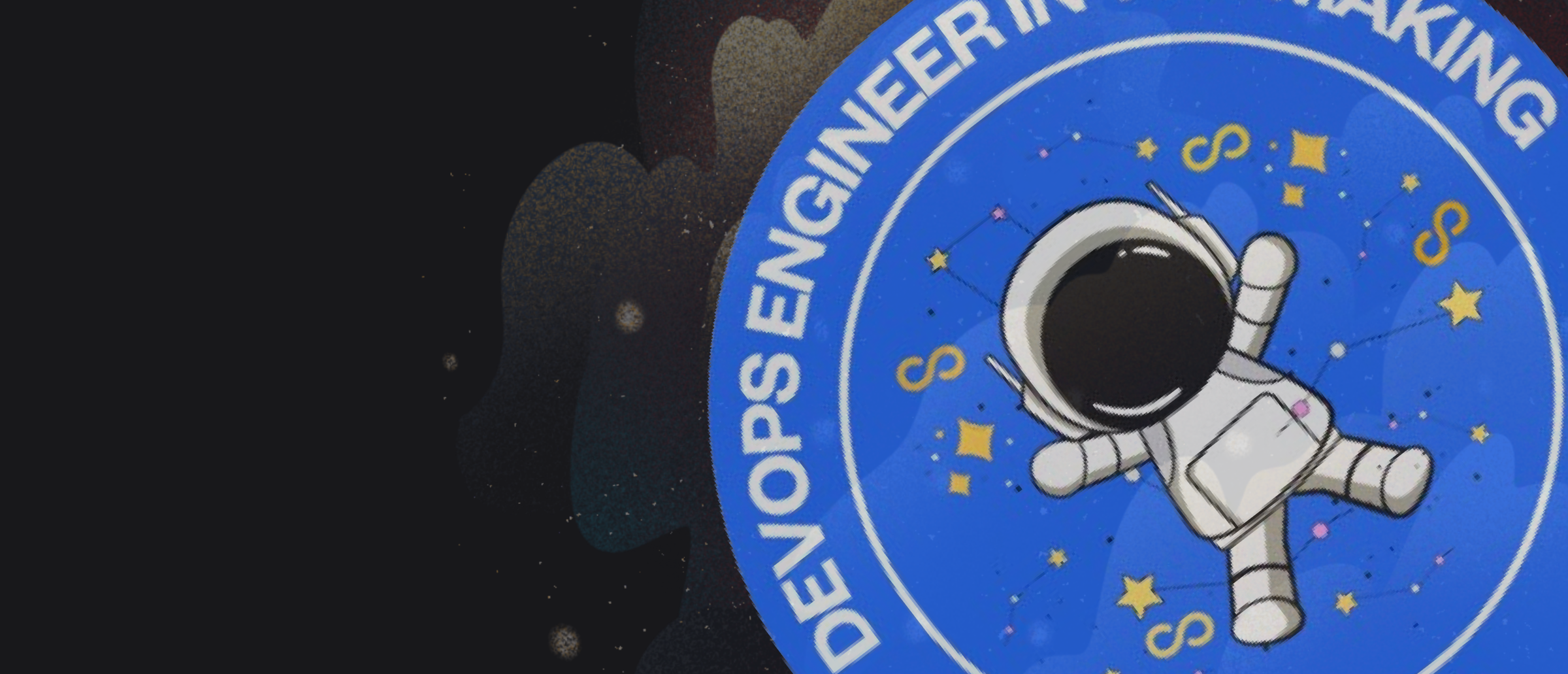
NEXTWORK

DevOps x AWS
Series VI

Automate with CloudFormation

Dahri Hadri



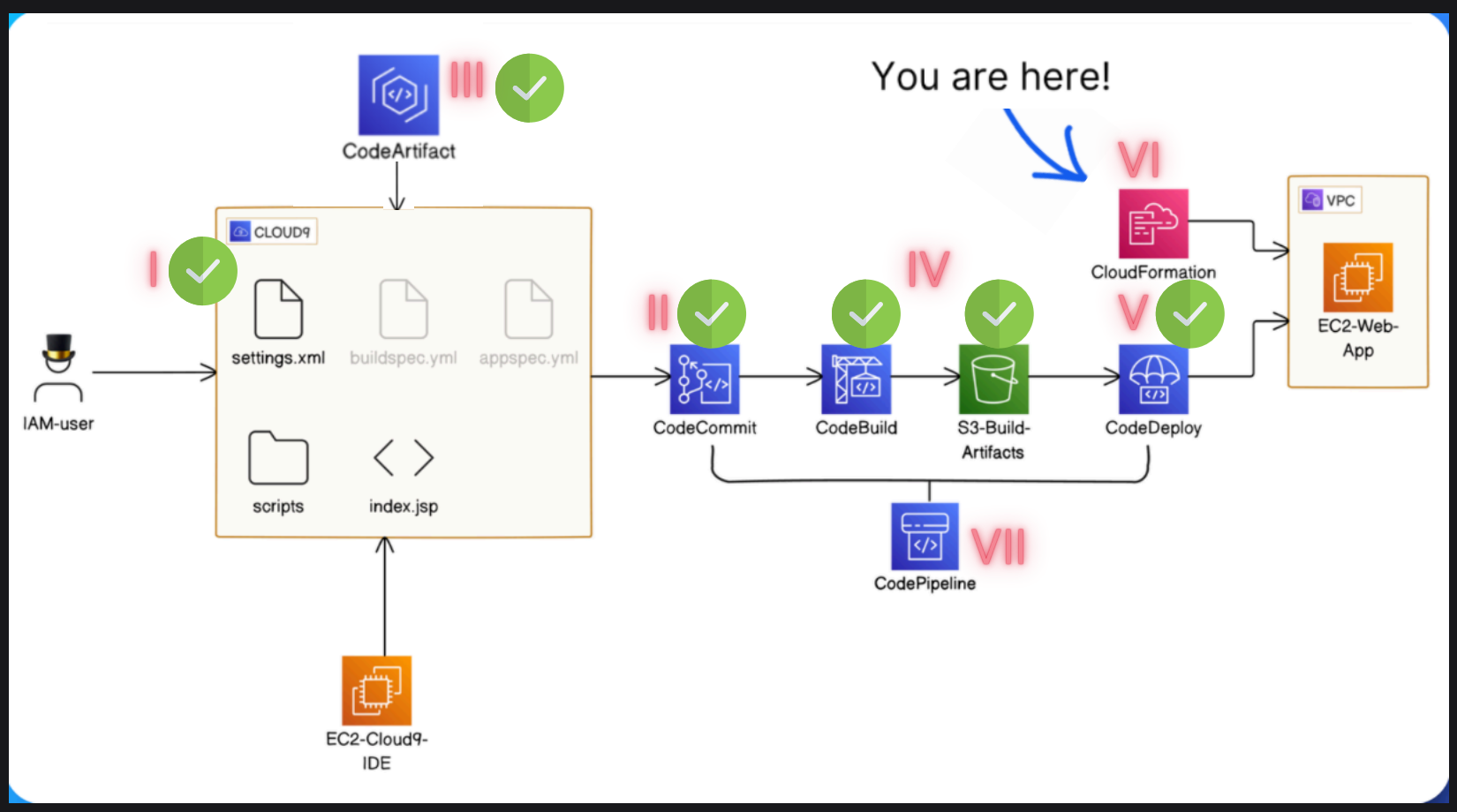
Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

DevOps x AWS

For this DevOps x AWS series, I am sharing 7 projects. In this SEVEN-project series, I will create a CI/CD pipeline to build and deploy a simple web application using AWS Code services. Here's what I'll build at the end of ALL seven projects:

I. Set up a Web App + IDE with Cloud9
II. Set Up A Git Repository with AWS CodeCommit
III. Secure Project Dependencies with AWS CodeArtifact
IV. Package an App with AWS CodeBuild
V. Deploy an App witth AWS CodeDeploy
VI. Automate with AWS CloudFormation
VII. CI/CD Pipeline with AWS CodePipeline



Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

Introducing AWS CloudFormation!

What it does & how it’s useful

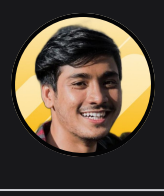
AWS CloudFormation automates resource creation and updates using coded templates, saving time and reducing human error. Developers and teams use AWS CloudFormation because it ensures consistent setups, automates deployments, and integrates with other AWS services seamlessly.

How I’m using it in today’s project

I'm using AWS CloudFormation to automate the setup of my web application's infrastructure and CI/CD pipeline. This ensures that all necessary resources are created consistently and efficiently, reducing manual setup and potential errors.

This project took me...

1.5 hours to complete, including setup and testing of all resources. Documentation took me around 40 minutes to write, ensuring detailed and clear instructions for each step.


Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

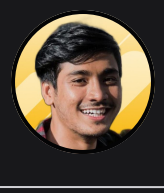
Setting up a template

- A CloudFormation template is a text file in JSON or YAML format that describes the AWS resources you want to create and manage in your stack.

