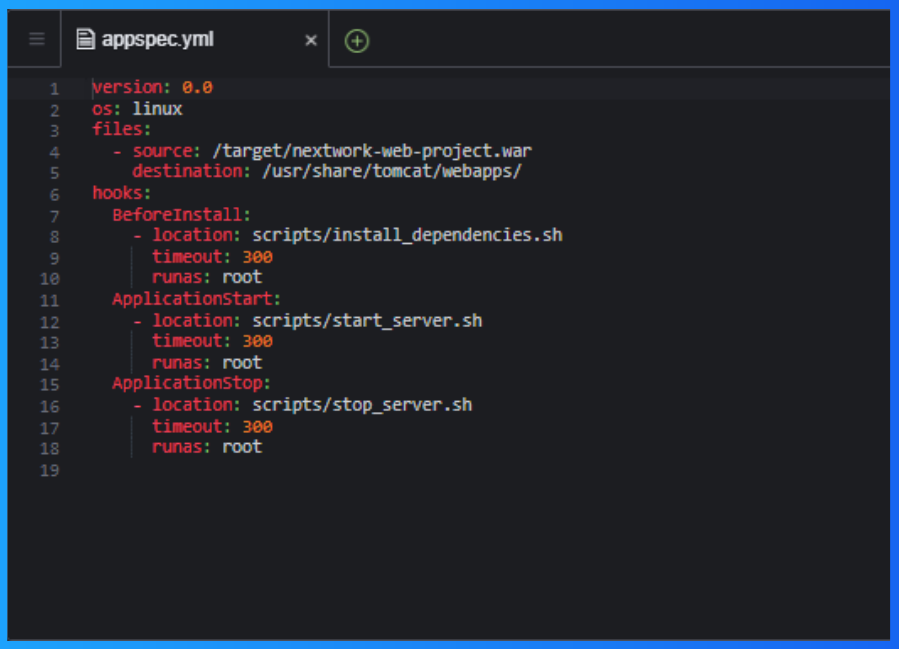
NextWork.org

Deploy an App with CodeDeploy

ZA zandiletsh20@gmail.com



ZA
zandiletsh20@gmail.com
NextWork Student
NextWork.org

Introducing today's project!

What is AWS CodeDeploy?

AWS CodeDeploy automates and manages the deployment of code to EC2 instances, Lambda functions, and on-premises servers. It supports various deployment strategies, handles rollbacks, and integrates with other AWS services.

How I'm using AWS CodeDeploy in this project

Used the AWS CodeDeploy in this project to automate the deployment of the web app by setting up an application, configuring deployment targets, using an S3 bucket for the latest WAR file, and managing deployments with the CodeDeploy Agent.

One thing I didn't expect...

Seeing an error when opening the website, how I resolved this by using http and not https and check if the own IP address hasn't changed.

This project took me...

