Introduction to EMR

In this Lab Step you will create an S3 bucket specifically for use with Amazon EMR. An S3 bucket is a prerequisite for using EMR. S3 buckets are a common source of errors when you are first starting out exploring EMR. For example, some MapReduce jobs may need the output folder to already exist, others may need to create it during processing.

Instructions

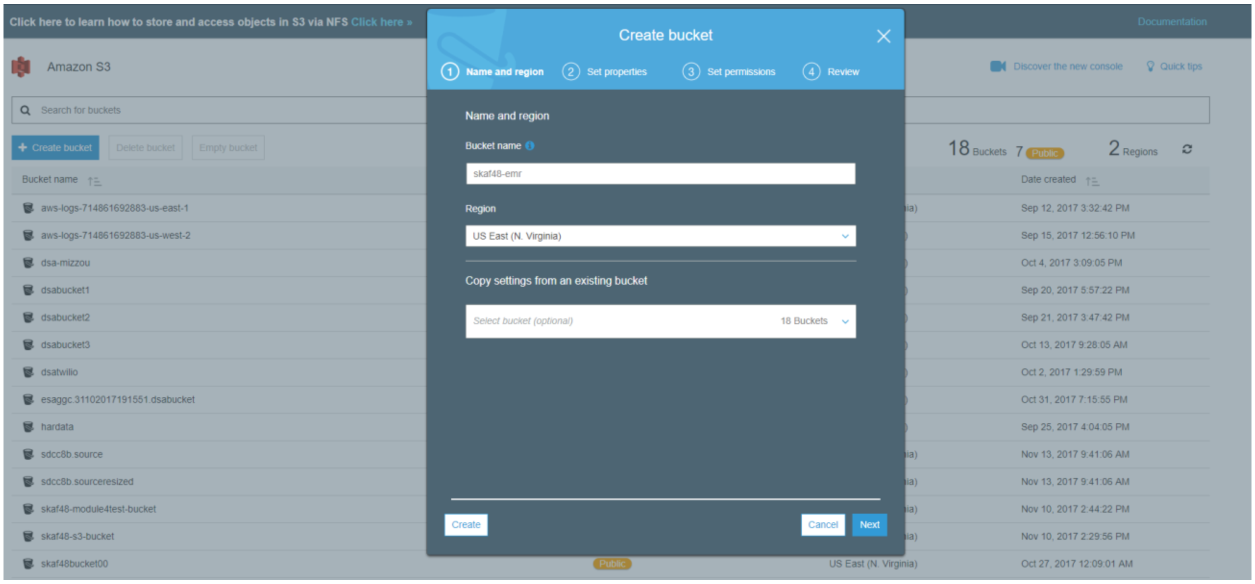
1. In the AWS Management Console, navigate to Services > Storage > S3.

2. Click Create bucket. Fill out the first screen of the Create bucket wizard:

Bucket name: <paw print>-emr (A S3 bucket name must be globally unique. You will be told if your bucket name is already used.

Region: US East

Click Create when ready. (Settings on other screens of the wizard are not required.)



Name and region

Bucket name: skaf48-emr

Region: US East (N. Virginia)

Copy settings from an existing bucket: Select bucket (optional) 18 Buckets

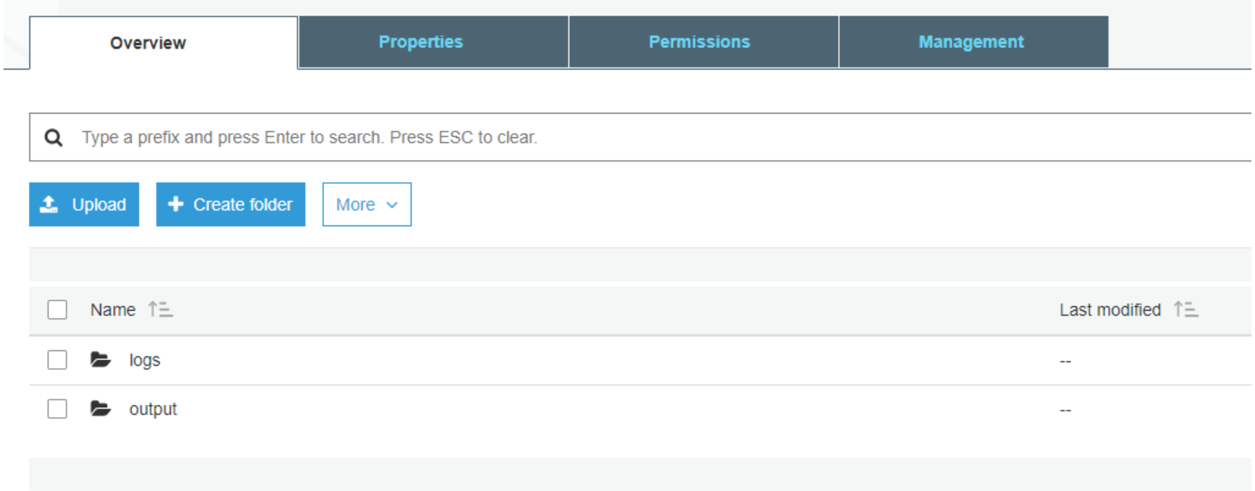
Bucket name			Date created
aws-logs-714861692883-us-east-1			Sep 12, 2017 3:32:42 PM
aws-logs-714861692883-us-west-2			Sep 15, 2017 12:56:10 PM
dsa-mizzou			Oct 4, 2017 3:09:05 PM
dsabucket1			Sep 20, 2017 5:57:22 PM
dsabucket2			Sep 21, 2017 3:47:42 PM
dsabucket3			Oct 13, 2017 9:28:05 AM
dsatwilio			Oct 2, 2017 1:29:59 PM
esaggc.31102017191551.dsabucket			Oct 31, 2017 7:15:55 PM
hardata			Sep 25, 2017 4:04:05 PM
sdcc8b.source			Nov 13, 2017 9:41:06 AM
sdcc8b.sourceresized			Nov 13, 2017 9:41:06 AM
skaf48-module4test-bucket			Nov 10, 2017 2:44:22 PM
skaf48-s3-bucket			Nov 10, 2017 2:29:56 PM
skaf48bucket00	Public	US East (N. Virginia)	Oct 27, 2017 12:09:01 AM

3. Beneath your <paw print>-emr - # bucket, click Create folder.

New folder name: output

4. Click Create folder again and create another folder at the same level as output:

New folder name: logs



□	Name	Last modified
□	logs	--
□	output	--

So far you have created one S3 bucket with necessary folders to store EMR processing logs and the results of a MapReduce job.

