

---

# **Amazon Elastic MapReduce**

**Getting Started Guide**

**API Version 2009-03-31**



# Amazon Elastic MapReduce: Getting Started Guide

Copyright © 2011 Amazon Web Services LLC or its affiliates. All rights reserved.

## Table of Contents

Get Started with Amazon Elastic MapReduce .....	1
Sign Up for Elastic MapReduce .....	2
Job Flow Essentials .....	9
Create a Streaming Job Flow .....	16
Create a Job Flow Using Hive .....	19
Restore Environment .....	26
Where Do I Go from Here? .....	28
Please Provide Feedback .....	32
About This Guide .....	33

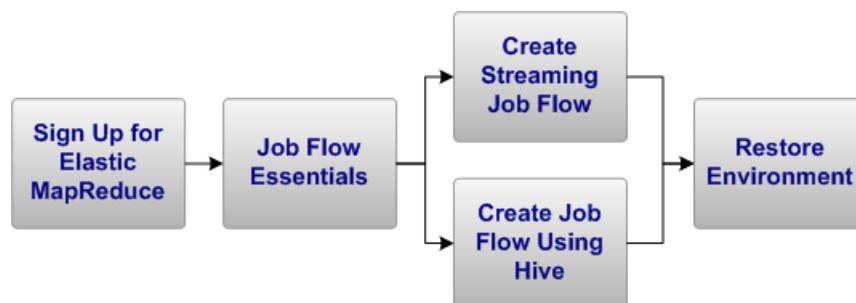
# Get Started with Amazon Elastic MapReduce

---

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

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

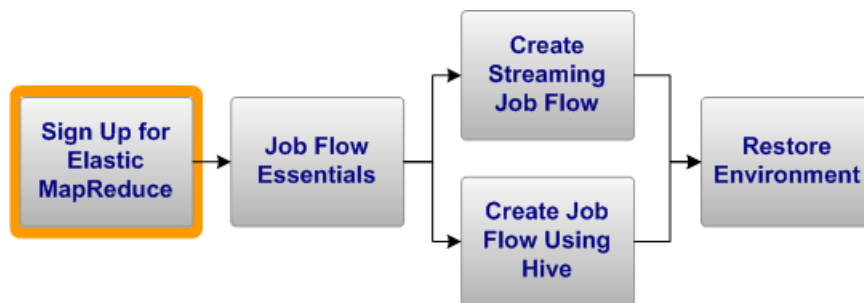
You can get started with Amazon Elastic MapReduce by following the tasks shown in the following diagram.



This guide walks you through launching and managing job flows. To start using Elastic MapReduce for the first time, move on to [Sign Up for Elastic MapReduce \(p. 2\)](#).

# Sign Up for Elastic MapReduce

---



---

## Topics

- [How to Get an Elastic MapReduce Account \(p. 2\)](#)
- [Install the Elastic MapReduce Command Line Interface \(p. 3\)](#)

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

## How to Get an Elastic MapReduce Account

This section explains how to sign up for an Elastic MapReduce account. This process creates an Amazon Web Service (AWS) account that gives you access to all Amazon Web Services, resources, forums, support, and usage reports. Signing up for Elastic MapReduce also automatically signs you up for Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3), which are closely integrated with Elastic MapReduce. You are not charged for any of the services unless you use them.

### To sign up for an Elastic MapReduce account

1. Go to <http://aws.amazon.com/elasticmapreduce/> and click **Sign Up for Elastic MapReduce**.
2. Enter your email address in the **My email address is:** field.
3. Log in to your AWS account:

- If you already have an AWS account, select **I am a returning user and my password is**, enter your password, and click **Sign in using our secure server**. Proceed to Step 4. below.
  - If you do not have an Amazon account, select **I am a new user** and click **Sign in using our secure server**. Follow the instructions to create an AWS account.
4. Review the information for **Sign Up For Amazon Elastic MapReduce**. If you accept the terms and conditions, click **Complete Sign Up** and follow the instructions on the subsequent pages.



#### Note

Part of the sign-up procedure for Amazon Elastic MapReduce includes receiving a telephone call and entering a PIN using the telephone keypad.

