

# AmazonEMRIntro

## Introduction

Amazon EMR provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances.

You can run popular distributed frameworks such as [Apache Spark](#), [HBase](#), [Presto](#), and [Flink](#) in Amazon EMR, and interact with data in other AWS (Amazon Web Services) data stores such as Amazon S3 and Amazon DynamoDB.

Amazon EMR securely and reliably handles a broad set of big data use cases, including log analysis, web indexing, data transformations (ETL), machine learning, financial analysis, scientific simulation, and bioinformatics.

References:

Amazon Elastic MapReduce

<https://aws.amazon.com/emr/>

Amazon EMR Management Guide

<http://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html>

Amazon Elastic MapReduce API

<http://docs.aws.amazon.com/ElasticMapReduce/latest/API/Welcome.html>

Amazon EMR Release Guide

<http://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hadoop-application.html>

Apache Hadoop

<http://hadoop.apache.org/>

Programming Elastic MapReduce Using AWS Services to Build an End-to-End Application, Kevin Schmidt, Christopher Phillips, O'Reilly Media, December 2013.

<http://shop.oreilly.com/product/0636920029304.do>

## Key components

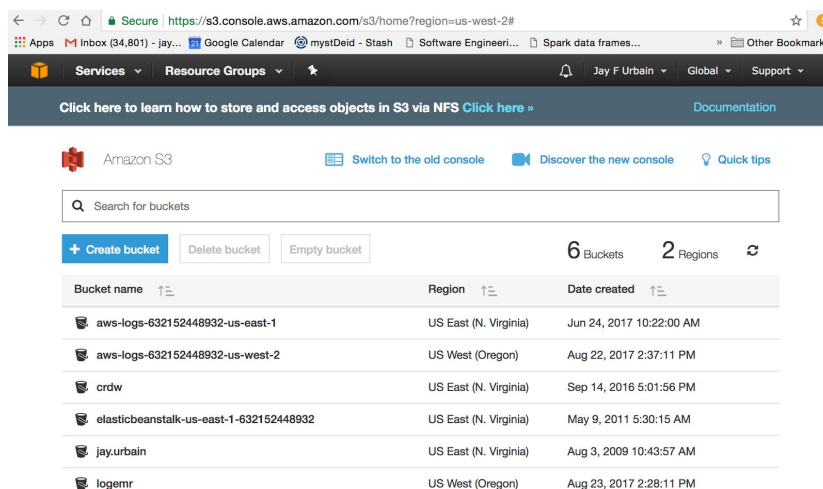
### *Amazon Elastic MapReduce (EMR)*

Amazon EMR is an in-the-cloud platform of the Hadoop framework. Amazon EMR makes heavy use of the Amazon Simple Storage Service (S3) to store analysis results and host data sets for processing, and leverages Amazon Elastic Cloud Compute (EC2) resources to run applications.

There is an additional charge of about 30 percent for the EMR EC2 instances. To read Amazon's overview of EMR, visit the [Amazon EMR web page](#).

### *Amazon Simple Storage Service (S3)*

Amazon S3 is the persistent storage for AWS. It provides a web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the Web. There are some restrictions. Data in S3 must be stored in named buckets, and any single object can be no more than 5 terabytes in size. The data stored in S3 is highly durable and is stored in multiple facilities and multiple devices within a facility.



There are standard REST- and SOAP-based web service APIs to interact with files stored on S3.

Amazon S3's permanent storage will be used to store data sets and computed result sets generated by Amazon EMR Job Flows. Applications built with Amazon EMR need to use some S3 services for data storage.

## ***Amazon Elastic Compute Cloud (EC2)***

Amazon EC2 makes it possible to run multiple instances of virtual machines on demand inside any one of the AWS regions. You can start as many or as few instances as you need without having to buy or rent physical hardware like in traditional hosting services. In the case of Amazon EMR, this means we can scale the size of your Hadoop cluster to any size needed. Individual EC2 instances come in a variety of sizes and specifications to meet the needs of different types of applications. There are instances tailored for high CPU load, high memory, high I/O, etc.

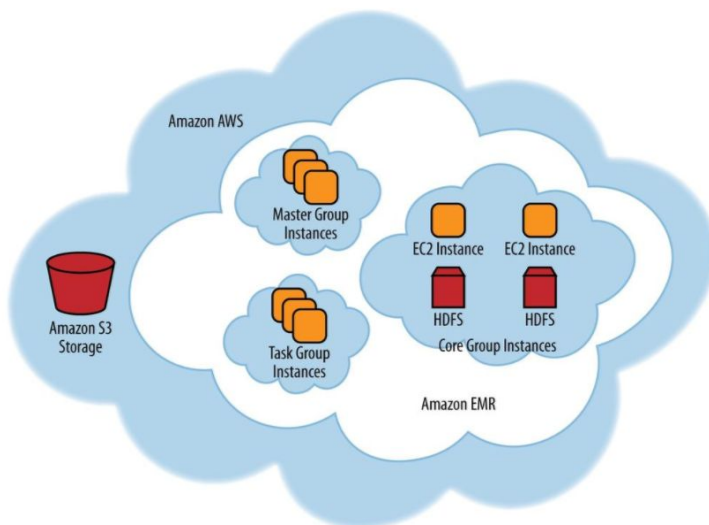
To read Amazon's overview of EC2, visit the [Amazon EC2 web page](#). Amazon EC2 instances are used as part of an Amazon EMR cluster.

## Amazon Elastic MapReduce Architecture

Amazon EMR allows users to launch and use resizable Hadoop clusters inside of Amazon's infrastructure. Amazon EMR, like Hadoop, can be used to analyze large data sets. EMR greatly simplifies the setup and management of the cluster of Hadoop and MapReduce components.

EMR instances use Amazon's pre-built and customized EC2 instances, which can take full advantage of Amazon's infrastructure and other AWS services. These EC2 instances are invoked when a new Job Flow is started to form an EMR cluster. A *Job Flow* is Amazon's term for the complete data processing that occurs through a number of compute steps in Amazon EMR. A Job Flow is specified by the MapReduce application and its input and output parameters.

<https://aws.amazon.com/emr/>



Amazon EMR performs the computational analysis using the MapReduce framework. The MapReduce framework splits the input data into smaller fragments, or shards, that are distributed to the nodes that compose the cluster



### ***Master group instance***

The master group instance manages the Job Flow and allocates all the needed executables, JARs, scripts, and data shards to the core and task instances. The master node monitors the health and status of the core and task instances and also collects the data from these instances and writes it back to Amazon S3.

The master group instances serve a critical function in our Amazon EMR cluster. If a master node is lost, you lose the work in progress by the master *and* the core and task nodes to which it had delegated work.

### ***Core group instance***

Core group instance members run the map and reduce portions of our Job Flow, and store intermediate data to the Hadoop Distributed File System (HDFS) storage in our Amazon EMR cluster.

The master node manages the tasks and data delegated to the core and task nodes. Due to the HDFS storage aspects of core nodes, a loss of a core node will result in data loss and possible failure of the complete Job Flow.

### ***Task group instance***

The task group is optional. It can do some of the dirty computational work of the map and reduce jobs, but does not have HDFS storage of the data and intermediate results.

The lack of HDFS storage on these instances means the data needs to be transferred to these nodes by the master for the task group to do the work in the Job Flow.

## **Amazon EMR and the Hadoop Ecosystem**

Amazon EMR uses Hadoop and its MapReduce framework at its core. Accordingly, many of the other core Apache Software Foundation projects that work with Hadoop also work with Amazon EMR. There are also many other AWS services that may be useful when you're running and monitoring Amazon EMR applications. For example:

### ***Hive***

Hive is a distributed data warehouse that allows you to create a Job Flow using a SQL-like language. Hive can be run from a script loaded in S3 or interactively inside of a running EMR instance.

### ***Pig***

Pig is a data flow language. The language is called *Pig Latin*. Pig scripts can be loaded into S3 and used to perform the data analysis in a Job Flow. Pig, like Hive, is one of the Job Flow types that can be run interactively inside of a running EMR instance.

### ***Amazon Cloudwatch***

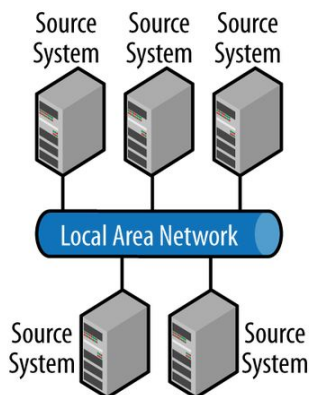
Cloudwatch allows you to monitor the health and progress of Job Flows. It also allows you to set alarms when metrics are outside of normal execution parameters.

## Amazon Elastic MapReduce Versus Traditional Hadoop Installs

Amazon EMR uses S3 storage for the input and output of data sets to be processed and analyzed. In order to process data, you need to transport it to Amazon's cloud into S3 buckets.

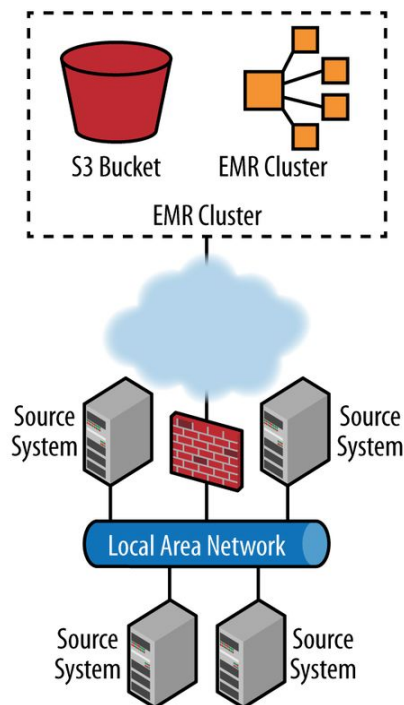
In the traditional Hadoop install, data transport between the current source locations and the Hadoop cluster may be colocated in the same data center on high-speed internal networks. This lowers the data transport barriers and the amount of time to get data into Hadoop for analysis.

### Traditional Hadoop Installation



Traditional Hadoop installation with data transferred to Hadoop over dedicated low latency local area network.

