Amazon Elastic MapReduce

Getting Started Guide

API Version 2009-03-31

Amazon Web Services

Amazon Elastic MapReduce: Getting Started Guide

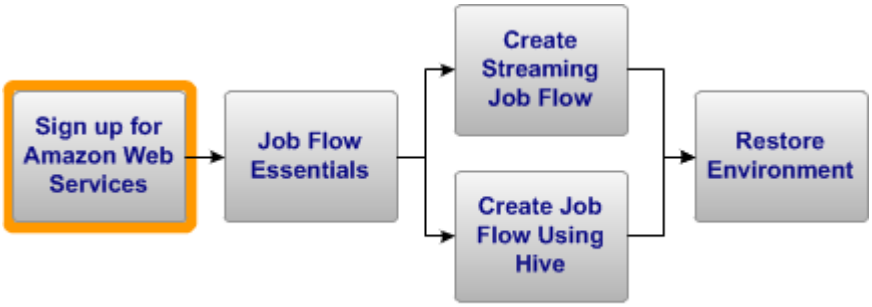
Amazon Web Services

Get Started with Amazon EMR

This *Amazon EMR Getting Started Guide* provides a high-level overview of the features found in Amazon Elastic MapReduce (Amazon EMR). After reading this guide, you should understand the basics of Amazon EMR. These examples show you how to use the Amazon EMR command line interface to create Hadoop streaming and Hive job flows, and how to use the Amazon EMR console to monitor and debug running job flows.

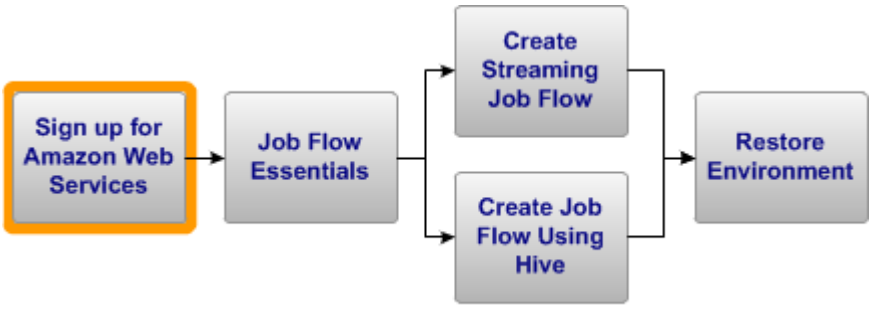
Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to process large amounts of data efficiently. Amazon EMR uses Hadoop processing combined with several AWS products to do such tasks as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehousing.

You can get started with Amazon Elastic MapReduce (Amazon EMR) by following the tasks shown in the following diagram.



This guide walks you through launching and managing job flows. To start using Amazon EMR for the first time, move on to Sign Up and Install the Command Line Interface (p. 2).

Sign Up and Install the Command Line Interface



Topics

This section describes the AWS account creation tasks and system configuration you need to perform before using Amazon Elastic MapReduce (Amazon EMR).

Creating an Amazon Web Services Account

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

> **Note**
> When you create an account, AWS automatically signs up the account for all services. You are charged only for the services you use.

To create an AWS account

1. Go to http://aws.amazon.com, and then click **Sign Up Now**.
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.

Install the Amazon EMR Command Line Interface

Topics

You can create job flows consisting of multiple steps using the Amazon EMR command line interface (CLI). The Amazon EMR console supports creating only single-step job flows. This document primarily describes how to manage job flows with the Amazon EMR CLI. For more information about how to use the Amazon EMR console and the Amazon EMR API, see the Amazon Elastic MapReduce Developer Guide and the Amazon Elastic MapReduce API Reference.

Installing Ruby

The Amazon EMR CLI requires Ruby 1.8.7 and is not compatible with later versions of Ruby. After you have installed Ruby, unzip elastic-mapreduce-ruby.zip into a directory, and the Amazon EMR CLI is ready to use.

To install Ruby

1. Download and install Ruby 1.8.7:

   - Linux and UNIX users can download Ruby from http://www.ruby-lang.org/en/news/2010/06/23/ruby-1-8-7-p299-released/ and install Ruby by entering the command:

     ```
     sudo apt-get install ruby-full
     ```

   - Windows users can install Ruby 1.8.7 from http://rubyforge.org/frs/?group_id=167&release_id=28426. During the installation process, select the checkboxes to add Ruby executables to your PATH environmental variable and to associate .rb files with this Ruby installation.
   - Mac OS X comes with Ruby installed. You can check the version as shown in the following step.

2. Verify that Ruby is running by typing the following at the command prompt:

   ```
   ruby -v
   ```

The Ruby version is shown, confirming that you installed Ruby. The output should be similar to the following:

```
ruby 1.8.7 (2012-02-08 patchlevel 358) [universal-darwin11.0]
```

Installing the Command Line Interface

To download the Amazon EMR CLI

1. Create a new directory to install the Amazon EMR CLI into. From the command-line prompt, enter the following:

```
mkdir elastic-mapreduce-cli
```

2. Download the Amazon EMR files:

   a. Go to http://aws.amazon.com/developertools/2264. If you are not logged in to AWS, enter your AWS account credentials when prompted.
   b. Click **Download**.
   c. Save the file in your newly created directory.

To install the Amazon EMR CLI

1. Navigate to your `elastic-mapreduce-cli` directory.
2. Unzip the compressed file:

   - Linux, UNIX, and Mac OS X users, from the command-line prompt, enter the following:

```
unzip elastic-mapreduce-ruby.zip
```

   - Windows users, from Windows Explorer, open the `elastic-mapreduce-ruby.zip` file and select **Extract all files**.

Configuring Credentials

The Amazon EMR credentials file can provide information required for many commands. You can also store command parameters in the file so you don't have to repeatedly enter that information at the command line each time you create a job flow.

Your credentials are used to calculate the signature value for every request you make. Amazon EMR automatically looks for your credentials in the file `credentials.json`. It is convenient to edit the `credentials.json` file and include your AWS credentials. An AWS key pair is a security credential similar to a password, which you use to securely connect to your instance when it is running. We recommend that you create a new key pair to use with this guide.

