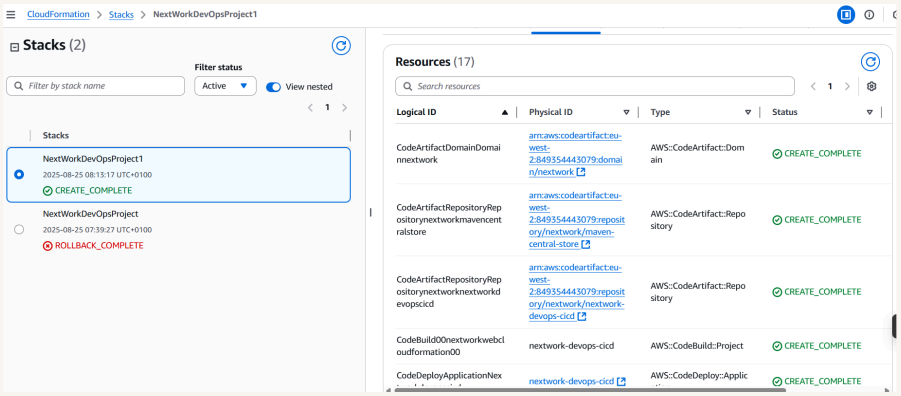
nextwork.org

Infrastructure as Code with CloudFormation

MA Marzena Pugo



MA
Marzena Pugo
NextWork Student
nextwork.org

Introducing Today's Project!

In this project, I will demonstrate how to set up a Cloud Formation template. I am doing this project to learn about infrastructure as a code and how I can apply this to my own architecture (my CI/CD architecture). By the end of this project resources as: CodeArtifact, CodeBuild, CodeDeploy, CodeConnection and more will be defined together in a single template.

Key tools and concepts

Services I used where: CodeFormation, and the code suite of services. The key concepts I learnt include Infrastructure as a code, how to use CloudFormation IAC generator, resolving circular dependencies and resources and the resources that might depend on other resources to be created first. I also learnt how to write my own resource definition in the secret mission.

Project reflection

This project took me approximately 3 hours to complete. The most challenging part was editing theCloudFormation template and finding all resources ID. It was most rewarding to deploy the template and se them live in my AWS account.

This project is part six of a series of DevOps projects where I'm building a CI/CD pipeline.

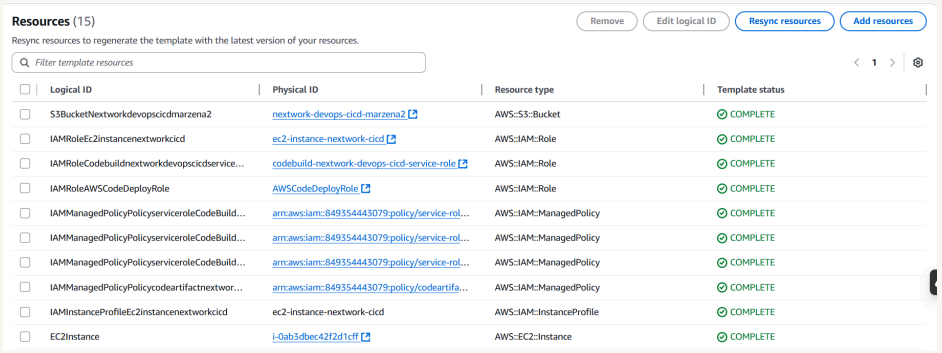
MA
Marzena Pugo
NextWork Student

Generating a CloudFormation Template

The IaC Generator is IaC Generator (Infrastructure as Code Generator) is a tool in CloudFormation that helps make writing templates much, much faster. It scans your AWS account, discovers all your resources, and generates the code for the resources you want to put in a CloudFormation template. This means you won't have to write code from scratch

A CloudFormation template is a text file that defines resources I want to deploy inside. The resources that I did add to my template include CodeArtifact domain and repositories, CodeDeploy application, IAM roles and policies ad S3 bucket.

