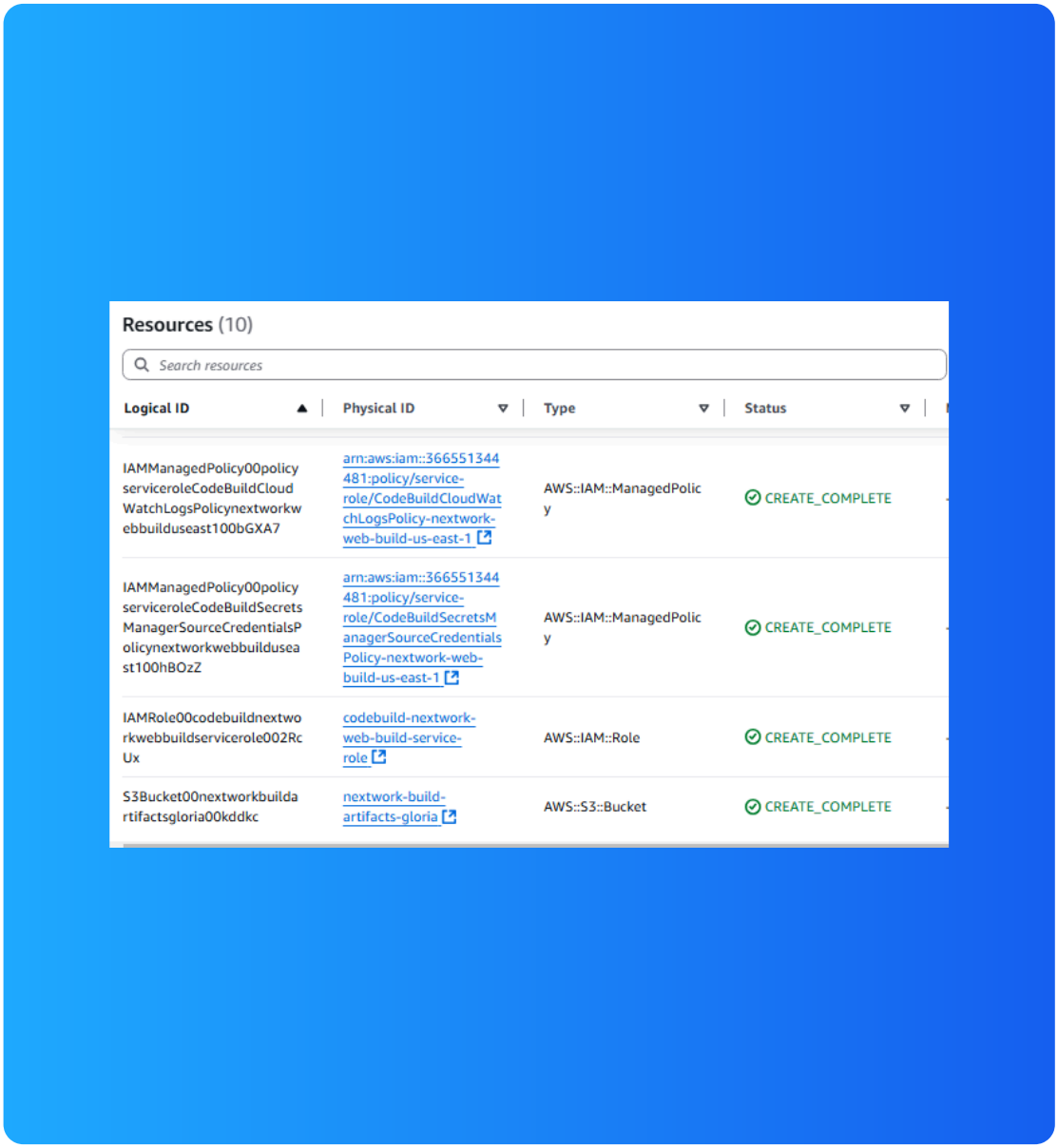
NextWork.org

Automate with CloudFormation

Gloria



Gloria
NextWork Student

Introducing today's project!

What is AWS CloudFormation?