To create your credentials file

1. Create a file named `credentials.json` in the directory where you unzipped the Amazon EMR CLI.
2. Add the following lines to your credentials file:

```
{
"access_id": "Your AWS Access Key ID",
```

```
"private_key": "Your AWS Secret Access Key",
"keypair": "Your key pair name",
"key-pair-file": "The path and name of your PEM file",
"log_uri": "A path to a bucket you own on Amazon S3, such as, s3n://mylog-uri/",
"region": "The region of your job flow, either us-east-1, us-west-2, us-west-1, eu-west-1, ap-northeast-1, ap-southeast-1, ap-southeast-2, or sa-east-1"
}
```

Note the name of the region. Use this region to create your Amazon EC2 key pair and your Amazon S3 bucket.

The next sections explain how to create and find your credentials.

AWS Security Credentials

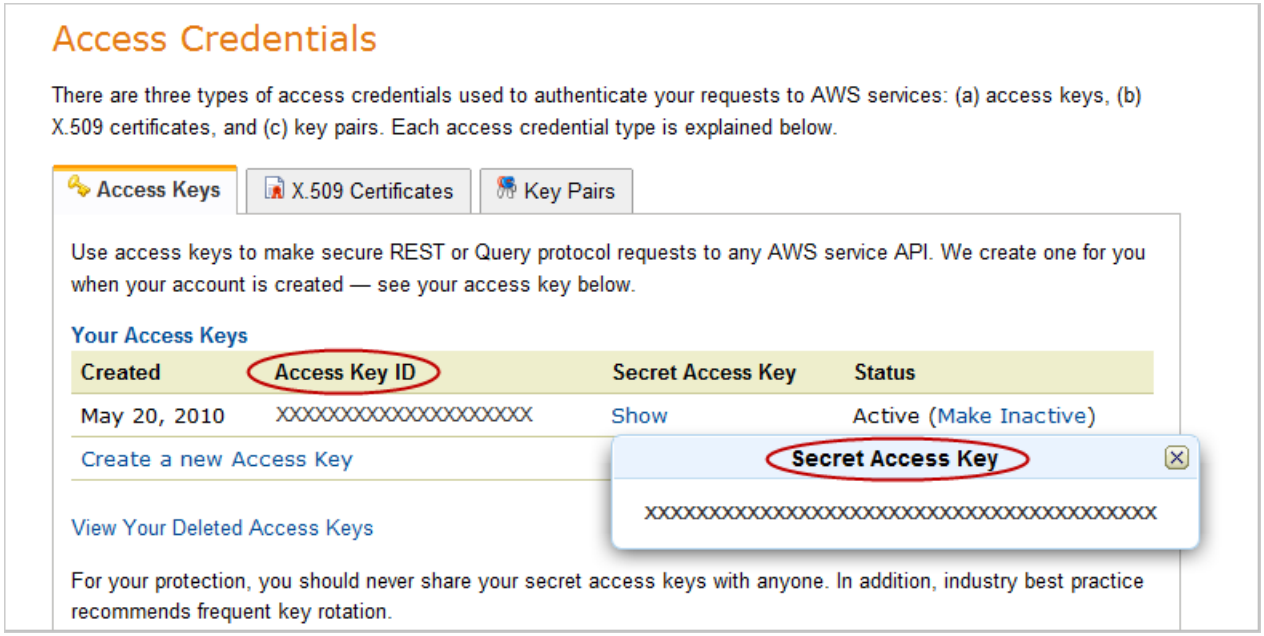
AWS uses security credentials to help protect your data. This section, shows you how to view your security credentials so you can add them to your `credentials.json` file.

AWS assigns you an Access Key ID and a Secret Access Key. You include your Access Key ID in all AWS service requests to identify yourself as the sender of the request.

Note
Your Secret Access Key is a shared secret between you and AWS. Keep this ID secret; we use it to bill you for the AWS services you use. Never include the ID in your requests to AWS and never email the ID to anyone even if an inquiry appears to originate from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your Secret Access Key.

To locate your AWS Access Key ID and AWS Secret Access Key

1. Go to the AWS web site at http://aws.amazon.com.
2. Click **My Account** to display a list of options.
3. Click **Security Credentials** and log in to your AWS account. Your **Access Key ID** is displayed in the **Access Credentials** section. Your **Secret Access Key** remains hidden as a further precaution.
4. To display your Secret Access Key, click **Show** in the **Your Secret Access Key** area, as shown in the following figure.



Set your *access_key* parameter to the value of your Access Key ID and set your *private_key* parameter to the value of your Secret Access Key.

To create an Amazon EC2 key pair

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the **EC2 Dashboard**, select the **Region** you used in your credentials.json file, then click **Key Pair**.
3. On the **Key Pairs** page, click **Create Key Pair**.
4. Enter a name for your key pair, such as, **mykeypair**.
5. Click **Create**.
6. Save the resulting PEM file in a safe location.

In your `credentials.json` file, change the *keypair* parameter to your Amazon EC2 key pair name and change the *key-pair-file* parameter to the location and name of your PEM file. This PEM file is what the CLI uses as the default for the Amazon EC2 key pair for the Amazon EC2 instances it creates when it launches a job flow.

Amazon S3 Bucket

The *log-uri* parameter specifies a location in Amazon S3 for the Amazon EMR results and log files from your job flow. The value of the *log-uri* parameter is an Amazon S3 bucket that you create for this purpose.

To create an Amazon S3 bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Click **Create Bucket**.
3. In the **Create a Bucket** dialog box, enter a bucket name, such as **mylog-uri**.

This name should be globally unique, and cannot be the same name used by another bucket. For more information about valid bucket names, see http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html.

