Getting Started with AWS

Analyzing Big Data

Getting Started with AWS: Analyzing Big Data

Table of Contents

Overview

The term *big data* has become so common, it defies clear definition. A prevailing theme in any context is that big data is difficult data: difficult to store in traditional databases, difficult to process on standard servers, and difficult to analyze with typical applications. Even "smaller" data can exhibit a complexity that requires a new approach. As you explore more sources and types of data, you'll also need to identify tools and techniques for managing it efficiently and extracting real value.

This guide illustrates two uses of Amazon Web Services to process big data. *Getting Started: Sentiment Analysis* (p. 5) shows you how to use Hadoop to evaluate Twitter data. *Getting Started: Web Server Log Analysis* (p. 17) shows you how to query Apache web server logs with Hive.

Key AWS Services for Big Data

With Amazon Web Services, you pay only for the resources you use. Instead of maintaining a cluster of physical servers and storage devices that are standing by for possible use, you can create resources when you need them. AWS also supports popular tools like Hadoop, and makes it easy to provision, configure, and monitor clusters for running those tools.

The following table shows how Amazon Web Services can help you manage big data.

Challenges	Amazon Web Services	Benefits
Data sets can be very large. Storage can become expensive, and data corruption and loss can have far-reaching implications.	Amazon Simple Storage Service (Amazon S3)	Amazon S3 can store large amounts of data, and its capacity can grow to meet your needs. It is highly redundant and secure, protecting against data loss and unauthorized use. Amazon S3 also has an intentionally small feature set to keep its costs low.

Challenges	Amazon Web Services	Benefits
Maintaining a cluster of physical servers to process data is expensive and time-consuming.	Amazon Elastic Compute Cloud (Amazon EC2)	When you run an application on a virtual Amazon EC2 server, you pay for the server only while the application is running, and you can increase the number of servers — within minutes, not hours or days — to meet the processing needs of your application.
Hadoop and other open-source big-data tools can be challenging to configure, monitor, and operate.	Amazon EMR	Amazon EMR handles cluster configuration, monitoring, and management. Amazon EMR also integrates open-source tools with other AWS services to simplify large-scale data processing in the cloud, so you can focus on data analysis and extracting value.

Let's look at two examples of how AWS can help you work with big data.

Getting Set Up

Before you use AWS for the first time, complete the steps in this section:

- Step 1: Sign Up for Amazon Web Services (AWS) (p. 3)
- Step 2: Create a Key Pair (p. 3)

You'll need to complete these steps only once. Your account and key pair will work in both of the tutorials in this guide: *Getting Started: Sentiment Analysis* (p. ?) and *Getting Started: Web Server Log Analysis* (p. ?).

Step 1: Sign Up for Amazon Web Services (AWS)

When you create an AWS account, your account will automatically be signed up for all AWS services. You pay only for the services that you use.

If you have an AWS account already, skip to the next step. If you don't have an AWS account, use the following procedure to create one.

To sign up for an AWS account

1. Go to http://aws.amazon.com, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Step 2: Create a Key Pair

You'll need to create a key pair to connect to Amazon EC2 instances. For security reasons, EC2 instances use a public/private key pair, rather than a user name and password, to authenticate connection requests. The public key half of this pair is embedded in the instance, so you can use the private key to log in securely without a password.

In this step, you'll use the AWS Management Console to create a key pair. Later, you'll use this key pair to connect to the Amazon EC2 instances that are used in the tutorials.

To generate a key pair

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the top navigation bar, in the region selector, click **US East (N. Virginia)**.
3. In the left navigation pane, under **Network and Security**, click **Key Pairs**.
4. Click **Create Key Pair**.
5. Type **`mykeypair`** in the new **Key Pair Name** box and then click **Create**.
6. Download the private key file, which is named `mykeypair.pem`, and keep it in a safe place. You will need it to access any instances that you launch with this key pair.

   Important
   If you lose the key pair, you cannot connect to your Amazon EC2 instances.

   For more information about key pairs, see Getting an SSH Key Pair in the *Amazon Elastic Compute Cloud User Guide*.

Getting Started: Sentiment Analysis

Topics

Sentiment analysis refers to various methods of examining and processing data in order to identify a subjective response, usually a general mood or a group's opinions about a specific topic. For example, sentiment analysis can be used to gauge the overall positivity of a blog or a document, or to capture constituent attitudes toward a political candidate.

Sentiment data is often derived from social media services and similar user-generated content, such as reviews, comments, and discussion groups. The data sets thus tend to grow large enough to be considered "big data."

Suppose your company recently released a new product and you want to assess its reception among consumers. You know that social media can help you capture a broad sample of public opinion, but you don't have time to monitor every mention. You need a better way to determine aggregate sentiment.

Amazon EMR integrates open-source data processing frameworks with the full suite of Amazon Web Services. The resulting architecture is scalable, efficient, and ideal for analyzing large-scale sentiment data, such as tweets over a given time period.

In this tutorial, you'll launch an AWS CloudFormation stack that provides a script for collecting tweets. You'll store the tweets in Amazon S3 and customize a mapper file for use with Amazon EMR. Then you'll create an Amazon EMR cluster that uses a Python natural language toolkit, implemented with a Hadoop streaming job, to classify the data. Finally, you'll examine the output files and evaluate the aggregate sentiment of the tweets.

This tutorial typically takes less than an hour to complete. You pay only for the resources you use. The tutorial includes a cleanup step to help ensure that you don't incur additional costs. You may also want to review the Pricing (p. 28) topic.

Important
Before you begin, make sure you've completed the steps in Getting Set Up (p. 3).

Click **Next** to start the tutorial.

Step 1: Create a Twitter Developer Account

In order to collect tweets for analysis, you'll need to create an account on the Twitter developer site and generate credentials for use with the Twitter API.

To create a Twitter developer account

1. Go to https://dev.twitter.com/user/login and log in with your Twitter user name and password. If you do not yet have a Twitter account, click the **Sign up** link that appears under the **Username** field.
2. If you've already used the Twitter developer site to generate credentials and register applications, skip to the next step.

