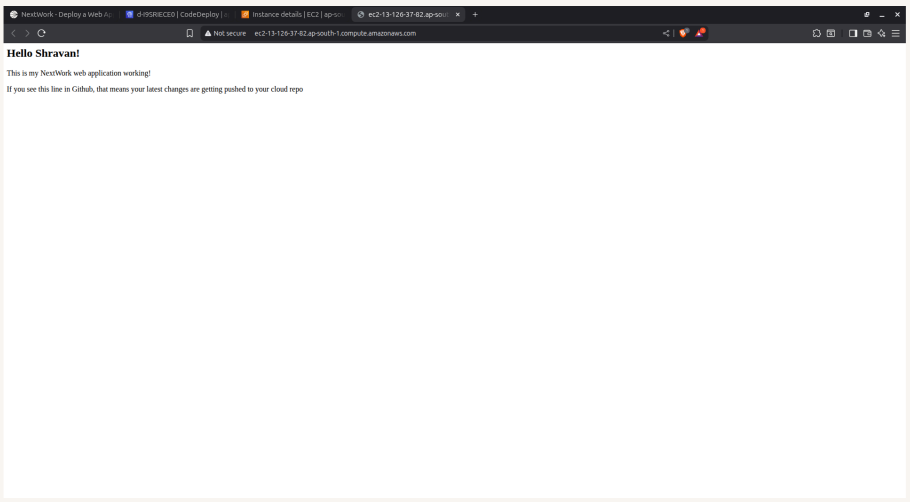
nextwork.org

Deploy a Web App with CodeDeploy

Shravan Kumar Satapathy



Shravan Kumar Satapathy
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate automating web app deployment using AWS CodeDeploy. I'm doing this project to learn how CodeDeploy automates releases, ensuring faster, consistent, and less error-prone deployments, even handling rollbacks.

Key tools and concepts

Services I used were AWS CodeDeploy, EC2, CloudFormation, S3, and IAM. Key concepts I learnt include Continuous Deployment, Infrastructure as Code, appspec.yml, deployment scripts, CodeDeploy Agent, and EC2 instance targeting via tags.

Project reflection

This project took me approximately 150 minutes. The most challenging part was preparing deployment scripts and troubleshooting initial deployment failures. It was most rewarding to successfully deploy the web app and verify its live functionality.

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, Day 6, very soon to continue expanding the pipeline's capabilities.

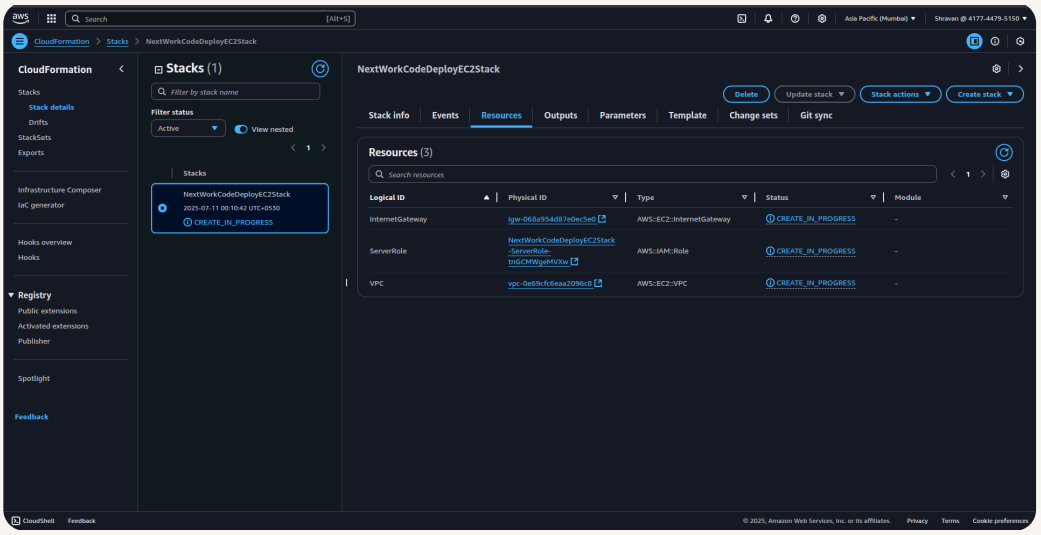
Deployment Environment

To set up for CodeDeploy, I launched an EC2 instance and VPC using CloudFormation because the EC2 hosts the live web app, and the VPC provides the essential, isolated network for secure deployment and user access.

Instead of launching these resources manually, I used AWS CloudFormation to deploy my EC2 instance and VPC. When I need to delete these resources, I will simply delete the CloudFormation stack, which removes all associated resources as a single unit.