AWS CloudFormation is a service that helps automate the setup of AWS resources using templates. It’s useful because it saves time, ensures consistency, and allows easy updates or deletion of resources as a group.

How I used CloudFormation in this project

I used AWS CloudFormation to define and deploy infrastructure resources like IAM roles, policies, S3 buckets, CodeBuild projects, and CodeArtifact repositories using a template. This automated the process, ensuring consistency and quick deployment.

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

One thing I didn't expect in this project was the circular dependency error in the CloudFormation template. It highlighted how important it is to manage resource creation order and dependencies carefully to avoid deployment issues.

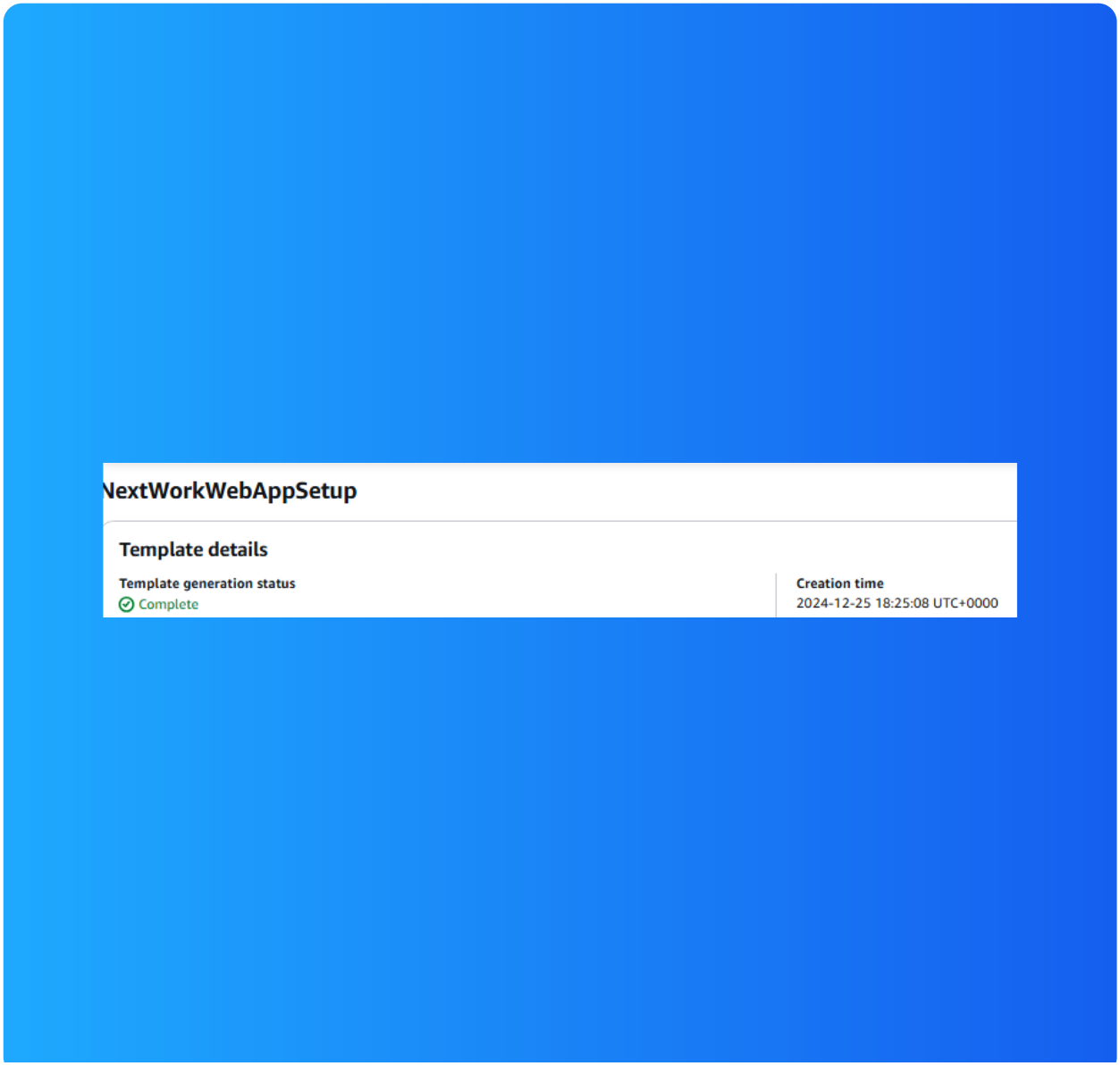
This project took me...