Creating an EMR Cluster

Introduction

An EMR cluster will have EC2 instances doing the job in the background. Amazon will take care of deploying and managing the instances in the EMR cluster. Depending on the size of the data set that needs to be processed, the number and size of the instances will vary. A cluster has three types of nodes:

Master - Manages the cluster and all software components needed to distribute both data and tasks to slave nodes. The master also tracks the status of tasks and the overall health of the cluster. A slave node can be a Core or Task node.

Core - A slave node that includes software to run tasks and store data. (Data is stored in a Hadoop Distributed File System (HDFS))

Task - A slave node that includes software to run tasks but does not store data.

A cluster typically contains a single master node, and one or more core slave nodes. A cluster can consist of a single Master node however. Optionally, task nodes may be used to help scale the cluster for more processing power.

Since we are dealing with small log files, this Lab will use one master and one core node. It's important to realize that in this Lab Step you will create, configure and launch your EMR cluster.

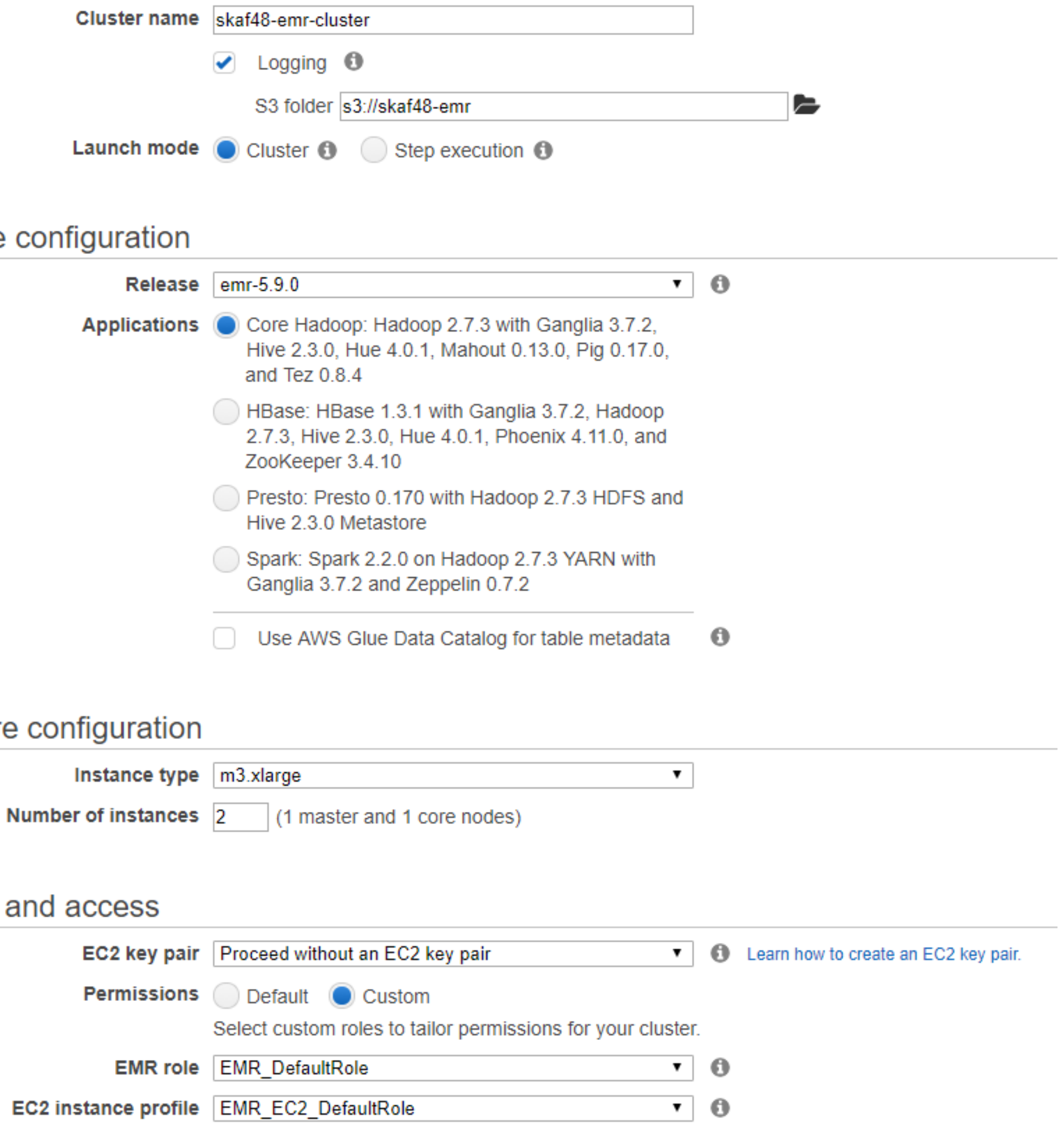
Amazon offers different ways to configure your cluster and their workloads. In this Lab you will configure your cluster in Launch mode. In Launch mode a running cluster can then accept and execute tasks you submit. The cluster continues to run until you explicitly terminate it. Many production environments run in Step execution mode. In Step execution mode the cluster starts, runs the "steps" (tasks) and terminates itself.

Since this is the first time for you using EMR, it is recommended to set up your initial configuration using "Cluster" as the launch mode. The continuous nature of this mode allows you to add a task, run it, and then tinker around debugging and even connecting to the instances if needed. You can then add the task again and repeat debugging efforts if needed!

Instructions

1. From the Amazon Management Console, navigate to Services > Analytics > EMR. You are placed in the Cluster list section of the EMR console.

