Introduction to AWS

Elastic Map Reduce (EMR)

Elastic Map Reduce 5.2.0

Installed:

- Hadoop 2.7.3
- Hive 2.1
- HBase 1.2.3
- Spark 2.0.2

Contents of releases:
http://docs.aws.amazon.com//ElasticMapReduce/latest/ReleaseGuide/emr-whatsnew.html

Example running sample Hive script

Steps to start using AWS EMR

Setup

- create keys for access,
- create buckets for data
- launch a cluster

Use data available for examples on AWS

Use a sample Hive script

Step 1: Sign Up for AWS

If you do not have an AWS account, use the following procedure to create one.

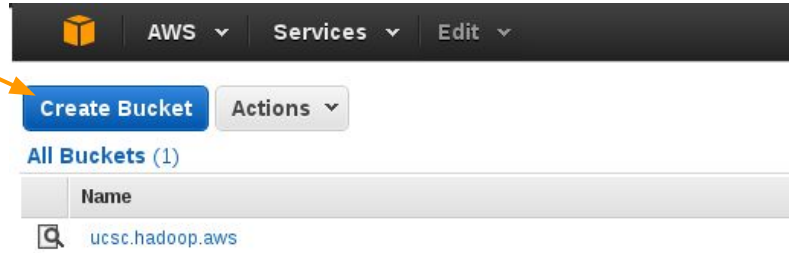
To sign up for AWS

1. Open https://aws.amazon.com/emr/ and
2. Click Get started with Amazon EMR
3. Follow the on-screen instructions - create an account.

Step 2: Create a place for your data

Open the s3 console: https://console.aws.amazon.com/s3

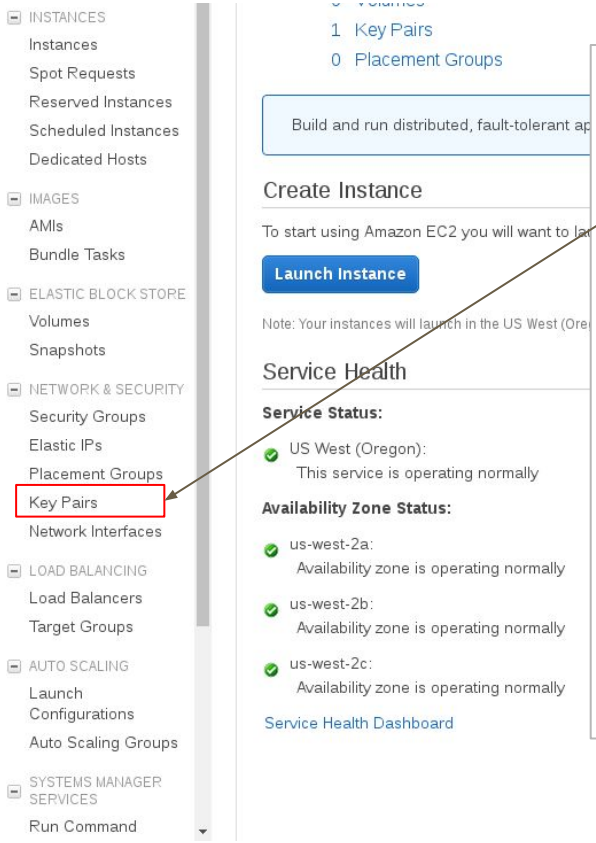
- Click



  - Give your bucket a unique name - e.g. ucsc.hadoop.aws
    - Only lowercase letters, numbers, periods (.), and hyphens (-)
    - Cannot end in numbers
  - Enable logging
- Click on the bucket you just created and create folders:
    - logs
    - output

Step 3: Open EC2 Management Console:

https://console.aws.amazon.com/ec2/



In left navigation panel,

- **select** NETWORK & SECURITY -> Key Pairs

In the next screen, click "**Create Key Pair**"

- provide a name for the key pair
- save the resulting pem file

Note: Location defaults to the "*Oregon*" region. Oregon region is referred to as "us-west-2" in file paths

Further information on key pairs see: Amazon EC2 Key Pairs

Step 4: Launch a cluster

Open the EMR console: https://console.aws.amazon.com/elasticmapreduce/.

1. Click Create cluster.

2. On the cluster configuration page, accept the defaults *except*:
   - For the hardware configuration, choose m1.medium (cheaper)
   - For EC2 key pair, choose the key pair that you created.

