



# DevOps x AWS

## Series IV

# Package an App with CodeBuild



Dahri Hadri

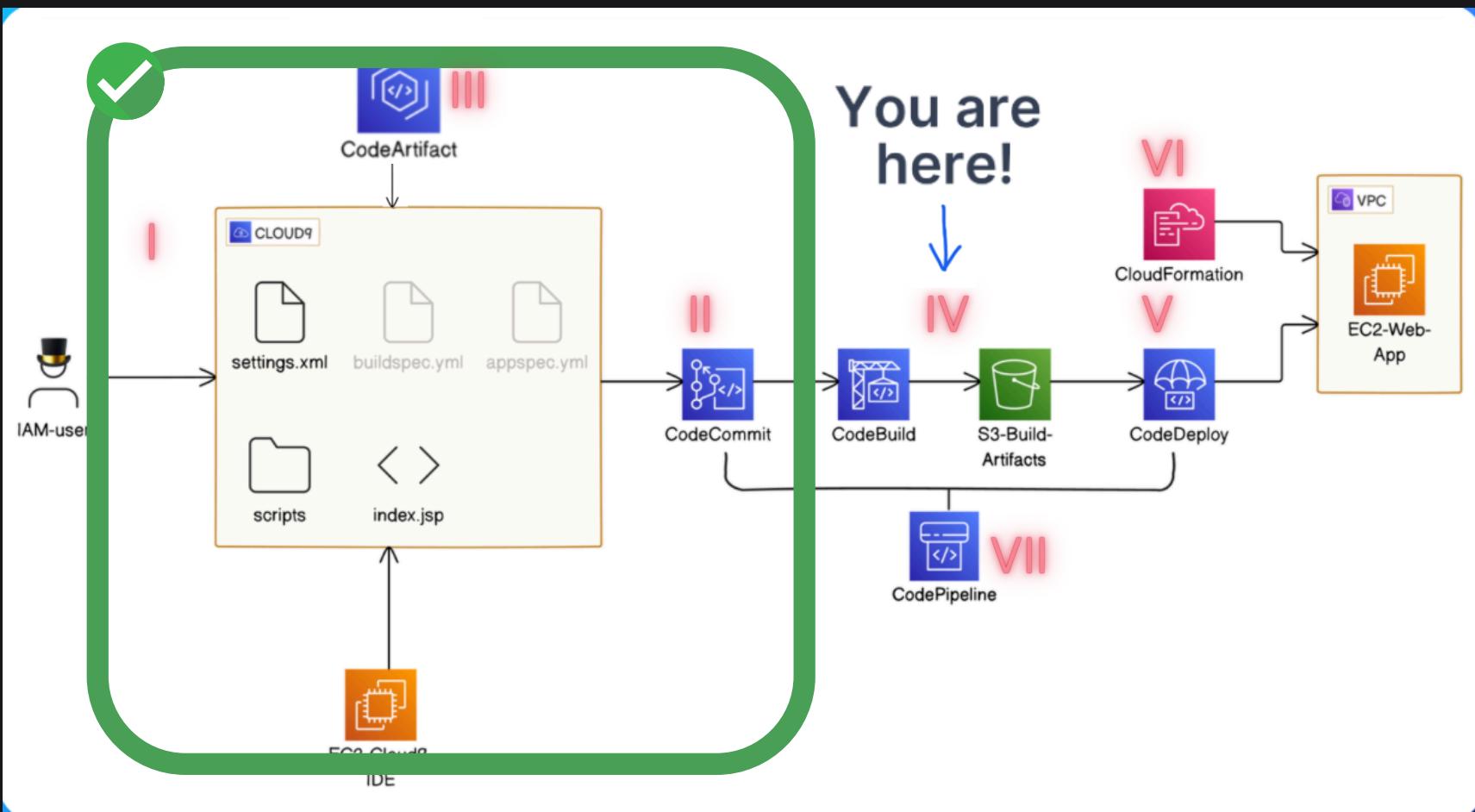


Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

# 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 with AWS CodeDeploy
- VI. Automate with AWS CloudFormation
- VII. CI/CD Pipeline with AWS CodePipeline





Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

# Introducing AWS CodeBuild!

## What it does & how it's useful

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces deployable software packages.

Developers and teams use AWS CodeBuild because it automates the build process, integrates seamlessly with other AWS services, and scales continuously with your project needs.

## How I'm using it in today's project

I'm using AWS CodeBuild in this project to automate the build process for our Java web application. It compiles the source code, and produces a deployable package, streamlining our CI/CD pipeline.