4. Select the **Region** for your bucket.

If your Amazon EMR region is...	Select the Amazon S3 region...
us-east-1	**US Standard**
us-west-2	**Oregon**
us-west-1	**Northern California**
eu-west-1	**Ireland**
ap-northeast-1	**Japan**
ap-southeast-1	**Singapore**
sa-east-1	**Sao Paulo**
us-gov-west-1	**GovCloud**

Note
To use the AWS GovCloud region, contact your AWS business representative. You can't create an AWS GovCloud account on the AWS website. You must engage directly with AWS and sign an AWS GovCloud (US) Enterprise Agreement. For more information, see the AWS GovCloud (US) Product Page.

5. Click **Create**.

Note
If you enable logging in the **Create a Bucket** wizard, it enables only bucket access logs, not Amazon EMR job flow logs.

You have created a bucket with the URI **`s3n://mylog-uri/`**.

After creating your bucket, set the appropriate permissions on it. Typically, you give yourself (the owner) read and write access and give authenticated users read access.

To set permissions on an Amazon S3 bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** pane, right-click the bucket you just created.
3. Select **Properties**.
4. In the **Properties** pane, select the **Permissions** tab.
5. Click **Add more permissions**.
6. Select **Authenticated Users** in the **Grantee** field.
7. To the right of the **Grantee** field, select **List**.
8. Click **Save**.

You have now created a bucket and assigned it permissions. Set your *`log-uri`* parameter to this bucket's URI as the location for Amazon EMR to upload your logs and results.

SSH Setup and Configuration

Configure your SSH credentials for use with either SSH or PuTTY. This step is required.

To configure your SSH credentials

- Configure your computer to use SSH:
  - Linux and UNIX users, set the permissions on the PEM file for your Amazon EC2 key pair. For example, if you saved the file as **`mykeypair.pem`**, the command looks like:

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

  - Windows users
    a. Windows users use PuTTY to connect to the master node. Download PuTTYgen.exe to your computer from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.
    b. Launch PuTTYgen.
    c. Click **Load**. Select the PEM file you created earlier.
    d. Click **Open**.
    e. Click **OK** on the **PuTTYgen Notice** telling you the key was successfully imported.
    f. Click **Save private key** to save the key in the PPK format.
    g. When PuTTYgen prompts you to save the key without a pass phrase, click **Yes**.
    h. Enter a name for your PuTTY private key, such as, **`mykeypair.ppk`**.
    i. Click **Save**.
    j. Exit the PuTTYgen application.

Verify installation of the Amazon EMR CLI

- In the directory where you installed the CLI, run the following commands from the command line:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --version
```

  - Windows users:

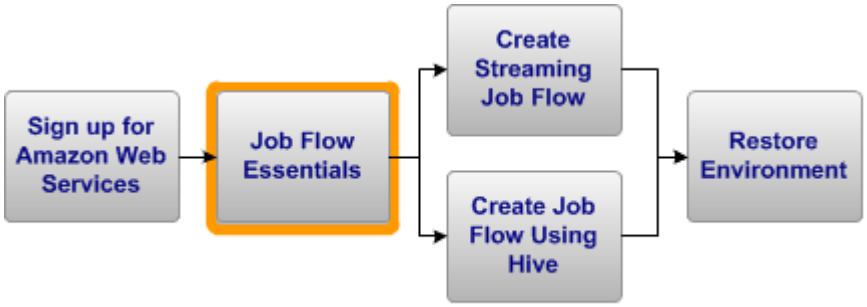
```
ruby elastic-mapreduce --version
```

    If the CLI is correctly installed and the credentials properly configured, the CLI should display its version number represented as a date. The output should look similar to the following:

```
Version 2012-12-17
```

Now that you have signed up for Amazon EMR, installed the Amazon EMR CLI, and configured your settings, move on to Job Flow Essentials (p. 9).

Job Flow Essentials



Topics

This section provides general information on how to create and manage job flows using the Amazon EMR command line interface (CLI).

Amazon Elastic MapReduce (Amazon EMR) takes care of provisioning an Amazon EC2 cluster, terminating it, moving the data between it and Amazon S3, and optimizing Hadoop. Amazon EMR removes most of the details of setting up the hardware and networking required by the server cluster, such as monitoring the setup, configuring Hadoop, and executing the job flow.

Creating a Job Flow

Using the Amazon EMR CLI, you can construct a job flow that will continue to run until you terminate it. This process is useful for debugging. When a step fails, you can add another step to your active job flow without having to incur the shutdown and startup cost of a new job flow.

Typically, a step involves performing relatively simple operations on very large amounts of data. A step corresponds roughly to one algorithm that manipulates the data. A job flow typically consists of multiple steps. The output of one step often becomes the input of the next. A sequence of one or more steps is called a *job flow*.

The following command starts a job flow that consumes resources until you terminate it.

To create a job flow

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --create --alive
```

  - Windows users:

```
ruby elastic-mapreduce --create --alive
```

The output will look similar to:

```
Created job flow JobFlowID
```

This command launches a job flow running on a single m1.small instance. The `--alive` option tells the job flow to keep running even when it has finished all its steps.

A unique job flow ID is assigned to each newly created job flow. You use the job flow ID to identify and manage your job flow.

Managing a Job Flow

This section presents several methods to identify and manage your job flows.

List All Amazon EMR Commands

You can use the `--help` parameters to list all of the commands available in the Amazon EMR CLI.

To list all Amazon EMR commands

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --help
```

  - Windows users:

```
ruby elastic-mapreduce --help
```

For more information on each of the Amazon EMR commands, see the Amazon Elastic MapReduce Developer Guide.

List All Job Flows

You can use the *--list* parameter to list all of your job flows for the past two weeks.

To list all job flows

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --list
```

  - Windows users:

```
ruby elastic-mapreduce --list
```

The response looks similar to the following:

```
JobFlowID     STARTING
   Development Job Flow  (requires manual termination)
```

For details on job flow STATES and additional methods to list job flows, see the Amazon Elastic MapReduce Developer Guide.

Retrieve Information About a Specific Job Flow

You can get information about a job flow using the *--describe* option and the associated job flow ID.

