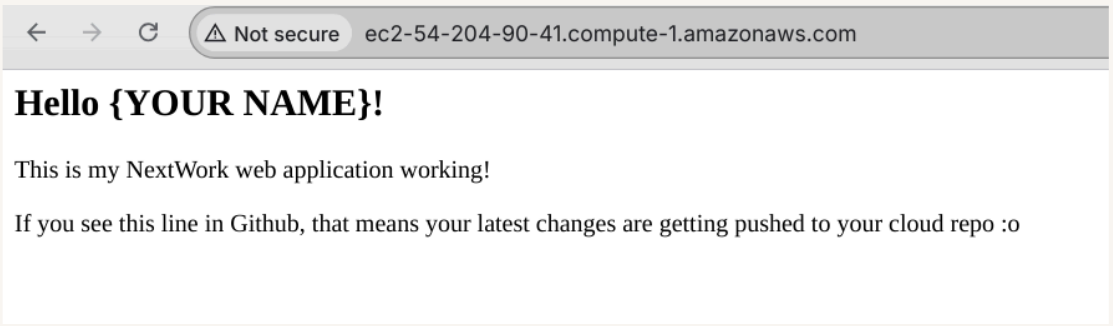
Deploy a Web App with CodeDeploy

Shadrack Kalukwo



S
Shadrack Kalukwo
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to use code deploy to deploy a web app. We are doing this project to learn how deployment works and how it can be automated using a combination of code deploy and deployment scripts. We'll se a live website.

Key tools and concepts

Services I used were ec2, CodeDeploy, CodeBuild, CodeArtifact, IAM, S3, CloudFormation, CodeConnection, VSCode, GitHub. Key concepts I learnt include; deployment, deployment groups, appspec, deploymentscripts and the importance of rebuilding project.

Project reflection

This project took me approximately 3 hours including troubleshooting. The most challenging part was understanding the separation of the deployment instance from the development instance.It was most rewarding to see the deployed web app.

This project is part five of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow

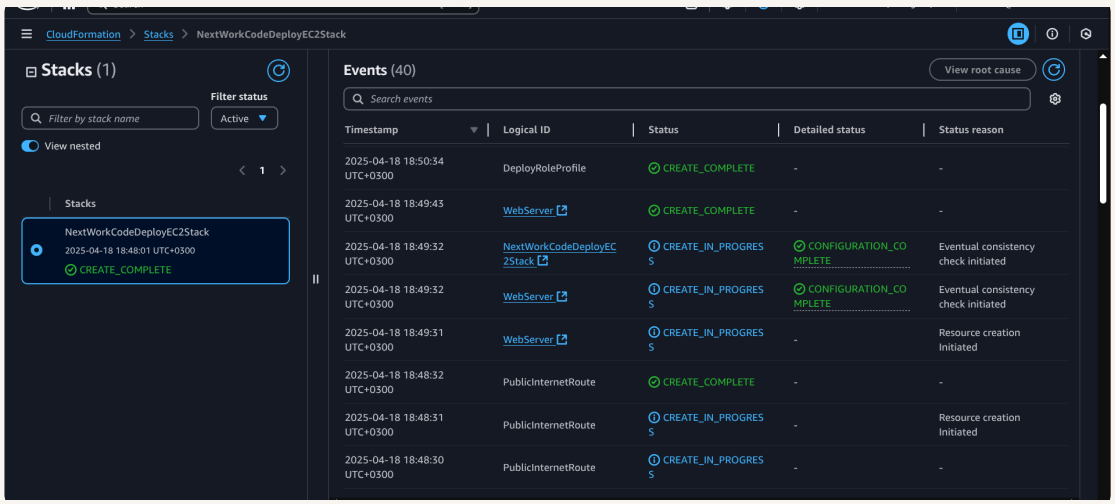
Shadrack Kalukwo
NextWork Student

Deployment Environment

To set up for CodeDeploy, I launched an EC2 instance and VPC because they will make up our production environment. We need to separate our development environment(writing code) and our production environment (deploying code)

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

Other resources created in this template include VPC, ec2, route tables, igw, IAM roles. They're also in the template because of setting up the networking environment of our deployment instance.



S
Shadrack Kalukwo
NextWork Student

Deployment Scripts

Scripts are mini programs that automates the running of commands. To set up CodeDeploy, I also wrote scripts to automate deployment commands. i.e These are the commands that our deployment ec2 instance needs to run inorder to host our web app.

