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

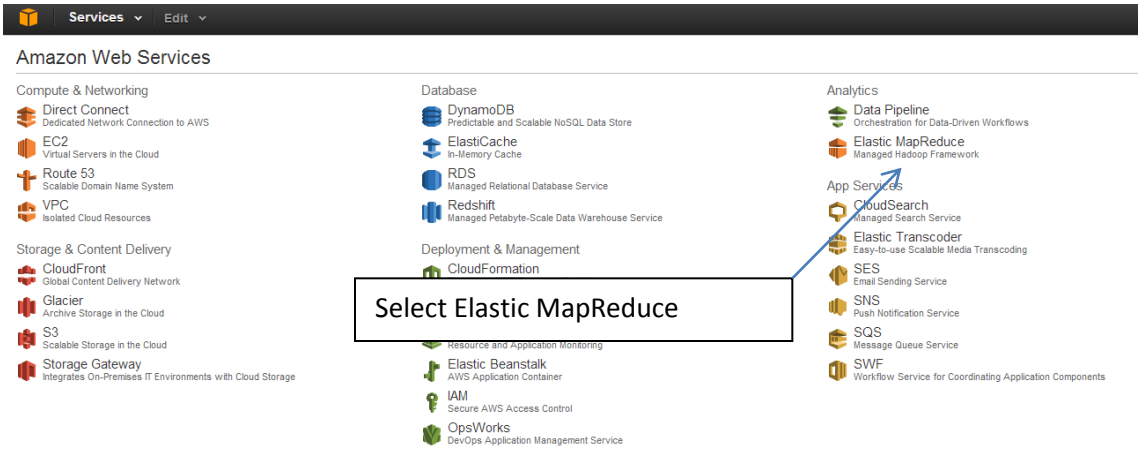
   OpenCV (2.4.5) (You can use the installcv shell script bootstrapimg.sh – copy from s3://cciemory/bootstrap/)

   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.

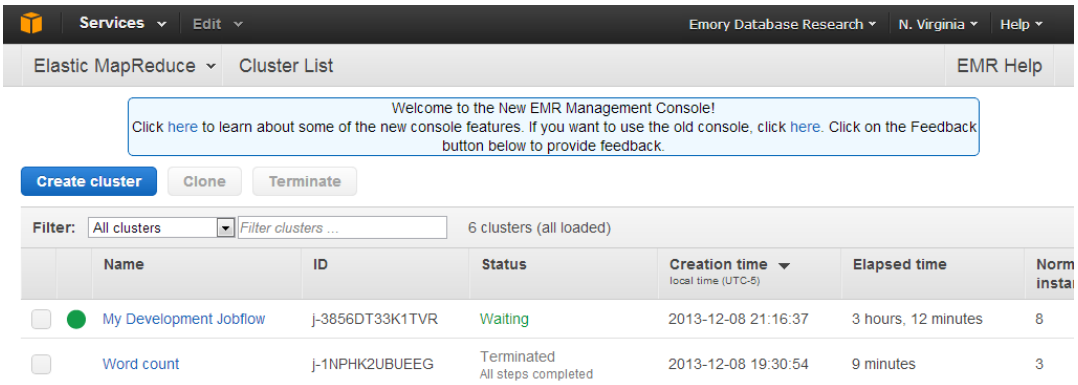
Running from Web Interface

1) Login to the AWS Console Home:

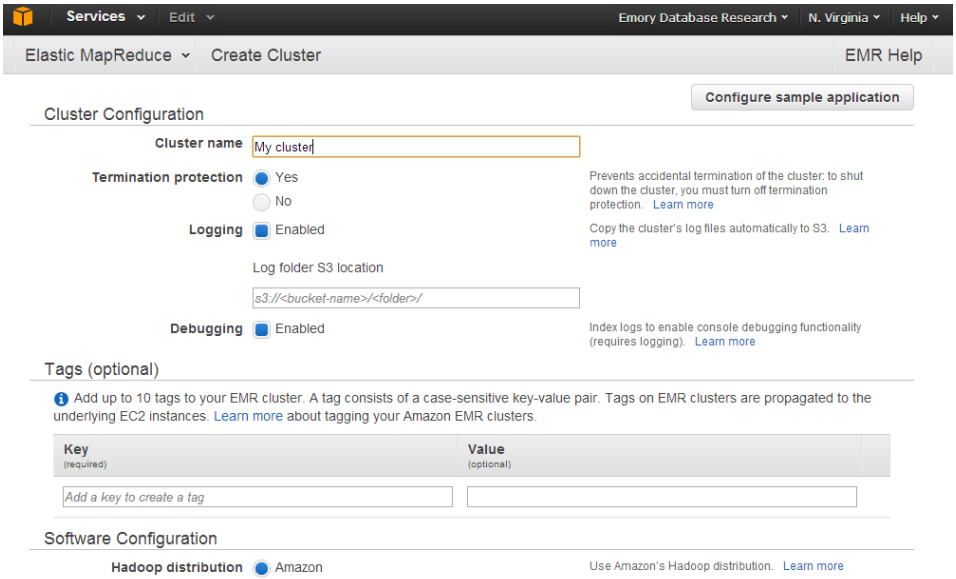


2) Select **Create Cluster**:

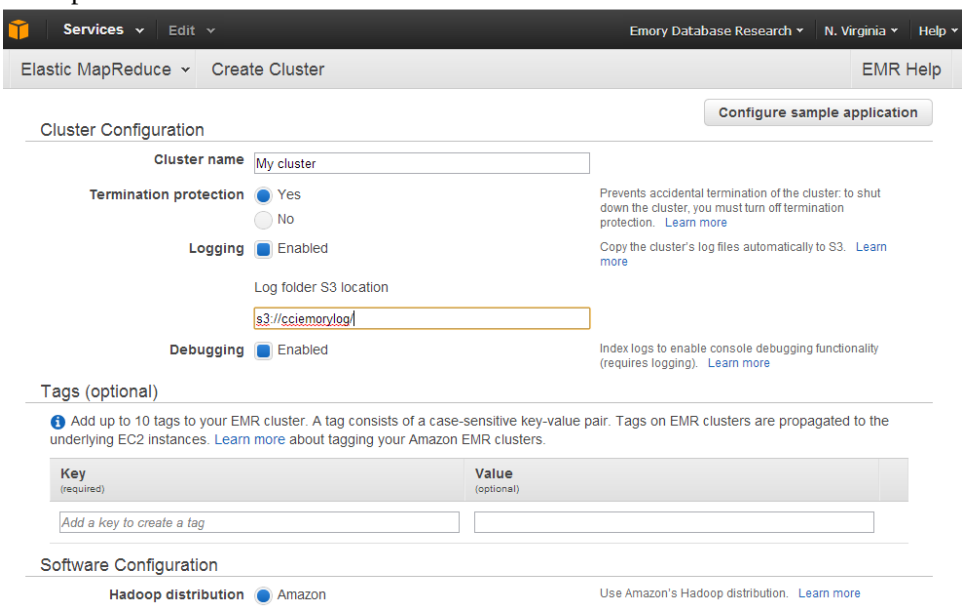
****** Make sure to select the region on the top right to be N. Virginia (U.S. East), not the Oregon (U.S. West) ****8