3. Choose Create cluster.

Step 5a: Launching a job

In the Amazon EMR console at https://console.aws.amazon.com/elasticmapreduce/.

1. In **Cluster List**, select the name of your cluster.
2. Scroll to the **Steps** section and expand it, then choose **Add step**. This will bring up:

Step type	Hive program
Name	Hive program
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	type or browse to the output bucket that you created above.
Arguments	leave the field blank.
Action on failure	Accept the default (**Continue**)

After you have defined the step, click **"Add"**

Step 5c. The running job

Initially, the step appears in the console with a status of **Pending**.

AWS Services Edit

Elastic MapReduce Cluster List

Create cluster | View details | Clone | Terminate

Filter: All clusters | Filter clusters ... | 1 cluster (all loaded)

Name	ID	Status	Creation time (UT
My cluster	j-WBF3SMVRL61C	Waiting	2015-08-01 11:36 (

Summary

Master public DNS: ec2-52-10-84-9.us-west-2.compute.amazonaws.com

Termination protection: Off Change

Tags: -- View All / Edit

Hardware Resize

Master: Running 1 m3.xlarge

Core: Running 2 m3.xlarge

Task: --

View cluster details | View monitoring details

Steps Add Step View all interactive jobs

Name	Status	Start time (UTC-7)	Elapsed time
Hive program	Pending		
Hive program	Completed	2015-08-01 12:23 (UTC-7)	1 minute
Setup hadoop debugging	Completed	2015-08-01 12:23 (UTC-7)	3 seconds

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

To update the status, choose **Refresh**

Step 6. Check out the completed job

click on the completed step to see the job info

refresh until you see logs

drill down to see attempt logs

Hive program

Status: Completed
ID: s-2X20TLV2VA408
Log files: No logs created yet
JAR location: command-runner.jar
Main class: None
Arguments: hive-script --run-hive-script --args -f s3://us-west-2.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q -d INPUT=s3://us-west-2.elasticmapreduce.samples -d OUTPUT=s3://ucsc.hadoop.aws/output/
Action on failure: Continue

Start time: 2015-08-05 10:28 (UTC-7)
Elapsed time: 58 seconds

Jobs > **Tasks**
Tasks for: s-2X20TLV2VA408, Job 1438456875247_0002
Task summary: 2 total tasks - 2 completed, 0 running, 0 failed, 0 pending, 0 cancelled.

Filter:

Task	Type	State	Start time (UTC-7)	Actions
r_000000	REDUCE	COMPLETED	2015-08-05 10:29:37 (UTC-7)	View attempts
m_000000	MAP	COMPLETED	2015-08-05 10:29:09 (UTC-7)	View attempts

Cancel

Step 6: If your job failed: debugging

ID	Name	Status	Start time (UTC-8)	Elapsed time	Log files	Actions
s-3UWRRLDSIDGLS	Hive program	Failed	2016-12-05 14:16 (UTC-8)	2 minutes	controller | syslog* | stderr | stdout	View jobs

JAR location: command-runner.jar
Main class: None
Arguments: hive-script --run-hive-script --args -f s3://us-west-2.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q -d INPUT=s3://us-west-2.elasticmapreduce.samples -d "OUTPUT=s3://ucsc.hadoop.emr/new folder/"
Action on failure: Continue

Click on the refresh icon

and then check the stderr log.

Step 6: debugging a failed job - the stderr log

https://aws-logs-487740761633-us-west-2.s3-us-west-2.amazonaws.com/elasticmapreduce/j-12LCDLEWT9HGC/steps/s-3UWRRLDSIDGLS/stderr.gz

Apps | Bookmarks | amazon EMR | java | blogs | personal | sysadmin | UCSC lectures | UCSC websites | todo | summer2016 | Fall2016 | bio d