2. Click Create cluster. You are placed in the Create Cluster - Quick Options screen:



Cluster name skaf48-emr-cluster

☑ Logging

S3 folder s3://skaf48-emr

Launch mode ● Cluster ○ Step execution

Software configuration

Release emr-5.9.0

Applications

● Core Hadoop: Hadoop 2.7.3 with Ganglia 3.7.2, Hive 2.3.0, Hue 4.0.1, Mahout 0.13.0, Pig 0.17.0, and Tez 0.8.4

○ HBase: HBase 1.3.1 with Ganglia 3.7.2, Hadoop 2.7.3, Hive 2.3.0, Hue 4.0.1, Phoenix 4.11.0, and ZooKeeper 3.4.10

○ Presto: Presto 0.170 with Hadoop 2.7.3 HDFS and Hive 2.3.0 Metastore

○ Spark: Spark 2.2.0 on Hadoop 2.7.3 YARN with Ganglia 3.7.2 and Zeppelin 0.7.2

□ Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type m3.xlarge

Number of instances 2 (1 master and 1 core nodes)

Security and access

EC2 key pair Proceed without an EC2 key pair

Learn how to create an EC2 key pair.

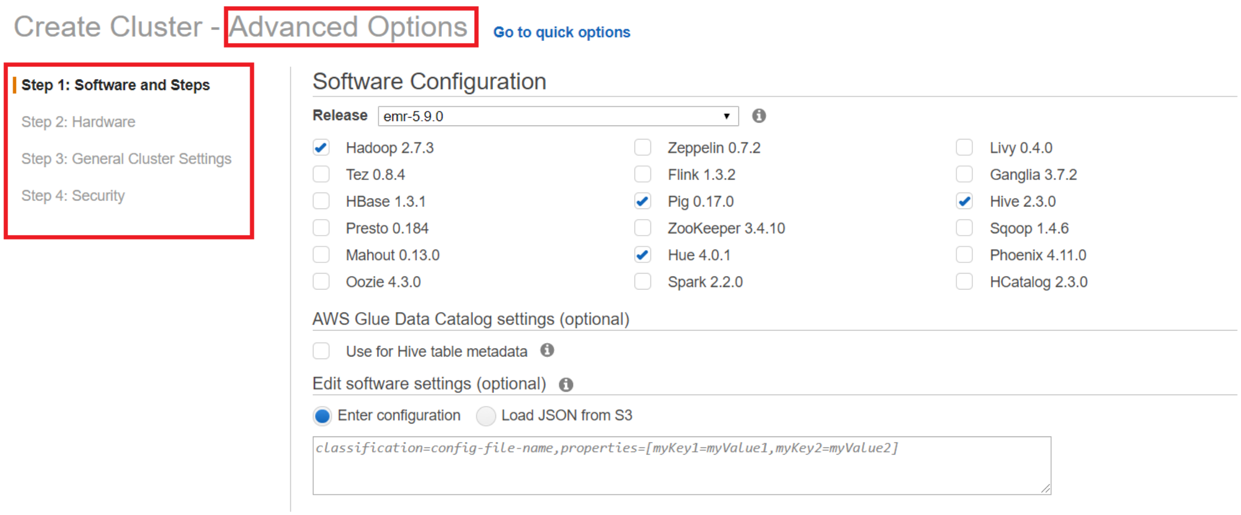
Permissions ○ Default ● Custom

Select custom roles to tailor permissions for your cluster.

EMR role EMR_DefaultRole

EC2 instance profile EMR_EC2_DefaultRole

3. Click Go to Advanced Options:



Create Cluster - Advanced Options Go to quick options

- **Step 1: Software and Steps**
- Step 2: Hardware
- Step 3: General Cluster Settings
- Step 4: Security

Software Configuration

Release emr-5.9.0

☑ Hadoop 2.7.3	□ Zeppelin 0.7.2	□ Livy 0.4.0
□ Tez 0.8.4	□ Flink 1.3.2	□ Ganglia 3.7.2
□ HBase 1.3.1	☑ Pig 0.17.0	☑ Hive 2.3.0
□ Presto 0.184	□ ZooKeeper 3.4.10	□ Sqoop 1.4.6
□ Mahout 0.13.0	☑ Hue 4.0.1	□ Phoenix 4.11.0
□ Oozie 4.3.0	□ Spark 2.2.0	□ HCatalog 2.3.0

AWS Glue Data Catalog settings (optional)

□ Use for Hive table metadata

Edit software settings (optional)

● Enter configuration ○ Load JSON from S3

```
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]
```

Advanced Options provides greater configuration options broken down into a multi-step wizard with a screen for four similar topics (Software, Hardware, General, Security) that the Quick Options includes.

4. Take a moment to look over the options on the Software and Steps screen. Don't worry about the details for the moment. When ready to proceed, click Next to see the Hardware configuration options. Click Next and look over all the options for each of the Advanced Options steps. Click Previous several times to return to Step 1: Software and Steps. (Important! Do not create your cluster yet!) This activity is simply meant to acquaint you with the type of options available when configuring your EMR cluster. It should give you a healthier understanding of the infrastructure and software underneath that Amazon provides and manages in order to make MapReduce simpler for you to use.

5. Click Go to quick options. For your first EMR cluster. Fill out the following for each section:

General Configuration

- Cluster name: Enter <paw print>-emr-cluster
- Logging: Leave checked (default)
- S3 folder: Enter the name of the unique S3 bucket previously created. For example: s3://<paw print>-emr
- Launch mode: Cluster (default)

Software Configuration

- Release: emr-5.6.0 (Many of the latest releases are supported.)

- Applications: Core Hadoop (default. Notice the software and versions that are included with the application package.)

Hardware Configuration

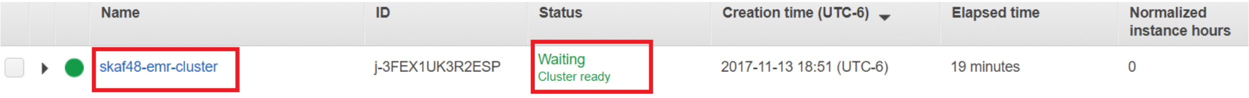
- Instance type: m3.xlarge (A larger instance type is not needed. Too small of an instance will have memory issues when the instance gets bootstrapped.)
- Number of instances: Change this to 2 (One master and one core node will suffice for this Lab.)
- Note: If you changed the instance type to a smaller computed optimized instance (c1.medium) the hardware specifications would be too limited for it to be a master node and process the task. The cluster may run but not have enough memory to process any sizable workloads. Determining the ideal instance size for your environment is a skill that some argue is part science, part art.