- I created a CloudFormation template using the IaC generator, which scans and captures your existing AWS resources to automate their setup and management.

- Surprisingly, not all resources could be added to my template. For my web app project, the resources I couldn't add to a template were CodeCommit repository, and CodeBuild project.

- However, the resources that I could add to my template were:
  - CodeArtifact domain
  - Local CodeArtifact repository
  - Upstream CodeArtifact repository
  - IAM Policy for CodeBuild CloudWatch Logs
  - IAM Service Role for CodeBuild
  - S3 bucket



Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

A peek into my created CloudFormation template!

aws | Services | Search [Alt+S] | Sydney | dahrihadri-IAM-Admin @ 6374-2337-5624

Template generation status
Complete

Creation time
2024-07-23 09:55:11 UTC+0800

Template ID
arn:aws:cloudformation:ap-southeast-2:637423375624:generatedTemplate/bdf0f1d2-499a-4315-ba26-46067e353878

Updated time
2024-07-23 09:55:25 UTC+0800

▼ Configurations

Deletion policy
DELETE

Built off existing stack
-

Update replace policy
DELETE

Template definition | Template resources | AWS CDK

Template

YAML | Download | Import to stack

⚠ Some optional properties were not included in the template. You can download the template and edit it to add the optional properties if they are relevant for your resources. Learn more

View warning details | Import edited template

Copy

```
---
Metadata:
  TemplateId: "arn:aws:cloudformation:ap-southeast-2:637423375624:generatedTemplate/bdf0f1d2-499a-4315-ba26-46067e353878"
Resources:
  IAMManagedPolicy00policycodeartifactnextworkconsumerpolicy00AMJWH:
    UpdateReplacePolicy: "Delete"
    Type: "AWS::IAM::ManagedPolicy"
    DeletionPolicy: "Delete"
    Properties:
      ManagedPolicyName: "codeartifact-nextwork-consumer-policy"
      Path: "/"
      Description: "Provides permissions to read from CodeArtifact."
      Groups: []
      PolicyDocument:
        Version: "2012-10-17"
        Statement:
        - Resource: "*"
          Action:
          - "codeartifact:GetAuthorizationToken"
          - "codeartifact:GetRepositoryEndpoint"
          - "codeartifact:ReadFromRepository"
```

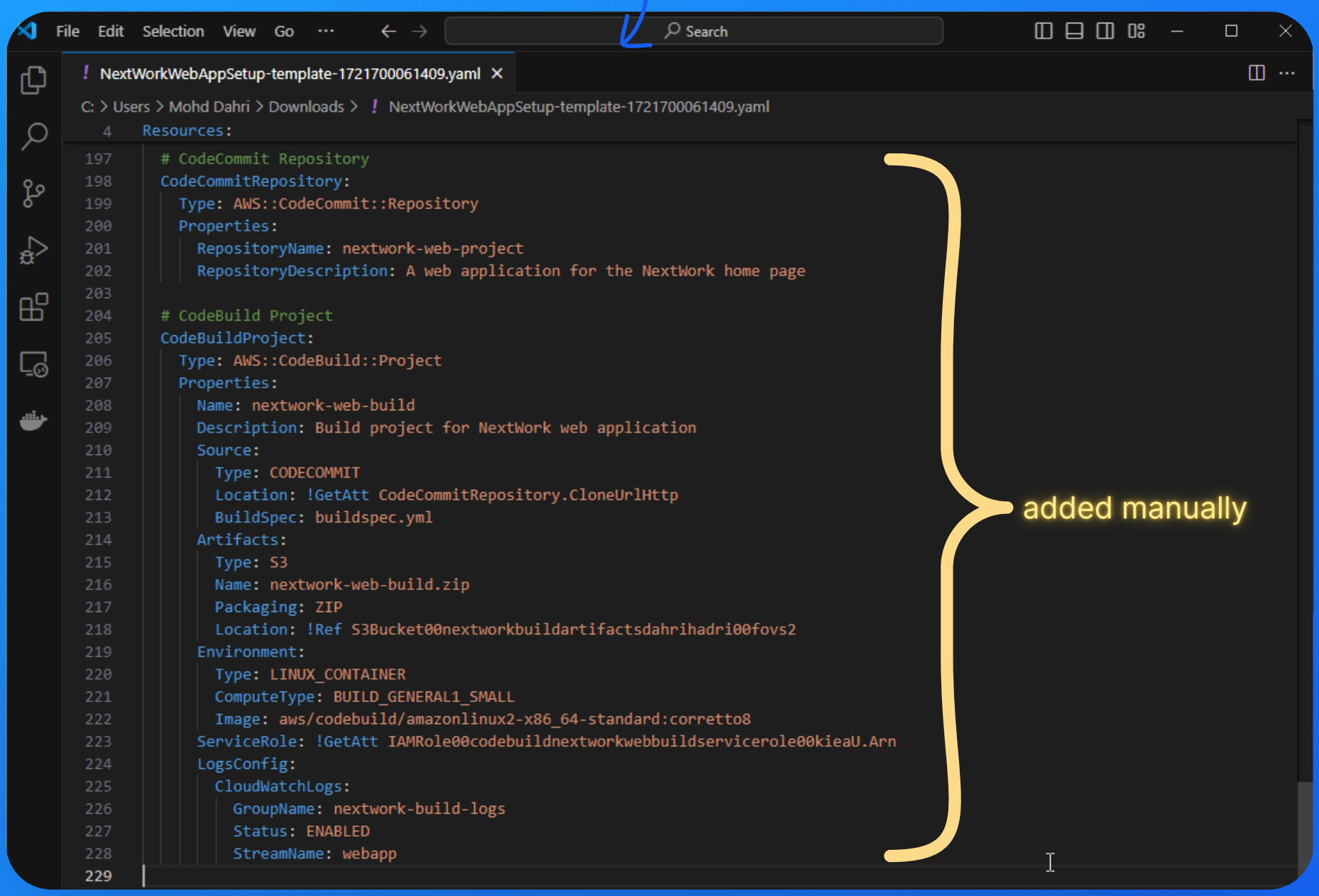
CloudShell | Feedback | © 2024, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences

Dahri Hadri
linkedin.com/in/dahrihadri

Manually adding resources

- After downloading the generated template, I manually defined two more resources: CodeCommit Repository and CodeBuild Project.
- I had to manually define these because the IaC generator couldn't include them due they still require manual steps once I've deployed them.
- I also had to make sure the references were consistent in this template, which meant:
  - Replacing the placeholder for the CodeBuild service role with the actual service role name.
  - Replacing the placeholder for the S3 bucket with the actual bucket configuration name.

Editing the CloudFormation template for a seamless setup



Dahri Hadri
linkedin.com/in/dahrihadri

My first template test

- Before testing my template, I had to delete existing resources in my AWS account to avoid conflicts, such as CodeCommit repositories, CodeArtifact domains, and IAM roles.

- To test my CloudFormation template, I created a stack. A stack is a collection of AWS resources created and managed together using CloudFormation.

- The result of this first test was an error because CloudFormation couldn’t find the IAM role it was trying to use. This occurred as policies were being attached before the role was fully created.

CloudFormation Error: IAM Role Missing

2024-07-23 11:02:52 UTC+0800	IAMManagedPolicy00policyserviceroleCodeBuildBasePolicynextworkwebbuildapsoutheast200C4Dsd	CREATE_FAILED	-	Resource handler returned message: "The role with name codebuild-nextwork-web-build-service-role cannot be found. (Service: Iam, Status Code: 404, Request ID: c46512fc-ce93-4824-935f-b99cae3dbc09)" (RequestToken: 956a7901-448e-916e-6bde-71140f5001a1, HandlerErrorCode:

Dahri Hadri
linkedin.com/in/dahrihadri

Fixing template errors

- To fix the error, I went back into my CloudFormation template to make some manual edits.

- I added the line, *DependsOn: CodeBuildServiceRole*, across my template. This line ensures that CloudFormation will create the IAM role first before attempting to create and attach the IAM policies or use the role in the CodeBuild project.

- This was added to four different parts of my template:
  - The three ***IAM policy configurations***.
  - The ***CodeBuild project configuration***.
-]

The DependsOn line added to one of my IAM policy definitions

```
207   # CodeBuild Project
208   CodeBuildProject:
209     DependsOn: "IAMRole00codebuildnextworkwebbuildservicerole00kieaU"
210     Type: AWS::CodeBuild::Project
211     Properties:
212       Name: nextwork-web-build
213       Description: Build project for NextWork web application
214       Source:
215         Type: CODECOMMIT
216         Location: !GetAtt CodeCommitRepository.CloneUrlHttp
217         BuildSpec: buildspec.yml
218       Artifacts:
219         Type: S3
220         Name: nextwork-web-build.zip
221         Packaging: ZIP
222         Location: !Ref S3Bucket00nextworkbuildartifactsdahrihadri00fovs2
223       Environment:
224         Type: LINUX_CONTAINER
225         ComputeType: BUILD_GENERAL1_SMALL
226         Image: aws/codebuild/amazonlinux2-x86_64-standard:corretto8
```

Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

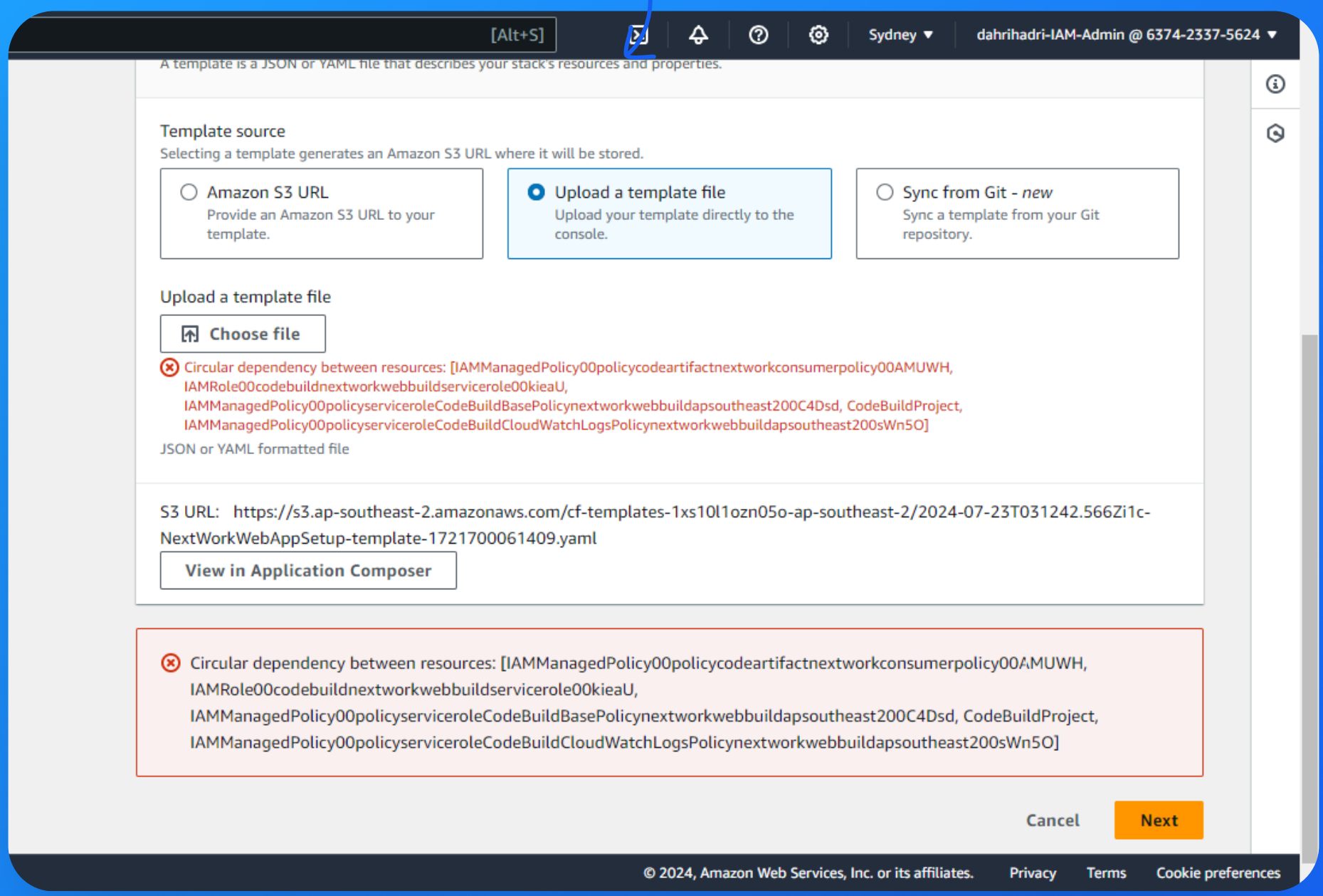
My second template test

- I gave my CloudFormation template another test! But this time, I couldn't create the stack because of an error. This is known as a circular dependency.

- This error means CloudFormation is stuck in a loop because some resources depend on each other to be created first. It's like a circular wait where resources are waiting for each other to be ready.

- To fix this error, I removed the circular references in my CloudFormation template by deleting the lines that referenced IAM policies in the IAM role configuration. This break in dependency allowed CloudFormation to process the stack without the circular loop issue.