## This project took me...

This project took me about 40 minutes to complete, including setting up AWS CodeBuild. Documentation took me around 30 minutes to write and polish, ensuring it's clear and comprehensive for future reference.



# Set up an S3 bucket

- I started my project by creating an S3 bucket because it serves as a centralized storage for build artifacts, ensuring easy access and management.
- The key artifact that this S3 bucket will capture is called **Web Application Resource, WAR file**.
- This file is important because it contains the compiled and packaged application ready for deployment.

My S3 bucket!

The screenshot shows the AWS S3 console interface. At the top, there are tabs for "General purpose buckets" (which is selected) and "Directory buckets". Below the tabs, a message says "Buckets are containers for data stored in S3." There is a search bar labeled "Find buckets by name". On the right side, there are buttons for "Create bucket", "Empty", "Delete", and "Copy ARN". The main table lists two buckets:

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">cf-templates-1xs10l1ozn05o-ap-southeast-2</a>	Asia Pacific (Sydney) ap-southeast-2	<a href="#">View analyzer for ap-southeast-2</a>	July 4, 2024, 14:27:20 (UTC+08:00)
<a href="#">nextwork-build-artifacts-dahrihadri</a>	Asia Pacific (Sydney) ap-southeast-2	<a href="#">View analyzer for ap-southeast-2</a>	July 11, 2024, 15:07:57 (UTC+08:00)



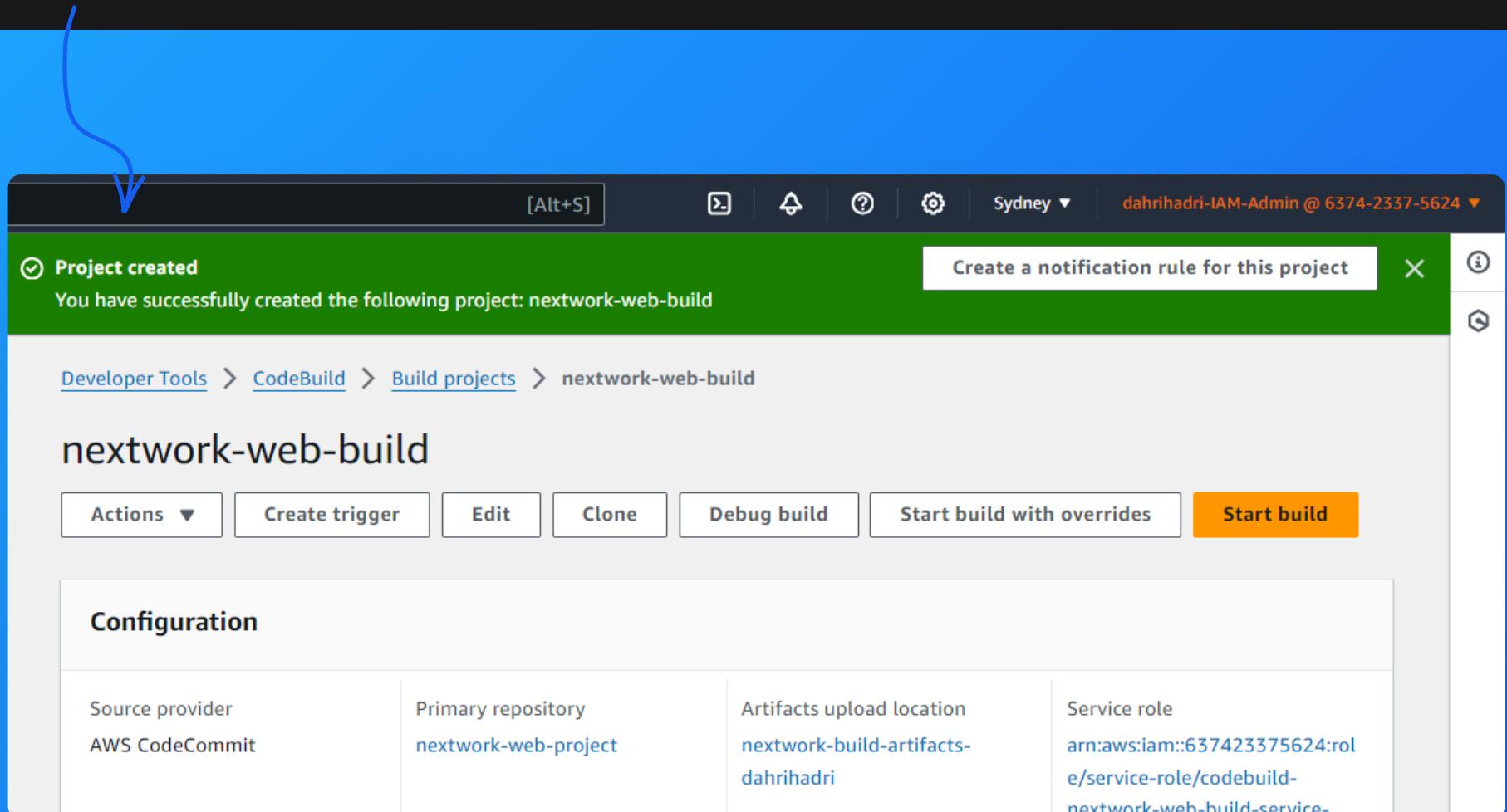
Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