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
FailedPredicateException(identifier,{useSQL11ReservedKeywordsForIdentifier()}?)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.identifier(HiveParser_IdentifiersParser.java:11914)
        at org.apache.hadoop.hive.ql.parse.HiveParser.identifier(HiveParser.java:51833)
        at org.apache.hadoop.hive.ql.parse.HiveParser.columnNameType(HiveParser.java:42051)
        at org.apache.hadoop.hive.ql.parse.HiveParser.columnNameTypeOrPKOrFK(HiveParser.java:42308)
        at org.apache.hadoop.hive.ql.parse.HiveParser.columnNameTypeOrPKOrFKList(HiveParser.java:37938)
        at org.apache.hadoop.hive.ql.parse.HiveParser.createTableStatement(HiveParser.java:5259)
        at org.apache.hadoop.hive.ql.parse.HiveParser.ddlStatement(HiveParser.java:2763)
        at org.apache.hadoop.hive.ql.parse.HiveParser.execStatement(HiveParser.java:1756)
        at org.apache.hadoop.hive.ql.parse.HiveParser.statement(HiveParser.java:1178)
        at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:204)
        at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:166)
        at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:404)
        at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:329)
        at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1158)
        at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1253)
        at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1084)
        at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1072)
        at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:232)
        at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:183)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:399)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:335)
        at org.apache.hadoop.hive.cli.CliDriver.processReader(CliDriver.java:429)
        at org.apache.hadoop.hive.cli.CliDriver.processFile(CliDriver.java:445)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:748)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:714)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:641)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
FAILED: ParseException line 3:2 Failed to recognize predicate 'Date'. Failed rule: 'identifier' in column specification
Command exiting with ret '64'
```

Background: Actual Hive script

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (
  Date 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
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  "input.regex" = "^(?!#)([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^
]+)\\s+[^\(]+[\(]([^\;]+).*\%20([^\/]+)[\/](.*)$"
) LOCATION '${INPUT}/cloudfront/data';

-- Total requests per operating system for a given time frame
INSERT OVERWRITE DIRECTORY '${OUTPUT}/os_requests/' SELECT os, COUNT(*) count FROM cloudfront_logs WHERE
date BETWEEN '2014-07-05' AND '2014-08-05' GROUP BY os;
```

Background: What the Hive script does

- Creates a Hive table named cloudfront_logs.
  - Reads the CloudFront log files from Amazon S3 and parses them.
  - Writes the parsed results to a Hive table, cloudfront_logs.
- Submits a Hive query against the table to count the total requests per OS for a given time frame.
- Writes the query results to the Amazon S3 output bucket.

How do I fix it?

Ideas?

Running on an older version of EMR

Cluster: Another old cluster Running Running step

Connections: Enable Web Connection – Spark History Server, Resource Manager ... (View All)
Master public DNS: ec2-35-165-24-84.us-west-2.compute.amazonaws.com SSH
Tags: -- View All / Edit

Summary
ID: j-3GKC7HOQ1GFUM
Creation date: 2016-12-05 15:08 (UTC-8)
Elapsed time: 16 minutes
Auto-terminate: No
Termination protection: Off Change

Configuration Details
Release label: emr-4.0.0
Hadoop distribution: Amazon 2.6.0
Applications: Hive 1.0.0, Mahout 0.10.0, Pig 0.14.0, Spark 1.4.1
Log URI: s3://aws-logs-487740761633-us-west-2/elasticmapreduce/
EMRFS consistent view: Disabled

Network and Hardware
Availability zone: us-west-2a
Subnet ID: subnet-3d0d914a
Master: Running 1 m1.medium
Core: Running 2 m1.medium
Task: --

Security and Access
Key name: ucsc.hadoop.aws
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-d30adab7 (ElasticMapReduce-master)
Security groups for Core & Task: sg-dc0adab8 (ElasticMapReduce-slave)

Monitoring

Hardware

Steps

Add step | Clone step | Cancel step

Steps

View all interactive jobs | View all jobs

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

ID	Name	Status	Start time (UTC-8)	Elapsed time	Log files	Actions
s-SF8MCNMDK7TH	Hive program	Completed	2016-12-05 15:24 (UTC-8)	2 minutes	View logs	View jobs

JAR location: command-runner.jar
Main class: None
Arguments: hive-script --run-hive-script --args -f s3://us-west-2.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q -d INPUT=s3://us-west-2.elasticmapreduce.samples -d OUTPUT=s3://ucsc.hadoop.aws/output/
Action on failure: Continue

Step 7: Stop spending money

- **Terminate your cluster**
- Go to https://console.aws.amazon.com/elasticmapreduce/
  1. On the **Cluster List** page, select your cluster and choose **Terminate**.
  2. By default, clusters created using the console are launched with termination protection enabled, so you must disable it. In the **Terminate clusters** dialog, for **Termination protection**, choose **Change**.
  3. Choose **Off** and then confirm the change.
  4. Choose **Terminate**.