This project took me 1 hour and 30 minutes to complete because of the errors encountered, like the circular dependency issue and fixing resource references, which required extra time to debug and resolve.

Gloria
NextWork Student

CloudFormation templates

A CloudFormation template is a JSON or YAML file that defines AWS resources and their configurations. It acts as a blueprint for creating and managing resources like EC2, S3, and VPCs, automating setup, ensuring consistency, and saving time.



Gloria
NextWork Student

IaC generator

I created a CloudFormation template using the IaC generator, which helps write infrastructure as code. It automates defining AWS resources and configurations, making deployments faster, consistent, and easy to manage while reducing manual effort.

Not all resources could be added to my template

The resources I couldn't add to a template were Visual Studio Code (VSCode) and GitHub. These are external tools and repositories, not directly managed by CloudFormation templates, which focus on AWS infrastructure resources.

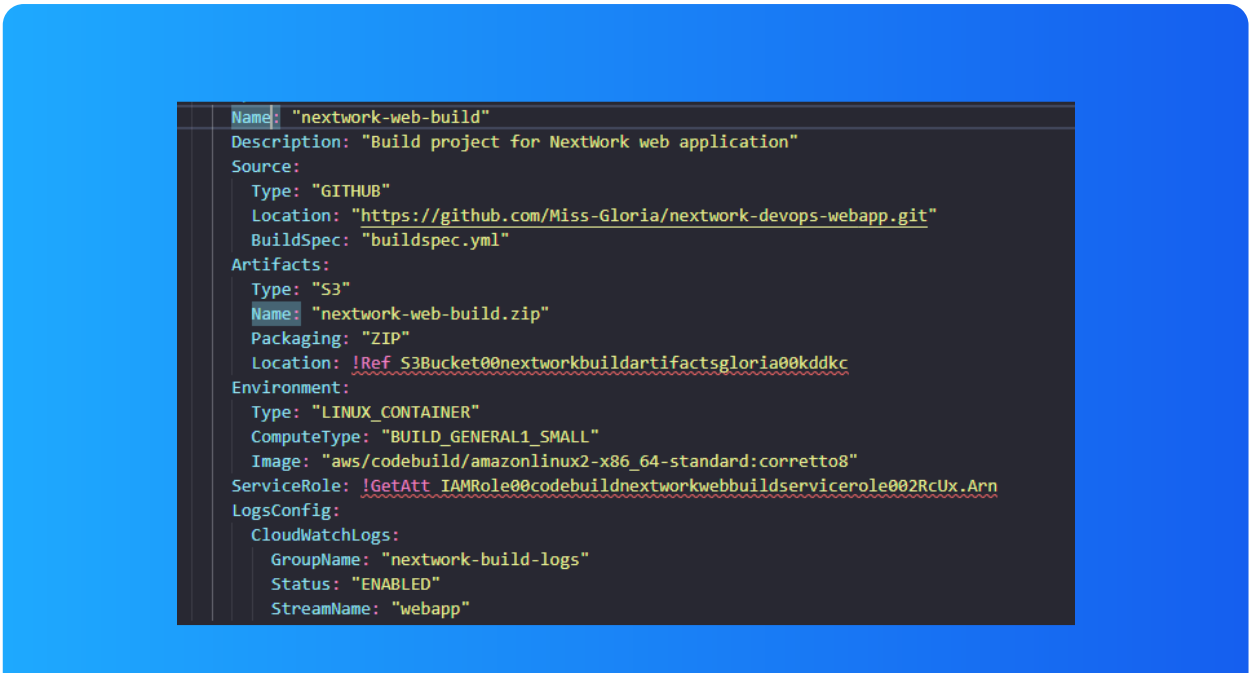
The resources I added to my template includes CodeArtifact domain nextwork, nextwork-packages, maven-central-store, IAM policy codeartifact-nextwork-consumer-policy, service role codebuild-nextwork-web-build-service-role, and S3 bucket.