60 mins

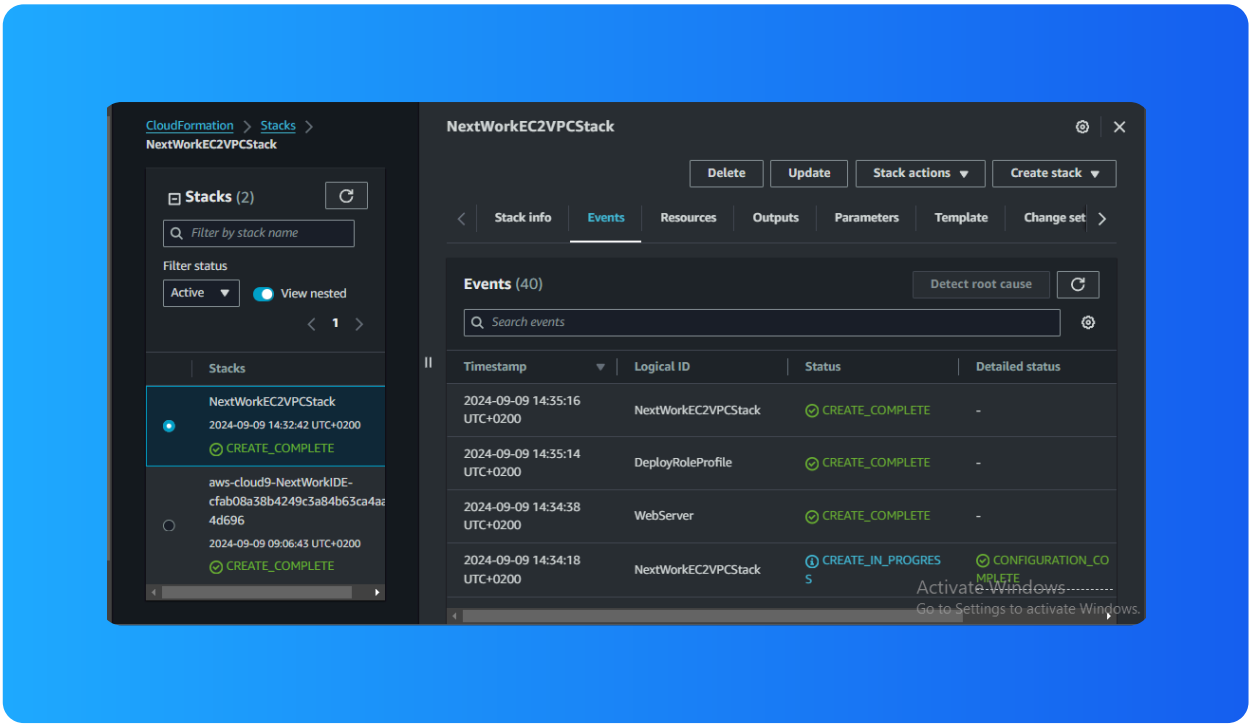
ZA
zandiletsh20@gmail.com
NextWork Student

Set up an EC2 instance

I set up an EC2 instance and VPC because deploying the web app needs a separate environment (my production environment). Setting up an EC2 instance and VPC provides a secure and flexible environment for running applications in the cloud.

We manage production and development environments separately because it is important for developers to test code without interfering with what the users see.

To set up my EC2 instance and VPC, I used AWS Cloudformation.



ZA
zandiletsh20@gmail.com
NextWork Student
NextWork.org

Bash scripts

Scripts are automate terminal tasks by running a series of commands in sequence. This makes repetitive tasks, like software installation or program startup, faster and easier by executing all steps with a single command.