Security and access

- EC2 key pair: Select Proceed without an EC2 key pair (You will not need to SSH into the instance for this lab.)
- Permissions: Change to Custom. Then select the default EC2 and EMR roles
  - EMR_DefaultRole
  - EMR_EC2_DefaultRole

Click Create cluster when ready to proceed.

6. Click Clusters in the left hand navigation pane. All clusters are listed. Eventually, the cluster you just created should settle into a Status of Waiting - Cluster ready:



			Name	ID	Status	Creation time (UTC-6) ▾	Elapsed time	Normalized instance hours
□	▸	●	skaf48-emr-cluster	j-3FEX1UK3R2ESP	Waiting Cluster ready	2017-11-13 18:51 (UTC-6)	19 minutes	0

The amount of time for your cluster to be ready for use varies. Several factors can influence timing, including the size and number of instances, and whether debugging is turned on or not. Your cluster will probably take about 10 minutes before it's ready.

Until now you learned how to configure, create, launch your cluster. You were also exposed to the many different configuration options available to you when setting up your cluster.

You have a running Amazon EMR cluster and are ready to give it work to do!

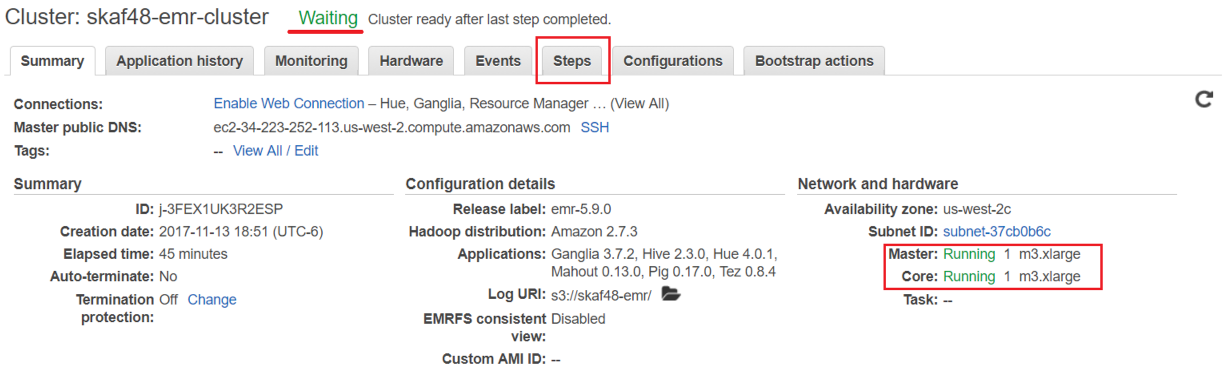
Adding a Step to your running Cluster

Introduction

In this Lab Step we will use an example Hive script that processes Amazon CloudFront (CF) log files. The script looks at the CF logs, identifies the different operating systems (OS) that make requests through CF, and tabulates how many for each OS. The hive script and the sample CF logs are made available on a public S3 bucket.

Instructions

1. Click on your running cluster name (<paw print>-emr-cluster) from the **Cluster list** section of the EMR console. A summary of the cluster is shown, along with several possible action buttons:



Cluster: skaf48-emr-cluster Waiting Cluster ready after last step completed.

Connections: Enable Web Connection – Hue, Ganglia, Resource Manager … (View All)
Master public DNS: ec2-34-223-252-113.us-west-2.compute.amazonaws.com SSH
Tags: -- View All / Edit

Summary
ID: j-3FEX1UK3R2ESP
Creation date: 2017-11-13 18:51 (UTC-6)
Elapsed time: 45 minutes
Auto-terminate: No
Termination protection: Off Change

Configuration details
Release label: emr-5.9.0
Hadoop distribution: Amazon 2.7.3
Applications: Ganglia 3.7.2, Hive 2.3.0, Hue 4.0.1, Mahout 0.13.0, Pig 0.17.0, Tez 0.8.4
Log URI: s3://skaf48-emr/
EMRFS consistent view: Disabled
Custom AMI ID: --

Network and hardware
Availability zone: us-west-2c
Subnet ID: subnet-37cb0b6c
Master: Running 1 m3.xlarge
Core: Running 1 m3.xlarge
Task: --

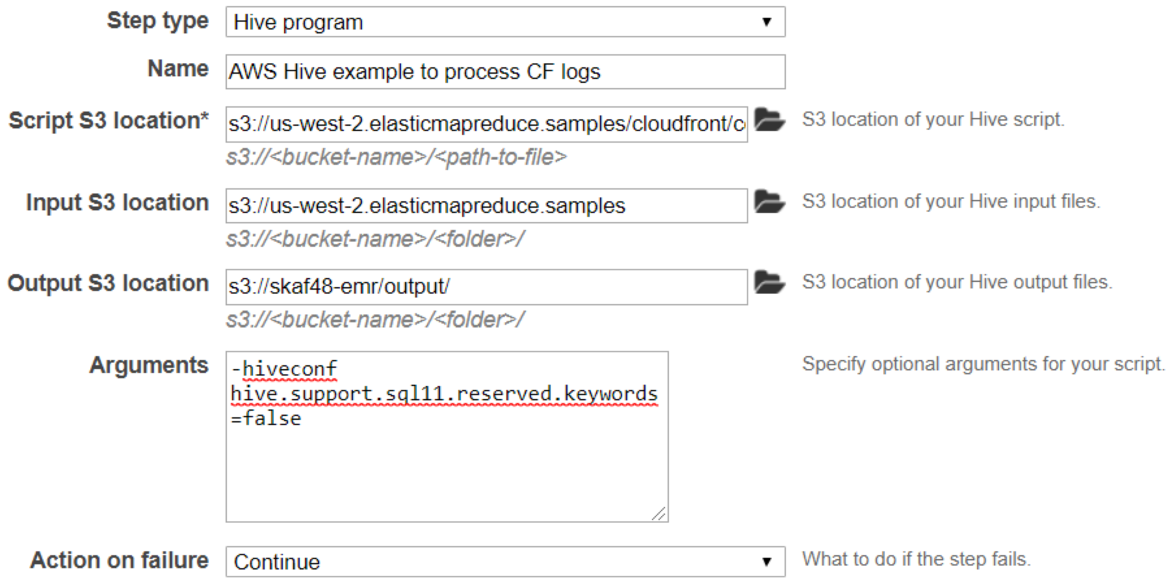
2. Click on steps and then click on Add step in order to submit the example script for processing by your EMR cluster.