Terminating cluster

Elastic MapReduce > Cluster List > Cluster Details

EMR He

Add step | Resize | Clone | Terminate

Cluster: My cluster **Terminating** Terminated by user request

Connections: --
Master public DNS: ec2-52-10-84-9.us-west-2.compute.amazonaws.com SSH
Tags: --

Summary

ID: j-WBF3SMVRL61C
Creation date: 2015-08-01 11:36 (UTC-7)
Elapsed time: 3 days, 23 hours
Auto-terminate: No
Termination protection: Off

Configuration Details

Release label: emr-4.0.0
Hadoop distribution: Amazon 2.6.0
Applications: Hive 1.0.0, Mahout 0.10.0, Pig 0.14.0, Spark 1.4.1
Log URI: s3://ucsc.hadoop.aws/logs/
EMRFS consistent view: Disabled

Network and Hardware

Availability zone: us-west-2a
Subnet ID: subnet-3d0d914a
Master: Running 1 m3.xlarge
Core: Running 2 m3.xlarge
Task: --

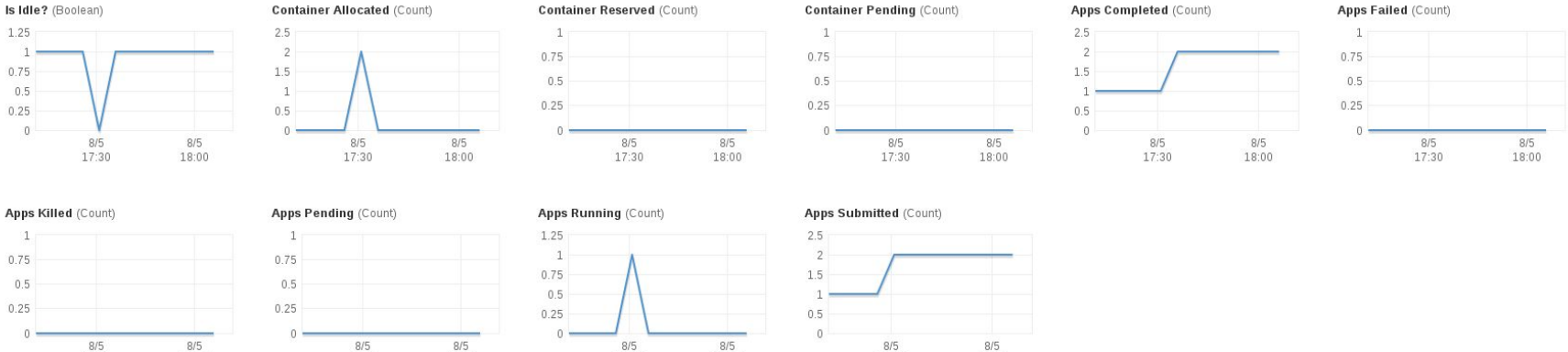
Security and Access

Key name: ucsc.hadoop.aws
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-d30adab7 (ElasticMapReduce-master)
Security groups for Core & Task: sg-dc0adab8 (ElasticMapReduce-slave)

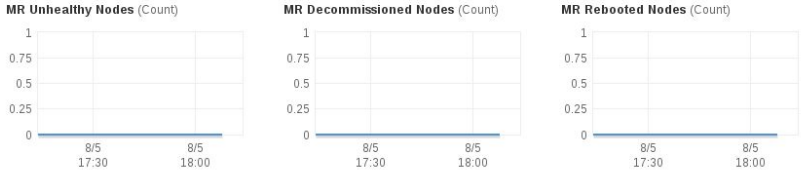
Monitoring

Graph size: Large | Start: 1 Hours Ago | End: 0 Hours Ago | Submit | All graphs are displayed in the UTC time zone.

Cluster Status | Node Status | IO



Metadata still available…



Hardware

Add task instance group

Instance Groups

Filter: Filter instance groups … 2 instance groups (all loaded)

ID	Name	Status	Type	Instance Type	Count	Bid Price	Actions
ig-1RTDMC643I3LC	Core Instance Group	Terminated	CORE	m3.xlarge	0 (2 Requested)		View EC2 instances
ig-35NODBUP41X7E	Master Instance Group	Terminated	MASTER	m3.xlarge	0 (1 Requested)		View EC2 instances

Steps

Add step | Clone step

Steps

View all interactive jobs | View all jobs

Filter: All steps Filter steps … 3 steps (all loaded)

ID	Name	Status	Start time (UTC-7)	Elapsed time	Log files	Actions
s-2X20TLV2VA408	Hive program	Completed	2015-08-05 10:28 (UTC-7)	58 seconds	View logs	View jobs
s-2DQ6NYJ6K21Y5	Hive program	Completed	2015-08-01 12:23 (UTC-7)	1 minute	View logs	View jobs
s-344DKO44M217K	Setup hadoop debugging	Completed	2015-08-01 12:23 (UTC-7)	3 seconds	View logs	View jobs

Terminated clusters

Amazon EMR preserves metadata information about completed clusters for two months.

