

Introduction to AWS CloudFormation Designer

Lab Overview

AWS CloudFormation Designer is a graphic tool for creating, viewing and modifying AWS CloudFormation templates. With Designer, you can diagram your template resources using a drag-and-drop interface, and then edit details using the integrated JSON and YAML editor. Whether you are a new or an experienced AWS CloudFormation user, AWS CloudFormation Designer can help you quickly see the interrelationship between a template's resources and easily modify templates.

In this lab you create an AWS CloudFormation stack using the AWS CloudFormation Designer. You use Designer to add resources and edit basic Parameters, Mappings, and Outputs. You use Designer to validate and launch the stack, then view the running stack. You then clean-up your resources by deleting the stack.

Topics Covered

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

- Create a simple single-instance stack using AWS CloudFormation Designer.
 - Add and edit Mappings, Resource Parameters, and Outputs.
 - Validate the stack and save your basic template.
 - Launch the stack and view the running stack and its Outputs.
 - Delete and clean up when the stack is no longer required.
-

Prerequisites

This lab assumes that you have an introductory-level knowledge of Amazon EC2 and AWS CloudFormation.

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:

- **Note:** A hint, tip, or important guidance.
- **CAUTION:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).

- **Refresh:** A time when you might need to refresh a web browser page or list to show new information.
 - **Copy edit:** A time when copying a command, script, or other text to a text editor (to edit specific variables within it) might be easier than editing directly in the command line or terminal.
-

About the Technologies

AWS CloudFormation

AWS CloudFormation enables you to create and provision AWS infrastructure deployments predictably and repeatedly. It helps you leverage AWS products such as Amazon EC2, Amazon Elastic Block Store, Amazon SNS, Elastic Load Balancing, and Auto Scaling to build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS infrastructure. AWS CloudFormation enables you to use a template file to create and delete a collection of resources together as a single unit (a stack).

AWS CloudFormation Designer

With AWS CloudFormation Designer, you can view the contents of any valid AWS CloudFormation template as a diagram. You can easily see the resources that are included in the template and how they're connected without reading JSON. If you need to view template details, you can use the integrated JSON text editor.

By using AWS CloudFormation Designer, you spend less time manually coding your templates and more time designing your AWS infrastructure. In AWS CloudFormation Designer, you drag

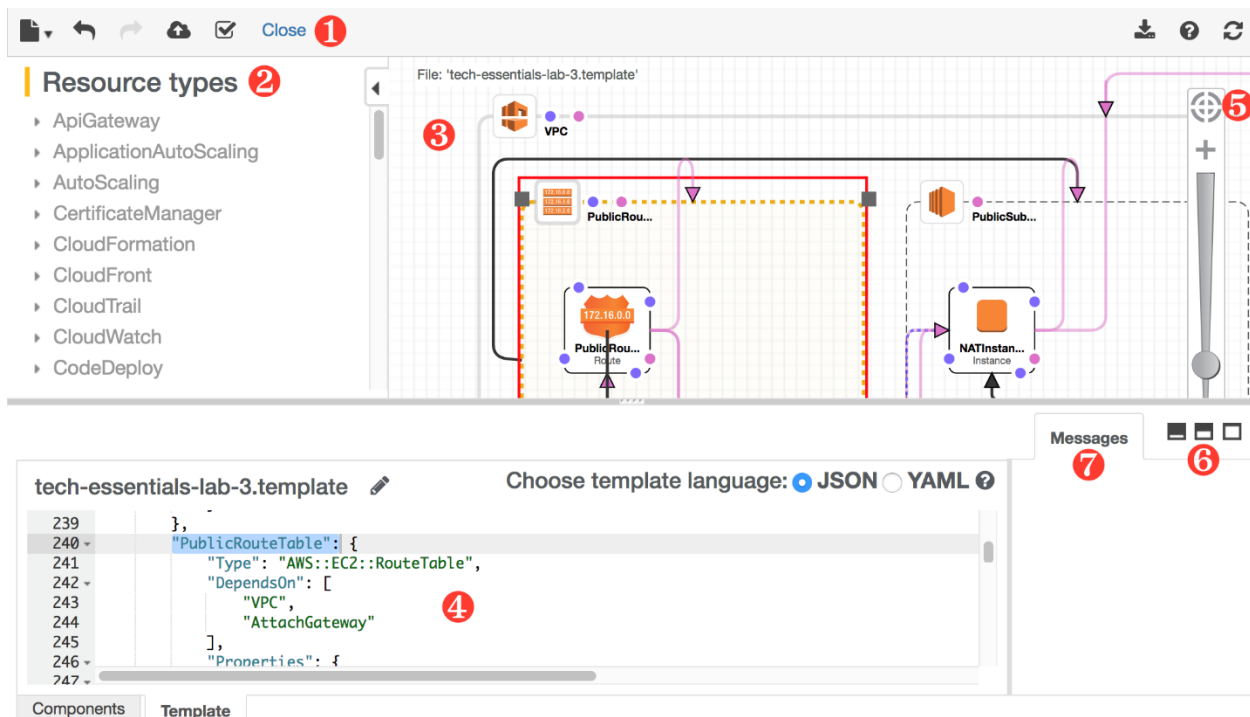
and drop new resources to add them to your template, and you drag connections between resources to establish relationships. AWS CloudFormation Designer automatically modifies the JSON. The integrated JSON text editor provides an auto-complete feature for resource property names.

The AWS CloudFormation Designer interface consists of several panes. The canvas pane shows a diagram of your template resources so that you can see them and their relationships at a glance. To add resources to your template, you drag resources from the **Resources types** pane onto the **canvas** pane. Use the **JSON editor** pane to specify template details, such as resource properties or template parameters. After you've modified the template, you can save its layout to a local file or to an S3 bucket.

AWS CloudFormation Designer cannot show or modify running *resources* in your stacks; use it for creating, modifying, and saving *templates* only.

AWS CloudFormation Designer Tool components

Here is an overview of the AWS CloudFormation Designer interface:



1. Toolbar

The toolbar provides quick access to commands for common actions, such as opening and saving templates, undoing or redoing changes, creating a stack, and validating your template.

2. Resource types pane

The Resource types pane lists all of the template resources, categorized by their AWS service name, that you can add to your template by dragging them to the canvas.

3. Canvas

The canvas pane displays your template resources as a diagram. You use it to add or remove resources, create relationships between resources and arrange their layout. The changes that you make in the canvas automatically modify the template's JSON.

4. JSON editor

In the JSON editor, you specify the details of your template, such as resource properties or template parameters. When you select an item in the canvas, the related JSON is highlighted in the editor.

5. Fit to window button

This resizes the canvas pane to fit your template's diagram.

6. Full or split screen buttons

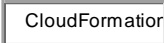
These select different views of AWS CloudFormation Designer. You can select a full-screen view of the canvas, a full-screen view of the JSON editor, or a split-screen view of the canvas and editor.

7. Messages pane

The Errors pane displays validation errors when you open, validate, or attempt to create a stack with an invalid template.

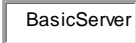
Task 1: Add an Instance Resource

In this task, you use the AWS CloudFormation Designer drag-and-drop interface to add an Amazon EC2 instance.

3. At the top of the AWS Management Console, in the search bar, search for and choose .
4. If you see the message *The redesigned AWS CloudFormation console is available now*, click **Try it out now and provide us feedback**.
5. In the left navigation pane, choose **Designer**.

If **Designer** is not visible, choose the **Navigation** icon and then select **Designer** from the list.

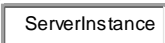
The **Parameters** tab in the lower part of the screen contains the JSON Editor.

6. Choose the pencil icon, then:
 - Change the template name to .
 - Press **Enter**

The **Resource types** pane on the left contains a list of all AWS resources that can be included in a template.

7. In the **Resource types** pane:
 - Expand **EC2**
 - Drag an **Instance** resource to blank designer canvas

The JSON template will auto-populated the JSON code for the resource in the lower pane.

