



nextwork.org

Deploy a Web App with CodeDeploy

VE

vedikagangil@gmail.com

Hello Vedika!

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 :)

Introducing Today's Project!

I will launch a deployment environment using AWS CloudFormation. Then write deployment scripts to automate deployment commands. Next deploy my web app with CodeDeploy and see it live! Then implement a disaster recovery technique.

Key tools and concepts

Services I used were AWS CodeBuild, CodeDeploy, CodePipeline, S3, CloudWatch, IAM, EC2, and VPC. Key concepts I learnt include CI/CD automation, infrastructure as code (IaC) via CloudFormation, deployment strategies.

Project reflection

This project took me 2 hours from start to finish.

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!

Deployment Environment

To set up for CodeDeploy, I launched an EC2 instance and VPC because the instance serves as the deployment target for my application, while the VPC provides isolated networking to control access and ensure secure communication between resources.

Instead of launching these resources manually, I used AWS CloudFormation to automate the deployment of my EC2 instance and VPC with a template. When I need to delete these resources, I simply delete the CloudFormation stack which removes all resource

Other resources created in this template include security groups, IAM roles, auto-scaling groups, and subnets. They're also in the template because each resource performs a very important role.

VE

vedikagangil@gmail.com

NextWork Student

nextwork.org

The screenshot shows the AWS CloudFormation console with the 'CloudFormation' navigation bar. The main view displays the 'Stacks' section, where a single stack named 'NextWorkCodeDeployEC2Stack' is listed with a status of 'CREATE_COMPLETE'. On the right, the 'Resources' tab is selected for this specific stack. The 'Resources' table lists 11 items:

Logical ID	Physical ID	Type	Status
DeployRoleProfile	NextWorkCodeDeployEC2Stack-DeployRoleProfile-bWV9k2UHmKX	AWS::IAM::InstanceProfile	CREATE_COMPLETE
InternetGateway	igw-02ba359d64e4fc02b	AWS::EC2::InternetGateway	CREATE_COMPLETE
PublicInternetRoute	rtb-0069600631e4a37ac0.0/0	AWS::EC2::Route	CREATE_COMPLETE
PublicRouteTable	rtb-0069600631e4a37ac0.0/0	AWS::EC2::RouteTable	CREATE_COMPLETE

Deployment Scripts

Scripts are sets of automated commands that execute tasks without manual input. To set up CodeDeploy, I also wrote scripts to install dependencies, configure the server, and start/stop my application during deployment phases.

`install_dependencies.sh` will automatically install all required software/packages on the EC2 instance during deployment. This ensures the app's environment matches the build environment, preventing "works on my machine" issues.

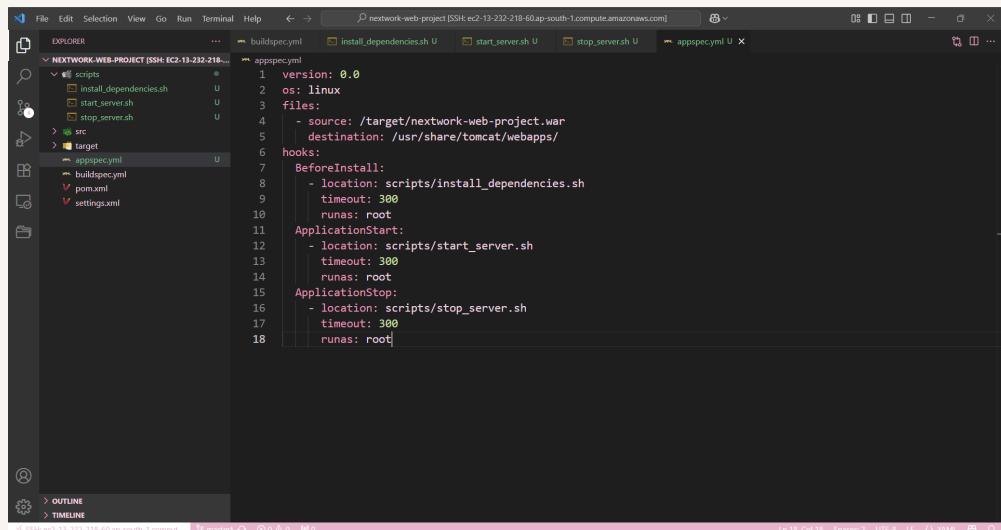
`start_server.sh` will launch your application (e.g., start a Node.js/Spring Boot server) on the EC2 instance after deployment. It ensures your app runs persistently (often using `systemd` or `nohup`) and binds to the correct port.

This script safely stops web server services by first checking if they're running. It uses `pgrep` to check for running processes of Apache (`httpd`) and Tomcat, and only attempts to stop the services if they are actually active.

appspec.yml

Then, I wrote an appspec.yml file to instruct CodeDeploy on how to deploy my application. The key sections in appspec.yml are files and hooks.

