AWS CloudFormation

Getting Started Guide

API Version 2010-05-15

AWS CloudFormation: Getting Started Guide

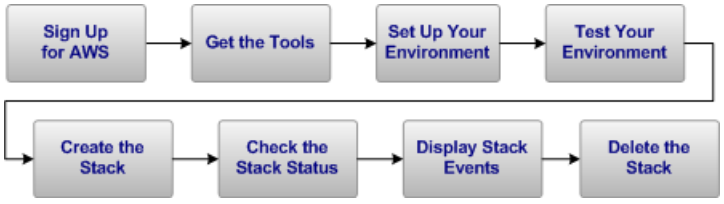
Table of Contents

Get Started with AWS CloudFormation

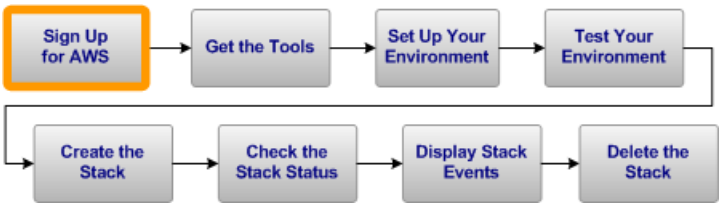
AWS CloudFormation allows you to describe your AWS infrastructure requirements in a *template*—a simple text file, which you use to create a *stack* (a collection of AWS resources you want to manage as a group). You use the template to define all the AWS resources you want in your stack. This can include Amazon Elastic Compute Cloud instances, Amazon Relational Database Service database instances, load balancers, and Amazon Simple Queue Service queues. In addition, AWS CloudFormation allows you to configure Amazon CloudWatch alarms and Auto Scaling triggers, enabling you to deploy and configure a scalable application in a single operation that conforms to your operational procedures.

AWS CloudFormation creates stacks from templates atomically, honoring resource dependencies and dealing with transient failures. It ensures that either all resources of the stack are created, or if there is an issue creating the stack, the stack is fully cleaned up. You don't have to delete any partial stacks or orphaned AWS resources.

This is the *AWS CloudFormation Getting Started Guide*. This guide walks you through launching and connecting to your first AWS CloudFormation application.



Sign Up for Amazon Web Services



Signing up for AWS CloudFormation also automatically signs you up for other AWS products you need, such as Amazon EC2, Amazon Relational Database Service and Amazon Simple Notification Service. You're not charged for any services unless you use them.

Note

AWS CloudFormation is a free service. However, you are changed for the AWS resources you include in your stacks at the current rates for each. For more information about AWS pricing, go to the detail page for each product on http://aws.amazon.com.

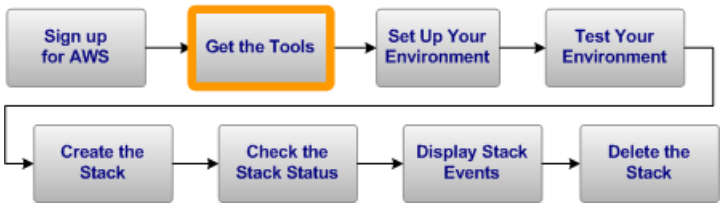
To sign up for AWS CloudFormation

1. Go to http://aws.amazon.com/cloudformation and click **Sign Up for AWS CloudFormation**.
2. Follow the on-screen instructions.

If you don't already have an AWS account, you'll be prompted to create one when you sign up for AWS CloudFormation.

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

Get the Tools



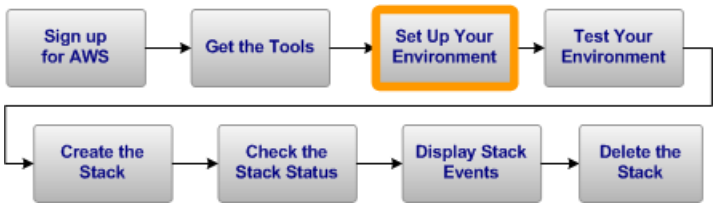
To control your AWS CloudFormation stacks, you will need to download the command line tools. You also need to get the sample template to do the tasks in this guide.

To get the command line tools

1. Download the command line tools from https://s3.amazonaws.com/cloudformation-cli/AWSCloudFormation-cli.zip. Extract them to a convenient location on your workstation.
2. If you prefer, you can download the WordPress sample template from https://s3.amazonaws.com/cloudformation-templates-us-east-1/WordPress-1.0.0.template and save it to a convenient location.

