Class: AWS Administration

Lab: Cloud Formation

Version: 6

Author: Curt Chambers

**Lab Synopsis**

Create and implement a simple Cloud Formation template to build a stack in AWS.

**Topics**

* Customize a basic Cloud Formation (CF) Template.
* Deploy the template from the GUI and CLI.
* See CF stack status in the GUI.
* Roll back the stack after deployment.
* See the same stack in YAML vs JSON

**Step 1**

Obtain the base Cloud Formation template here:

<https://drive.google.com/open?id=1rvM5IqTHPpvqo9EKYWO3oG-mfUlV9Hf6>

OR clone from Github if you want to try some version control:

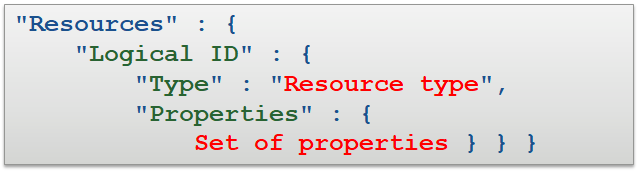
git clone https://github.com/curtchambers/cloudformation-lab.git

Open the Cloud Formation JSON template in your VS Code editor and have a look at all the sections. This template deploys the following:

First – the very top statement “AWSTemplateFormatVersion”: “2010-09-09” is a reserved statement in Amazon Cloud Formation indicating the version number of the template format – NOT your version. As such leave it alone 😊

The “Resources” entries specify all of the AWS services or components we’ll be creating – lets take a look at the one for creating the VPC itself. A complete breakdown of how resources are built is here:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resources-section-structure.html>



Add a comma between resources in the resource section!

"VPC": { *“VPC” is the Logical ID and must be unique in the template.*

"Type": "AWS::EC2::VPC", *This is the ARN for the resource type we want to create.*

"Properties": { *Property data required to create the resource.*

"CidrBlock": "10.0.0.0/16",

"EnableDnsHostnames": true,

"Tags": [ *Create a Key+Value pair to tag a name on the resource.*

{

"Key": "Name",

"Value": "CF Lab VPC"

}

]

}

},

When you create the various Logical IDs for each resource you build in the template those can be referenced elsewhere in the template as needed.

Study the remainder of the template until you completely understand what is being built in the resource section.

Note how some things must be done in the template. The Internet Gateway is created and THEN attached.

The public route table is

(1) created (PublicRouteTable)

(2) has a route actually added (PublicRoute)

(3) is associated with PublicSubnet1 (PublicSubnetRouteTableAssociation1)

Note the “Ref” statements and how they REFerence the logical IDs of the other parts of the template.

Lastly in the “Outputs” section two pieces of data are given back to the account user who created the stack:

The ID of the VPC and the Availability Zone PublicSubnet1 is in.

**Step 2**

Open the AWS console go to the Cloud Formation service and upload and create the stack from the template. You can choose to upload the template to an S3 bucket if you like OR just directly upload the template file (which WILL store it in an S3 bucket anyhow).

There is both the original Cloud Formation Stack Creation interface and the option of switching to the NEW stack creator interface – I would suggest just switching to the new one. We are not specifying a region for this stack so this VPC will be created in whatever region you happen to be in.

**Stack Name:** Name of your choosing

Look through the “stack options” page to understand what the different options do. Note the ability to add tags to all the resources. We will not need to specify an IAM role and instead use our default account credentials.

**Notification Options:** You could from this screen created an SNS ( Simple Notification Service) topic that users or even applications could subscribe to so that they are notified when the stack is completed. Very useful for triggering other processes – either by users or in an automated fashion by applications.

Create the stack & watch the progress in the Cloud Formation console – hit refresh every few seconds.

After the stack is completed locate the outputs tab for a screenshot below:

**< SCREENSHOT OF THE OUTPUTS TAB SHOWING THE VPC AND AVAILABILITY ZONE INFO >**

**< INSERT SCREENSHOT HERE >**

After the screenshot proceed to DELETE the subnet associated with the VPC that was created & wait 2 minutes.

Return to the Cloud Formation screen and under the action menu enable “Detect Drift” & wait 30 seconds.

Under the action menu select “View Drift Results” – changes to the resources as specified in the original template are called “drift” – Cloud Formation can detect those!

**< SCREENSHOT OF THE RESULTS OF THE VIEW DRIFT RESULTS >**

**< INSERT SCREENSHOT HERE >**

Under the actions menu delete the stack.

**Step 3**

Once the deletion is complete we’ll now create the same stack using the same CF template from the command line. From your AWS Admin VM run the following command – note this is ALL ON ONE LINE:

aws cloudformation create-stack --stack-name *mystackname* --template-body file://\path\to\document\cloudformationtemplate.json

The command line will respond with something similar to this:

*{*

*"StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/myteststack/330b0120-1771-11e4-af37-50ba1b98bea6"*

*}*

I would suggest moving your JSON CF template to your root directory to make it easier to reference with the file: parameter. IF your json document was stored on the root level of your C: drive on a Windows machine you would access it as such:

[File://\mydocument.json](../../../../C:/mydocument.json)

To see the event history of the stack build run the following command:

aws cloudformation describe-stack-events --stack-name mystacksname

To actually see what your stack built and their IDs in AWS run the following:

aws cloudformation list-stack-resources --stack-name *mystackname*

To delete the stack run the following:

aws cloudformation delete-stack --stack-name *mystackname*

Wait a minute and then run this to list any stacks:

aws cloudformation list-stacks --stack-status-filter CREATE\_COMPLETE

**< SCREENSHOT OF THE RESULTS OF THE LIST-STACKS COMMAND >**

**< INSERT SCREENSHOT HERE >**

**Now try it in YAML !**

Easy method - take your JSON template and convert it automatically to YAML at:

<https://www.json2yaml.com/>

Study the resultant YAML version differences. Most AWS users prefer the clarity of YAML. Examples of how to deploy different resources in AWS are USUALLY available in both YAML and JSON in the documentation.

Now write YAML code to add an additional public subnet to the existing template. If you look at the template it will require the following:

Resources Section:

* new subnet with appropriate settings (select a different AZ as well)
* new public subnet route table association

Outputs Section:

* Create an output for AZ2 to show those details

**Show the YAML code below to add the additional subnet and related pieces to the existing Template:**

Now Run the Template (either from the CLI or the GUI) and test your new addition. Troubleshoot and correct your code as needed.

Delete your stack after getting it to successfully run.

**Step 3 – Clean Up on Aisle S3 !**

Interestingly when you upload a CF json template from your local machine it doesn’t just load directly into Cloud Formation. It is saved into a auto-created bucket in your S3 storage. Anyone who has access to your account can see it in the bucket.

To clean up locate that bucket in S3 and delete all of the templates you see.