# Set up a CodeBuild project

When creating a project in CodeBuild, there were 5 key configurations I set up:

1. **Source**, which means where the project's code or files originate. I chose AWS CodeCommit, my cloud repository for project files.
2. **Environment**, which means the resources and configurations needed to build the web app. I choose an EC2 instance running Amazon Linux 2 with Java Corretto 8 as the runtime environment.
3. **Buildspec**, which are the commands executed during the build process in the terminal. I configured it to "use a buildspec file."
4. **Artifacts**, which means determining where the build outputs will be stored. I selected my previously created S3 bucket to store the build artifacts.
5. **Logs**, which means configuring where the build logs are saved. I selected Amazon CloudWatch to monitor and review build logs in real-time.

My completed project ready for the first build!



The screenshot shows the AWS CodeBuild console with a green header bar indicating a 'Project created' message: 'You have successfully created the following project: nextwork-web-build'. Below the header, the navigation path is 'Developer Tools > CodeBuild > Build projects > nextwork-web-build'. The main title is 'nextwork-web-build'. A row of buttons includes 'Actions', 'Create trigger', 'Edit', 'Clone', 'Debug build', 'Start build with overrides', and a prominent orange 'Start build' button. A 'Configuration' section displays four items: 'Source provider' (AWS CodeCommit), 'Primary repository' (nextwork-web-project), 'Artifacts upload location' (nextwork-build-artifacts-dahrihadri), and 'Service role' (arn:aws:iam::637423375624:role/service-role/codebuild-nextwork-web-build-service). The entire screenshot is set against a blue background.



Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

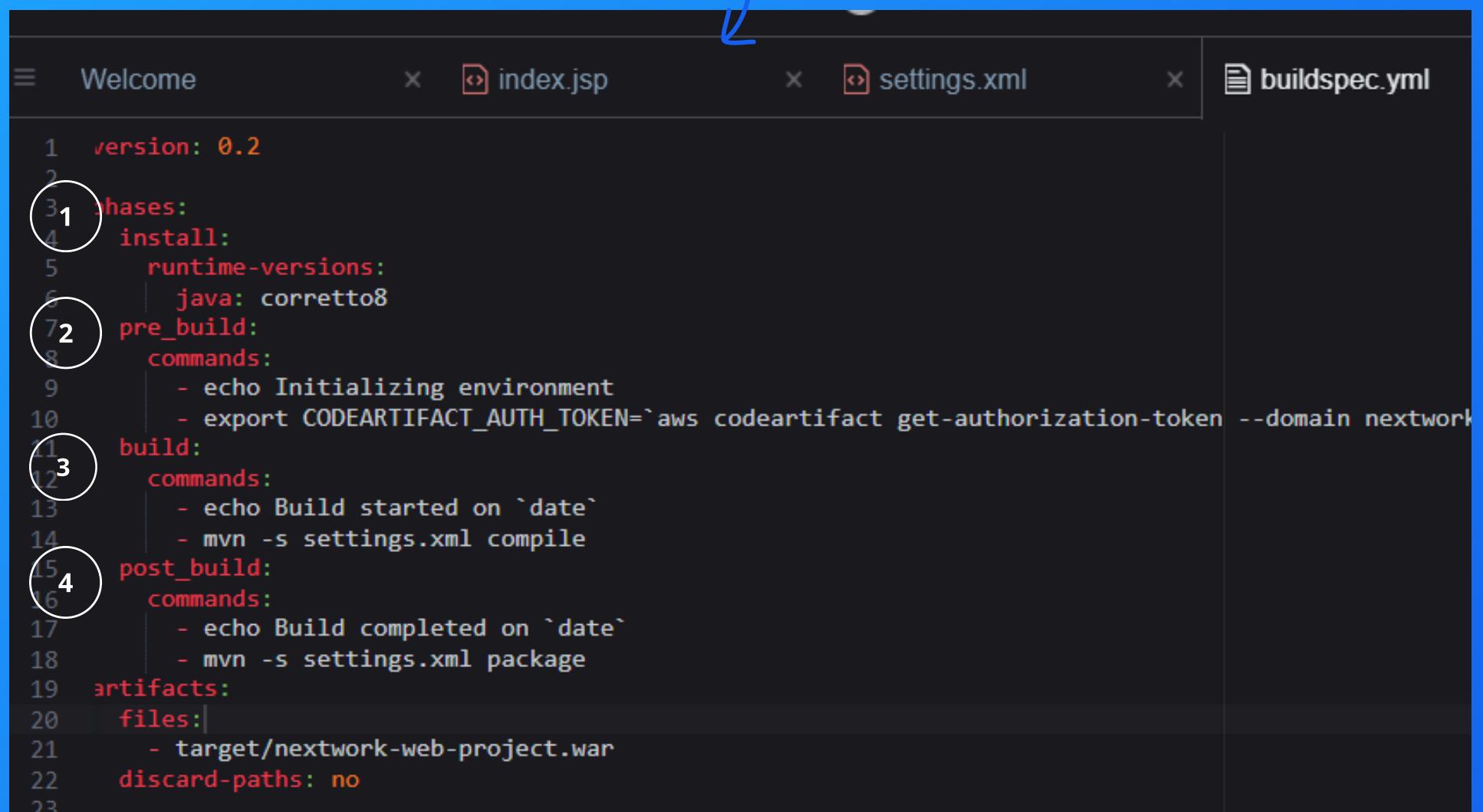
# Create a buildspec.yml file

I created a buildspec.yml file at root repository.

This file contains four phases that tells our build environment what commands to run. These four phases are:

1. **Install:** This phase installs dependencies and sets up the environment. In this case, it specifies the Java runtime version.
2. **Pre-build:** This phase prepares the build environment before the actual build commands are run. It includes initializing the environment and exporting an authorization token for AWS CodeArtifact.
3. **Build:** This is the main phase where the actual build commands are executed. It includes compiling the project.
4. **Post-build:** This phase runs commands after the build, such as packaging the build artifacts.

## A peek into my buildspec.yml



```
1  version: 0.2
2
3  phases:
4    install:
5      runtime-versions:
6        java: corretto8
7    pre_build:
8      commands:
9        - echo Initializing environment
10       - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-token --domain nextwork
11    build:
12      commands:
13        - echo Build started on `date`
14        - mvn -s settings.xml compile
15    post_build:
16      commands:
17        - echo Build completed on `date`
18        - mvn -s settings.xml package
19    artifacts:
20      files:
21        - target/nextwork-web-project.war
22    discard-paths: no
23
```

