



Big Data Workshop

Amazon Kinesis, Amazon Redshift,
Amazon Elastic Map Reduce

June 9th, 2014

Lab 1: Amazon Kinesis	3
Step 1: Create the Kinesis Stream	3
Step 2: Create the Producer and Visualizer	5
Lab 2: Amazon Redshift	9
Step 1: Create the Redshift Cluster	9
Step 2: Create the Redshift Connector	14
Lab 3: Amazon Elastic Map Reduce	21
Step 1: Create the S3 bucket	21
Step 2: Create the EMR cluster	26
Step 3: Hive	30
Step 4: Pig	33
Step 5: COPY to Redshift	36
Appendix A: Taking it further	37
Appendix B: Architecture Diagrams	38
Amazon Kinesis	38
Amazon Redshift	39
Amazon Elastic Map Reduce	40

Lab 1: Amazon Kinesis

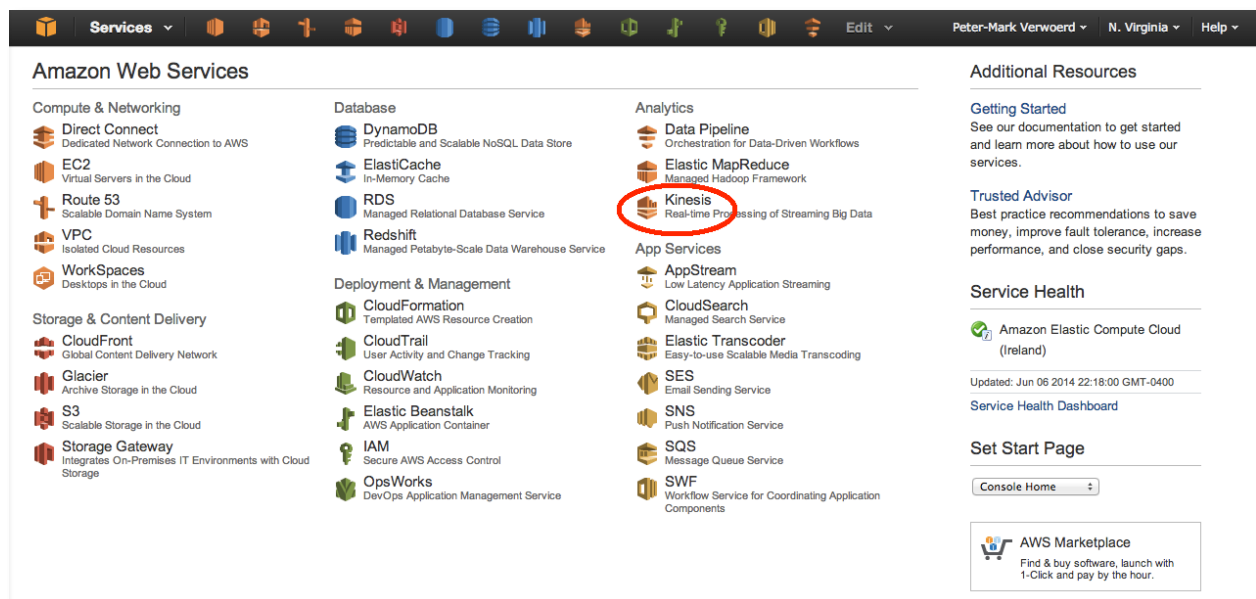
In this first Lab, you will create a Kinesis Stream on the AWS console. Once it is running, you will then use a Cloudformation template to launch an application that will produce random data to feed in to the Kinesis Stream (“producer”) as well as an application using the Kinesis Client Library to consume the data from the Kinesis Stream and visualize it (“visualizer”).

Step 1: Create the Kinesis Stream

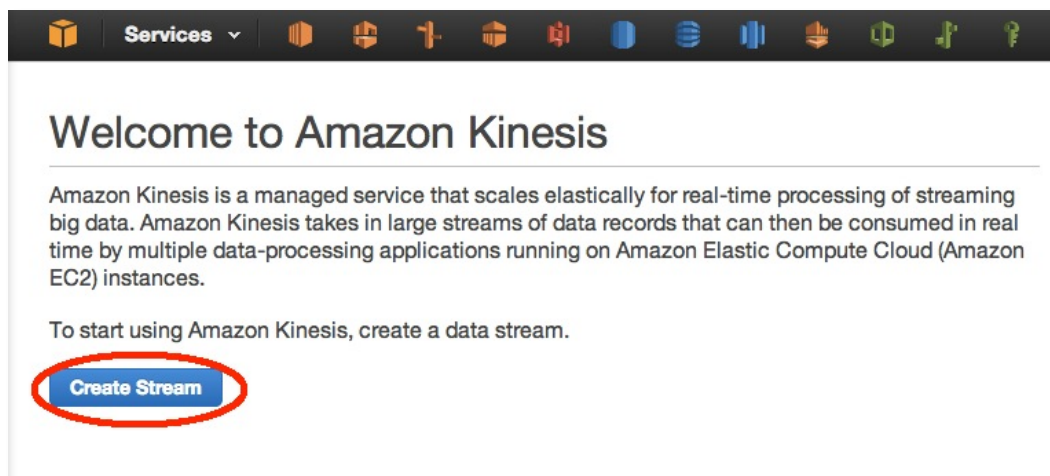
Start at the AWS Console home page:

<https://console.aws.amazon.com/console/home>

From the AWS Console home page, select Kinesis



Click “Create Stream”



Give it any name you wish (remember it for later). Put in 2 shards and click “Create”.

Amazon Kinesis

Create Stream

A stream is composed of multiple shards, each of which provides a fixed unit of capacity. The total capacity of the stream is the sum of the capacities of its shards. Each shard corresponds to 1 MB/s of write capacity and 2 MB/s of read capacity. See the [Amazon Kinesis Developer Guide](#) for more information on estimating number of shards needed for your stream. Note that the cost of the stream is also a function of the number of shards. To learn more about the stream, see the [Amazon Kinesis Pricing Page](#)

Stream Name*

☐ Help me decide how many shards I need

Number of Shards*

Values calculated based on the number of shards entered above:

	Read:	Write:
Total Stream Capacity:	- MB/s	- MB/s
Max Transactions/second:	-	-

The Stream Name identifies the stream and is used to access the data written to the stream

Use the shard calculator to estimate the number of shards needed for the stream

You can change the number of shards in the stream without re-creating the stream

* Required information

Cancel

Create

Step 2: Create the Producer and Visualizer

Here you will create the Kinesis producer and consumer using a Cloudformation template. Click on the link below:

[KinesisPublisherLab](#)

This will open the AWS Console to the Cloudformation page:

Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

Stack

An AWS CloudFormation stack is a collection of related resources that you provision and update as a single unit.

Name

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more.](#)

Source

- ☐ Select a sample template
- ☐ Upload a template to Amazon S3
 - No file chosen
- ☒ Specify an Amazon S3 template URL
 -

Scroll down and click “Next”:

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more.](#)

Source

- ☐ Select a sample template
- ☐ Upload a template to Amazon S3
 - No file chosen
- ☒ Specify an Amazon S3 template URL
 -

On this page, leave the defaults. You don't need to add a key name at this point. Add the Kinesis Stream name you created in step one and click "Next".

Parameters

InstanceType EC2 instance type

KeyName (Optional) Name of an existing EC2 KeyPair to enable SSH access to the instance. If this is not provided you will not be able to SSH on to the EC2 instance.

KinesisStream The name of the Kinesis Stream already created

SSHLocation The IP address range that can be used to SSH to the EC2 instances

Cancel Previous **Next**

You don't have to add tags at this point, but it's a good best practice to do so. You could do something like "Name" for the Key and "Kinesis Lab" for the Value. Many users will further tag with Keys like "Environment" or "User" with respective Values. Once you have entered a tag (if you want), click "Next".

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 10 unique key-value pairs for each stack. [Learn more.](#)

	Key (127 characters maximum)	Value (255 characters maximum)	
1	<input type="text" value="Name"/>	<input type="text" value="Kinesis Lab"/>	<input type="button" value="+"/>

► Advanced

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

Cancel Previous **Next**

On the final page, scroll down and make sure to select the acknowledgement check box. Then click "Create".

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::Policy]

This template might include Identity and Access Management (IAM) resources, which can include groups, IAM users, and IAM roles with certain permissions. Ensure that the template you are using is from a trusted source. [Learn more.](#)

☒ I acknowledge that this template might cause AWS CloudFormation to create IAM resources.

Cancel Previous **Create**

The Cloudformation console will then load and you will see this:

<div> <div>Create Stack</div> <div>Update Stack</div> <div>Delete Stack</div> </div>			
Filter: Active ▾ By Name: <input type="text"/>			
Stack Name	Created Time	Status	Description
<input type="checkbox"/> KinesisPublisherLab	2014-06-07 19:36:25 UTC-0400	CREATE_IN_PROGRESS	The Amazon Kinesis and Data Visualization Lab

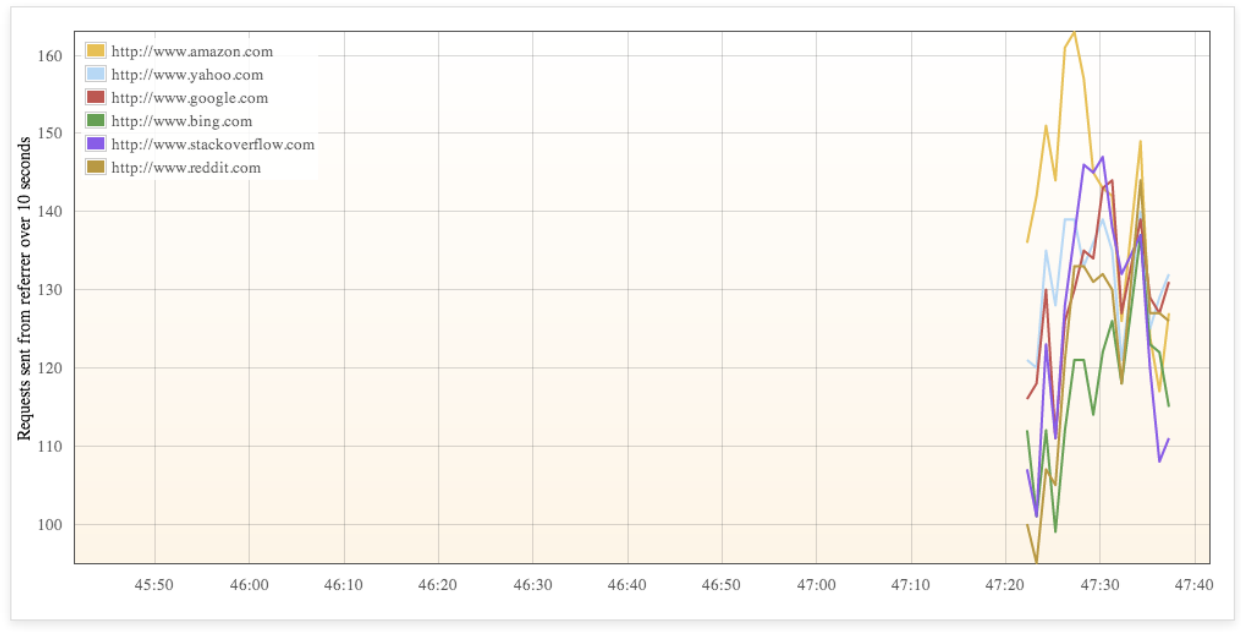
Wait until the status changes from “CREATE_IN_PROGRESS” to “CREATE_COMPLETE” as below:

<div> <div>Create Stack</div> <div>Update Stack</div> <div>Delete Stack</div> </div>			
Filter: Active ▾ By Name: <input type="text"/>			
Stack Name	Created Time	Status	Description
<input type="checkbox"/> KinesisPublisherLab	2014-06-07 19:39:10 UTC-0400	CREATE_COMPLETE	The Amazon Kinesis and Data Visualization Lab

To verify the Cloudformation stack launched successfully, go to the “Outputs” tab and select the URL in the output and open it in a new browser tab/window.

