Using AWS CloudFormation for Automation

**SPL-BE-100-CEUCFA-1 - Version 1.0.0**

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

This lab provides you with hands-on experience using AWS CloudFormation, Amazon Elastic Compute Cloud (Amazon EC2) and the AWS Command Line Interface (CLI). You review an incomplete CloudFormation template and are challenged to fill in the missing parameters, allowing the stack to launch an Amazon EC2 instance, and user data install and run nginx. Once the stack creates you verify that nginx is running on the Amazon EC2 instance.

OBJECTIVES

By the end of this lab, you will be able to do the following:

* Review an incomplete CloudFormation template.
* Update the CloudFormation template to create an Amazon EC2 instance.
* Configure the instance to use user data to install nginx.
* Create a CloudFormation stack with an Amazon EC2 resource.
* Test that the nginx web server runs successfully on the Amazon EC2 instance.

TECHNICAL KNOWLEDGE PREREQUISITES

* Familiarity with the AWS Cloud9 interface is helpful as you open a file to view its contents and modify a commands with the text editor.
* Experience using a terminal session is helpful.

DURATION

This lab requires *30* minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Additional information:** Where to find more information.
* **Consider:** A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand.
* **Task complete:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Review an incomplete CloudFormation template that creates an Amazon EC2 instance**

In this task, you connect to an AWS Cloud9 environment that is used as the Integrated Development Environment (IDE) to review and update the incomplete CloudFormation template that has been prebuilt for you. You identify the properties that have been defined and determine which parameters need to be updated for the template to successfully create a new CloudFormation stack and the resource(s) defined.

TASK 1.1: REVIEW THE INCOMPLETE CLOUDFORMATION TEMPLATE

Connect to the AWS Cloud9 environment provisioned as part of this lab. This is the IDE that you use to review and update the CloudFormation template as well as run AWS CLI commands to create the CloudFormation stack which creates the Amazon EC2 instance.

1. Copy the **Cloud9Environment** URL link from the **Lab information** section to the left of these instructions and paste it into a new browser tab.

The browser takes you to the AWS Cloud9 environment that you use during this lab.

1. You do not need the **Cloud9 Welcome screen** or any of the other default tabs that appear when you first launch the IDE, so choose the **X** next to each tab to close them. This section is where you can update files throughout this lab.

**Consider:** You are working in another AWS Cloud9 environment similar to other labs. The only difference is the files that you see in the file tree. If you need a refresher, take a moment to familiarize yourself with the **AWS Cloud9 IDE** interface by expanding the *Cloud9 review* section.

**Cloud9 review**

1. In the file tree, open the file named **cloudformation.template**.
2. Review the CloudFormation template and see if you can identify the specific properties and what they do.
3. Once you have an idea what the template does, review the detailed overview section below to see if your assumptions are correct.

**Detailed overview**

To make this CloudFormation template complete and functional, the missing values (represented by ???) need to be filled in.

**Task complete:** You have reviewed the incomplete CloudFormation template and identified the properties that require updating for the template to successfully create a new stack with an Amazon EC2 resource.

**Task 2: Update the incomplete CloudFormation template**

In this task, you are challenged with updating the incomplete CloudFormation template with the missing details so that it can create a new CloudFormation stack and any defined resource(s).

TASK 2.1: UPDATE THE RESOURCES SECTION OF THE CLOUDFORMATION TEMPLATE

The resources section of the template contains three properties that need to be updates. These properties are specified by ??? (three question marks).

1. The **cloudformation.template** file should already be opened, if not, open it now for editing.
2. Update the values for the following properties from the **Resources** section located to the left of these instructions:

* For *ImageId*, use the value for **AMI**.
* For *SecurityGroupIds*, use the value for **SecurityGroupID**.
* For *SubnetId*, use the value for **PublicSubnet**.

TASK 2.2: UPDATED THE OUTPUT SECTION OF THE CLOUDFORMATION TEMPLATE

There is one last update required for this CloudFormation template to be complete and functional. The output section is missing details to provide the public IP address of the instance when processed.

You can retrieve the public IP address of an Amazon EC2 instance by accessing the instance’s *return values*. This is a list of attributes that can be returned when you use either the !GetAtt or !Sub intrinsic function.

* *!GetAtt* is specifically designed to retrieve attributes from CloudFormation resources.
* *!Sub* is more versatile and can be used for string substitutions, including embedding *!GetAtt* functions within it to replace placeholders with actual resource attributes.

**Note:** When to use one over the other often comes down to the specific needs of the CloudFormation task at hand. If you’re just fetching an attribute, *!GetAtt* is straightforward. If you’re embedding that attribute in a more complex string or need string manipulation, *!Sub* becomes very handy. As you can see, in your CloudFormation template it is using the !Sub intrinsic function because it is creating a string for a proper URL address.

When the output is processed, it provides the full URL of the instances’s public IP address as *http://1.1.1.1.1* (where 1.1.1.1 is updated with the actual public IP address).

1. Review the Amazon EC2 URL for [Return values](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html#aws-properties-ec2-instance-return-values), to see the list of available attributes.

As you can see, one of the Amazon EC2 returnable attributes is the *PublicIp*.

To provide the public IP address in the outputs section, requires the use of the !Sub intrinsic function coupled with the *logical ID* of the Amazon EC2 resource, and the *returnable attribute* (PublicIp in this case) using dot notation. Dot notation provides a concise and readable way to access the properties and methods of objects across various programming languages.

**Learn more:** Refer to *A Beginners Guide to Dot Notation* in the **Additional resources** section for more information.