   You don't need to download it unless you want to inspect it. You will use the template URL later in this guide.

Set Up Your Environment



You need to configure your development environment to use the command line tools. When you set your environment variables, use the declaration method appropriate for your operating system or shell.

To configure your development environment

1. Create an environment variable `AWS_CLOUDFORMATION_HOME` specifying the fully-qualified pathname to the extracted archive (for example, `AWS_CLOUDFORMATION_HOME=/home/user/tools/aws/cloudformation/AWSCloudFormation-1.0`) as its value.
2. Add `${AWS_CLOUDFORMATION_HOME}/bin` to your *$PATH* environment variable (in Windows, add `%AWS_CLOUDFORMATION_HOME%\bin` to your *%PATH%* environment variable).
3. Navigate to the `credential-file-path.template` file in the root folder of the extracted archive, open it and add your AWS credentials to the appropriate lines in the file.

   ```
   AWSAccessKeyId=your access key
   AWSSecretKey=your secret key
   ```

   Save and close the file.

   Tip

   You can get your AWS Account's credentials by going to the AWS Security Credentials page. After you sign in, you can find the access keys located in the **Access Credentials** section of the page.

4. Limit the file permissions to only the file owner (e.g., use `chmod 600` on the file if you're using Linux/UNIX).

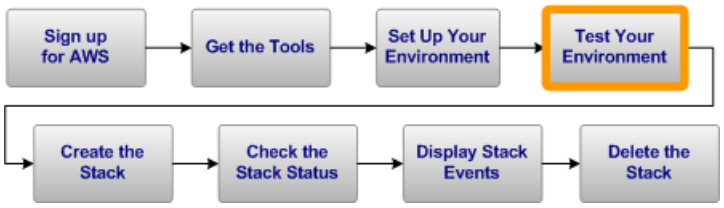
Caution

Your AWS Account's Secret Access Key is a secret which only you and AWS should know. It is important to keep it confidential to protect your AWS Account. Store it securely in a safe place. Never include it in your requests to AWS, and never e-mail it to anyone. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your AWS Account's Secret Access Key.

5. Create an environment variable `AWS_CREDENTIAL_FILE` (specifying as its value the fully-qualified filename you just edited).
6. Verify that your *`$JAVA_HOME`* environment variable is set (*`%JAVA_HOME%`* on Windows), and that your *`$PATH`* environment variable includes `$JAVA_HOME/bin` (*`%PATH%`* and `%JAVA_HOME%\bin` on Windows).

   All AWS command line tools require Java 1.5 or newer.

Test Your Environment



Before you continue with this sample, you should test your tools and validate the example template.

To test your tools

1. Test the environment by using the `cfn-cmd` command to display the basic tool help text:

   ```
   cfn-cmd
   ```

   The output should resemble the following example.

   ```
   Command Name                  Description
   ------------                  -----------

   cfn-create-stack              Create a new stack.
   cfn-delete-stack              Delete a stack.
   cfn-describe-stack-events     Describe events for one or more stacks.
   cfn-describe-stack-resources  Describe the resources for a stack.
   cfn-describe-stacks           Return information about one or more stacks.
   cfn-get-template              Get the template used to create a stack.
   cfn-validate-template         Validate a template and return parameters.
   help
   version                       Prints the version of the CLI tool and the
   API.

         For help on a specific command, type '<commandname> --help'
   ```

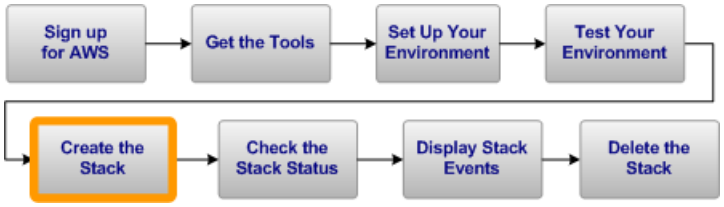
2. Test that you can reach the AWS CloudFormation service with the `cfn-validate-template` command and the WordPress template you downloaded. The command and response should resemble the following.