Install_dependencies will help our ec2 instance, install the dependencies it needs to host the web app like tomcat the webserver.

start_server.sh will start up the two servers that we've installed in our ec2 instance. 1 it starts up tomcat responsible for running java app and 2 it starts apache the web server responsible for handling web traffic and passing requests to apache.

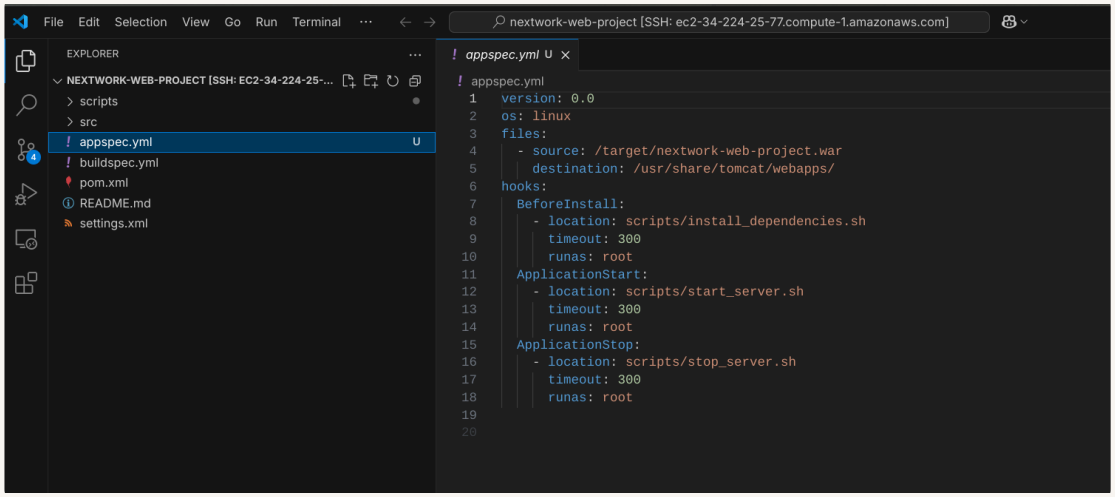
stop_server.sh will stop our apache and tomcat servers when they are nolonger to serve the web app to a user.

Shadrack Kalukwo
NextWork Student

appspec.yml

Then, I wrote an appspec.yml file to give CodeDeploy the instructions for deploying our web app. The key sections in appspec.yml are before install i.e. use the install_dependencie.sh, Application insatll i.e start-server.sh and stop_server.sh

I also updated buildspec.yml to tell CodeBuild that it should also package the new appspec.yml and set of scripts that we generated inside the build artiact i.e. compressed .war file



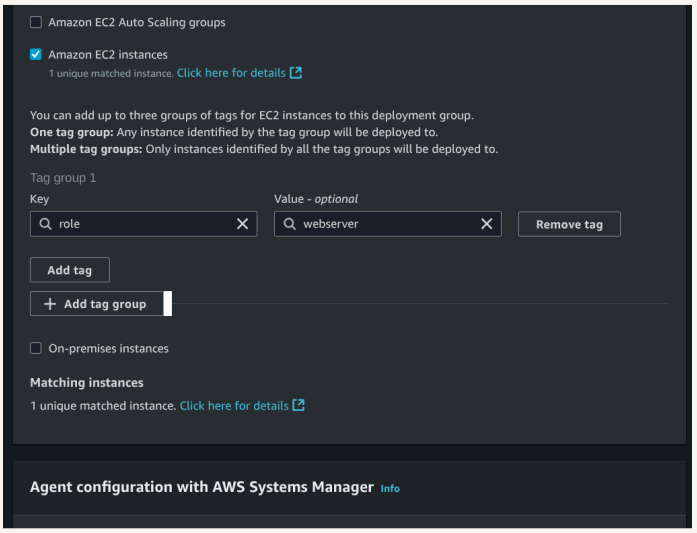
S
Shadrack Kalukwo
NextWork Student

Setting Up CodeDeploy

A deployment group is a group of ec2 instances that you can deploy to and a colllection of settings that determine how you want to deploy your web app. A CodeDeploy application is simply a folder that holds together all the deployment groups.

To set up a deployment group, you also need to create an IAM role to give CodeDeploy the permission to access the ec2 instances that it needs to coordinate, otherwise CodeDeploy doesn't have access to ec2, hence not being able to intruct the ec2 .