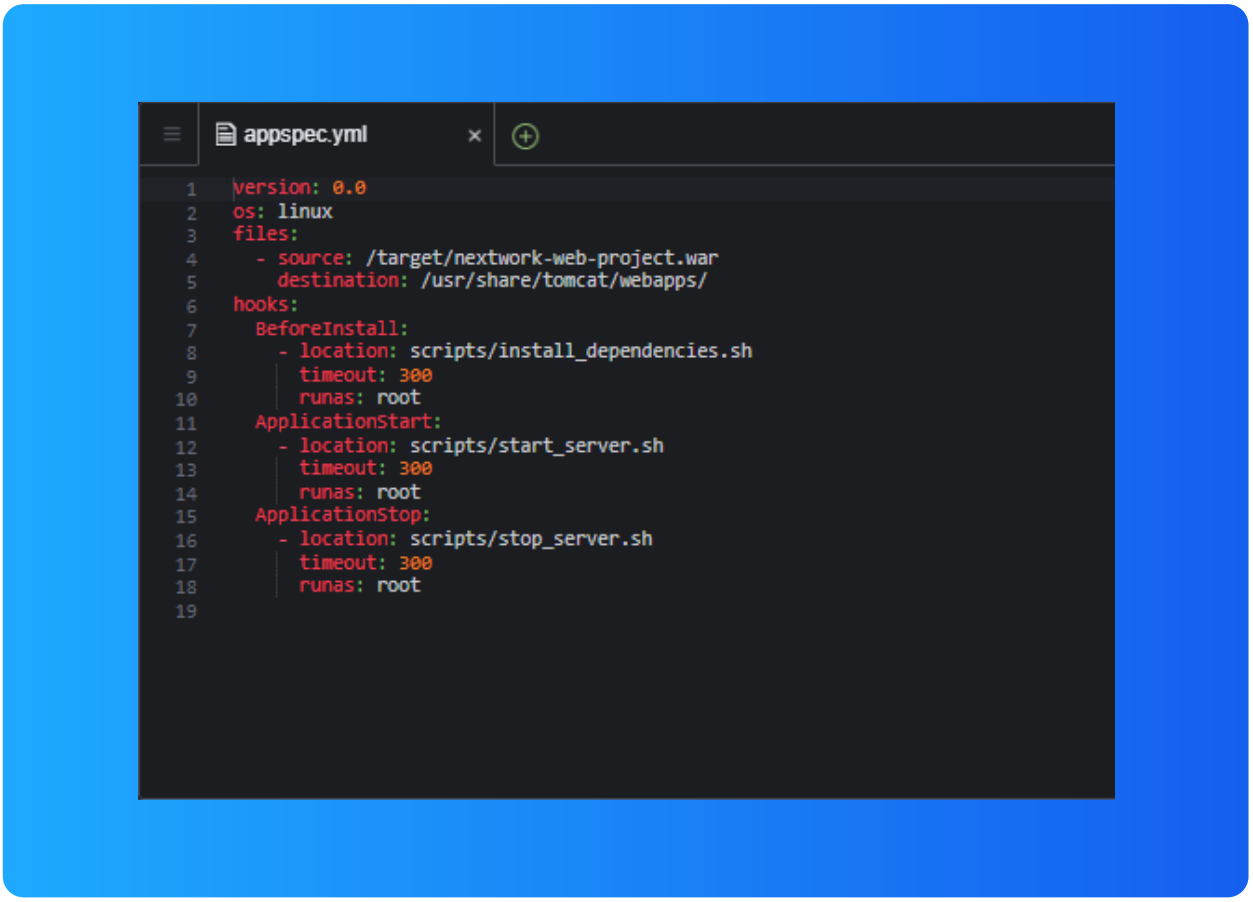
I used three scripts for my project's deployment

The first script I created was install_dependencies.sh installs Tomcat and HTTPD(web servers)

