



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

# Deploy an App with CodeDeploy



Sadeesha Perera





# Introducing AWS CodeDeploy!

## What it does & how it's useful

AWS CodeDeploy is an automated deployment service that streamlines application deployment across multiple compute platforms on AWS. Developers and teams use AWS CodeDeploy because it provides flexible content deployment, minimizes downtime, and works with existing code directly from Amazon S3 and other repositories.

## How I'm using it in today's project

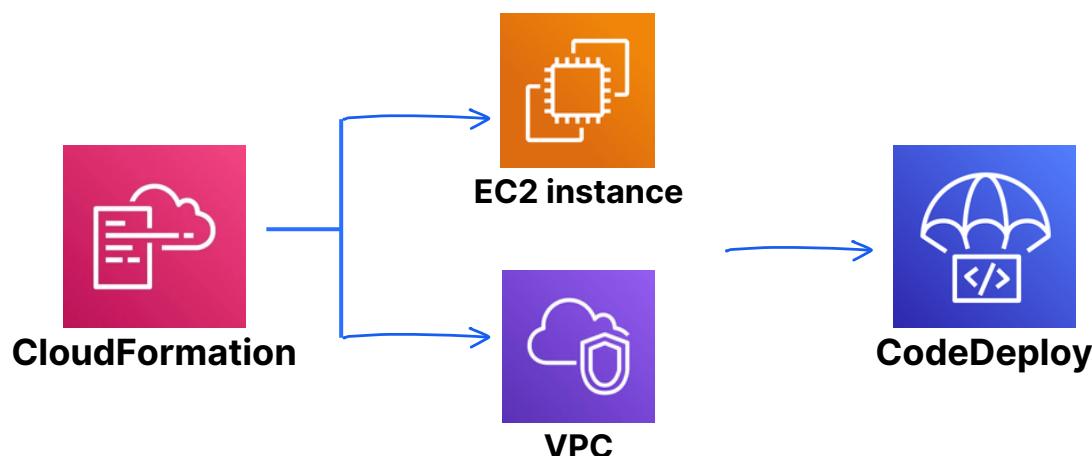
I'm using AWS CodeDeploy in this project to streamline the development and deployment of my web application with simplifying configuration across various platforms.

## This project took me...

I spent an hour setting up and completing this project, and documentation took me an extra 30 minutes. In total 1 hour and 30 minutes.

# Set up an EC2 instance

- I set up an EC2 instance and VPC because deploying my web app requires a separate environment (my production environment).
- The production environment is different to the build environment because deploying my web app will require different tools/dependencies in order to function properly. It is also best practice to separate environment for developers to test code without interfering with what users see.
- To set up my EC2 instance and VPC, I used AWS CloudFormation.
- The diagram below illustrates the flow of using CloudFormation to create two new resources(EC2 instances, VPC) that will deploy and run my web app (which we'll see in action when I set up AWS CodeDeploy).



A peek into my CloudFormation stack's deployment!

The screenshot shows the AWS CloudFormation console with a blue header bar. Below it, the 'Stacks' section displays a list of stacks. One stack, 'NextWorkEC2VPCStack', is highlighted with a teal border and has a green checkmark next to 'CREATE\_COMPLETE'. A blue arrow points from the text above to this stack. To the right, a detailed table provides more information about the stack's resources:

Timestamp	Logical ID	Status	Detailed status	Status
2024-07-14 00:11:48 UTC+0530	NextWorkEC2VPCStack	CREATE_COMPLETE	-	-
2024-07-14 00:11:46 UTC+0530	DeployRoleProfile	CREATE_COMPLETE	-	-
2024-07-14 00:11:10 UTC+0530	WebServer	CREATE_COMPLETE	-	-
2024-07-14 00:10:41 UTC+0530	NextWorkEC2VPCStack	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Event check in

At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Cookie preferences'.

# Write bash scripts

- Scripts are collections of commands in a file i.e. when you run a script, you are telling the terminal to run (line by line) what i've written in the script
- Bash is a type of script language.
- I created 3 scripts for this the deployment process:  
a.install\_dependencies.sh installs Tomcat and HTTPD (web servers).  
b.start\_server.sh starts up my web servers.  
c.stop\_server.sh stops the web servers.

A peek into appspec.yml!

```
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
```

# CodeDeploy's IAM Role

- I created an IAM service role for CodeDeploy because CodeDeploy needs access to other services like EC2 in order to successfully deploy my web app
- To set up CodeDeploy's IAM role, I attached an AWS managed Policy called `AWSCodeDeployRole` which automatically adds default permissions that CodeDeploy often needs.

