



nextwork.org

Deploy a Web App with CodeDeploy



Joseph Cole-Showers

A screenshot of a web browser window. The address bar shows a non-secure connection to 'ec2-18-171-171-21...'. The page content displays a 'Hello Joseph!' message and a note about GitHub pushes.

Hello Joseph!

This is my NextWork web application working!

If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :c

Introducing Today's Project!

In this project, I will demonstrate how to use CodeDeploy to deploy a web app.

Key tools and concepts

Services I used include: CodeDeploy, CodeBuild, CodeArtifact, IAM, EC2, CloudFormation, S3, CodeConnection.

Project reflection

This project took me approximately 3hrs. The most challenging part was understanding the deployment group and app creation.

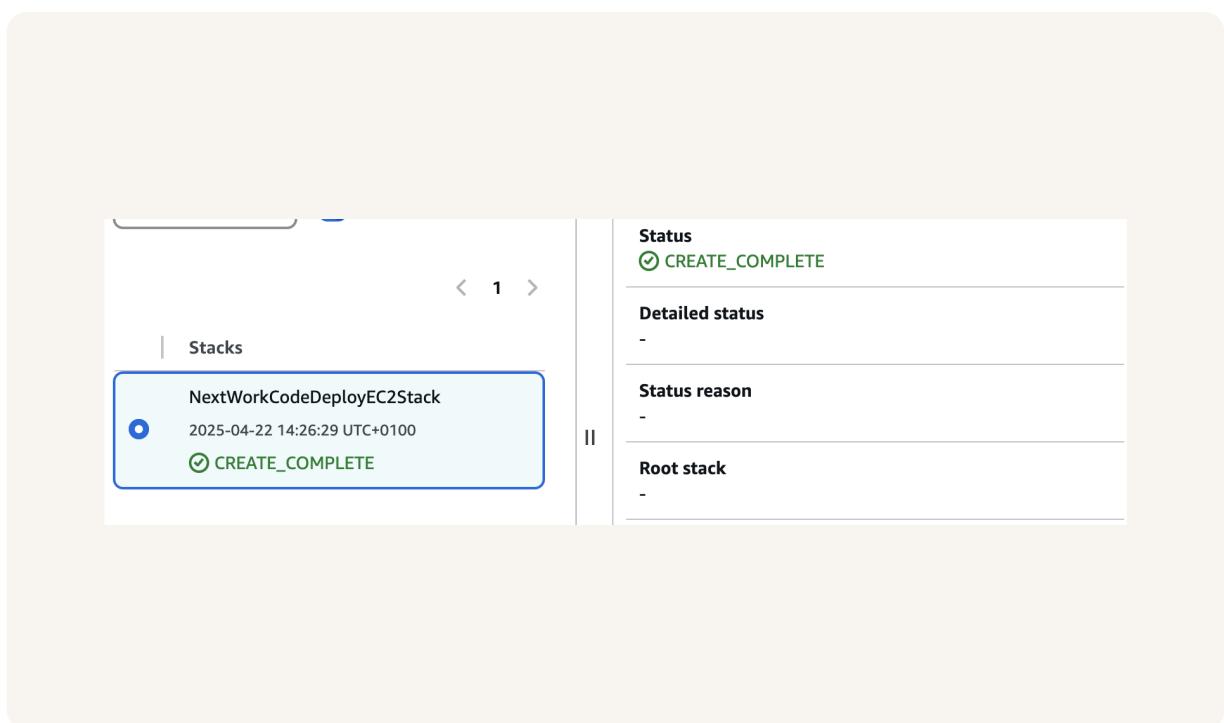
This project is part five of a series of DevOps projects where I'm building a CI/CD pipeline.

Deployment Environment

To set up for CodeDeploy I launched an EC2 instance and VPC because they will make up my production env. I need to separate the dev env where I'm writing the code with the production env, where I'm deploying the webapp.

Instead of launching these resources manually, I used CloudFormation. When I need to delete these resources created by CF, I can simply delete the entire stack, this will automatically delete all the resources inside the stack.

Other resources created in this template include a VPC, RouteTable, SecurityGroups, Subnet.



Deployment Scripts

Scripts are mini programs that helps to automate the running of commands. To setup CodeDeploy I also wrote scripts to automate the deployment commands. i.e Commands that the deployment ec2 instance needs to run in order to host the web app.

install_dependencies will help my EC2 instance install all the dependencies it needs to host a web app.