3. Select Hive program in the Step type drop-down menu. The context changes based on the Step type:

4. Fill out the step details as follows:

- Name: Enter AWS Hive example to process CF logs
- Script S3 location: s3://us-west-2.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q
- Input S3 location: s3://us-west-2.elasticmapreduce.samples
- Output S3 location: s3://<paw print>-emr/output/
- Arguments: -hiveconf hive.support.sql11.reserved.keywords=false
- Action on failure: Select Continue (If the processing fails continuing will allow basic debugging, reconfiguration, and additional attempts.)

Tip: Take your time filling out the fields, it's pretty easy to misconfigure the settings above. For example, incorrect path to the S3 bucket, forgetting to create or include the output folder, doubling up on the s3:// protocol in the path, etc. Your Add step dialog should look similar to:



Step type	Hive program ▾	
Name	AWS Hive example to process CF logs	
Script S3 location*	s3://us-west-2.elasticmapreduce.samples/cloudfront/c *s3://<bucket-name>/<path-to-file>*	S3 location of your Hive script.
Input S3 location	s3://us-west-2.elasticmapreduce.samples *s3://<bucket-name>/<folder>/*	S3 location of your Hive input files.
Output S3 location	s3://skaf48-emr/output/ *s3://<bucket-name>/<folder>/*	S3 location of your Hive output files.
Arguments	-hiveconf hive.support.sql11.reserved.keywords =false	Specify optional arguments for your script.
Action on failure	Continue ▾	What to do if the step fails.

Click **Add** when ready to proceed.

6. Return to the **Cluster list**. The **Status** changes from **Waiting - Cluster ready** to **Running**:

Select the running cluster name (<paw print>-emr-cluster). The summary information shows the cluster status **Running** and the **Step Running** as well:

You have a running EMR cluster. The Hive script uploaded is provided by Amazon, along with example CloudFront logs. Processing log files is a very common Big Data use case. The logs used in the example were from CloudFront, but they could have been from another AWS service such as CloudTrail, or Apache HTTP access logs etc. Similarly, the example Hive script could have been a Pig program, a streaming Python program, or a custom Java application (JAR file).

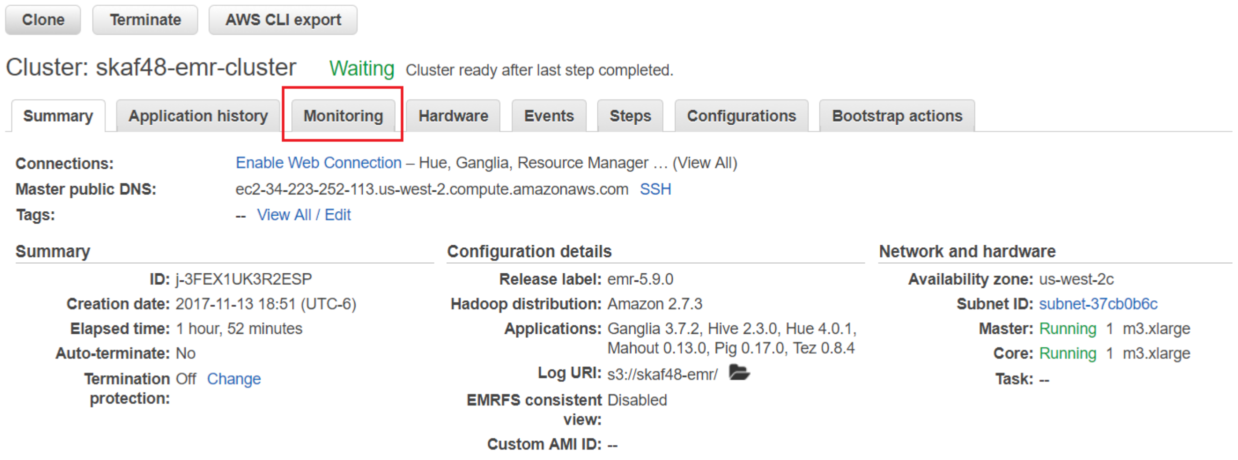
Viewing the EMR Cluster and Step Results

As with any program, viewing and confirming proper results is a common last phase in the process. There are several ways to view results and even monitor progress before, during and after job execution. In this Lab Step you will explore several ways to view results within the EMR console.

Instructions

1. Click **Cluster list** in the left pane of the EMR console, then click on the name of your running cluster.

2. Expand the **Monitoring** section:



Cluster: skaf48-emr-cluster Waiting Cluster ready after last step completed.

Connections: Enable Web Connection – Hue, Ganglia, Resource Manager ... (View All)
Master public DNS: ec2-34-223-252-113.us-west-2.compute.amazonaws.com SSH
Tags: -- View All / Edit

Summary

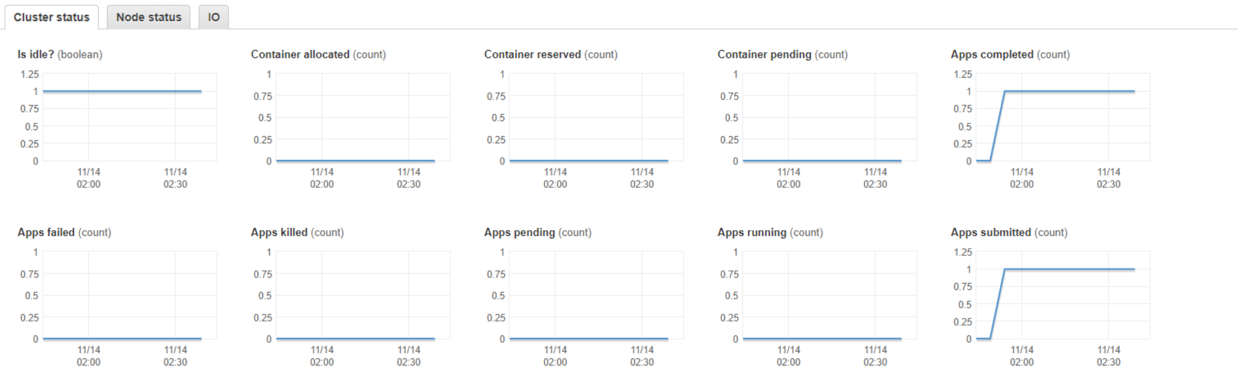
ID: j-3FEX1UK3R2ESP
Creation date: 2017-11-13 18:51 (UTC-6)
Elapsed time: 1 hour, 52 minutes
Auto-terminate: No
Termination protection: Off Change