Manually adding resources

After downloading the generated template, I manually defined two more resources: the CodeBuild project and the GitHub repository.The CodeBuild project compiles and packages the application, while the GitHub repository serves as the source for the app

I had to manually define these because the IaC generator cannot automatically configure resources like CodeBuild projects and GitHub repositories, as they require specific details, such as repository URLs and build settings.

I made two edits: replaced `CodeBuildServiceRole` with the correct IAM role name to ensure proper permissions and updated `ArtifactsBucket` with the S3 bucket name to correctly store build artifacts.



G

Gloria
NextWork Student

Testing my template

Before testing my template, I removed resources like the CodeCommit repo, two CodeArtifact repos, CodeArtifact domain, CodeBuild project, S3 bucket, IAM role, and three IAM policies. This was to stop errors from CloudFormation during creation.

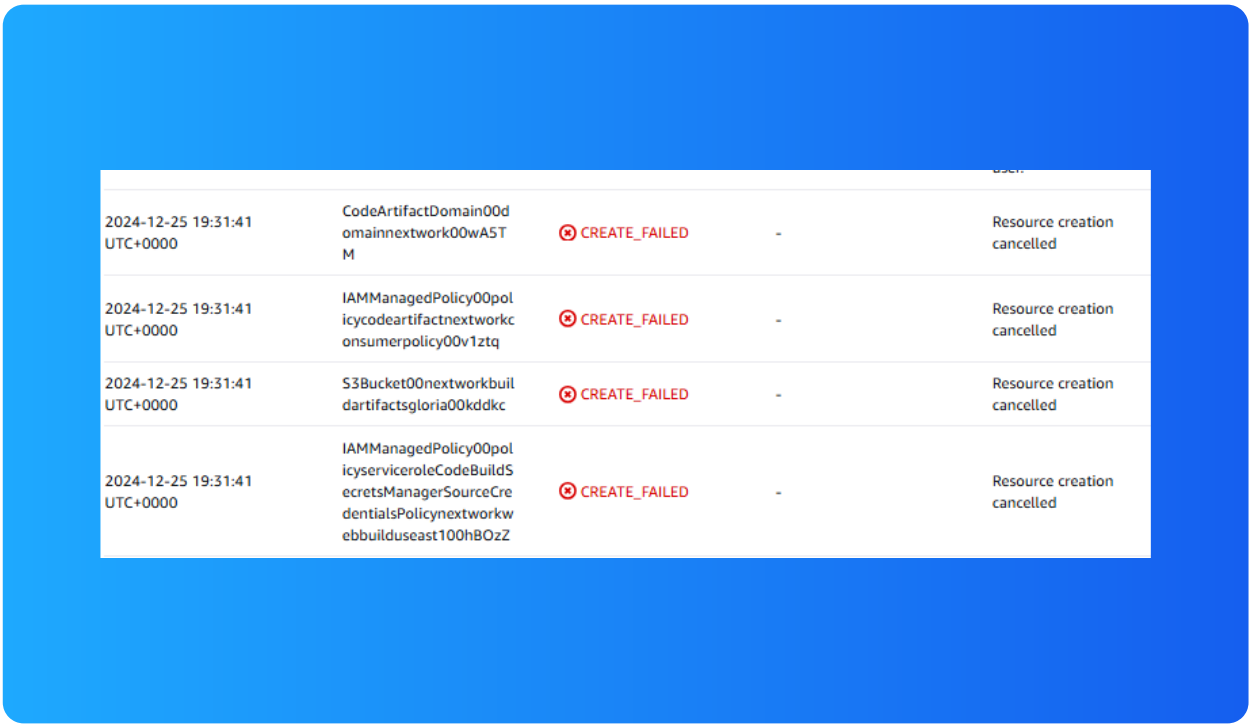
A stack is a group of AWS resources that are managed together. It is created using a CloudFormation template, and all resources in the stack are deployed, updated, or deleted as a single unit.This helps organize and automate infrastructure management

The error message is saying that CloudFormation couldn't find the IAM role codebuild-nextwork-web-build-service-role while trying to attach policies to it. This happened because the role wasn't fully created before the policies were processed.