### Amazon EMR Installation



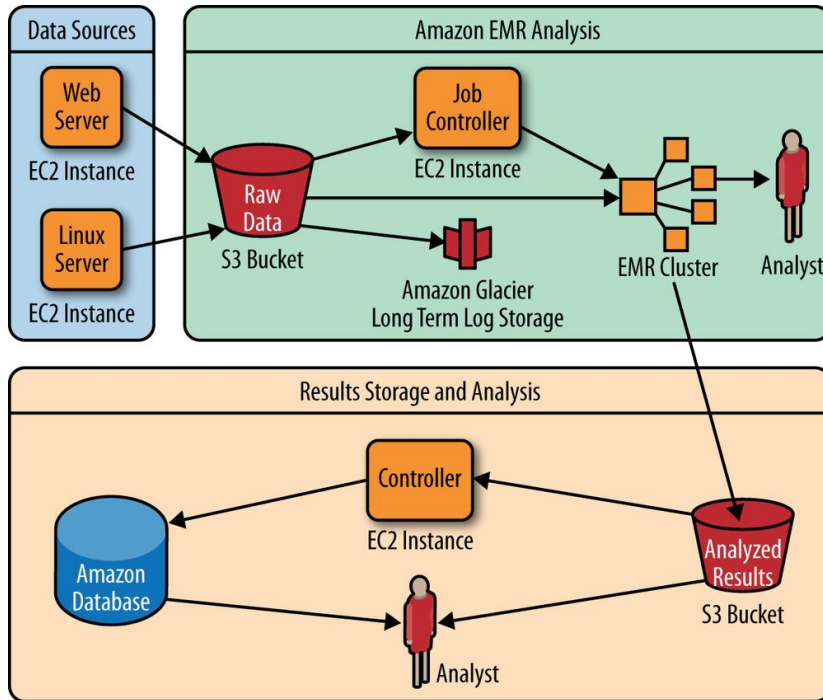
Amazon EMR installation with data transferred across higher latency internet and competing with traffic for delivery.

*Note: Amazon has S3 Import and Export services.*



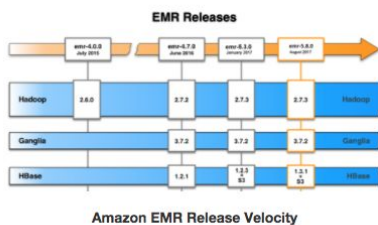
## Application Building Blocks

- ingest large volumes of log data,
- perform real-time and
- batch analysis, and ultimately produce results that can be shared with end users.



## Application Integration

### Use Your Favorite Open Source Applications



With versioned releases on Amazon EMR, you can easily select and use the latest open source projects on your EMR cluster, including applications in the Apache Hadoop and Spark ecosystems. Software is installed and configured by Amazon EMR, so you can spend more time on increasing the value of your data without worrying about infrastructure and administrative tasks.

Apache Hadoop

Apache Spark

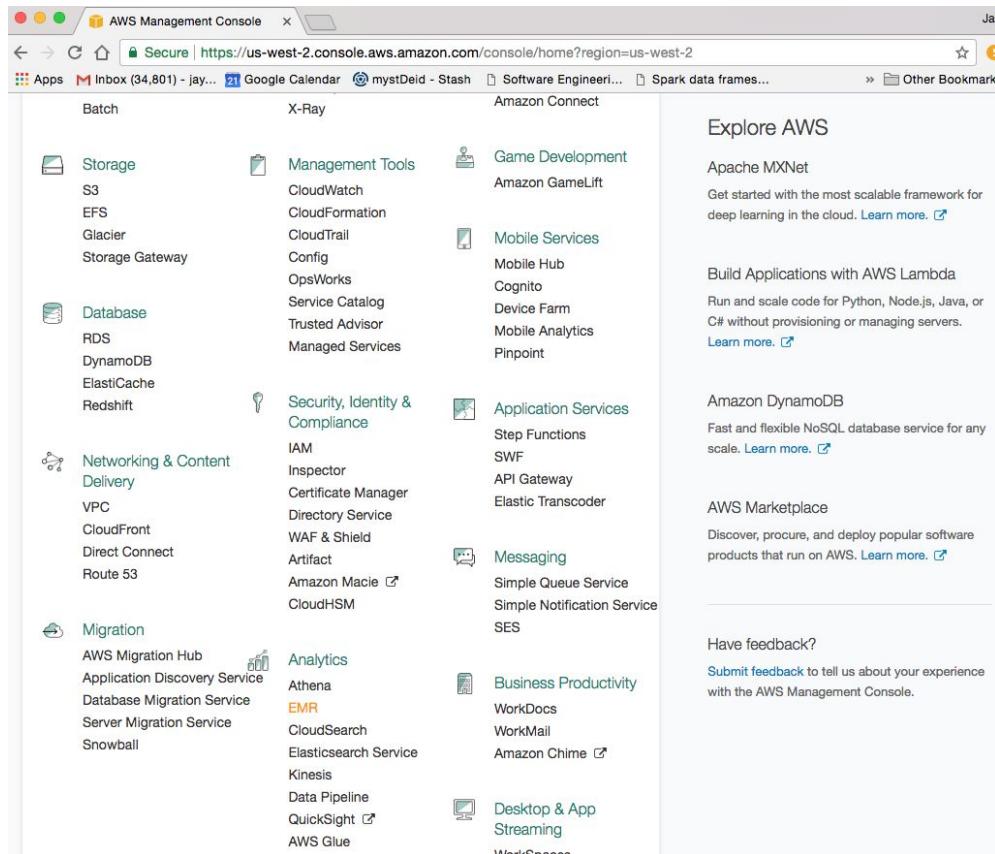
Apache HBase

Presto

## Creating and deploying a MapReduce App on Amazon EMR

Go to [aws.amazon.com](https://aws.amazon.com) create an AWS account.

Log on to AWS. You should see the following services.



### Amazon EMR Management Guide

<http://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html>

### Amazon Elastic MapReduce API

<http://docs.aws.amazon.com/ElasticMapReduce/latest/API/Welcome.html>

### Amazon EMR Release Guide

<http://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hadoop-application.html>

Basic steps for creating a job:

- Upload
- Create
- Monitor

Standard approach:

1. Upload data from S3 (cloud storage) into HDFS (Hadoop File System)
2. Can do MapReduce directly against S3, or you can upload your data from S3 into the HDFS file systems that's part of the EC2 (Elastic Cloud Compute) instances with the JBMs with the Hadoop daemons installed on them.

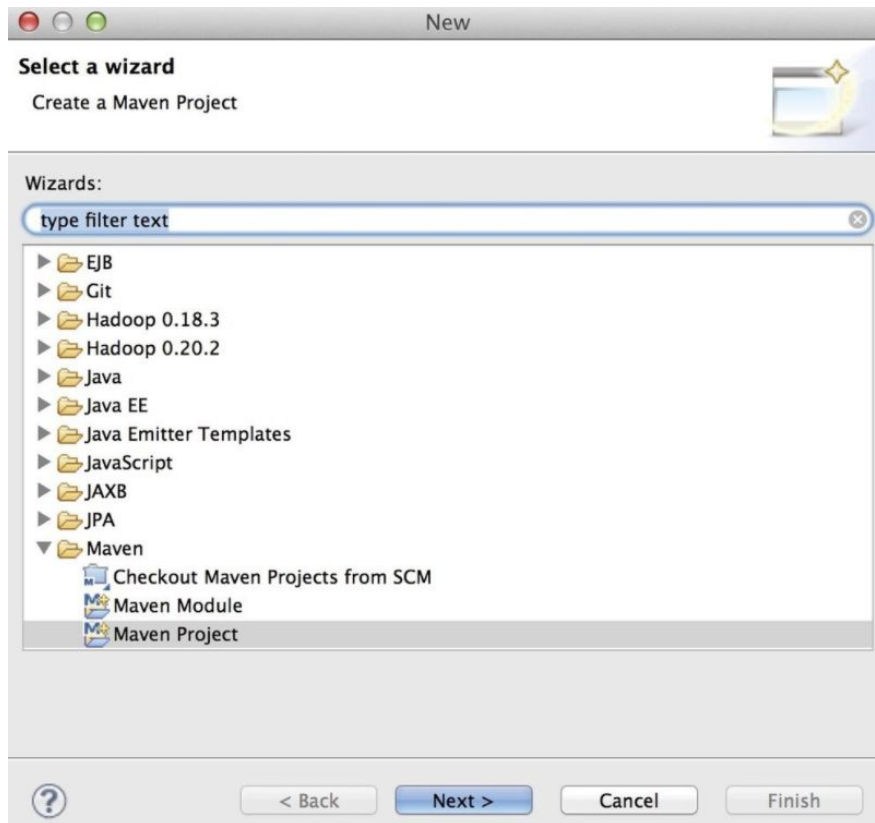
## **Building MapReduce Applications**

This example uses Eclipse Java IDE and the Eclipse Maven plug-in, [m2eclipse](#), to manage application dependencies.

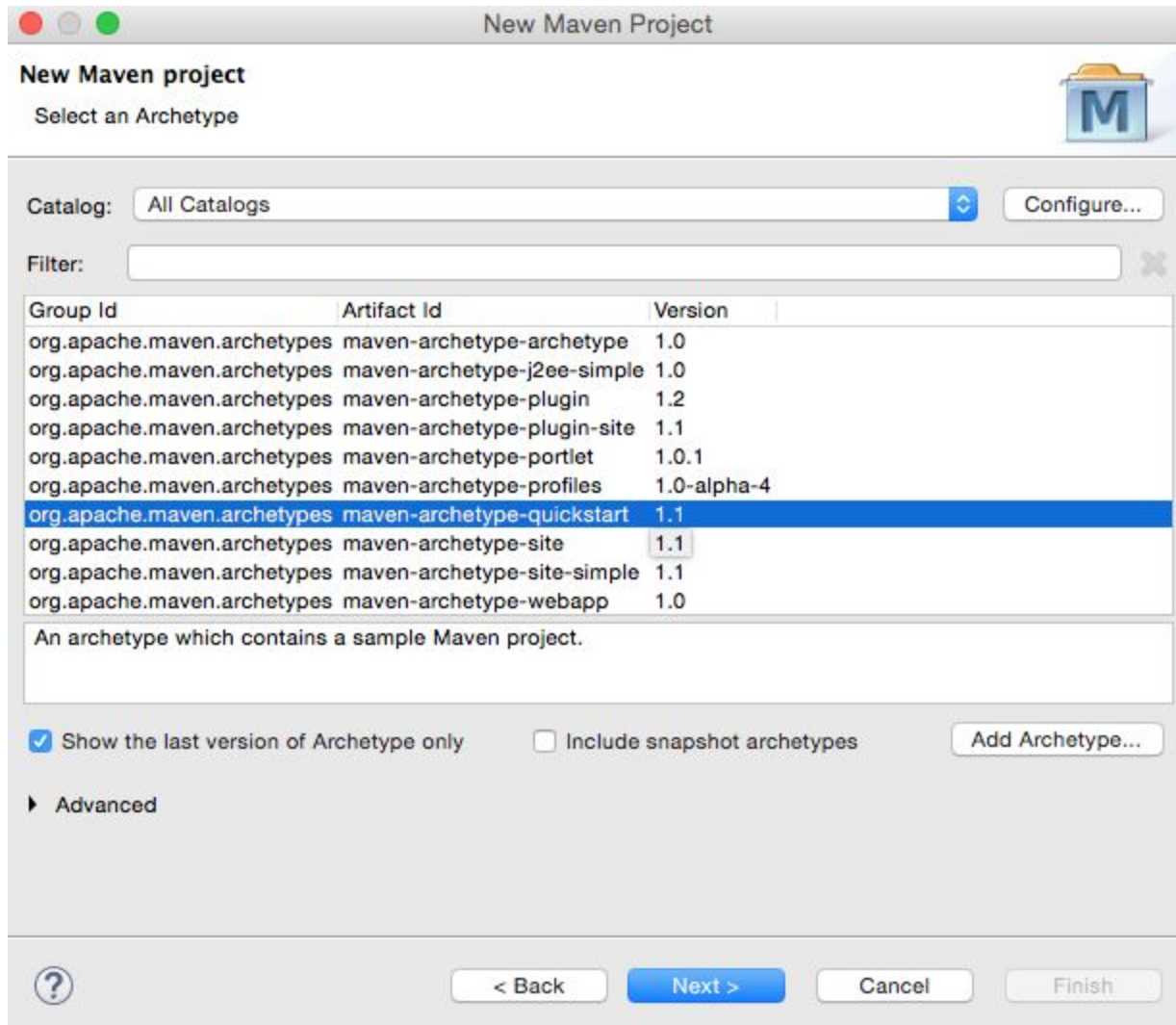
You can install the *m2eclipse plug-in* through the *Install New Software* option inside of Eclipse.