Configuration details

Release label: emr-5.9.0
Hadoop distribution: Amazon 2.7.3
Applications: Ganglia 3.7.2, Hive 2.3.0, Hue 4.0.1, Mahout 0.13.0, Pig 0.17.0, Tez 0.8.4
Log URI: s3://skaf48-emr/
EMRFS consistent view: Disabled
Custom AMI ID: --

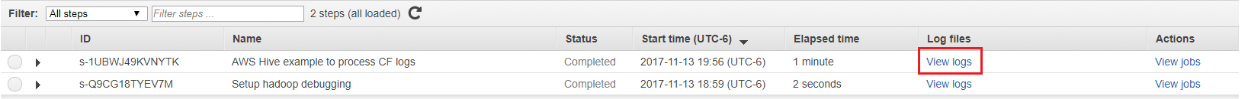
Network and hardware

Availability zone: us-west-2c
Subnet ID: subnet-37cb0b6c
Master: Running 1 m3.xlarge
Core: Running 1 m3.xlarge
Task: --



In the example above, the history reveals one application that was submitted, and one that is completed successfully. None of the applications failed. Currently, no apps are pending or running.

3. Expand the **Steps** section:



Filter: All steps ▾ Filter steps ... 2 steps (all loaded)

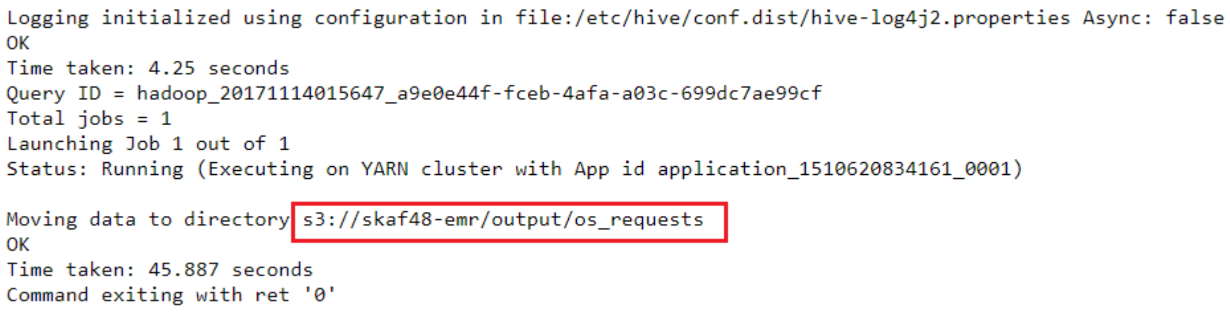
		ID	Name	Status	Start time (UTC-6)	Elapsed time	Log files	Actions
○	▸	s-1UBWJ49KVNYTK	AWS Hive example to process CF logs	Completed	2017-11-13 19:56 (UTC-6)	1 minute	View logs	View jobs
○	▸	s-Q9CG18TYEV7M	Setup hadoop debugging	Completed	2017-11-13 18:59 (UTC-6)	2 seconds	View logs	View jobs

4. Click the **View logs** link for the Step you just ran. There are three log files (**controller, stderr**, and **stdout**). These logs files are stored in S3 bucket's **logs** folder. If the log files are not yet available, the display conveys "no logs created yet" adjacent to the refresh icon.

5. Click **stdout**. Stdout is not that interesting for a Step that ran to successful completion, but it's good to know it's there.

6. Click **controller**. The controller log file contains more information than most will want to see, including Linux shell environment information, stdout and stderr log files locations on the instance, and the entire hadoop command line for running the Hive script, etc. This log file can be helpful if troubleshooting efforts are required.

7. Click **stderr**. The stderr log file contains very useful information, even if troubleshooting is not required:

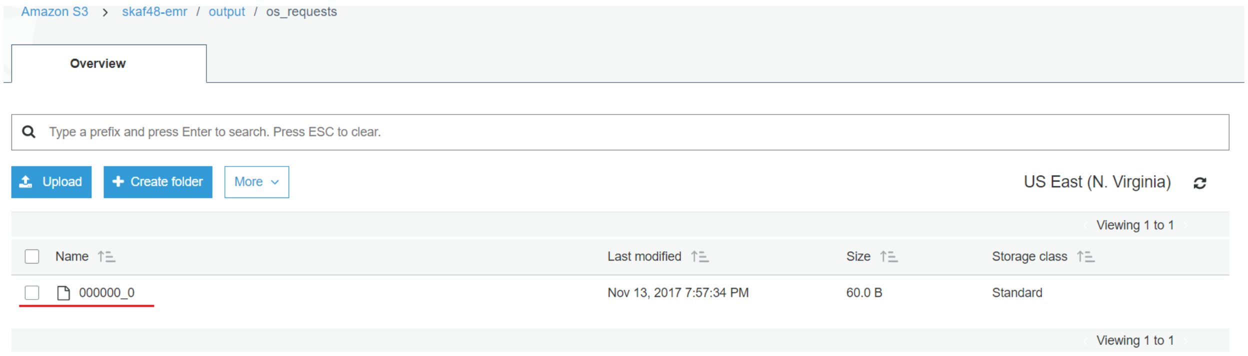


```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
OK
Time taken: 4.25 seconds
Query ID = hadoop_20171114015647_a9e0e44f-fceb-4afa-a03c-699dc7ae99cf
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1510620834161_0001)

Moving data to directory s3://skaf48-emr/output/os_requests
OK
Time taken: 45.887 seconds
Command exiting with ret '0'
```

Note the location in S3 where the output was moved to.