   If you have not yet used the Twitter developer site, you'll be prompted to authorize the site to use your account. Click **Authorize app** to continue.
3. Go to the Twitter applications page at https://dev.twitter.com/apps and click **Create a new application**.
4. Follow the on-screen instructions. For the application **Name**, **Description**, and **Website**, you can enter any text — you're simply generating credentials to use with this tutorial, rather than creating a real application.
5. On the details page for your new application, you'll see a **Consumer key** and **Consumer secret**. Make a note of these values; you'll need them later in this tutorial. You may want to store your credentials in a text file.
6. At the bottom of the application details page, click **Create my access token**. Make a note of the **Access token** and **Access token secret** values that appear, or add them to the text file you created in the preceding step.

If you need to retrieve your Twitter developer credentials at any point, you can go to https://dev.twitter.com/apps and select the application you created for the purposes of this tutorial.

Step 2: Create an Amazon S3 Bucket for the Amazon EMR Files

Amazon EMR jobs typically use Amazon S3 buckets for input and output data files, as well as for any mapper and reducer files that aren't provided by open-source tools. For the purposes of this tutorial, you'll create your own Amazon S3 bucket in which you'll store the data files and a custom mapper.

To create an Amazon S3 bucket using the console

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/home.
2. Click **Create Bucket**.
3. Enter a name for your bucket, such as **`mysentimentjob`**.

Note
To meet Hadoop requirements, Amazon S3 bucket names used with Amazon EMR are restricted to lowercase letters, numbers, periods (.), and hyphens (-).

4. Leave the **Region** set to **US Standard** and click **Create**.
5. Click the name of your new bucket in the **All Buckets** list.
6. Click **Create Folder**, then type `input`. Press **Enter** or click the check mark.
7. Repeat this step to create another folder called `mapper` at the same level as the input folder.
8. For the purposes of this tutorial (to ensure that all services can use the folders), you should make the folders public. Select the check boxes next to your folders. Click **Actions**, then click **Make Public**. Click **OK** to confirm that you want to make the folders public.

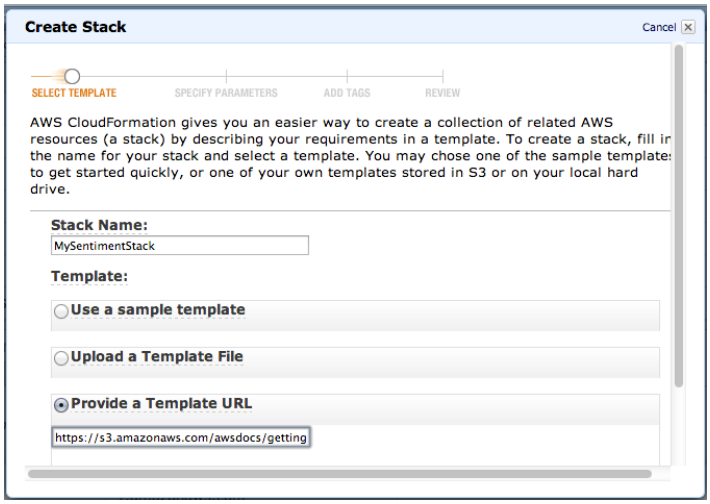
Make a note of your bucket and folder names — you'll need them in later steps.

Step 3: Collect and Store the Sentiment Data

In this step, you'll use an AWS CloudFormation template to launch an instance, then use the tools on the instance to collect data via the Twitter API. You'll also use a command-line tool to store the collected data in the Amazon S3 bucket you created.

To launch the AWS CloudFormation stack

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. Make sure **US East (N. Virginia)** is selected in the region selector of the navigation bar.
3. Click **Create Stack**.
4. In the **Stack Name** box, type any name that will help you identify your stack, such as `MySentimentStack`.
5. Under **Template**, select **Provide a Template URL**. Type `https://s3.amazonaws.com/awsdocs/gettingstarted/latest/sentiment/sentimentGSG.template` in the box (or copy the URL from this page and paste it in the box). Click **Continue**.



6. On the **Specify Parameters** page, enter your AWS and Twitter credentials. The **Key Pair** name must match the key pair you created in the US-East region in Step 2: Create a Key Pair (p. 3).

   For best results, copy and paste the Twitter credentials from the Twitter developer site or the text file you saved them in.

Note
The order of the Twitter credential boxes on the **Specify Parameters** page may not match the display order on the Twitter developer site. Make sure you're pasting the correct value in each box.

7. Select the check box to acknowledge that the template may create IAM resources, then click **Continue**. Click **Continue** again on the **Add Tags** page.
8. Review your settings, making sure your Twitter credentials are correct. You can make changes to the settings by clicking the **Edit** link for a specific step in the process.
9. Click **Continue** to launch the stack. A confirmation window opens. Click **Close**.
10. The confirmation window closes, returning you to the AWS CloudFormation console. Your new AWS CloudFormation stack appears in the list with its status set to **CREATE_IN_PROGRESS**.

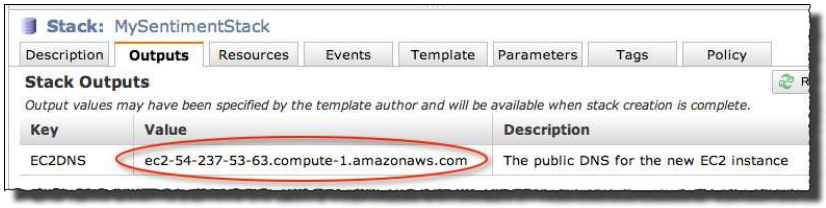
Note
Your stack will take several minutes to launch. Make sure to click **Refresh** on the **Stacks** page to see whether the stack has been successfully created.

For more information about AWS CloudFormation, go to Walkthrough: Updating a Stack.

To collect tweets using your AWS CloudFormation stack

When your stack shows the status CREATE_COMPLETE, it's ready to use.

1. Click the **Outputs** tab in the bottom pane to get the IP address of the Amazon EC2 instance that AWS CloudFormation created.



2. Connect to the instance via SSH, using the user name `ec2-user`. For more information about connecting to an instance and configuring your SSH credentials and tools, see Connecting to Your Linux/UNIX Instances Using SSH or Connecting to Linux/UNIX Instances from Windows Using PuTTY. (Disregard the sections that describe how to transfer files.)
3. In the SSH window, type the following command:

```
cd sentiment
```

4. The instance has been preconfigured with Tweepy, an open-source package for use with the Twitter API. Python scripts for running Tweepy appear in the `sentiment` directory. To ensure that they are present, type the following command:

```
ls
```

You should see files named `collector.py` and `twaiter.py`, as well as `twitterparams.py`.

