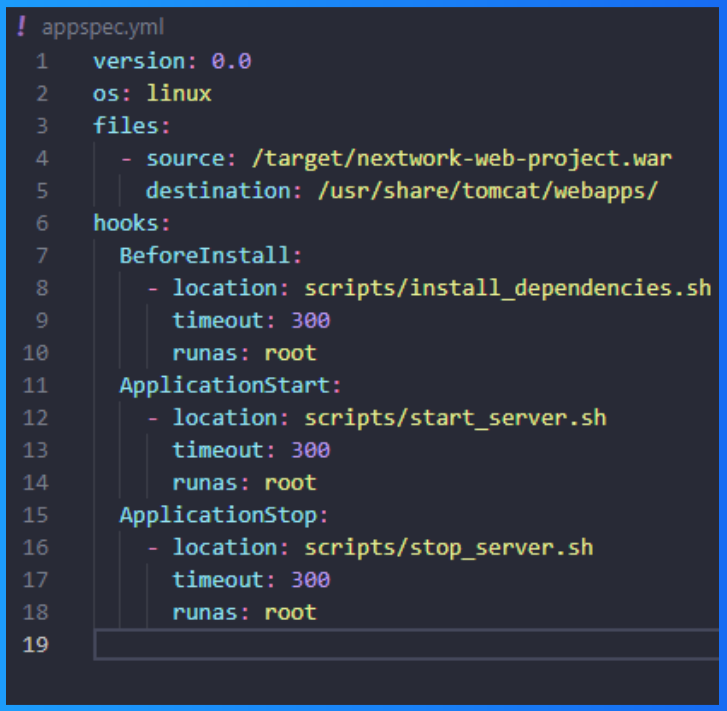
NextWork.org

Deploy an App with CodeDeploy

Gloria



G
Gloria
NextWork Student

Introducing today's project!

What is AWS CodeDeploy?

AWS CodeDeploy automates deploying applications to EC2, on-premises servers, or other platforms. It ensures consistency, supports rollbacks, saves time, and reduces errors. It’s useful for managing scalable deployments and maintaining applications

How I'm using AWS CodeDeploy in this project

I used AWS CodeDeploy to deploy my web app to an EC2 instance. It automated fetching the build artifacts from an S3 bucket, updating the application on the target instance, and ensuring a smooth and consistent deployment process.

One thing I didn't expect...

One thing I didn’t expect in this project was how easy AWS CodeDeploy made things. It worked well with S3 and EC2, handled tasks automatically, and saved time. It showed me how AWS tools make hard jobs simple and fast.

This project took me...

This project took me around 40 minutes to complete. Setting up resources, configuring AWS CodeDeploy, deploying the app, and testing everything went smoothly thanks to the clear instructions and automation provided by AWS.

