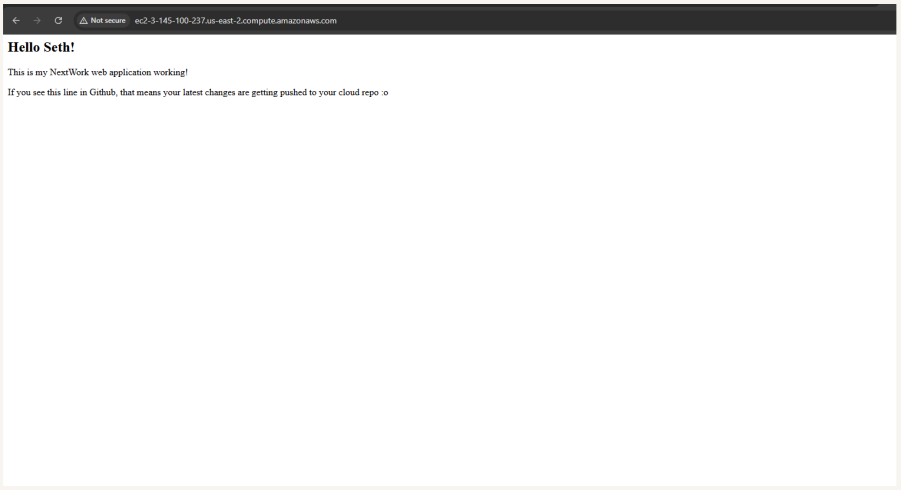
Deploy a Web App with CodeDeploy

Seth Sekyere



Seth Sekyere

Introducing Today's Project!

In this project, I will demonstrate how to automate deployments using AWS CodeDeploy. I’m doing this project to learn how to release applications reliably, minimize downtime, and integrate deployment into a CI/CD pipeline.

Key tools and concepts

Services I used were AWS CodeDeploy, EC2, S3, and CloudFormation. Key concepts I learnt include continuous deployment, deployment scripts, appspec.yml, lifecycle hooks, Infrastructure as Code, and automating application rollouts.

Project reflection

This project took me approximately 2 hours. The most challenging part was writing and configuring the deployment scripts correctly. It was most rewarding to see my web app successfully deployed and running on the EC2 instance.

This project is part five of a DevOps series where I'm building a CI/CD pipeline! I'll be working on the next project tonight to continue learning deployment strategies and automation.

Seth Sekyere

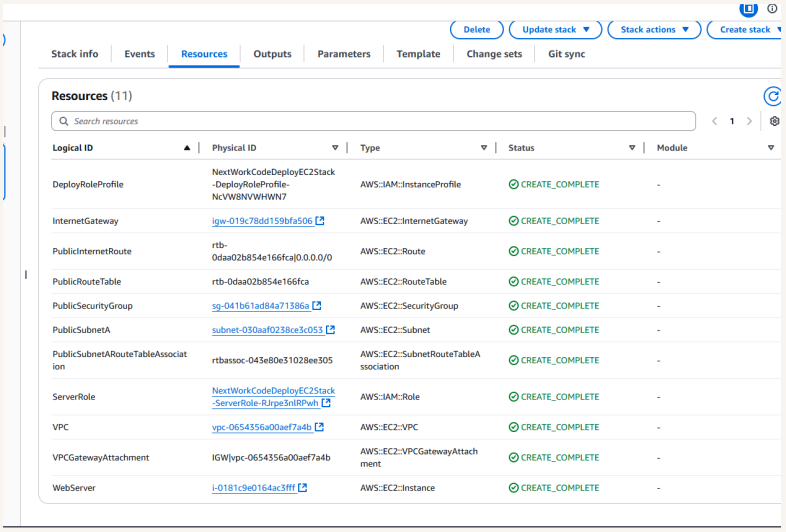
Deployment Environment

To set up for CodeDeploy, I launched an EC2 instance and VPC because the EC2 will host my web app and the VPC provides the secure networking environment it needs for deployment and access.

Instead of launching these resources manually, I used AWS CloudFormation. When I need to delete these resources, I can simply delete the stack, and CloudFormation removes everything automatically.

Other resources created in this template include a subnet, route tables, an internet gateway, and a security group. They're included to provide networking, routing, internet access, and firewall rules for the EC2 instance.

Seth Sekyere



Seth Sekyere

Deployment Scripts

Scripts are text files containing commands that automate tasks. To set up CodeDeploy, I also wrote scripts to install dependencies, start and stop the server, and configure the environment for deploying my web app.

The `install_dependencies.sh` script will install all software needed to run the web app (Tomcat and Apache) and configure Apache as a reverse proxy so it forwards web traffic to Tomcat.

The `start_server.sh` script will start Tomcat and Apache services and configure them to automatically run on server reboot, ensuring the web application stays up and accessible.