Error Alert: Circular Dependency Detected 🚨



Dahri Hadri
linkedin.com/in/dahrihadri

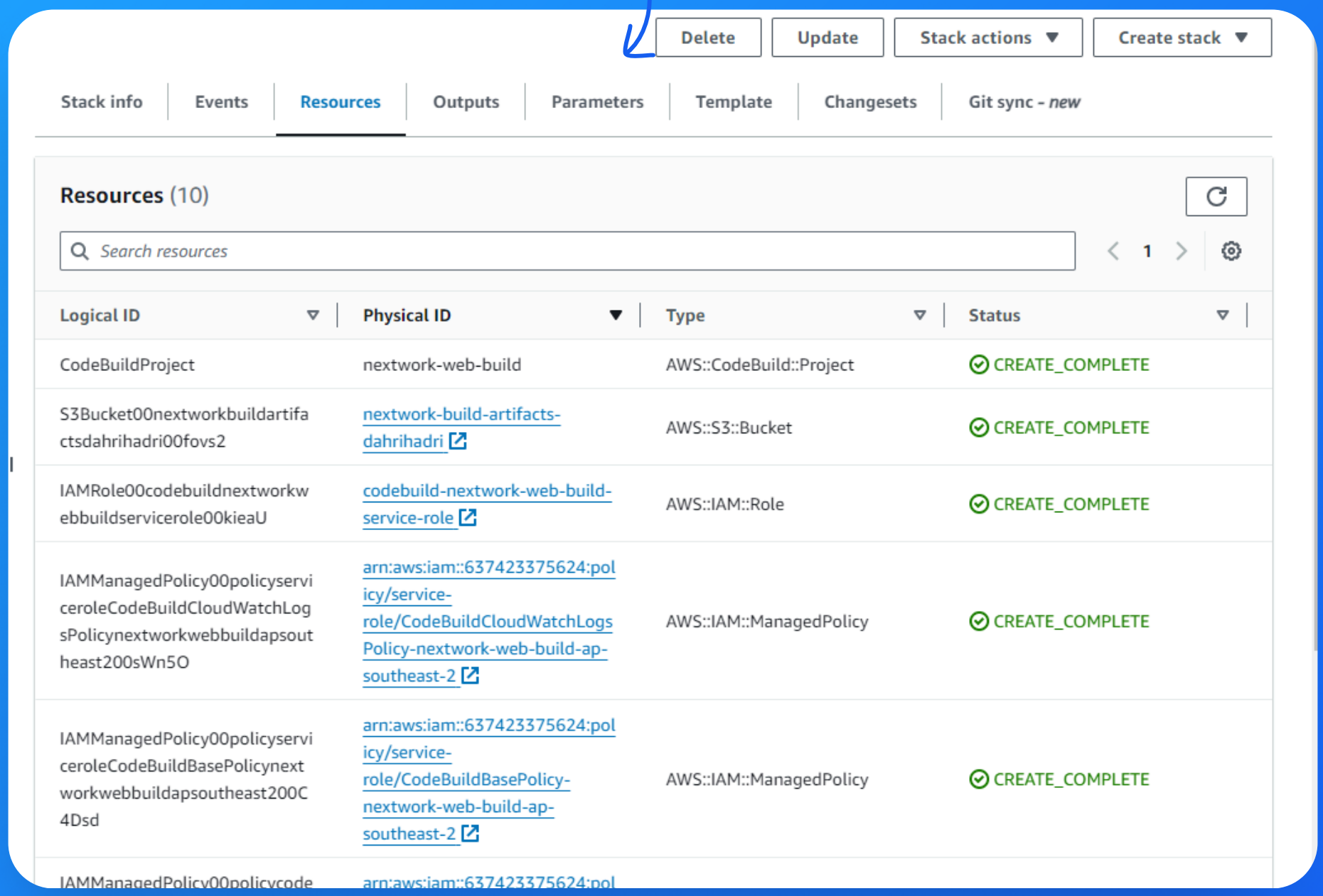
DevOps x AWS
Series VI
NEXTWORK

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.

- Not all the resources in the list had a shortcut URL because some resources, like the S3 artifacts bucket, are deployed in a specific region and CloudFormation can't produce a direct link to them.

My created resources. Looking flashy!