5. To collect tweets, type the following command, where *`term1`* is your search term.

```
python collector.py term1
```

To use a multi-word term, enclose it in quotation marks. Examples:

```
python collector.py kindle
python collector.py "kindle fire"
```

The collector script is not case sensitive.

6. Press **Enter** to run the collector script. Your SSH window should show the message "Collecting tweets. Please wait."

   The script collects 500 tweets, which may take several minutes. If you're searching for a subject that is not currently popular on Twitter (or if you edited the script to collect more than 500 tweets), the script will take longer to run. You can interrupt it at any time by pressing **Control+C**.

   When the script has finished running, your SSH window will show the message "Finished collecting tweets."

   Note
   If your SSH connection is interrupted while the script is still running, reconnect to the instance and run the script with nohup (e.g., `nohup python collector.py > /dev/null &`).

To store the collected tweets in Amazon S3

Your sentiment analysis stack has been preconfigured with s3cmd, a command-line tool for Amazon S3. You'll use s3cmd to store your tweets in the bucket you created earlier.

1. In your SSH window, type the following command. (The current directory should still be `sentiment`. If it's not, use `cd` to navigate to the `sentiment` directory.)

   ```
   ls
   ```

   You should see a file named `tweets.`*`date-time`*`.txt`, where *`date`* and *`time`* reflect when the script was run. This file contains the ID numbers and full text of the tweets that matched your search terms.

2. To copy the Twitter data to Amazon S3, type the following command, where *`tweet-file`* is the file you identified in the previous step and *`your-bucket`* is the name of the Amazon S3 bucket you created earlier.

   ```
   s3cmd put tweet-file s3://your-bucket/input/
   ```

   Example:

   ```
   s3cmd put tweets.Nov12-1227.txt s3://mysentimentjob/input/
   ```

   Important
   Be sure to include the trailing slash, to indicate that input is a folder. Otherwise, Amazon S3 will create an object called `input` in your base S3 bucket.

3. To verify that the file was uploaded to Amazon S3, type the following command:

   ```
   s3cmd ls s3://your-bucket/input/
   ```

   You can also use the Amazon S3 console at https://console.aws.amazon.com/s3/ to view the contents of your bucket and folders.

Step 4: Customize the Amazon EMR Mapper

When you create your own Hadoop streaming programs, you'll need to write mapper and reducer executables as described in Process Data with a Streaming Cluster in the *Amazon Elastic MapReduce Developer Guide*. For this tutorial, we've provided a mapper script that you can customize for use with your Twitter search term.

To customize the mapper

1. On your local system, open a text editor of your choice. Copy and paste the following script into a new file.

```
#!/usr/bin/python

import cPickle as pickle
import nltk.classify.util
from nltk.classify import NaiveBayesClassifier
from nltk.tokenize import word_tokenize
import sys

sys.stderr.write("Started mapper.\n");


def word_feats(words):
    return dict([(word, True) for word in words])


def subj(subjLine):
    subjgen = subjLine.lower()
    # Replace term1 with your subject term
    subj1 = "term1"
    if subjgen.find(subj1) != -1:
        subject = subj1
        return subject
    else:
        subject = "No match"
        return subject


def main(argv):
    classifier = pickle.load(open("classifier.p", "rb"))
    for line in sys.stdin:
        tolk_posset = word_tokenize(line.rstrip())
        d = word_feats(tolk_posset)
        subjectFull = subj(line)
        if subjectFull == "No match":
            print "LongValueSum:" + " " + subjectFull + ": " + "\t" + "1"
        else:
            print "LongValueSum:" + " " + subjectFull + ": " + classifi
er.classify(d) + "\t" + "1"


if __name__ == "__main__":
    main(sys.argv)
```

2. Edit the following line in the script:

```
subj1 = "term1"
```

   Replace *term1* with the search term you used in Step 3: Collect and Store the Sentiment Data (p. 7). Example:

```
subj1 = "kindle"
```

   Important
   Make sure you don't change any of the spacing in the file. Incorrect indentation will cause the Hadoop streaming program to fail.

   Save the edited file. You may also want to review the script generally, to get a sense of how mappers can work.

   Note
   In your own mappers, you'll probably want to fully automate the configuration. The manual editing in this tutorial is for purposes of illustration only. For more details about creating Amazon EMR work steps and bootstrap actions, go to Create Bootstrap Actions to Install Additional Software and Steps in the Amazon Elastic MapReduce Developer Guide.

3. Go to the Amazon S3 console at https://console.aws.amazon.com/s3/ and locate the `mapper` folder you created in Step 2: Create an Amazon S3 Bucket for the Amazon EMR Files (p. 6).
4. Click **Upload** and follow the on-screen instructions to upload your customized mapper file.
5. Make the mapper file public: select it, then select **Actions** and then **Make Public**.

Step 5: Create an Amazon EMR Cluster

Important
This tutorial reflects changes made to the Amazon EMR console in November 2013. If your console screens do not match the images in this guide, switch to the new version by clicking the link that appears at the top of the console:

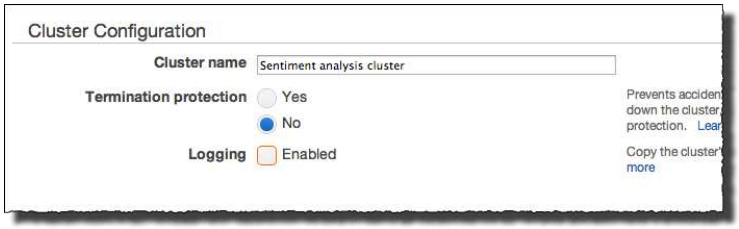
Amazon EMR allows you to configure a cluster with software, bootstrap actions, and work steps. For this tutorial, you'll run a Hadoop streaming program. When you configure a cluster with a Hadoop streaming program in Amazon EMR, you specify a mapper and a reducer, as well as any supporting files. The following list provides a summary of the files you'll use for this tutorial.

- For the mapper, you'll use the file you customized in the preceding step.
- For the reducer method, you'll use the predefined Hadoop package `aggregate`. For more information about the aggregate package, go to the Hadoop documentation.
- Sentiment analysis usually involves some form of natural language processing. For this tutorial, you'll use the Natural Language Toolkit (NLTK), a popular Python platform. You'll use an Amazon EMR bootstrap action to install the NLTK Python module. Bootstrap actions load custom software onto the instances that Amazon EMR provisions and configures. For more information, go to Create Bootstrap Actions in the *Amazon Elastic MapReduce Developer Guide*.
- Along with the NLTK module, you'll use a natural language classifier file that we've provided in an Amazon S3 bucket.
- For the job's input data and output files, you'll use the Amazon S3 bucket you created (which now contains the tweets you collected).