- so even if you terminate the cluster, everything about the job persists

The console does not provide a way to delete completed clusters from the console; these are automatically removed after two months.

Step 8a. Go to S3 to see the output files

Open the s3 console: https://console.aws.amazon.com/s3

AWS Services Edit mia corey

Upload Create Folder Actions None Properties Tr

All Buckets / ucsc.hadoop.aws

Name	Storage Class	Size	Last Modified
logs	--	--	--
output	--	--	--
output_$folder$	Standard	0 bytes	Sat Aug 01 12:24:41 GMT-700 2015

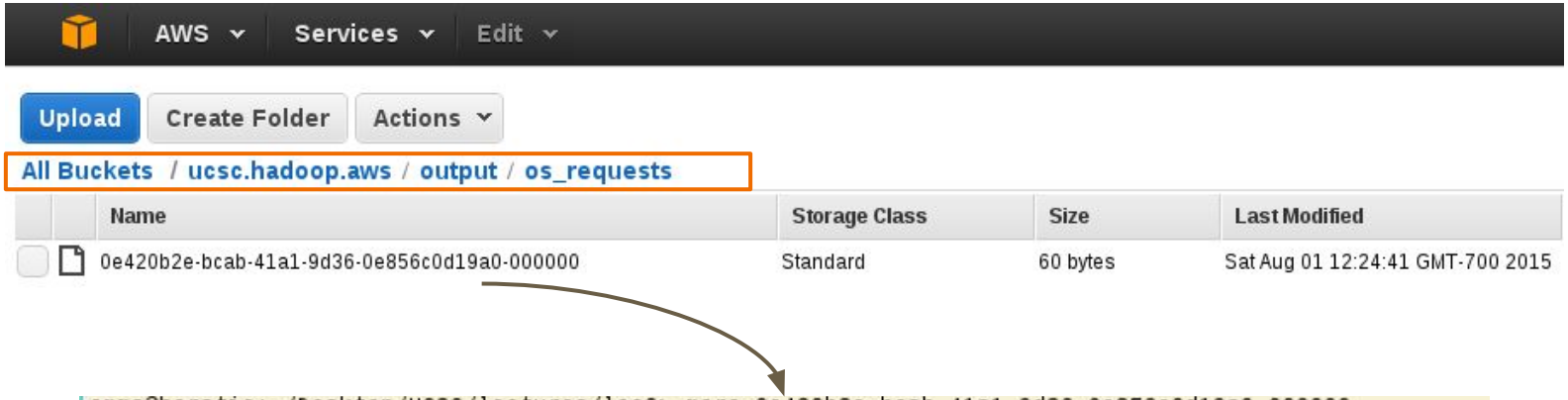
click on “output”

Bucket: ucsc.hadoop.aws

Bucket: ucsc.hadoop.aws
Region: Oregon
Creation Date: Sat Aug 01 11:11:42 GMT-700 2015
Owner: mcorey

- Permissions
- Static Website Hosting
- Logging
- Events
- Versioning
- Lifecycle
- Cross-Region Replication
- Tags
- Requester Pays

Step 8b. Drill down to the job output




```
emma@horatio:~/Desktop/UCSC/lectures/lec6> more 0e420b2e-bcab-41a1-9d36-0e856c0d19a0-000000
Android855
Linux813
MacOS852
OSX799
Windows883
iOS794
```

Step 8b. Drill down on aws-log-...-us-west-2

Drill down to the logs for the cluster (**j-3GKC7HOQ1GFUM**)

Upload | Create Folder | Actions

All Buckets / aws-logs-487740761633-us-west-2 / elasticmapreduce / j-3GKC7HOQ1GFUM

Name	Storage Class	Size	Last Modified
containers	--	--	--
em	--	--	--
hadoop-mapreduce	--	--	--
node	--	--	--
steps	--	--	--

What are they? Various type of logs...

