



[nextwork.org](http://nextwork.org)

# Deploy a Web App with CodeDeploy



Saish Nar

A screenshot of a web browser window. The address bar shows a warning icon followed by "Not secure" and the URL "ec2-65-0-133-130.ap-south-1.compute.amazonaws.com". The main content area displays a simple web page with the following text:

**Hello saish !**

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 how to use CodeDeploy to deploy a web app! I'm doing this project to learn about how deployment works, and how it can be automated using a combination of CodeDeploy and deployment scripts by the end of this project

## Key tools and concepts

Services I used were, AWS Codebuild, CodeArtifact, CodeDeploy, Github, VS Code and AWS S3. Key concepts included how to create a codedeploy application and create a deployment group for successful ci deployment. It was truly a wonderful experience.

## Project reflection

This project took me approximately 3hrs to complete. The most challenging part was to re-deploy the stop\_server.sh script , It was most rewarding to study the errors and troubleshoot those errors and see the success message of deployment.

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 in the next 24 hours to smash the next challenge.



**Saish Nar**  
NextWork Student

[nextwork.org](http://nextwork.org)

# Deployment Environment

To set up for CodeDeploy, I launched an EC2 instance and VPC because Instead of clicking around the AWS console to set up resources, I can write a single template file that describes everything you need - your EC2 instance, security groups, databases

Instead of launching these resources manually, I used AWS Cloudformation. When I need to delete these resources I can simply delete the CloudFormation stack. This will automatically delete all the resources that are inside that stack.

Other resources created in this template include networking resources like VPCs, internet gateways, route tables and subnets. They're also in this template because By defining these networking resources in the template for controlling both traffic.



**Saish Nar**  
NextWork Student

[nextwork.org](http://nextwork.org)

The screenshot shows the AWS CloudFormation console with the stack named "NextWorkCodeDeployEC2Stack". The "Resources" tab is selected, displaying 11 resources:

Logical ID	Physical ID	Type	Status
DeployRoleProfile	2Stack-DeployRoleProfile-IOv2xKEYDjxY	AWS::IAM::InstanceProfile	<span>CREATE_IN_PROGRESS</span>
InternetGateway	igw-0a16edf9d3eeef202c	AWS::EC2::InternetGateway	<span>CREATE_COMPLETE</span>
PublicInternetRoute	rtb-014050361f2d409c3 0.0/0	AWS::EC2::Route	<span>CREATE_COMPLETE</span>
PublicRouteTable	rtb-014050361f2d409c3	AWS::EC2::RouteTable	<span>CREATE_COMPLETE</span>
PublicSecurityGroup	sg-08d638887cd88d51a	AWS::EC2::SecurityGroup	<span>CREATE_COMPLETE</span>



**Saish Nar**

NextWork Student

[nextwork.org](http://nextwork.org)

---

# Deployment Scripts

Scripts are like mini-programs that automate tasks. To set up CodeDeploy, I also wrote scripts to automate deployment commands (i.e. these are the commands that my deployment EC2 instance needs to run in order to host my web app).

`install_dependencies.sh` will help my EC2 instance install all the dependencies it needs to host a web app, like Tomcat (a web server).

`start_server.sh` will start both Tomcat (my Java application server) and Apache (my web server) and makes sure they'll restart automatically if the EC2 instance ever reboots.

`stop_server.sh` is a script that stops my Apache and Tomcat servers when they're no longer needed to serve the web app to a user.

**Saish Nar**  
NextWork Student

[nextwork.org](http://nextwork.org)

## appspec.yml

Then, I wrote an appspec.yml file to give CodeDeploy the instructions for deploying my web app! The key sections in appspec.yml are BeforeInstall, ApplicationStart (i.e. start\_server.sh script) and ApplicationStop (i.e. stop\_server.sh script).