1. Update the *Website* output to include the *PublicIp* Amazon EC2 returnable attribute as shown below:

Value: !Sub 'http://${WebServer.PublicIp}'

1. Save the changes to your *cloudformation.template* file.

**Task complete:** You have filled in the missing property values for the Amazon EC2 resource in the incomplete CloudFormation template. You also added the missing Amazon EC2 PublicIp attribute required to retrieve the Amazon EC2 instance’s public IP address.

**Task 3: Create a CloudFormation stack and verify the user data processed successfully**

In this task, you create a CloudFormation stack using your CloudFormation template which creates an Amazon EC2 instance. Once the stack finishes, you run an AWS CLI command to retrieve the *Output* to see the nginx URL that gets created as part of the stack creation process. Then you copy and paste the URL into a browser to verify the nginx server is functional and responding to web requests.

TASK 3.1: CREATE THE CLOUDFORMATION STACK USING THE AWS CLI

Using the AWS CLI, you can create a CloudFormation stack and the associated resource(s) specified in the CloudFormation template file. To launch a CloudFormation stack using a template file via the AWS CLI, you use the *aws cloudformation create-stack* command.

Here is an example of the basic syntax:

**Example:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is an EXAMPLE ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aws cloudformation create-stack --stack-name YOUR\_STACK\_NAME --template-body file://PATH\_TO\_YOUR\_TEMPLATE-FILE

Where:

* *YOUR\_STACK\_NAME* is the name you want to give your CloudFormation stack.
* PATH\_TO\_YOUR\_TEMPLATE\_FILE is the local path to your CloudFormation template file.
  + **Note:** The *file://* prefix is important as it lets the CLI know you are providing the file path.

1. **Command:** Create the CloudFormation stack using the following AWS CLI command in the AWS Cloud9 terminal:

aws cloudformation create-stack --stack-name nginx-Web-Server --template-body file://~/environment/cloudformation.template

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"StackId": "arn:aws:cloudformation:us-west-2:111111111111:stack/nginx-Web-Server/bab88ad0-388b-11ee-aa2c-02f2933b4349"

}

You can check the status of a stack using the *aws cloudformation describe-stacks* command. You can limit the results to just the stack status using the global *–query* option.

1. **Command:** Check the status of the CloudFormation stack creation with the following command:

aws cloudformation describe-stacks --stack-name "nginx-Web-Server" --query "Stacks[0].StackStatus"

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

"CREATE\_IN\_PROGRESS"

"CREATE\_IN\_PROGRESS"

"CREATE\_IN\_PROGRESS"

"CREATE\_COMPLETE"

Initially, it shows as *CREATE\_IN\_PROGRESS*. It may take a few times running the command but eventually the status changes to *CREATE\_COMPLETE*.

**Note:** If you encounter an error when creating your stack, review the following troubleshooting steps:

**Troubleshooting steps:**

Once it has a status of *CREATE\_COMPLETE*, run another *aws cloudformation describe-stacks* command and query the Outputs section to see the output value (the URL) for the Website output key.

1. **Command:** To retrieve the URL for the nginx web server, run the following command:

aws cloudformation describe-stacks --stack-name "nginx-Web-Server" --query "Stacks[0].Outputs"

**Expected output:** Your IP address is going to be a different value than seen in the example below.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is EXAMPLE OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[

{

"OutputKey": "Website",

"OutputValue": "http://54.218.70.234",

"Description": "URL to site"

}

]

TASK 3.2: VERIFY THE USER DATA PROCESSED SUCCESSFULLY BY TESTING THE NGINX WEB SERVER

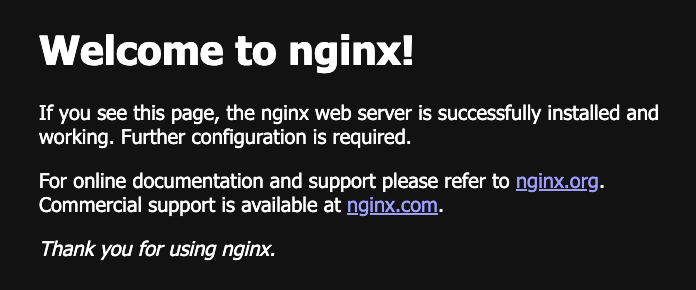
You can now open the link to the public IP address of the Amazon EC2 instance to confirm the user data script ran successfully and that the nginx web server is running and accepting web requests.

1. Hover over the **URL** value, once the curser changes to a hand perform a left mouse/pad action. From the options listed you can choose either **Open** or **Copy**.

* If you chose *Open*, it opens a new browser tab to that URL.
* If you chose **Copy**, you have to manually open a new browser tab and paste in the URL.

**Note:** Choosing *Open in Preview* does not work and result in an blank error page.

You should see a welcome message as show below:



This indicates that the user data ran and completed successfully when you created the instance using CloudFormation.

**Task complete:** You have created a CloudFormation stack using the CloudFormation template file that you updated. You retrieved the nginx URL and verified that the user data processed successfully by viewing the nginx default home page running locally on the Amazon EC2 instance you created using CloudFormation.

**Conclusion**

You have done the following:

* Reviewed an incomplete CloudFormation template.
* Updated the CloudFormation template to create an Amazon EC2 instance.
* Configured the instance to use user data to install nginx.
* Created a CloudFormation stack with an Amazon EC2 resource.
* Tested that the nginx web server ran successfully on the Amazon EC2 instance.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional Resources**

* [Dot notation](https://en.wikipedia.org/wiki/Dot_notation)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).