<div> <div>Overview</div> <div>Outputs</div> <div>Resources</div> <div>Events</div> <div>Template</div> <div>Parameters</div> <div>Tags</div> <div>Stack Policy</div> </div>		
Key	Value	Description
URL	http://ec2-██████████.compute-1.amazonaws.com	URL to the sample application's visualization

On the page that you open, you will see a visualization of the data like below:



Congratulations! Kinesis is ingesting data from the producer and the visualizer is reading data from the Kinesis Stream and displaying it.

Lab 2: Amazon Redshift

In this section you will create an Amazon Redshift database cluster, another EC2 Kinesis Client Library application and use the Kinesis Connector library in that application to write to the Redshift database. If you haven't already, make sure you install a client that will be able to communicate with Redshift. SQL Workbench/J is recommended as it will work on any system with Java. The instructions for installing it are here:

<http://docs.aws.amazon.com/redshift/latest/mgmt/connecting-using-workbench.html>

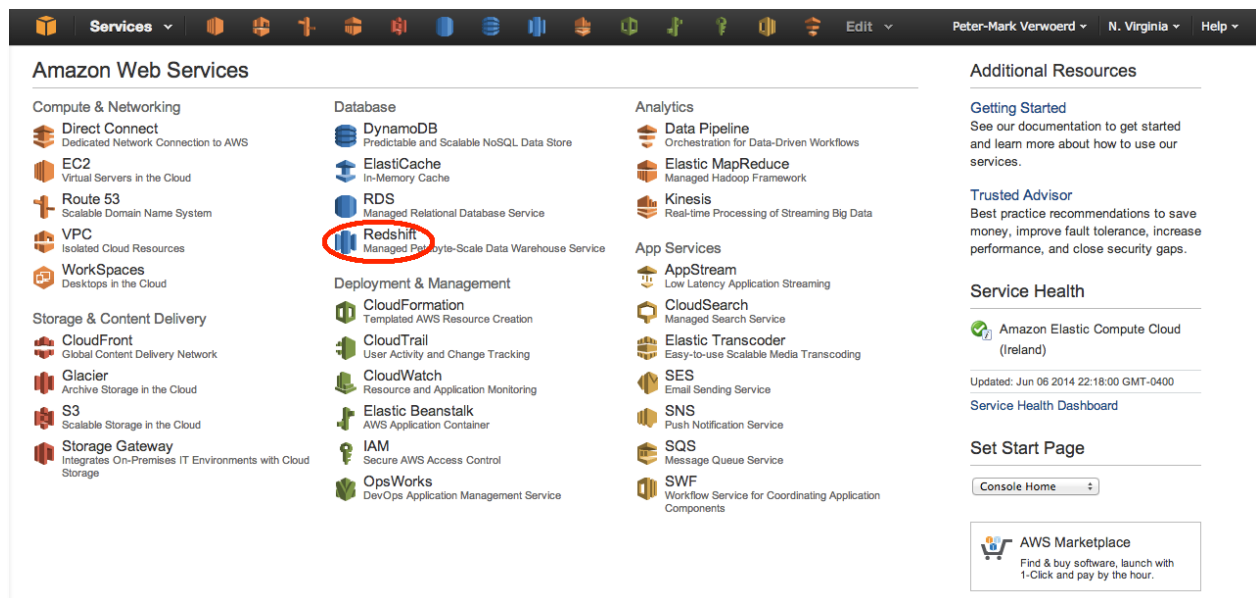
You may be required to download the Postgres JDBC Driver. It is available here:

<http://docs.aws.amazon.com/redshift/latest/mgmt/configure-jdbc-connection.html>

You will be able to test if it has been successfully installed at the end of Step 1.


Step 1: Create the Redshift Cluster

Go back to the Amazon Web Services console main page and select Amazon Redshift:



Click “Launch Cluster”

[Amazon Redshift](#)
[Clusters](#)
[Snapshots](#)
[Security](#)
[Parameter Groups](#)
[Reserved Nodes](#)
[Events](#)



Welcome to Amazon Redshift

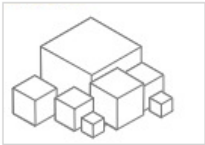
You do not appear to have any clusters in the US East (N. Virginia) region.

Amazon Redshift is a fast and powerful, fully managed, petabyte-scale data warehouse service in the cloud. Amazon Redshift offers you fast query performance when analyzing virtually any size data set using the same SQL-based tools and business intelligence applications you use today. With a few clicks in the AWS Management Console, you can launch a Redshift cluster, starting with a few hundred gigabytes of data and scaling to a petabyte or more, for under \$1,000 per terabyte per year.

Launch Cluster

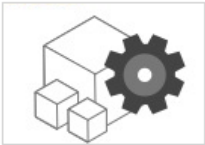
Get up and running immediately

Create Cluster



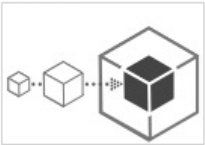
[Learn More](#)

Manage & Configure



[Learn More](#)

Load & Query Data



[Learn More](#)

On this page you will set the parameters necessary to start your cluster. Give your cluster an identifier - any will do, within the limits described on the page. Then give a database name. Though it's optional, you may want to give it a name you find it easy to remember. You may change the port from the default port if you wish or if your laptop will block connections on 5439 (this can happen). Port 8192 is a common alternative. Provide the username and password within the limits described. Be sure to remember your password - it won't be displayed anywhere. Click "Continue".

CLUSTER DETAILS

NODE CONFIGURATION

ADDITIONAL CONFIGURATION

REVIEW

Provide the details of your cluster. Fields marked with * are required.

Cluster Identifier*	<input type="text"/>	This is the unique key that identifies a cluster. This parameter is stored as a lowercase string. (e.g. my-dw-instance)
Database Name	<input type="text"/>	Optional. A default database named dev is created for the cluster. Optionally, specify a custom database name (e.g. mydb) to create an additional database.
Database Port*	<input type="text" value="5439"/>	Port number on which the database accepts connections.
Master User Name*	<input type="text"/>	Name of master user for your cluster. (e.g. awsuser)
Master User Password*	<input type="password"/>	Password must contain 8 to 64 printable ASCII characters excluding: /, ", ', \, and @. It must contain 1 uppercase letter, 1 lowercase letter, and 1 number.
Confirm Password*	<input type="password"/>	Confirm Master User Password.

Cancel

Continue

On the next page, you will be able to select the size and type of your cluster. The 4 instance type options are: dw1.xlarge, dw1.8xlarge, dw2.large, dw2.8xlarge. The difference between the dw1 and dw2 types is the type of storage. The dw1 family have magnetic hard drives while the dw2 family have SSD, or flash, hard drives. For this lab, you should select the dw2.large. You will also need to add a number of nodes of the cluster. 2 nodes are more than sufficient for the work you will be doing. Once entered, click “Continue”.

CLUSTER DETAILS **NODE CONFIGURATION** ADDITIONAL CONFIGURATION REVIEW

Choose a number of nodes and Node Type below. Number of Compute Nodes is required for multi-node clusters.

Node Type dw1.xlarge Specifies the compute, memory, storage, and I/O capacity of the cluster's nodes.

CPU 4.4 EC2 Compute Units (2 virtual cores) per node

Memory 15 GiB per node

Storage 2TB HDD storage per node

I/O Performance Moderate

Cluster Type Multi Node

Number of Compute Nodes* Compute nodes store your data and execute your queries. In addition to your compute nodes, a leader node will be added to your cluster, free of charge. The leader node is the access point for ODBC/JDBC and generates the query plans executed on the compute nodes.

Maximum 32

Minimum 2

Cancel Previous Continue

On the next page, scroll down to the bottom. Change the “Create CloudWatch Alarm” option from Yes to No. When using Redshift normally, you should leave this enabled. An alarm that will alert you between when disk usage is between 70% - 80% is a good best practice. But since this lab does not come close to that, and it will slow down the process, deselect if for now.

Optionally, create a basic alarm for this cluster.

Create CloudWatch Alarm Yes Create a CloudWatch alarm to monitor the disk usage of your cluster.

Disk Usage Threshold 80% Threshold at which the alarm will trigger when disk usage across all nodes reaches this percentage.

Use Existing Topic No Use an existing SNS topic or create a new one. SNS is a Simple Notification Service which will send email notifications to the recipients of the SNS topic when the alarm triggers.

Topic test-default-alarms Name of the SNS topic that will be created.

Recipients Recipients of this SNS topic. If you have multiple recipients, separate the recipients with a comma.

Cancel Previous Continue

When you've changed it to "No", then click "Continue".

Optionally, create a basic alarm for this cluster.

Create CloudWatch Alarm Create a CloudWatch alarm to monitor the disk usage of your cluster.

Scroll to the bottom of the next page. If you've selected 2 dw2.large instances for your cluster, you will see the same price as in the screen shot below. Click "Create Cluster".

⚠ You will start accruing charges as soon as your cluster is active.
 Amazon Redshift is not part of the AWS Free Tier. The on-demand hourly rate for this cluster will be **\$0.50**. If you have purchased Reserved Nodes in this region for this node type which are currently active your rate will differ. For more information see [Amazon Redshift Pricing](#) and [Reserved Nodes Documentation](#).

Once you return to the list of Redshift Clusters, the status of your new cluster will be "creating". The creation process can take several minutes.

Clusters

Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
redshift-lab	creating	unknown	unknown	0

Once it becomes "available", as below, finish the walkthrough from the documentation to create a JDBC connection to your cluster. Once you have successfully connected, leave the connection open and proceed to Step 2.

Clusters

Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
redshift-lab	available	healthy	no	1

Step 2: Create the Redshift Connector

In this section, as in Step 2 in the first lab, you will use CloudFormation to launch the Redshift connector on EC2. Click on the link below:

[RedshiftConnectorLab](#)

to begin.

This will open the AWS Console to the CloudFormation page.

Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

Stack

An AWS CloudFormation stack is a collection of related resources that you provision and update as a single unit.

Name

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more.](#)

Source ☐ Select a sample template

☐ Upload a template to Amazon S3

No file chosen

☒ Specify an Amazon S3 template URL

Scroll down and click

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more.](#)

Source ☐ Select a sample template

☐ Upload a template to Amazon S3

No file chosen

☒ Specify an Amazon S3 template URL

[Cancel](#) [Next](#)

On this page, you will configure everything that the Redshift connector application needs to connect to the Redshift cluster. You will not need to change the Instance Type, add a KeyName, or change the SSH Location. If you are familiar with any of those parameters however, feel free to change them.