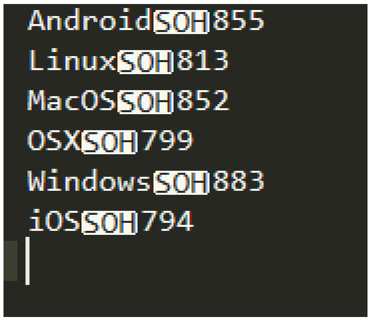
8. Navigate in your S3 bucket to output > os_requests. The file containing the results is shown:



Amazon S3 > skaf48-emr / output / os_requests

□	Name	Last modified	Size	Storage class
□	000000_0	Nov 13, 2017 7:57:34 PM	60.0 B	Standard

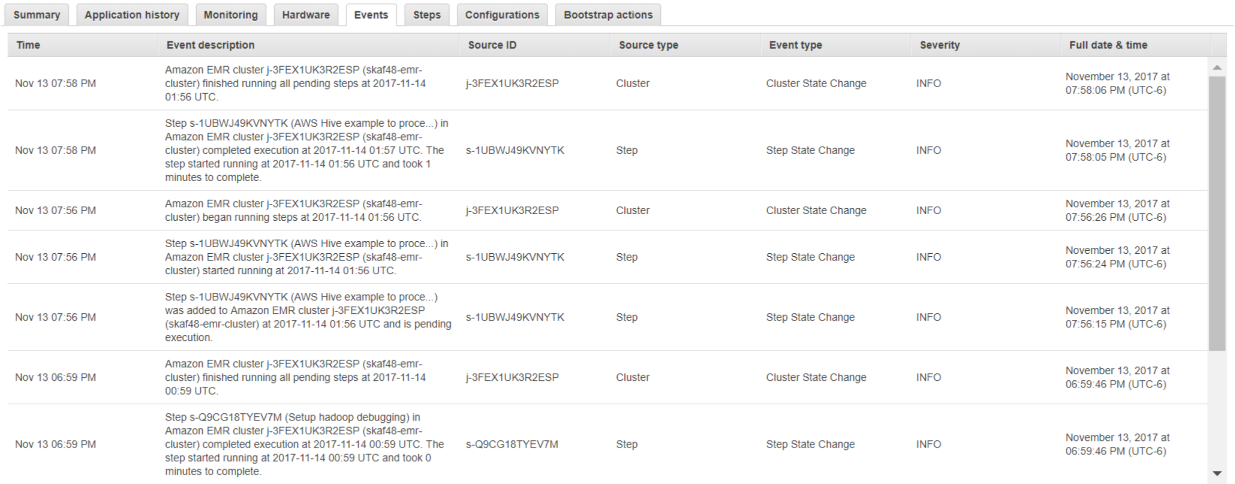
9. Click the file name link (00000_0) then the **Open** button to see the results:



```
AndroidSOH855
LinuxSOH813
MacOSSOH852
OSXSOH799
WindowsSOH883
iOSSOH794
```

The Hive script calculated the totals for each operating system that accessed CloudFront in the sample dataset. In the example shown, about 800 for each of the different operating systems (Android, Linux, ... iOS).

10. Return to the EMR console, select your running cluster then expand the **Events** section:



Time	Event description	Source ID	Source type	Event type	Severity	Full date & time
Nov 13 07:58 PM	Amazon EMR cluster j-3FEX1UK3R2ESP (skaf48-emr-cluster) finished running all pending steps at 2017-11-14 01:56 UTC.	j-3FEX1UK3R2ESP	Cluster	Cluster State Change	INFO	November 13, 2017 at 07:58:06 PM (UTC-6)
Nov 13 07:58 PM	Step s-1UBWJ49KVNYTK (AWS Hive example to proce...) in Amazon EMR cluster j-3FEX1UK3R2ESP (skaf48-emr-cluster) completed execution at 2017-11-14 01:57 UTC. The step started running at 2017-11-14 01:56 UTC and took 1 minutes to complete.	s-1UBWJ49KVNYTK	Step	Step State Change	INFO	November 13, 2017 at 07:58:05 PM (UTC-6)
Nov 13 07:56 PM	Amazon EMR cluster j-3FEX1UK3R2ESP (skaf48-emr-cluster) began running steps at 2017-11-14 01:56 UTC.	j-3FEX1UK3R2ESP	Cluster	Cluster State Change	INFO	November 13, 2017 at 07:56:26 PM (UTC-6)
Nov 13 07:56 PM	Step s-1UBWJ49KVNYTK (AWS Hive example to proce...) in Amazon EMR cluster j-3FEX1UK3R2ESP (skaf48-emr-cluster) started running at 2017-11-14 01:56 UTC.	s-1UBWJ49KVNYTK	Step	Step State Change	INFO	November 13, 2017 at 07:56:24 PM (UTC-6)
Nov 13 07:56 PM	Step s-1UBWJ49KVNYTK (AWS Hive example to proce...) was added to Amazon EMR cluster j-3FEX1UK3R2ESP (skaf48-emr-cluster) at 2017-11-14 01:56 UTC and is pending execution.	s-1UBWJ49KVNYTK	Step	Step State Change	INFO	November 13, 2017 at 07:56:15 PM (UTC-6)
Nov 13 06:59 PM	Amazon EMR cluster j-3FEX1UK3R2ESP (skaf48-emr-cluster) finished running all pending steps at 2017-11-14 00:59 UTC.	j-3FEX1UK3R2ESP	Cluster	Cluster State Change	INFO	November 13, 2017 at 06:59:46 PM (UTC-6)
Nov 13 06:59 PM	Step s-Q9CG18TYEV7M (Setup hadoop debugging) in Amazon EMR cluster j-3FEX1UK3R2ESP (skaf48-emr-cluster) completed execution at 2017-11-14 00:59 UTC. The step started running at 2017-11-14 00:59 UTC and took 0 minutes to complete.	s-Q9CG18TYEV7M	Step	Step State Change	INFO	November 13, 2017 at 06:59:46 PM (UTC-6)

Both EMR cluster and Steps executed are logged in a very readable format, that includes a date/time stamp. The example shown above shows INFO level events, but additional **Severity** can be shown as well. (Such as DEBUG.) Both logging and debugging can be enabled during the cluster creation process. Debugging requires that logging is also enabled.