Gloria
NextWork Student

Unpacking the first error

My first template test failed because CloudFormation tried to attach policies to the codebuild-nextwork-web-build-service-role before the role was fully created.This happened because the role and policies were being created together hence the error.



2024-12-25 19:31:41 UTC+0000	CodeArtifactDomain00domainnextwork00wA5TM	CREATE_FAILED	-	Resource creation cancelled
2024-12-25 19:31:41 UTC+0000	IAMManagedPolicy00policycodeartifactnextworkconsumerpolicy00v1ztq	CREATE_FAILED	-	Resource creation cancelled
2024-12-25 19:31:41 UTC+0000	S3Bucket00nextworkbuildartifactsgloria00kddkc	CREATE_FAILED	-	Resource creation cancelled
2024-12-25 19:31:41 UTC+0000	IAMManagedPolicy00policyserviceroleCodeBuildSecretsManagerSourceCredentialsPolicynextworkwebbuilduseast100hBOzZ	CREATE_FAILED	-	Resource creation cancelled

To fix this error, I edited my CloudFormation template.

I added the line DependsOn: IAMRole00codebuildnextworkwebbuildservicerole002RcUx across the template to ensure that the IAM role was created first before attempting to attach any policies, resolving the dependency issue.

Gloria
NextWork Student

Fixing the first error

The DependsOn attribute means CloudFormation will create the IAM role first before attaching the policies. This ensures the resources are created in the correct order, preventing errors caused by trying to attach policies to a role that doesn't exist

The DependsOn line was added to four different parts of my template: the CodeBuildBasePolicy, CodeBuildCloudWatchLogsPolicy, CodeBuildSecretsManagerSourceCredentialsPolicy, and codeartifact-nextwork-consumer-policy sections.

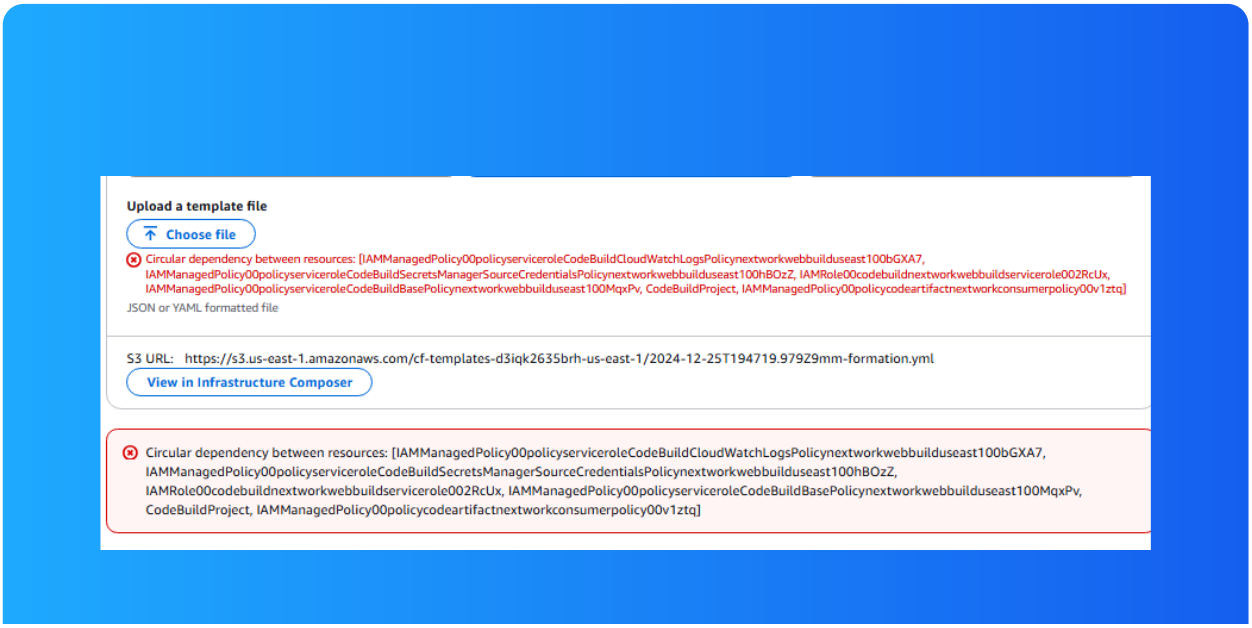
Gloria
NextWork Student

Second template test

I gave my CloudFormation template another test! This time, it failed due to a circular dependency error. The IAM role and policies referenced each other, causing CloudFormation to struggle with the resource creation order, resulting the error.