Note that the files used in this tutorial are for illustration purposes only. When you perform your own sentiment analysis, you'll need to write your own mapper and build a sentiment model that meets your needs. For more information about building a sentiment model, go to Learning to Classify Text in *Natural Language Processing with Python*, which is provided for free on the NLTK site.

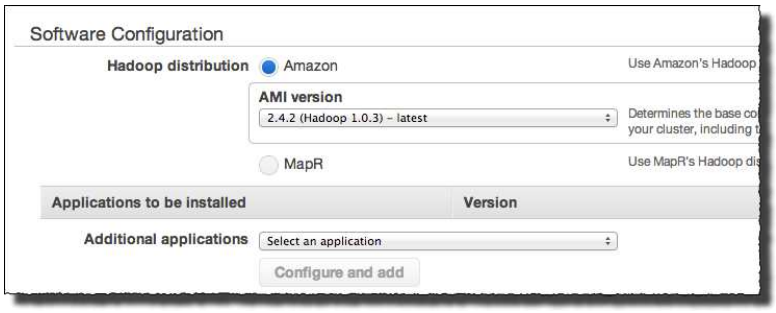
To create an Amazon EMR cluster using the console

1. Open the Amazon EMR console at https://console.aws.amazon.com/elasticmapreduce/.
2. Click **Create cluster**.
3. In the **Cluster Configuration** section, type a **Cluster name** or use the default value of **`My cluster`**. Set **Termination protection** to **No** and clear the **Logging enabled** check box.

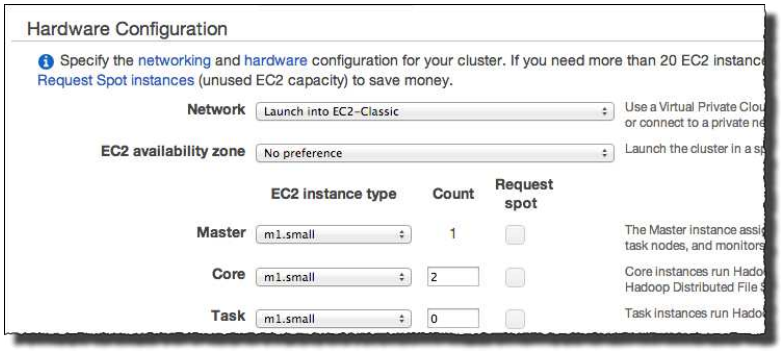


Note
In a production environment, logging and debugging can be usefull tools for analyzing errors or inefficiencies in Amazon EMR steps and applications. For more information on how to use logging and debugging in Amazon EMR, go to Troubleshooting in the *Amazon Elastic MapReduce Developer Guide*.

4. In the **Software Configuration** section, leave the default **Hadoop distribution** setting: **Amazon** and latest **AMI version**. Under **Applications to be installed**, click each X to remove Hive and Pig from the list.



5. In the **Hardware Configuration** section, leave the default settings. The default instance types, an m1.small master node and two m1.small core nodes, will help keep the cost of this tutorial low.



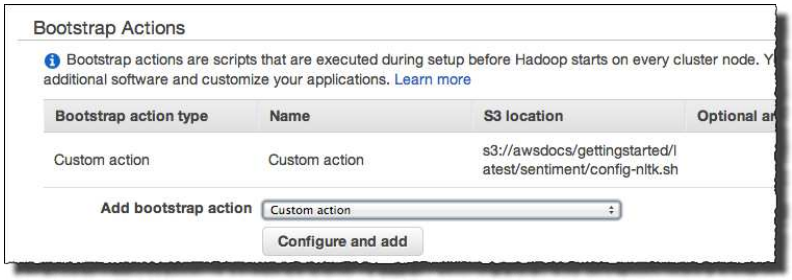
Note
When you analyze data in a real application, you may want to increase the size or number of these nodes to improve processing power and optimize computational time. You may also want to use spot instances to further reduce your Amazon EC2 costs. For more

information about spot instances, go to Lowering Costs with Spot Instances in the *Amazon Elastic MapReduce Developer Guide*.

6. In the **Security and Access** section, select the **EC2 key pair** you created earlier. Leave the default IAM settings.

7. In the **Bootstrap Actions** section, in the **Add bootstrap action** list, select **Custom action**. You'll add a custom action that installs and configures the Natural Language Toolkit on the cluster.
8. In the **Add Bootstrap Action** popup, enter a **Name** for the action or leave it set to **Custom action**. In the **Amazon S3 Location** box, type **`s3://awsdocs/gettingstarted/latest/sentiment/config-nltk.sh`** (or copy and paste the URL from this page), and then click **Add**. (You can also download and review the shell script, if you'd like.)

The **Bootstrap Actions** section should now show the custom action you added.



9. In the **Steps** section, you'll define the Hadoop streaming job. In the **Add step** list, select **Streaming program**, then click **Configure and add**.
10. In the **Add Step** popup, configure the job as follows, replacing *`your-bucket`* with the name of the Amazon S3 bucket you created earlier:

Name	Sentiment analysis
Mapper	s3://*`your-bucket`*/mapper/sentiment.py
Reducer	aggregate
Input S3 location	s3://*`your-bucket`*/input
Output S3 location	s3://*`your-bucket`*/output *(make sure this folder does not yet exist)*

Arguments	-cacheFile s3://awsdocs/gettingstarted/latest/sentiment/classifier.p#classifier.p
Action on failure	Continue

Click **Add**. The **Steps** section should now show the parameters for the streaming program.

11. Below the step parameters, set **Auto-terminate** to **Yes**.
12. Review the cluster settings. If everything looks correct, click **Create cluster**.

A summary of your new cluster will appear, with the status **Starting**. It will take a few minutes for Amazon EMR to provision the Amazon EC2 instances for your cluster.

Step 6: Examine the Sentiment Analysis Output

When your cluster's status in the Amazon EMR console is **Waiting: Waiting after step completed**, you can examine the results.