The second script I created was start_server.sh, which starts up the web servers

The third script I created was stop_server.sh which stops the web servers

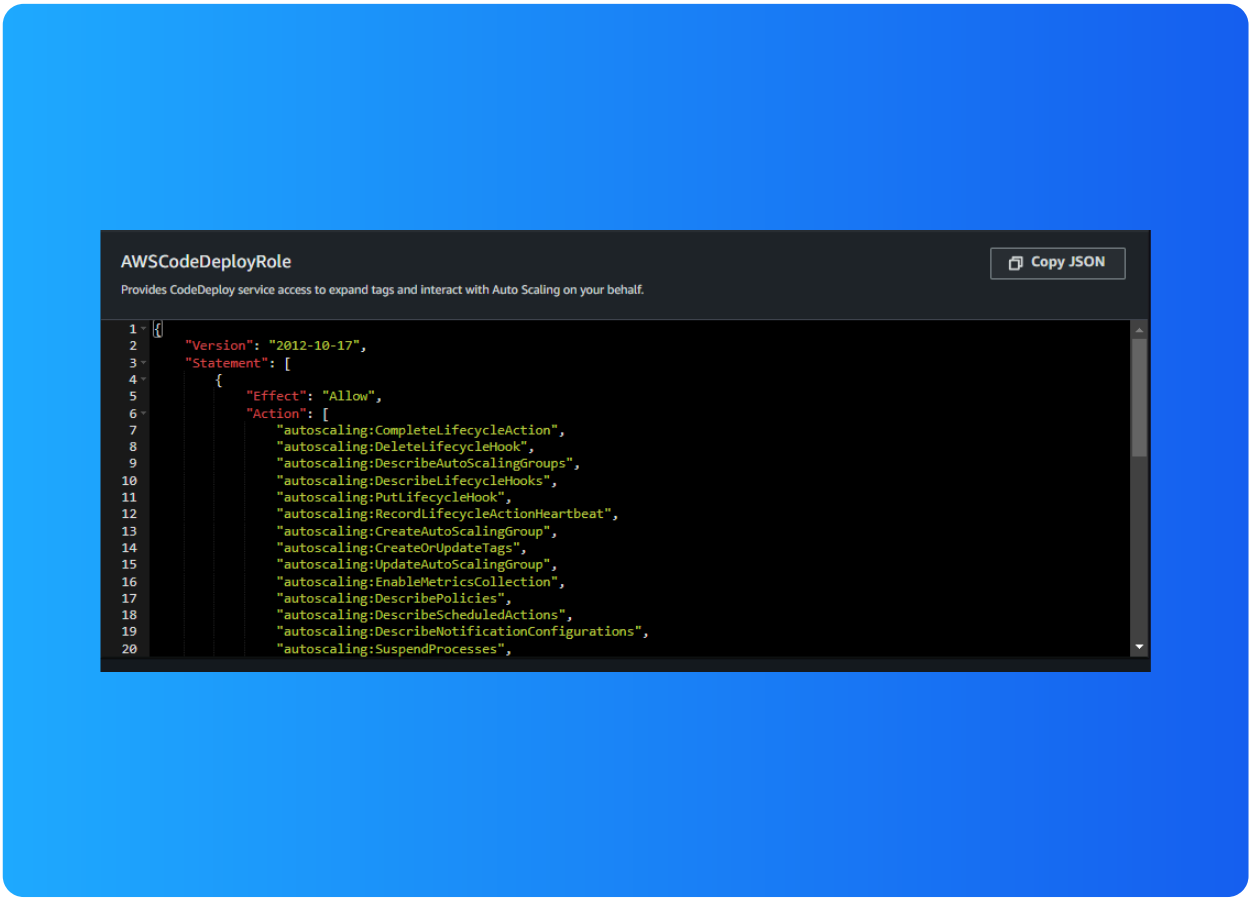
Bash scripts



CodeDeploy’s IAM Role

I created an IAM service role for CodeDeploy because I needed to grant CodeDeploy access to EC2 instances by applying the AWS Managed AWSCodeDeployRole policy, which currently does not have the necessary permissions.

To set up CodeDeploy’s IAM role, I created the role in the IAM console, chose CodeDeploy as the serive and use case. AWSCodeDeployRole policy is the default policy suggested.



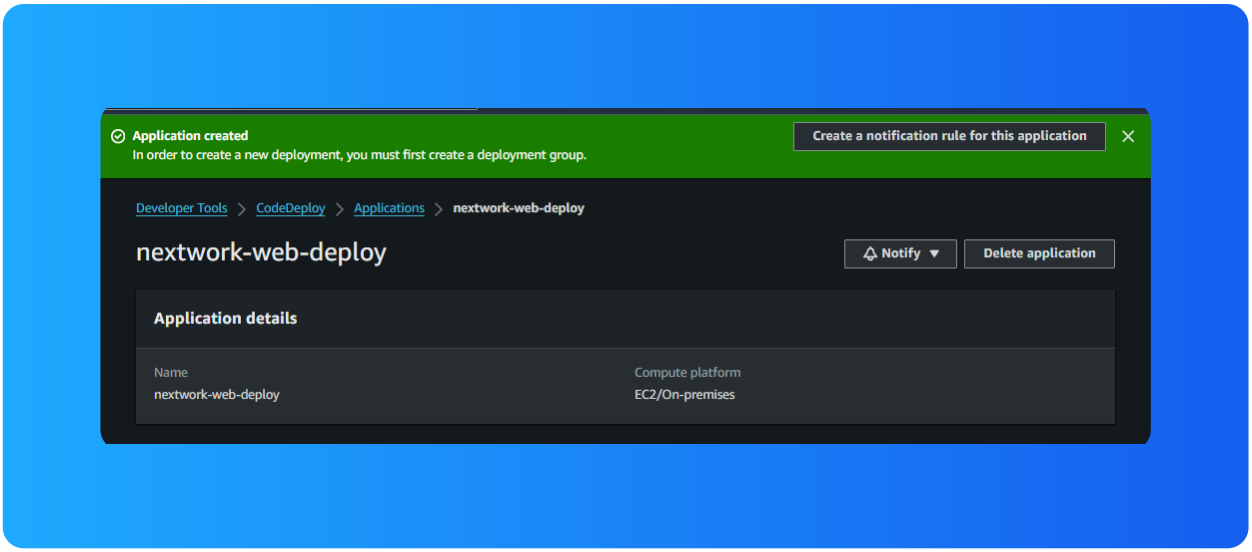
ZA
zandiletsh20@gmail.com
NextWork Student