## Install the Elastic MapReduce Command Line Interface

### Topics

- [Installing Ruby](#) (p. 3)
- [Installing the Command Line Interface](#) (p. 4)
- [Configuring Credentials](#) (p. 5)
- [SSH Setup and Configuration](#) (p. 8)

You can create job flows consisting of multiple steps using the Elastic MapReduce command line interface (CLI). The Elastic MapReduce tab supports creating only single-step job flows. This document primarily describes how to manage job flows with the Elastic MapReduce CLI. Details on how to use the AWS management console, the Elastic MapReduce tab, and the Elastic MapReduce API are available in the [Amazon Elastic MapReduce Developer Guide](#) and the [Amazon Elastic MapReduce API Reference](#).

## Installing Ruby

The Elastic MapReduce command line interface requires Ruby 1.8. After you have installed Ruby, unzip `elastic-mapreduce-ruby.zip` into a directory, and the Elastic MapReduce CLI is ready to use.

### To install Ruby

1. Download and install Ruby 1.8:
  - 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 from [http://rubyforge.org/frs/?group\\_id=167&release\\_id=28426](http://rubyforge.org/frs/?group_id=167&release_id=28426). Ensure the Ruby directory is in your `PATH`.
2. Verify that Ruby is running by typing the following at the command prompt:

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

```
$ ruby -v
```

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

```
C:\ruby>ruby -v
```

The Ruby version is shown, confirming you installed Ruby.

## Installing the Command Line Interface

### To download the Elastic MapReduce CLI

1. Create a local directory for the CLI in the Ruby directory:

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

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

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

```
C:\ruby>mkdir elastic-mapreduce-cli
```

2. Download the Elastic MapReduce 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 Elastic MapReduce CLI

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

- Linux and UNIX 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.

## Configuring Credentials

The Elastic MapReduce credentials file can provide information required for many commands. It is convenient to store command parameters in the file to save you from the trouble of repeatedly entering the information.

Your credentials are used to calculate the signature value for every request you make. Elastic MapReduce 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 your `elastic-mapreduce-cli` directory.
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-1, or eu-west-1]"
}
```

Note the name of the Region. You will 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 **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.

- To display your Secret Access Key, click **Show** in the **Your Secret Access Key** area, as shown in the following figure.

The screenshot shows the AWS Management Console interface. At the top, there's the Amazon Web Services logo and navigation links. Below that, a navigation bar includes 'AWS', 'Products', 'Developers', 'Community', 'Support', and 'Account'. The 'Account' section is expanded, showing 'Security Credentials' as the active sub-section. The main content area is titled 'Security Credentials' and includes a welcome message for 'Test User'. It explains that AWS services are secured and lists three types of credentials: Access Credentials, Sign-In Credentials, and Account Identifiers. A link 'Find out which AWS Security Credentials you need' is provided. Below this, the 'Access Credentials' section is highlighted, with a sub-section for 'Access Keys'. This section contains a table of 'Your Access Keys' with columns for 'Created', 'Access Key ID', 'Secret Access Key', and 'Status'. A 'Show' button is next to the 'Secret Access Key' column, which is circled in red. A tooltip is visible over the 'Show' button, also with 'Secret Access Key' circled in red. The tooltip shows a masked secret access key.

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

- Go to the **Amazon EC2** tab of the AWS management console. If you are not logged in to AWS, enter your AWS account credentials when prompted.
- From the **EC2 Dashboard**, select the **Region** you used in your credentials.json file, then click **Key Pair**.



#### Note

The Region *Asia Pacific* is not supported by Elastic MapReduce.

- On the **Key Pairs** page, click **Create Key Pair**.
- Enter a name for your key pair, such as, `mykeypair`.
- Click **Create**.
- 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.

## Amazon S3 Bucket

The `log-uri` parameter specifies a location in Amazon S3 for the Elastic MapReduce 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. Go to the **Amazon S3** tab at <https://console.aws.amazon.com/s3/home>. If you are not logged in to AWS, enter your AWS Account credentials when prompted.
2. Click **Create Bucket**.  
The **Create a Bucket** dialog box opens.
3. Enter a bucket name, such as `mylog-uri`.  
This name should be globally unique, and cannot be the same name used by another bucket.



#### Note

For details on valid bucket names, go to <http://docs.amazonwebservices.com/AmazonS3/latest/dev/BucketRestrictions.html>.

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

If your Elastic MapReduce Region is...	Select the Amazon S3 Region...
us-east-1	US Standard
us-west-1	Northern California
eu-west-1	Ireland

5. Click **Create**.



#### Note

If you enable logging in the **Create a Bucket** wizard, it enables only bucket access logs, not Elastic MapReduce 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. If not already there, go to the Amazon S3 tab at <https://console.aws.amazon.com/s3/home>. If you are not logged in to AWS, enter your AWS account credentials when prompted.
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 Elastic MapReduce to upload your logs and results.

## SSH Setup and Configuration

Configure your SSH credentials for use with either ssh or PuTTY.

### 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. 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.

Windows users need to install PuTTY to connect remotely to the master node.

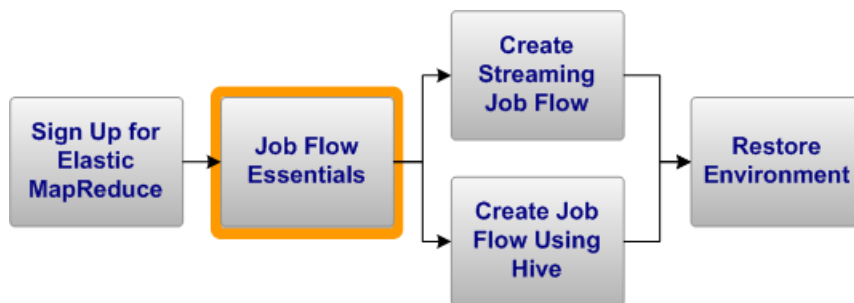
### To download PuTTY

- Windows users only, download PuTTY to your computer from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. You use PuTTY to connect via ssh to your master node as part of the sample job flow using Hive.

Now that you have signed up for Amazon Elastic MapReduce, installed the Amazon Elastic MapReduce CLI, and configured your settings, move on to [Job Flow Essentials \(p. 9\)](#).

# Job Flow Essentials

---



---

## Topics

- [Creating a Job Flow \(p. 9\)](#)
- [Managing a Job Flow \(p. 10\)](#)
- [Terminate a Job Flow \(p. 14\)](#)

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

Elastic MapReduce takes care of provisioning an Amazon EC2 cluster, terminating it, moving the data between it and Amazon S3, and optimizing Hadoop. Elastic MapReduce 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 Elastic MapReduce 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

- Enter the following commands from the command-line prompt:
  - Linux and UNIX users:

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

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>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 Elastic MapReduce Commands

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

#### To list all Elastic MapReduce commands

- Enter the following commands from the command-line prompt:
  - Linux and UNIX users:

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

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce --help
```