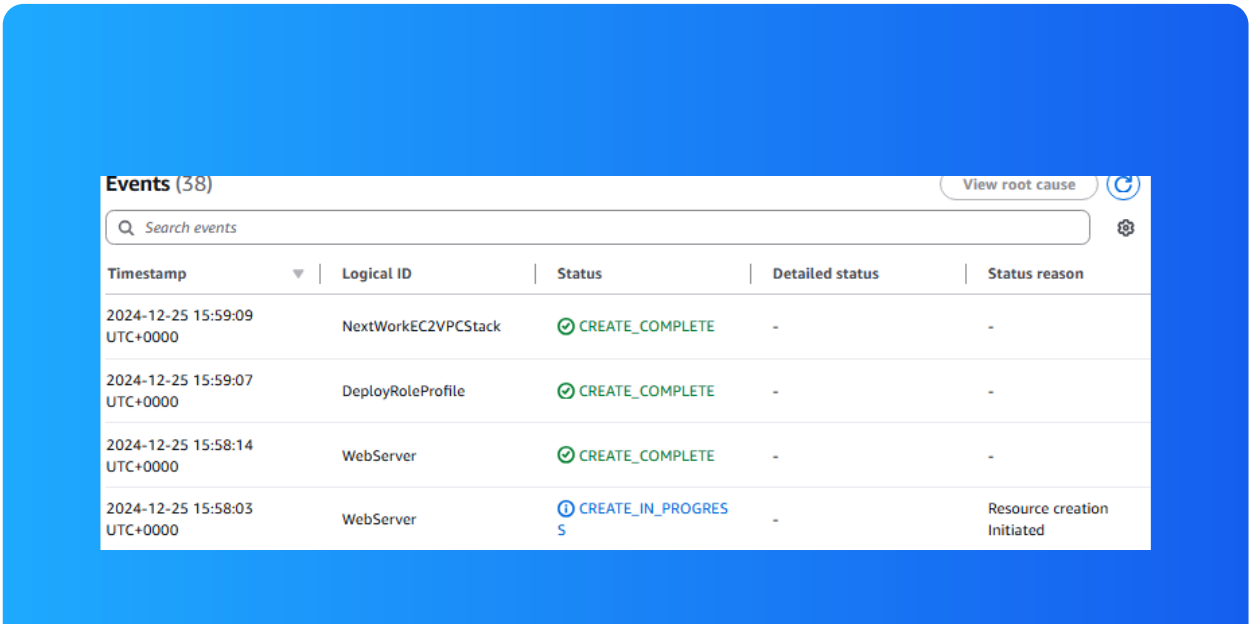
Gloria
NextWork Student

Set up an EC2 instance

I set up an EC2 instance and VPC because they provide a secure, isolated environment to run applications.The EC2 instance acts as a virtual server for hosting the web app, while the VPC ensures network-level security and control and safe environment.

We manage production and development environments separately to avoid risks. Development is for testing and changes, while production is for live users.This separation prevents errors from affecting users, improves performance, and secure operations.

To set up my EC2 instance and VPC, I used AWS CloudFormation. It automated the creation of resources through a template, ensuring everything was configured properly, saving time, and reducing manual errors.



G
Gloria
NextWork Student

Bash scripts

Scripts are files containing a series of commands that automate tasks.Bash is a Unix shell and command language used to execute these scripts or run commands interactively.Together, they help automate processes, simplify workflows, and manage systems

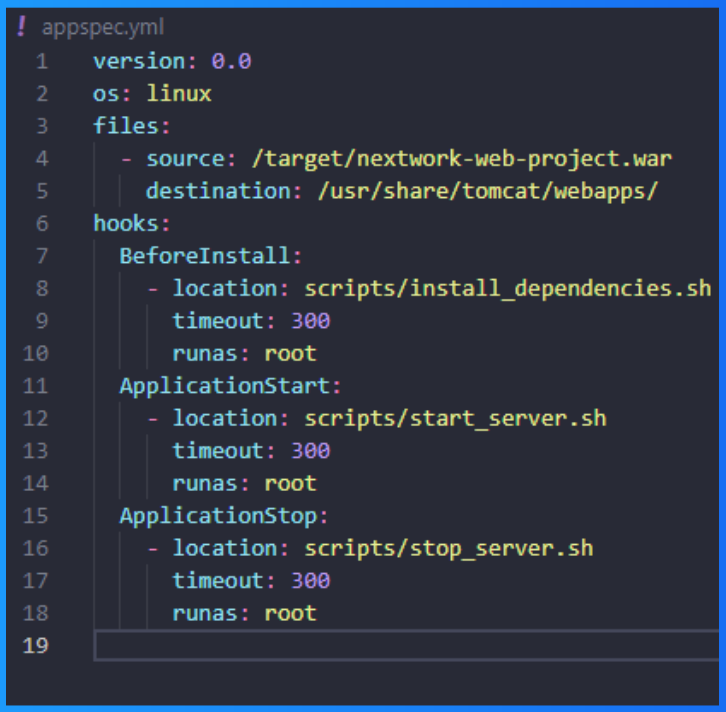
I used three scripts for my project's deployment

The first script I created was a Bash script to set up a Tomcat server and configure an Apache HTTP server as a reverse proxy. It installs Tomcat and Apache, then creates a configuration file to direct traffic to the web app hosted on Tomcat.

The second script I created was to start and enable the Tomcat and Apache HTTP services. It ensures both services are running and set to start automatically on system boot, keeping the web application and reverse proxy operational consistently.

The third script I created was a Bash script to stop running Apache and Tomcat services. It checks if each service is running using `pgrep` and stops them if found active. This script helps ensure proper shutdown before maintenance or redeployment.