To include the dependencies needed to build the MapReduce applications:

Create a Maven project inside of Eclipse by selecting *File*→*New*→*Other*. The Maven project option should be available after you install the m2eclipse plug-in.



Select the program and project name of your application when going through the Eclipse New Project Wizard.



**New Maven Project**  
Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

Name	Value

► Advanced

*FYI: Maven naming conventions:*

<https://maven.apache.org/guides/mini/guide-naming-conventions.html>

After the project is created, the Hadoop dependencies will need to be added to the project so the application can make use of the Hadoop base classes, types, and methods. You can add the Hadoop core dependencies by opening the *pom.xml* file that is in the root of the project.

## Add the following to your project pom.xml text

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.jayurbain.emr</groupId>
  <artifactId>logEMR</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>com.jayurbain.emr</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-shade-plugin</artifactId>
        <executions>
          <execution>
            <phase>package</phase>
            <goals>
              <goal>shade</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <finalName>uber-${artifactId}-${version}</finalName>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <dependencies>
    <!-- deid -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-client</artifactId>
      <version>2.2.0</version>
    </dependency>
  </dependencies>
</project>
```

In your application:

Create a package *com.jayurbain.emr*

Create a new Java class file *WordCount.java* as follows:

```
//////////
package com.jayurbain.emr.wordcount;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {

    public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value, Context context) throws IOException,
        InterruptedException {
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                word.set(tokenizer.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
```



```

    public void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
}

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();

    Job job = new Job(conf, "wordcount");

    job.setJarByClass(WordCount.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    job.setMapperClass(Map.class);
    job.setReducerClass(Reduce.class);

    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

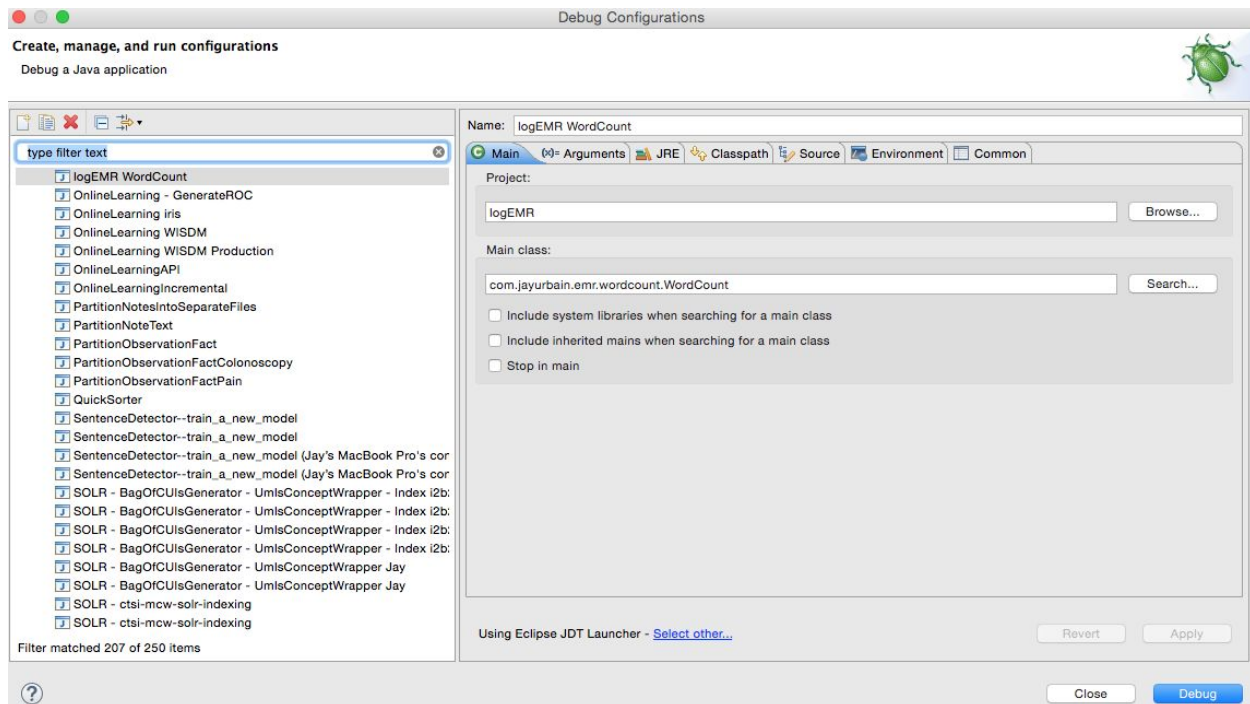
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.waitForCompletion(true);
}