Note

You need to use a bucket in the same region as the region you are using. The default region is us-east-1.

```
cfn-validate-template --template-url https://s3.amazonaws.com/cloudformation-
templates-us-east-1/WordPress-1.0.0.template --headers

PARAMETERS  PARAMETER_NAME   DEFAULT_VALUE      NOECHO  DESCRIPTION
PARAMETERS  WordPressUser    admin            true    The WordPress database
 admin account username
PARAMETERS  WordPressDBPort  8443               false   TCP/IP port for the
 WordPress database
PARAMETERS  OperatorEmail    nobody@amazon.com  false   Email address to
notify if there are any operational issues
PARAMETERS  WordPressPwd     admin            true    The WordPress database
 admin account password
PARAMETERS  WordPressDBName  wordpress        false   The WordPress database
 name
PARAMETERS  GroupSize        1                  false   The initial number
 of EC2 instances for the WordPress web server
PARAMETERS  WebServerPort    8888               false   TCP/IP port for the
 WordPress web server
PARAMETERS  InstanceType     m1.small           false   The type of EC2
instances used for the WordPress web server
PARAMETERS  KeyName                             false   Name of an existing
 EC2 KeyPair to enable SSH access into the WordPress web server
```

Create the Stack



You create your stack based on the *WordPress-1.0.0* file you downloaded in the previous task. The template contains several AWS resources including a LoadBalancer, an Amazon Relational Database Service instance, and an Auto Scaling group.

To create the stack

1. Create a complete WordPress stack using the `cfn-create-stack` command.

```
PROMPT> cfn-create-stack MyWPTestStack --template-url
https://s3.amazonaws.com/cloudformation-templates-us-east-1/WordPress-
1.0.0.template --parameters="KeyName=EC2-KeyPair-Name"

arn:aws:cfn:us-east-1:165024647323:stack/MyWPTestStack/d3c07370-2391-11e0-
b93a-508be050d086
```

   The return value is the unique stack identifier. Though you get the stack identifier right away, it takes about ten minutes to create all its member resources.

   Note

   The template is configured to take the name of a valid EC2 key name, which will enable you to access the WordPress web server via SSH. If you supply a valid key name, the stack will be created. If you don't supply a valid key name, the stack will be rolled back. For information on getting a key pair, see *SSH Key Pair* at http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/using-credentials.html.

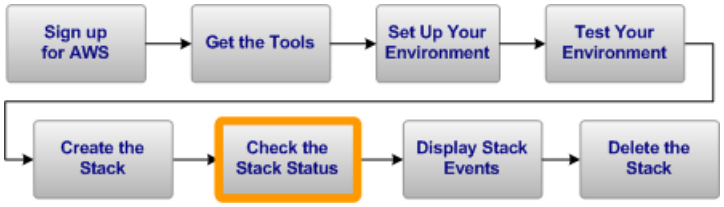
2. Use the `cfn-describe-stack-events` command to monitor your WordPress stack as it creates its components.

```
cfn-describe-stack-events MyWPTestStack --headers

PROMPT> cfn-describe-stack-events MyWPTestStack --headers

STACK_EVENT  STACK_NAME     LOGICAL_ID           RESOURCE_TYPE
                EVENT_TIME            RESOURCE_STATUS
RESOURCE_STATUS_REASON

STACK_EVENT  MyWPTestStack  ElasticLoadBalancer
AWS::ElasticLoadBalancing::LoadBalancer  2011-02-22T17:54:13Z  CREATE_COMPLETE
STACK_EVENT  MyWPTestStack  ElasticLoadBalancer
AWS::ElasticLoadBalancing::LoadBalancer  2011-02-22T17:54:09Z
CREATE_IN_PROGRESS
STACK_EVENT  MyWPTestStack  EC2SecurityGroup     AWS::EC2::SecurityGroup
                2011-02-22T17:54:09Z  CREATE_COMPLETE
STACK_EVENT  MyWPTestStack  EC2SecurityGroup     AWS::EC2::SecurityGroup
                2011-02-22T17:54:05Z  CREATE_IN_PROGRESS
STACK_EVENT  MyWPTestStack  MyWPTestStack        AWS::CloudFormation::Stack
              2011-02-22T17:53:58Z  CREATE_IN_PROGRESS    User Initiated
```

The example shows that the most recent event was the creation of an RDS security group.

The stack isn't fully created yet. Next, you use `cfn-describe-stacks` to check the stack status.

Check the Stack Status



You can use the `cfn-describe-stacks` command to discover when your stack has successfully completed the creation process.

To check your stack status

1. Use the `cfn-describe-stacks` command, specifying the stack name.

   In the example, the stack name is *`MyWPTestStack`*.