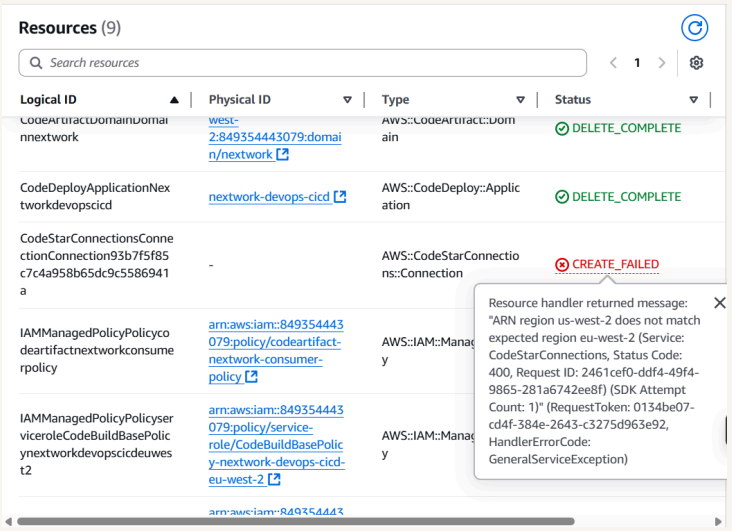
The resources I could'nt add to my template were CodeBuild projects and CodeDeploy Deployment group. This is because both resources are quite complicated in nature. There are a lot of settings that I will need to configure. The IaC generator is not yet capable to these settings so you'd have to write these definitions manually.



Template Testing

Before testing my template, I deleted the existing resources from my account because they share the same names as the resources in our generated temlate. This would cause an error - so we're deleting the existing resources to avoid that conflict /overlap.

I tested my template by launching a stack using the template I generated . The result of my first test was CREATE_FAiLED because my IAM Policy tried to attach to an IAM role that did not exist yet. This is because the role is getting created in the exact same template- so it was not yet ready by the time the Policy wanted to attach to it.



MA
Marzena Pugo
NextWork Student

DependsOn

To resolve the error, I have opened my template in a code editor to update the template manually. The DependsOn attribute means a resource needs to wait for another resource in the same template to be created first.

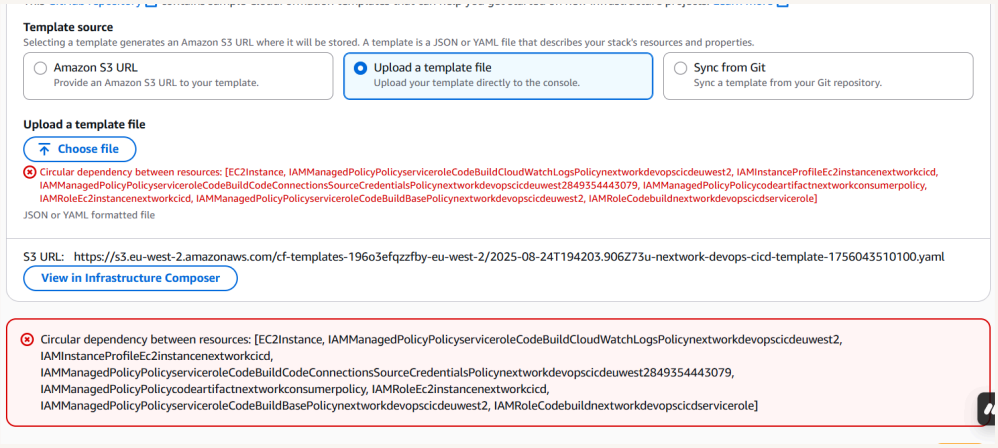
The DependsOn line was added to four different parts of my template: all 4 policies for codebuild that are policies that allow the access to: Code Artifact, Code Connection, Cloud Watch and EC2 Instance which is under CodeBuild policy. For CodeArtifact policy I have to define 2 DependsOn line, it dependOn CodeBuild service and EC2 instance role.