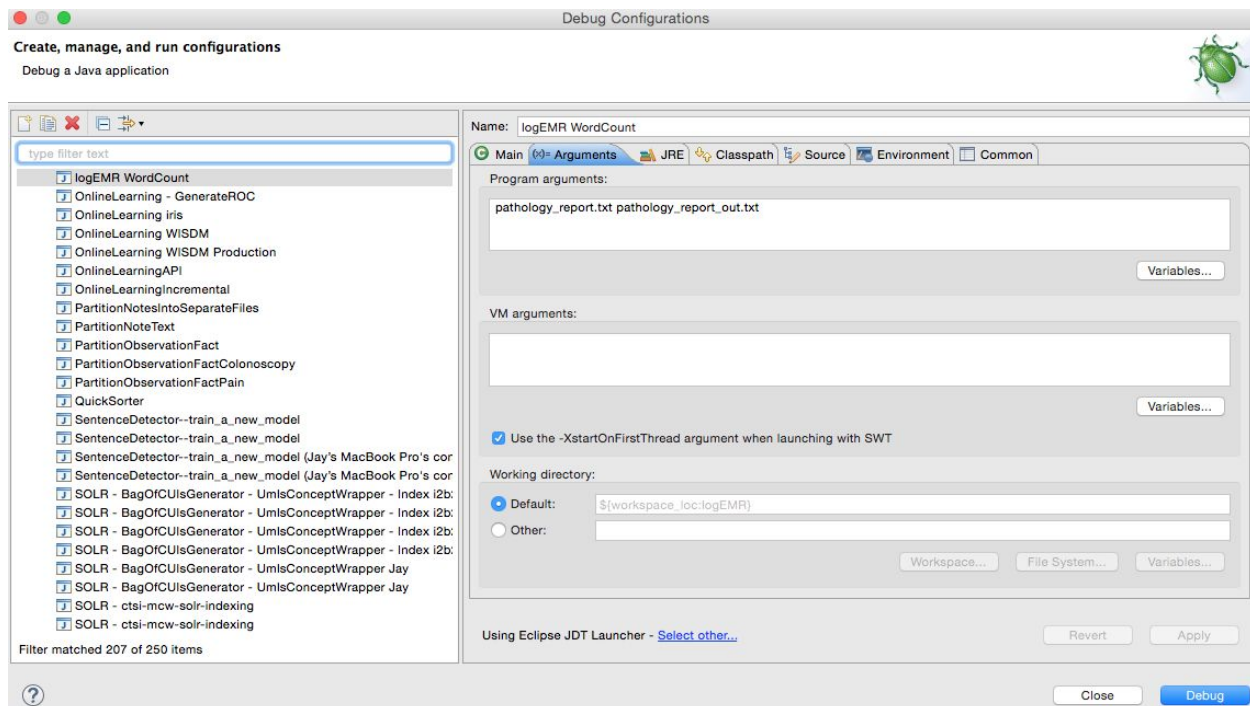
}
//////////

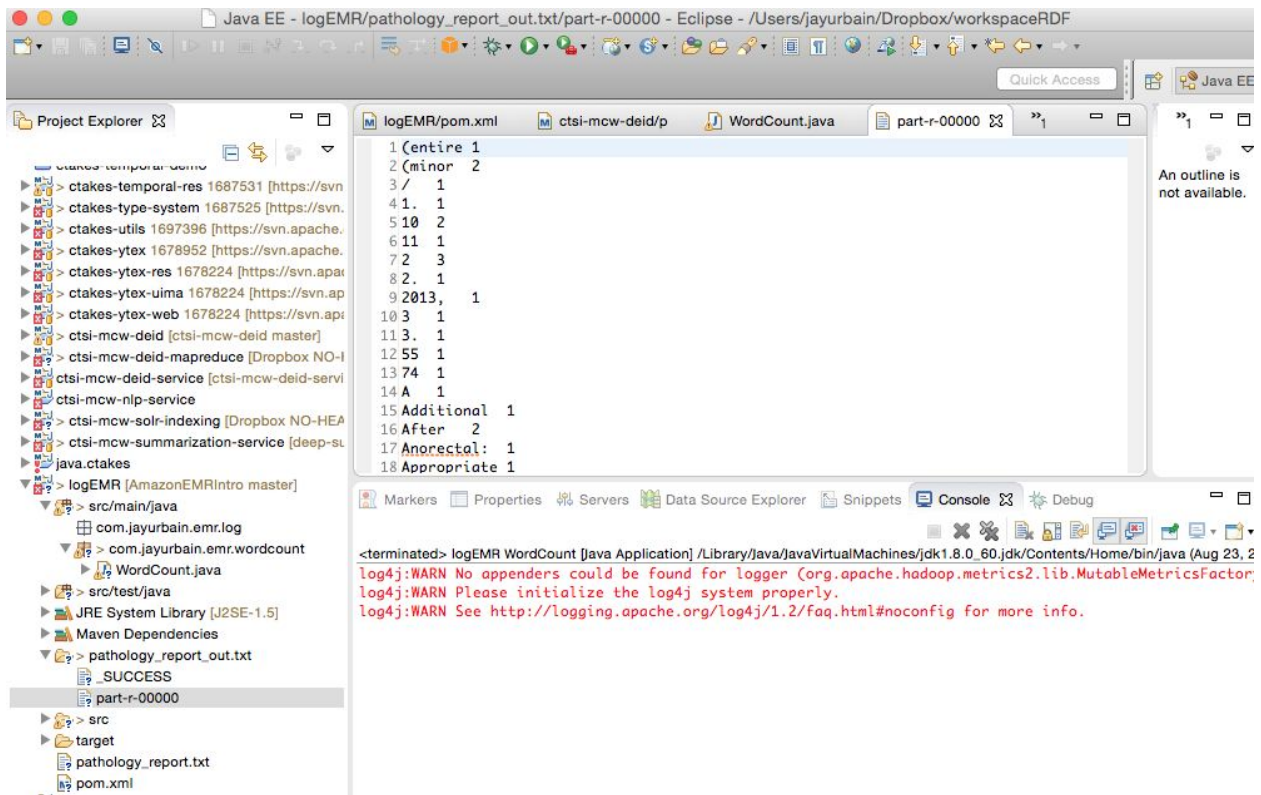
```

You can run the application locally first to test the application. Parameters: input text file, and output text file.

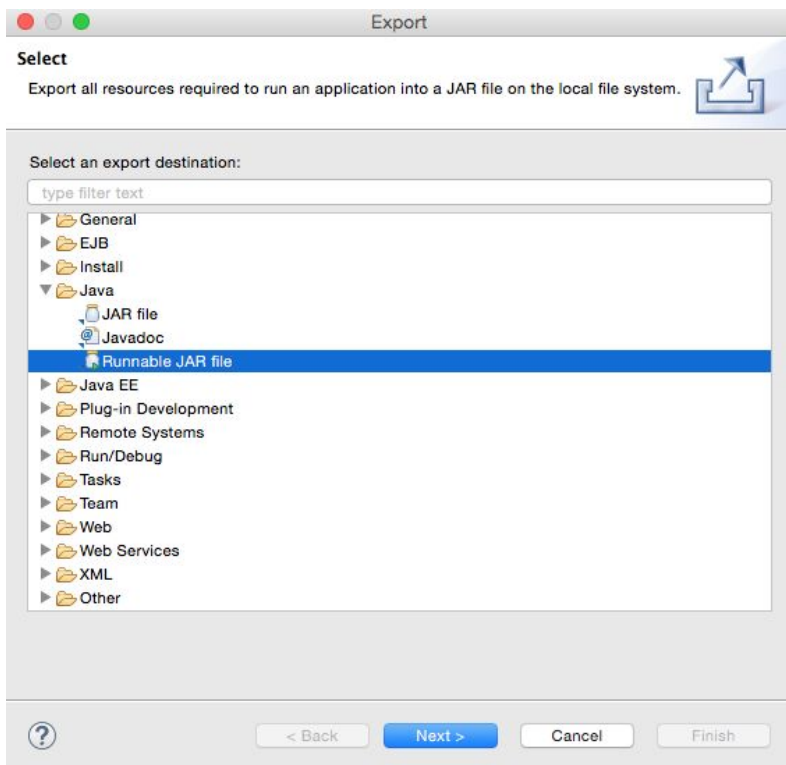
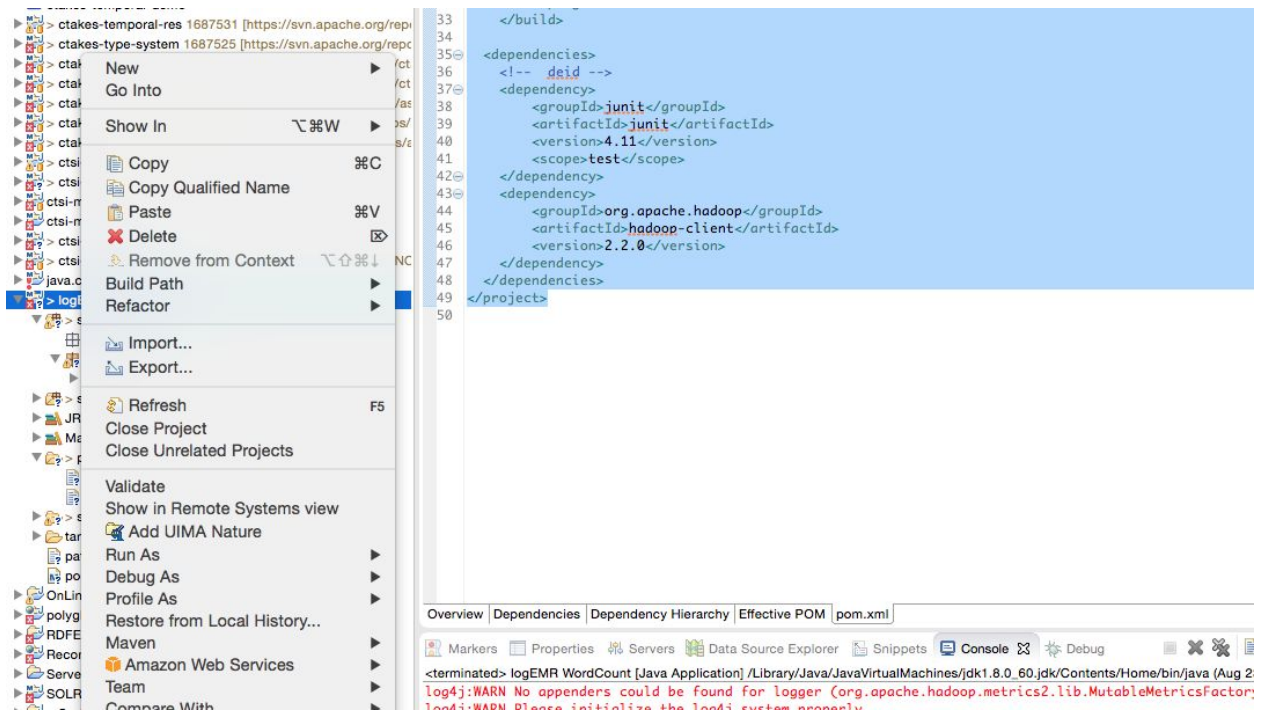


Select arguments.

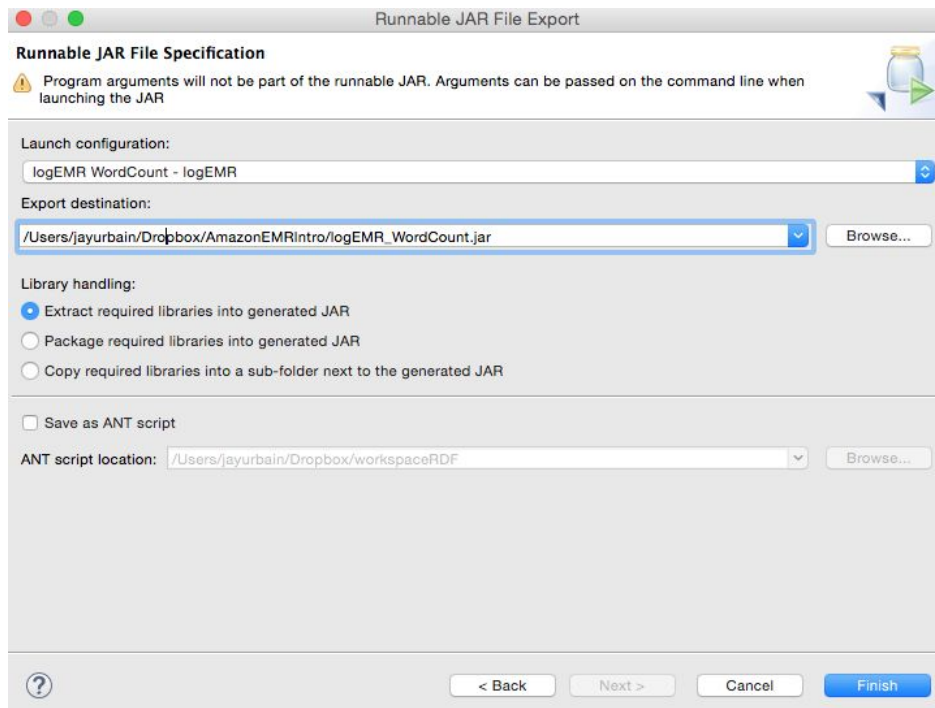




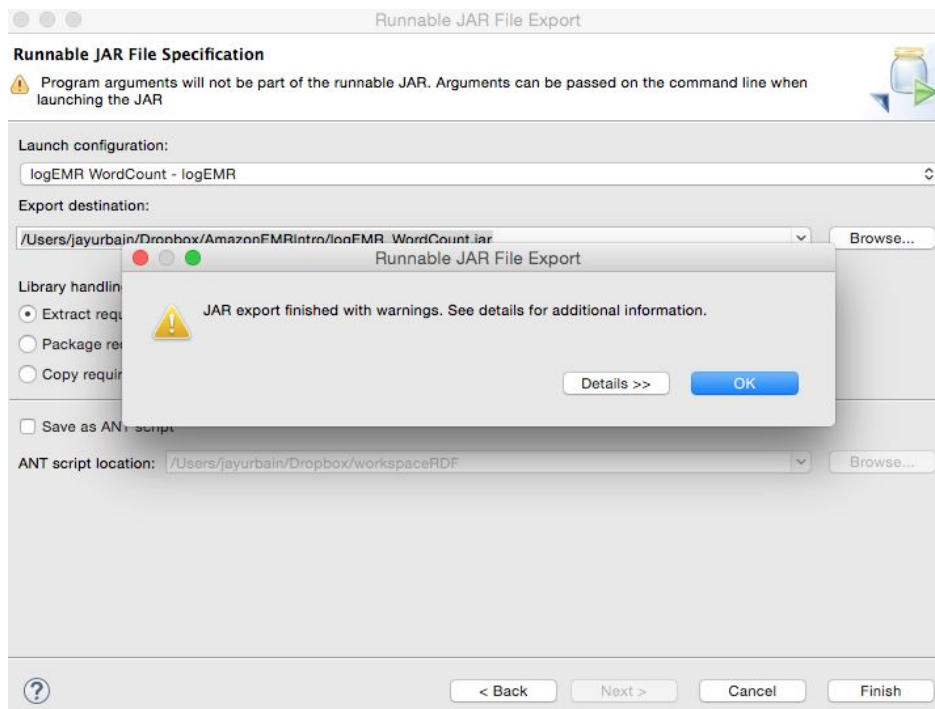
## Export as executable jar file:



Select launch configuration, and export destination.



Select Ok.



Just hit Ok. Only duplicate packaging warnings.

To run the app on a Hadoop with local file system. (See end of this file for installing Hadoop locally).

