



Infrastructure as Code with CloudFormation

MA

Marzena Pugo

The screenshot shows the AWS CloudFormation console interface. On the left, the 'Stacks' section displays two stacks: 'NextWorkDevOpsProject1' (status: CREATE_COMPLETE) and 'NextWorkDevOpsProject' (status: ROLLBACK_COMPLETE). On the right, the 'Resources' section lists 17 resources, all in a 'CREATE_COMPLETE' status. The resources include:

Logical ID	Type	Status
arnaws:codeartifact:eu-west-2:849354445079:domain/nnextwork	AWS::CodeArtifact::Domain	CREATE_COMPLETE
arnaws:codeartifact:eu-west-2:849354443079:repository/nextrawork/maven-central-store	AWS::CodeArtifact::Repository	CREATE_COMPLETE
arnaws:codeartifact:eu-west-2:849354443079:repository/nextrawork/maven-devops-cicd	AWS::CodeArtifact::Repository	CREATE_COMPLETE
CodeBuildLogGroup00nextrawebcloudformation00	AWS::CodeBuild::Project	CREATE_COMPLETE
CodeDeployApplicationNextwork-devops-cicd	AWS::CodeDeploy::Application	CREATE_COMPLETE

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 were: 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 the CloudFormation template and finding all resources ID. It was most rewarding to deploy the template and see 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.

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.

The screenshot shows a web-based interface for managing CloudFormation resources. At the top, there's a header with 'Resources (15)' and buttons for 'Remove', 'Edit logical ID', 'Resync resources', and 'Add resources'. Below the header is a search bar labeled 'Filter template resources'. The main area is a table with the following columns: Logical ID, Physical ID, Resource type, and Template status. The table lists 15 resources, each with a checkbox next to it. Most resources are marked as 'COMPLETE' in the status column, except for one which is 'IN_PROGRESS'. The resources listed include various AWS services like S3, IAM, and EC2, along with some custom names like 'nextwork-devops-cicd-marzena2' and 'ec2-instance-nextwork-cicd'.

Resources (15)			
Resync resources to regenerate the template with the latest version of your resources.			
Logical ID	Physical ID	Resource type	Template status
S3BucketNextworkdevops cicdmarzena2	nextwork-devops-cicd-marzena2	AWS::S3::Bucket	COMPLETE
IAMRoleEc2Instance nextworkcicd	ec2-instance-nextwork-cicd	AWS::IAM::Role	COMPLETE
IAMRoleCodebuild nextworkdevops cicd service...	codebuild-nextwork-devops-cicd-service-role	AWS::IAM::Role	COMPLETE
IAMRoleAWSCodeDeployRole	AWSCodeDeployRole	AWS::IAM::Role	COMPLETE
IAMManagedPolicyPolicyservice roleCodeBuild...	arn:aws:iam:849354443079:policy/service-rol...	AWS::IAM::ManagedPolicy	COMPLETE
IAMManagedPolicyPolicyservice roleCodeBuild...	arn:aws:iam:849354443079:policy/service-rol...	AWS::IAM::ManagedPolicy	COMPLETE
IAMManagedPolicyPolicyservice roleCodeBuild...	arn:aws:iam:849354443079:policy/service-rol...	AWS::IAM::ManagedPolicy	COMPLETE
IAMManagedPolicyPolicycodeartifact nextwork...	arn:aws:iam:849354443079:policy/codeartifa...	AWS::IAM::ManagedPolicy	COMPLETE
IAMInstanceProfileEc2Instance nextworkcicd	ec2-instance-nextwork-cicd	AWS::IAM::InstanceProfile	COMPLETE
EC2Instance	i-0ab3dbe42f2d1cff	AWS::EC2::Instance	COMPLETE

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 template. 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.

Resources (9)				
Logical ID	Physical ID	Type	Status	
CodeArtifactDomainNameNextwork	west-2:849354443079:domain/nextwork	AWS::CodeArtifact::Domain	DELETE_COMPLETE	
CodeDeployApplicationNextworkdevops cicd	nextwork-devops-cicd	AWS::CodeDeploy::Application	DELETE_COMPLETE	
CodeStarConnectionsConnectionConnection93b7f5f85c7c4a958b65dc9c5586941a	-	AWS::CodeStarConnections::Connection	CREATE_FAILED	<div>Resource handler returned message: "ARN region us-west-2 does not match expected region eu-west-2 (Service: CodeStarConnections, Status Code: 400, Request ID: 2461cef0-ddf4-49f4- 9865-281a6742ee8f) (SDK Attempt Count: 1)" (RequestToken: 0134be07- cd4f-384e-2643-c3275d963e92, HandlerErrorCode: GeneralServiceException)</div>
IAMManagedPolicyPolicycodeartifactorynextworkconsumerpolicy	arn:aws:iam:849354443079:policy/codeartifact-nextwork-consumer-policy	AWS::IAM::ManagedPolicy		
IAMManagedPolicyPolicyservericeroleCodeBuildBasePolicynextworkdevops cicdeuwestt2	arn:aws:iam:849354443079:policy/service-role/CodeBuildBasePolicy-nextwork-devops-cicd-eu-west-2	AWS::IAM::ManagedPolicy		
	arn:aws:iam:849354443079:policy/service-role/CodeBuildBasePolicy-nextwork-devops-cicd-eu-west-2			

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.