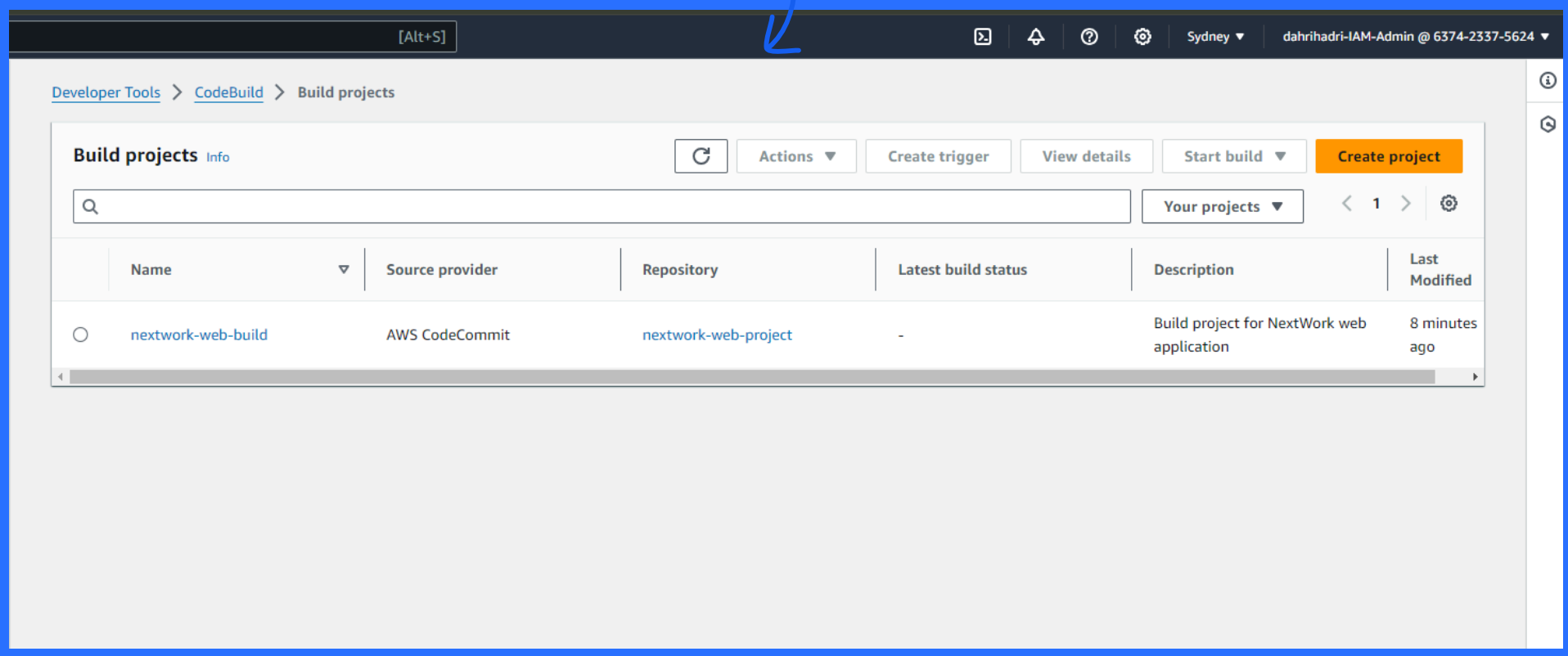


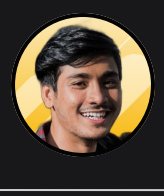
Dahri Hadri
linkedin.com/in/dahrihadri

Other resources 👀

- I can find the other resources that CloudFormation has created for me:
  - My CodeArtifact domain
  - My CodeArtifact repostiory
  - My S3 Artifacts bucket
  - My CodeCommit repository
  - My CodeBuild project
- For example, I able to find my CodeBuild project in the CodeBuild console!

CodeBuild project in the CodeBuild console!





Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

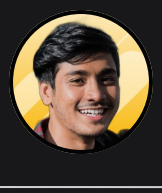
My key learnings

1. CloudFormation template is a blueprint for defining AWS resources, while a stack is a deployed instance of that template, managing and provisioning those resources.

2. Not all resources can be automatically included in an IaC generator because some need manual setup or additional configuration beyond basic template definitions.

3. Even after using an IaC generator, errors during deployment can occur due to circular dependencies and missing references in the template.

4. Something I didn’t expect was encountering circular dependencies, where resources seemed stuck waiting for each other to be created.



Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

My CI/CD pipeline so far...

1. **AWS Cloud9** is responsible for IDE setup and development environment management.

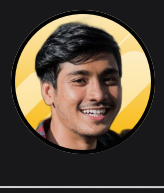
2. **AWS CodeCommit** is responsible for hosting secure and scalable Git repositories in the AWS cloud, facilitating collaborative software development workflows.

3. **AWS CodeArtifact** is responsible for securely storing and managing software packages and dependencies, ensuring reliable and scalable artifact management.

4. **AWS CodeBuild** is responsible for automating the build and testing of code in the cloud, providing scalable and efficient infrastructure for continuous integration and delivery (CI/CD) pipelines.