Other resources created in this template include Subnet, Route Tables, Internet Gateway, and Security Group. They're also in the template because it builds a complete, secure, and configurable network infrastructure, crucial for complex web apps.

Shravan Kumar Satapathy
NextWork Student



Deployment Scripts

Scripts are mini-programs that automate tasks by running command sequences. To set up CodeDeploy, I also wrote scripts to automate deployment, installing dependencies like Tomcat and Apache, and configuring them to host my web app for access.

install_dependencies will set up all software required to run the web app by installing Tomcat and Apache. It then configures these programs to work together, making the website accessible to visitors on the internet.

start_server.sh will start both Tomcat and Apache, the Java application and web servers. It also configures them to automatically restart if the EC2 instance reboots, ensuring continuous application availability.

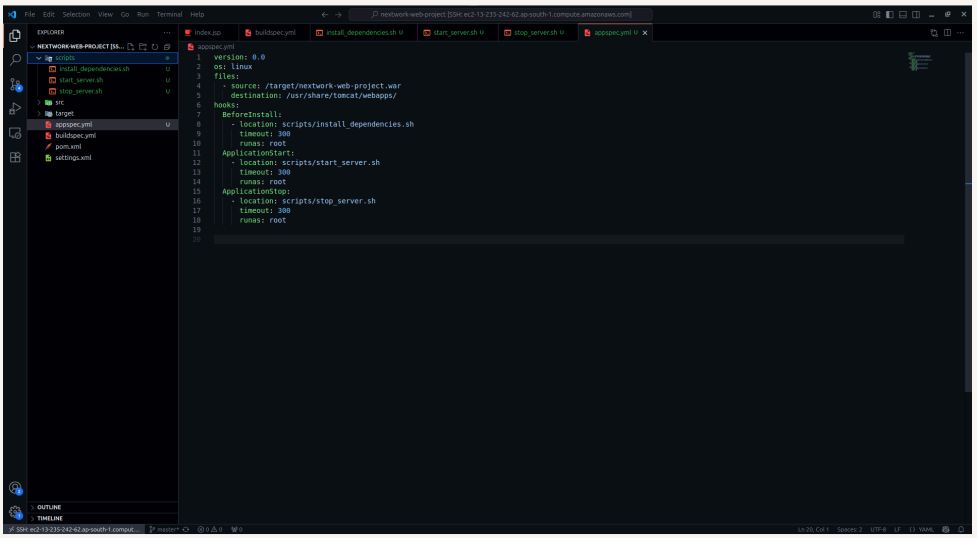
stop_server.sh will safely stop web server services (Apache and Tomcat) by first checking if they are actively running. This prevents errors from attempting to stop non-existent processes, making the deployment script more robust.

Shravan Kumar Satapathy
NextWork Student

appspec.yml

Then, I wrote an appspec.yml file to instruct CodeDeploy on deployment. The key sections in appspec.yml are: version, os, files (source/destination mapping), and hooks for running scripts at deployment lifecycle events like install or start.

I also updated buildspec.yml because CodeDeploy needs the appspec.yml and scripts folder to properly deploy the application. I modified the artifacts section to include appspec.yml and scripts/**/* in the final build package.

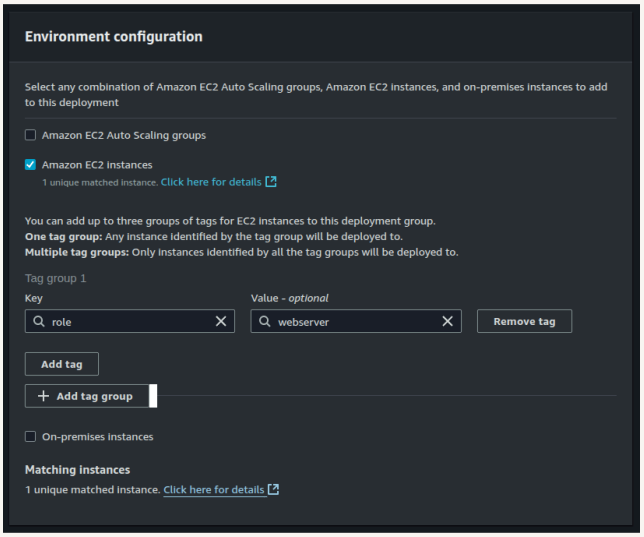


Setting Up CodeDeploy

A deployment group is a collection of EC2 instances for deploying an application, defining its deployment strategy. A CodeDeploy application is the main container organizing all deployment resources and groups for one specific application.

To set up a deployment group, you also need to create an IAM role to grant CodeDeploy permissions to manage AWS resources like EC2, S3 artifacts, Auto Scaling, and CloudWatch. This adheres to the principle of least privilege.