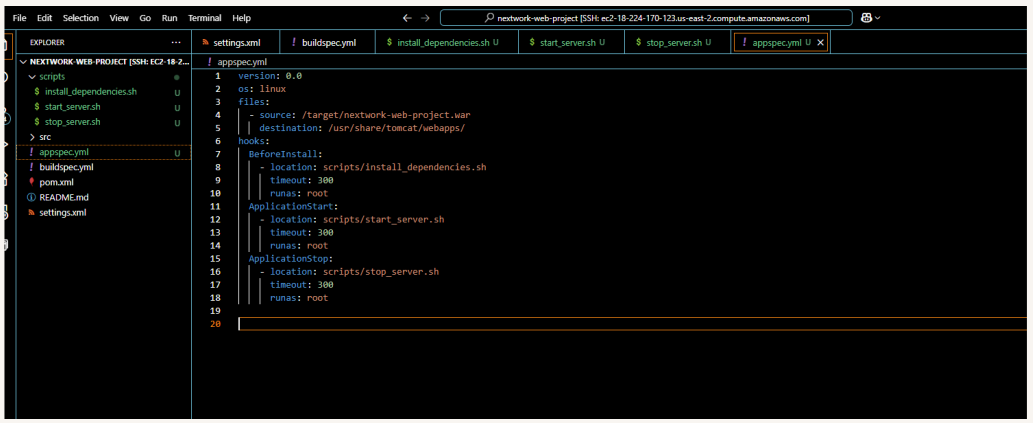
The `stop_server.sh` script will check if Apache (`httpd`) and Tomcat are running, and if they are, it will stop their services. This ensures the web application and its server processes are safely shut down.

Seth Sekyere

appspec.yml

Then, I wrote an `appspec.yml` file to tell CodeDeploy how to deploy my application. The key sections in `appspec.yml` are version, os, files, and hooks, which define what files to copy and which scripts to run at each deployment stage

I also updated buildspec.yml because I needed to include the appspec.yml file and the scripts folder as artifacts, so CodeDeploy would have all the files required to deploy my application correctly.



Seth Sekyere

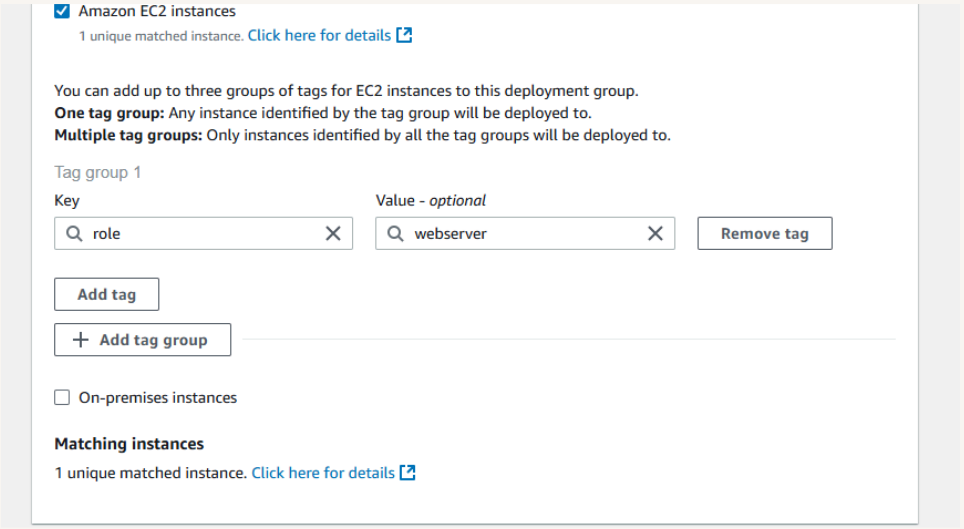
Setting Up CodeDeploy

A deployment group is the set of instances where the app is deployed with specific settings. A CodeDeploy application is the main container that organizes all its deployment groups and revisions.

To set up a deployment group, you also need to create an IAM role to give CodeDeploy permission to access EC2 instances, read artifacts from S3, and write logs to CloudWatch.

Tags are helpful for identifying and organizing resources. I used the tag `Environment=Production` to tell CodeDeploy which EC2 instance should receive the deployment.

Seth Sekyere

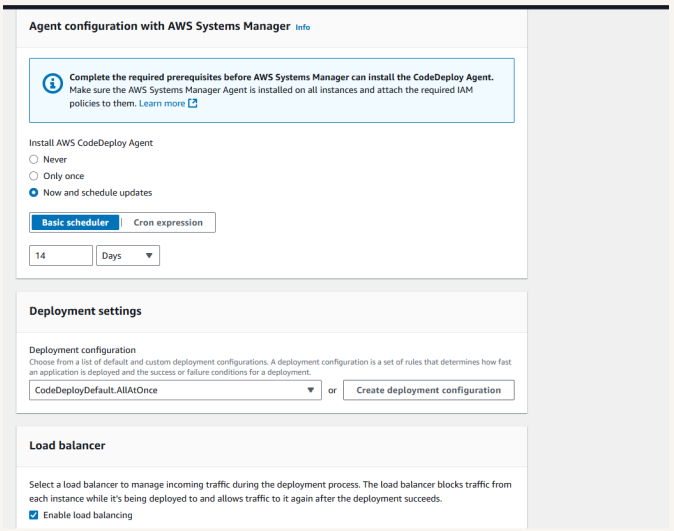


Seth Sekyere

Deployment configurations

Another key setting is the deployment configuration, which affects how quickly and safely your app is rolled out. I used CodeDeployDefault.AllAtOnce, so the app deploys to all instances at once—fast but risky for multiple instances.

In order to connect CodeDeploy with an EC2 instance, a CodeDeploy Agent is installed on the instance. It receives deployment instructions from CodeDeploy and executes them, ensuring the application is deployed correctly.



Seth Sekyere

Success!

A CodeDeploy deployment is a single rollout of your application, specifying which revision to deploy and how. The difference to a deployment group is that the group defines where and how deployments happen, while a deployment is an actual execution

I had to configure a revision location, which means telling CodeDeploy where to find the build artifacts for deployment. My revision location was the S3 bucket storing the WAR/zip file of my web app.

To check that the deployment was a success, I visited the public URL of my EC2 instance. I saw my web app running correctly, confirming the WAR file was deployed and the servers were up.

Seth Sekyere