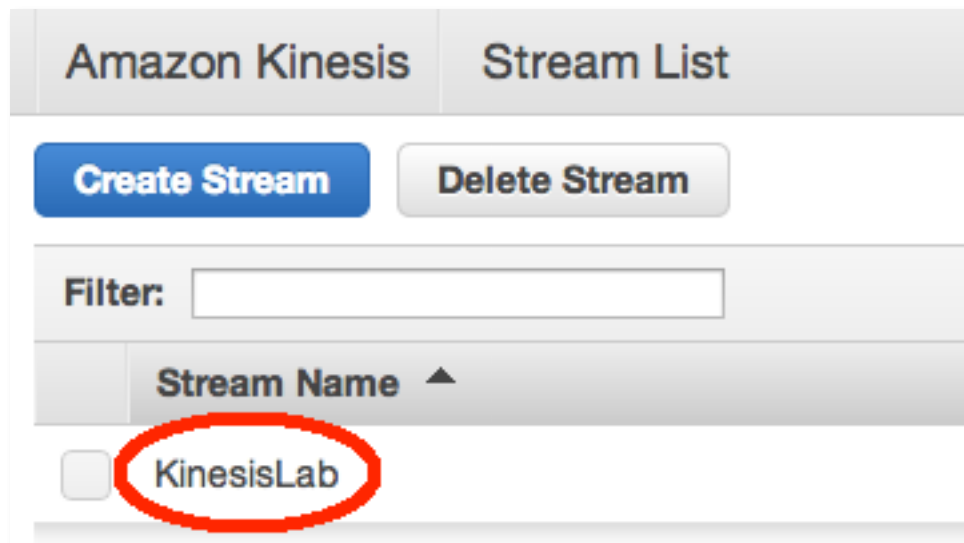
Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

Parameters

InstanceType	<input type="text" value="t1.micro"/>	EC2 instance type
KeyName	<input type="text"/>	(Optional) Name of an existing EC2 KeyPair to enable SSH access to the instance. If this is not provided you will not be able to SSH on to the EC2 instance.
KinesisStream	<input type="text"/>	The name of the Kinesis Stream already created
RedshiftDatabase	<input type="text"/>	The database of the Redshift cluster already created.
RedshiftPassword	<input type="text"/>	The password associated with the user account for the Redshift cluster already created.
RedshiftURL	<input type="text"/>	The URL for the Redshift cluster already created.
RedshiftUsername	<input type="text"/>	The user name for the Redshift cluster already created.
SSHLocation	<input type="text" value="0.0.0.0/0"/>	The IP address range that can be used to SSH to the EC2 instances.

[Cancel](#)
[Previous](#)
[Next](#)

First, add the name of the Kinesis Stream. If don't remember it, you can open the console in a new tab, and navigate to Kinesis again.



Of course, use the name of the Kinesis Stream you created in the field selected below:

Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

Parameters

InstanceType	<input type="text" value="t1.micro"/>	EC2 Instance type
KeyName	<input type="text"/>	(Optional) Name of an existing EC2 KeyPair to enable SSH access to the instance. If this is not provided you will not be able to SSH on to the EC2 instance.
KinesisStream	<input type="text"/>	The name of the Kinesis Stream already created
RedshiftDatabase	<input type="text"/>	The database of the Redshift cluster already created.
RedshiftPassword	<input type="text"/>	The password associated with the user account for the Redshift cluster already created.
RedshiftURL	<input type="text"/>	The URL for the Redshift cluster already created.
RedshiftUsername	<input type="text"/>	The user name for the Redshift cluster already created.
SSHLocation	<input type="text" value="0.0.0.0/0"/>	The IP address range that can be used to SSH to the EC2 instances.

Cancel Previous Next

Next, add the information about your Redshift cluster. From the AWS Console home page, navigate to the Redshift page. Select the cluster you created.

Clusters

[Launch Cluster](#)

Cluster	Cluster Status	DB Health	In Maintenance	Recent E
redshift-lab	available	healthy	no	1

Scroll down to the middle of the page so you can see the “Cluster Database Properties”. The name that you need to add to the Cloudformation template is the “Database Name” - importantly, **not** the Cluster Name. The “Redshift URL” is the “Endpoint”. And the “Redshift Username” is the “Master Username”.

Cluster Database Properties

Endpoint: `redshift-lab.cpvfdvyaudct.us-east-1.redshift.amazonaws.com`

Port: `8192`

Database Name: `labdatabase`

Master Username: `dbroot`

Encrypted: `No`

JDBC URL: `jdbc:postgresql://redshift-lab.cpvfdvyaudct.us-east-1.redshift.amazonaws.com:8192/labdatabase?tcpKeepAlive=true`

ODBC URL: `Driver={PostgreSQL};
Server=redshift-lab.cpvfdvyaudct.us-east-1.redshift.amazonaws.com;
Database=labdatabase;
UID=dbroot;
PWD=insert_your_master_user_password_here; Port=8192`

Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

Parameters

InstanceType	<input type="text" value="t1.micro"/>	EC2 Instance type
KeyName	<input type="text"/>	(Optional) Name of an existing EC2 KeyPair to enable SSH access to the instance. If this is not provided you will not be able to SSH on to the EC2 instance.
KinesisStream	<input type="text"/>	The name of the Kinesis Stream already created
RedshiftDatabase	<input type="text"/>	The database of the Redshift cluster already created.
RedshiftPassword	<input type="text"/>	The password associated with the user account for the Redshift cluster already created.
RedshiftURL	<input type="text"/>	The URL for the Redshift cluster already created.
RedshiftUsername	<input type="text"/>	The user name for the Redshift cluster already created.
SSHLocation	<input type="text" value="0.0.0.0/0"/>	The IP address range that can be used to SSH to the EC2 instances.

[Cancel](#) [Previous](#) [Next](#)

The Redshift Password is the password you provided when you created the Redshift cluster. After you add it, click “Next”.

Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

Parameters

InstanceType	<input type="text" value="t1.micro"/>	EC2 Instance type
KeyName	<input type="text"/>	(Optional) Name of an existing EC2 KeyPair to enable SSH access to the instance. If this is not provided you will not be able to SSH on to the EC2 instance.
KinesisStream	<input type="text"/>	The name of the Kinesis Stream already created
RedshiftDatabase	<input type="text"/>	The database of the Redshift cluster already created.
RedshiftPassword	<input type="password"/>	The password associated with the user account for the Redshift cluster already created.
RedshiftURL	<input type="text"/>	The URL for the Redshift cluster already created.
RedshiftUsername	<input type="text"/>	The user name for the Redshift cluster already created.
SSHLocation	<input type="text" value="0.0.0.0/0"/>	The IP address range that can be used to SSH to the EC2 instances.

Cancel Previous **Next**

As before, it is not necessary to add tags to this Cloudformation stack, but again, it is a good practice to do so. Once you have done so, or not, click “Next”.

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 10 unique key-value pairs for each stack. [Learn more.](#)

	Key (127 characters maximum)	Value (255 characters maximum)	
1	<input type="text" value="Name"/>	<input type="text" value="Redshift Lab"/>	<input type="button" value="+"/>

► Advanced

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

Cancel Previous **Next**

On the final page, don't forget to click to acknowledge the creation of IAM permissions and then click "Create"

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::User, AWS::IAM::Policy, AWS::IAM::AccessKey]

This template might include Identity and Access Management (IAM) resources, which can include groups, IAM users, and IAM roles with certain permissions. Ensure that the template you are using is from a trusted source. [Learn more.](#)

☐ I acknowledge that this template might cause AWS CloudFormation to create IAM resources.

[Cancel](#)
[Previous](#)
[Create](#)

You will then return to the Cloudformation console and the status of your Cloudformation stack will be "CREATE_IN_PROGRESS".

Create Stack

Update Stack

Delete Stack

Filter: Active

By Name:

	Stack Name	Created Time	Status	Description
<input type="checkbox"/>	RedshiftConnectorLab	2014-06-07 22:31:44 UTC-0400	CREATE_IN_PROGRESS	The Amazon Kinesis to Amazon Redshift Lab

Once the status has gone to "CREATE_COMPLETE", switch to your SQL client.

Create Stack

Update Stack

Delete Stack

Filter: Active

By Name:

	Stack Name	Created Time	Status	Description
<input type="checkbox"/>	RedshiftConnectorLab	2014-06-07 22:31:44 UTC-0400	CREATE_COMPLETE	The Amazon Kinesis to Amazon Redshift Lab

In the SQL client, run the following statement:

```
SELECT referrer, count(*) FROM Kinesisbasictable WHERE resource LIKE '%index%' GROUP BY referrer;
```

You should get results in the form of the following:

referrer	count
http://www.google.com	167930
http://www.bing.com	166582
http://www.reddit.com	168990
http://www.amazon.com	166784
http://www.yahoo.com	168074
http://www.stackoverflow.com	166942

Try running the query again a minute or so later to make sure the numbers are increasing:

referrer	count
http://www.reddit.com	172329
http://www.bing.com	169903
http://www.stackoverflow.com	170530
http://www.yahoo.com	171567
http://www.amazon.com	170403
http://www.google.com	171325

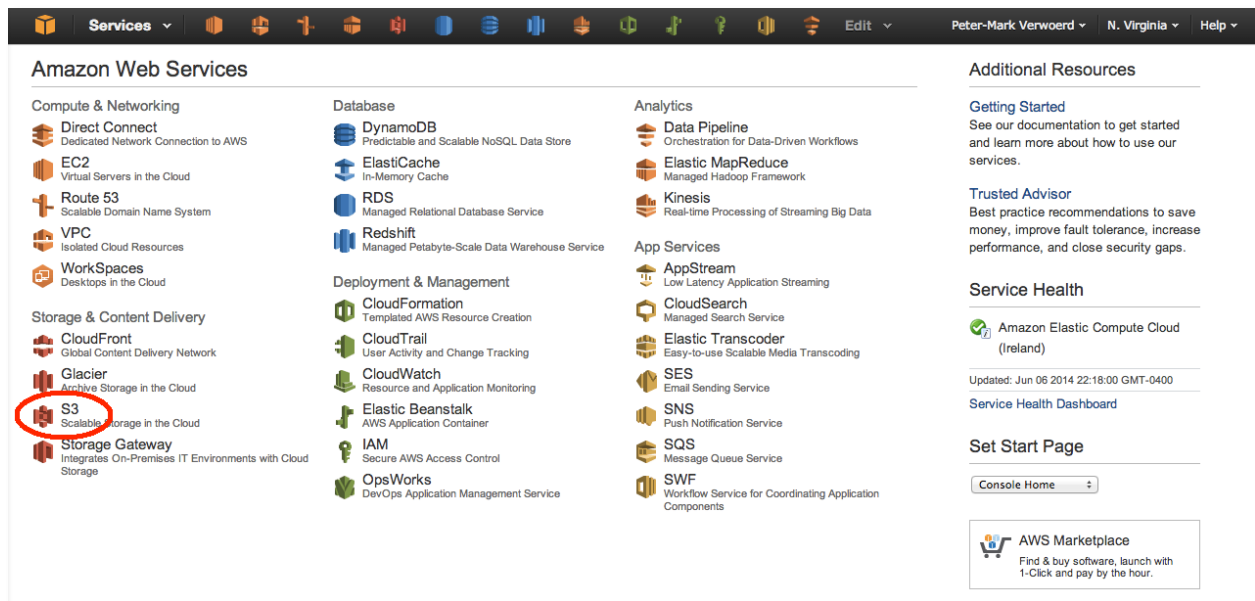
If the numbers are increasing, then congratulations! You now have a producer sending data to a Kinesis Stream, being read by a Kinesis Client Library enabled application, which is using the Kinesis Redshift Connector to write to a Redshift data warehouse. Feel free to experiment with querying the data in the Kinesisbasictable.

Lab 3: Amazon Elastic Map Reduce

In this section you will create an S3 bucket to keep a small amount of fake “user” data. Then you will create an Elastic Map Reduce (EMR) cluster to analyze and transform the user data. Then add the transformed data to your existing Redshift cluster.

Step 1: Create the S3 bucket

Go to the AWS Console home page and navigate to the S3 page.



Here, click “Create Bucket”



Give the bucket any name you wish, just make sure it only contains letters, numbers, hyphens or periods. Also make sure the region you select is “US-Standard”. For historical reasons, this is a different name than the rest of the region in US-East, but it is the corresponding S3 region for US East. Also for historical reasons, US Standard allows a wider variety of names for its buckets than the rest of the regions. For the purposes of this lab and for best practices, it’s recommended to use the stricter naming policy. Logging isn’t necessary at this point. Once you’ve named your bucket, click “Create”.