```
$ hadoop jar logEMR_WordCount patholgoy_report.txt pathology_report_out.txt
```

Run (for hdfs file system):

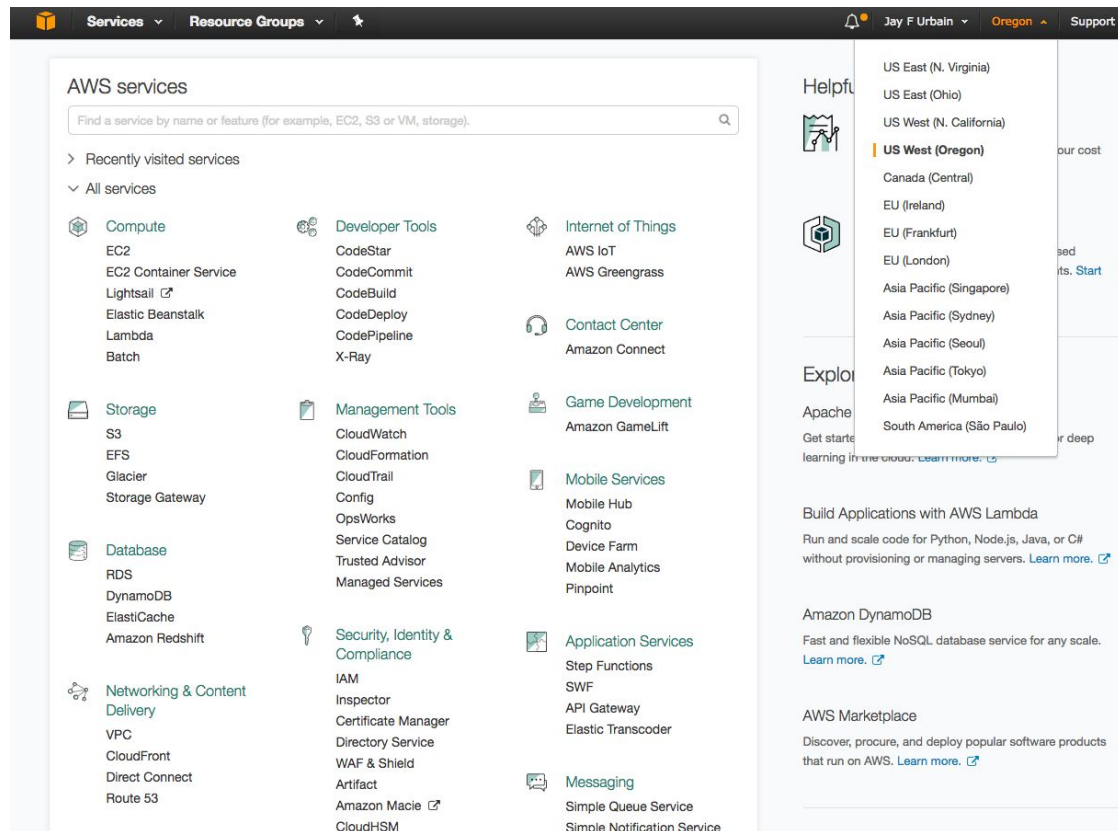
```
$hadoop jar logEMR_WordCount hdfs:///xxx//patholgoy_report.txt  
hdfs:///xxx//pathology_report_out.txt
```

In either case, result file will be in pathology\_report\_out.txt/part-r-00000 (or something similar).

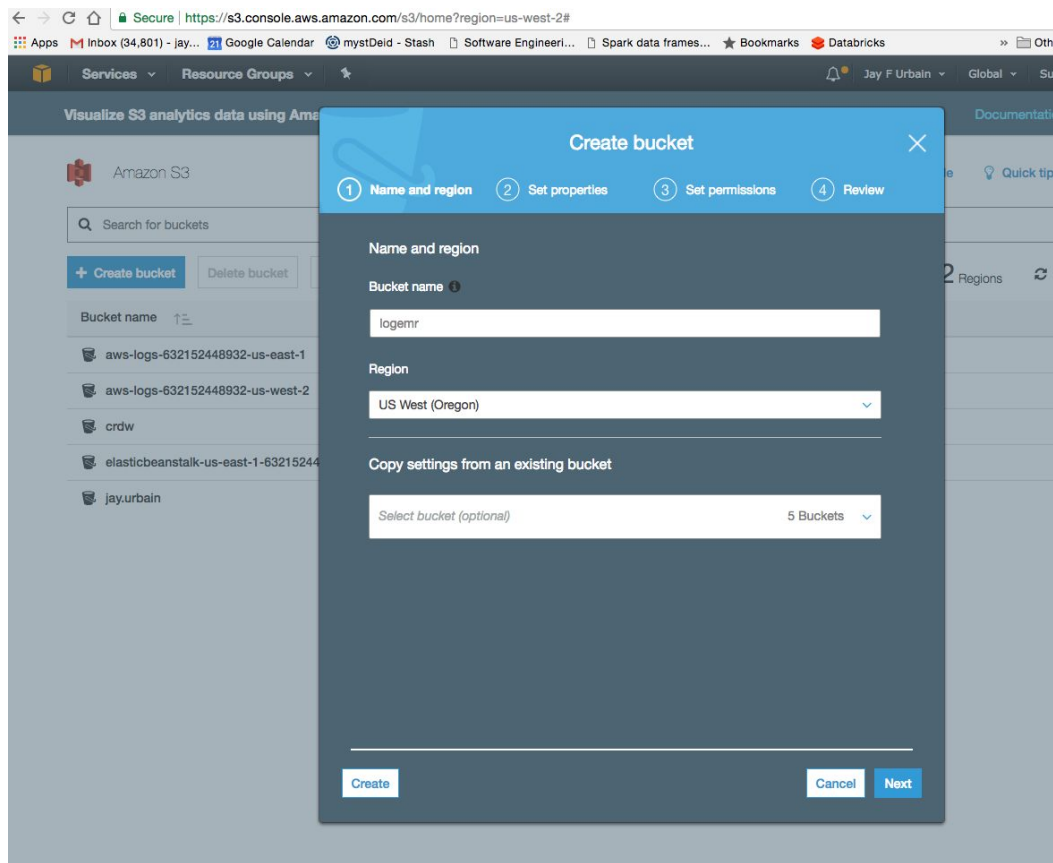
## To run on AmazonEMR:

Sign into AWS Console.

Select S3. Take note of the region. In this case **US West (Oregon)**



Create an S3 bucket.





You can skip the following. Or set properties. They can be changed later.

### Create bucket

✓ Name and region

2 Set properties

3 Set permissions

4 Review

#### Versioning

Keep multiple versions of an object in the same bucket.

[Learn more](#)

✓ Enabled

#### Logging

Set up access log records that provide details about access requests.

[Learn more](#)

✓ Enabled

#### Tags

Use tags to track your cost against projects or other criteria.

[Learn more](#)

0 Tags

Previous

Next

Select next.

### Create bucket

✓ Name and region

✓ Set properties

3 Set permissions

4 Review

#### Manage users

User ID	Objects	Object permissions	
jay@upstreamdev.com(Owner)	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	×

#### Manage public permissions

Do not grant public read access to this bucket (Recommended) ▾

#### Manage system permissions

Grant Amazon S3 Log Delivery group write access to this bucket ▾

Previous

Next

Select Create bucket.

### Create bucket

✓ Name and region

✓ Set properties

✓ Set permissions

4 Review

Name and regionEdit

Bucket name

logemr

Region

US West (Oregon)

PropertiesEdit

Versioning

Enabled

Logging

Enabled

Tagging

0 Tags

PermissionsEdit

Users

1

Public permissions

Disabled

System permissions

Enabled

Previous

Create bucket

You should see the following.

Visualize S3 analytics data using Amazon QuickSight [Learn More »](#)

Documentation

Amazon S3

[Switch to the old console](#) [Discover the new console](#) [Quick tips](#)

Search for buckets

Create bucket

Delete bucket

Empty bucket

6 Buckets

2 Regions

Bucket name	Region	Date created
aws-logs-632152448932-us-east-1	US East (N. Virginia)	Jun 24, 2017 10:22:00 AM
aws-logs-632152448932-us-west-2	US West (Oregon)	Aug 22, 2017 2:37:11 PM
crdw	US East (N. Virginia)	Sep 14, 2016 5:01:56 PM
elasticbeanstalk-us-east-1-632152448932	US East (N. Virginia)	May 9, 2011 5:30:15 AM
jay.urbain	US East (N. Virginia)	Aug 3, 2009 10:43:57 AM
logemr	US West (Oregon)	Aug 23, 2017 2:28:11 PM

Select your logerr bucket and upload your jar and *text input* file.

Google Calendar mystDeid - Stash Software Engineer... Spark data frames...

Services Resource Groups

Jay F Urbain Global

Amazon S3 logemr

Overview Properties

Upload

1 Select files 2 Set permissions 3 Set properties 4 Review

Drag and drop here OR

Add files

Upload Next

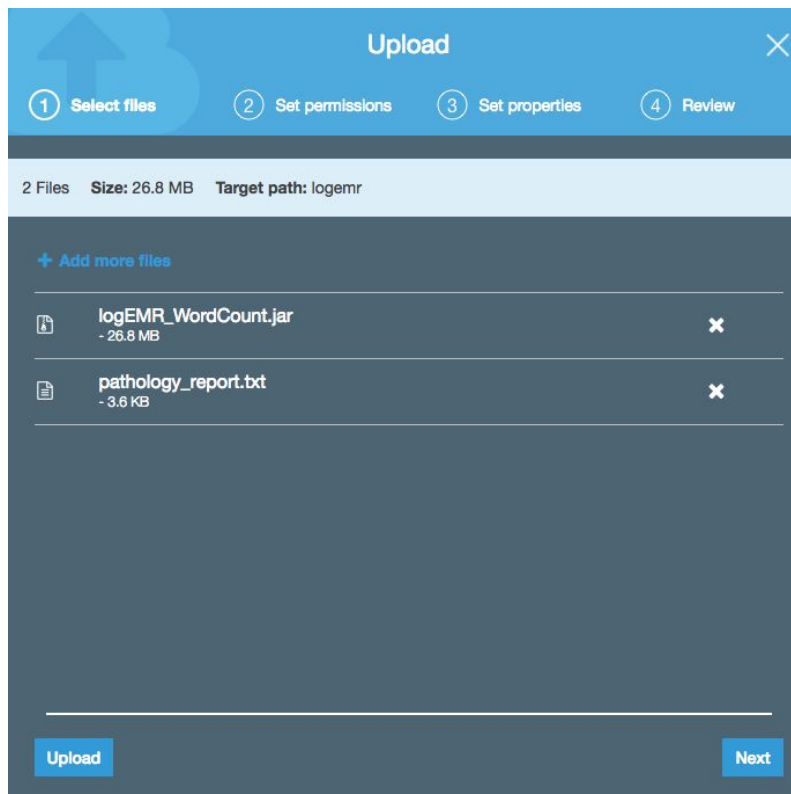
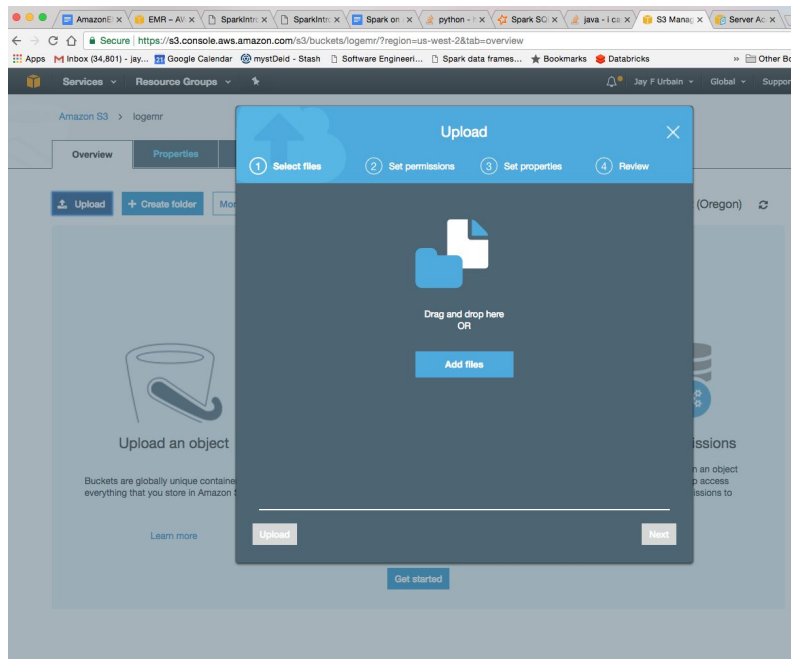
Get started

tensorflow / nmt

AmazonEMRIntro

Search

File list: .DS\_Store, .git, amazonemrmasterclass...351-1va1-app6892.pdf, Ex\_Files\_Amazon\_Web\_Data, Ex\_Files\_Amazon\_Web\_Data.zip, logEMR\_WordCount.jar, README.md, workspaceEMR, logEMR, .classpath, .gitignore, .project, .settings, pathology\_report\_out.txt, pathology\_report.txt, pom.xml, src, target



Upload

1 Select files

2 Set permissions

3 Set properties

4 Review

2 Files   Size: 26.8 MB   Target path: logemr

Manage users

User ID	Objects	Object permissions
jay@upstreamdev.com(Owner)	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write <span>×</span>

Manage public permissions

Do not grant public read access to this object(s) (Recommended) ▼

Upload

Previous

Next

Upload

✓

Select files

✓

Set permissions

3

Set properties

4

Review

2 Files

Size: 26.8 MB

Target path: logemr

Storage class

Choose one depending on your use case scenario and performance access requirements.

☒ Standard

☐ Standard-IA

☐ Reduced redundancy

Encryption

Protect data at rest by using Amazon S3 master-key or by using AWS KMS master-key.

☒ None

☐ Amazon S3 master-key

☐ AWS KMS master-key

Metadata


Metadata is a set of name-value pairs. You cannot modify object metadata after it is uploaded.

Header	Value
--------	-------


Upload


Previous


Next





Upload



 Select files

 Set permissions

 Set properties

 Review

Files

Edit

2 Files

Size: 26.8 MB

Permissions

Edit

1 grantees

Properties

Edit

Encryption

Storage class

No

Standard

Metadata

Previous

Upload

Sure asks a lot of questions. Select Upload.

*Be patient.*



Services

Resource Groups

Jay F Urbain

Global

Sup

Amazon S3 > logemr

Overview

Properties

Permissions

Management

Q

Type a prefix and press Enter to search. Press ESC to clear.

Upload

Create folder

More

AllDeleted objects

US West (Oregon)

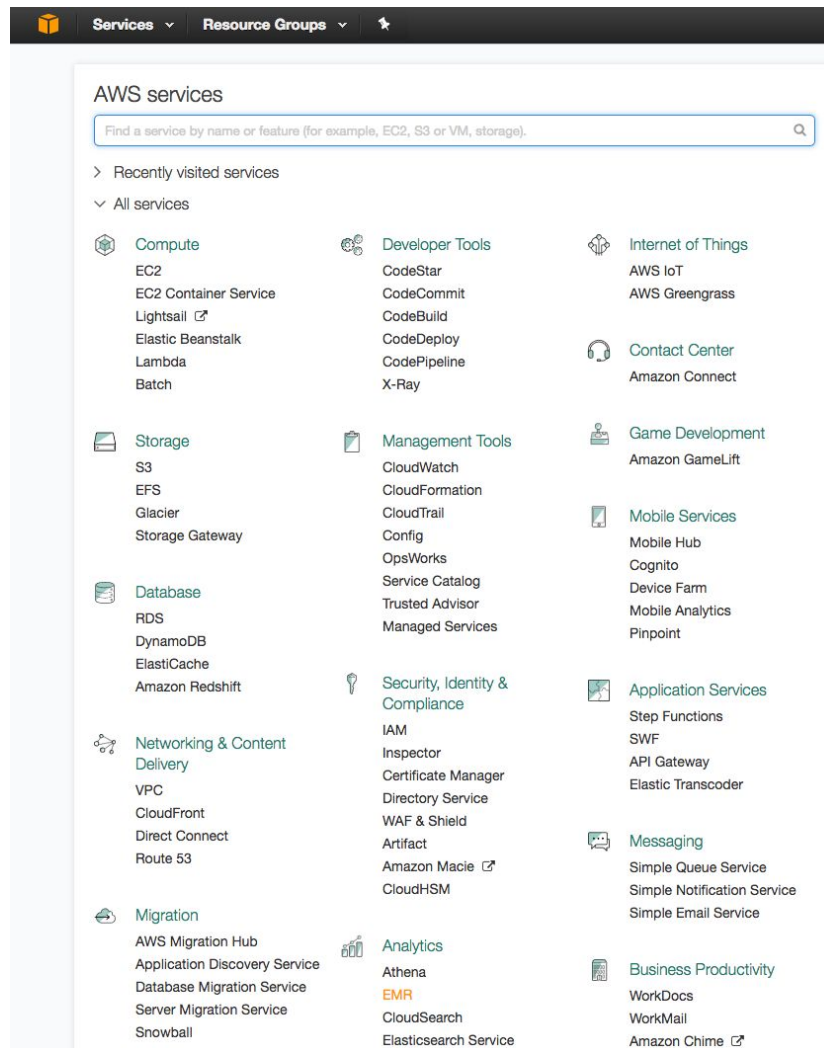
Viewing 1 to 2

<input type="checkbox"/>	Name	Last modified	Size	Storage class
<input type="checkbox"/>	logEMR_WordCount.jar	Aug 23, 2017 2:34:48 PM	26.8 MB	Standard
<input type="checkbox"/>	pathology_report.txt	Aug 23, 2017 2:35:36 PM	3.6 KB	Standard

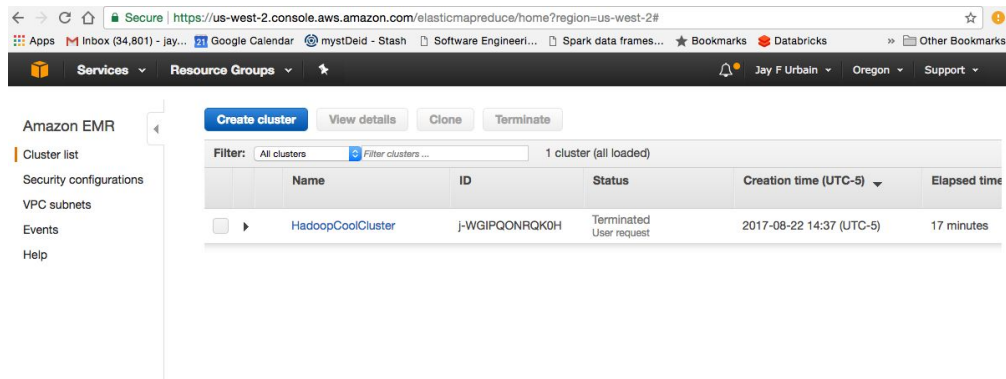
Viewing 1 to 2

tada!

## Select EMR from AWS Services



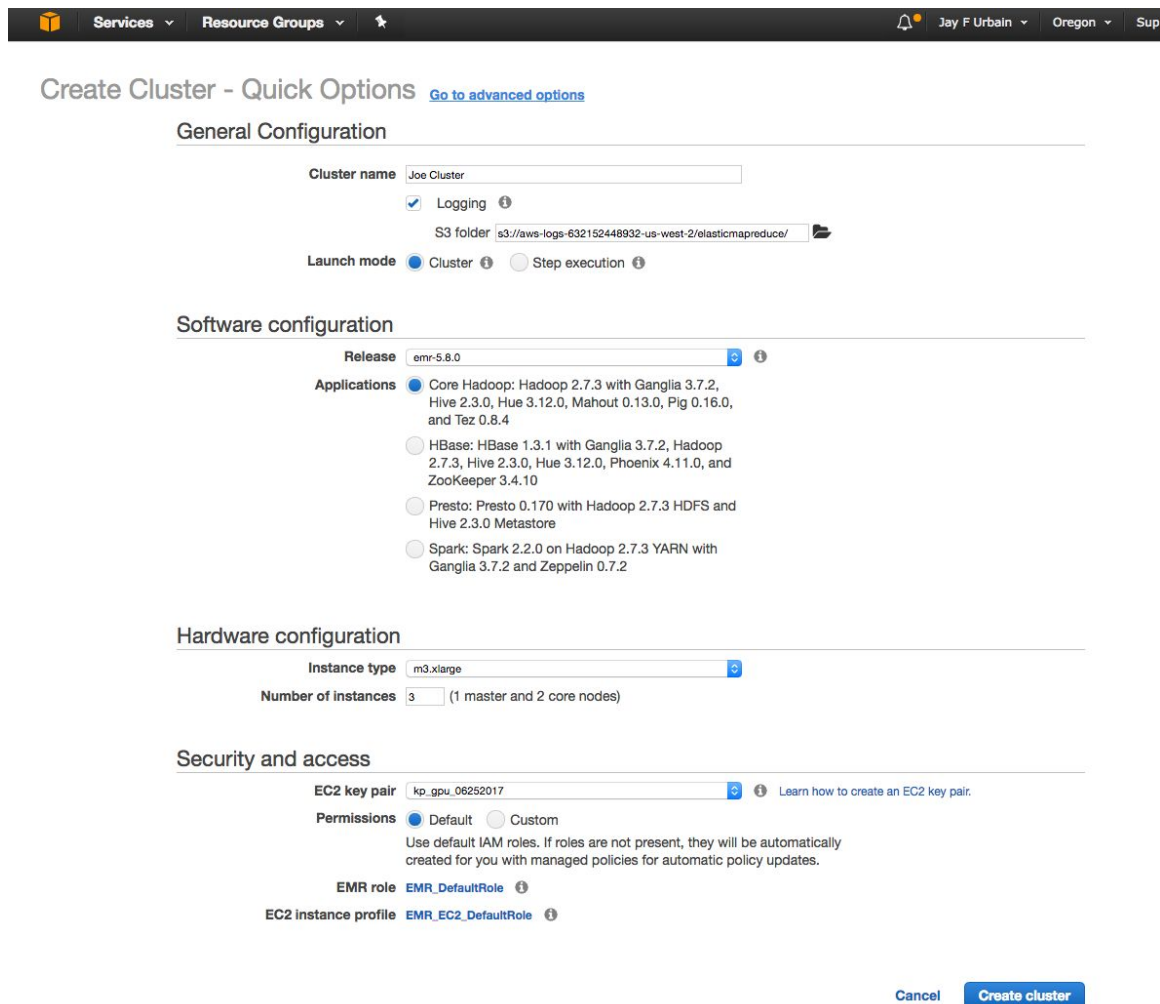
## Select Create Cluster:



The screenshot shows the Amazon EMR console interface. On the left is a navigation menu with options: Cluster list, Security configurations, VPC subnets, Events, and Help. The main area displays a table of clusters. At the top, there are buttons for 'Create cluster', 'View details', 'Clone', and 'Terminate'. The table has columns for Name, ID, Status, Creation time (UTC-5), and Elapsed time. One cluster is listed: 'HadoopCoolCluster' with ID 'j-WGIPQONRQK0H', status 'Terminated User request', creation time '2017-08-22 14:37 (UTC-5)', and elapsed time '17 minutes'.

Name	ID	Status	Creation time (UTC-5)	Elapsed time
HadoopCoolCluster	j-WGIPQONRQK0H	Terminated User request	2017-08-22 14:37 (UTC-5)	17 minutes

Make the following selections (do NOT select Create cluster until you scroll down):



The screenshot shows the 'Create Cluster - Quick Options' form. It is divided into four sections: General Configuration, Software configuration, Hardware configuration, and Security and access. At the bottom are 'Cancel' and 'Create cluster' buttons.

**General Configuration**

- Cluster name: Joe Cluster
- ☒ Logging
- S3 folder: s3://aws-logs-632152448932-us-west-2/elasticmapreduce/
- Launch mode: ☒ Cluster ☐ Step execution

**Software configuration**

- Release: emr-5.8.0
- Applications: ☒ Core Hadoop: Hadoop 2.7.3 with Ganglia 3.7.2, Hive 2.3.0, Hue 3.12.0, Mahout 0.13.0, Pig 0.16.0, and Tez 0.8.4. Other options include HBase, Presto, and Spark.

**Hardware configuration**

- Instance type: m3.xlarge
- Number of instances: 3 (1 master and 2 core nodes)

**Security and access**

- EC2 key pair: kp\_gpu\_06252017
- Permissions: ☒ Default ☐ Custom
- EMR role: EMR\_DefaultRole
- EC2 instance profile: EMR\_EC2\_DefaultRole

Select [Go to advanced options](#)

## Create a custom JAR

The screenshot shows the 'Create Cluster - Advanced Options' page in the AWS EMR console. The 'Software Configuration' section is active, showing a list of software releases and their associated components. The 'Release' dropdown is set to 'emr-5.8.0'. The components listed are:

Component	Selected
Hadoop 2.7.3	<input checked="" type="checkbox"/>
Flink 1.3.1	<input type="checkbox"/>
Pig 0.16.0	<input checked="" type="checkbox"/>
ZooKeeper 3.4.10	<input type="checkbox"/>
Zeppelin 0.7.2	<input type="checkbox"/>
Ganglia 3.7.2	<input type="checkbox"/>
Hive 2.3.0	<input checked="" type="checkbox"/>
Sqoop 1.4.6	<input type="checkbox"/>

An 'Add Step' modal is open, showing the configuration for a 'Custom JAR' step. The fields are:

- Step type:** Custom JAR
- Name:** logEMR\_WordCount.jar
- JAR location:** s3://logemr/logEMR\_WordCount.jar
- Arguments:** s3://logemr/pathology\_report.txt, s3://logemr/pathology\_report\_out.txt
- Action on failure:** Terminate cluster

The modal also includes a 'Cancel' button and an 'Add' button.

Services

Resource Groups

Jay F Urbain

Oregon

Support

## 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-5.8.0

☒ Hadoop 2.7.3  
☐ Flink 1.3.1  
☒ Pig 0.16.0  
☐ ZooKeeper 3.4.10  
☒ Hue 3.12.0  
☐ Spark 2.2.0

☐ Zeppelin 0.7.2  
☐ Ganglia 3.7.2  
☒ Hive 2.3.0  
☐ Sqoop 1.4.6  
☐ Phoenix 4.11.0  
☐ HCatalog 2.3.0

☐ Tez 0.8.4  
☐ HBase 1.3.1  
☐ Presto 0.170  
☐ Mahout 0.13.0  
☐ Oozie 4.3.0

Edit software settings (optional)

☒ Enter configuration ☐ Load JSON from S3

`classification=config-file-name,properties={myKey1=myValue1,myKey2=myValue2}`

### Add steps (optional)

Name	Action on failure	JAR location	Arguments
logEMR_WordCount.jar	Terminate cluster	s3://logemr/logEMR_WordCount.jar	s3://logemr/pathology_report.txt s3://logemr/pathology_report_out.txt

Step type Custom JAR Configure

☐ Auto-terminate cluster after the last step is completed

Cancel

Next

Select next.

Services

Resource Groups

Jay F Urbain

Oregon

## Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

### General Options

Cluster name My cluster

☒ Logging

S3 folder s3://aws-logs-632152448932-us-west-2/elasticmapreduce/

☒ Debugging

☒ Termination protection

Scale down behavior Terminate at instance hour

### Tags

Key	Value (optional)
<i>Add a key to create a tag</i>	

### Additional Options

☐ EMRFS consistent view

Custom AMI ID None

Bootstrap Actions

Cancel

Previous

Next

Some serious computer power here.

Select terminate at task completion

Services Resource Groups

Create Cluster - Advanced Options [Go to quick options](#)

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

### General Options

Cluster name

☒ Logging ⓘ  
S3 folder

☒ Debugging ⓘ

☒ Termination protection ⓘ

Scale down behavior  ⓘ

### Tags ⓘ

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

### Additional Options

☐ EMRFS consistent view ⓘ

Custom AMI ID  ⓘ

**Bootstrap Actions**

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](#)

Add bootstrap action

Next.

ServicesResource Groups

Jay F UrbainOregonSupport

Create Cluster - Advanced OptionsGo to quick options

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Security Options

EC2 key pairProceed without an EC2 key pair

☒ Cluster visible to all IAM users in account

Permissions

☒ Default☐ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR roleEMR\_DefaultRole

EC2 instance profileEMR\_EC2\_DefaultRole

Auto Scaling roleEMR\_AutoScaling\_DefaultRole

Encryption Options

EC2 Security Groups

No EC2 key pair has been selected, so you will not be able to SSH to this cluster or connect to HUE (unless you are using a VPN). Learn how to create an EC2 Key Pair.

CancelPreviousCreate cluster

Create cluster


Services ▾ Resource Groups ▾ ⌵

Amazon EMR

Cluster list  
Security configurations  
VPC subnets  
Events  
Help

Add step Resize Clone Terminate AWS CLI export

Cluster: My cluster **Starting**

<b>Connections:</b> -- <b>Master public DNS:</b> -- <b>Tags:</b> -- <a href="#">View All / Edit</a>	
<b>Summary</b> <b>ID:</b> j-1ZX4K2HUZFT6N <b>Creation date:</b> 2017-08-23 14:44 (UTC-5) <b>Elapsed time:</b> 0 seconds <b>Auto-terminate:</b> No <b>Termination protection:</b> On <a href="#">Change</a>	<b>Configuration Details</b> <b>Release label:</b> emr-5.8.0 <b>Hadoop distribution:</b> Amazon 2.7.3 <b>Applications:</b> Hive 2.3.0, Pig 0.16.0, Hue 3.12.0 <b>Log URI:</b> s3://aws-logs-632152448932-us-west-2/elasticmapreduce/  <b>EMRFS consistent view:</b> Disabled <b>Custom AMI ID:</b> --