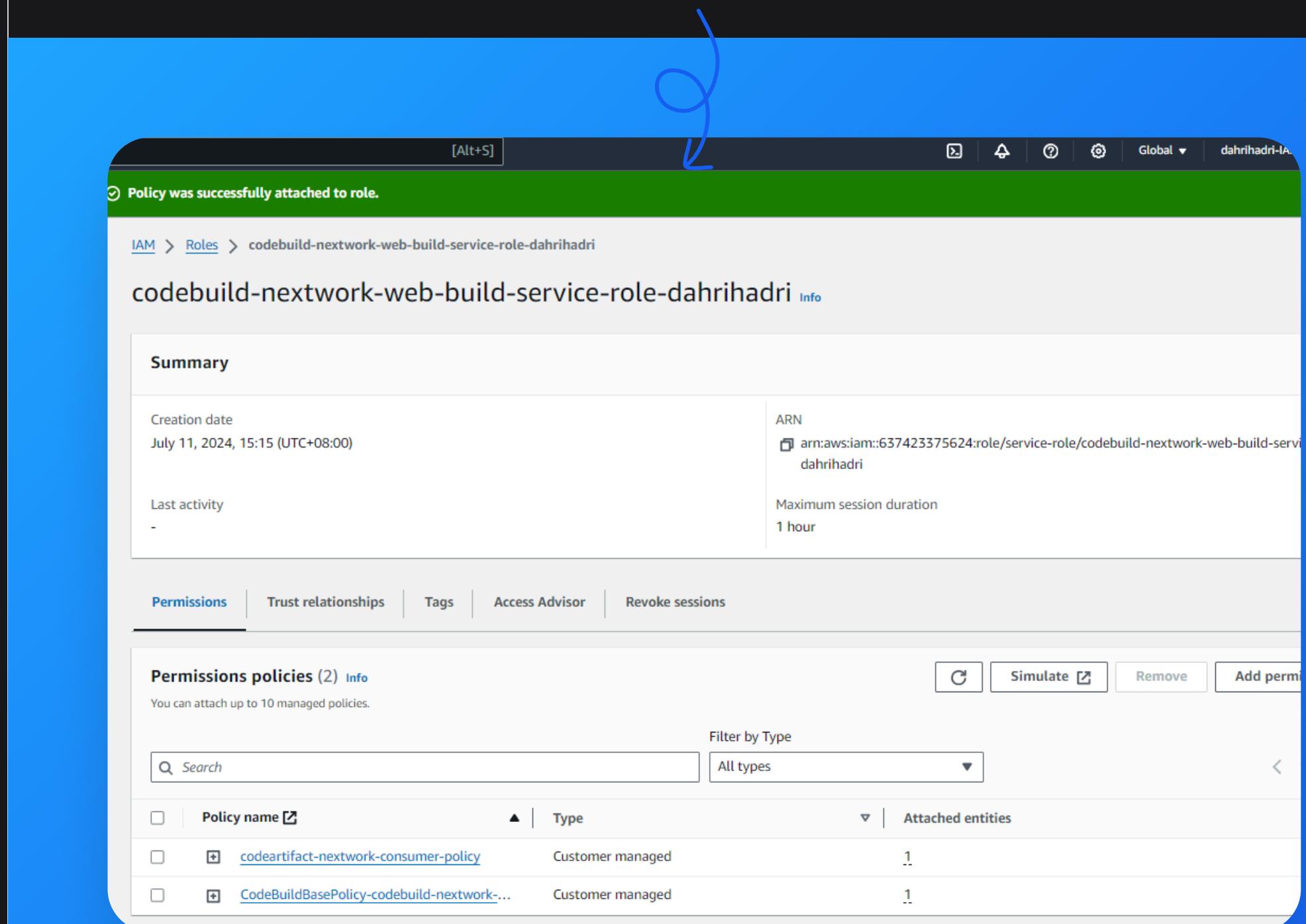


Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

# Edit CodeBuild's IAM role

- Before I start building my web app project (exciting!), I modified my CodeBuild project's service role first. This role was first created when I set up the CodeBuild project.
- I attached a new policy called `codeartifact-nextwork-consumer-policy` to my CodeBuild project's IAM role. This means the CodeBuild service role now has permissions to interact with CodeArtifact repositories, facilitating secure artifact management and dependency retrieval during builds

Updating permission policies for my CodeBuild project's IAM role.



The screenshot shows the AWS IAM Roles page. A green success message at the top states: "Policy was successfully attached to role." Below this, the role name is displayed: "codebuild-nextwork-web-build-service-role-dahrihadri". The "Summary" section provides basic information: Creation date (July 11, 2024, 15:15 (UTC+08:00)), ARN (arn:aws:iam::637423375624:role/service-role/codebuild-nextwork-web-build-service-role-dahrihadri), and Maximum session duration (1 hour). The "Permissions" tab is selected, showing two attached policies: "codeartifact-nextwork-consumer-policy" (Customer managed) and "CodeBuildBasePolicy-codebuild-nextwork-...". Other tabs include "Trust relationships", "Tags", "Access Advisor", and "Revoke sessions".

