access Management (IAM) roles** section, configure the following:

* For **Amazon EMR service role**, select **EMR\_DefaultRole** from the **Service role** dropdown menu.
* For **EC2 instance profile for Amazon EMR**, select **EMR\_EC2\_DefaultRole** from the **Instance profile** dropdown menu.

1. Choose **Create cluster**.

**Note:** The cluster will take approximately five minutes to launch. Please continue reading while waiting for the cluster to launch.

HADOOP APPLICATIONS

The default configuration automatically installs several default applications on the cluster:

* **Apache Hadoop** is an open-source software project that can be used to efficiently process large datasets. Instead of using one large computer to process and store the data, Hadoop uses clusters of commodity hardware to analyze massive data sets in parallel.
* The **Ganglia** open-source project is a scalable, distributed system designed to monitor clusters and grids while minimizing the impact on their performance. Ganglia can generate reports and view the performance of the cluster as a whole, as well as inspect the performance of individual nodes.
* **Apache Tez** is a framework for creating a complex directed acyclic graph (DAG) of tasks for processing data. In some cases, it is used as an alternative to Hadoop MapReduce. For example, Pig and Hive workflows can run using Hadoop MapReduce or they can use Tez as an execution engine.
* **Hive** is an open-source data warehouse and analytic package that runs on top of a Hadoop cluster. Hive scripts use an SQL-like language called Hive QL (query language) that abstracts programming models and supports typical data warehouse interactions. Hive enables you to avoid the complexities of writing Tez jobs based on directed acyclic graphs (DAGs) or MapReduce programs in a lower level computer language, such as Java.
* **Hue** (Hadoop User Experience) is an open-source, web-based, graphical user interface for use with Amazon EMR and Apache Hadoop. Hue groups together several different Hadoop ecosystem projects into a configurable interface for your Amazon EMR cluster.
* **Pig** is an open-source Apache library that runs on top of Hadoop. The library takes SQL-like commands written in a language called Pig Latin and converts those commands into Tez jobs based on directed acyclic graphs (DAGs) or MapReduce programs. You do not have to write complex code using a lower level computer language, such as Java.

DATA TO BE PROCESSED

Once your cluster is ready, you use the cluster to process log data from Amazon CloudFront.

**Amazon CloudFront** is a web service that speeds up distribution of static and dynamic web content, such as .html, .css, .php, and image files. CloudFront delivers content through a worldwide network of data centers called *edge locations*. When a user requests content through CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance. If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately. If the content is not in that edge location, CloudFront retrieves it from an Amazon S3 bucket or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

Amazon CloudFront can produce *access logs* that show all data requested by users. The log files can grow very large, so Hadoop is an ideal way to process and analyze the log files.

Here is a sample of the log data:

2017-07-05 20:05:47 SEA4 4261 10.0.0.15 eabcd12345678.cloudfront.net /test-image-2.jpeg Mozilla/5.0%20(MacOS;%20U;%20Windows%20NT%205.1;%20en-US;%20rv:1.9.0.9)%20Gecko/2009040821%20Chrome/3.0.9

The data contains:

| **Field** | **Sample data** | **Definition** |
| --- | --- | --- |
| Date | 2017-07-05 | The date on which the event occurred. |
| Time | 20:05:47 | The time when the CloudFront server finished responding to the request (in UTC). |
| Edge Location | SEA4 | The edge location that served the request. Each edge location is identified by a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. |
| Bytes | 4261 | The total number of bytes that CloudFront served to the viewer in response to the request, including headers. |
| IP | 10.0.0.15 | The IP address of the viewer that made the request. |
| Method | GET | The HTTP access method: DELETE, GET, HEAD, OPTIONS, PATCH, POST, or PUT. |
| Host | abcd.cloudfront.net | The domain name of the CloudFront distribution. |
| URI | /test-image-2.jpeg | The portion of the URI that identifies the path and object |
| Status | 200 | An HTTP status code (eg 200 = success). |
| Referrer | - | The name of the domain that originated the request. |
| User Agent | Mozilla/5.0… | The User-Agent header identifies the source of the request, such as the type of device and browser that submitted the request and, if the request came from a search engine, which search engine. |

In the following task, you run a Hive job on your cluster to analyze this data.