To examine the streaming program results

1. Go to the Amazon S3 console at https://console.aws.amazon.com/s3/home and locate the bucket you created in Step 2: Create an Amazon S3 Bucket for the Amazon EMR Files (p. 6). You should see a new `output` folder in your bucket. You may need to click the refresh arrow in the top right corner to see the new bucket.
2. The job output will be split into several files: an empty status file named `_SUCCESS` and several `part-xxxxx` files. The `part-xxxxx` files contain sentiment measurements generated by the Hadoop streaming program.
3. To download an output file, select it in the list, then click **Actions** and select **Download**. Right-click the link in the pop-up window to download the file.

   Repeat this step for each output file.
4. Open the files in a text editor. You'll see the total number of positive and negative tweets for your search term, as well as the total number of tweets that did not match any of the positive or negative terms in the classifier (usually because the subject term was in a different field, rather than in the actual text of the tweet).

   Example:

```
kindle: negative      13
kindle: positive     479
No match:              8
```

In this example, the sentiment is overwhelmingly positive. In most cases, the positive and negative totals will be closer together. For your own sentiment analysis work, you'll want to collect and compare data over several time periods, possibly using several different search terms, to get as accurate a measurement as possible.

Step 7: Clean Up

To prevent your account from accruing additional charges, you should terminate the resources you used in this tutorial.

To delete the AWS CloudFormation stack

1. Go to the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. In the **AWS CloudFormation Stacks** section, select your sentiment stack.
3. Either click the **Delete Stack** button, or right-click your selected stack and click **Delete Stack**.
4. Click **Yes, Delete** in the confirmation dialog that appears.

   Note
   After stack deletion has begun, you can't cancel the process. The stack will proceed to the state **DELETE_IN_PROGRESS**. After the stack has been deleted, it will have the state **DELETE_COMPLETE**.

Because you ran a Hadoop streaming program and set it to auto-terminate after running the steps in the program, the cluster should have been automatically terminated when processing was complete.

To ensure the Amazon EMR cluster was terminated

1. If you are not already viewing the cluster list, click **Cluster List** in the **Elastic MapReduce** menu at the top of the Amazon Elastic MapReduce console.
2. In the cluster list, make sure the **Status** of your cluster is **Terminated**.

To terminate an Amazon EMR cluster

1. If you are not already viewing the cluster list, click **Cluster List** in the **Elastic MapReduce** menu at the top of the Amazon Elastic MapReduce console.
2. In the cluster list, select the box to the left of the cluster name, and then click **Terminate**. In the confirmation pop-up that appears, click **Terminate**.

The next step is optional. It deletes the key pair you created earlier. You are not charged for key pairs. If you are planning to explore Amazon EMR further or complete the other tutorial in this guide, you should retain the key pair.

To delete a key pair

1. In the Amazon EC2 console navigation pane, select **Key Pairs**.
2. In the content pane, select the key pair you created, then click **Delete**.

The next step is optional. It deletes two security groups created for you by Amazon EMR when you launched the cluster. You are not charged for security groups. If you are planning to explore Amazon EMR further, you should retain them.

To delete Amazon EMR security groups

1. In the Amazon EC2 console navigation pane, click **Security Groups**.
2. In the content pane, click the **ElasticMapReduce-slave** security group.
3. In the details pane for the ElasticMapReduce-slave security group, click the **Inbound** tab. Delete all actions that reference ElasticMapReduce. Click **Apply Rule Changes**.
4. In the content pane, click **ElasticMapReduce-slave**, and then click **Delete**. Click **Yes, Delete** to confirm. (This group must be deleted before you can delete the ElasticMapReduce-master group.)
5. In the content pane, click **ElasticMapReduce-master**, and then click **Delete**. Click **Yes, Delete** to confirm.

You've completed the sentiment analysis tutorial. Be sure to review the other topics in this guide for more information about Amazon Elastic MapReduce.

Getting Started: Web Server Log Analysis

Topics

Suppose you host a popular e-commerce website. In order to understand your customers better, you want to analyze your Apache web logs to discover how people are finding your site. You'd especially like to determine which of your online ad campaigns are most successful in driving traffic to your online store.

The web server logs, however, are too large to import into a MySQL database, and they are not in a relational format. You need another way to analyze them.

Amazon EMR integrates open-source applications such as Hadoop and Hive with Amazon Web Services to provide a scalable and efficient architecture for analyzing large-scale data, such as Apache web logs.

In the following tutorial, we'll import data from Amazon S3 and create an Amazon EMR cluster from the AWS Management Console. Then we'll connect to the master node of the cluster, where we'll run Hive to query the Apache logs using a simplified SQL syntax.

This tutorial typically takes less than an hour to complete. You pay only for the resources you use. The tutorial includes a cleanup step to help ensure that you don't incur additional costs. You may also want to review the Pricing (p. 28) topic.

> **Important**
> Before you begin, make sure you've completed the steps in Getting Set Up (p. 3).

Click **Next** to start the tutorial.

Step 1: Create a Cluster Using the Console

Important
This tutorial reflects changes made to the Amazon EMR console in November 2013. If your console screens do not match the images in this guide, switch to the new version by clicking the link that appears at the top of the console:

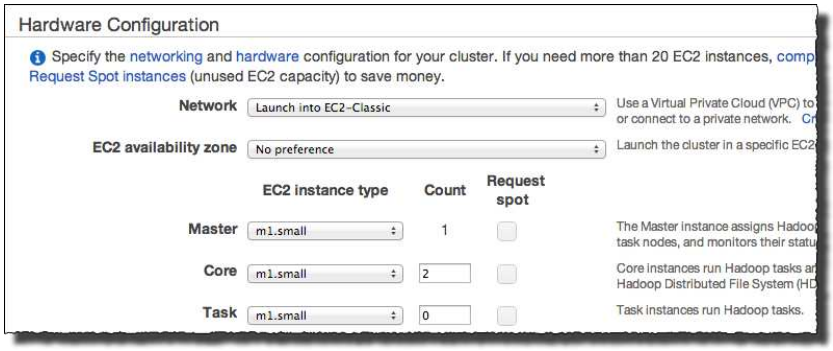
To create a cluster using the console

1. Sign in to the AWS Management Console and open the Amazon Elastic MapReduce console at https://console.aws.amazon.com/elasticmapreduce/.
2. Click **Create Cluster**.
3. In the **Cluster Configuration** section, type a **Cluster name** or use the default value of `My cluster`. Set **Termination protection** to **No** and clear the **Logging** check box.