To get information about your job flow

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --describe --jobflow JobFlowID
```

  - Windows users:

```
ruby elastic-mapreduce --describe --jobflow JobFlowID
```

The response looks similar to the following:

```
{
  "JobFlows": [
    {
      "BootstrapActions": [],
      "SupportedProducts": [],
      "JobFlowId": "j-2Z2DREHZJG0IM",
      "Instances": {
        "MasterPublicDnsName": "ec2-107-22-71-101.compute-1.amazonaws.com",
        "InstanceGroups": [
          {
            "State": "RUNNING",
            "InstanceType": "m1.small",
            "EndDateTime": null,
            "CreationDateTime": 1357324832.0,
            "Market": "ON_DEMAND",
            "BidPrice": null,
            "InstanceRequestCount": 1,
            "LaunchGroup": null,
            "InstanceRole": "MASTER",
            "StartDateTime": 1357325011.0,
            "InstanceGroupId": "ig-3SQQQNPT1HF25",
            "LastStateChangeReason": "",
            "ReadyDateTime": 1357325109.0,
            "Name": "Master Instance Group",
            "InstanceRunningCount": 1
          }
        ],
        "KeepJobFlowAliveWhenNoSteps": true,
        "MasterInstanceType": "m1.small",
        "NormalizedInstanceHours": 1,
        "Ec2SubnetId": null,
        "HadoopVersion": "1.0.3",
        "Ec2KeyName": "mykeypair",
        "InstanceCount": 1,
        "SlaveInstanceType": null,
        "Placement": {
          "AvailabilityZone": "us-east-1d"
        },
        "TerminationProtected": false,
        "MasterInstanceId": "i-a3a973d2"
      },
      "ExecutionStatusDetail": {
        "State": "WAITING",
        "EndDateTime": null,
        "CreationDateTime": 1357324832.0,
        "StartDateTime": 1357325113.0,
        "LastStateChangeReason": "Waiting for steps to run",
        "ReadyDateTime": 1357325113.0
      },
      "Steps": [],
      "LogUri": "s3n:\/\/syne-test\/logs\/",
      "AmiVersion": "2.3.1",
      "VisibleToAllUsers": false,
      "Name": "Development Job Flow (requires manual termination)",
      "JobFlowRole": null
    }
  ]
}
```

For details on job flow parameter names and values, see the Amazon Elastic MapReduce Developer Guide and the Amazon Elastic MapReduce API Reference.

Debugging Job Flows

To use Amazon EMR debugging you must specify an Amazon S3 bucket location in your `credentials.json` file. You specified the *`log_uri`* parameter in the file you created as part of the Configuring Credentials (p. 4) step.

You access Amazon EMR log files either by using the Amazon EMR console or by viewing them directly from the Amazon S3 console.

> **Note**
> A five-minute delay occurs between when the log files stop being written and when they are available on Amazon S3.

Hadoop debugging is also available to identify issues and problems in your job flows. For details on how to enable and configure Hadoop debugging, see the Amazon Elastic MapReduce Developer Guide.

Adding Steps to a Streaming Job Flow

You can add steps to a job flow if the *`RunJobFlow`* parameter *`KeepJobFlowAliveWhenNoSteps`* is set to *`True`*. This value keeps the Amazon EC2 cluster engaged even after the successful completion of a job flow. The default setting for *`KeepJobFlowAliveWhenNoSteps`* is *`True`* and can be verified using the *`--describe --jobflow`* ***`JobFlowID`*** commands. To identify your job flow ID, refer to the preceding Retrieve Information About a Specific Job Flow (p. 11) section.

To add a step to a job flow

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce -j JobFlowID --stream \
--mapper  s3://elasticmapreduce/samples/wordcount/wordSplitter.py \
--input   s3://elasticmapreduce/samples/wordcount/input \
--output  s3n://myawsbucket/output \
--reducer aggregate
```

  - Windows users:

```
ruby elastic-mapreduce -j JobFlowID --stream --mapper s3://elasticmapre
duce/samples/wordcount/wordSplitter.py --input s3://elasticmapre
duce/samples/wordcount/input --output  s3n://myawsbucket/output --reducer
 aggregate
```

The *`--stream`* command adds a streaming step using the specified parameters for input, output, mapper and reducer. In the Amazon EMR console, *Hadoop streaming* is a feature of Hadoop that lets you create and run job flows using any executable program or script as Hadoop mappers and reducers. You can view the step you just added in the Amazon EMR console from either the CLI or the Amazon EMR console.

To view a job flow from the Amazon EMR 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 **Refresh**.
3. Click the job flow with the added step.
4. In the **Details** pane at the bottom of the window, click the **Steps** tab.

Information about the step you added is displayed in the **Steps** tab.

Terminate a Job Flow

Once you finish working with a job flow, you terminate it so you are no longer being charged for using AWS resources.

To terminate a job flow

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --terminate JobFlowID
```

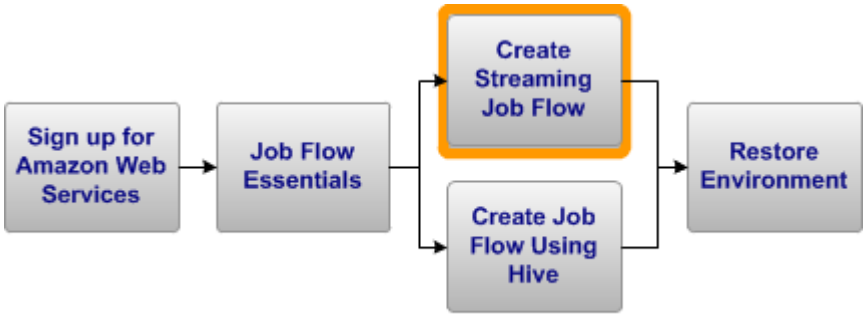
  - Windows users:

```
ruby elastic-mapreduce --terminate JobFlowID
```

Congratulations! You have successfully created and terminated an Amazon EMR instance and learned about a few of the options available to you.

Now that you know how to create, debug, and terminate a job flow, move on to Create a Streaming Job Flow (p. 15).

Create a Streaming Job Flow



This example shows how to use Hadoop streaming to count the number of times that a word occurs in a data set. This type of job flow is appropriate if you want to search a large number of logs for a particular error or you want to know the number of blog posts made for each user name. Hadoop streaming enables you to execute MapReduce programs written in languages such as Python, Ruby, and PHP.

To count the occurrence of words, you need a mapper function that iterates through the input data and outputs word-count pairs. You can create a mapper function in Python as shown in the following example:

```
#!/usr/bin/python
import sys
import re

def main(argv):
    pattern = re.compile("[a-zA-Z][a-zA-Z0-9]*")
    for line in sys.stdin:
        for word in pattern.findall(line):
            print "LongValueSum:" + word.lower() + "\t" + "1"


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

To run the Hadoop streaming job with Amazon Elastic MapReduce (Amazon EMR), this mapper function must be uploaded to Amazon S3.