For more information on each of the Elastic MapReduce 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

- Enter the following commands from the command-line prompt:

- Linux and UNIX users:

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

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>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

- Enter the following commands from the command-line prompt:

- Linux and UNIX users:

```
$ ./elastic-mapreduce --describe --jobflow [JobFlowID]
```

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce --describe --jobflow  
[JobFlowID]
```

The response looks similar to the following:

```
{
  "JobFlows": [
    {
      "Name": "Development Job Flow (requires manual termination)",
      "LogUri": "s3n:\\\\YourBucket\\FileName\\",
      "ExecutionStatusDetail": {
        "StartDateTime": null,
        "EndDateTime": null,
        "LastStateChangeReason": "Starting instances",
        "CreationDateTime": DateTimeStamp,
        "State": "STARTING",
        "ReadyDateTime": null
      },
      "Steps": [],
      "Instances": {
        "MasterInstanceId": null,
        "Ec2KeyName": "KeyName",
        "NormalizedInstanceHours": 0,
        "InstanceCount": 5,
        "Placement": {
          "AvailabilityZone": "us-east-1a"
        },
        "SlaveInstanceType": "m1.small",
        "HadoopVersion": "0.20",
        "MasterPublicDnsName": null,
        "KeepJobFlowAliveWhenNoSteps": true,
        "InstanceGroups": [
          {
            "StartDateTime": null,
            "SpotPrice": null,
            "Name": "Master Instance Group",
            "InstanceRole": "MASTER",
            "EndDateTime": null,
            "LastStateChangeReason": "",
            "CreationDateTime": DateTimeStamp,
            "LaunchGroup": null,
            "InstanceGroupId": "InstanceGroupID",
            "State": "PROVISIONING",
            "Market": "ON_DEMAND",
            "ReadyDateTime": null,
            "InstanceType": "m1.small",
            "InstanceRunningCount": 0,
            "InstanceRequestCount": 1
          },
          {
            "StartDateTime": null,
            "SpotPrice": null,
            "Name": "Task Instance Group",
            "InstanceRole": "TASK",
            "EndDateTime": null,
            "LastStateChangeReason": "",
            "CreationDateTime": DateTimeStamp,
            "LaunchGroup": null,
            "InstanceGroupId": "InstanceGroupID",
            "State": "PROVISIONING",
            "Market": "ON_DEMAND",
            "ReadyDateTime": null,
            "InstanceType": "m1.small",
```