Note
In a production environment, logging and debugging can be useful tools for analyzing errors or inefficiencies in Amazon EMR steps or programs. For more information on how to use logging and debugging in Amazon EMR, go to Troubleshooting in the *Amazon Elastic MapReduce Developer Guide*.

4. In the **Software Configuration** section, leave the default **Hadoop distribution** setting: **Amazon** and latest **AMI version**. Under **Applications to be installed**, leave the default **Hive** settings. Click the X to remove Pig from the list.

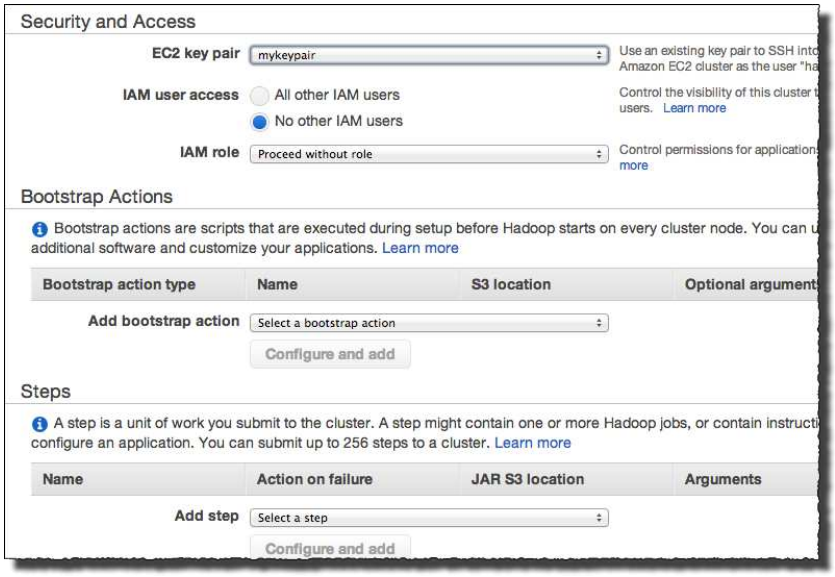
5. In the **Hardware Configuration** section, leave the default settings. The default instance types, an m1.small master node and two m1.small core nodes, will help keep the cost of this tutorial low.



Note
When you analyze data in a real application, you may want to increase the size or number of these nodes to improve processing power and reduce computational time. You may also want to use spot instances to further reduced your Amazon EC2 costs. For more information about spot instances, go to Lowering Costs with Spot Instances in the *Amazon Elastic MapReduce Developer Guide*.

6. In the **Security and Access** section, select the **EC2 key pair** you created in the preceding step. Leave the default IAM settings.

   Leave the default **Bootstrap Actions** and **Steps** settings. Bootstrap actions and steps allow you to customize and configure your application. For this tutorial, we will be using Hive, which is already installed on the AMI, so no addition configuration is needed.



7. Review the settings. If everything looks correct, click **Create cluster**.

   A summary of your new cluster will appear, with the status STARTING. It will take a few minutes for Amazon EMR to provision the Amazon EC2 instances for your cluster.

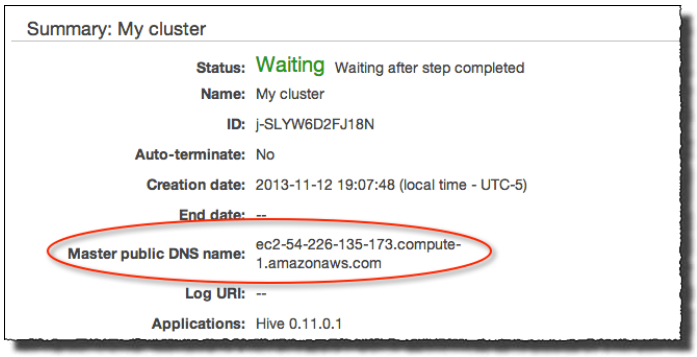
Step 2: Connect to the Master Node

When the cluster in the Amazon EMR console is WAITING, the master node is ready for you to connect to it. First you'll need to get the DNS name of the master node and configure your connection tools and credentials.

To locate the DNS name of the master node

- If you're not currently viewing the Cluster Details page, first select the cluster on the Cluster List page.

  On the Cluster Details page, you'll see the **Master public DNS name** Make a note of the DNS name; you'll need it in the next step.



You can use secure shell (SSH) to open a terminal connection to the master node. An SSH application is installed by default on most Linux, Unix, and Mac OS installations. Windows users can use an application called PuTTY to connect to the master node. Platform-specific instructions for configuring a Windows application to open an SSH connection are provided later in this topic.

You must first configure your credentials, or SSH will return an error message saying that your private key file is unprotected, and it will reject the key. You need to do this step only the first time you use the private key to connect.

To configure your credentials on Linux/Unix/Mac OS X

1. Open a terminal window. On most computers running Mac OS X, you'll find the terminal at Applications/Utilities/Terminal. On many Linux distributions, the path is Applications/Accessories/Terminal.
2. Set the permissions on the PEM file for your Amazon EC2 key pair so that only the key owner has permissions to access the key. For example, if you saved the file as `mykeypair.pem` in your home directory, you can use this command:

```
chmod og-rwx ~/mykeypair.pem
```

To connect to the master node using Linux/Unix/Mac OS X

1. In the terminal window, enter the following command, where the value of the `-i` parameter indicates the location of the private key file you saved in of Step 2: Create a Key Pair (p. 3). In this example, the key is assumed to be in your home directory.

```
ssh hadoop@master-public-dns-name \
-i ~/mykeypair.pem
```

2. You'll see a warning that the authenticity of the host can't be verified. Type **`yes`** to continue connecting.

If you're using a Windows-based computer, you'll need to install an SSH client in order to connect to the master node. In this tutorial, we'll use PuTTY. If you have already installed PuTTY and configured your key pair, you can skip this procedure.

To install and configure PuTTY on Windows

1. Download PuTTYgen.exe and PuTTY.exe to your computer from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.
2. Launch PuTTYgen.
3. Click Load. Select the PEM file you created earlier. You may have to change the search parameters from file of type “PuTTY Private Key Files (*.ppk) to “All Files (*.*)”.
4. Click **Open**.
5. On the PuTTYgen Notice telling you the key was successfully imported, click **OK**.
6. To save the key in the PPK format, click **Save private key**.
7. When PuTTYgen prompts you to save the key without a pass phrase, click **Yes**.
8. Enter a name for your PuTTY private key, such as mykeypair.ppk.