The permissions granted by my CodeDeploy IAM role.

The screenshot shows the AWS IAM Policies page. A blue arrow points from the text above to the policy name "AWSCodeDeployRole". The policy details are as follows:

Policy name	Type
<a href="#">AWSCodeDeployRole</a>	AWS managed

**AWSCodeDeployRole**

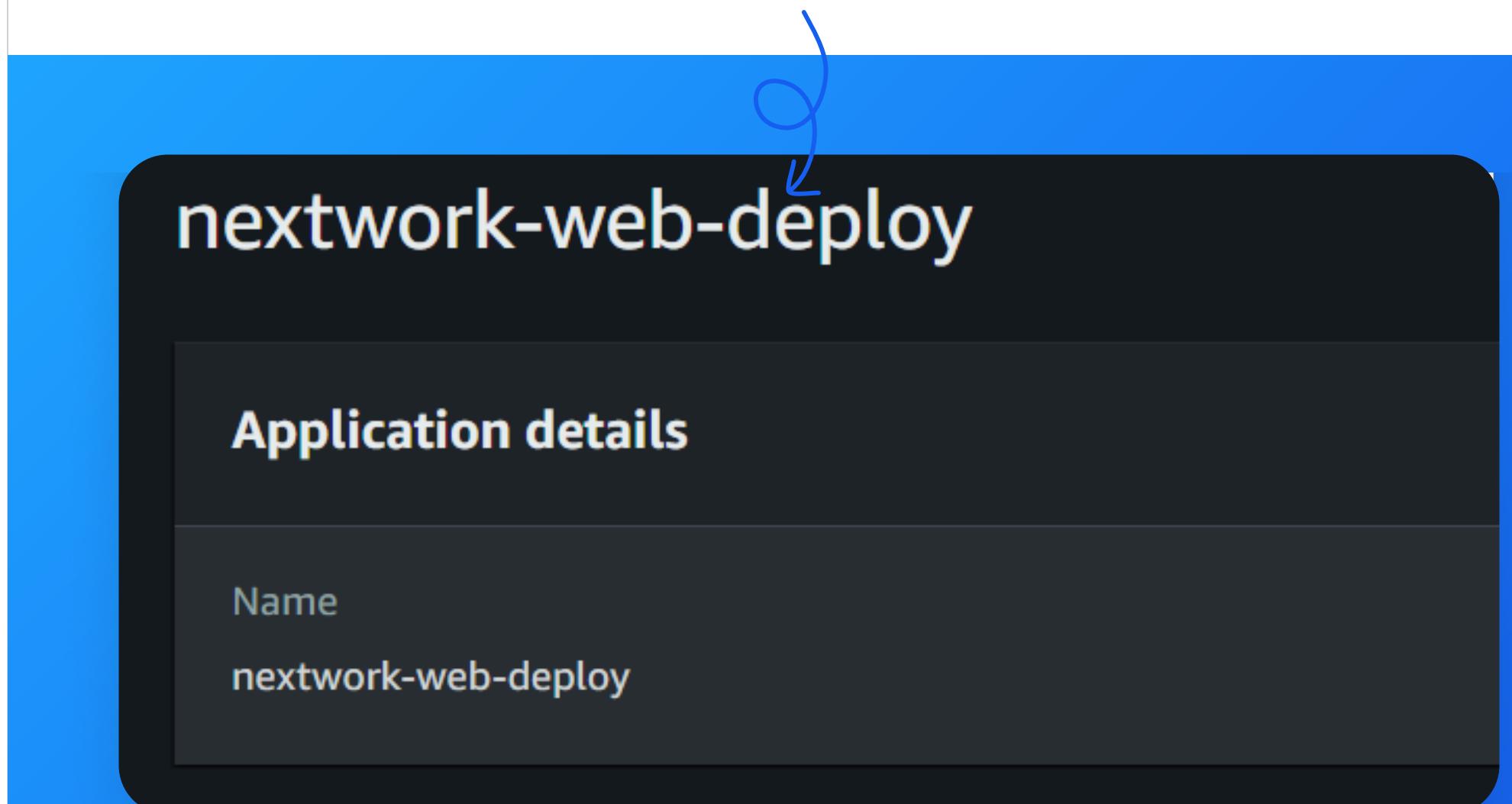
Provides CodeDeploy service access to expand tags and interact with Auto Scaling on your behalf.