8. Choose the pencil icon, then:
 - Rename the instance to .
 - Press **Enter**
9. At the top-right of the screen, choose the refresh icon.

Congratulations! You have successfully added an Instance resource using the drag-and-drop interface of the AWS CloudFormation Designer.

Task 2: Set Parameters

Parameters are input values that you specify when creating a stack. Parameters are useful for passing-in values to avoid hard-coded values in templates. For example, you do not need to hard-code a server *instance type* in the template; instead, you can use a parameter to specify the instance type when you create the stack. This allows the same template to be used to create multiple servers with different instance types.

In this task, you use the AWS CloudFormation Designer to set parameters for the EC2 instance resource.

10. Select on an open area in the AWS CloudFormation Designer canvas.

Selecting on an open area in the canvas allows you to edit template-level components.

11. In the JSON editor pane, choose the **Parameters** tab if not already selected.
12. Copy the JSON code below and paste the block into the JSON editor, **replacing everything that is currently there**.

```
{
  "Parameters": {
    "InstanceType": {
      "Description": "Server EC2 instance type",
      "Type": "String",
      "Default": "t3.micro",
      "AllowedValues": [
        "t3.micro",
        "t2.micro"
      ],
    }
  }
}
```

```
        "ConstraintDescription": "must be a valid EC2 instance type."
    }
}
}
```

Examine the code. This code adds parameters for selecting an **Instance Type**.

13. Update the canvas by choosing the refresh icon.

When you see a message in the canvas pane that the Designer tool is *out of date* and needs to be refreshed, simply use the refresh icon to update the canvas.

Congratulations! You have successfully used the AWS CloudFormation Designer to set parameters for the EC2 instance resource.

Task 3: Add Mappings

Mappings are a set of keys that are associated with a set of name-value pairs. They are useful for specifying values based on an input parameter value. In this lab, you use a mapping to select an *AMI ID* for an EC2 instance based on the *region* where the stack is deployed.

14. In the JSON editor pane, choose the **Mappings** tab.
15. Copy the JSON code below and paste the block into the JSON editor, **replacing everything that is currently there**.

```
{
  "Mappings" : {
    "AWSRegionToAMI": {
      "us-east-1": { "AMI": "ami-c58c1dd3" },
      "us-east-2": { "AMI": "ami-4191b524" },
      "us-west-1": { "AMI": "ami-7a85a01a" },
    }
  }
}
```



```
"us-west-2": { "AMI": "ami-4836a428" },
"ca-central-1": { "AMI": "ami-0bd66a6f" },
"eu-west-1": { "AMI": "ami-01ccc867" },
"eu-west-2": { "AMI": "ami-b6daced2" },
"eu-central-1": { "AMI": "ami-b968bad6" },
"sa-east-1": { "AMI": "ami-37cfad5b" },
"ap-southeast-1": { "AMI": "ami-fc5ae39f" },
"ap-southeast-2": { "AMI": "ami-162c2575" },
"ap-south-1": { "AMI": "ami-52c7b43d" },
"ap-northeast-1": { "AMI": "ami-923d12f5" },
"ap-northeast-2": { "AMI": "ami-9d15c7f3" }
}
}
```

Examine the JSON code. This code lists the value to use for an AMI for each region.

16. Update the canvas by choosing the refresh icon.

Congratulations! You have successfully used the AWS CloudFormation Designer to add Mappings to select an AMI ID for the EC2 instance resource based on the region where the stack is deployed.

Task 4: Add Outputs

Outputs declare values that you want available to a *describe-stacks* API call or through the AWS CloudFormation console stack **Outputs** tab. In this task, you add Outputs to display the PublicIP address for the EC2 instance resource.

17. In the JSON editor pane, choose the **Outputs** tab.
18. Copy the JSON code below and paste the block into the JSON editor, **replacing everything that is currently there**.

```
{
  "Outputs": {
    "PublicIP": {
      "Value": {
        "Fn::GetAtt": [
          "ServerInstance",
          "PublicIp"
        ]
      },
      "Description": "Newly created server IP address"
    }
  }
}
```

Examine the JSON code. It is defining an output called *PublicIP* that shows the Public IP of the EC2 instance.