This error means CloudFormation couldn't create the resources because the IAM role and its policies needed each other to be created first. This caused a loop, making it unclear which one to create first, so the stack failed.

To fix this error, I removed the references to the IAM policies in the `ManagedPolicyArns` section of the IAM role. This broke the circular dependency and allowed CloudFormation to create the resources in the correct order without any issues.



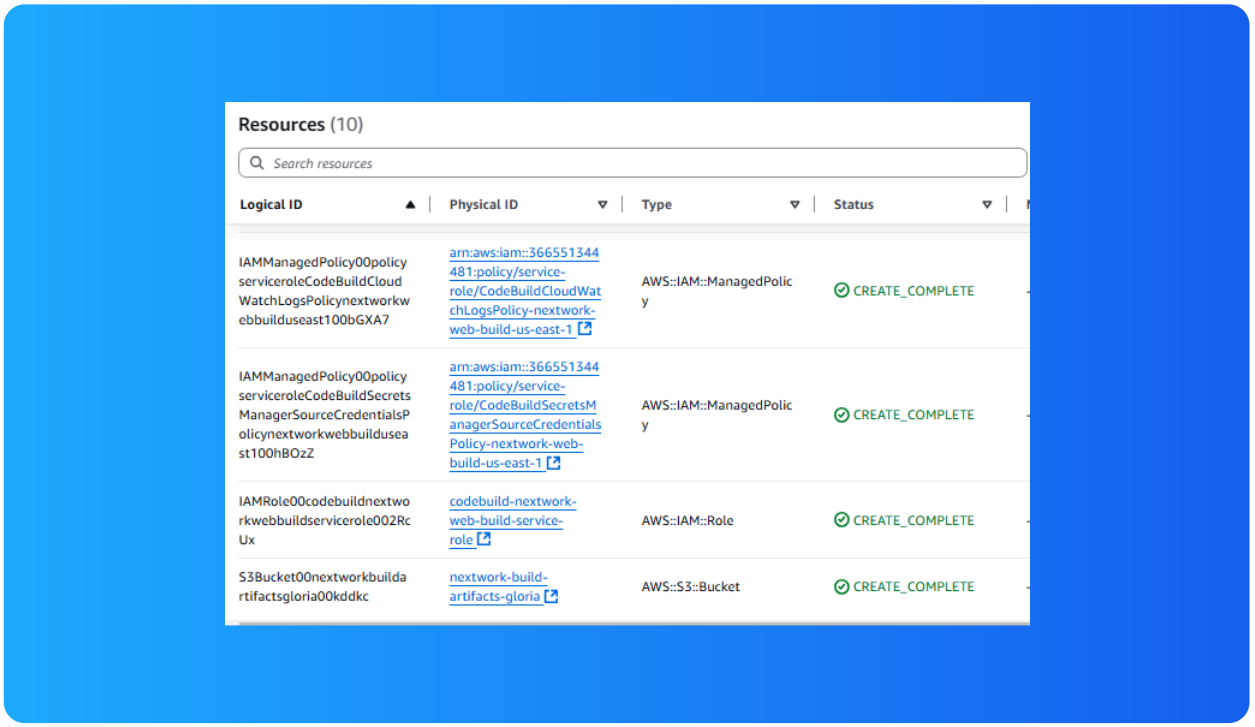
Gloria
NextWork Student

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 in the CloudFormation console. This tab shows a list of all the resources created as part of the stack, along with their current statuses and details.

Not all the resources in the list had a shortcut URL because some, like IAM roles or policies, don't have direct web interfaces. These resources are managed through other AWS services or APIs and don't require a clickable link in the console.



Everyone should be in a job they love.

Check out nextwork.org for more projects