```
        "InstanceRunningCount": 0,  
        "InstanceRequestCount": 2  
    },  
    {  
        "StartDateTime": null,  
        "SpotPrice": null,  
        "Name": "Core Instance Group",  
        "InstanceRole": "CORE",  
        "EndDateTime": null,  
        "LastStateChangeReason": "",  
        "CreationDateTime": DateTimeStamp,  
        "LaunchGroup": null,  
        "InstanceGroupId": "InstanceGroupID",  
        "State": "PROVISIONING",  
        "Market": "ON_DEMAND",  
        "ReadyDateTime": null,  
        "InstanceType": "m1.small",  
        "InstanceRunningCount": 0,  
        "InstanceRequestCount": 2  
    }  
],  
    "MasterInstanceType": "m1.small"  
},  
    "BootstrapActions": [],  
    "JobFlowId": "JobFlowID"  
}  
]  
}
```

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 Elastic MapReduce 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 \(p. 5\)](#) step.

You access Elastic MapReduce log files either by using the Elastic MapReduce tab in the AWS Management Console or by viewing them directly from the Amazon S3 tab.



### 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 using default parameter values to a job flow

- Enter the following commands from the command-line prompt:
  - Linux and UNIX users:

```
$ ./elastic-mapreduce -j JobFlowID --stream
```

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce -j JobFlowID --stream
```

The `--stream` command adds a streaming step using default parameters. in the AWS Management 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 from the Elastic MapReduce tab from either the CLI or the AWS Management Console.

### To view a job flow from the AWS management console

1. Go to the Elastic MapReduce tab at <https://console.aws.amazon.com/elasticmapreduce/home>. If you are not logged in to AWS, enter your AWS account credentials when prompted.
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

- Enter the following commands from the command-line prompt:
  - Linux and UNIX users:

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

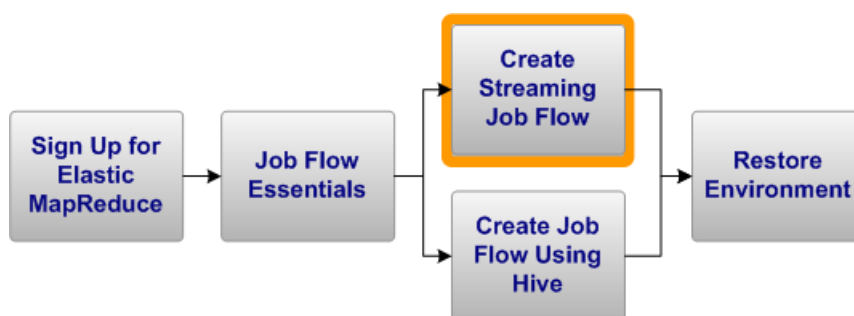
- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce --terminate JobFlowID
```

Congratulations! You have successfully created and terminated an Elastic MapReduce 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. 16).

# 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):
    line = sys.stdin.readline()
    pattern = re.compile("[a-zA-Z][a-zA-Z0-9]*")
    try:
        while line:
            for word in pattern.findall(line):
                print "LongValueSum:" + word.lower() + "\t" + "1"
            line = sys.stdin.readline()
    except "end of file":
        return None
    if __name__ == "__main__":
        main(sys.argv)
```

To run the Hadoop streaming job with Amazon Elastic MapReduce, 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:

- Linux and UNIX 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://my-  
bucket] \  
  --reducer aggregate
```

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby 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://my-  
bucket] \  
  --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 Elastic MapReduce tab in the AWS management console.

### To view the streaming job flow

1. Go to the Elastic MapReduce tab at <https://console.aws.amazon.com/elasticmapreduce/home>. If you are not logged in to AWS, enter your AWS account credentials when prompted.
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 Elastic MapReduce 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 Elastic MapReduce log files, click **Close**.

You can find additional Elastic MapReduce 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](#).



### Tip

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.

### To view job flow results

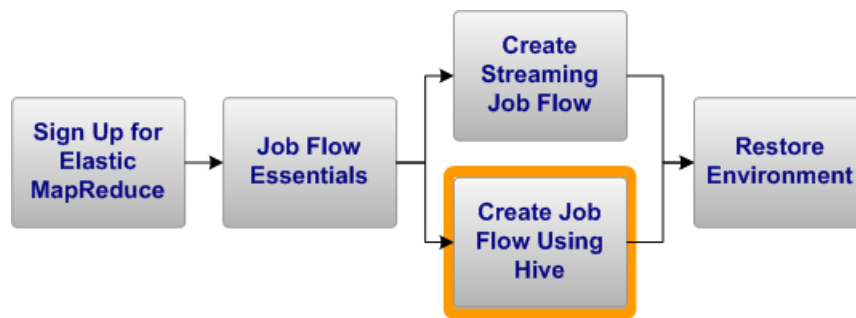
1. Go to the Amazon S3 tab at <https://console.aws.amazon.com/s3/home>. If you are not logged in to AWS, enter your AWS account credentials when prompted.
2. Navigate to the Amazon S3 bucket you referenced in `--output`.

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 \(p. 19\)](#).

# Create a Job Flow Using Hive

---



## Topics

- [Create a Hive Script \(p. 19\)](#)
- [Launch a Job Flow Using Hive \(p. 21\)](#)

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 Elastic MapReduce*  
<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.

```
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 Elastic MapReduce 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

- Enter the following commands from the command-line prompt:
  - Linux and UNIX users:

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

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce --create --alive --
name "Hive Job Flow" --hive-interactive
```

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 Elastic MapReduce tab in the AWS management console.

### To list all active job flows using the CLI

- Enter the following commands from the command-line prompt:

- Linux and UNIX users:

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

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>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. Enter the following commands from the command-line prompt:

- Linux and UNIX users:

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

Use the *job flow ID* of the sample job flow.

- Windows users:

- a. Start PuTTY.

- 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`.

## Amazon Elastic MapReduce Getting Started Guide Launch a Job Flow Using Hive

- 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 Elastic MapReduce 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.

```
hadoop@domU-12-31-39-07-D2-14:~$ 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://my-bucket/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

- Enter the following commands from the command-line prompt:
  - Linux and UNIX users:

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

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce --terminate JobFlowID
```

### To view the results of your job flow

1. Go to the Amazon S3 tab at <https://console.aws.amazon.com/s3/home>. If you are not logged in to AWS, enter your AWS account credentials when prompted.
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.

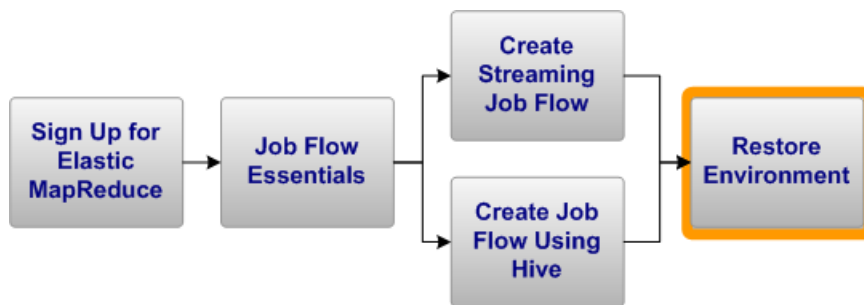
You can find additional Elastic MapReduce log files in the Amazon S3 bucket you specified in your `credentials.json` file.

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 \(p. 26\)](#).

# Restore Environment

---



You have completed the Elastic MapReduce 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 Elastic MapReduce and Amazon S3 services.

## Stop Elastic MapReduce Job Flows

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

### To list all active job flows

- Enter the following commands from the command-line prompt:
  - Linux and UNIX users:

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

- Windows users:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce --list --active
```

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

### To terminate a job flow

- 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:

```
C:\ruby\elastic-mapreduce-cli>ruby elastic-mapreduce --terminate [job flow ID]
```

Terminating all job flows will remove all associated Amazon EC2 instances.

## Remove Log Files

By specifying the *log-uri* as part of the [Configuring Credentials \(p. 5\)](#) step, all of your job flows generated Elastic MapReduce logs and saved them to Amazon S3.