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": [
7                  "autoscaling:CompleteLifecycleAction",
8                  "autoscaling>DeleteLifecycleHook",
9                  "autoscaling:DescribeAutoScalingGroups",
10                 "autoscaling:DescribeLifecycleHooks",
11                 "autoscaling:PutLifecycleHook",
12                 "autoscaling:RecordLifecycleActionHeartbeat",
13                 "autoscaling>CreateAutoScalingGroup",
14                 "autoscaling>CreateOrUpdateTags",
15                 "autoscaling:UpdateAutoScalingGroup",
16                 "autoscaling:EnableMetricsCollection",
17                 "autoscaling:DescribePolicies",
18                 "autoscaling:DescribeScheduledActions",
19                 "autoscaling:DescribeNotificationConfigurations",
20                 "autoscaling:SuspendProcesses",
21                 "autoscaling:ResumeProcesses"
22             ]
23         }
24     ]
25 }
```

# Create a CodeDeploy app

- A CodeDeploy application means a saved configuration template on how to deploy my web app.
- To create a CodeDeploy application, I had to select a computing platform, which determines the environment in which my web application will be hosted. With every compute platform comes its own set of requirements for deploying a web app.
- The compute platform I chose was EC2, because i have used an EC2 instance as my web server. Choosing EC2 gives me the most amount of control out of all three compute options. it also benefits my learning because it exposes me to more options.

My CodeDeploy app ready for a deployment!



# Set up a deployment group

- A deployment group means the configuration set for a specific scenario.
- To create my deployment group, I set up a:
  - **Service role** is an IAM Role that I am using for my deployment group to give CodeDeploy access to my web app server EC2 instance.
  - **Deployment type**, which is how deployment will be managed, I chose in place, i.e. my webserver EC2 instance will deploy the latest web app right away without needing to create a new environment
  - **Environment configuration**, which means the type of servers that will be used to deploy my web app - in this case, I just used Amazon EC2 instances
  - **CodeDeploy Agent**, which is a helper for communicating with my EC2 web server and making sure they carry out the instructions for deployment.
  - **Deployment settings**, which is whether deployment is staggered out or done all out once.
- For the load balancer setting, I disabled the load balancer.
  - This was because I only have one web server deploying my web app.

Sadeesha Perera  
linkedin.com/sadeesha-perera

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

Setting up the service role + deployment type.

**Service role**

Enter a service role  
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

arn:aws:iam::992382852391:role/NextWorkCodeDeployRole

**Deployment type**

Choose how to deploy your application

In place  
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

Blue/green  
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the

Setting up the CodeDeploy Agent.

**Agent configuration with AWS Systems Manager** Info

**Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent.**  
Make sure that the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. [Learn more](#)

Install AWS CodeDeploy Agent

Never  
 Only once  
 Now and schedule updates

[Basic scheduler](#) [Cron expression](#)

14 Days ▾

# First deployment failed 😞

d-9WW0CHW77

C Copy deployment Retry deployment

The overall deployment failed because too many individual instances failed deployment, too few healthy instances are available for deployment, or some instances in your deployment group are experiencing problems.

**Deployment status**

Installing application on your instances  
0 of 1 instances updated ✖ Failed

**Deployment details**

Application	Deployment ID	Status
nextwork-web-deploy	d-9WW0CHW77	<span style="color: red;">✖ Failed</span>
Deployment configuration	Deployment group	Initiated by

☰ appspec.yml x install\_depend Welcome x