Tags are helpful for identifying the instances that we'll be deploying the webapp. We used a tag role=webserver to automatically match the ec2 we deployed. With new ec2's, we tag the new instance with the same tag to add to the same deployment grp.

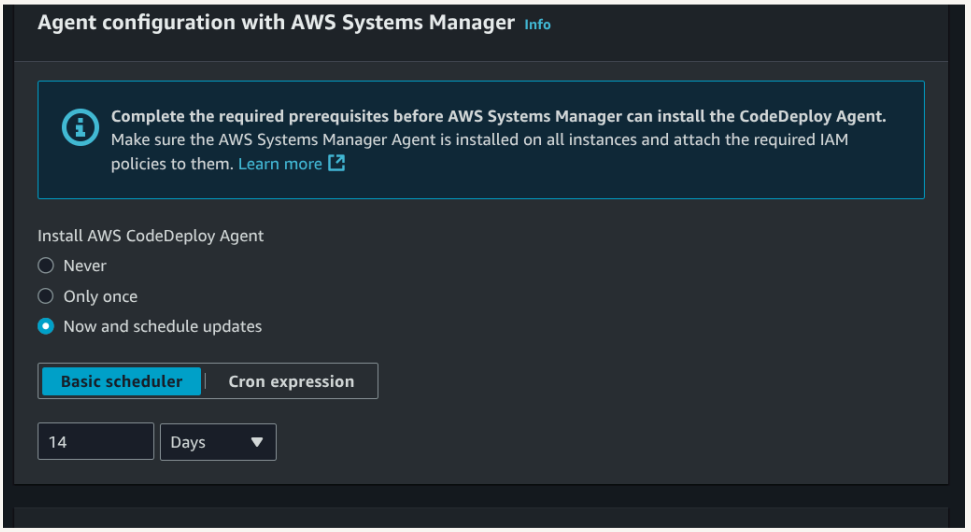

Amazon EC2 Auto Scaling groups
Amazon EC2 instances
1 unique matched instance. Click here for details
You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.
Tag group 1
Key
role
Value - optional
webserver
Remove tag
Add tag
Add tag group
On-premises instances
Matching instances
1 unique matched instance. Click here for details
Agent configuration with AWS Systems Manager Info

S
Shadrack Kalukwo
NextWork Student

Deployment configurations

Another key settings is the deployment configuration, which affects how quickly and risky we are deploying the webapp. We used CodeDeploy default all at once, so any changes that CodeDeploy deploys affects all the instances at once. It is the fastest

In order to connect the deployment instance with CodeDeploy, a deployment agent is set up to recieve instructions from CodeDeploy and make sure that the commands in appspec.yml are run.



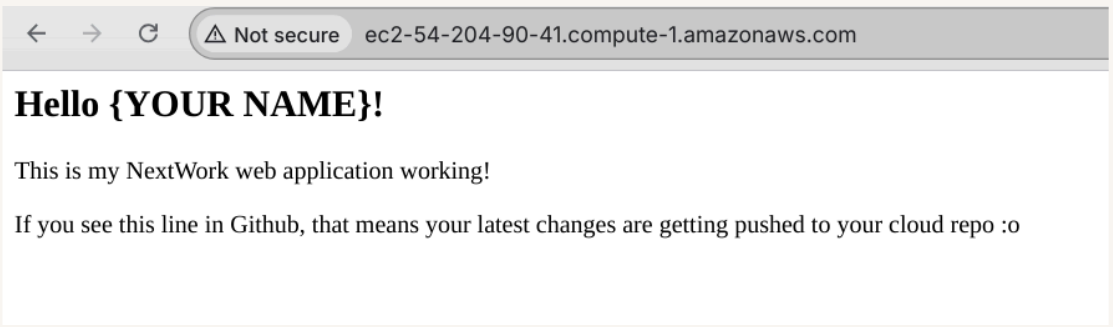
S
Shadrack Kalukwo
NextWork Student

Success!

A CodeDeploy deployment is a specific update to our application that we are deploying to our users. The difference between a deployment and a deployment group is that the group is like the settings file and deployment is the specific update rolled

'I had to configure a revision location, which means where our web app's war file ready to be deployed lives. Our revision location is the S3 bucket we've created and linked with CodeBuild.

To check that the deployment was a success, I visited the ipv4 dns address of our ec2 instance, we saw a live web that's working and serving our web app code to end users.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects