



# Automate with CloudFormation

n



ajv.singh3107@gmail.com

Resources (10)						
Logical ID	Physical ID	Type	Status	Module		
CodeArtifactDomain00domainnextwork00W3alo	arn:aws:codeartifact:ap-south-1:011528261245:domain/nextwork	AWS::CodeArtifact::Domain	CREATE_COMPLETE	-		
CodeArtifactRepository00repositorynextworkmavencentralstore001nnkP	arn:aws:codeartifact:ap-south-1:011528261245:repository/nextwork/maven-central-store	AWS::CodeArtifact::Repository	CREATE_COMPLETE	-		
CodeArtifactRepository00repositorynextworknextworkpackages000ORFL	arn:aws:codeartifact:ap-south-1:011528261245:repository/nextwork/nextwork-packages	AWS::CodeArtifact::Repository	CREATE_COMPLETE	-		
CodeBuildProject	nextwork-web-build	AWS::CodeBuild::Project	CREATE_COMPLETE	-		
CodeCommitRepository	9efd4085-c063-4423-9eb7-d4771500bd98	AWS::CodeCommit::Repository	CREATE_COMPLETE	-		
IAMManagedPolicy00policycodeartifactnextworkconsumerpolicy00JYcpF	arn:aws:iam::011528261245:policy/codeartifact-nextwork-consumer-policy	AWS::IAM::ManagedPolicy	CREATE_COMPLETE	-		
IAMManagedPolicy00policyserviceRoleCodeBuildBasePolicynextworkwebbuildapsouth100b8fon	arn:aws:iam::011528261245:policy/service-role/CodeBuildBasePolicy-nextwork-web-build-ap-south-1	AWS::IAM::ManagedPolicy	CREATE_COMPLETE	-		
IAMManagedPolicy00policyserviceRoleCodeBuildCloudWatchLogsPolicynextworkwebbuildapsouth100CBFg	arn:aws:iam::011528261245:policy/service-role/CodeBuildCloudWatchLogsPolicy-nextwork-web-build-ap-south-1	AWS::IAM::ManagedPolicy	CREATE_COMPLETE	-		

# Introducing today's project!

## What is AWS CloudFormation?

AWS CloudFormation automates the creation and management of AWS resources using code. It allows you to define infrastructure in templates, ensuring consistent, repeatable, and scalable deployments while reducing manual errors and saving time.

## How I used CloudFormation in this project

I used AWS CloudFormation to automate the deployment of resources like IAM roles, CodeBuild, and S3 buckets, ensuring efficient infrastructure management and repeatability in today's project.

## One thing I didn't expect in this project was...

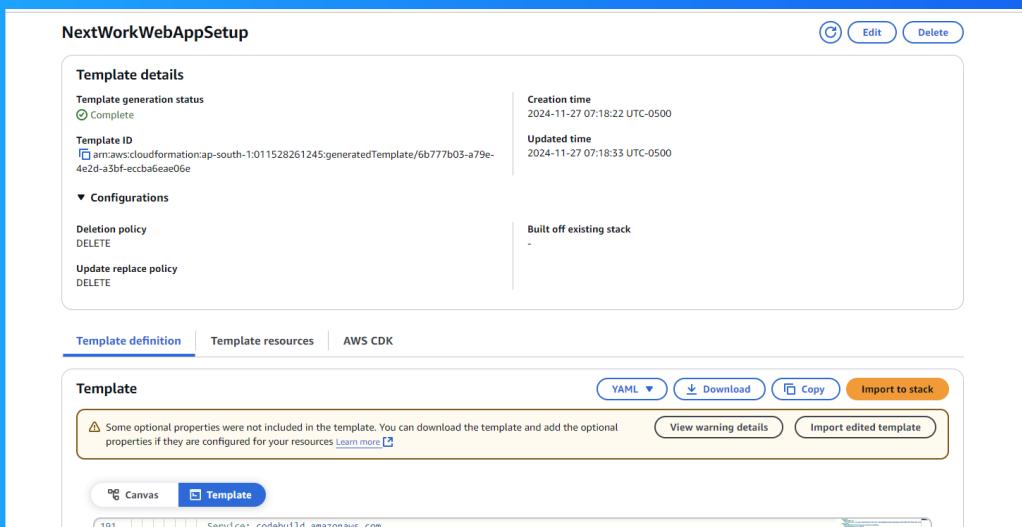
One thing I didn't expect was encountering IAM role creation issues in CloudFormation. The initial failure occurred due to the timing of policy attachment, which required the addition of the `DependsOn` attribute to ensure the IAM role was created.

## This project took me...

This project took me around 2 hours to complete. It involved creating and troubleshooting the CloudFormation template, resolving IAM role creation issues, and ensuring the proper sequencing of resources to ensure successful deployment.

# CloudFormation templates

A CloudFormation template is a JSON or YAML file that defines AWS resources and their configurations. It automates infrastructure provisioning, making deployments consistent, repeatable, and scalable by describing the resources in code.