I also updated buildspec.yml because the original version didn't include the post_build phase to zip artifacts for CodeDeploy. I added these two lines: appspec.yml, scripts/**/*



```
version: 0.0
os: linux
files:
  - source: /target/nextwork-web-project.war
    destination: /usr/share/tomcat/webapps/
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

Setting Up CodeDeploy

A deployment group is a set of EC2 instances/auto-scaling groups (or Lambda/EKS targets) where CodeDeploy deploys your app. A CodeDeploy application is the container for deployment groups, representing the overall app being deployed.

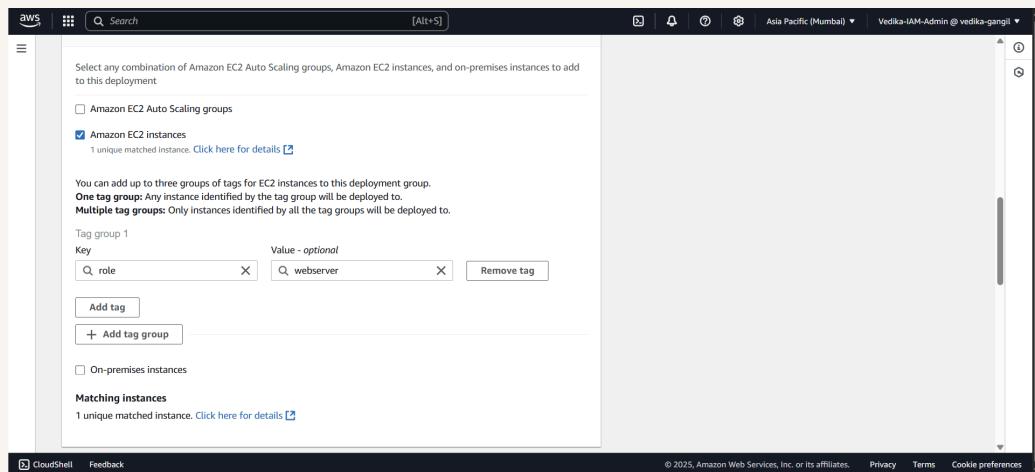
To set up a deployment group, you also need to create an IAM role to grant CodeDeploy permissions to access your EC2 instances, read S3 artifacts, and interact with other AWS services (like Auto Scaling or Lambda).

ags are helpful for dynamically grouping EC2 instances without hardcoding instance IDs. I used the tag CodeDeploy: true to let the deployment group auto-discover and target only the instances marked for deployment, ensuring flexibility.

VE

vedikagangil@gmail.com

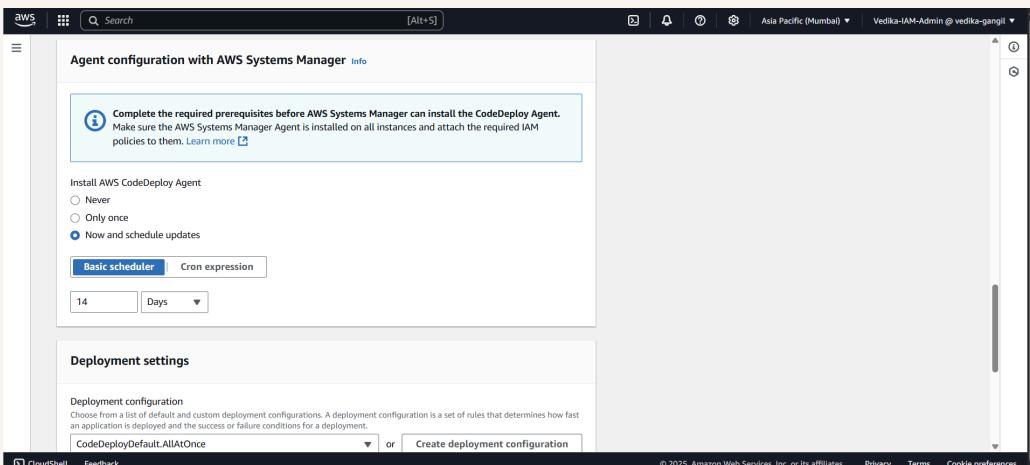
NextWork Student

nextwork.org

Deployment configurations

Another key setting is the deployment configuration, which affects how CodeDeploy rolls out updates across instances. I used CodeDeployDefault.AllAtOnce.

In order to connect your EC2 instances to CodeDeploy's deployment system, a CodeDeploy Agent is also set up to run on each target instance. This lightweight service listens for AWS deployment commands, executes the lifecycle hooks defined in your yml



Success!

A CodeDeploy deployment is a single execution that installs a specific application revision onto the instances defined in a deployment group. The difference to a deployment group is that the group is the target while the deployment is the action.

I had to configure a revision location, which means specifying where CodeDeploy should fetch the application code/artifacts (e.g., S3 bucket, GitHub repo). My revision location was an S3 bucket storing the zipped build artifacts from CodeBuild.

To check that the deployment was a success, I visited the AWS CodeDeploy console's deployment history. I saw the status marked as "Succeeded," verified all lifecycle events completed without errors, and confirmed the app was running.

VE

vedikagangil@gmail.com

NextWork Student

nextwork.org

Hello Vedika!

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 :o



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