You can save this Python script to your own Amazon S3 location. For your convenience, this example is stored on Amazon S3 at the location `s3://elasticmapreduce/samples/wordcount/wordSplitter.py`.

The sample input for this job flow is available at `s3://elasticmapreduce/samples/wordcount/input`.

This example uses the built-in reducer called *`aggregate`*. This reducer adds up the counts of words being output by the wordSplitter mapper function. It knows to use data type Long from the prefix on the words.

To run a streaming job flow

- Enter the following commands from the command-line prompt in the directory where you installed the CLI. Each time you run a Hadoop streaming job flow you must specify a new *`--output`* location or the job flow will fail. You can specify a folder within an existing bucket as well as create a new bucket.

  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --create --stream \
  --mapper  s3://elasticmapreduce/samples/wordcount/wordSplitter.py \
  --input   s3://elasticmapreduce/samples/wordcount/input \
  --output  A path to a bucket you own on Amazon S3, such as, s3n://myaws
bucket \
  --reducer aggregate
```

  - Windows users:

```
ruby elastic-mapreduce --create --stream  --mapper  s3://elasticmapre
duce/samples/wordcount/wordSplitter.py --input   s3://elasticmapre
duce/samples/wordcount/input --output  A path to a bucket you own on Amazon
 S3, such as, s3n://myawsbucket --reducer aggregate
```

The output will look similar to:

```
Created job flow JobFlowID
```

This sample may take several minutes to run. You can monitor the job flow from the CLI as described in the Retrieve Information About a Specific Job Flow (p. 11) step or from the Amazon EMR console.

To view the streaming job flow

1. Sign in to the AWS Management Console and open the Amazon Elastic MapReduce console at https://console.aws.amazon.com/elasticmapreduce/.
2. Click **Refresh**.
3. Click the Hadoop streaming job flow. The Hadoop streaming job flow is listed with a *STATE*.
4. Click **Debug**.

   If the job flow *STATE* is `COMPLETED`, links to the Amazon EMR log files are displayed.
5. If the job flow is not completed, click **Close**, wait a minute, and then attempt Step **4** again.

Note
The Actions column has a link to **View Jobs**. Clicking this link displays an alert. Jobs, Tasks, and Task Attempts are not available because you did not enable debugging when you created this job flow. You must enable and configure Hadoop debugging to create these additional results.

6. After you have viewed the Amazon EMR log files, click **Close**.

You can find additional Amazon EMR log files in the Amazon S3 bucket you specified in your `credentials.json` file.

For information about the contents of these logs, see the Amazon Elastic MapReduce Developer Guide.

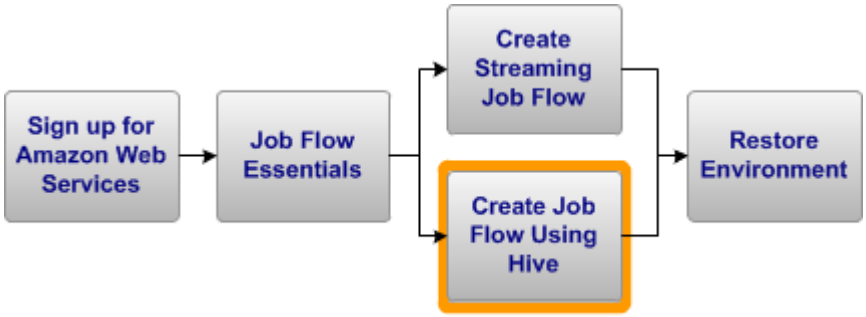
To view job flow results

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Navigate to the Amazon S3 bucket you referenced in `--output`.

Your job flow results are stored in a text file. The results file contains a list of all words found with the number of times the word occurred in the data set.

Now that you have completed a Hadoop streaming job flow, move on to Create a Job Flow Using Hive (p. 18).

Create a Job Flow Using Hive



Topics

This sample Hive script combines advertisement impression and click log data to evaluate the success of targeted online advertising. The script combines the two sets of log data, places the information into a Hive cluster, and outputs the results to a specified directory. The following script processes all impressions that occurred between 2009-04-13 8:00 and 2009-04-13 9:00 and were referred by twitter.com from a Mozilla browser.

A detailed description of this business problem can be found in the tutorial, *Contextual Advertising Using Hive and Amazon EMR*
http://developer.amazonwebservices.com/connect/entry!default.jspa?categoryID=269&externalID=2855.

Hive provides tools to summarize data, query, and analyze large data sets stored in Hadoop files. It provides a simple query language called Hive QL which is based on SQL. Hive allows traditional map/reduce programmers to plug in custom mappers and reducers for more sophisticated analysis.

Create a Hive Script

For your convenience, this sample script is stored on Amazon S3 at `s3://elasticmapreduce/samples/hive-ads`. You can also save this script to your own Amazon S3 location and change the Hive command appropriately.

Sample data for this job flow is available at `s3://elasticmapreduce/samples/hive-ads/libs/twitter-impressions.q`.

The commented script follows:

- A custom SerDe is used to read the advertisement impressions data. A SerDe is a serializer/deserializer and handles converting the data from one format to another. For more information about how Hive uses a SerDe, go to https://cwiki.apache.org/confluence/display/Hive/SerDe.

```
ADD JAR ${SAMPLE}/libs/jsonserde.jar ;
```

- An external table is created to instruct Hive on how to organize the advertisement impressions data.