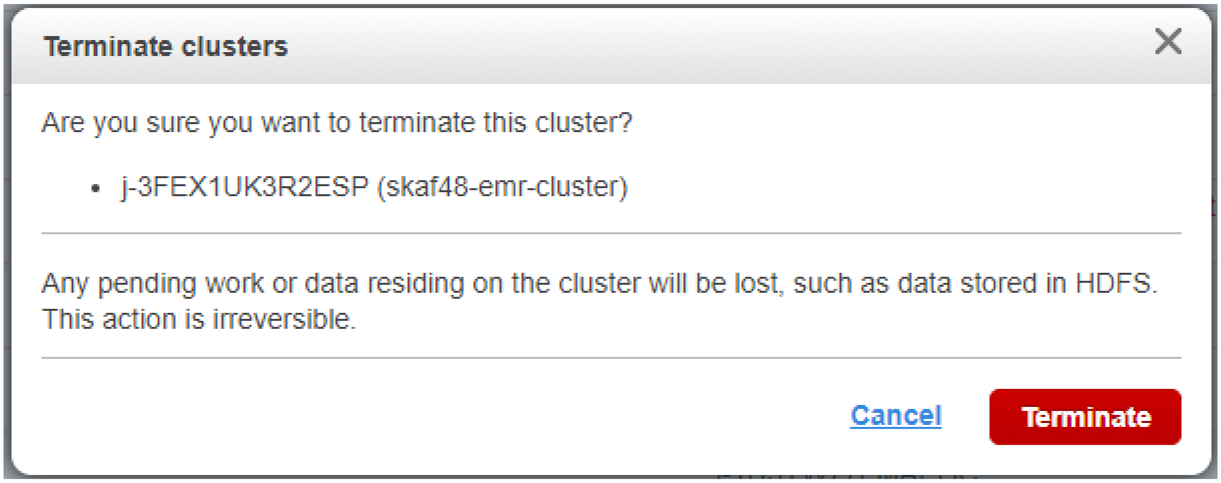
You have seen how well thought out the Amazon EMR console is. Right from within the console you can check status in a convenient and simple to read format. Further, you can see the overall health of your cluster, as well as the health of apps already executed or currently executing.

In EC2 you can stop instances when you are done working for the day, or the week, then start them again when needed. For clusters that are started in "Cluster" launch mode rather than "Step execution" mode, you can not stop them. They must be manually terminated.

Instructions

1. Navigate to the **Cluster list** section of the EMR console.

2. Select the checkbox of your running cluster, then click the **Terminate** button. You are presented with a **Terminate clusters** dialog:



Terminate clusters

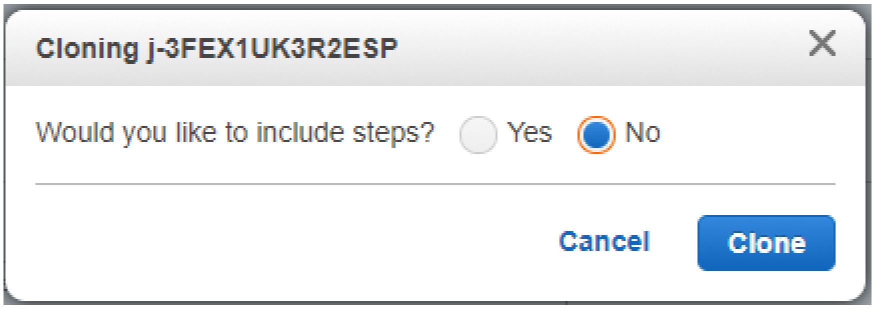
Are you sure you want to terminate this cluster?

- j-3FEX1UK3R2ESP (skaf48-emr-cluster)

Any pending work or data residing on the cluster will be lost, such as data stored in HDFS. This action is irreversible.

3. Click **Terminate**. The status changes to **Terminating - User request**. It takes approximately 1-2 minutes for the cluster **Status** to transition to **Terminated - User request**. Although you cannot use the same cluster again the way you can with a stopped EC2 instance, you can clone it and start the cloned cluster.

4. Select the checkbox for the terminated cluster, then click **Clone**. A Clone dialog window will ask **if you would like to include steps?**



Cloning j-3FEX1UK3R2ESP

Would you like to include steps? ○ Yes ● No

5. Change the radio button to **No** and and click **Clone**.

You are placed in the last step of the **Create Cluster - Advanced Options** wizard. In the same manner that you can configure **Software, Hardware, General Cluster** and **Security Settings** when creating the original cluster, you can do the same here. For example, if you wanted to install Sqoop on the new cluster you could do so on the **Software** step. If you suspect your application is using too much memory on the Master node, you could change the instance type on the **Hardware** step. Similarly, you could add Core nodes, change the **Cluster name**, where it logs to, **Termination**protection, and so on. (You can navigate steps 1 to 4 of the wizard using the **Next** and **Previous** buttons.)

6. Click **Create cluster** on the **Security** step (screen 4 of the wizard). The cloned cluster transitions to **Starting**. The cloned cluster will be available for use in about 10 minutes. (Exact timing varies.)

7. Click **Cluster list** in the left navigation of the EMR console, then expand the cluster that is **Starting**.

Note: If you had answered "Yes" to include steps earlier, the steps would be copied to the new cluster:

Note that all steps previously run by the original cluster are copied over to the clone. That includes steps that completed successfully and also failed steps. In this lab you will start from a fresh cluster with no steps carried over from the original one.

Technical Detour

One last helpful scenario to consider before summarizing this Lab Step and moving on: Imagine your debug efforts while running in **Cluster Launch** mode are completed. Now you want to run in **Step execution** mode, where the cluster will automatically terminate after the Steps processed complete.

The following is a summary of how to accomplish this (*but please don't do this in the current lab*):

- Select the cluster and clone it. (Don't include the steps from the cloned cluster.)
- On the **Software and Steps** wizard, go to the optional **Add steps** section
- Check the **Auto-terminate cluster after the last step is completed** checkbox
- Select the proper **Step type** (Streaming, Hive, etc.) and click **Configure.** Add the Step as you normally would.
- Page through the remainder of the wizard, confirming all looks correct, then create the cluster.

As you would expect, the cluster should Start, transition to Running, run the Step (or Steps) and then self terminate. You learned how to terminate a cluster, then clone and start up the new clone at your own convenience. Not only can this save on cost, it is a great way to tweak the cluster configuration as needed and run additional MapReduce jobs, etc. You also learned how you could clone a cluster and change various configuration options including the launch mode.