



nextwork.org

Deploy a Web App with CodeDeploy



Gaurav Balpande

Hello Gaurav Balpande!

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

Introducing Today's Project!

In this project, I will demonstrate the use of CodeDeploy I'm doing this project to learn about the complete process of cicd by doing the deployment of the web app using CodeDeploy. It is necessary because web app become available on web due to cicd.

Key tools and concepts

Services I used were Codebuild,Codedeploy ,Iam,Ec2 and Github Key concepts I learnt include Codedeploy which consist of all the cicd pipeline from making of web app to accessing it through internet.

Project reflection

This project took me approximately 3 hrsThe most challenging part was the last deploy process because i had been stucked there for more than an hour It was most rewarding to get the webpage for our 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 may be tomorrow!

Deployment Environment

To set up for CodeDeploy, I launched an EC2 instance and VPC because we need to have separate Ec2 instance for development and production. We had created production Ec2 instance with cloudformation to reduce the effort and add VPC for public access.

Instead of launching these resources manually, I used CloudFormation When I need to delete these resources we just need to delete the CloudFormation stack.

Other resources created in this template include VPC,subnets,Route table,internet gateway and security group They're also in the template because it provide extra level of security and also it need has to internet access.



CloudFormation > Stacks > NextWorkCodeDeployEC2Stack

CloudFormation

- Stacks
 - Stack details**
 - Drifts
 - StackSets
 - Exports
- Infrastructure Composer
- Hooks overview
- Registry**
 - Public extensions
 - Activated extensions
 - Publisher
- Spotlight

Stacks (1)

Filter status Active

View nested

Stacks

NextWorkCodeDeployEC2Stack
2025-04-23 00:49:25 UTC+0530
CREATE_IN_PROGRESS

Events (34)

Search events

Timestamp	Logical ID	Status	Detailed status
2025-04-23 00:49:54 UTC+0530	PublicInternetRoute	CREATE_COMPLETE	-
2025-04-23 00:49:53 UTC+0530	PublicInternetRoute	CREATE_IN_PROGRESS	-
2025-04-23 00:49:52 UTC+0530	PublicInternetRoute	CREATE_IN_PROGRESS	-
2025-04-23 00:49:52 UTC+0530	PublicRouteTable	CREATE_COMPLETE	-
2025-04-23 00:49:50 UTC+0530	WebServer	CREATE_IN_PROGRESS	-
2025-04-23 00:49:49 UTC+0530	PublicSecurityGroup	CREATE_COMPLETE	-
2025-04-23 00:49:47 UTC+0530	DeployRoleProfile	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE
2025-04-23 00:49:47 UTC+0530	DeployRoleProfile	CREATE_IN_PROGRESS	-
2025-04-23 00:49:47 UTC+0530	PublicSubnetARouteTabl	CREATE_COMPLETE	-

Deployment Scripts

Scripts are file like structure that contain command that has to be run on terminal To set up CodeDeploy, I also wrote scripts to install dependncies that are required to host our web app like apache and tomcat.

install_dependencies will install Tomcat and httpd. Tomcat is the server for java web app and httpd is the Apache http server to behave as a reverse proxy for Tomcat.

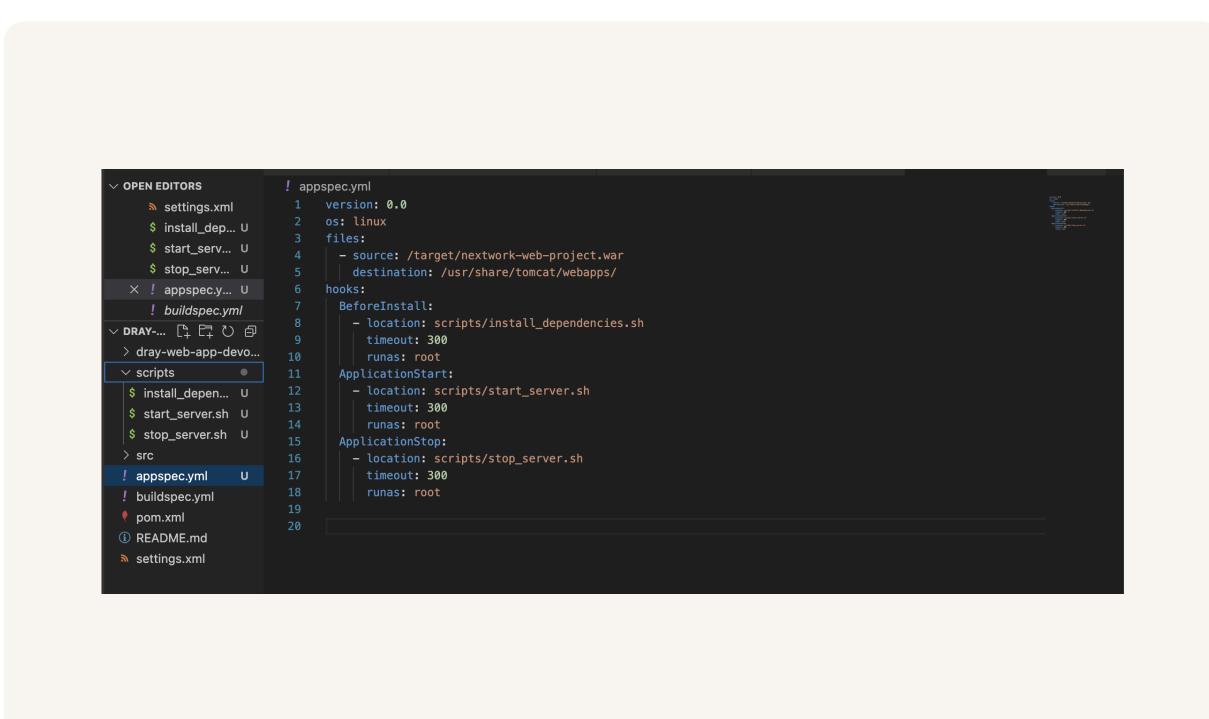
start_server.sh will start the services like Tomcat and httpd. It is also reponsible for restarting the services whenever Ec2 instance is reboot.