```
CREATE EXTERNAL TABLE impressions (
   requestBeginTime string, adId string, impressionId string, referrer string,

    userAgent string, userCookie string, ip string
  )
  PARTITIONED BY (dt string)
  ROW FORMAT
    serde 'com.amazon.elasticmapreduce.JsonSerde'
    with serdeproperties ( 'paths'='requestBeginTime, adId, impressionId,
    referrer, userAgent, userCookie, ip' )
  LOCATION '${SAMPLE}/tables/impressions' ;
```

- A single partition table is created and partitioned based on time.

```
ALTER TABLE impressions ADD PARTITION (dt='2009-04-13-08-05');
```

- Temporary tables are created in the job flow's local HDFS partition to store intermediate advertisement impressions and click data.

```
CREATE TABLE tmp_impressions (
   requestBeginTime string, adId string, impressionId string, referrer string,

    userAgent string, userCookie string, ip string
  )
  STORED AS SEQUENCEFILE ;
```

- Data from the advertisement impressions table for a specified time period is inserted into the partitioned table.

```
INSERT OVERWRITE TABLE tmp_impressions
    SELECT
      from_unixtime(cast((cast(i.requestBeginTime as bigint) / 1000) as int))
 requestBeginTime,
      i.adId, i.impressionId, i.referrer, i.userAgent, i.userCookie, i.ip
    FROM
      impressions i
    WHERE
      i.dt   '{DAY}-${HOUR}-00' and i.dt < '{NEXT_DAY}-${NEXT_HOUR}-00'
;
```

- Specific impression data is stored in an output table on Amazon S3.

```
CREATE EXTERNAL TABLE output_impressions (
   requestBeginTime string, adId string, impressionId string, referrer string,

    userAgent string, userCookie string, ip string
    )
    PARTITIONED BY (day string, hour string)
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY '\t'
    LINES TERMINATED BY '\n'
    STORED AS TEXTFILE
    LOCATION '${OUTPUT}/impressions'
  ;
```

- The output table is populated with all advertisement impressions referred by twitter.com through a Mozilla browser during the specified time period.

```
INSERT OVERWRITE TABLE output_impressions PARTITION (day='${DAY}',
hour='${HOUR}')
  SELECT
    i.requestBeginTime, i.adId, i.impressionId, i.referrer, i.userAgent,
i.userCookie, i.ip
  FROM
    tmp_impressions i
  WHERE
    i.referrer = 'twitter.com' and i.userAgent like '%Mozilla%'
    ;
```

Launch a Job Flow Using Hive

To run the job flow with Hive, create an Amazon Elastic MapReduce (Amazon EMR) job flow using the CLI, log in to the job flow's master node, and then launch the Hive script.

To create a job flow using Hive

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --create --alive --name "Hive Job Flow" --hive-inter
active
```

  - Windows users:

```
ruby elastic-mapreduce --create --alive --name "Hive Job Flow" --hive-in
teractive
```

The output will look similar to:

```
Created job flow JobFlowID
```

This job flow takes a few minutes to transition from the *STARTING* to the *WAITING* state. You can monitor the job flow from the CLI as described in the Retrieve Information About a Specific Job Flow (p. 11) step or from the Amazon EMR console.

To list all active job flows using the CLI

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

    ```
    ./elastic-mapreduce --list --active
    ```

  - Windows users:

    ```
    ruby elastic-mapreduce --list --active
    ```

The list of active job flows initially looks similar to the following:

```
JobFlowID      STARTING
     Hive Job Flow
   PENDING        Setup Hive
```

When the job flow is ready to accept the Hive script, it looks similar to:

```
JobFlowID      WAITING        ec2-184-72-128-177.compute-1.amazonaws.com
     Hive Job Flow
   COMPLETED      Setup Hive
```

The DNS to the master node and the root login are required to connect to the master node. The DNS can be found in the output of an active job flow. In this sample, the DNS is `ec2-184-72-128-177.compute-1.amazonaws.com`. The root login or username is `hadoop`.

When the job flow is in the *WAITING* state, connect to the master node using SSH.

To connect to the master node

1. In the directory where you installed the CLI, enter the following commands from the command-line prompt:
   - Linux, UNIX, and Mac OS X users:

     ```
     ./elastic-mapreduce --ssh --jobflow JobFlowID
     ```

     Use the *job flow ID* of the sample job flow.
   - Windows users:

a. Start PuTTY. (For more information about how to install PuTTY and use it to connect to an EC2 instance, such as the master node, go to Appendix D: Connecting to a Linux/UNIX Instance from Windows using PuTTY in the Amazon Elastic Compute Cloud User Guide.)
b. Select **Session** in the **Category** list. Enter **hadoop@*DNS*** in the **Host Name** field. The input looks similar to *hadoop@ec2-184-72-128-177.compute-1.amazonaws.com*.
c. In the **Category** list, expand **Connection**, expand **SSH**, and then select **Auth**. The **Options controlling SSH authentication** pane appears.
d. Click **Browse** for **Private key file for authentication**, and select the private key file you generated earlier. If you are following this guide, the file name is **mykeypair.ppk**.
e. Click **OK**.
f. Click **Open** to connect to your master node.
g. A **PuTTY Security Alert** pops up. Click **Yes**.

When you successfully connect to the master node, the output looks similar to the following:

```
Using username "hadoop".
Authenticating with public key "imported-openssh-key"
Linux domU-12-31-39-01-5C-F8 2.6.21.7-2.fc8xen #1 SMP Fri Feb 15 12:39:36
EST 2008 i686
------------------------------------------------------------------------
------

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/

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

2. Run the sample Hive script with the following command.