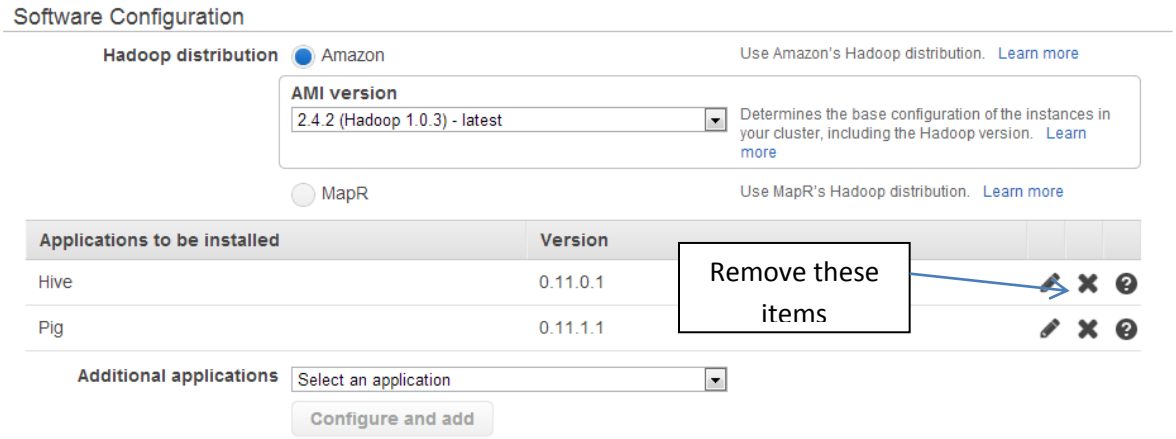
3) Enter a cluster name and location for log files:



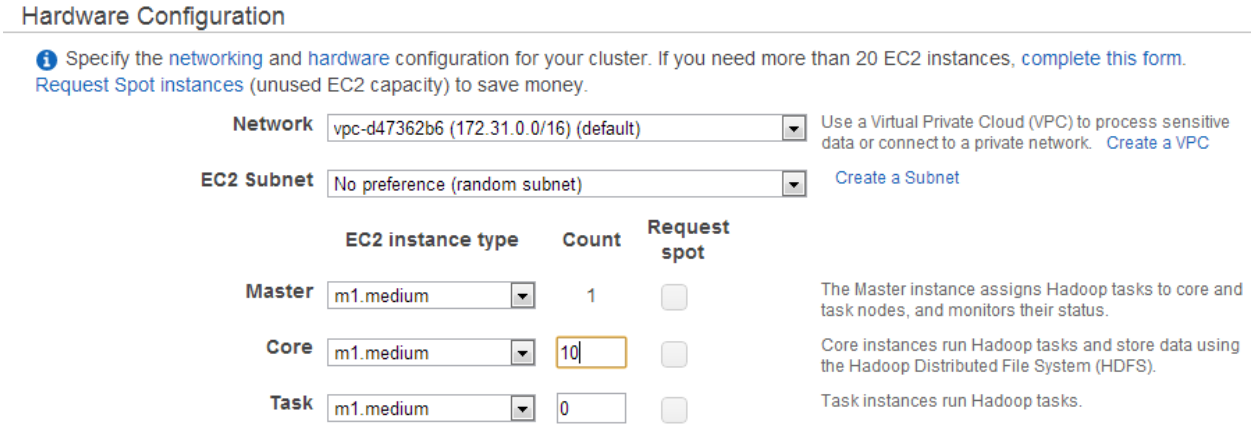
Example:



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.

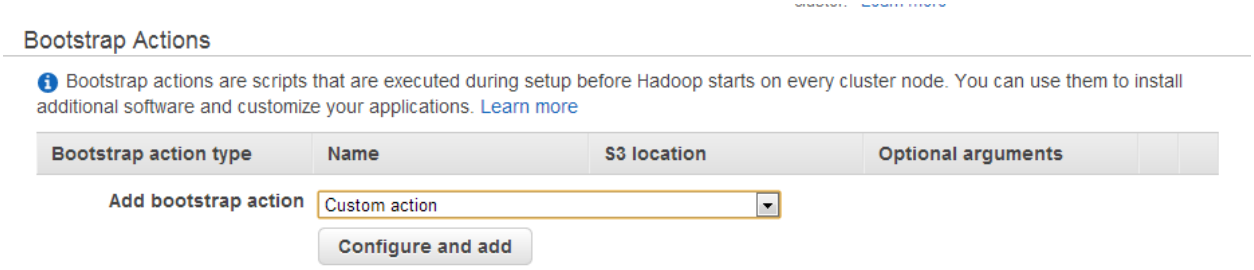


6) Select Hardware Configuration matching your preference:

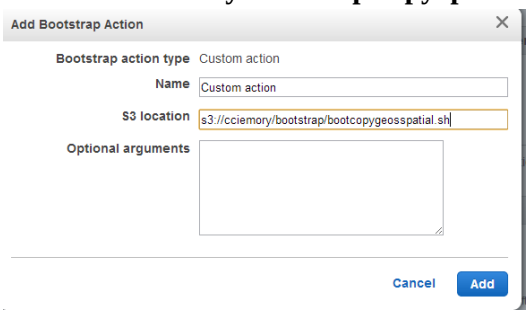


** The description of EC2 instances can be found on http://aws.amazon.com/ec2/instance-types/instance-details/ .
** The default settings for the numbers of mappers and reducers can be found on http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/TaskConfiguration.html
** Be advised that 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.

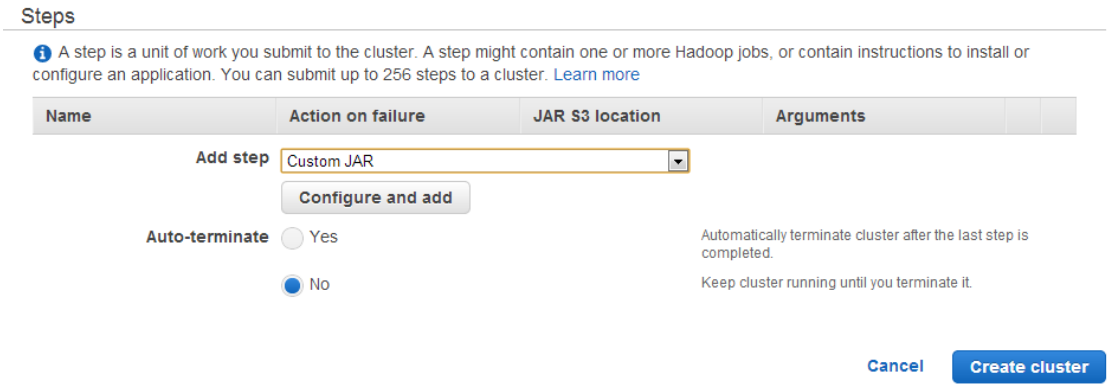
7) Select a boostrap script:



Example: choose Custom action -> Configure and add:
Enter **s3://cciemory/bootstrap/copyopencvlib.sh**

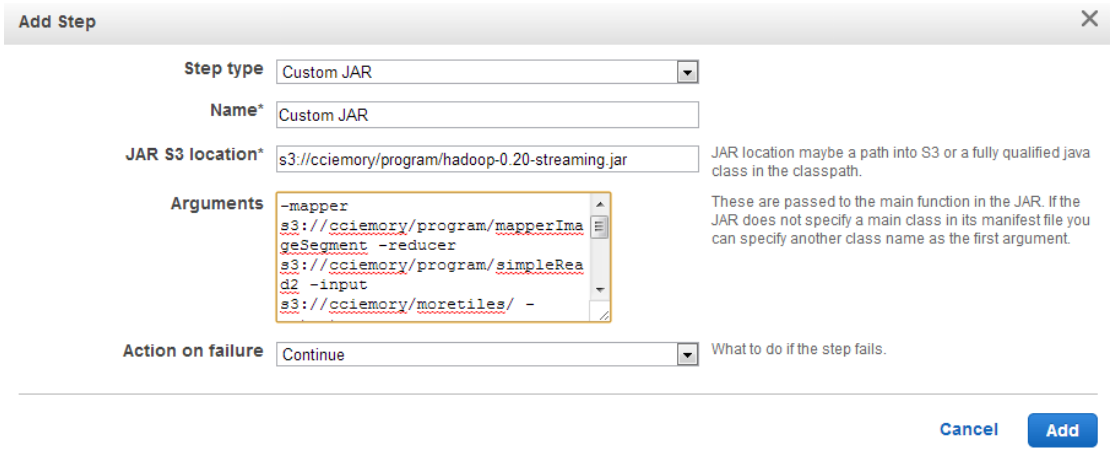


8) Select Custom JAR Select steps: Click on Configure and add:



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.