Bash scripts

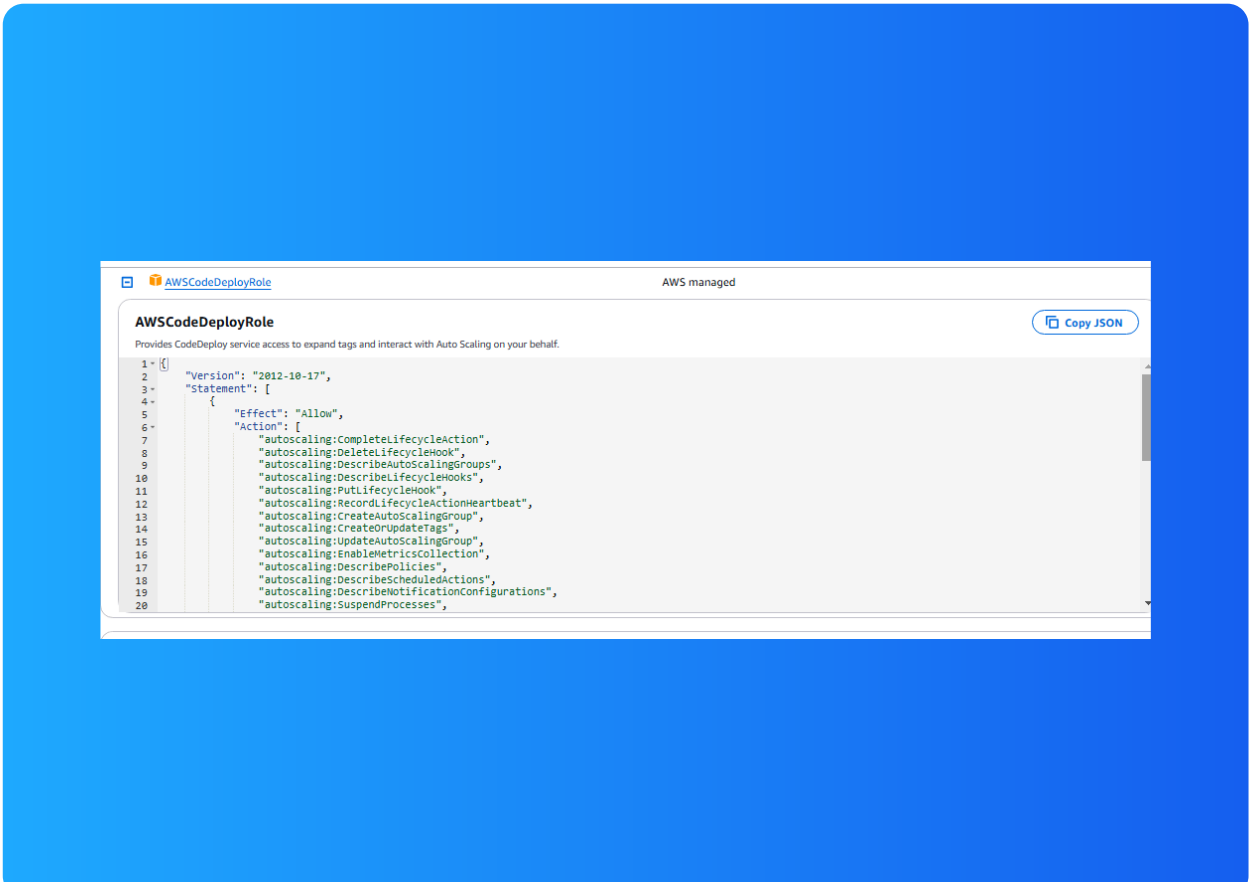


Gloria
NextWork Student

CodeDeploy's IAM Role

I created an IAM service role for CodeDeploy because it needs permissions to deploy to EC2. The AWSCodeDeployRole policy grants these permissions securely, saving time by providing a ready-made policy instead of manually defining permissions.

To set up CodeDeploy's IAM role, I chose CodeDeploy as the service, selected its use case, and attached the AWSCodeDeployRole policy. This predefined policy allows CodeDeploy to securely manage EC2 instances for deployments without manual permission.