Create a Bucket - Select a Bucket Name and RegionCancel

A bucket is a container for objects stored in Amazon S3. When creating a bucket, you can choose a Region to optimize for latency, minimize costs, or address regulatory requirements. For more information regarding bucket naming conventions, please visit the [Amazon S3 documentation](#).

Bucket Name:

Region:

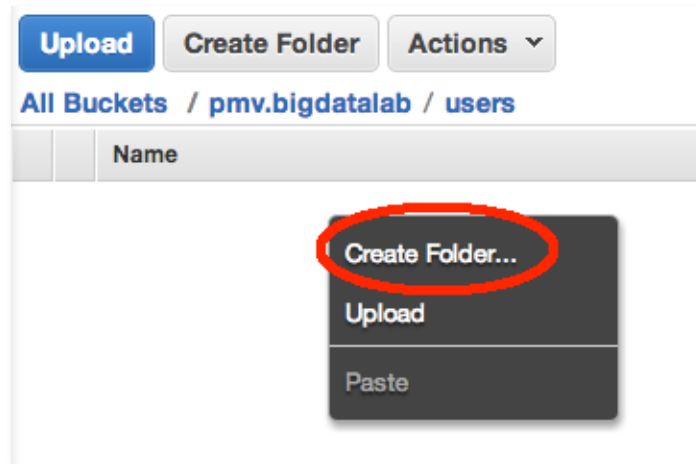
[Set Up Logging >](#) **Create** [Cancel](#)

On the S3 console, select the bucket you've just created.

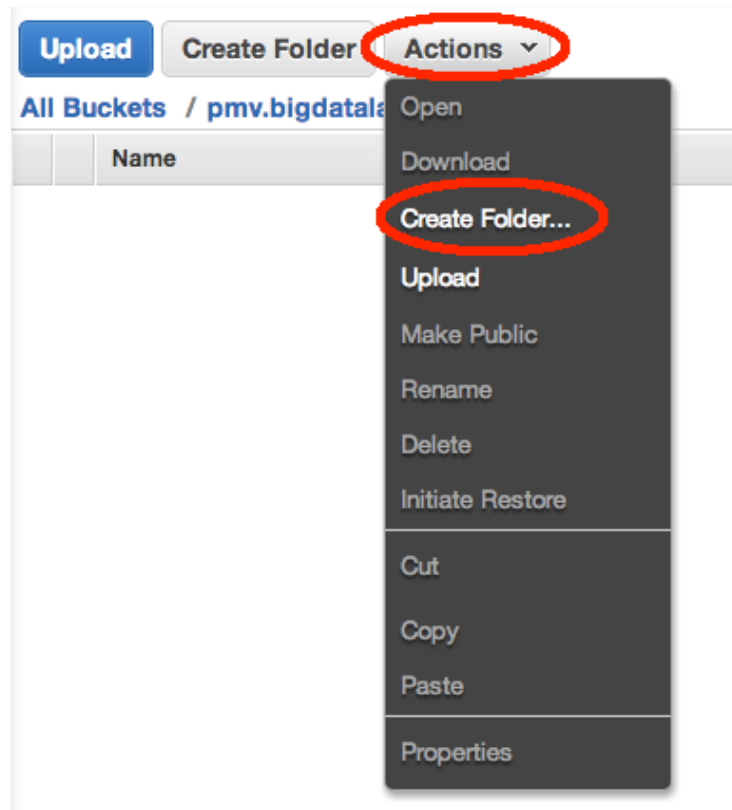
The screenshot shows the AWS S3 console interface. At the top, there are buttons for 'Create Bucket' and 'Actions'. Below this is the 'All Buckets' section, which displays a list of buckets. The bucket 'pmv.bigdatalab' is highlighted in blue, and its name is circled in red. Below the bucket list, there is a section for the selected bucket 'pmv.bigdatalab'. This section includes buttons for 'Upload', 'Create Folder', and 'Actions', as well as tabs for 'None', 'Properties', and 'Transfers'. A message states 'The bucket 'pmv.bigdatalab' is empty'.

Name	Storage Class	Size	Last Modified
The bucket 'pmv.bigdatalab' is empty			

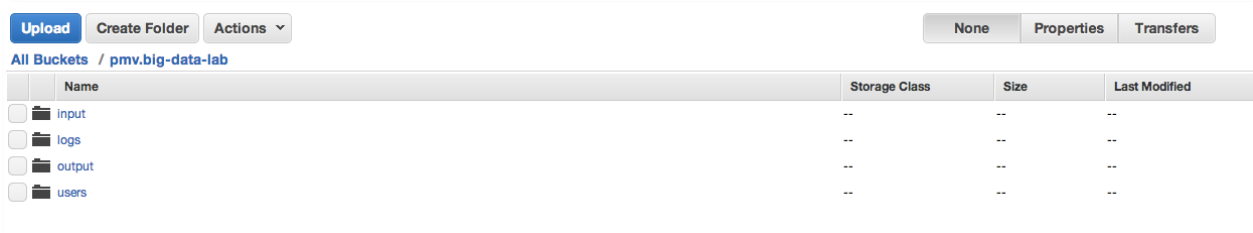
In that bucket, create 4 folders. To create a folder, you can either right click



or use the menu at the top left and select the “Create folder” option.



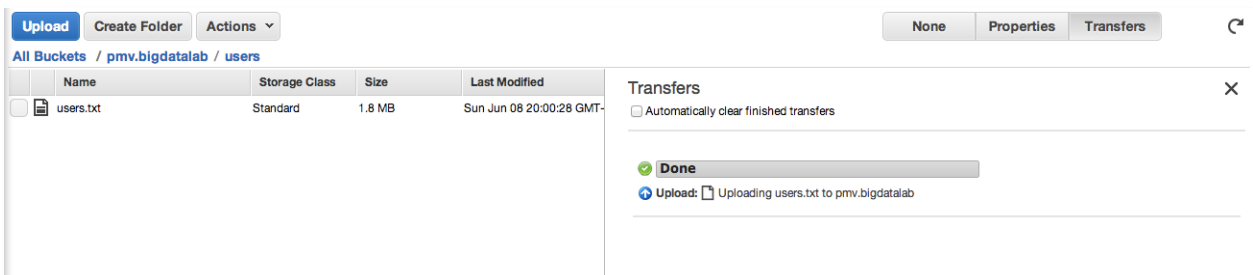
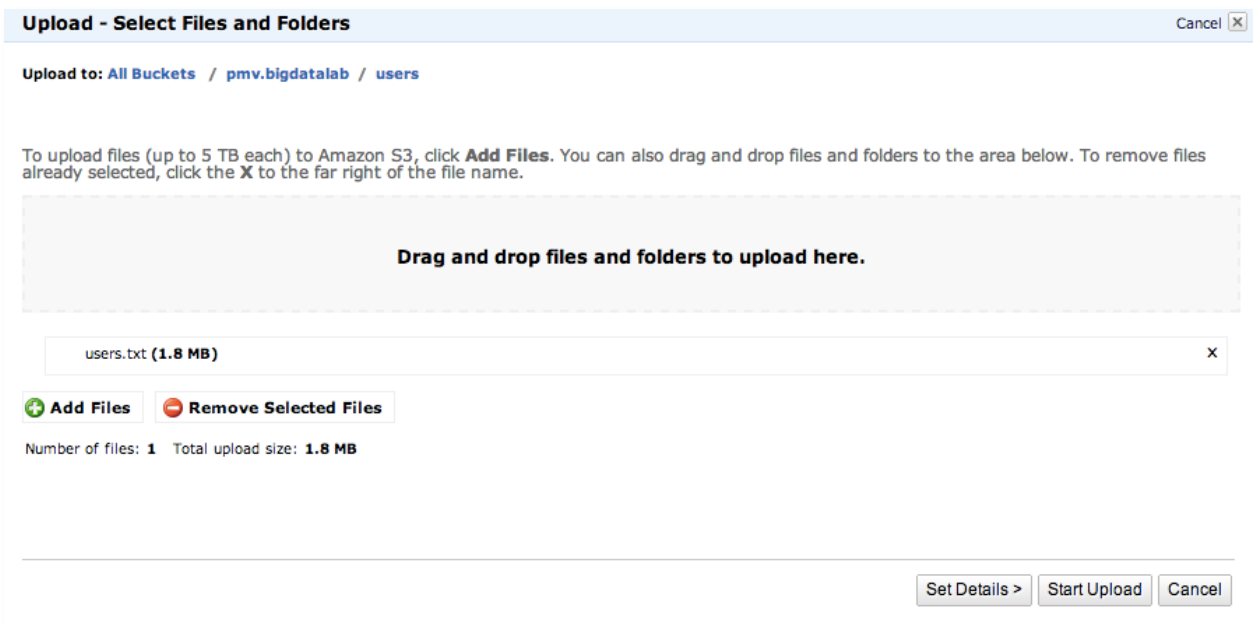
The 4 folders you should make are “input”, “logs”, “users”, and “output”.



Next, download the following link to your local desktop - it's the fake user data you will be analyzing in this part:

bit.ly/awsbdlab2

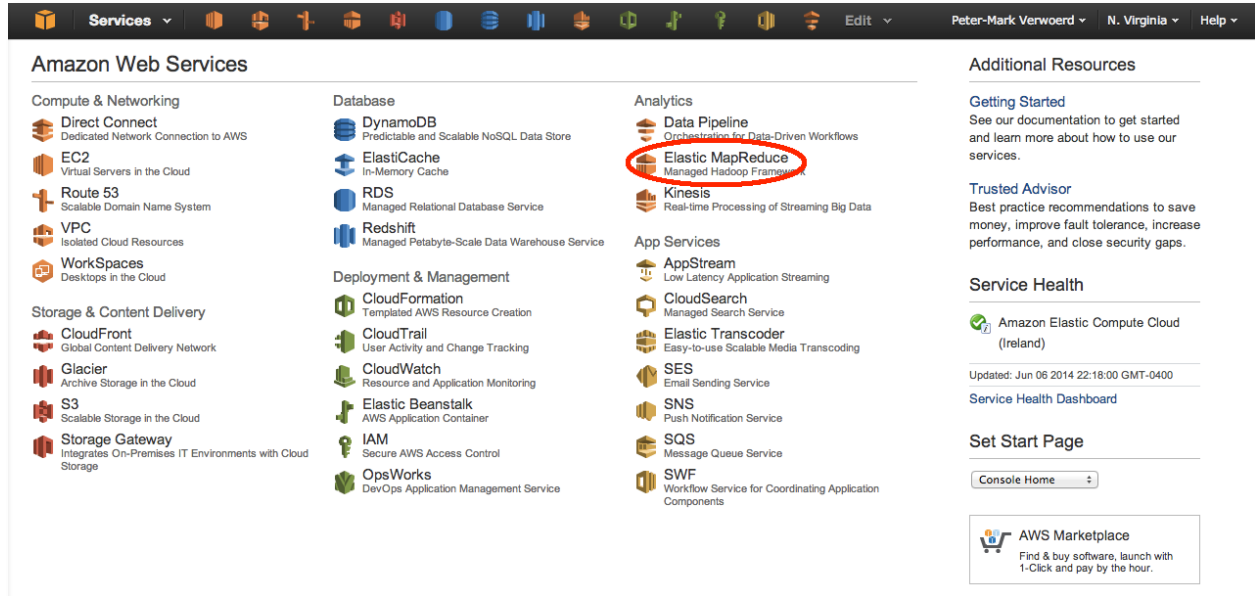
Now, add it to the “users” folder in your S3 bucket.