**Task complete:** You successfully launched a Hadoop cluster using Amazon EMR.

**Task 3: Process Your Sample Data by Running a Hive Script**

In this task, you run a Hive script in your cluster as a *step* in the Amazon EMR console to process your sample data. In Amazon EMR, a *step* is a unit of work that contains one or more Hadoop jobs. You can submit *steps* when you create the cluster or when the cluster is running (if it is a long-running cluster).

1. **Caution:** Wait until your cluster is showing a status of **Waiting**.

**Refresh:** If your cluster is not yet in the *Waiting* state, wait for 10 minutes and choose the refresh  icon. Try again every few minutes until the cluster status is *Waiting*.

1. Choose the **Steps** tab.
2. Choose **Add step**.

On the **Add step** page, configure the following:

* For **Type**, choose **Hive program**.
* For **Name**, enter

Process logs

.

* For **Hive script location**, enter

s3://<REGION>.elasticmapreduce.samples/cloudfront/code/Hive\_CloudFront.q

 in the search textbox.

**Note:** Replace **<REGION>** with the value of **Region** located to the left of these instructions.

* For **Input Amazon S3 location - *optional***, enter

s3://<REGION>.elasticmapreduce.samples

 in the search textbox.

**Note:** Replace **<REGION>** with the value of **Region** located to the left of these instructions.

* For **Output Amazon S3 location - *optional***:
  + Choose **Browse S3**.
  + On the **Choose Amazon S3 location** pop-up window, select the Hadoop bucket name that was created earlier, and choose **Choose**.
* For **Arguments - *optional***, enter

-hiveconf hive.support.sql11.reserved.keywords=false

**Note:** This will allow column names that are the same as reserved words.

1. Choose **Add step**.

**Expected output:** The status of the step changes from *Pending* to *Running* to *Completed* as the step runs.

The step takes approximately one minute to run.

**Knowledge check:** What the script is doing?

**Answer:** The Hive script does the following:

* Creates a Hive table named *cloudfront\_logs*.
* Reads the CloudFront log files from Amazon S3 and parses the files using the Regular Expression Serializer/Deserializer (*RegEx SerDe*).
* Writes the parsed results to the *cloudfront\_logs* Hive table.
* Submits a HiveQL query against the data to retrieve the total requests per operating system for a given time frame.
* Writes the query results to your Amazon S3 output bucket.

**File contents:** The Hive code that creates the table is:

CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront\_logs (

DateObject Date,

Time STRING,

Location STRING,

Bytes INT,

RequestIP STRING,

Method STRING,

Host STRING,

Uri STRING,

Status INT,

Referrer STRING,

OS String,

Browser String,

BrowserVersion String

)

**File contents:** The Hive code that parses the log files using the RegEx SerDe looks like:

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'

WITH SERDEPROPERTIES (

"input.regex" = "^(?!#)([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+[^\(]+[\(](https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/spl-166/v1.0.23.prod-afbac65a/instructions/en\_us/[^\;]+).\*\%20([^\/]+)[\/](https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/spl-166/v1.0.23.prod-afbac65a/instructions/en\_us/.\*)$"

) LOCATION '${INPUT}/cloudfront/data/';

**File contents:** The HiveQL query that calculates *requests by operating system* is:

INSERT OVERWRITE DIRECTORY '${OUTPUT}/os\_requests/'

SELECT

os,

COUNT(\*) count

FROM cloudfront\_logs

WHERE dateobject

BETWEEN '2014-07-05' AND '2014-08-05'

GROUP BY os;

The sample data set contains approximately 5000 rows of data. This same process, however, can be used to process *millions of rows of data* in parallel across multiple nodes.

1. Wait for the status of the step to change to **Completed**.

**Refresh:** To update the status, choose the refresh  icon.

Your results are now available for viewing.

**Task complete:** You successfully ran a Hive script in your cluster as a *step* in the Amazon EMR console to process your sample data.

**Task 4: View the Results**

After the step completes successfully, the query output produced by the Hive script is stored in the Amazon S3 bucket that you specified when you submitted the step.

In this task, you view the results of the query output produced by the Hive script.

1. At the top of the AWS Management Console, in the search bar, search for and choose

S3

.

1. Choose the link to your *Hadoop* bucket.
2. Choose the **os\_requests** folder.

The Hive query results are stored in text files.

1. Download each file.
2. Open the files using a text editor such as WordPad (Windows), TextEdit (Mac OS), or gEdit (Linux).