5. **AWS CodeDeploy** is responsible to automates deployment processes, ensuring consistent and reliable application updates across EC2 instances and on-premises servers.

6. AWS CloudFormation is responsible orchestrating infrastructure as code to automate resource creation.

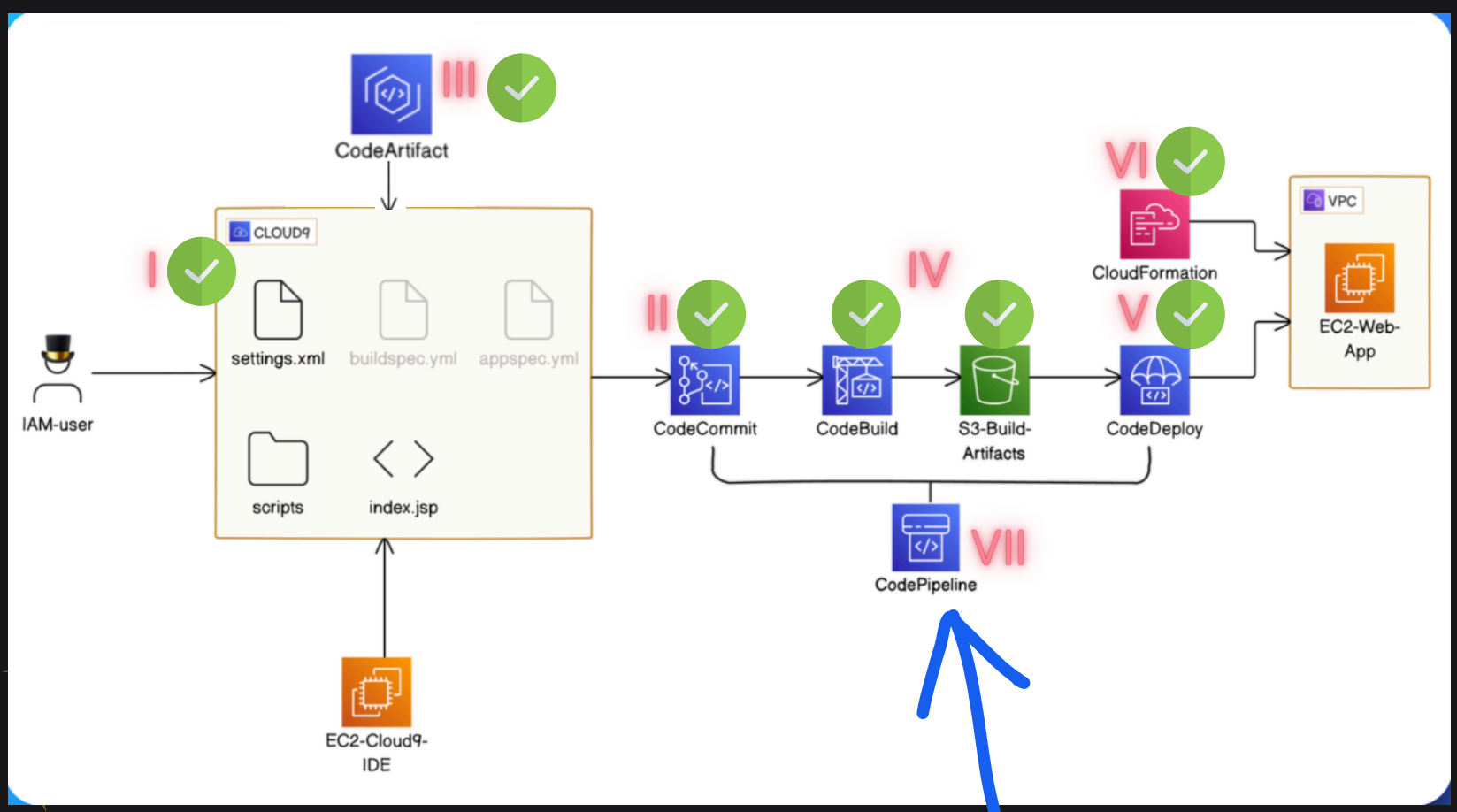


Dahri Hadri
linkedin.com/in/dahrihadri

DevOps x AWS
Series VI
NEXTWORK

Great!

we are done with series VI



I will build the FINAL part of this CI/CD pipeline - CodePipeline - in the next project!

NEXTWORK

Find this helpful?

Like this post

yes!

Leave a comment

Save for later

Let's connect!

Dahri Hadri

@dahrihadri

https://www.linkedin.com/in/dahrihadri

Ask me about it