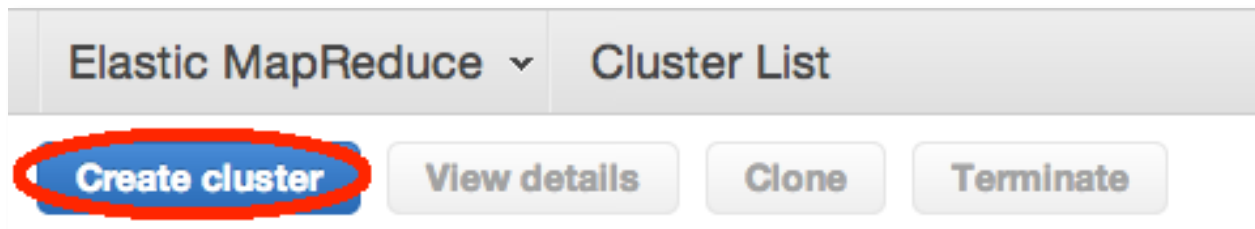
Your S3 bucket is set up and ready to use now.

Step 2: Create the EMR cluster

Go to the AWS Console home page and navigate to the EMR page.



On the EMR console page, click “Create cluster”



On this page you will define your EMR cluster. Give it any name you wish. You will not need Termination protection for this lab, so you can switch that option. Keep logging enabled - this is a best practice as well as helping debugging, if necessary. You can select the bucket you created in Step 1, in the logging folder. As before, you don't have to use tagging, but it is a best practice that is recommended.

Cluster Configuration Configure sample application

Cluster name

Termination protection ☒ Yes ☐ No
Prevents accidental termination of the cluster: to shut down the cluster, you must turn off termination protection. [Learn more](#)

Logging ☒ Enabled
Copy the cluster's log files automatically to S3. [Learn more](#)

Log folder S3 location

s3://<bucket-name>/<folder>/

Debugging ☒ Enabled
Index logs to enable console debugging functionality (requires logging). [Learn more](#)

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

The rest of the setup is fine with the defaults, but just to go over what's there. The first section below Tagging is the software configuration. The AMI version is the Amazon release of each version of Hadoop that we support. Currently, the default is Hadoop 1.0.3, but there is also support for several versions of Hadoop 2. We also support the MapR Hadoop distribution. Hive and Pig are both installed by default on new clusters. Since you will be using them both, leave them as is. You could also optionally add HBase, a database, and Ganglia, distributed monitoring, here, but they aren't necessary. With Hadoop 2, you could also install Impala, a distributed query engine, but this is also not necessary.

Software Configuration

Hadoop distribution ☒ Amazon Use Amazon's Hadoop distribution. [Learn more](#)

AMI version
 Determines the base configuration of the instances in your cluster, including the Hadoop version. [Learn more](#)

☐ MapR Use MapR's Hadoop distribution. [Learn more](#)

Applications to be installed	Version			
Hive	0.11.0.1			
Pig	0.11.1.1			

Additional applications

Configure and add

Next you can configure the type of cluster in this section, VPC or not, Availability Zones and which type of instances you want to use for your cluster. The defaults here are fine.

Hardware Configuration

i Specify the [networking](#) and [hardware](#) configuration for your cluster. If you need more than 20 EC2 instances, [complete this form](#).
[Request Spot instances](#) (unused EC2 capacity) to save money.

Network Launch into EC2-Classic Use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network. [Create a VPC](#)

EC2 availability zone No preference Launch the cluster in a specific EC2 Availability Zone.

	EC2 instance type	Count	Request spot	
Master	m1.small	1	<input type="checkbox"/>	The Master instance assigns Hadoop tasks to core and task nodes, and monitors their status.
Core	m1.small	2	<input type="checkbox"/>	Core instances run Hadoop tasks and store data using the Hadoop Distributed File System (HDFS).
Task	m1.small	0	<input type="checkbox"/>	Task instances run Hadoop tasks.

Here is where you could add an EC2 key pair if you wanted to log in to any of the instances in your cluster, or add access to IAM users, or add EC2 role to your instances. All the defaults here are fine.

Security and Access

EC2 key pair Proceed without an EC2 key pair Use an existing key pair to SSH into the master node of the Amazon EC2 cluster as the user "hadoop". [Learn more](#)

IAM user access ☐ All other IAM users ☒ No other IAM users Control the visibility of this cluster to other IAM users. [Learn more](#)

EC2 role Proceed without role Control permissions for applications on the cluster. [Learn more](#)

Bootstrap actions are scripts to install on your cluster. We will not need to add anything now.

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
<p>Add bootstrap action Select a bootstrap action</p> <p>Configure and add</p>			

Finally, Steps are scripts you can have Hadoop run when it starts. You will not need any for this lab, so just click “Create cluster”.

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
<div> Add step <div>Select a step</div> <div>Configure and add</div> </div>			

Auto-terminate
☐ Yes
☒ No

Automatically terminate cluster after the last step is completed.

Keep cluster running until you terminate it.

Cancel

Create cluster

Your cluster will be in the “Starting” status. Your cluster will then take a few minutes to start.

Elastic MapReduce
Cluster List > Cluster Details

Add step
Resize
Clone
Terminate

Cluster: My cluster
Starting
Provisioning Amazon EC2 capacity

It will be ready to go when the cluster status is “Waiting”.

Elastic MapReduce
Cluster List > Cluster Details

Add step
Resize
Clone
Terminate

Cluster: My cluster
Waiting
Waiting after step completed

Master public DNS: ec2-75-101-239-196.compute-1.amazonaws.com

Tags: -- View All / Edit

Step 3: Hive

In this step, you will run a Hive script to extract data from the users data file. First, copy the following in to a text editor:

```
ADD JAR s3://pmv.public/emr/jsonserde.jar ;
CREATE EXTERNAL TABLE users (userid int, username string, firstname string,
lastname string, city string, state string, email string, phone string,
likesports string, liketheatre string, likeconcerts string, likejazz string,
likeclassical string, likeopera string, likerock string, likevegas string,
likebroadway string, likemusicals string)
ROW FORMAT
    serde 'com.amazon.elasticmapreduce.JsonSerde'
    with serdeproperties ( 'paths'='userid, username, firstname, lastname,
city, state, email, phone, likesports, liketheatre, likeconcerts, likejazz,
likeclassical, likeopera, likerock, likevegas, likebroadway, likemusicals' )
LOCATION "${INPUT}" ;
INSERT OVERWRITE DIRECTORY "${OUTPUT}"
SELECT * FROM users WHERE likesports = "true";
```

This script first loads a JSON serializer/deserializer (or serde). Then it creates a table in a location referenced by an input, that will be added shortly. It then writes the output of a select to another referenced argument for the output.

Save the file and give it a name along the lines of “users.sql”.

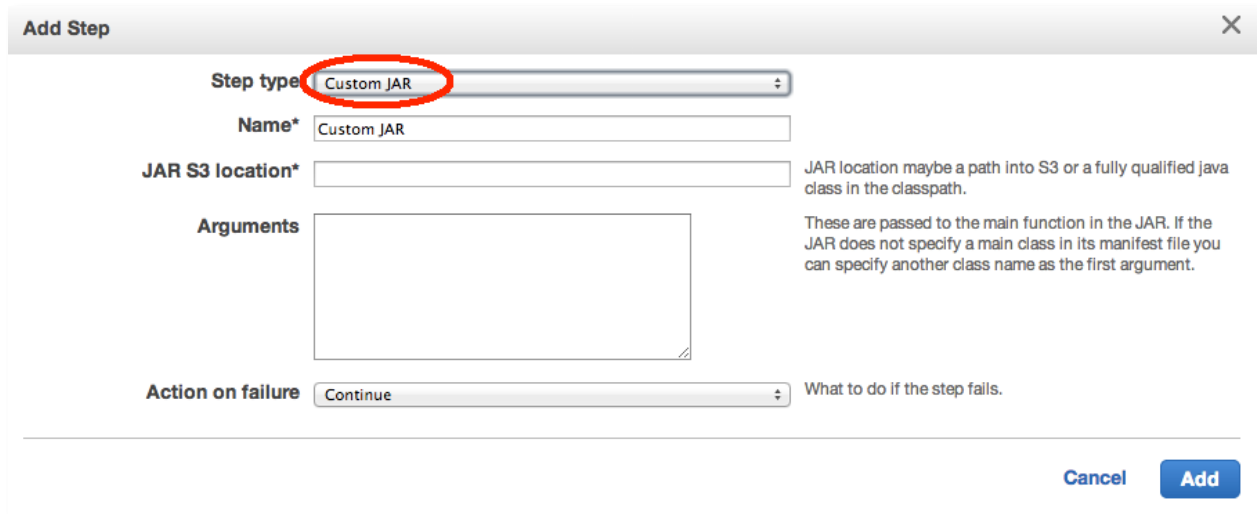
Upload the file to the “input” folder in the S3 bucket you created for the lab.

Then return to the AWS Console page for the EMR cluster you created. Select “Add step”.

The screenshot shows the AWS EMR console interface for a cluster named 'My cluster' in a 'Waiting' state. The top navigation bar includes 'Elastic MapReduce', 'Cluster List', and 'Cluster Details'. Below the navigation bar are buttons for 'Add step', 'Resize', 'Clone', and 'Terminate'. The cluster status is 'Waiting' with a note 'Waiting after step completed'. The console displays various details about the cluster, including Master public DNS, Tags, Summary, Configuration Details, Security/Network, and Hardware. The 'Steps' section is expanded, showing a table with columns for ID, Name, Status, and Action. The 'Add step' button is highlighted with a red circle.

Summary	Configuration Details	Security/Network	Hardware
ID: j-D09I8TFANBGC Creation date: 2014-06-08 21:31 (UTC-4) Elapsed time: 1 hour, 55 minutes Auto-terminate: No Termination protection: Off	AMI version: 2.4.2 Hadoop distribution: Amazon 1.0.3 Applications: Hive 0.11.0.1, Pig 0.11.1.1 Log URI: s3://pmv.log/emr/	Availability zone: us-east-1c Subnet ID: -- Key name: -- EC2 role: -- Visible to all users: None	Master: Running 1 m1.small Core: Running 2 m1.small Task: --

Change the “Step type” to “Hive program”



The screenshot shows the 'Add Step' dialog box. The 'Step type' dropdown is set to 'Custom JAR' and is circled in red. The 'Name' field contains 'Custom JAR'. The 'JAR S3 location*' field is empty. The 'Arguments' field is a large empty text area. The 'Action on failure' dropdown is set to 'Continue'. To the right of the 'JAR S3 location*' field, there is explanatory text: 'JAR location maybe a path into S3 or a fully qualified java class in the classpath.' Below the 'Arguments' field, there is more text: 'These are passed to the main function in the JAR. If the JAR does not specify a main class in its manifest file you can specify another class name as the first argument.' At the bottom right, there are 'Cancel' and 'Add' buttons.