   ```
   PROMPT> cfn-describe-stacks MyWPTestStack

   STACK  MyWPTestStack  CREATE_COMPLETE  WordPress is web software you can
   use to create a beautiful website or blog.
   URL=http://WordPress-ElasticL-1IDEWGIID2CF8-362665508.us-east-
   1.elb.amazonaws.com/wp-admin/install.php  2011-02-22T18:28:06Z
   ```

   If you do not specify the stack name, AWS CloudFormation reports the status of all your running stacks.

   The example shows the status as *`CREATE_COMPLETE`*, which means that the stack has finished loading.

   The URL for your new stack is available as an output from the stack. In this case, it's `http://WordPress-ElasticL-1IDEWGIID2CF8-362665508.us-east-1.elb.amazonaws.com/wp-admin/install.php`.

2. After your stack status is *`CREATE_COMPLETE`*, use the `cfn-describe-stack-resources` command to view the resources that have been provisioned as part of the stack.

   ```
   PROMPT> cfn-describe-stack-resources --stack-name MyWPTestStack --headers
   ```

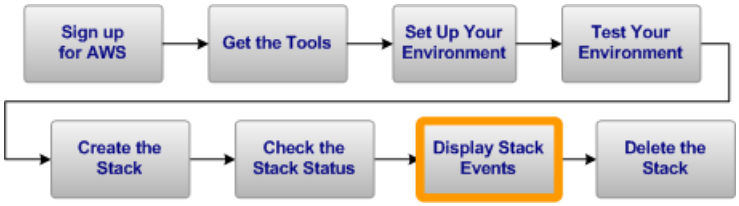
   The return value contains the resource types, their IDs, and information about their status.

```
STACK_RESOURCE  LOGICAL_ID                  PHYSICAL_ID
                   TYPE                                          CREATED_TIME
        STATUS
STACK_RESOURCE  EC2SecurityGroup            MyWPTestStack-EC2SecurityGroup-
MCQ4H1YSB08E      AWS::EC2::SecurityGroup                   2011-02-
22T18:28:15Z  CREATE_COMPLETE
STACK_RESOURCE  ElasticLoadBalancer        WordPress-ElasticL-1IDEWGIID2CF8
               AWS::ElasticLoadBalancing::LoadBalancer  2011-02-22T18:28:18Z
  CREATE_COMPLETE

            ... more stack resources ...
```

Each resource is tagged with its stack name (`MyWPTestStack`). This makes it easy for you to identify which resources were created as part of a stack when you use other AWS tools.

For example, if you had the Amazon RDS tools installed, you could use the command `rds-describe-db-instances` command to report all your active instances. This would return a report on all DB instances, whether created by AWS CloudFormation or not. The output from that command includes stack name tags. You can therefore filter on *`MyWPTestStack`* to identify the DB instances that AWS CloudFormation has created.

Display Stack Events



The `cfn-describe-stack-events` command enables you to see the progress a stack makes as it is created (or deleted).

To check stack events

- Use the `cfn-describe-stack-events` command followed by the name of the stack.
  The following example shows the most recent stack events. AWS CloudFormation reports the newest events first.

```
PROMPT> cfn-describe-stack-events WordPressTestStack

STACK_EVENT  STACK_NAME     LOGICAL_ID      RESOURCE_TYPE EVENT_TIME
RESOURCE_STATUS        RESOURCE_STATUS_REASON
STACK_EVENT  MyWPTestStack  MyWPTestStack   AWS::CloudFormation::Stack
2011-02-22T18:41:20Z  CREATE_COMPLETE
STACK_EVENT  MyWPTestStack  CPUAlarmHigh    AWS::CloudWatch::Alarm
2011-02-22T18:41:11Z  CREATE_COMPLETE
STACK_EVENT  MyWPTestStack  CPUAlarmHigh    AWS::CloudWatch::Alarm
2011-02-22T18:41:04Z  CREATE_IN_PROGRESS

... earlier stack events ...
```

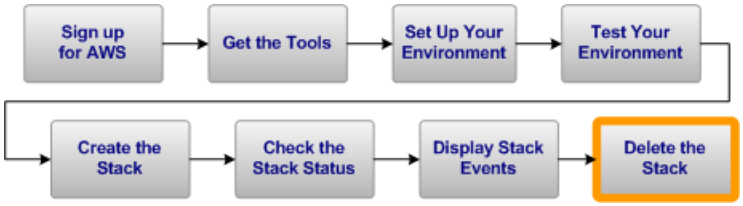
  The stack name is optional. If you do not specify a stack name, the events for all your running stacks are returned.