CodeDeploy application

A CodeDeploy application means a logical entity that manages and deploys software updates across various targets, such as EC2 instances or Lambda functions.

To create a CodeDeploy application, I had to select a compute platform, which means choosing the type of infrastructure where the application will be deployed. The compute platform determines how CodeDeploy interacts with the deployment targets.

The compute platform I chose was EC2 because of the full control over virtual servers, allowing us to install, configure, and manage everything needed to run our web application and learn about server setup in detail.



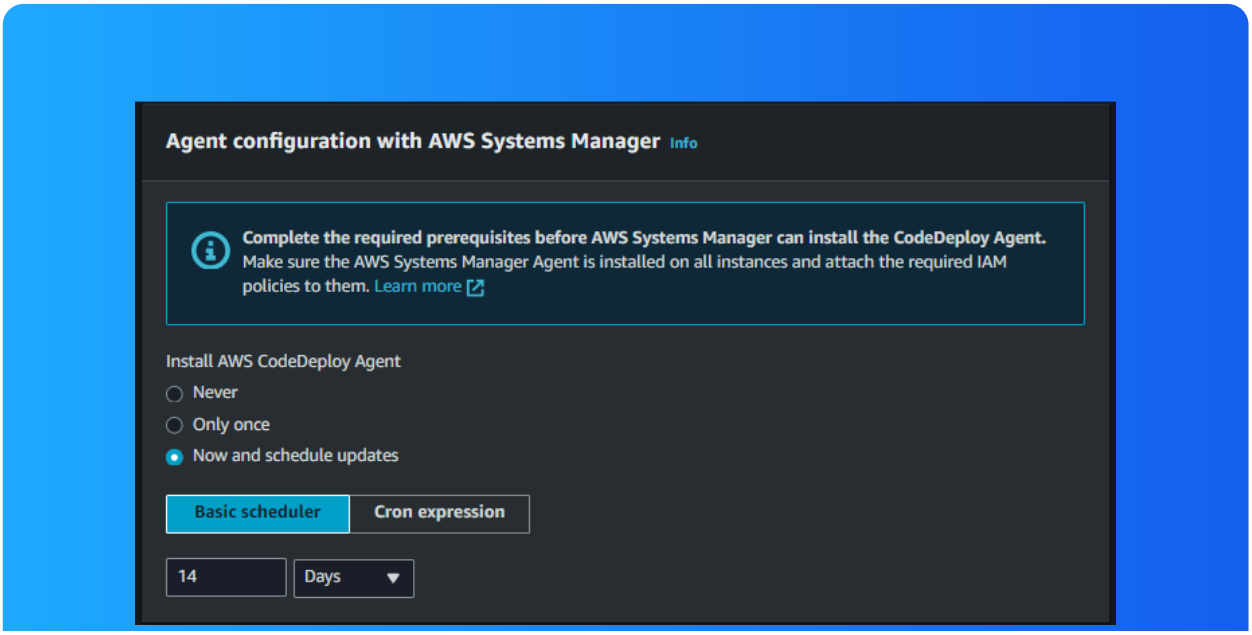
Deployment group

A deployment group means in CodeDeploy is a set of target instances or Lambda functions where your application is deployed. It includes configurations for deployment strategy, health checks, and resource identification to manage and apply updates.