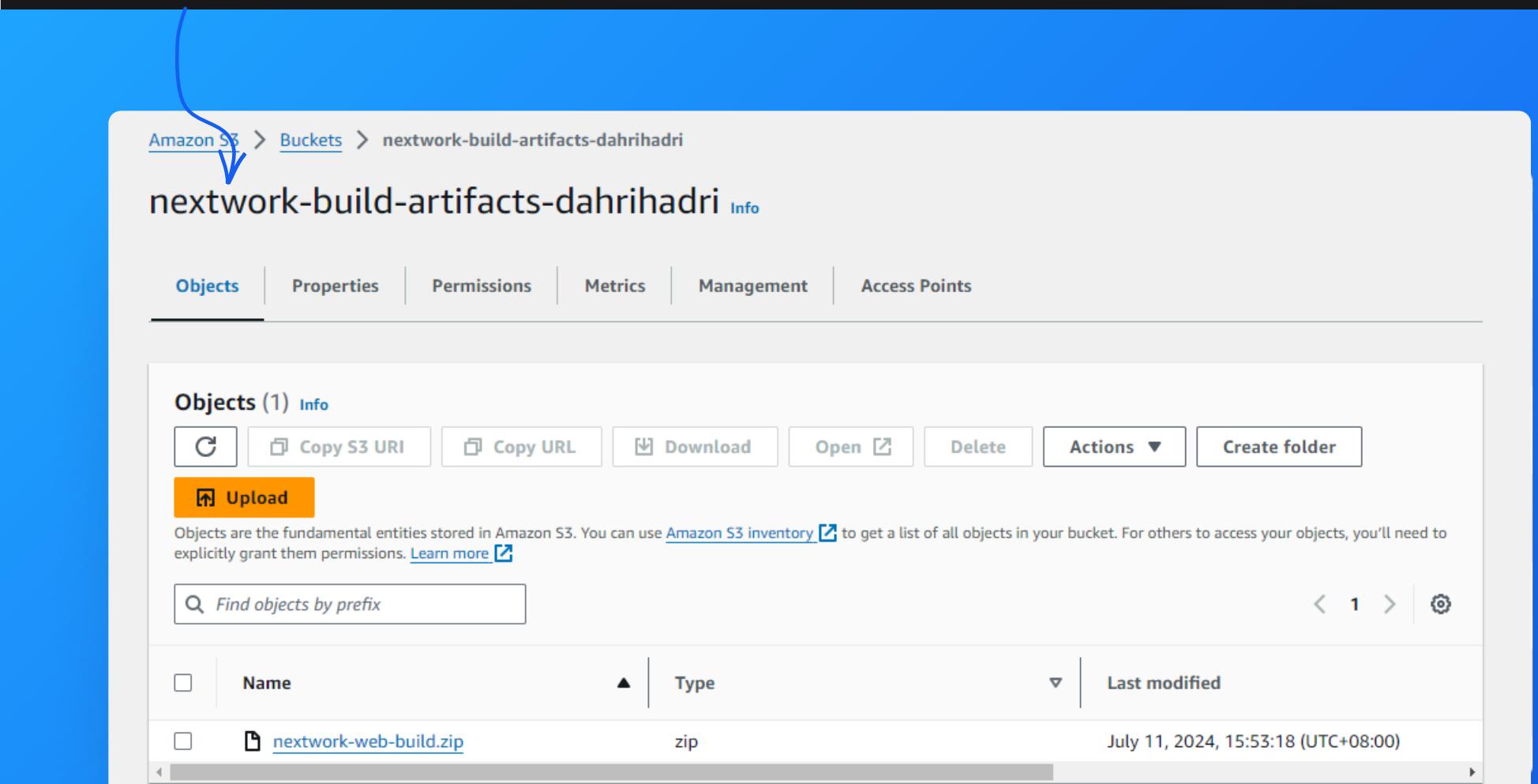


Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

# My first project build

- To build my project, all I had to do was click the "Start build" button in the CodeBuild console
- The build process in CodeBuild took approximately 5 minutes to complete.
- Once the build is complete, I checked my S3 bucket **nextwork-build-artifacts-dahrihadri**. I saw the **nextwork-web-build.zip** file, which verified that the build was completed successfully.

My completed project ready for the first build!



Amazon S3 > Buckets > nextwork-build-artifacts-dahrihadri

nextwork-build-artifacts-dahrihadri [Info](#)

Objects (1) [Info](#)

[C](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions ▾](#) [Create folder](#)

[Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

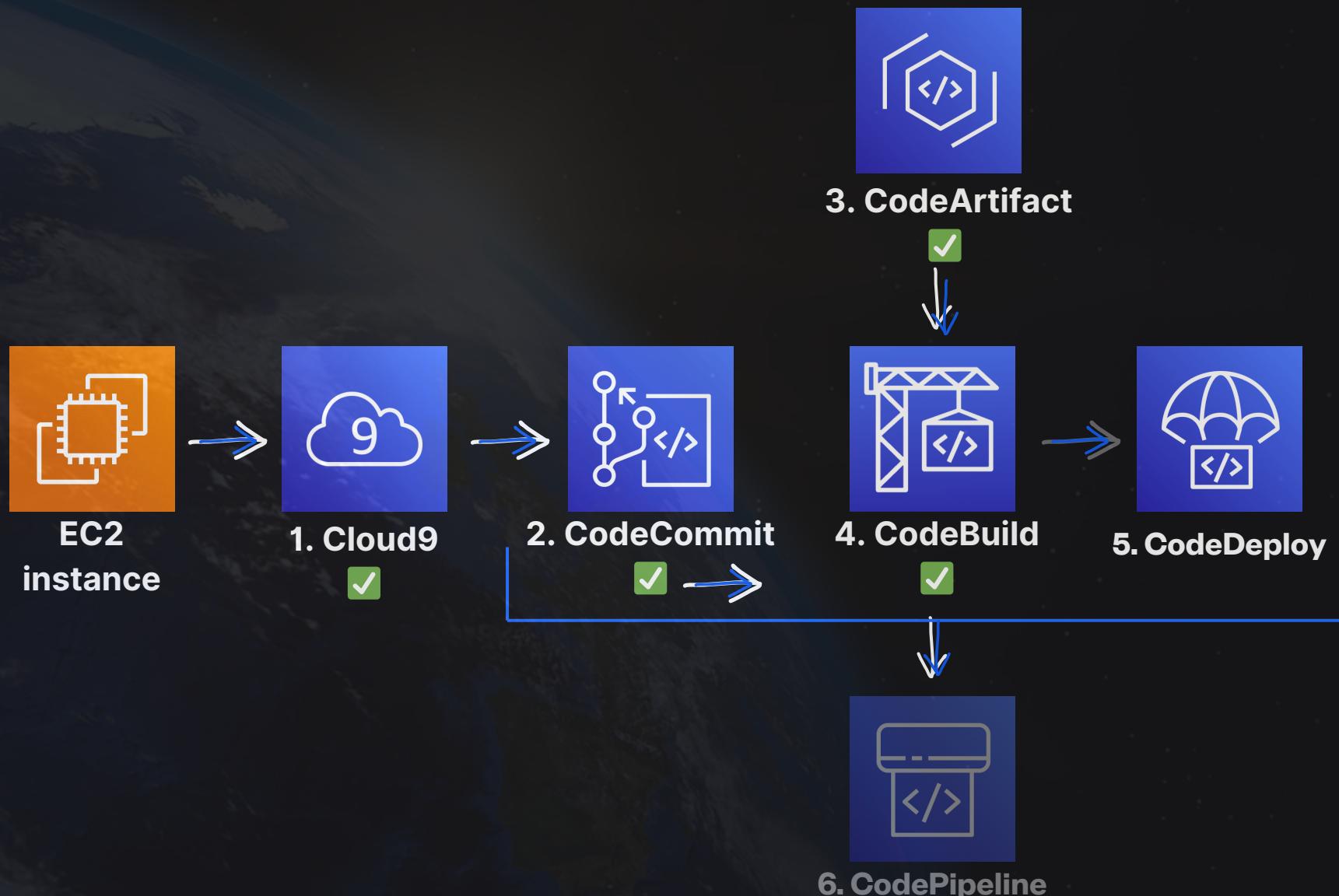
<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	<a href="#">nextwork-web-build.zip</a>	zip	July 11, 2024, 15:53:18 (UTC+08:00)



Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

# 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.





Dahri Hadri  
[linkedin.com/in/dahrihadri](https://linkedin.com/in/dahrihadri)

# My key learnings

1

The build process, or 'building', means compiling source code, running tests, and packaging artifacts into deployable formats. It's essential for ensuring code quality and readiness for deployment in a CI/CD pipeline.

2

The buildspec.yml file defines the build phases and commands for AWS CodeBuild. It controls how CodeBuild runs, including environment setup, build commands, and artifact generation, ensuring consistent and automated build processes.

3

Even though CodeBuild creates a new service role for my build environment, I still have to modify the role's permission policies because the default permissions may not align perfectly with my specific project requirements. Customizing policies ensures that CodeBuild can access necessary resources like S3 buckets, CodeArtifact repositories, or other AWS services essential for my build process without unnecessary restrictions or security risks.