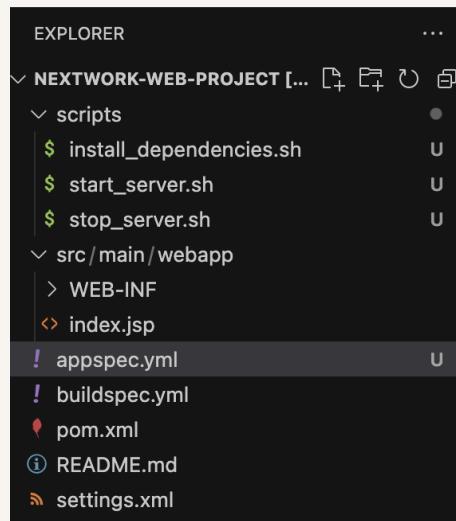
start_server.sh is another script that will start up the 2 servers that I'm using in the EC2 instance.

stop_server.sh will safely stop web server services by first checking if they're running.

appspec.yml

Then, I wrote an appspec.yml file to give CodeDeploy the instructions on how to deploy the web app. The key sections in appspec.yml are - beforeInstall, ApplicationStart, ApplicationStop

I also updated buildspec.yml to tell CodeBuild that it should also package up the new appspec.yml and setup scripts that we generated inside the build artifact (i.e the compressed war file).



Setting Up CodeDeploy

A deployment group is a group of EC2 instances that I can deploy to and also the collection of settings that determine how I want to deploy the webapp. A CodeDeploy app is a folder that holds together all the deployment groups for the same apps.

To set up a deployment group, you also need to create an IAM role to give CodeDeploy the permission to access the EC2 instances it needs. Otherwise, CodeDeploy doesn't have access to EC2 - not able to give instances the instructions to deploy the web

Tags are helpful for identifying the instances that will be deploying the web app. I used the tag role -> webserver, to automatically match the EC2 instance I deployed with the CloudFormation template.

Tag group 1

Key

X

Value - optional

X

Remove tag

Add tag

+ Add tag group

On-premises instances

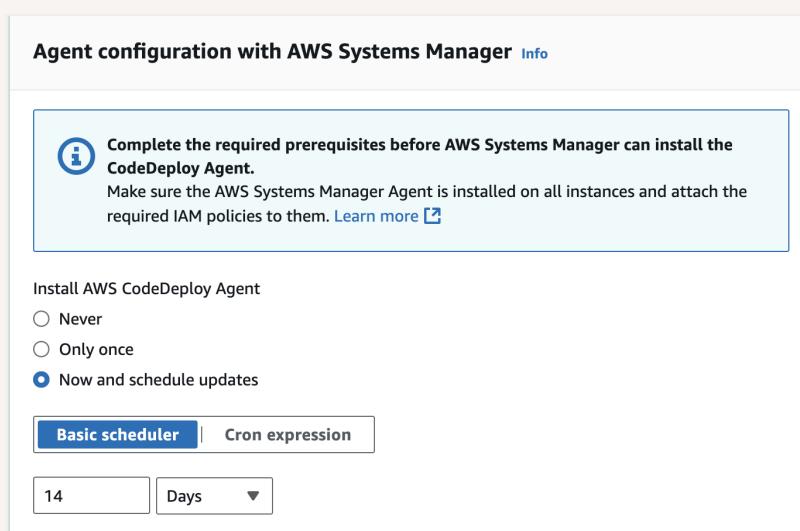
Matching instances

1 unique matched instance. [Click here for details](#) 

Deployment configurations

Another key setting is the deployment configuration, which affects how quickly and risky I'm deploying the web app. I used CodeDeployDefault.AllAtOnce, so any changes codedeploy deploys affects all instances in the deployment group at once.

In order to connect the deployment instance with Codedeploy a CodeDeploy Agent is set up to receive instructions from CodeDeploy and make sure that the commands in appspec.yml runs.



Success!

A CodeDeploy deployment is a specific update to the app that I'm deploying to users. The difference to a deployment group is that the group is like a settings file and the deployment is like a specific update we make using that settings file.

I had to configure a revision location, which means where the webapps war file (i.e the compressed file) ready to deploy, where it lives. My revision location was the S3 bucket I created and linked with Codebuild.

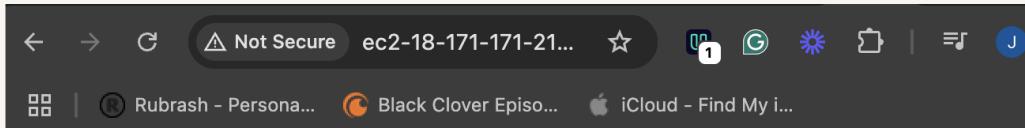
To check that the deployment was a success, I visited the ipv4 dns address of the EC2 instance. I saw a live web app that is working and serving the web app code to end users.

J

Joseph Cole-Showers

NextWork Student

nextwork.org



Hello Joseph!

This is my NextWork web application working!

If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :c



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