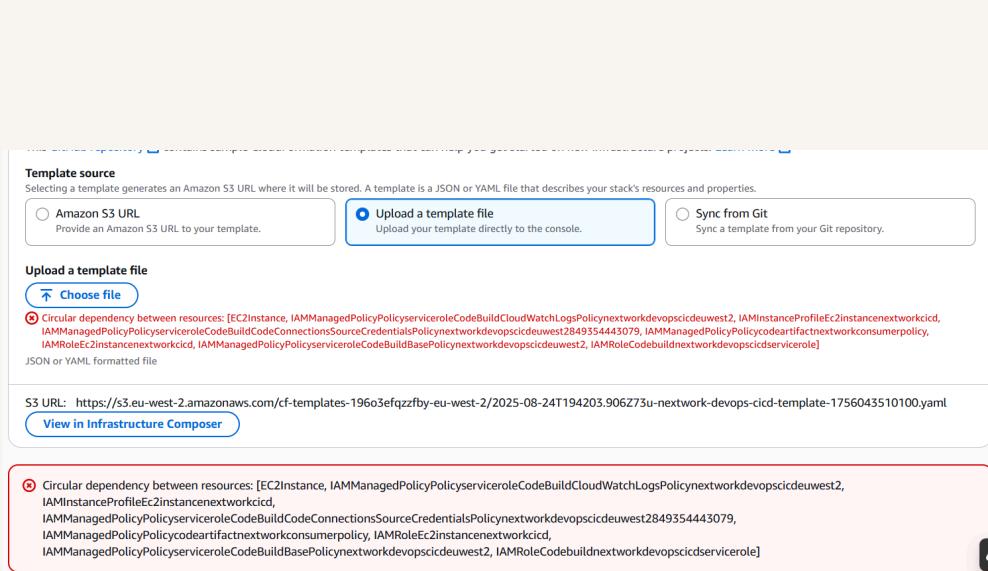
The DependsOn line was added to four different parts of my template: all 4 policies for codebuild that are policies that allow the access to: Code Artifact, Code Connection, Cloud Watch and EC2 Instance which is under CodeBuild policy. For CodeArtifact policy I have to define 2 DependsOn line, it dependOn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

```
91 IAMManagedPolicyPolicyserviceroleCodeBuildCloudWatchLogsPolicynextworkdevops cicdeuwest2:
92     UpdateReplacePolicy: "Delete"
93     Type: "AWS::IAM::ManagedPolicy"
94     DeletionPolicy: "Delete"
95     DependsOn: "IAMRoleCodebuildnextworkdevops cicd servicerole"
96
97     Properties:
98         ManagedPolicyName: "CodeBuildCloudWatchLogsPolicy-nextwork-devops-cicd-eu-west-2"
99         Path: "/service-role/"
100        Description: "Policy used in trust relationship with CodeBuild"
101        Groups: []
102        PolicyDocument:
103            Version: "2012-10-17"
104            Statement:
105                - Resource:
106                    - "arn:aws:logs:eu-west-2:849354443079:log-group:/aws/codebuild/nextwork-devops-cicd"
107                    - "arn:aws:logs:eu-west-2:849354443079:log-group:/aws/codebuild/nextwork-devops-cicd:/*"
108                Action:
109                    - "logs:CreateLogGroup"
110                    - "logs:CreateLogStream"
111                    "#LogDeliveryForCloudWatchLogs"
```

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 ask the user the value of xyz when they launch the resources. In this example I created the parameters for my Github account information (account name and repository name).

```
# CodeBuild Project
CodeBuild00nextworkwebcloudformation00:
  Type: AWS::CodeBuild::Project
  DependsOn:
    - "IAMRoleCodebuildnextworkdevopscicdservicerole"
    - "S3BucketNextworkdevopscicdmarzena2"
  Properties:
    Name: nextwork-devops-cicd
    Description: Build project for NextWork web application
    Source:
      Type: GITHUB
      Location: !Sub "https://github.com/${GitHubRepoOwner}/${GitHubRepo}"
      BuildSpec: buildspec.yml
    Artifacts:
      Type: S3
      Name: nextwork-web-build.zip
      Packaging: ZIP
      Location: !Ref S3BucketNextworkdevopscicdmarzena2
      Path: /builds
    Environment:
```

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 in the template from IAM policies to the manually defined CodeBuild project.

The screenshot shows the AWS CloudFormation console interface. On the left, the 'Stacks' section displays two stacks: 'NextWorkDevOpsProject1' (status: CREATE_COMPLETE) and 'NextWorkDevOpsProject' (status: ROLLBACK_COMPLETE). On the right, the 'Resources' section lists 17 resources, all of which are in a 'CREATE_COMPLETE' status. The resources include various AWS services such as CodeArtifactDomain, CodeArtifactRepository, CodeBuildProject, and CodeDeployApplication.

Logical ID	Physical ID	Type	Status
CodeArtifactDomainDomainName	arn:aws:codeartifact:eu-west-2:849354443079:domain/nexwork	AWS::CodeArtifact::Domain	CREATE_COMPLETE
CodeArtifactRepositoryRepositoryName	arn:aws:codeartifact:eu-west-2:849354443079:repository/nexwork/maven-central-store	AWS::CodeArtifact::Repository	CREATE_COMPLETE
CodeArtifactRepositoryRepositoryName	arn:aws:codeartifact:eu-west-2:849354443079:repository/nexwork/nextwork-devops-cicd	AWS::CodeArtifact::Repository	CREATE_COMPLETE
CodeBuildProjectName	nexwork-devops-cicd	AWS::CodeBuild::Project	CREATE_COMPLETE
CodeDeployApplicationName	nexwork-devops-cicd	AWS::CodeDeploy::Application	CREATE_COMPLETE



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