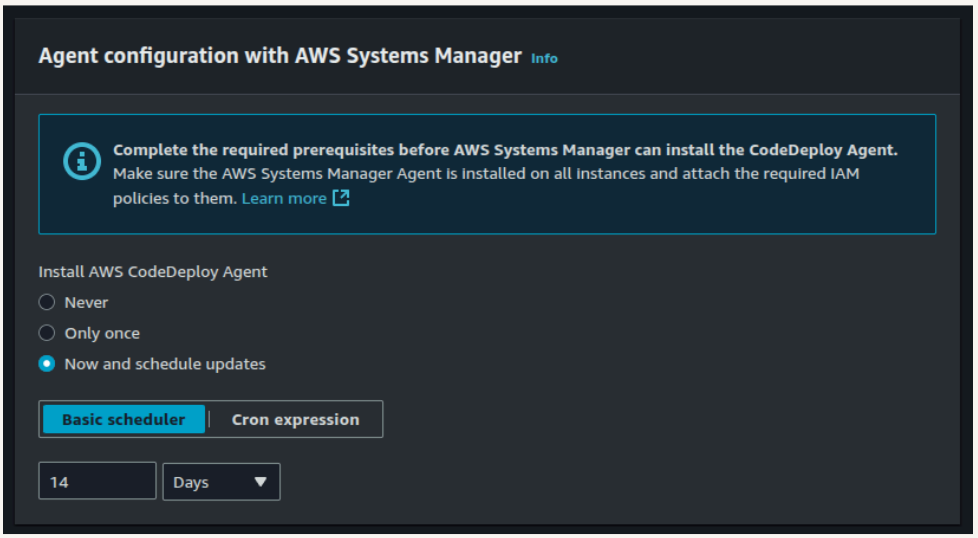
Tags are helpful for CodeDeploy to efficiently identify target EC2 instances. I used the tag role: webserver to automatically find and deploy to the correct instance, ensuring flexibility, self-documentation, and integration with CloudFormation.


Environment configuration
Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment
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

Deployment configurations

Another key setting is the deployment configuration, which affects rollout speed. I used CodeDeployDefault.AllAtOnce, so the app deploys to all instances simultaneously. This is fastest and suitable for our single-instance project.

In order to connect CodeDeploy with my EC2 instance, a CodeDeploy Agent is also set up to receive deployment instructions and carry them out on the instance. It also ensures the agent software stays up to date with automatic 14-day updates.



Success!

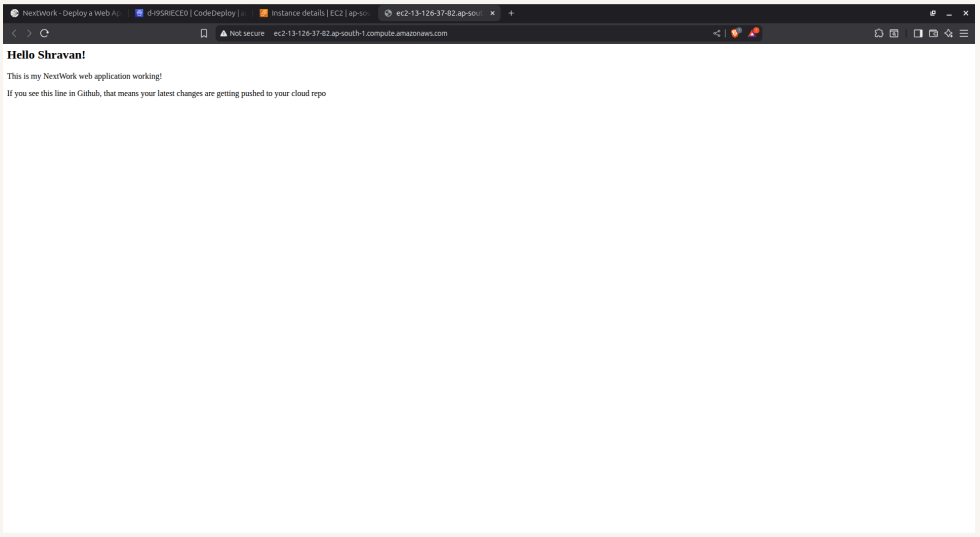
A CodeDeploy deployment is a single application update, with a specific revision and settings. The difference to a deployment group is that the group defines where and how to deploy, while a deployment is the actual execution.

I had to configure a revision location, which means specifying the place CodeDeploy finds my application's build artifacts. My revision location was the S3 URI of the nextwork-devops-cicd-artifact.zip file in my nextwork-devops-cicd S3 bucket.

To check that the deployment was a success, I visited the Public IPv4 DNS of my EC2 instance in a web browser. I saw the web application live, confirming the successful deployment.

Shravan Kumar Satapathy
NextWork Student

nextwork.org



The place to learn & showcase your skills

Check out nextwork.org for more projects