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

### To delete a file on Amazon S3

1. Go to the Amazon S3 tab at <https://console.aws.amazon.com/s3/home>. If you are not logged in to AWS, enter your AWS account credentials when prompted.
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 successfully launched, connected to, and terminated a job flow. For more information about Amazon Elastic MapReduce and how to continue, see [Where Do I Go from Here? \(p. 28\)](#).

# Where Do I Go from Here?

---

## Topics

- [Other Ways to Access Elastic MapReduce](#) (p. 28)
- [Learn More About Elastic MapReduce](#) (p. 29)
- [Learn More About Hadoop](#) (p. 30)
- [Elastic MapReduce Resources](#) (p. 30)

Amazon Elastic MapReduce 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 Elastic MapReduce.

## Other Ways to Access Elastic MapReduce

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

### Continue Using the Command Line Interface

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

### Use the Console

The Elastic MapReduce administration tab in the AWS Management Console includes many functions besides just monitoring debug output. To learn more about how to use Elastic MapReduce through the management console, go to the [Amazon Elastic MapReduce Developer Guide](#). The console also has help to assist you.

### Code Directly to the Web Service API

If you want to write code directly to the Elastic MapReduce Query API, go to the [Amazon Elastic MapReduce Developer Guide](#). The guide describes how to create and authenticate API requests, and

how to use Elastic MapReduce through the APIs. For a complete description of all the API actions, go to the [Amazon Elastic MapReduce API Reference](#).

## Learn More About Elastic MapReduce

This section lists additional features in Elastic MapReduce and tells you where to find more information. You can also find additional information about Elastic MapReduce in the [Elastic MapReduce 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. For more information on using streaming job flows with Elastic MapReduce consider the following tutorial:

- Tutorial: Finding Similar Items with Amazon Elastic MapReduce, 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. For more information on using Hive with Elastic MapReduce consider the following:

- Tutorial: Contextual Advertising using Apache Hive and Amazon Elastic MapReduce with High Performance Computing instances <http://aws.amazon.com/articles/2855>
- Video: Getting started with Hive on Amazon Elastic MapReduce <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 Elastic MapReduce 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 AWS Management Console or the Elastic MapReduce 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 Elastic MapReduce Job Flow <http://aws.amazon.com/articles/3938>

## Hadoop Debugging

In addition to Elastic MapReduce logging, you also have the option to generate detailed Hadoop logs. Hadoop logging must be enabled when a job flow is created and you must sign up for Amazon 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/>.

## Elastic MapReduce Resources

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

**Amazon Elastic MapReduce Getting Started Guide**  
**Elastic MapReduce Resources**

---

Resource	Description
<a href="#">Amazon Elastic MapReduce Getting Started Guide</a>	This document. Provides a quick tutorial of the service based on a simple use case. Examples and instructions are included.
<a href="#">Amazon Elastic MapReduce Developer Guide</a>	Provides conceptual information about Elastic MapReduce and describes how to use Elastic MapReduce features.
<a href="#">Amazon Elastic MapReduce API Reference</a>	Contains a technical description of all Elastic MapReduce APIs.
<a href="#">Amazon Elastic MapReduce Quick Reference Card</a>	Describes all of the command line parameters and their options.
<a href="#">Elastic MapReduce Technical FAQ</a>	Covers the top questions developers have asked about this product.
<a href="#">Elastic MapReduce Release Notes</a>	Gives a high-level overview of the current release, and notes any new features, corrections, and known issues.
<a href="#">AWS Developer Resource Center</a>	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
<a href="#">AWS Management Console</a>	Enables you to perform most of the functions of Elastic MapReduce and other AWS products without programming.
<a href="#">Discussion Forums</a>	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
<a href="#">AWS Support Center</a>	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).
<a href="#">AWS Premium Support Information</a>	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.
<a href="#">Elastic MapReduce Product Information</a>	The primary web page for information about Elastic MapReduce.
Form for questions related to your AWS account: <a href="#">Contact Us</a>	This form is <i>only</i> for account questions. For technical questions, use the Discussion Forums.
<a href="#">Conditions of Use</a>	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 by completing our [Getting Started Survey](#).

Thank you.

# About This Guide

---

This is the *Amazon Elastic MapReduce Getting Started Guide*. It was last updated on January 16, 2011.