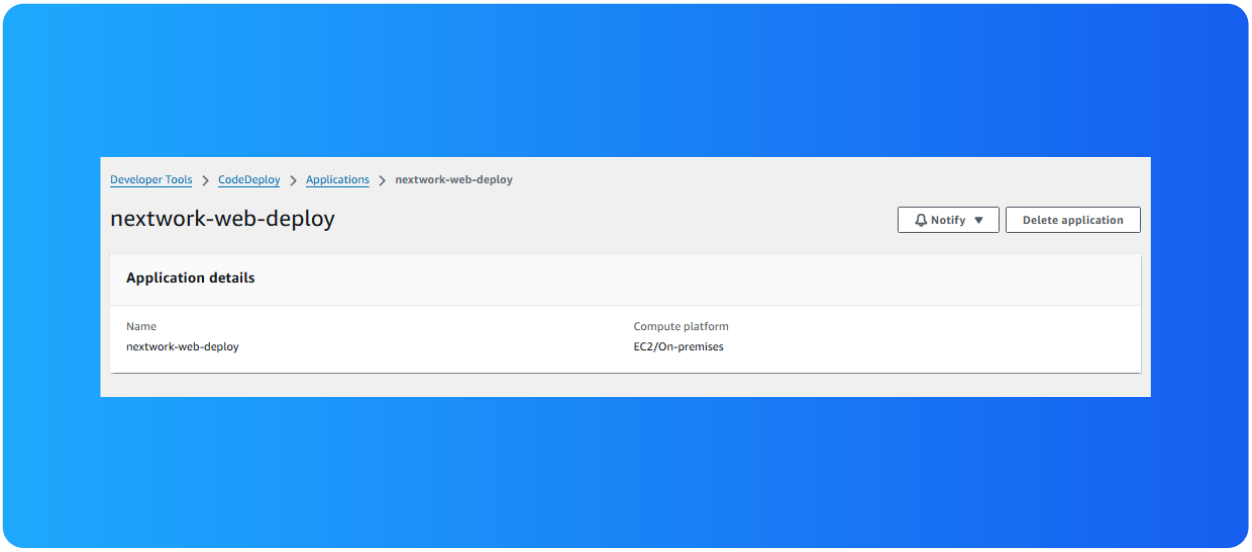
Gloria
NextWork Student

CodeDeploy application

A CodeDeploy application means a resource that acts as a container for deployment settings and configurations. It organizes the deployment process, linking target instances and configurations to ensure updates are deployed correctly and efficiently.

To create a CodeDeploy application, I had to select a compute platform, which means choosing the environment for deployment, such as EC2/On-Premises, Lambda, or ECS. This defines where CodeDeploy manages and deploys updates.

The compute platform I chose was EC2/On-premises because it gives control over virtual servers to set up, configure, and deploy web apps. It's great for learning how to manage servers, ensuring flexibility and a deeper understanding of deployment.



Gloria
NextWork Student

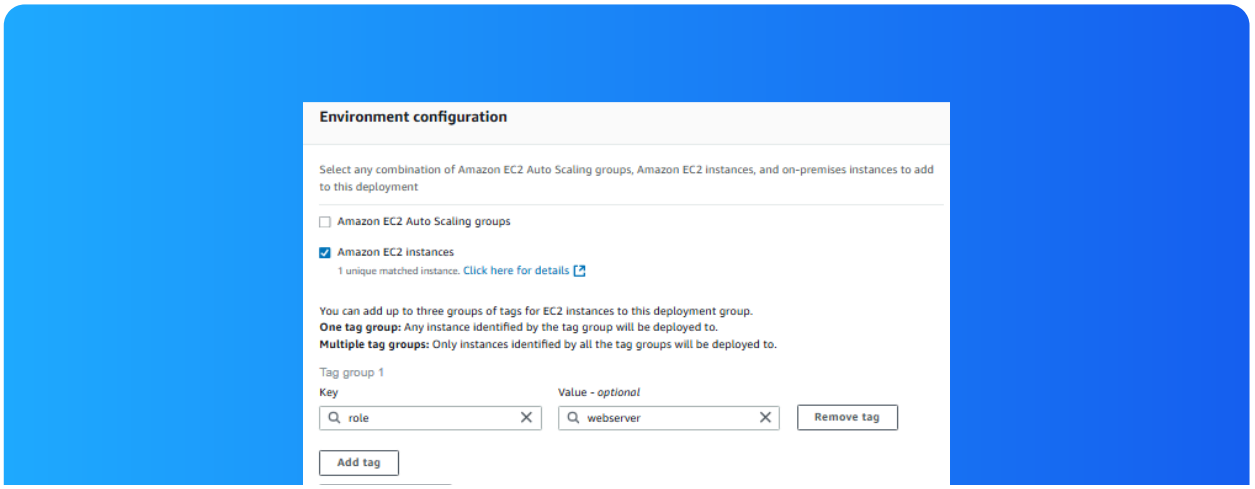
Deployment group

A deployment group means a collection of settings and resources in CodeDeploy that specifies where and how to deploy your application. It includes the target instances, service roles, tags, deployment type, and settings to manage the deployments.