To connect to the master node using Windows/Putty

1. Start PuTTY.
2. In the **Category** list, click **Session**. In the **Host Name** box, type hadoop@*DNS*. The input will look similar to hadoop@ec2-184-72-128-177.compute-1.amazonaws.com.
3. In the **Category** list, expand **Connection**, expand **SSH**, and then click **Auth**.
4. In the **Options controlling SSH authentication** pane, click **Browse for Private key file for authentication**, and then select the private key file that you generated earlier. If you are following this guide, the file name is `mykeypair.ppk`.
5. Click **Open**.
6. To connect to the master node, click **Open**.
7. In the PuTTY Security Alert window, click **Yes**.

> **Note**
> For more information about how to install PuTTY and use it to connect to an EC2 instance, go to Connecting to Linux/UNIX Instances from Windows Using PuTTY in the *Amazon Elastic Compute Cloud User Guide*.

When you've successfully connected to the master node via SSH, you'll see a welcome message and prompt similar to the following:

```
--------------------------------------------------------------------------------

Welcome to Amazon EMR running Hadoop and Debian/Lenny.

Hadoop is installed in /home/hadoop. Log files are in /mnt/var/log/hadoop. Check
/mnt/var/log/hadoop/steps for diagnosing step failures.

The Hadoop UI can be accessed via the following commands:

  JobTracker    lynx http://localhost:9100/
  NameNode      lynx http://localhost:9101/

--------------------------------------------------------------------------------
hadoop@ip-10-245-190-34:~$
```

Step 3: Start and Configure Hive

Apache Hive is a data warehouse application you can use to query Amazon EMR cluster data with a SQL-like language. Because Hive was listed in the **Applications to be installed** when we created the cluster, it's ready to use on the master node.

To use Hive interactively to query the web server log data, you'll need to load some additional libraries. The additional libraries are contained in a Java archive file named `hive_contrib.jar` on the master node. When you load these libraries, Hive bundles them with the map-reduce job that it launches to process your queries.

To learn more about Hive, go to http://hive.apache.org/.

To start and configure Hive on the master node

1. On the command line of the master node, type `hive`, and then press Enter.
2. At the `hive>` prompt, type the following command, and then press Enter.

```
hive> add jar /home/hadoop/hive/lib/hive_contrib.jar;
```

Wait for a confirmation message similar to the following:

```
Added /home/hadoop/hive/lib/hive_contrib.jar to class path
Added resource: /home/hadoop/hive/lib/hive_contrib.jar
```

Step 4: Create the Hive Table and Load Data into HDFS

In order for Hive to interact with data, it must translate the data from its current format (in the case of Apache web logs, a text file) into a format that can be represented as a database table. Hive does this translation using a serializer/deserializer (SerDe). SerDes exist for a variety of data formats. For information about how to write a custom SerDe, go to the Apache Hive Developer Guide.

The SerDe we'll use in this example uses regular expressions to parse the log file data. It comes from the Hive open-source community and can be found at https://github.com/apache/hive/blob/trunk/contrib/src/java/org/apache/hadoop/hive/contrib/serde2/RegexSerDe.java. (This link is provided for reference only; for the purposes of this tutorial, you do not need to download the SerDe.).

Using this SerDe, we can define the log files as a table, which we'll query using SQL-like statements later in this tutorial.

To translate the Apache log file data into a Hive table

- Copy the following multiline command. At the `hive` command prompt, paste the command, and then press Enter.

```
CREATE TABLE serde_regex(
  host STRING,
  identity STRING,
  user STRING,
  time STRING,
  request STRING,
  status STRING,
  size STRING,
  referer STRING,
  agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  "input.regex" = "([^ ]*) ([^ ]*) ([^ ]*) (-|\\[[^\\]]*\\]) ([^
\"]*|\"[^\"]*\") (-|[0-9]*) (-|[0-9]*)(?: ([^ \"]*|\"[^\"]*\") ([^
\"]*|\"[^\"]*\"))?",
  "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s"
)
LOCATION 's3://elasticmapreduce/samples/pig-apache/input/';
```

In the command, the LOCATION parameter specifies the location of a set of sample Apache log files in Amazon S3. To analyze your own Apache web server log files, you would replace the Amazon S3 URL above with the location of your own log files in Amazon S3. To meet the requirements of Hadoop, Amazon S3 bucket names used with Amazon EMR must contain only lowercase letters, numbers, periods (.), and hyphens (-).

After you run the command above, you should receive a confirmation like this one:

```
Found class for org.apache.hadoop.hive.contrib.serde2.RegexSerDe
OK
Time taken: 12.56 seconds
hive>
```

Once Hive has loaded the data, the data will persist in HDFS storage as long as the Amazon EMR cluster is running, even if you shut down your Hive session and close the SSH connection.

Step 5: Query Hive

You're ready to start querying the Apache log file data. Here are some sample queries to run.

Count the number of rows in the Apache webserver log files.

```
select count(1) from serde_regex;
```

Return all fields from one row of log file data.

```
select * from serde_regex limit 1;
```

Count the number of requests from the host with an IP address of 192.168.1.198.

```
select count(1) from serde_regex where host="192.168.1.198";
```

To return query results, Hive translates your query into a Hadoop MapReduce job and runs it on the Amazon EMR cluster. Status messages will appear as the Hadoop job runs.

Hive SQL is a subset of SQL; if you know SQL, you'll be able to easily create Hive queries. For more information about the query syntax, go to the Hive Language Manual.

Step 6: Clean Up

To prevent your account from accruing additional charges, you should terminate the cluster when you are done with this tutorial. Because you used the cluster interactively, it has to be manually terminated.

To disconnect from Hive and SSH

1. In your SSH window or client, press Click CTRL+C to exit Hive.
2. At the SSH command prompt, type `exit`, and then press Enter. You can then close the terminal or PuTTY window.

```
exit
```

To terminate an Amazon EMR cluster

1. If you are not already viewing the cluster list, click **Cluster List** at the top of the Amazon Elastic MapReduce console.
2. In the cluster list, select the box to the left of the cluster name, and then click **Terminate**. In the confirmation pop-up that appears, click **Terminate**.