19. Update the canvas by choosing the refresh icon.

Congratulations! You have successfully used the AWS CloudFormation Designer to add Outputs to display the PublicIP address for the EC2 instance resource.

Task 5: Specify Resource Properties

Many resources have **properties** that define their configuration, such as selecting the instance type for an EC2 instance. In this task, you specify additional parameters for the EC2 instance.

20. Choose the **ServerInstance** icon on the canvas.

The Properties pane shows that there are no properties currently configured for the instance.

21. Copy the JSON code below and paste the block into the JSON editor, **replacing everything that is currently there**.

22.

```

{
  "Resources": {
    "ServerInstance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "InstanceType": {
          "Ref": "InstanceType"
        },
        "ImageId": {
          "Fn::FindInMap": [
            "AWSRegionToAMI",
            {
              "Ref": "AWS::Region"
            },
            "AMI"
          ]
        }
      }
    }
  }
}

```

Examine the JSON code. The *InstanceType* uses the value passed in via the Parameter defined earlier. The *ImageId* uses the Mapping to select an appropriate AMI for the region being used by the stack.

You can now save your template.

22. In the CloudFormation Designer
Toolbar:

- Choose the document icon
- Choose **Save**

23. Choose **Local file**.

24. Choose **Save**.

At this point, you see a prompt to choose the location where you can save the template file locally.

After this lab is finished, you can study this fully-valid and reusable CloudFormation template.

25. In the AWS CloudFormation Designer
toolbar:

- Validate your template by choosing .

You should see the message: *Template is valid*.

Congratulations! You now have a complete AWS CloudFormation template that you can use to create a basic server. In the following task, you use this template to create a stack.

Task 6: Create the Stack

To create a stack, you can launch the AWS CloudFormation Create Stack wizard from AWS CloudFormation Designer. Once the stack is deployed, you have a basic server up and running.

26. In the AWS CloudFormation Designer toolbar, create your stack by choosing . AWS CloudFormation Designer saves the open template in an S3 bucket, then launches the AWS CloudFormation Create Stack wizard. AWS CloudFormation automatically populates the template URL.

27. On the **Create stack** page, choose **Next**.

28. On the **Specify stack details** page, configure:

- **Stack name:**

29. Choose **Next**.

30. On the **Configure stack options** page, navigate to the bottom of the screen and choose **Next**.

31. On the **Review** page, review the configuration and choose **Submit**.

It takes a few minutes for AWS CloudFormation to create your stack.

32. Choose the refresh icon in the top-right of the screen.

33. Monitor the progress of your stack by choosing the **Events** tab.

Once the stack reaches the **CREATE_COMPLETE** status, your stack has been deployed.

Congratulations! You have successfully created your stack using the AWS CloudFormation Designer.

Task 7: View the Running Stack

In the task, you review your stack and examine the Outputs section.

34. Choose the **Outputs** tab.

The PublicIP of the EC2 instance is displayed.

You can also explore your stack template by examining the **Template** tab as required.

Congratulations! You have successfully reviewed your stack and examined the Outputs section.

Task 8: Delete the Stack

In this task, you delete the stack. Deleting the stack also deletes all the resources created by the stack template.

35. Choose **Delete**.

36. On the **Delete BasicServerStack?** pop-up window, choose **Delete stack**.

You can monitor progress of the deletion by choosing the **Events** tab.

37. Choose the **refresh** icon every 30 seconds to update the status.

Once the resources have been deleted, the stack will be removed from the list.

Congratulations! You have successfully deleted your stack and all its associated resources.

Conclusion

Congratulations! You now have successfully learned how to:

- Create a simple single-instance stack using AWS CloudFormation Designer.
 - Add and edit Mappings, Resource Parameters, and Outputs.
 - Validate the stack and save your basic template.
 - Launch the stack and view the running stack and its Outputs.
 - Delete and clean up when the stack is no longer required.
-