# IaC generator

I created a CloudFormation template using the IaC generator, which defines AWS resources and their configurations as code. This ensures consistent, automated deployments, simplifies infrastructure management, and reduces manual errors.

## Not all resources could be added to my template

The resources I couldn't add to a template were specific configurations for CodeDeploy, such as the deployment group and agent setup. These require manual adjustments after deployment for precise application and environment integration.

The resources that I could add to my template include an AWS CodeArtifact domain, two repositories (Nextwork Packages and Maven Central Store), an IAM managed policy, an IAM role for CodeBuild, and an S3 bucket named Nextwork Build Artifacts Amber.



# Manually adding resources

After downloading the template, I added a CodeCommit Repository ('nextwork-web-project') to host the source code and a CodeBuild Project ('nextwork-web-build') to compile and package the NextWork web application, connecting it to S3 for artifacts.

I had to manually define these because the IaC generator doesn't support certain resources like CodeCommit repositories and CodeBuild projects, which are custom to the project's build process. These resources needed explicit inclusion in the template.

I also had to make sure the references were consistent, so I made two edits: 1. Replaced CodeBuildServiceRole with the correct service role ARN reference. 2. Replaced ArtifactsBucket with the correct S3 bucket reference for artifacts.

# Testing my template

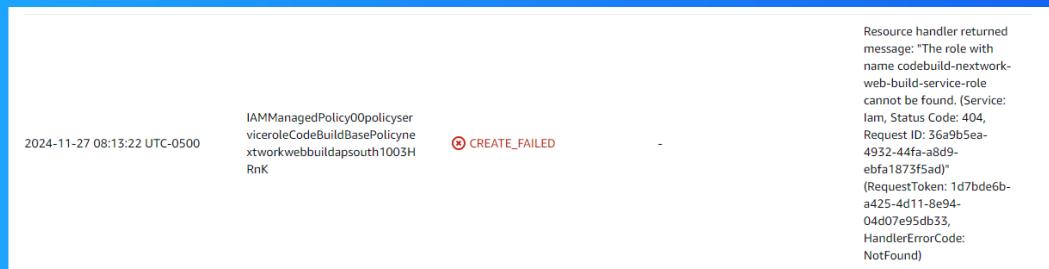
Before testing my template, I ensured that any existing resources were deleted or cleaned up to prevent conflicts. This included verifying that previous IAM roles, policies, and resources like S3 buckets were either removed or properly configured.

A stack is a collection of AWS resources that you manage as a single unit. It's created and managed through AWS CloudFormation, allowing you to deploy, update, and delete resources together, ensuring consistency across your infrastructure.

The result of my first test was a failure due to a missing IAM role. CloudFormation couldn't find the role "codebuild-nextwork-web-build-service-role" because it was being created and attached to policies simultaneously, causing the error.

# Unpacking the first error

My first template test failed because CloudFormation attempted to attach policies to the "codebuild-nextwork-web-build-service-role" IAM role before it was fully created, leading to an error since the role didn't exist yet during the provisioning.



## To fix this error, I edited my CloudFormation template.

I added the line `DependsOn: "IAMRole00codebuildnextwork-web-buildservicerole006vOvu"` across my template to ensure CloudFormation waits for the IAM role to be fully created before attempting to attach the IAM policies, resolving the dependency issue

# Fixing the first error

The DependsOn attribute means CloudFormation creates the IAM role first before attaching the IAM policies. This ensures the resources are created in the correct order, preventing errors where policies are attached to a non-existent role.

The DependsOn line was added to four different parts of my template: 1)After the IAM role definition, 2)For each IAM policy definition, 3)For the CodeBuild project, and 4)For the S3 bucket definition, ensuring the correct creation for each resource.