The next step is optional. It deletes the key pair you created earlier. You are not charged for key pairs. If you are planning to explore Amazon EMR further or complete the other tutorial in this guide, you should retain the key pair.

To delete a key pair

1. In the Amazon EC2 console navigation pane, select **Key Pairs**.
2. In the content pane, select the key pair you created, then click **Delete**.

The next step is optional. It deletes two security groups created for you by Amazon EMR when you launched the cluster. You are not charged for security groups. If you are planning to explore Amazon EMR further, you should retain them.

To delete Amazon EMR security groups

1. In the Amazon EC2 console navigation pane, click **Security Groups**.
2. In the content pane, click the **ElasticMapReduce-slave** security group.
3. In the details pane for the ElasticMapReduce-slave security group, click the **Inbound** tab. Delete all actions that reference ElasticMapReduce. Click **Apply Rule Changes**.
4. In the content pane, click **ElasticMapReduce-slave**, and then click **Delete**. Click **Yes, Delete** to confirm. (This group must be deleted before you can delete the ElasticMapReduce-master group.)
5. In the content pane, click **ElasticMapReduce-master**, and then click **Delete**. Click **Yes, Delete** to confirm.

Variations

The tutorials in this guide are just two examples of how you can use Amazon EMR to work with big data. This page summarizes some other options you may want to explore.

Variations for Hadoop

- **Analyze Sentiment by Location**

  In this tutorial, we analyzed only the content of the tweets. In your own work, you may want to collect geographic data, creating data sets for specific regions or ZIP codes in order to analyze sentiment by location.

  For more information about the geographic aspects of Twitter data, see the Twitter API resource documentation, such as the description of the reverse_geocode resource.
- **Explore Corpora and Data**

  The Natural Language Toolkit includes several corpora, ranging from Project Gutenberg selections to patient information leaflets. The Stanford Large Network Dataset Collection includes various types of sentiment data, as well as Amazon product review data. A wealth of movie review data is available from Cornell. You may find that working with a more focused set of data, rather than general Twitter data, yields more accurate results.
- **Try Other Classifier Models**

  This tutorial applied a Naive Bayesian classifier to the sentiment data. The Natural Language Toolkit supports several classification algorithms, including maxent (maximum entropy) and support vector machines (SVMs) via the scikitlearn module. Depending on the type of data you're analyzing and the features you choose to evaluate, other algorithms may produce better output. For more information, read Learning to Classify Text in the book *Natural Language Processing with Python*.

Variations for Hive

- **Script Your Hive Queries**

  Interactively querying data is the most direct way to get results, and interactive queries can help you explore data and refine your approach. Once you've created a set of queries that you want to run regularly, you can automate the process by saving your Hive commands as a script and uploading the

script to Amazon S3. For more information on how to launch a cluster using a Hive script, go to Launch a Hive Cluster in the *Amazon Elastic MapReduce Developer Guide*.

- **Use Pig Instead of Hive to Analyze Your Data**

  Amazon EMR provides access to many open-source tools, including Pig, which uses a language called Pig Latin to abstract map-reduce jobs. For an example of how to analyze log files with Pig, go to Parsing Logs with Apache Pig and Amazon Elastic MapReduce.

- **Create Custom Applications to Analyze Your Data**

  If you're not able to find an open-source tool that meets your needs, you can write a custom Hadoop map-reduce application and run it on Amazon EMR. For more information, go to Run a Hadoop Application to Process Data in the *Amazon Elastic MapReduce Developer Guide*.

  Alternatively, you can create a Hadoop streaming job that reads data from standard input. For an example, see Getting Started: Sentiment Analysis (p. 5) in this guide. For more details, go to Launch a Streaming Cluster in the *Amazon Elastic MapReduce Developer Guide*.

Pricing

The AWS Simple Monthly Calculator helps you estimate your monthly bill. It provides a per service cost breakdown, as well as an aggregate monthly estimate. You can also use the calculator to see an estimation and breakdown of costs for common solutions.

To estimate costs using the AWS Simple Monthly Calculator

1. Go to http://calculator.s3.amazonaws.com/calc5.html.
2. In the navigation pane, select a web service you currently use or plan to use. Enter your estimated monthly usage for that service. Click **Add To Bill** to add the cost to your total. Repeat this step for each web service you use.
3. To see the total estimated monthly charges, click the tab labeled **Estimate of Your Monthly Bill**.

For additional information, download the whitepaper How AWS Pricing Works. Pricing details are also available for each service. Example: Amazon Simple Storage Service Pricing

Related Resources

The following table lists related resources that you'll find useful as you work with AWS services.

Resource	Description
AWS Products and Services	A comprehensive list of products and services AWS offers.
Documentation	Official documentation for each AWS product including service introductions, service features, and API references, and other useful information.
AWS Architecture Center	Provides the necessary guidance and best practices to build highly scalable and reliable applications in the AWS cloud. These resources help you understand the AWS platform, its services and features. They also provide architectural guidance for design and implementation of systems that run on the AWS infrastructure.
AWS Economics Center	Provides access to information, tools, and resources to compare the costs of Amazon Web Services with IT infrastructure alternatives.
AWS Cloud Computing Whitepapers	Features a comprehensive list of technical AWS whitepapers covering topics such as architecture, security, and economics. These whitepapers have been authored either by the Amazon team or by AWS customers or solution providers.
Videos and Webinars	Previously recorded webinars and videos about products, architecture, security, and more.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and AWS Premium Support. (subscription required).
AWS Premium Support Information	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.

Resource	Description
Form for questions related to your AWS account: Contact Us	This form is *only* for account questions. For technical questions, use the Discussion Forums.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Document History

This document history is current as of the 2013-11-14 release of *Getting Started with AWS Analyzing Big Data with AWS*.

Change	Description	Release Date
Added Sentiment Analysis tutorial	Added new tutorial illustrating the use of Amazon EMR for analyzing the sentiment of Twitter data.	14 November 2013
Updated Amazon EMR console references	Updated to reflect improvements to the Amazon EMR console.	13 November 2013
Updated Amazon EC2 and Amazon EMR Pricing	Updated to reflect reductions in Amazon EC2 and Amazon EMR service fees.	6 March 2012
Updated Amazon S3 Pricing	Updated to reflect reduction in Amazon S3 storage fees.	9 February 2012
New content	Created new document.	12 December 2011