You need to add 3 locations to this step. The first is the location of the script you just uploaded to S3. The format is:

`s3://<YOUR-BUCKET>/input/users.sql`

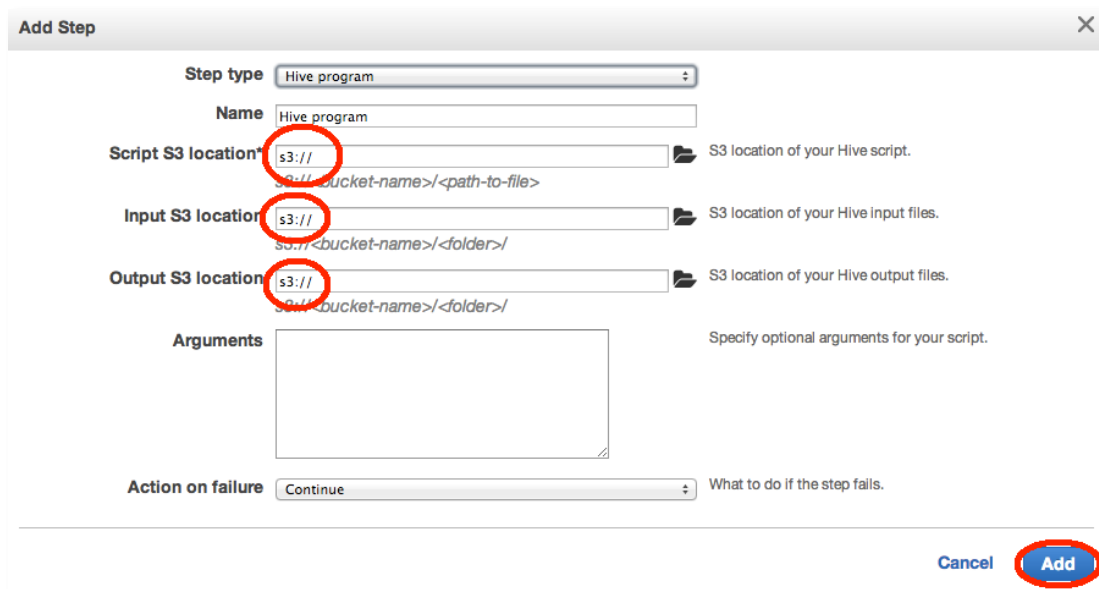
using your bucket name and the name you gave the script. The input location is:

`s3://<YOUR-BUCKET>/users/`

Note that you don't want to specific the file. Hive reads in folders, not files. The output location is:

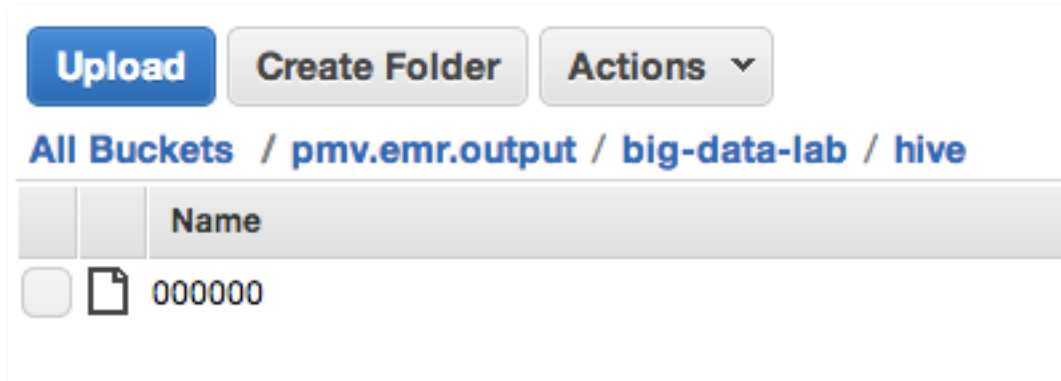
`<s3://<YOUR-BUCKET>/output/hive/`

After you've added the information necessary, click “Add”.



The screenshot shows the 'Add Step' dialog box with 'Step type' set to 'Hive program'. The 'Name' field contains 'Hive program'. The 'Script S3 location*' field is filled with 's3://' and is circled in red. Below it, a placeholder text reads 's3://<bucket-name>/<path-to-file>'. The 'Input S3 location' field is also filled with 's3://' and circled in red, with a placeholder 's3://<bucket-name>/<folder>'. The 'Output S3 location' field is filled with 's3://' and circled in red, with a placeholder 's3://<bucket-name>/<folder>'. The 'Arguments' field is empty. The 'Action on failure' dropdown is set to 'Continue'. To the right of the 'Script S3 location*' field, there is explanatory text: 'S3 location of your Hive script.' Below the 'Input S3 location' field, there is text: 'S3 location of your Hive input files.' Below the 'Output S3 location' field, there is text: 'S3 location of your Hive output files.' Below the 'Arguments' field, there is text: 'Specify optional arguments for your script.' At the bottom right, there are 'Cancel' and 'Add' buttons, with the 'Add' button circled in red.

The EMR cluster will take a minute or so to run the script. Once the step has completed, you can check the location in S3.



Step 4: Pig

In this step, you will run a Pig script to extract data from the users data file. First, copy the following in to a text editor:

```
USERS = LOAD '$INPUT' USING
JsonLoader('userid:int,username:chararray,firstname:chararray,lastname:charar
ray,city:chararray,state:chararray,email:chararray,phone:chararray,likesports
:chararray,liketheatre:chararray,likeconcert:chararray,likejazz:chararray,lik
eclassical:chararray,likeopera:chararray,likerock:chararray,likevegas:chararr
ay,likebroadway:chararray,likemusicals:chararray');
STORE USERS into '$OUTPUT' USING PigStorage('|');
```

This script loads the data from S3 referenced in the input, mapping the fields in the JSON loader. It will then write to the output location, changing the format to pipe delimited.

Save the file and give it a name along the lines of “users.pig”.

Upload the file to the “input” folder in the S3 bucket you created for the lab.

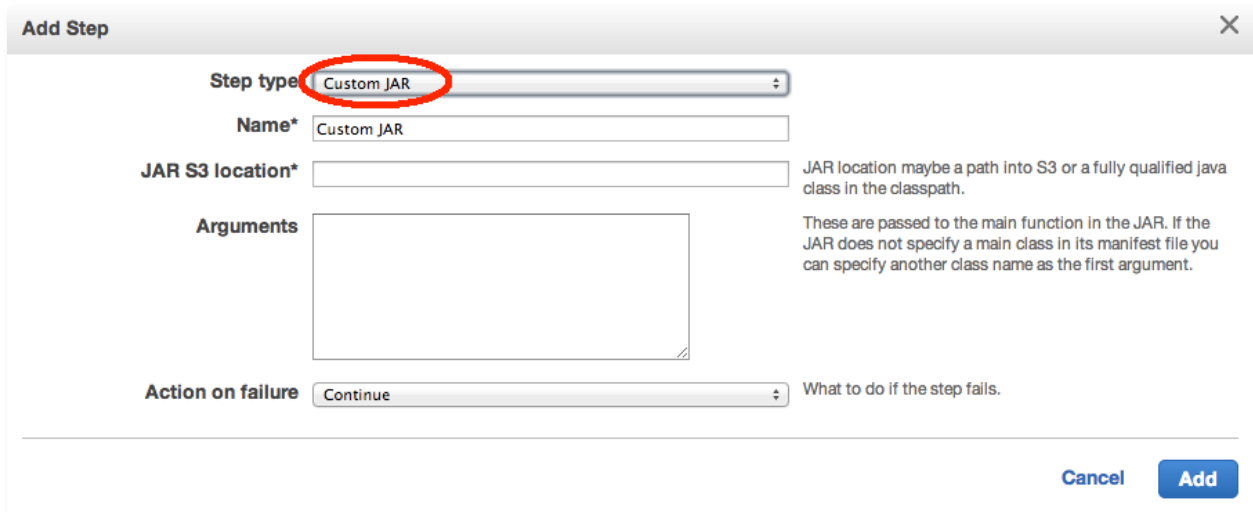
Then return to the AWS Console page for the EMR cluster you created. Select “Add step”.

The screenshot shows the AWS Management Console interface for an EMR cluster. At the top, there are navigation tabs: 'Elastic MapReduce', 'Cluster List', and 'Cluster Details'. The 'Cluster Details' tab is selected. Below the navigation, there are buttons: 'Add step', 'Resize', 'Clone', and 'Terminate'. The cluster name is 'My cluster' and its state is 'Waiting'. The 'Add step' button is circled in red. The console displays various details about the cluster, including its ID, creation date, and configuration details. The 'Steps' section is expanded, showing the 'Add step' button.

Summary	Configuration Details	Security/Network	Hardware
ID: j-D09I8TFANBGC Creation date: 2014-06-08 21:31 (UTC-4) Elapsed time: 1 hour, 55 minutes Auto-terminate: No Termination protection: Off	AMI version: 2.4.2 Hadoop distribution: Amazon 1.0.3 Applications: Hive 0.11.0.1, Pig 0.11.1.1 Log URI: s3://pmv.log/emr/	Availability zone: us-east-1c Subnet ID: -- Key name: -- EC2 role: -- Visible to all users: None	Master: Running 1 m1.small Core: Running 2 m1.small Task: --

Below the details, there are sections for 'Monitoring' and 'Steps'. The 'Steps' section is expanded, showing the 'Add step' button, which is circled in red.

Change the “Step type” to “Pig program”



The screenshot shows the 'Add Step' dialog box. The 'Step type' dropdown is set to 'Custom JAR' and is circled in red. The 'Name' field contains 'Custom JAR'. The 'JAR S3 location*' field is empty. The 'Arguments' field is a large empty text area. The 'Action on failure' dropdown is set to 'Continue'. To the right of the 'JAR S3 location*' field, there is a note: 'JAR location maybe a path into S3 or a fully qualified java class in the classpath.' Below the 'Arguments' field, there is another note: 'These are passed to the main function in the JAR. If the JAR does not specify a main class in its manifest file you can specify another class name as the first argument.' At the bottom right, there are 'Cancel' and 'Add' buttons.

You need to add 3 locations to this step. The first is the location of the script you just uploaded to S3. The format is:

`s3://<YOUR-BUCKET>/input/users.pig`

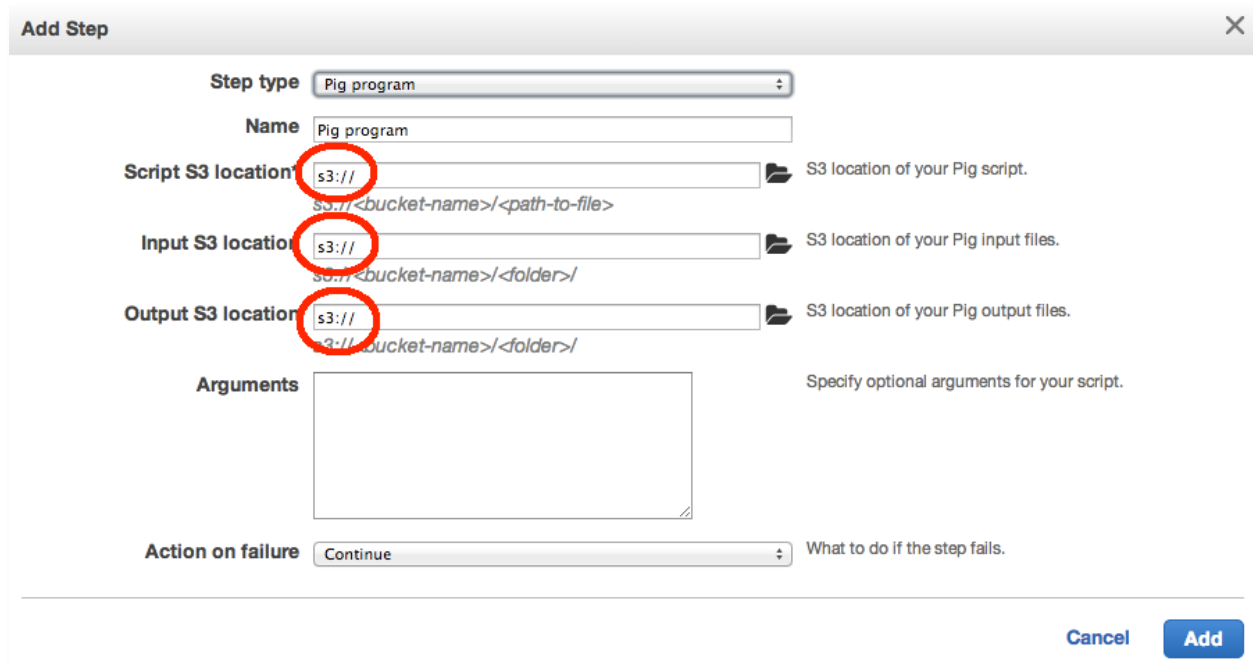
using your bucket name and the name you gave the script. The input location is:

`s3://<YOUR-BUCKET>/users/users.txt`

Note unlike Hive, you need to reference the specific file. The output location is:

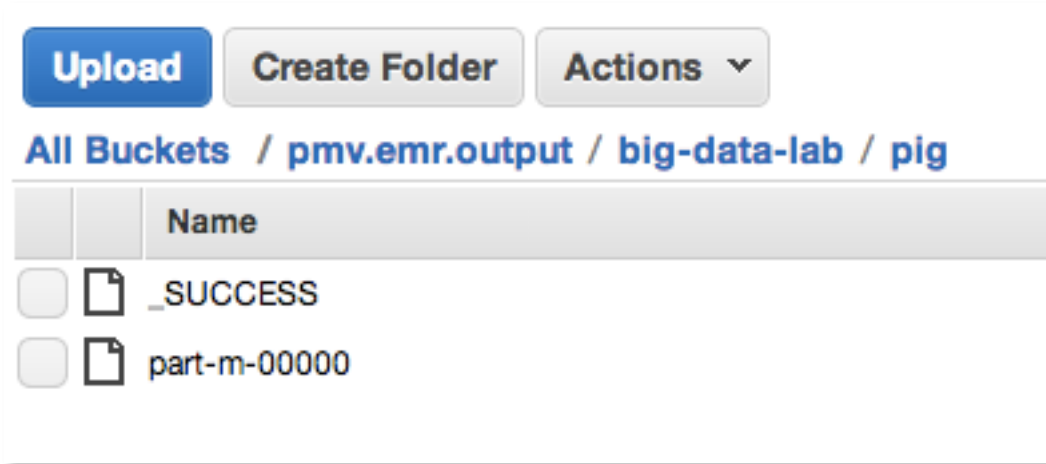
`<s3://<YOUR-BUCKET>/output/pig/`

After you’ve added the information necessary, click “Add”.



The screenshot shows the 'Add Step' dialog box with 'Step type' set to 'Pig program'. The 'Name' field contains 'Pig program'. The 'Script S3 location*' field is filled with 's3://' and is circled in red. Below it, a placeholder text reads 's3://<bucket-name>/<path-to-file>'. The 'Input S3 location*' field is also filled with 's3://' and circled in red, with a placeholder 's3://<bucket-name>/<folder>/'. The 'Output S3 location*' field is filled with 's3://' and circled in red, with a placeholder 's3://<bucket-name>/<folder>/'. To the right of each S3 location field is a folder icon and a description: 'S3 location of your Pig script.', 'S3 location of your Pig input files.', and 'S3 location of your Pig output files.' respectively. The 'Arguments' field is empty. The 'Action on failure' dropdown is set to 'Continue'. At the bottom right, there are 'Cancel' and 'Add' buttons.

The EMR cluster will take a minute or so to run the script. Once the step has completed, you can check the location in S3.



Step 5: COPY to Redshift

In this final step, you will copy the users data that was converted by Pig in to your existing Redshift cluster.

In the SQL client connected to the Redshift cluster, create the table to hold the user data:

```
CREATE TABLE users (  
    userid int NOT NULL PRIMARY KEY,  
    username varchar(100) NOT NULL DISTKEY,  
    firstname varchar(100) NOT NULL SORTKEY,  
    lastname varchar(100) NOT NULL,  
    city varchar(100) NOT NULL,  
    state varchar(100) NOT NULL,  
    email varchar(100) NOT NULL,  
    phone varchar(100) NOT NULL,  
    likesports varchar(10) NOT NULL,  
    liketheatre varchar(10) NOT NULL,  
    likeconcert varchar(10) NOT NULL,  
    likejazz varchar(10) NOT NULL,  
    likeclassical varchar(10) NOT NULL,  
    likeopera varchar(10) NOT NULL,  
    likerock varchar(10) NOT NULL,  
    likevegas varchar(10) NOT NULL,  
    likebroadway varchar(10) NOT NULL,  
    likemusicals varchar(10) NOT NULL  
);
```

Once it's been created, run the COPY command in the SQL client:

```
COPY USERS FROM 's3://pmv.emr.output/big-data-lab/pig/out/part-m-00000'  
CREDENTIALS 'aws_access_key_id=<YOUR-ACCESS-KEY>;aws_secret_access_key=<YOUR-  
SECRET-ACCESS-KEY>' delimiter '|' COMPUPDATE ON;
```

Make sure you add your own AWS access key and secret key.

Using this command, Redshift will use its storage nodes to reach out to S3 in parallel and copy the data to the table. Using the “COMPUPDATE” option will make Redshift optimize the compression of the data on the table as it's being loaded.

The command should finish in a few minutes. Once it's done, you can try running some select statements. You could run the same command that was run in Hive:

```
select * from users where likesports = 'true';
```

This will return the same data set that Hive did. If you run the same statement again, you'll notice that it returns much more quickly. This is because Redshift compiles queries the first time they're run so that the each subsequent time the query is run, they are executed directly on the storage nodes.

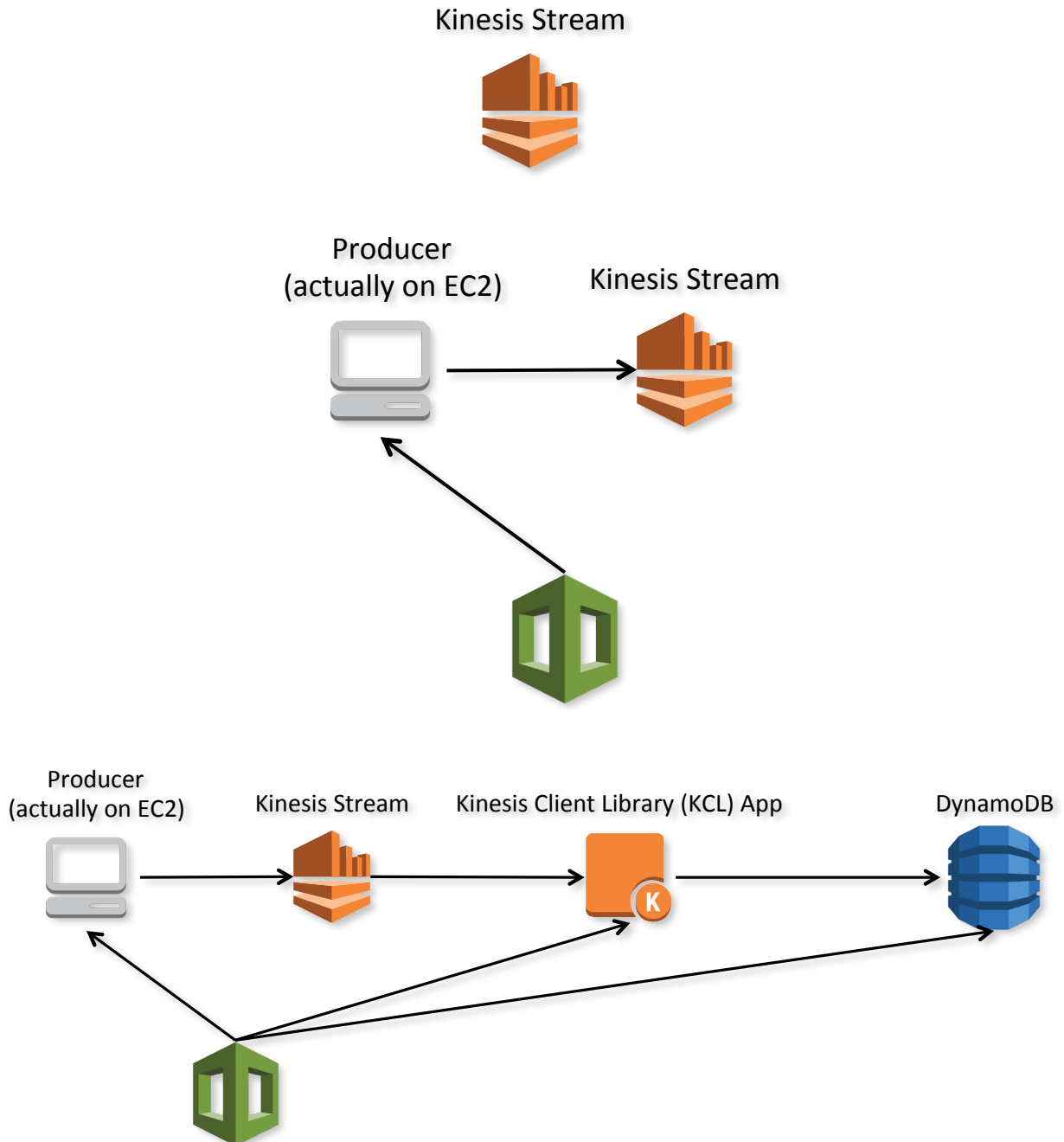
If your queries are returning successfully, congratulations! You've taken data in 1 format on S3, transformed it using EMR and loaded it in to your Redshift data warehouse.