In the output files, you should see the number of access requests by operating system:

This lab used a small sample of data. Normally, there would be significantly more rows of data in your CloudFront logs. Regardless of the size, the cluster automatically distributes the work across the cluster to complete the work required.

**Task complete:** You successfully viewed the results of the query output produced by the Hive script.

**Task 5 : Connect to the EMR cluster CLI and perform query using HiveQL**

In this task, you connect to the EMR cluster using SSH tool. Once you access the EMR cluster, you switch to Hive application and run query using HiveQL.

1. **Copy edit:** Copy the **CommandHostSessionManagementUrl** from the left of this instruction and paste it on a new tab of your browser.

**Expected output:** You will be redirected to the command host terminal.

1. **Command:** In the terminal, copy and paste below commands:

**Note:** Using the below command, you are retrieving the EMR cluster ID and then use that Cluster ID to get the public DNS of the cluster. Finally you are using the Cluster DNS value to SSH to the EMR Cluster 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-lab.pem hadoop@$HOST

1. Type

yes

 for the prompt.

**Expected output:** You will see the EMR terminal.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Warning: Permanently added 'ec2-35-160-218-246.us-west-2.compute.amazonaws.com,10.1.12.14' (ECDSA) to the list of known hosts.

Last login: Tue Jul 11 17:08:58 2023

\_\_| \_\_|\_ )

\_| ( / Amazon Linux 2 AMI

\_\_\_|\\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-2/

No packages needed for security; 1 packages available

Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE 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

EEEEEEEEEEEEEEEEEEEE MMMMMMM MMMMMMM RRRRRRR RRRRRR

1. **Command:** Type

hive

 in the terminal and press *enter*.

**Expected output:** You will be redirected to the hive application.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false

hive>

1. **Command:** Copy and paste the below HiveQL command and review the result.

SELECT

os,

COUNT(\*) count

FROM cloudfront\_logs

WHERE dateobject

BETWEEN '2014-07-05' AND '2014-08-05'

GROUP BY os;

**Expected output:** Here you can view the results that are similar to the previous task.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query ID = hadoop\_20230711172040\_1bc44b58-844c-4a29-b254-18f65f6d1aec

Total jobs = 1

Launching Job 1 out of 1

Status: Running (Executing on YARN cluster with App id application\_1689094149002\_0002)

----------------------------------------------------------------------------------------------

VERTICES MODE STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED

----------------------------------------------------------------------------------------------

Map 1 .......... container SUCCEEDED 1 1 0 0 0 0

Reducer 2 ...... container SUCCEEDED 2 2 0 0 0 0

----------------------------------------------------------------------------------------------

VERTICES: 02/02 [==========================>>] 100% ELAPSED TIME: 31.65 s

----------------------------------------------------------------------------------------------

OK

Linux 813

MacOS 852

OSX 799

iOS 794

Android 855

Windows 883

Time taken: 39.254 seconds, Fetched: 6 row(s)

**Task complete:** You successfully connected to the EMR cluster using SSH tool, accessed the EMR cluster, and switched to Hive application to run a query using HiveQL.

**Task 6: Terminate your Amazon EMR Cluster**

The cluster can be used to process many more jobs and can also be used interactively instead of submitting pre-built jobs.

In this task, you *terminate* the cluster, since it is no longer required for this lab.

1. Return to the **AWS Management Console** browser tab.
2. At the top of the AWS Management Console, in the search bar, search for and choose

EMR

.

1. In the left navigation pane, Choose **Clusters**.
2. Select  **My cluster**.
3. Choose **Terminate**.
4. On the **Terminate cluster** pop-up window, choose **Terminate**.

**Task complete:** You successfully terminated the EMR cluster.

**Conclusion**

You have successfully done the following:

* Launched a fully functional Hadoop cluster using Amazon EMR.
* Defined the schema and created a table for sample log data stored in Amazon S3.
* Analyzed the data using a *HiveQL* script and wrote the results back to Amazon S3.
* Downloaded and viewed the results on your computer.
* Connected to the Hive CLI and ran *HiveQL* query script to view the results.

**End lab**

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

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

**Additional resources**

* [Getting Started with AWS](https://aws.amazon.com/getting-started/)
* [Analytics on AWS](https://aws.amazon.com/big-data/datalakes-and-analytics/)
* [What is Amazon EMR?](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).