stop_server.sh will stop the services like Tomcat and httpd. Before stopping the service it will first check whether the services are active or not this is because we can't stop the stopped procses. It uses pgrep for checking the state of process.

appspec.yml

Then, I wrote an appspec.yml file to give instruction regarding our created scripts. The key sections in appspec.yml are the source which contain where actually our build package is located.

I also updated buildspec.yml because I want to store the folder scripts and file appspec.yaml for further operations. It will need this information because without it, Codedeploy will not know what to do with compiled code.



```
! appspec.yml
1 version: 0.0
2 os: linux
3 files:
4   - source: /target/nextwork-web-project.war
5     destination: /usr/share/tomcat/webapps/
6 hooks:
7   BeforeInstall:
8     - location: scripts/install_dependencies.sh
9       timeout: 300
10      runas: root
11   ApplicationStart:
12     - location: scripts/start_server.sh
13       timeout: 300
14      runas: root
15   ApplicationStop:
16     - location: scripts/stop_server.sh
17       timeout: 300
18      runas: root
19
20
```

Setting Up CodeDeploy

A deployment group is like a collection of different ways to deploy the web app like different EC2 instances and compute resources. A CodeDeploy application is used to place all the resources related to particular application in one place.

To set up a deployment group, you also need to create an IAM role to give permission to production Ec2 instance about accessing aws resources. we had created anew role which contain permission abot Ec2 insatnce,autoscalling,cloud-watch,s3.

Tags are helpful for using multiple insatnces I used the tag role to not hard code for insatnce name but instead use tag so that we can use multiple insatnces.



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	Value - optional
<input type="text" value="role"/> <input type="button" value="X"/>	<input type="text" value="webserver"/> <input type="button" value="X"/>
<input type="text"/> <input type="button" value="Q"/>	<input type="text"/> <input type="button" value="Q"/>

On-premises instances

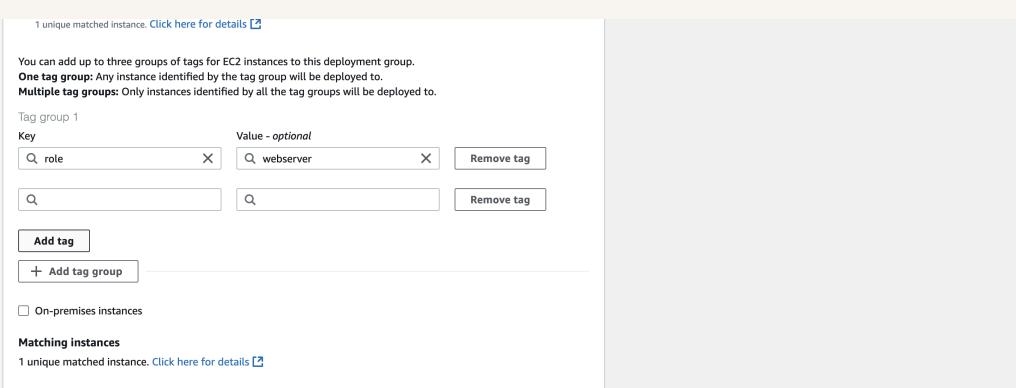
Matching instances

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

Deployment configurations

Another key settings is the deployment configuration, which affects the deployment I used CodeDeployDefault.AllAtOnce, so at a time web app has to be deployed at once. This is because it increases speed and we have just one instance so it not matter

In order to connect production Ec2 instance and CodeDeploy a CodeDeploy Agent is set up. A CodeDeploy Agent is also set up to create a extra level of control for connection between Ec2 instance and CodeDeploy.



Success!

A CodeDeploy deployment is a single update which has particular id and history. The difference to a deployment group is that deployment group includes everything related to a particular deployment of our app. CodeDeploy deployment comes under group.

I had to configure a revision location, which means the exact location where our artifact is stored. My revision location was in S3 bucket.

To check that the deployment was a success, I visited <http://ec2-43-204-38-204.ap-south-1.compute.amazonaws.com/>. I saw the java web app consisting of message "Hello Gaurav Balpande!"



Hello Gaurav Balpande!

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