```
hive \
    -d SAMPLE=s3://elasticmapreduce/samples/hive-ads \
    -d DAY=2009-04-13 -d HOUR=08 \
    -d NEXT_DAY=2009-04-13 -d NEXT_HOUR=09 \
    -d OUTPUT=A path to a bucket and a folder you own on Amazon S3, such
as, s3://myawsbucket/folder \
    -f s3://elasticmapreduce/samples/hive-ads/libs/twitter-impressions.q
```

The Hive script is added to the job flow. The output looks similar to the following:

```
10/08/20 14:57:34 WARN conf.Configuration: DEPRECATED: hadoop-site.xml found
 in the classpath.
Usage of hadoop-site.xml is deprecated. Instead use core-site.xml, mapred-
site.xml and hdfs-site.xml to
override properties of core-default.xml, mapred-default.xml and hdfs-de
```

```
fault.xml respectively
Hive history file=/mnt/var/lib/hive/tmp/history/hive_job_log_ha
doop_201008201457_1658787617.txt
Testing s3://elasticmapreduce/samples/hive-ads/libs/jsonserde.jar
converting to local s3://elasticmapreduce/samples/hive-ads/libs/jsonserde.jar
Added /mnt/var/lib/hive/downloaded_resources/s3_elasticmapreduce_samples_hive-
ads_libs_jsonserde.jar
to class path
Found class for com.amazon.elasticmapreduce.JsonSerde
OK
Time taken: 11.531 seconds

...


Starting Job = job_201008201445_0003, Tracking URL = http://domU-12-31-39-
01-5C-F8.compute-1.internal:
9100/jobdetails.jsp?jobid=job_201008201445_0003
Kill Command = /home/hadoop/.versions/0.20/bin/../bin/hadoop job  -
Dmapred.job.tracker=
domU-12-31-39-01-5C-F8.compute-1.internal:9001 -kill job_201008201445_0003
2010-08-20 14:59:07,714 Stage-2 map = 0%,  reduce = 0%
2010-08-20 14:59:22,254 Stage-2 map = 100%,  reduce = 0%
2010-08-20 14:59:31,450 Stage-2 map = 100%,  reduce = 33%
2010-08-20 14:59:37,608 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_201008201445_0003
Loading data to table output_impressions partition {day=2009-04-13, hour=08}
30 Rows loaded to output_impressions
OK
Time taken: 64.647 seconds
```

Your job flow step is completed.

To quit ssh or PuTTY

- Type **`exit`** and press **ENTER**.

To terminate a job flow

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

    ```
    ./elastic-mapreduce --terminate JobFlowID
    ```

  - Windows users:

    ```
    ruby elastic-mapreduce --terminate JobFlowID
    ```

To view the results of your job flow

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Navigate to the Amazon S3 bucket and path you referenced in your Hive script as part of `-d OUTPUT`. The results for this sample will be located in a text file in the folder `\impressions\day=2009-04-13\hour=08`.

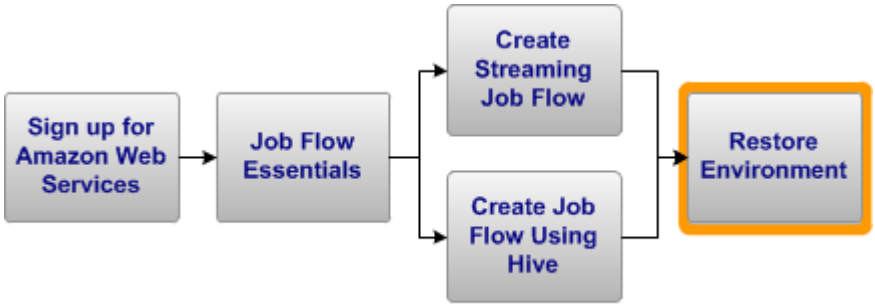
Your job flow results are stored in a text file.

You can find additional Amazon EMR log files in the Amazon S3 bucket you specified in your `credentials.json` file.

For information about the contents of these logs, see the Amazon Elastic MapReduce Developer Guide.

Now that you completed a job flow using Hive, find out how to clean up your resources so you do not incur any unnecessary charges. To do so, move on to Restore Environment (p. 25).

Restore Environment



You have completed the Amazon Elastic MapReduce (Amazon EMR) samples described in this guide.

To make sure you are not charged for any left-over services, delete any unwanted job flows and files from the Amazon EMR and Amazon S3 services.

Stop Amazon EMR Job Flows

You can verify that you are not using any Amazon EMR resources by listing your active job flows, and then terminating those you no longer need.

To list all active job flows

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux, UNIX, and Mac OS X users:

```
./elastic-mapreduce --list --active
```

  - Windows users:

```
ruby elastic-mapreduce --list --active
```

Use the job flow ID to identify each job flow you want to terminate.

To terminate a job flow

- In the directory where you installed the CLI, enter the following commands from the command-line prompt:
  - Linux and UNIX users, from the command-line prompt, enter the following:

```
./elastic-mapreduce --terminate job flow ID
```

  - Windows users, from the command-line prompt, enter the following:

```
ruby elastic-mapreduce  --terminate job flow ID
```

Terminating all job flows will remove all associated Amazon EC2 instances. Depending on the configuration of the job flow, it may take up to 5-20 minutes for the job flow to completely terminate and release allocated resources.

Remove Log Files

By specifying the *log-uri* as part of the Configuring Credentials (p. 4) step, all of your job flows generated Amazon EMR logs and saved them to Amazon S3.

If you no longer require the Amazon EMR log files, delete the files so you will not be charged for Amazon S3 storage.

To delete a file on Amazon S3

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Navigate to the bucket and folder specified as your *log-uri* by clicking the bucket name in the **Bucket** pane, and then clicking the folder in the **Objects and Folders** pane.
3. Click **Actions** and select **Delete** to delete a folder and all of its contents.

You are no longer being charged for any services you used as part of this tutorial.

Congratulations! You have successfully launched, connected to, and terminated a job flow. For more information about Amazon Elastic MapReduce (Amazon EMR) and how to continue using it, see Where Do I Go from Here? (p. 27).

Where Do I Go from Here?

Topics

Amazon Elastic MapReduce (Amazon EMR) is a rich service offering many features than are not covered in this guide, such as Hadoop logging& Pig, and Custom JAR job flows& Bootstrap Action&, and virtual private networking. This section provides links to additional resources, that will help you deepen your understanding of Amazon EMR.

Other Ways to Access Amazon EMR

This guide has shown you how to launch and terminate job flows using Amazon EMR. You can continue using Amazon EMR through the command line interface, or try one of the other interfaces.

Continue Using the Command Line Interface

To learn more about the Amazon EMR command line interface, refer to the Amazon Elastic MapReduce Developer Guide. The CLI offers full support of all the Amazon EMR functions without requiring you to code or use the Amazon EMR library.

Use the Amazon EMR Console

The Amazon EMR console includes many functions besides just monitoring debug output. To learn more about how to use the Amazon EMR console, go to the Amazon Elastic MapReduce Developer Guide. The Amazon EMR console also has help to assist you.

Code Directly to the Web Service API

If you want to write code directly to the Amazon EMR Query API, go to the Amazon Elastic MapReduce Developer Guide. The guide describes how to create and authenticate API requests, and how to use

Amazon EMR through the APIs. For a complete description of all the API actions, go to the Amazon Elastic MapReduce API Reference.

Learn More About Amazon EMR

This section lists additional features in Amazon EMR and tells you where to find more information. You can also find additional information about Amazon EMR in the Amazon EMR Articles & Tutorials area of the AWS web site.

Streaming Job Flows

The sample streaming job flow provided in this guide highlights the basic capabilities of Amazon Elastic MapReduce (Amazon EMR). For more information on using streaming job flows with Amazon EMR consider the following tutorial:

- Tutorial: Finding Similar Items with Amazon EMR, Python, and Hadoop Streaming http://aws.amazon.com/articles/2294

Job Flows Using Hive

The sample job flow with Hive provided in this guide highlights the basic capabilities of using Hive with Amazon Elastic MapReduce (Amazon EMR). For more information on using Hive with Amazon EMR consider the following:

- Tutorial: Contextual Advertising using Apache Hive and Amazon EMR with High Performance Computing instances http://aws.amazon.com/articles/2855
- Video: Getting started with Hive on Amazon EMR http://aws.amazon.com/articles/2862

Job Flows Using Pig

Pig is an open-source Apache library that runs on top of Hadoop. The library takes SQL-like commands written in a language called Pig Latin and converts these commands into MapReduce job flows. Pig enables you to create queries using familiar SQL-like commands and syntax, avoiding the complexities of writing MapReduce algorithms using a lower-level language, such as Java. While you can execute one Pig Latin command at a time, it is far more common to write a script of Pig Latin commands that accomplish a task. Elastic MapReduce can use such scripts when you upload them to Amazon S3.

For more information on using Pig with Elastic Map Reduce consider the following:

- Tutorial: Parsing Logs with Apache Pig and Elastic MapReduce http://aws.amazon.com/articles/2729
- Video: Getting Started with Apache Pig on Elastic MapReduce http://aws.amazon.com/articles/2735

Job Flows Using Custom JAR files

A custom JAR job flow runs a compiled Java program that you have uploaded to Amazon S3. The program should be compiled against the version of Hadoop you want to launch and you should submit Hadoop jobs using the Hadoop JobClient interface.

For more information on using Elastic MapReduce with custom JAR files, consider the following tutorial.

- Tutorial: How to Create and Debug an Amazon EMR Job Flow http://aws.amazon.com/articles/3938

Job Flows Using Cascading

Cascading is an open-source project providing an API for defining and executing complex, scale-free, and fault tolerant data processing work flows on Hadoop.

For more information on using Cascading with Elastic Map Reduce consider the following tutorial.

- Tutorial: Cascading Multitool http://aws.amazon.com/jobflows/2293

Bootstrap Actions

Bootstrap actions are programs that you run on all nodes of a job flow prior to starting Hadoop. With bootstrap actions you can do the following:

- Install software on the node
- Modify the default Hadoop site configuration
- Change the way Java parameters use Hadoop daemons

You can specify a bootstrap action in the Amazon EMR console or the Amazon EMR command line client when starting job flows. Several predefined bootstrap actions are available, including Configure Hadoop, Configure Daemons, and Run-if.

For more information on Bootstrap Actions, see the Amazon Elastic MapReduce Developer Guide or refer to the following tutorial.

- Tutorial: How to Create and Debug an Amazon EMR Job Flowhttp://aws.amazon.com/articles/3938

Hadoop Debugging

In addition to Amazon EMR logging, you also have the option to generate detailed Hadoop logs. Hadoop logging must be enabled when a job flow is created and will use SimpleDB to store the logs.

For more information on Hadoop debugging, see the Amazon Elastic MapReduce Developer Guide.

Learn More About Hadoop

Apache Hadoop is an open-source Java software framework that supports data processing of large data sets using server clusters.

For more information on the Hadoop framework, go to http://hadoop.apache.org/core/.

Amazon EMR Resources

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

Resource	Description
Amazon Elastic MapReduce Getting Started Guide	This document. Provides a quick tutorial of the service based on a simple use case. Examples and instructions are included.

Resource	Description
Amazon Elastic MapReduce Developer Guide	Provides conceptual information about Amazon EMR and describes how to use Amazon EMR features.
Amazon Elastic MapReduce API Reference	Contains a technical description of all Amazon EMR APIs.
	Describes all of the command line parameters and their options.
Amazon EMR Technical FAQ	Covers the top questions developers have asked about this product.
Amazon EMR Release Notes	Gives a high-level overview of the current release, and notes any new features, corrections, and known issues.
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
Amazon EMR console	Enables you to perform most of the functions of Amazon EMR and other AWS products without programming.
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 (if you are subscribed to this program).
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.
Amazon EMR Product Information	The primary web page for information about Amazon EMR.
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.

Please Provide Feedback

Your input is important to help make our documentation helpful and easy to use. Please tell us about your experience getting started with Amazon Elastic MapReduce (Amazon EMR) by completing our Getting Started Survey.

Thank you.

Document History

This documentation is associated with the 2009-03-31 release of Amazon Elastic MapReduce. This guide was was last updated on 09 April 2013.

The following table describes the important changes since the last release of the *Amazon Elastic MapReduce Getting Started Guide*.

Change	Description	Release Date
Public Release	This is the first release of the *Amazon Elastic MapReduce Getting Started Guide*.	In this release.