ECV-CF-205
Quickly Deploy Your Application through CloudFormation

2018.03.09

Version 1.1

Agenda

About this lab

Scenario

The following procedures help you quickly deploy your application through AWS CloudFormation. Modify a YAML script so that it will change original security group inbound rule, and then to confirm the update stack become effective.

AWS CloudFormation

What is AWS CloudFormation?

AWS CloudFormation is a service that helps you model and set up your AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and AWS CloudFormation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources and figure out what's dependent on what; AWS CloudFormation handles all of that.

The workshop's region will be in 'Virginia'

Prerequisites

- Sign-in a AWS account, and make sure you have select N.Virginia region.
- Make sure you have an Amazon EC2 key pair in the N.Virginia region. If you don't have, see below link:
  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair

Lab tutorial

1.1 Run the Original Template

1.1. Use your web browser to view the template, and then use the browser's Copy and Save this file to a location on your local computer's hard drive. https://github.com/awslabs/aws-cloudformation-templates/blob/master/aws/services/EC2/EC2InstanceWithSecurityGroupSample.yaml

1.2. Open this file in a text editor (preferably one that is JSON- or YAML or AWS CloudFormation- aware). In the particular, note the following features of this file:

- This template creates an EC2 security group for the instance to give you SSH access.

You will try to change security group inbound rule and confirm change status in the following steps.

1.3. On the **Services** menu, click **CloudFormation**.

1.4. Click **Create Stack**.

1.5. On the **Select Template** page, click the **Upload a template to Amazon S3** option. Click **Choose File** or **Browse** and select the template that you have been saving on your local hard drive.

1.6. Click **Next**.

1.7. On the **Specify Detail** page, use the following values:

- **Stack name: SecurityGroupCFTemplate**
- **KeyName:** select an existing EC2 keypair

1.8. Click **Next**.

1.9. On the **Options** page, create tags to associate with the resources created by the Template.Enter the following values:

- **Key: Name**
- **Value: EC2Instance**

1.10. Click **Next**.

1.11. On the **Review** page, review your input, and then click **Create**

1.12. On the **AWS CloudFormation Dashboard**, select the entry for your stack, and when the **Status** value on the **Events** tab is *CREATE_COMPLETE*, continue to the next procedure.

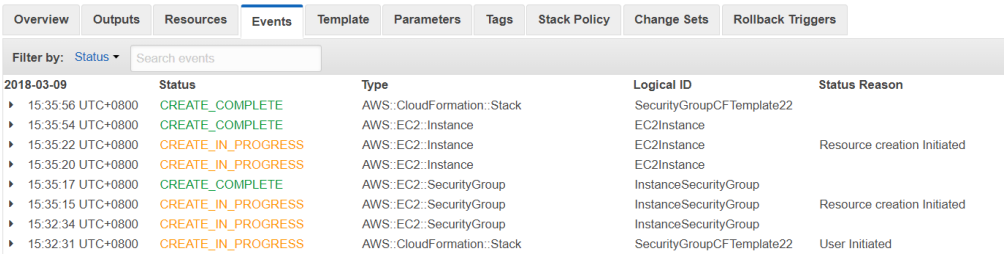

Figure1: AWS CloudFormation event status

1.13. On the **Services** menu, click **EC2**.

1.14. Click the **Running Instances** link.

1.15. Verify that your **EC2Instance**'s Security Groups is now only allow SSH traffic.

1.2 Change Security Group Inbound Rule to Template

1.16. Open your CloudFormation template just download in a text editor.

1.17. Search in the template for the string **SecurityGroupIngress**. Find the entire value of the below command:

```
IpProtocol: tcp
FromPort: '22'
ToPort: '22'
CidrIp: !Ref 'SSHLocation'
```

1.18. Copy the following text to replace above command in your template:

```
IpProtocol: tcp
FromPort: '80'
ToPort: '80'
CidrIp: 0.0.0.0/0
```

1.19. Save the template as text file.

1.3 Update Your Existing Stack

1.20. On the **Services** menu, click **CloudFormation**.

1.21. Select the template (**SecurityGroupCFTemplate**) that you created earlier.

1.22. For **Actions**, click **Update Stack**.

1.23. On the **Select Template** page, click the **Upload a template to Amazon S3** option, click **Choose File**, and then select your modified template from your hard drive.

1.24. Click **Next**.

Note: At this point, AWS CloudFormation may detect an error in your template. AWS CloudFormation can detect two types of errors before executing your template:

- Syntax errors. Common culprits of syntax errors are un balanced brackets and missing commas after an object within a JSON array. Editing your file with a text editor that supports syntax checking ("linting") of JavaScript or JSON files can help detect and eliminate these errors early.
- Reference errors. AWS CloudFormation is intelligent about the way you refer to other elements within an AWS Template file. For example, if you try to use the `Ref` intrinsic function that before you can proceed further with creating your stack.

If you receive an error at this stage, use your text editor or IDE to attempt to fix it; or ask your instructor for solve it.

1.25. On the **Specify Details** page, for BastionInstance, type t2.small.

1.26. Click **Next**.

1.27. On the **Options** page, do not change any values.

1.28. Click **Next**.

1.29. Review the summary of your changes, and then click **Update**.

1.30. The status of your stack will change to *UPDATE_IN_PROGRESS* while the update takes place. Wait until it says *UPDATE_COMPLETE* before proceeding to the next step.

1.31. On the **Services** menu, click **EC2**.

1.32. Click the **Running Instances** link.

1.33. Verify that your **EC2Instance**'s Security Groups is being update to allow HTTP traffic.

Conclusion

Congratulations! You now have learned how to:

- Use AWS CloudFormation to quickly deploy your application.
- Change yaml format template.
- Update existing stack.