Two key configurations for a deployment group

Environment means the servers or resources where your application will run.In CodeDeploy, it defines the type of compute platform, like EC2 instances or containers, and specifies the setup for deploying and managing your application effectively.

A CodeDeploy Agent is software on EC2 instances that lets them communicate with CodeDeploy. It receives instructions, executes deployments, and ensures the app runs smoothly. Without it, CodeDeploy can’t manage deployments on the instance.



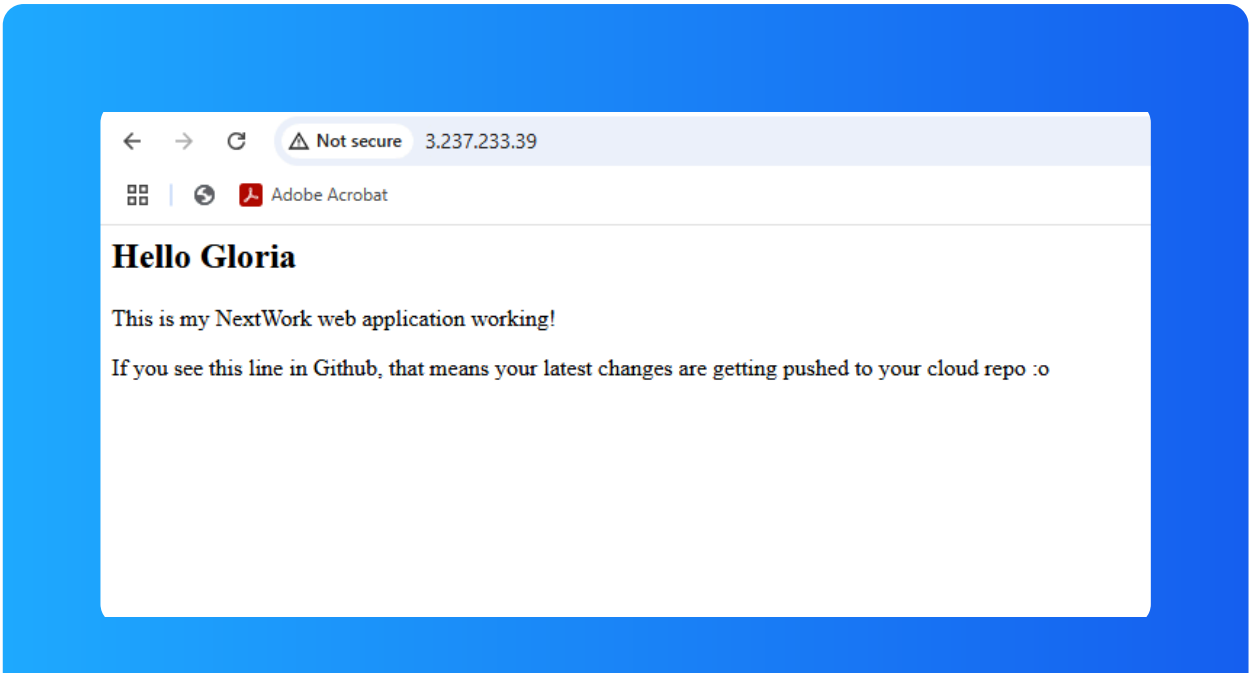
Gloria
NextWork Student

CodeDeploy application

To create my deployment, I had to set up a revision location, which means specifying where the application files are stored. It could be an S3 bucket, GitHub, or Bitbucket. CodeDeploy uses this location to fetch the files needed for deployment.

My revision location was my S3 bucket URI containing the zip file of my web app's build artifacts. CodeDeploy used this URI to fetch the application files for deployment to the target EC2 instance.

To visit my web app, I had to visit my EC2 console, select the WebServer EC2 instance, and open its public address. I then modified the URL from https to http to access the web app correctly.



Everyone should be in a job they love.

Check out nextwork.org for more projects