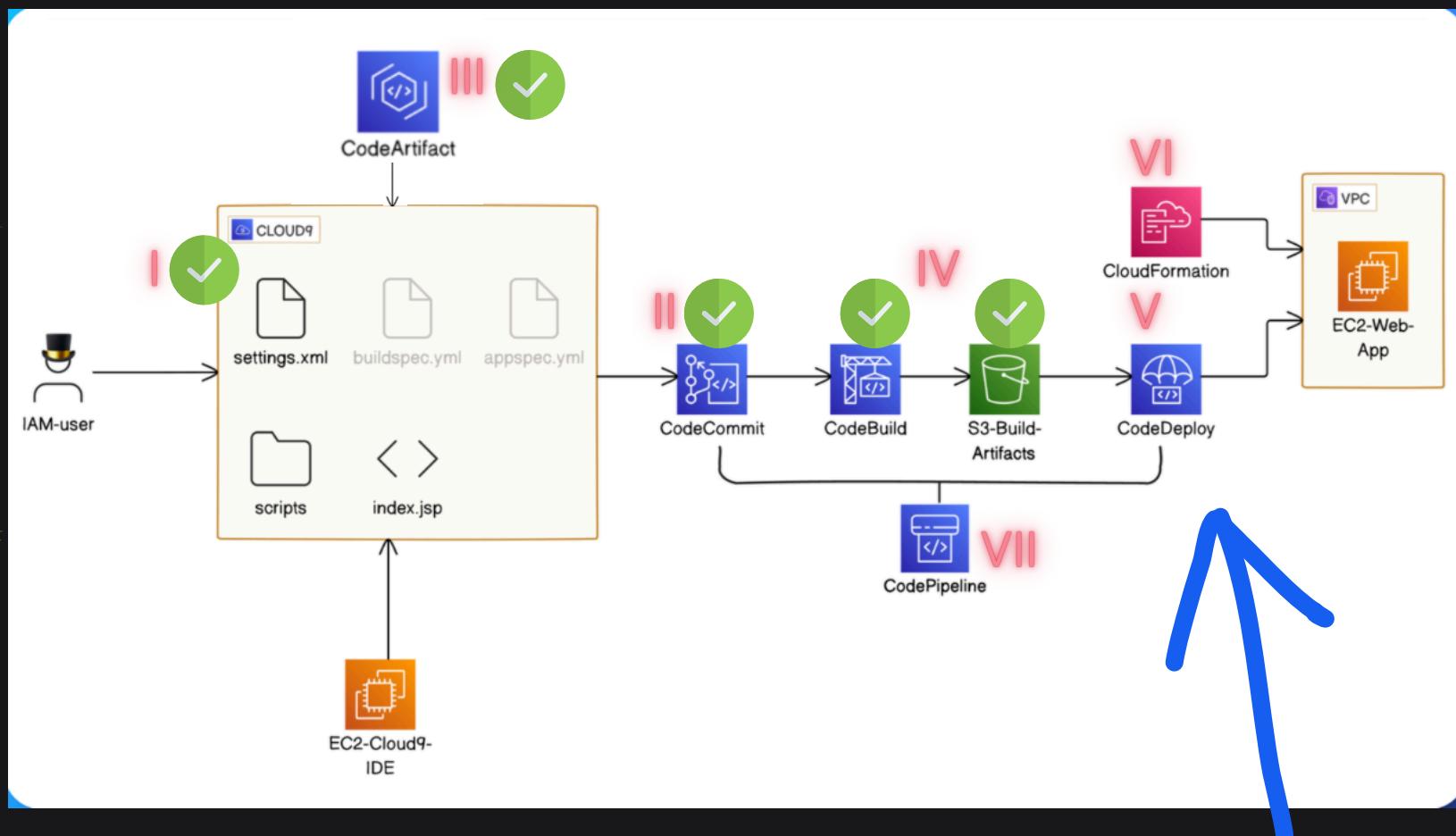
4

One thing I didn't expect was the level of configuration needed for integrating another services like CodeArtifact in my CodeBuild projects.



Dahri Hadri  
linkedin.com/in/dahrihadri

# Great! we are done with series IV



I will build Series V of this CI/CD pipeline—CodeDeploy—in the next project!



NEXTWORK

# Find this helpful?

-  Like this post
-  Leave a comment
-  Save for later
-  Let's connect!



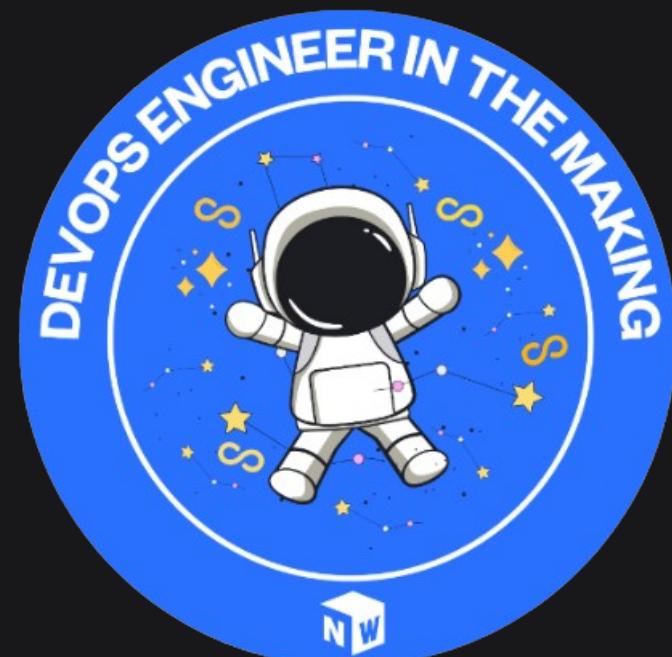
Dahri Hadri



@dahrihadri



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



Ask me about it