Appendix A: Taking it further

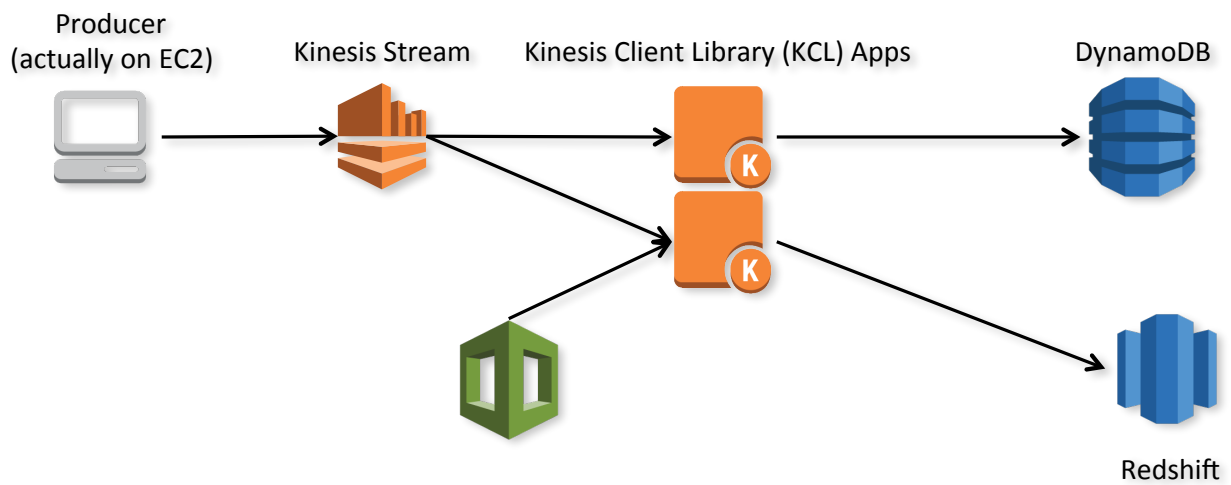
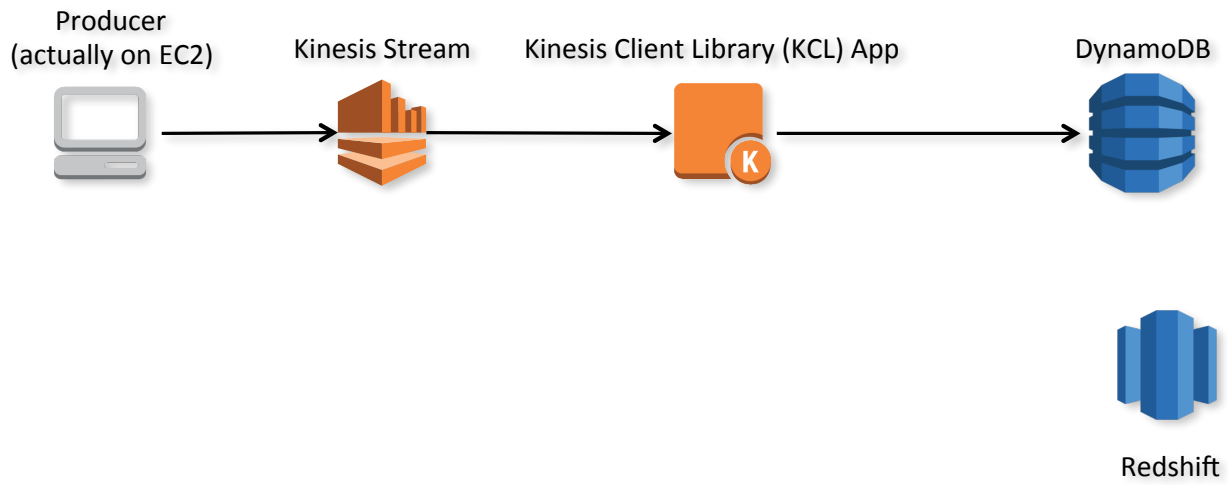
1. Re-write the consumer application from part 1 to write to S3 instead of Redshift directly.
2. Try to run a Business Intelligence (BI) tool on top of your Redshift cluster to visualize query results.
3. Connect to your EMR cluster via the command line and:
 - A. run the Hive script in interactive mode and/or make other queries.
 - B. run the Pig script in interactive mode and/or make other changes to the users.txt file.
4. Use Data Pipeline to create a pipeline that will launch an EMR cluster, run the Pig script and copy the data to the Redshift cluster.
5. Using DynamoDB
 - C. Query the tables using the API
 - D. Query the tables using Hive in EMR
 - E. Export the tables to Redshift using the COPY command.

Appendix B: Architecture Diagrams

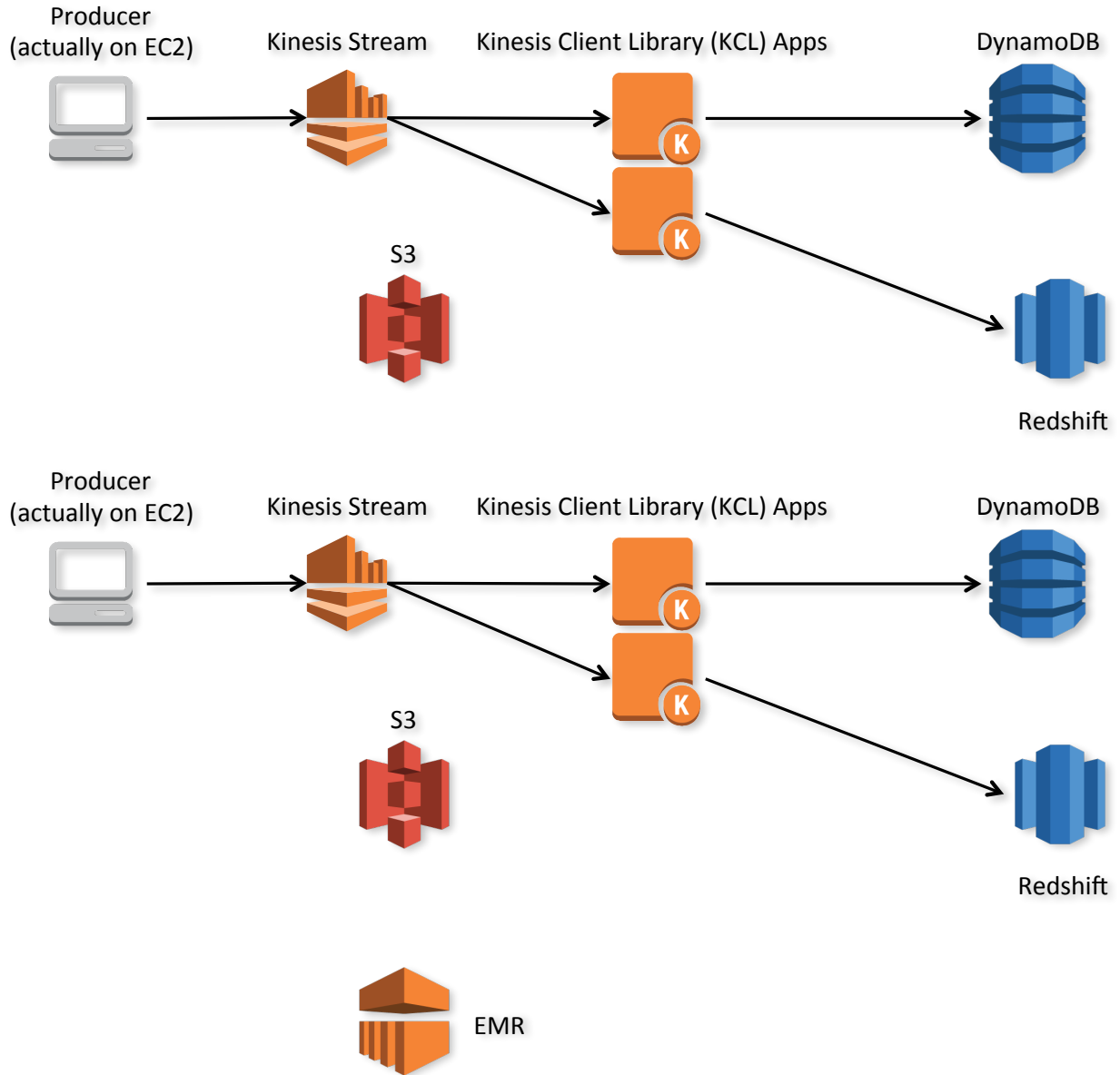
Amazon Kinesis



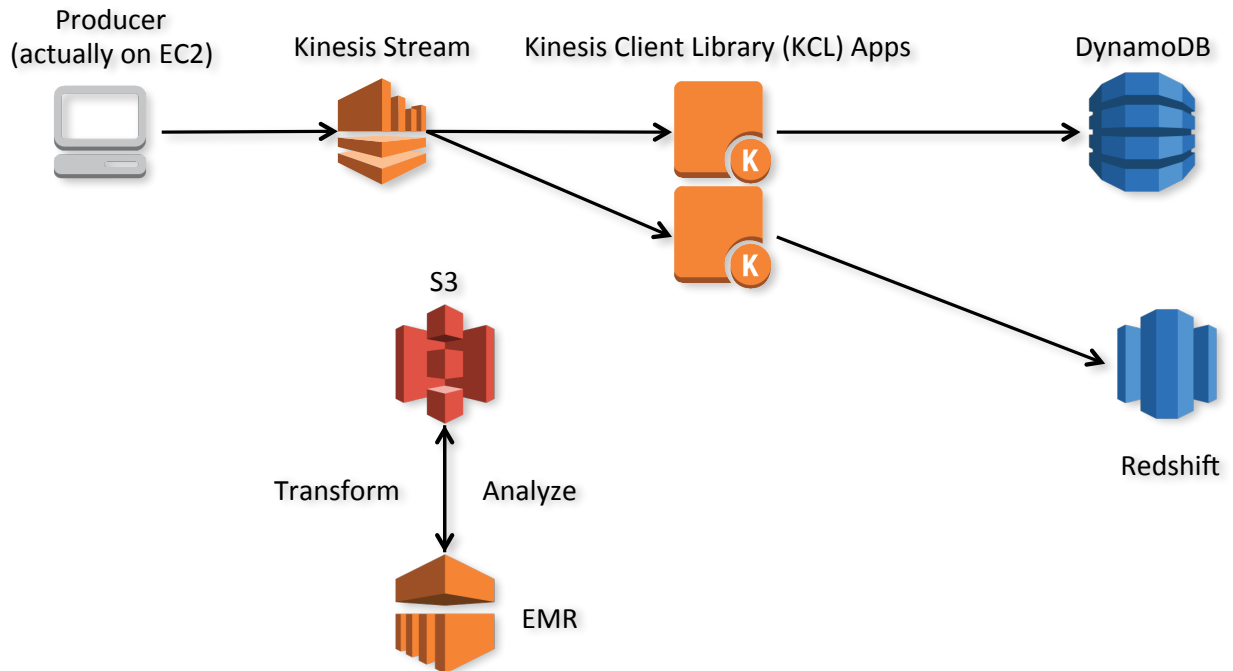
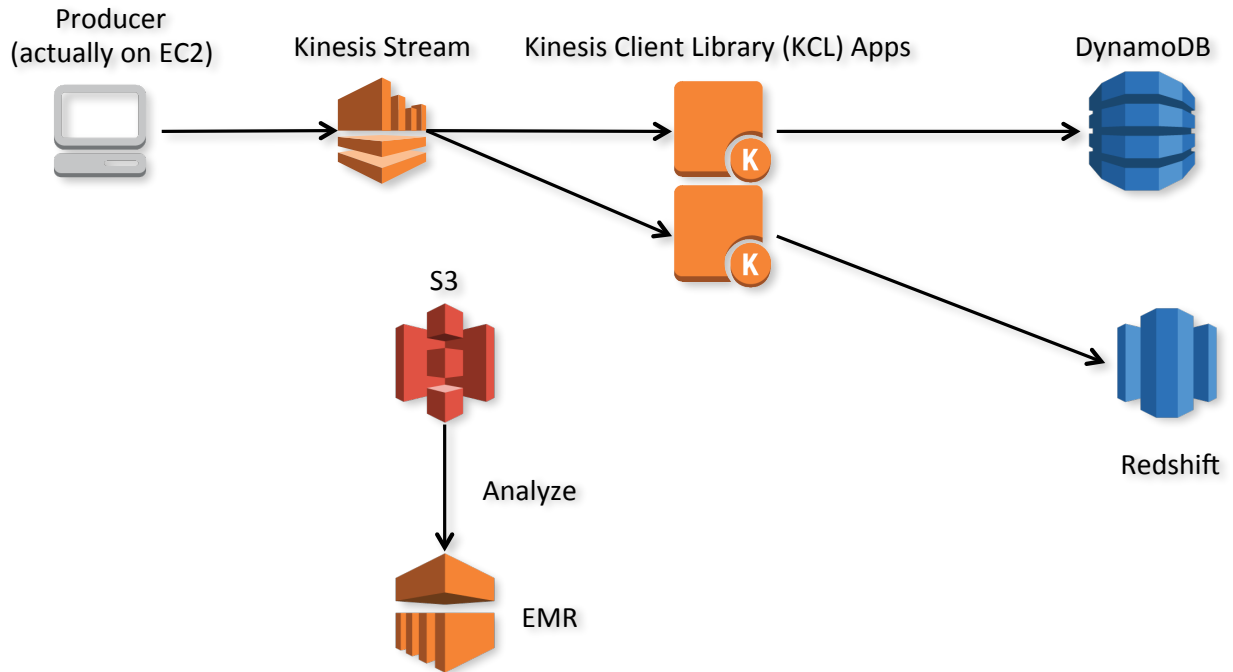
Amazon Redshift



Amazon Elastic Map Reduce



Big Data Workshop



Big Data Workshop

