

Running Image Analysis on Amazon Elastic Mapreduce

Required software

- Amazon EMR CLI: <http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-cli-install.html>
- Amazon S3 Client for file manipulation and/or transfer: s3cmd <http://s3tools.org/s3cmd> or S3Browser (<http://s3browser.com/>)

Source code compilation

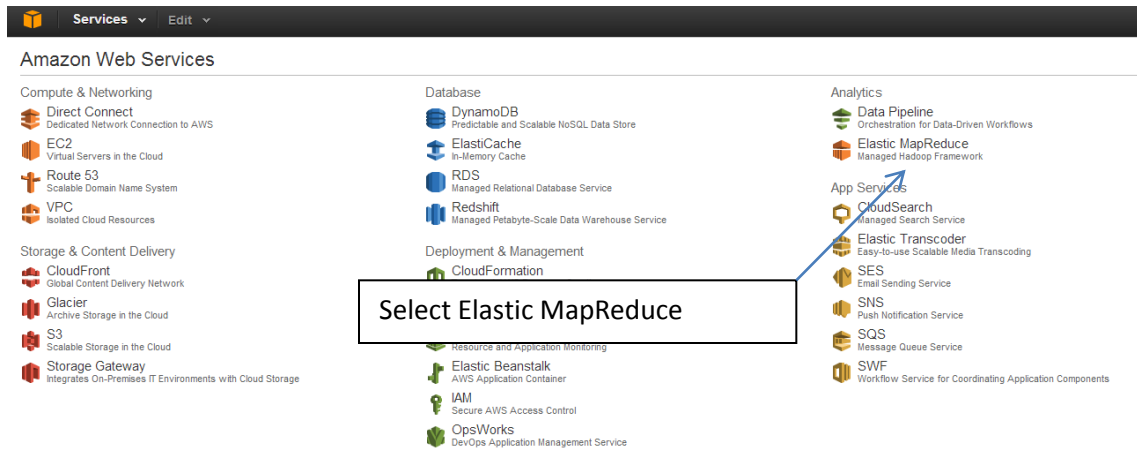
- 1) Amazon EMR CLI:
Create a cluster instance of EMR (EC2) and login into it using ssh via the AWS EMR CLI interface.
This can be done in shell:
ruby ./elastic-mapreduce --create --alive --name "Compiling/editing purposes" --num-instances=1 --master-instance-type=m1.medium
*** We use m1.medium, since m1.small instances has a 32-bit OS.
The output is a jobflow ID:
Login via CLI:
ruby ./elastic-mapreduce --ssh --jobflow jobflow_id
Example:
./elastic-mapreduce --ssh --jobflow j-3KNI2HN0YYQFO
The user will log in as a hadoop user.
Files on Amazon S3 can be accessed using hadoop command line interface.
- 2) Required libraries needed to modify and compile source code for LibHadoopGIS

OpenCV (2.4.5) (You can use the installcv shell script bootstrapping.sh – copy from s3://cciememory/bootstrap/)

Boost (1.48.0)
- 3) Compile programs and test on Amazon cluster.
Edit CMakeLists.txt
Run cmake .
Then
- 4) Upload the data back to Amazon S3 using hadoop fs commands.

Running from Web Interface

- 1) Login to the AWS Console Home:



2) Select Create Cluster:

***** Make sure to select the region on the top right to be N. Virginia (U.S. East), not the Oregon (U.S. West) *****

	Name	ID	Status	Creation time local time (UTC-5)	Elapsed time	Norm insta
<input type="checkbox"/>	My Development Jobflow	j-3856DT33K1TVR	Waiting	2013-12-08 21:16:37	3 hours, 12 minutes	8
<input type="checkbox"/>	Word count	j-1NPHK2UBUEEG	Terminated All steps completed	2013-12-08 19:30:54	9 minutes	3

3) Enter a cluster name and location for log files:

Cluster Configuration

Cluster name:

Termination protection: ☒ Yes ☐ No

Logging: ☒ Enabled

Log folder S3 location:

Debugging: ☒ Enabled

Tags (optional)

Add up to 10 tags to your EMR cluster. A tag consists of a case-sensitive key-value pair. Tags on EMR clusters are propagated to the underlying EC2 instances. [Learn more](#) about tagging your Amazon EMR clusters.