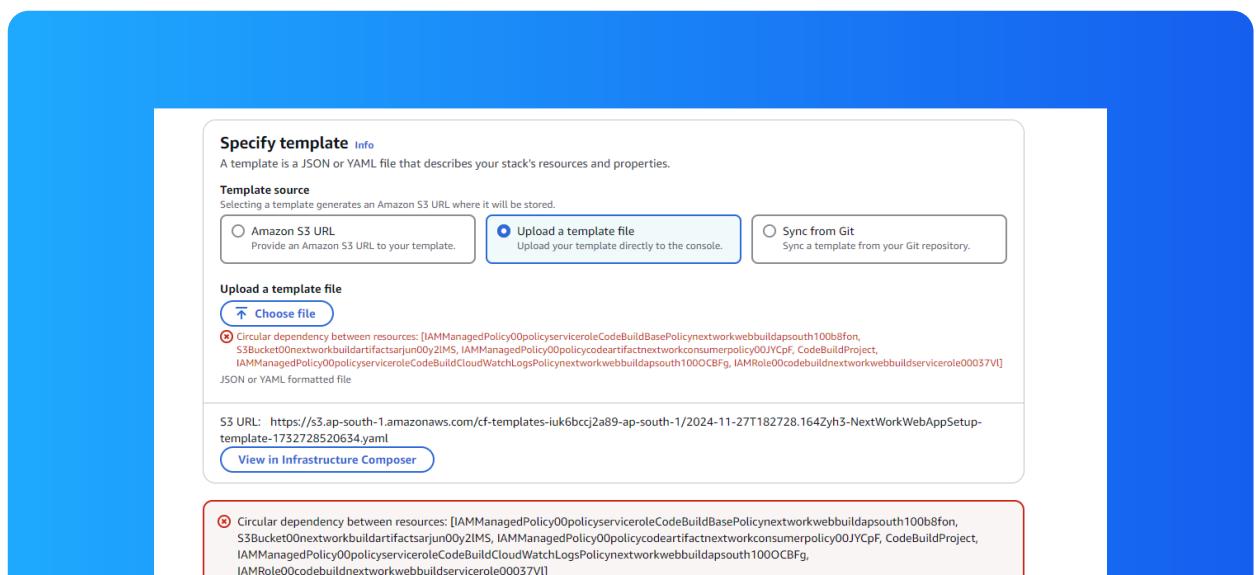
```
IAMManagedPolicy00policyserviceroleCodeBuildBasePolicynextworkwebbuildapsouth1003HRnK:  
  UpdateReplacePolicy: "Delete"  
  Type: "AWS::IAM::ManagedPolicy"  
  DeletionPolicy: "Delete"  
  DependsOn: "IAMRole00codebuildnextwork-web-buildservicerole00HmAhh"  
  Properties:  
    ManagedPolicyName: "CodeBuildBasePolicy-nextwork-web-build-ap-south-1"  
    Path: "/service-role/"  
    Description: "Policy used in trust relationship with CodeBuild"  
    Groups: []  
    PolicyDocument:  
      Version: "2012-10-17"
```

# Second template test

I gave my CloudFormation template another test! But this time, I couldn't create the stack because of a circular dependency error. This means resources depend on each other in a loop. I'll need to break this cycle by reordering resource creation.

This error means there's a circular dependency between resources in your CloudFormation template. One or more resources are referencing each other in a loop, which CloudFormation doesn't allow. You'll need to break this cycle by reordering resources.

I removed the ManagedPolicyArns section from the IAMRole00codebuild configuration to eliminate the circular dependency. Now, the IAM role won't reference the IAM policies directly, breaking the loop. This should allow the CloudFormation stack to deploy



# My final template test ☐

In my final test, creating the new stack was a great success

I could verify all the deployed resources by visiting the "Resources" tab. This tab showed a complete list of resources, their types, and their current status.

Not all resources had a shortcut URL, as some resources like IAM policies and roles aren't directly accessible. They're often referenced by other resources, and their management occurs indirectly.

Resources (10)					
Logical ID	Physical ID	Type	Status	Module	
CodeArtifactDomain00domainnex	arn:aws:codeartifact:ap-south-1:011528261245:domain/nextwork	AWS::CodeArtifact::Domain	CREATE_COMPLETE	-	
CodeArtifactRepository00repoitorynextworkmavencentralstore001nnKp	arn:aws:codeartifact:ap-south-1:011528261245:repository/nextwork/maven-central-store	AWS::CodeArtifact::Repository	CREATE_COMPLETE	-	
CodeArtifactRepository00repositorynextworknextworkpackages000ORfL	arn:aws:codeartifact:ap-south-1:011528261245:repository/nextwork/nextwork-packages	AWS::CodeArtifact::Repository	CREATE_COMPLETE	-	
CodeBuildProject	nextwork-web-build	AWS::CodeBuild::Project	CREATE_COMPLETE	-	
CodeCommitRepository	9efd4085-c063-4423-9eb7-ds7715006d88	AWS::CodeCommit::Repository	CREATE_COMPLETE	-	
IAMManagedPolicy00policycodetifactnextworkconsumerpolicy001YCPf	arn:aws:iam:011528261245:policy/codeartifact-nextwork-consumer-policy	AWS::IAM::ManagedPolicy	CREATE_COMPLETE	-	
IAMManagedPolicy00policyserviceCodeBuildBasePolicynextworkkwebuildapsouth100B8fon	arn:aws:iam:011528261245:policy/service-role/CodeBuildBasePolicy-nextwork-web-build-ap-south-1	AWS::IAM::ManagedPolicy	CREATE_COMPLETE	-	
IAMManagedPolicy00policyserviceCodeBuildCloudWatchLogsPolicynextworkwebbuildapsouth100CEBfg	arn:aws:iam:011528261245:policy/service-role/CodeBuildCloudWatchLogPolicy-nextwork-web-build-ap-south-1	AWS::IAM::ManagedPolicy	CREATE_COMPLETE	-	



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