Two key configurations for a deployment group

Environment means the settings that define how and where the application is deployed.

A CodeDeploy Agent is a component installed on EC2 instances or on-premises servers that communicates with CodeDeploy, executes deployment scripts, and reports deployment status.

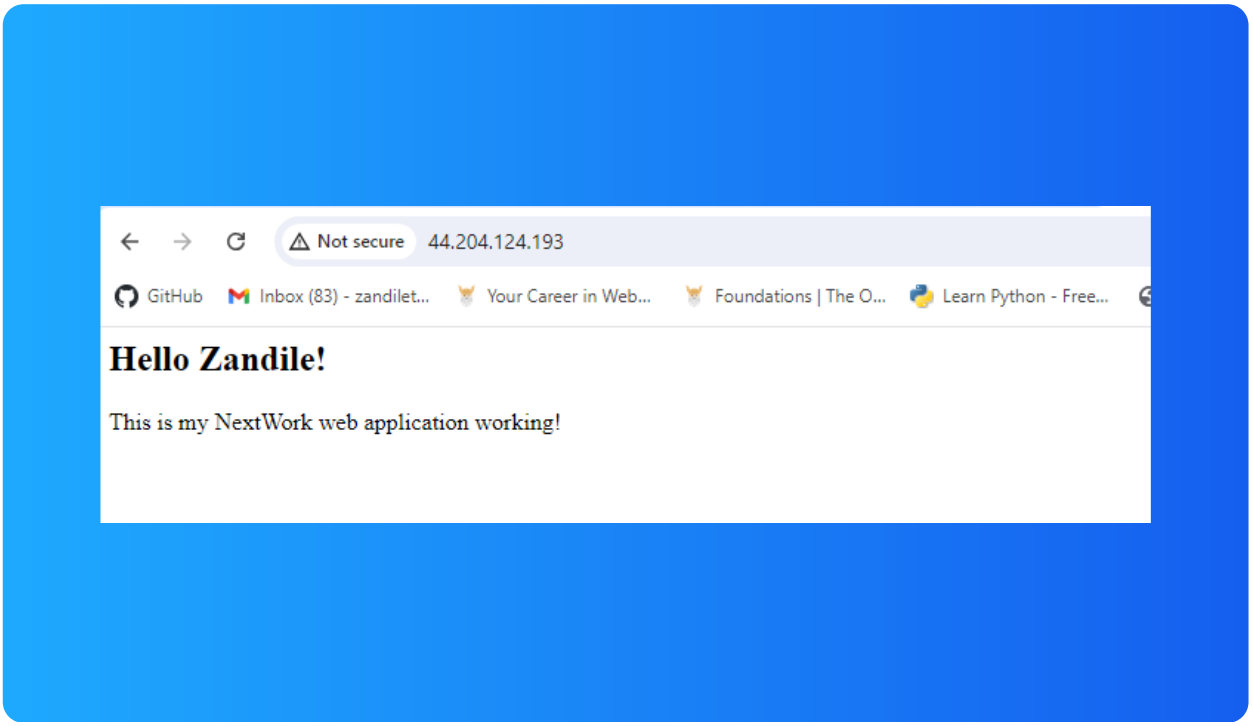


CodeDeploy application

To create my deployment, I had to set up a revision location, which means the source where application code or configuration files are stored. It specifies where CodeDeploy retrieves the files for deployment.

My revision location was my S3 bucket that stores our WAR file so CodeDeploy can access the latest version of the web app for deployment to the target instances.

To visit my web app, I had to visit the EC2 console, then selected the WebServer EC2 instance, then clicking the open address link to access the web app



Everyone should be in a job they love.

Check out nextwork.org for more projects