```
1 #!/bin/bash
2 sudo yum install tomcat -y
3 sudo yum -y install httpd
4 sudo cat <!> EOF > /etc/httpd/conf.d/tomcat_manager.conf
5 <VirtualHost *:80>
```

☰ appspec.yml x install\_depend x Welcome x

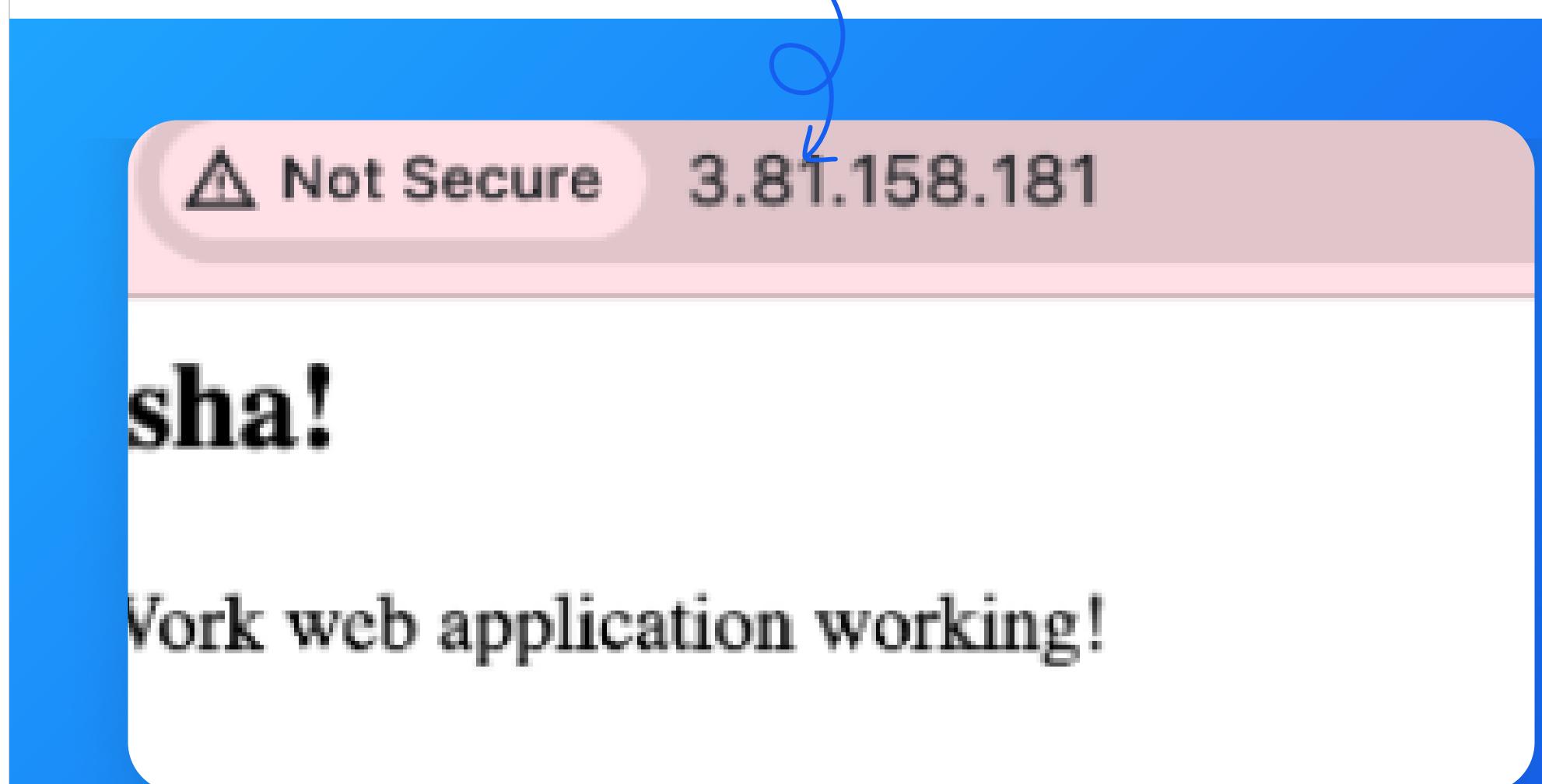
```
1 #!/bin/bash
2 sudo yum install tomcat -y
3 sudo yum -y install httpd
4 sudo cat << EOF > /etc/httpd/conf.d/tomcat_manager.conf
5 <VirtualHost *:80>
6   ServerName localhost
7   DocumentRoot /var/www/html
8   ErrorLog /var/log/httpd/error.log
9   CustomLog /var/log/httpd/access.log combined
10  
```

Reason for the error was a little syntax error 😊. I have to do the steps again manually because my CI/CD pipeline is still not implemented.

# Deployment success! 🚀

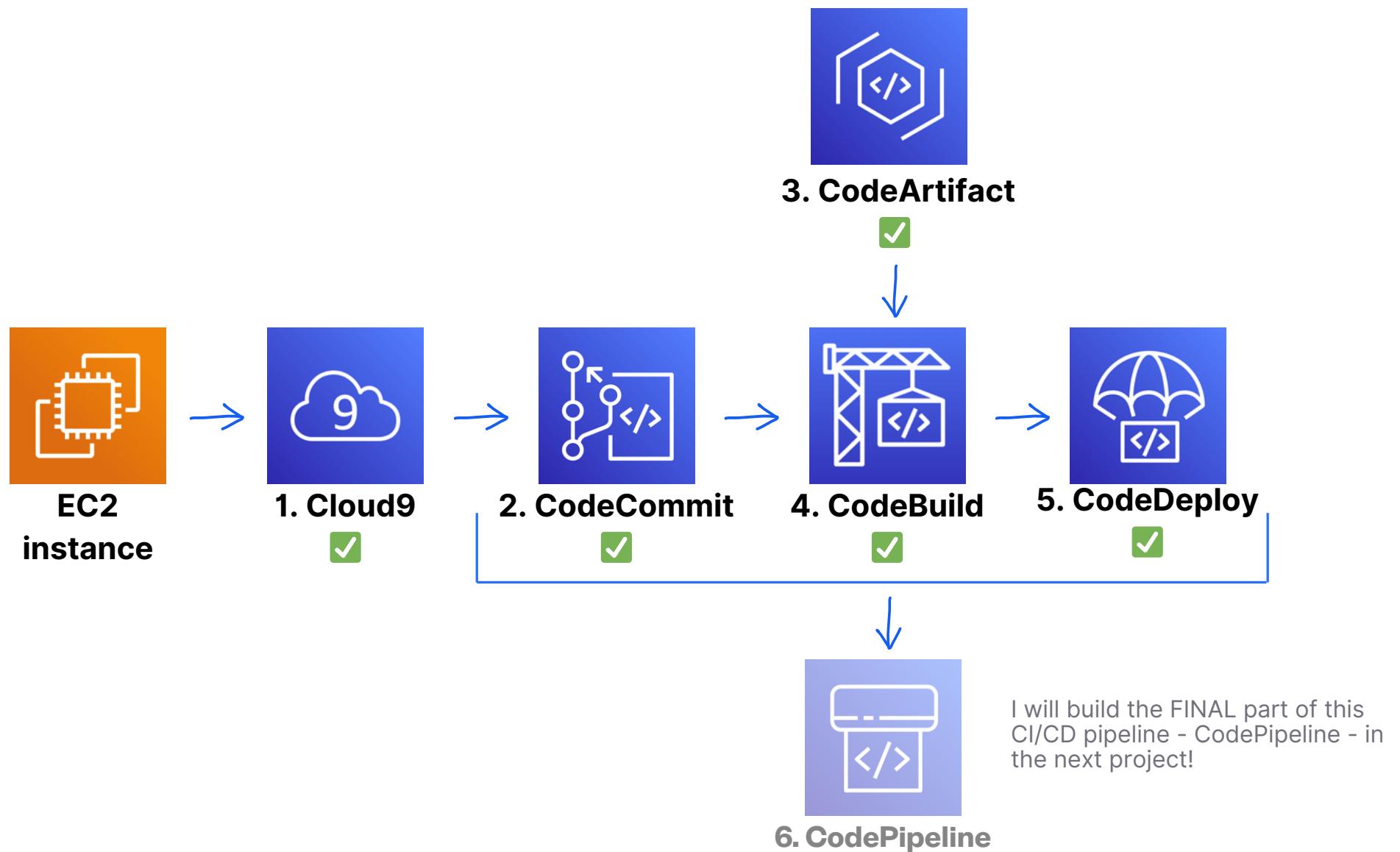
- After setting up my deployment group, I was ready to kick off my first deployment!
- To create my deployment, I had to set up a revision location, which means the location of my web app's build artefacts.
- My revision location was my Java WAR.zip file containing my compiled web app code. that was sitting in my S3 build artefact bucket.
- To visit my web app, I had to visit my web server which was in my EC2 console, as the EC2 instance's IPv4 address.

Amazing! I could see my web app taking shape.



# My CI/CD pipeline so far...

1. **AWS Cloud9** is responsible for writing running and debugging the code for my web app.
2. **AWS CodeCommit** is responsible for storing the repositories in the AWS cloud for my web app
3. **AWS CodeArtifact** is responsible for providing security for the packages and dependencies I used for the development of my web app
4. **AWS CodeBuild** is responsible for compiling the source code, run test, and packages for my web app deployment.
5. **AWS CodeDeploy** is responsible for automating the code deployment to my EC2 instance servers and bringing my web app live for public users.





# My key learnings

- 1** The deployment process means the method of building an application which includes, compiling, configuring and deploying.
- 2** I created a separate environment for deployment because so that I can run my test away from my live production environment.
- 3** To create a deployment, I had to set up different resources in AWS, which included: an EC2 instance, a VPC with CloudFormation, deployment of scripts, service roles, CodeDeploy application, and a deployment group.
- 4** One thing I didn't expect was the lengthy process it took to setup and configuration needed for deploying a webapp in AWS. But the process also helps to automate the whole build in the end.



**Everyone should be  
in a job they love. *yes!***

Check out [community.nextwork.org](http://community.nextwork.org) for more free projects