MA
Marzena Pugo
NextWork Student

Circular Dependencies

I gave my CloudFormation template another test! But this time there was another error : "Circular dependency". This error tells me that the CloudFormation does not know what resource should be deploy first. As it turns out my Policies depend on the roles but the Roles depend on the policies to be created. This puts CloudFormation in a loop, my template is unusable.

To fix the error I reviewed my template and tried to understand why all these Roles and Policies are referencing each other in the template. As it turns out the Role has a section "Managed PolicyArn" that references policy they use, then the Policy definitions have the "DependsOn" line that asks to wait for the rrules to be created first. To resolve this loop I simply need to delete all the "MangedPolicyArn" section off of my Roles.

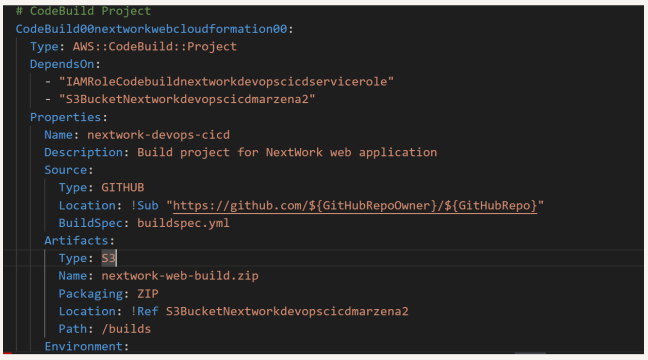


Manual Additions

In a project extension, I manually defined two more resources: a Codebuild Project and CodeDeploy deployment group.

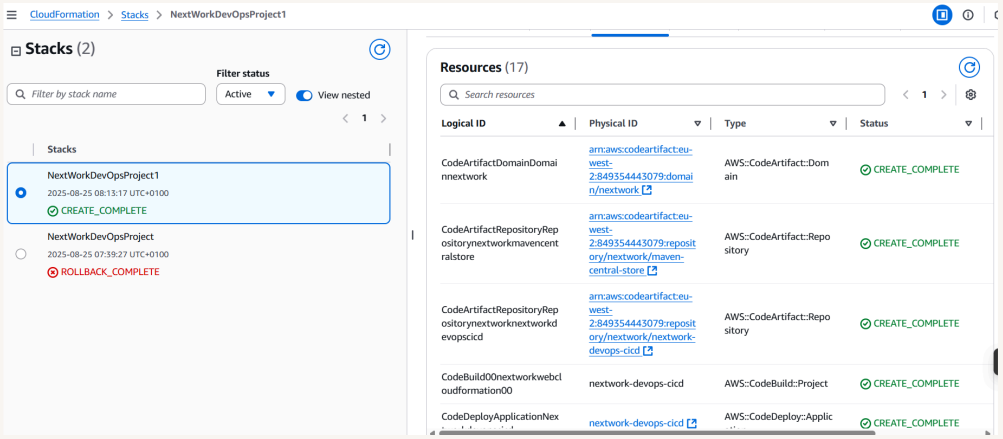
I also had to make sure the references were consistent in this template, so I edited the values for the s3 bucket ID, CodeDeploy application ID and and IAM service role IDs to match the ID of the resources in the template.

I also introduced Parameters, which are like form fields in a CloudFormation template- instead of hardcoding a value inside the template I can get the CloudFormation stack to aks the user the value of xyz when they launch te resources. In this example I created the parameters for my Github account information (account name and repository name).



Success!

I could verify all the deployed resources by visiting the hyperlinks provided for each deployed resource. I could see all of the resources deployed ni the template from IAM policies to the manuallly defined CodeBuid project.



The place to learn & showcase your skills

Check out nextwork.org for more projects