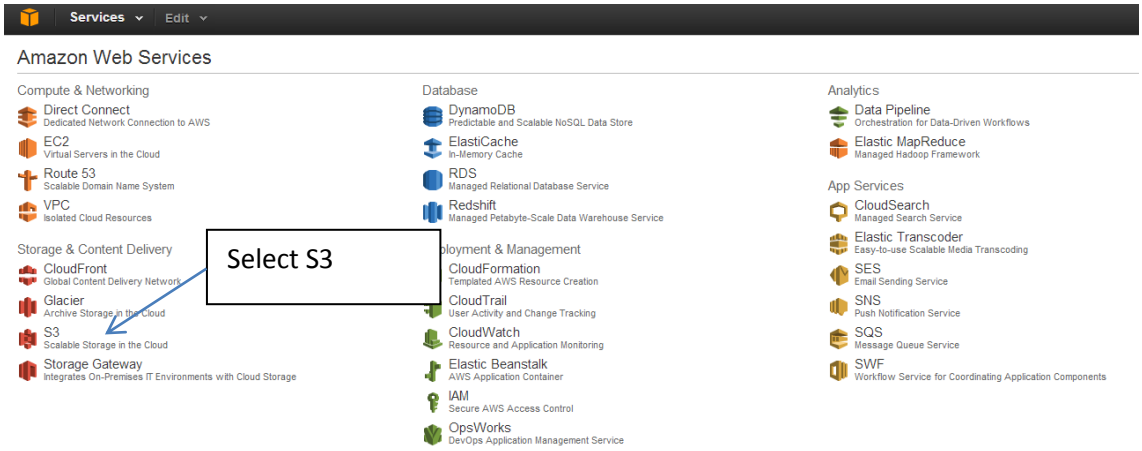


Example:
Custom jar: *s3://cciemory/program/hadoop-0.20-streaming.jar (Note that this is the streaming file that we updated by adding different mapreduce support input format and record readers to the original hadoop-0.20-streaming.jar)*

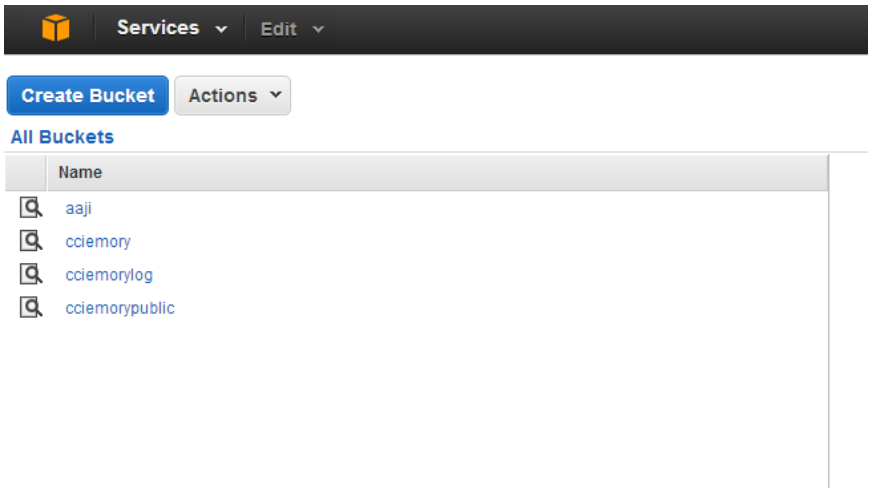
Mapper: *s3://cciemory/program/mapperImageSegment*
Reducer: *s3://cciemory/program/reducerImage*
Arguments: *-mapper s3://cciemory/program/mapperImageSegment -reducer s3://cciemory/program/simpleRead2 -input s3://cciemory/moretiles/ -output s3://cciemory/moretilesOutput/ -inputformat org.apache.hadoop.mapred.WholeFileInputFormat -inputreader org.apache.hadoop.mapred.WholeFileRecordReader -numReduceTasks 10 -cmdenv imagewidth=4096 -cmdenv imageheigh=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 imageheigh=4300-cmdenv imagebuffer=102

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



Select the bucket cciemory:



Note:

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