<b>Network and Hardware</b> <b>Availability zone:</b> -- <b>Subnet ID:</b> <a href="#">subnet-26a6ef6f</a> <b>Master:</b> <b>Provisioning</b> 1 m3.xlarge <b>Core:</b> <b>Provisioning</b> 2 m3.xlarge <b>Task:</b> --	<b>Security and Access</b> <b>Key name:</b> -- <b>EC2 instance profile:</b> EMR_EC2_DefaultRole <b>EMR role:</b> EMR_DefaultRole <b>Auto Scaling role:</b> EMR_AutoScaling_DefaultRole <b>Visible to all users:</b> All <a href="#">Change</a> <b>Security groups for</b> <a href="#">sg-d16d34ab</a> (ElasticMapReduce-Master: master) <b>Security groups for</b> <a href="#">sg-286b3252</a> (ElasticMapReduce-Core & Task: slave)

- ▶ Monitoring
- ▶ Hardware
- ▶ Steps
- ▶ Configurations
- ▶ Events
- ▶ Bootstrap Actions

Be patient. If you're bored, you can look at the instances starting:

Go back to services, select EC2, and then select EC2 instances running:



Services ▾ Resource Groups ▾ ⭐

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

**Instances**

Spot Requests

Reserved Instances

Scheduled Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Launch Instance Connect Actions ▾

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-01ba82c25758201c5	m3.xlarge	us-west-2a	running	2/2 checks ...
	i-0b696ec456ef47ced	m3.xlarge	us-west-2a	running	2/2 checks ...
	i-0e167329278f9f5a2	m3.xlarge	us-west-2a	running	2/2 checks ...

Go back to your clusters.

← → ↻ 🏠 https://us-west-2.console.aws.amazon.com/elasticmapreduce/home?region=us-west-2#cluster-list: ⭐ ⓘ

Apps 📧 Inbox (34,801) - Jay... 📅 Google Calendar 🌐 mystDeid - Stash 📁 Software Engineeri... 📁 Spark data frames... ⭐ Bookmarks 📊 Databricks ⚙️ Spark Programmin... » Other Bookmarks

Services ▾ Resource Groups ▾ ⭐

Amazon EMR

**Cluster list**

Security configurations

VPC subnets

Events

Help

Create cluster View details Clone Terminate

Filter: All clusters Filter clusters ... 2 clusters (all loaded)

Name	ID	Status	Creation time (UTC-5)	Elapsed time	Normalized instance hours
My cluster	j-1ZX4K2HUZFT6N	Starting	2017-08-23 14:44 (UTC-5)	6 minutes	0
HadoopCoolCluster	j-WGIPQONRQK0H	Terminated User request	2017-08-22 14:37 (UTC-5)	17 minutes	24

Check S3

Services ▾ Resource Groups ▾ ☆

Amazon S3 > logemr

Overview Properties Permissions Management

🔍 Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder More ▾ All Deleted objects

US West (Oregon) ↻

Viewing 1 to 3

<input type="checkbox"/>	Name ↑ ▾	Last modified ↑ ▾	Size ↑ ▾	Storage class ↑ ▾
<input type="checkbox"/>	📁 pathology_report_out.txt	--	--	--
<input type="checkbox"/>	📄 logEMR_WordCount.jar	Aug 23, 2017 2:34:48 PM	26.8 MB	Standard
<input type="checkbox"/>	📄 pathology_report.txt	Aug 23, 2017 2:35:36 PM	3.6 KB	Standard

Viewing 1 to 3

This is exciting! Select your output directory.

Services ▾ Resource Groups ▾ ☆

Amazon S3 > logemr / pathology\_report\_out.txt

Overview

🔍 Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder More ▾ All Deleted objects



US West (Oregon) ↻

Viewing 1 to 8

<input type="checkbox"/>	Name ↑ ▾	Last modified ↑ ▾	Size ↑ ▾	Storage class ↑ ▾
<input type="checkbox"/>	📄 _SUCCESS	Aug 23, 2017 2:54:33 PM	0 B	Standard
<input type="checkbox"/>	📄 part-r-00000	Aug 23, 2017 2:54:26 PM	409.0 B	Standard
<input type="checkbox"/>	📄 part-r-00001	Aug 23, 2017 2:54:28 PM	359.0 B	Standard
<input type="checkbox"/>	📄 part-r-00002	Aug 23, 2017 2:54:27 PM	494.0 B	Standard
<input type="checkbox"/>	📄 part-r-00003	Aug 23, 2017 2:54:30 PM	373.0 B	Standard
<input type="checkbox"/>	📄 part-r-00004	Aug 23, 2017 2:54:28 PM	316.0 B	Standard
<input type="checkbox"/>	📄 part-r-00005	Aug 23, 2017 2:54:31 PM	379.0 B	Standard
<input type="checkbox"/>	📄 part-r-00006	Aug 23, 2017 2:54:32 PM	457.0 B	Standard

Viewing 1 to 8

SUCCESS!!!

 **Services** ▾ **Resource Groups** ▾ 

Amazon S3 > logemr / pathology\_report\_out.txt

**\_SUCCESS** [Latest version ▾](#)

Overview

**Properties**

Permissions

Open

Download

Download as

Make public

Copy path

**Owner**  
jay@upstreamdev.com

**Last modified**  
Aug 23, 2017 2:54:33 PM

**Etag**  
d41d8cd98f00b204e9800998ecf8427e

**Storage class**  
Standard

**Server side encryption**  
None

**Size**  
0

**Link**  
[https://s3-us-west-2.amazonaws.com/logemr/pathology\\_report\\_out.txt/\\_SUCCESS](https://s3-us-west-2.amazonaws.com/logemr/pathology_report_out.txt/_SUCCESS)

You can download output files individually. S3 is not designed for browsing file contents.

Note: if you install the command line tools, you can do the following:  
`aws s3 sync s3://mybucket .`

Amazon S3 > logemr / pathology\_report\_out.txt

Overview

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder More All Deleted objects

<input type="checkbox"/>	Name	Last modified
<input type="checkbox"/>	._SUCCESS	Aug 23, 2017 2:54:33 PM
<input type="checkbox"/>	part-r-00000	Aug 23, 2017 2:54:26 PM
<input type="checkbox"/>	part-r-00001	Aug 23, 2017 2:54:28 PM
<input type="checkbox"/>	part-r-00002	Aug 23, 2017 2:54:27 PM
<input type="checkbox"/>	part-r-00003	Aug 23, 2017 2:54:30 PM
<input type="checkbox"/>	part-r-00004	Aug 23, 2017 2:54:28 PM
<input type="checkbox"/>	part-r-00005	Aug 23, 2017 2:54:31 PM
<input checked="" type="checkbox"/>	part-r-00006	Aug 23, 2017 2:54:32 PM

### part-r-00006

Download Copy path

Latest version

**Overview**

Key	part-r-00006
Size	457
Expiration date	N/A
Expiration rule	N/A
ETag	870db3fd574db62e8b12e82fc8f41dfa
Last modified	Aug 23, 2017 2:54:32 PM GMT-0500
Link	<a href="https://s3-us-west-2.amazonaws.com/logemr/pathology_report_out.txt/part-r-00006">https://s3-us-west-2.amazonaws.com/logemr/pathology_report_out.txt/part-r-00006</a>

**Properties**

Storage class	Standard
Encryption	None
Metadata	1
Tags	0 Tags

**Permissions**

Owner jay@upstreamdev.com

Object permissions

Read	2 Grantees
Write	1 Grantees

Object permissions

Read	1 Grantees
Write	1 Grantees