I also updated buildspec.yml because to tell CodeBuild that it should also package up the new appspec.yml and set up scripts that I created inside the bulld artifact (i.e. the compressed war file)

```
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 group of EC2 instances that you can deploy to, and also collection of settings that determine HOW you want to deploy your webapp. A CodeDeploy application is simply a folder that holds together all deployment groups for same app

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.

Tags are helpful for identifying so that CodeDeploy can identify the target EC2 instances for deployment. In my CloudFormation template, I tagged my EC2 instance with role: webserver. CodeDeploy will use this tag to find & deploy to the correct EC2.



**Saish Nar**  
NextWork Student

[nextwork.org](http://nextwork.org)

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

On-premises instances

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

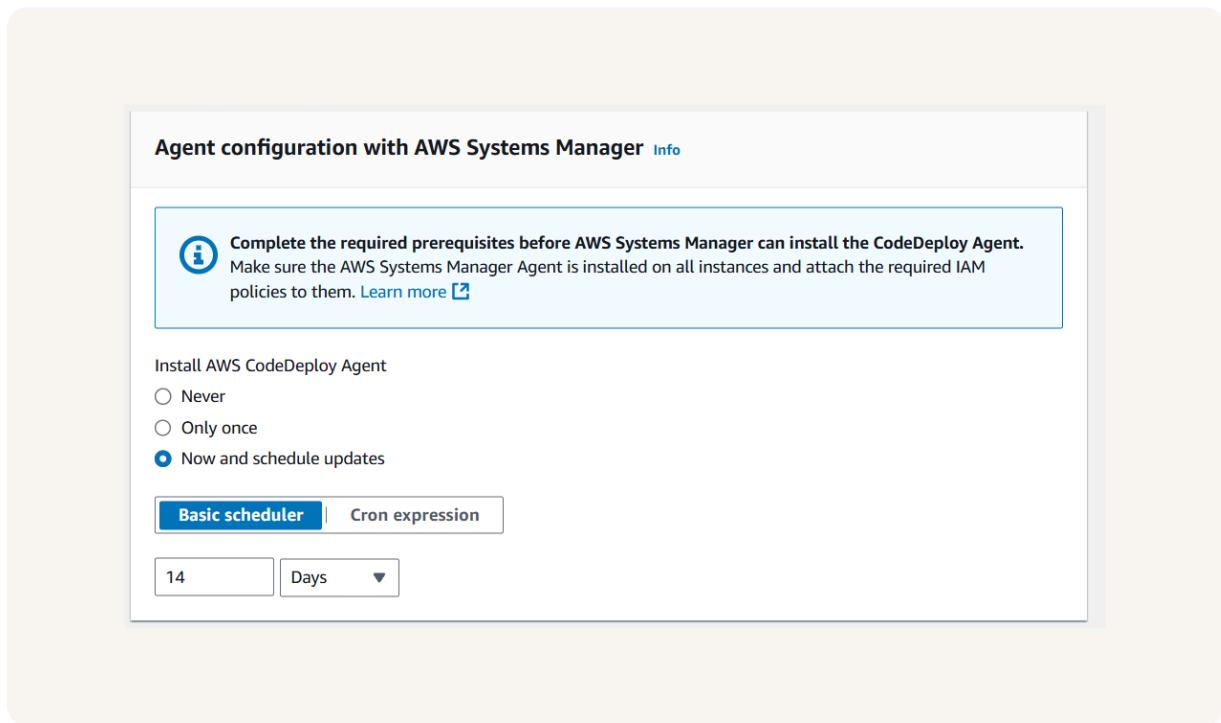
**Saish Nar**  
NextWork Student

[nextwork.org](http://nextwork.org)

# Deployment configurations

Another key setting is the deployment configuration, which affects how quickly and risky I'm deploying the web app. I used CodeDeployDefault.AllAtOnce, so any changes that CodeDeploy deploys affect ALL the instances in deployment group at once.

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





**Saish Nar**  
NextWork Student

[nextwork.org](http://nextwork.org)

# Success!

A CodaDeploy deployment is Specific update to my application that I am deploying to my users. The difference to deployment group is that group is like a settings file, and the deployment itself is like a specific update I roll out using settings file

I had to configure a revision location, which means where my web app's WAR file (i.e. the compressed file ready to be deployed) lives. My revision location is the S3 bucket that I've created and linked with CodeBuild.

To check that the deployment was a success, I visited the IPv4 DNS address of my deployment EC2 instance! I saw a live web app that's working and serving my web app code to end users.



**Saish Nar**

NextWork Student

[nextwork.org](http://nextwork.org)

← → ⌛ ⚠ Not secure ec2-65-0-133-130.ap-south-1.compute.amazonaws.com

**Hello saish !**

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](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