Location	Description
hadoop-mapreduce/	Job logs and the configuration XML file for each Hadoop job.
node/	Node logs, including bootstrap action, instance state, and application logs for the node. The logs for each node are stored in a folder labeled with the identifier of the EC2 instance of that node.
steps/*N*/	Step logs that contain information about the processing of the step. The value of *N* indicates the stepId assigned by Amazon EMR. For example, a step has two stages: s-1234ABCDEFGH and s-5678IJKLMNOP. The first step is located in /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/ and the second step in /mnt/var/log/hadoop/steps/s-5678IJKLMNOP/. The step logs written by Amazon EMR are as follows. ● **controller** — Information about the processing of the step. If your step fails while loading, you can find the stack trace in this log. ● **syslog** — Describes the execution of Hadoop jobs in the step. ● **stderr** — The standard error channel of Hadoop while it processes the step. ● **stdout** — The standard output channel of Hadoop while it processes the step.

Log for checking the configuration

AWS Services Edit

Upload Create Folder Actions

All Buckets / ucsc.hadoop.aws / logs / j-WBF3SMVRL61C / hadoop-mapreduce / history / 2015 / 08 / 01 / 000000

Name	Storage Class	Size	Last Modified
job_1438456875247_0001-1438457036179-hadoop-INSERT+OVER...	Standard	3.9 KB	Sat Aug 01 12:25:35 GMT-700 2015
job_1438456875247_0001_conf.xml.gz	Standard	23.7 KB	Sat Aug 01 12:25:35 GMT-700 2015


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><configuration>
<property><name>dfs.block.access.token.lifetime</name><value>600</value><source>hdfs-default.xml</source><source>job.xml</source></property>
<property><name>hive.skewjoin.key</name><value>100000</value><source>programatically</source><source>org.apache.hadoop.hive.conf.LoopingByteArrayInputStream@7fdf417c</source><source>job.xml</source></property>
<property><name>hive.index.compact.binary.search</name><value>true</value><source>programatically</source><source>org.apache.hadoop.hive.conf.LoopingByteArrayInputStream@7fdf417c</source><source>job.xml</source></pr
<property><name>mapreduce.map.log.level</name><value>INFO</value><source>mapred-default.xml</source><source>job.xml</source></property>
<property><name>dfs.namenode.lazypersist.file.scrub.interval.sec</name><value>300</value><source>hdfs-default.xml</source><source>job.xml</source></property>
<property><name>mapreduce.admin.user.env</name><value>LD_LIBRARY_PATH=$HADOOP_COMMON_HOME/lib/native:/usr/lib/hadoop-lzo/lib/native</value><source>mapred-site.xml</source><source>job.xml</source></property>
<property><name>file.bytes-per-checksum</name><value>512</value><source>core-default.xml</source><source>job.xml</source></property>
<property><name>mapreduce.client.completion.pollinterval</name><value>5000</value><source>mapred-default.xml</source><source>job.xml</source></property>
<property><name>yarn.nodemanager.linux-container-executor.cgroups.strict-resource-usage</name><value>false</value><source>yarn-default.xml</source><source>job.xml</source></property>
<property><name>yarn.log-aggregation-enable</name><value>false</value><source>yarn-site.xml</source><source>job.xml</source></property>
<property><name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name><value>org.apache.hadoop.mapred.ShuffleHandler</value><source>yarn-site.xml</source><source>job.xml</source></property>
<property><name>dfs.namenode.edit.log.autoroll.check.interval.ms</name><value>300000</value><source>hdfs-default.xml</source><source>job.xml</source></property>
<property><name>ipc.client.fallback-to-simple-auth-allowed</name><value>false</value><source>core-default.xml</source><source>job.xml</source></property>
<property><name>dfs.client.failover.connection.retries</name><value>0</value><source>hdfs-default.xml</source><source>job.xml</source></property>
<property><name>mapreduce.jobtracker.system.dir</name><value>${hadoop.tmp.dir}/mapred/system</value><source>mapred-default.xml</source><source>job.xml</source></property>
<property><name>yarn.scheduler.minimum-allocation-mb</name><value>256</value><source>yarn-site.xml</source><source>job.xml</source></property>
<property><name>mapreduce.task.profile.map.params</name><value>${mapreduce.task.profile.params}</value><source>mapred-default.xml</source><source>job.xml</source></property>
<property><name>mapreduce.map.memory.mb</name><value>1440</value><source>mapred-site.xml</source><source>job.xml</source></property>
<property><name>mapreduce.tasktracker.dns.interface</name><value>default</value><source>mapred-default.xml</source><source>job.xml</source></property>
<property><name>dfs.datanode.failed.volumes.tolerated</name><value>0</value><source>hdfs-default.xml</source><source>job.xml</source></property>
<property><name>yarn.label.enabled</name><value>true</value><source>yarn-site.xml</source><source>job.xml</source></property>
<property><name>hive.server2.authentication</name><value>NONE</value><source>programatically</source><source>org.apache.hadoop.hive.conf.LoopingByteArrayInputStream@7fdf417c</source><source>job.xml</source></propert
<property><name>_hive.tmp_table_space</name><value>/tmp/hive/hadoop/86cbc455-1812-445a-956d-3d5bf811790f/_tmp_space.db</value><source>programatically</source><source>job.xml</source></property>
<property><name>stream.stderr.reporter.prefix</name><value>reporter:</value><source>programatically</source><source>org.apache.hadoop.hive.conf.LoopingByteArrayInputStream@7fdf417c</source><source>job.xml</source></
<property><name>dfs.client.slow.io.warning.threshold.ms</name><value>30000</value><source>hdfs-default.xml</source><source>job.xml</source></property>
<property><name>hadoop.security.groups.cache.secs</name><value>300</value><source>core-default.xml</source><source>job.xml</source></property>
<property><name>yarn.nodemanager.env-whitelist</name><value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,HADOOP_YARN_HOME</value><source>yarn-default.xml</source><source>job.xml</source></property>
<property><name>hive.metastore.authorization.storage.checks</name><value>false</value><source>programatically</source><source>org.apache.hadoop.hive.conf.LoopingByteArrayInputStream@7fdf417c</source><source>job.xml<
<property><name>map.sort.class</name><value>org.apache.hadoop.util.QuickSort</value><source>mapred-default.xml</source><source>job.xml</source></property>
<property><name>dfs.namenode.safemode.threshold-pct</name><value>0.999f</value><source>hdfs-default.xml</source><source>job.xml</source></property>
<property><name>mapreduce.jobtracker.jobhistory.task.numberprogresssplits</name><value>12</value><source>mapred-default.xml</source><source>job.xml</source></property>
<property><name>datanucleus.storeManagerType</name><value>rdbms</value><source>programatically</source><source>org.apache.hadoop.hive.conf.LoopingByteArrayInputStream@7fdf417c</source><source>job.xml</source></prope
<property><name>dfs.short.circuit.shared.memory.watcher.interrupt.check.ms</name><value>60000</value><source>hdfs-default.xml</source><source>job.xml</source></property>
```

Another view of the syserr log

https://s3-us-west-2.amazonaws.com/aws-logs-487740761633-us-west-2/elasticmapreduce/j-3GKC7HOQ1GFUM/steps/s-SF8MCNMDK7TH/stderr.gz

Apps | Bookmarks | amazon EMR | java | blogs | personal | sysadmin | UCSC lectures | UCSC websites | todo | summer2016 | Fall2016 | bio da

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
OK
Time taken: 8.009 seconds
Query ID = hadoop_20161205232424_8cdf7e58-e416-49d8-8872-cac04eb4269c
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1480979756851_0001, Tracking URL = http://ip-172-31-34-7.us-west-2.compute.inter
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1480979756851_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2016-12-05 23:25:23,334 Stage-1 map = 0%,  reduce = 0%
2016-12-05 23:25:51,561 Stage-1 map = 13%,  reduce = 0%, Cumulative CPU 13.87 sec
2016-12-05 23:25:54,788 Stage-1 map = 27%,  reduce = 0%, Cumulative CPU 16.77 sec
2016-12-05 23:25:57,971 Stage-1 map = 33%,  reduce = 0%, Cumulative CPU 19.7 sec
2016-12-05 23:26:01,210 Stage-1 map = 40%,  reduce = 0%, Cumulative CPU 22.47 sec
2016-12-05 23:26:06,699 Stage-1 map = 53%,  reduce = 0%, Cumulative CPU 28.32 sec
2016-12-05 23:26:11,113 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 31.9 sec
2016-12-05 23:26:28,208 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 37.97 sec
MapReduce Total cumulative CPU time: 37 seconds 970 msec
Ended Job = job_1480979756851_0001
Moving data to: s3://ucsc.hadoop.aws/output/os_requests
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 37.97 sec   HDFS Read: 599 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 37 seconds 970 msec
OK
Time taken: 106.927 seconds
Command exiting with ret '0'
```

Drilling down in S3:

aws-logs-4877407671633
->elasticmapreduce
-> j-3GKC7HOQ1GFUM
-> steps
-> s-SF8MCNMDK7TH
-> stderr

This is the same log here

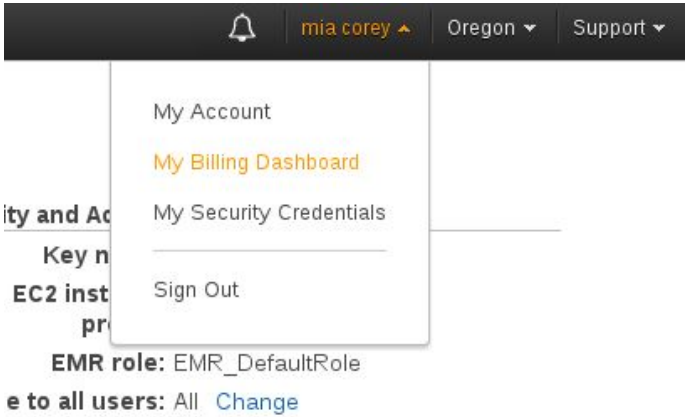
Misc: Downloads and job metadata

You can download *everything* in your S3 bucket

- simply double-click on the output files or logs

Step 9: Again, stop spending money

- Delete the S3 bucket at https://console.aws.amazon.com/s3
  - You cannot delete an Amazon S3 bucket that has items in it.
  - First, delete the logs and output folders, and then delete your bucket.
- Check your bill:



Step 9: REALLY - stop spending money

- Keeping two M3Large cluster and output data *idle* for 30 days.
  - **Total cost: $775**
- Cost for running the example (1 minute compute time) and cleaning up right away:
  - **Total cost: $3**

Pricing on AWS: https://aws.amazon.com/emr/pricing/

	Amazon EC2 Price	Amazon EMR Price
General Purpose - Current Generation		
m3.xlarge	$0.266 per Hour	$0.070 per Hour
m3.2xlarge	$0.532 per Hour	$0.140 per Hour
m4.large	$0.12 per Hour	$0.030 per Hour
m4.xlarge	$0.239 per Hour	$0.060 per Hour
m4.2xlarge	$0.479 per Hour	$0.120 per Hour
m4.4xlarge	$0.958 per Hour	$0.240 per Hour
m4.10xlarge	$2.394 per Hour	$0.270 per Hour
General Purpose - Previous Generation		
m1.small	$0.044 per Hour	$0.011 per Hour
m1.medium	$0.087 per Hour	$0.022 per Hour
m1.large	$0.175 per Hour	$0.044 per Hour
m1.xlarge	$0.350 per Hour	$0.088 per Hour

lowest price available today: 12/5/2016.

30 minutes to spin up.

Background: comments on the data

Sample data from Amazon CloudFront web distribution log files.

- The data is stored in Amazon S3 at **s3://*us-west-2*.elasticmapreduce.samples**
  - *us-west-2* is my region (I used the default)
- If you use a different region, the sample data is under that region.

Additional data at **http://aws.amazon.com/public-data-sets/**

- web crawl data
- genomic data
- ngrams (word co-occurrences) from Google books
- usenet data (news group data, anonymized)
- Fred (economic data)

and more...

Background: the Hive script

Used a Hive script to calculate the number of requests per OS in a given timeframe.

The script is stored in Amazon S3 at

s3://*us-west-2*.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q

where *us-west-2* is my region.

To access or download via HTTP:

us-west-2.elasticmapreduce.samples.s3.amazonaws.com/cloudfront/code/Hive_CloudFront.q

Important consoles

S3 Management: https://console.aws.amazon.com/s3/

- input data
- output results
- logs

EC2 Management: https://console.aws.amazon.com/ec2/

Cluster view: https://console.aws.amazon.com/elasticmapreduce/

- Launching a cluster
- Launching a job
- Viewing job metadata
- Terminating a cluster

Advanced

Connecting to the master node:

http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-connect-master-node-ssh.html