```
part-r-00000 x part-r-00001 x part-r-00002 x part-r-00003 x part-r-00004 x part-r-00005 x part-r-00006 x
1 1. 1
2 Additional 1
3 Colon: 7
4 Descending 1
5 Diagnosis:none 1
6 During 1
7 Histopathologic 1
8 IV 2
9 Left 1
10 Medications 1
11 Medications: 1
12 Midazolam 1
13 Normal 9
14 Olympus 1
15 Recommendations: 1
16 Sigmoid 1
17 The 3
18 [LOCATION] 1
19 achieve 1
20 are 1
21 average 2
22 desaturation. 2
23 difficult. 1
24 direct 2
25 discomfort, 2
26 examination, 1
27 for 4
28 help 1
29 here 2
30 ileum. 1
31 inadequate 1
32 lavage 1
33 liquid) 1
34 monitored 1
35 oximetry 1
36 place. 1
```

You can create a single file using cat:

```
Jays-MacBook-Pro-2:emr_output jayurbain$ ls
part-r-00000  part-r-00001  part-r-00002  part-r-00003  part-r-00004  part-r-00005  part-r-00006  part-r-al
Jays-MacBook-Pro-2:emr_output jayurbain$ cat part-r-00000 part-r-00001 part-r-00002 part-r-00003 part-r-00004 part-r-00005 part-r-00006 > part-r-all
Jays-MacBook-Pro-2:emr_output jayurbain$ ls
part-r-00000  part-r-00001  part-r-00002  part-r-00003  part-r-00004  part-r-00005  part-r-00006  part-r-al  part-r-all
Jays-MacBook-Pro-2:emr_output jayurbain$ cat part-r-all
After 2
CF 1
Currently 1
Details: 1
Grade 3
Impression: 1
MD 3
Physician 1
Prep 1
Procedure 2
Rectum: 1
Small 2
[PERSON] 5
and 10
and/or 2
completion 1
described 1
diet. 1
digital 1
evidence 1
found 1
hemorrhoids 2
home, 1
in 6
incrementally 1
lateral 1
made 1
obtain 1
obtained. 1
over 1
procedure 5
```

Return to EMR screen, make sure your cluster is dead. The meter is running, Select Terminate.

The screenshot shows the Amazon EMR console interface. On the left is a navigation menu with options: Amazon EMR, Cluster list, Security configurations, VPC subnets, Events, and Help. The main area displays a table of clusters. The table has columns: Name, ID, Status, Creation time (UTC-5), Elapsed time, and Normalized instance hours. There are 2 clusters listed: 'My cluster' (ID: j-1ZX4K2HUZF6N, Status: Waiting, Cluster ready) and 'HadoopCoolCluster' (ID: j-WGIPQONRQK0H, Status: Terminated, User request). A 'Terminate clusters' dialog box is open in the foreground, showing a message for 'My cluster' that termination protection is turned on and needs to be turned off. It includes a 'Change' link for termination protection, a warning about data loss, and 'Cancel' and 'Terminate' buttons.

Name	ID	Status	Creation time (UTC-5)	Elapsed time	Normalized instance hours
My cluster	j-1ZX4K2HUZF6N	Waiting Cluster ready	2017-08-23 14:44 (UTC-5)	44 minutes	0
HadoopCoolCluster	j-WGIPQONRQK0H	Terminated User request	2017-08-22 14:37 (UTC-5)	17 minutes	24

**Terminate clusters**

Cluster j-1ZX4K2HUZF6N (My cluster) has termination protection turned on. To terminate this cluster you first need to turn off termination protection.

Termination protection: Off [Change](#)

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

[Cancel](#) [Terminate](#)

## **AWS Command line tools**

You can install AWS command line tools and interact with AWS through a terminal window.

[https://aws.amazon.com/cli/?sc\\_channel=PS&sc\\_campaign=acquisition\\_US&sc\\_publisher=google&sc\\_medium=command\\_line\\_b&sc\\_content=aws\\_cli\\_p&sc\\_detail=aws%20cli&sc\\_category=command\\_line&sc\\_segment=159752350313&sc\\_matchtype=p&sc\\_country=US&s\\_kwcid=AL!4422!3!159752350313!p!!g!!aws%20cli&ef\\_id=WM1YAAAAHwO1Q7Z:20170823201900:s](https://aws.amazon.com/cli/?sc_channel=PS&sc_campaign=acquisition_US&sc_publisher=google&sc_medium=command_line_b&sc_content=aws_cli_p&sc_detail=aws%20cli&sc_category=command_line&sc_segment=159752350313&sc_matchtype=p&sc_country=US&s_kwcid=AL!4422!3!159752350313!p!!g!!aws%20cli&ef_id=WM1YAAAAHwO1Q7Z:20170823201900:s)

## Download Hadoop and run locally

<http://hadoop.apache.org/releases.html>

Download source tar ball.

You should install [Cygwin](#) or better yet, buy a Mac. When installing Cygwin, make sure to select the Bash shell and OpenSSL features to be able to develop and run the MapReduce examples locally on Windows systems.

Hadoop, Hive, and Pig require the JAVA\_HOME environment variable to be set. It is also typically good practice to have Java in the PATH so scripts and applications can easily find it. On a Linux machine, you can use the following command to specify these settings:

```
export JAVA_HOME=/usr/java/latest
export PATH=$PATH:$JAVA_HOME/bin
```

After you install Hadoop, it is convenient to add Hadoop to the path and define a variable that references the location of Hadoop for other scripts and routines that use it. The following example shows these variables being added to the .bash\_profile on a Linux system to define the home location and add Hadoop to the path:

```
$ export HADOOP_INSTALL=/home/user/hadoop-0.20.205.0
$ export PATH=$PATH:$HADOOP_INSTALL/bin
```

You can confirm the installation and setup of Hadoop by running it at the command line. The following example shows running the hadoop command line and the version installed:

```
$ hadoop version
Hadoop 0.20.205.0
Subversion https://svn.apache.org/repos/asf/hadoop/
```

common/branches/branch-0.20-security-205 -r 1179940  
Compiled by hortonfo on Fri Oct 7 06:26:14 UTC 2011