Key (required)	Value (optional)
<input type="text" value="Add a key to create a tag"/>	<input type="text"/>

Software Configuration

Hadoop distribution: ☒ Amazon [Learn more](#)

Example:

Cluster Configuration

Cluster name: My cluster

Termination protection: ☒ Yes ☐ No

Logging: ☒ Enabled

Log folder S3 location: s3://cciememorylog/

Debugging: ☒ Enabled

Tags (optional)

Software Configuration

Hadoop distribution: ☒ Amazon

- 4) Select the compatible AMI version (corresponding to different version of Hadoop).
Currently all versions should be compatible with LibHadoopGIS.
- 5) Remove Hive and Pig installation as we do not require them.

Software Configuration

Hadoop distribution: ☒ Amazon

AMI version: 2.4.2 (Hadoop 1.0.3) - latest

Applications to be installed

Applications to be installed	Version
Hive	0.11.0.1
Pig	0.11.1.1

Additional applications: Select an application

- 6) Select Hardware Configuration matching your preference:

Hardware Configuration

Network: vpc-d47362b6 (172.31.0.0/16) (default)

EC2 Subnet: No preference (random subnet)

EC2 instance type

	EC2 instance type	Count	Request spot
Master	m1.medium	1	<input type="checkbox"/>
Core	m1.medium	10	<input type="checkbox"/>
Task	m1.medium	0	<input type="checkbox"/>

** The description of EC2 instances can be found on <http://aws.amazon.com/ec2/instance-types/instance-details/> .

** The default settings for the numbers of mappers and reducers can be found on <http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/TaskConfiguration.html>

** Be advised that while you can pick any number of reducers, the maximum CPU cores available for computing might not be less than the number of reducers. In addition, the amount of memory available for mapper jobs will decrease as the number of reducers increases while the number of core and task instances does not change.

7) Select a bootstrap script:

Bootstrap Actions

i Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
-----------------------	------	-------------	--------------------

Add bootstrap action Custom action

Configure and add

Example: choose Custom action -> Configure and add:

Enter **s3://cciemory/bootstrap/copyopencvlib.sh**

Add Bootstrap Action

Bootstrap action type: Custom action

Name: Custom action

S3 location: s3://cciemory/bootstrap/bootcopygeospatial.sh

Optional arguments:

Cancel **Add**

8) Select Custom JAR Select steps: Click on Configure and add:

Steps

i A step is a unit of work you submit to the cluster. A step might contain one or more Hadoop jobs, or contain instructions to install or configure an application. You can submit up to 256 steps to a cluster. [Learn more](#)

Name	Action on failure	JAR S3 location	Arguments
------	-------------------	-----------------	-----------

Add step Custom JAR

Configure and add

Auto-terminate ☐ Yes ☒ No

Automatically terminate cluster after the last step is completed.

Keep cluster running until you terminate it.

Cancel **Create cluster**

9) Enter the locations of mappers, reducers, input and output directory, as well as other arguments.

Mapper: S3 location of mapper file

Reducer (need to enter parameters): S3 location of reducer file.

Input location: valid input location on Amazon s3.

Output location: The directory of the output should not exist on S3. It will be created by the EMR job.

Arguments: Specify the number of reduce tasks and other options as needed.

Custom jar: The location of the streaming jar file.

Add Step [X]

Step type: Custom JAR

Name*: Custom JAR

JAR S3 location*: s3://cciemory/program/hadoop-0.20-streaming.jar

Arguments: `-mapper s3://cciemory/program/mapperImageSegment -reducer s3://cciemory/program/simpleRead2 -input s3://cciemory/moretiles/ -output s3://cciemory/moretilesOutput/ -inputformat org.apache.hadoop.mapred.WholeFileInputFormat -inputreader org.apache.hadoop.mapred.WholeFileRecordReader -numReduceTasks 10 -cmdenv imagewidth=4096 -cmdenv imageheight=4096 -cmdenv imagebuffer=0`

Action on failure: Continue

What to do if the step fails.

Cancel Add

Example:

Custom jar: *s3://cciemory/program/hadoop-0.20-streaming.jar* (Note that this is the streaming file that we updated by adding different mapreduce support input format and record readers to the original *hadoop-0.20-streaming.jar*)

Mapper: *s3://cciemory/program/mapperImageSegment*

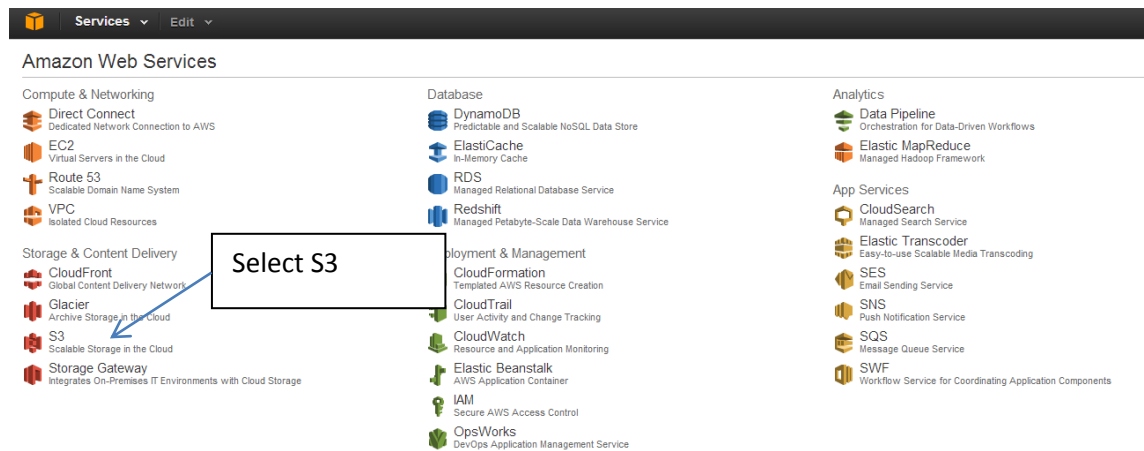
Reducer: *s3://cciemory/program/reducerImage*

Arguments: *-mapper s3://cciemory/program/mapperImageSegment -reducer s3://cciemory/program/simpleRead2 -input s3://cciemory/moretiles/ -output s3://cciemory/moretilesOutput/ -inputformat org.apache.hadoop.mapred.WholeFileInputFormat -inputreader org.apache.hadoop.mapred.WholeFileRecordReader -numReduceTasks 10 -cmdenv imagewidth=4096 -cmdenv imageheight=4096 -cmdenv imagebuffer=0*

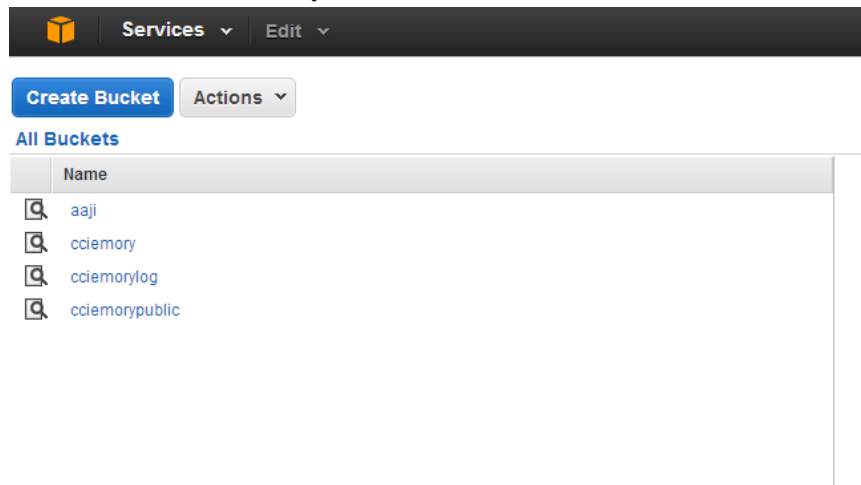
Or

-mapper s3://cciemory/program/mapperImageSegment -reducer s3://cciemory/program/simpleRead2 -input s3://cciemory/moretiles/ -output s3://cciemory/moretilesOutput/ -inputformat org.apache.hadoop.mapred.WholeFileInputFormat -inputreader org.apache.hadoop.mapred.WholeFileRecordReader -numReduceTasks 10 -cmdenv imagewidth=4300 -cmdenv imageheight=4300 -cmdenv imagebuffer=102

Result can be found on Amazon S3. Amazon S3 can be accessed from the Console Home:



Select the bucket cciememory:



Note:

This can be run from Command Line Interface (AWS CLI) using the arguments provided above.