  Tip

  You can use `cfn-describe-stack-events` anytime. By examining the most recent events, you can track the progress of stack and resource creations and deletions.

Delete the Stack



When you've finished exploring the new stack, be sure to delete it to avoid unnecessary charges.

To delete the stack

1. Use the `cfn-delete-stack` command to delete the stack.

```
PROMPT> cfn-delete-stack MyWPTestStack

Warning: Deleting a stack will lead to deallocation of all of the stack's
resources. Are you sure you want to delete this stack? [Ny]y
```

   In the example, the user confirmed the deletion with **y**.

2. After the delete begins, use the `cfn-describe-stacks` command to track progress of the deletion.

```
PROMPT> cfn-describe-stacks MyWPTestStack

STACK  MyWPTestStack  arn:aws:cfn:us-east-
1:165024647323:stack/MyWPTestStack/d3c07370-2391-11e0-b93a-508be050d086/
2011-01-19T06:03:25Z  DELETE_IN_PROGRESS  User Initiated  false
URL=http://WordPress-WordPres-MGO8851IMZLJ-701592678.us-east-
1.elb.amazonaws.com
```

   The status shows *`DELETE_IN_PROGRESS`* while the member AWS resources are being deleted. When the status returns *`no stacks found`*, the stack is fully deleted.

Tip

You can continue to use the cfn-describe-stacks command to monitor the deletion progress.

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 AWS CloudFormation by completing our Getting Started Survey.

Thank you.

Where Do I Go From Here?

Topics

AWS CloudFormation is a rich service offering many features not covered in this guide, such as support for a number of preconfigured templates, the ability to create templates from scratch, and a user interface integrated with the AWS Console. This section provides links to additional resources, that can help you deepen your understanding and use of AWS CloudFormation.

Continue Using the Command Line Interface

This guide introduced you to basic operations using several of the command line tools. To learn more about the AWS CloudFormation tools, refer to the AWS CloudFormation User Guide.

Using the AWS Management Console

This guide introduces you to AWS CloudFormation by using the command line tools. You can also use the AWS Management Console to perform the operations listed here. With the console, you can create, list, and delete your stacks; examine the state of each stack's components, get a stack's template, and delete stacks.

For more information, sign in to the AWS Management Console, and click the AWS CloudFormation tab.

Use and Customize the Standard AWS CloudFormation Templates

In this guide you created a stack based on the WordPress stack template. We provide over a dozen other templates. They enable you to create stacks that use most of the features that AWS CloudFormation

provides. In addition, you can use these standard templates as models to create your own customized stack templates. To learn more about using and modifying templates, go to the AWS CloudFormation User Guide.

AWS CloudFormation Related Resources

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

Resource	Description
User Guide	Provides detailed information on how to use AWS CloudFormation.
API Reference	Provides detailed information on how to use the AWS CloudFormation libraries from your code.
AWS Management Console	Enables you to perform most of the functions of AWS CloudFormation without programming.
AWS CloudFormation FAQs	Frequently asked questions about using AWS CloudFormation.
AWS CloudFormation Release Notes	Gives a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
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.
AWS CloudFormation Discussion Forum	A community-based forum for developers to discuss technical questions related to AWS CloudFormation.
Product information about AWS CloudFormation	The primary web page for information about AWS CloudFormation.
For questions related to your AWS account: Contact Us	This 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.

Document History

The following table describes the important changes to the AWS CloudFormation documentation. This documentation is associated with the 2010-05-15 release of AWS CloudFormation. This guide was last updated on 28 May 2011.

Change	Description	Release Date
New Feature	AWS CloudFormation now provides the cfn-list-stacks command, which enables you to list stacks filtered by stack status. Deleted stacks can be listed for up to 90 days after they have been deleted. .	in this release
New Features	The cfn-describe-stack-resources and cfn-get-template commands now enable you to get information from stacks which have been deleted for 90 days after they have been deleted. For more information, see ??? and ???.	in this release
New Link	AWS CloudFormation endpoint information is now located in the Amazon Web Services General Reference. For more information, go to Regions and Endpoints in Amazon Web Services General Reference.	2011-03-01
Initial Release	This is the initial public release of AWS CloudFormation.	2011-02-25