NextWork.org

Deploy an App with CodeDeploy

Sadeesha Perera


Sadeesha Perera
linkedin.com/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 we 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.


Sadeesha Perera
linkedin.com/sadeesha-perera

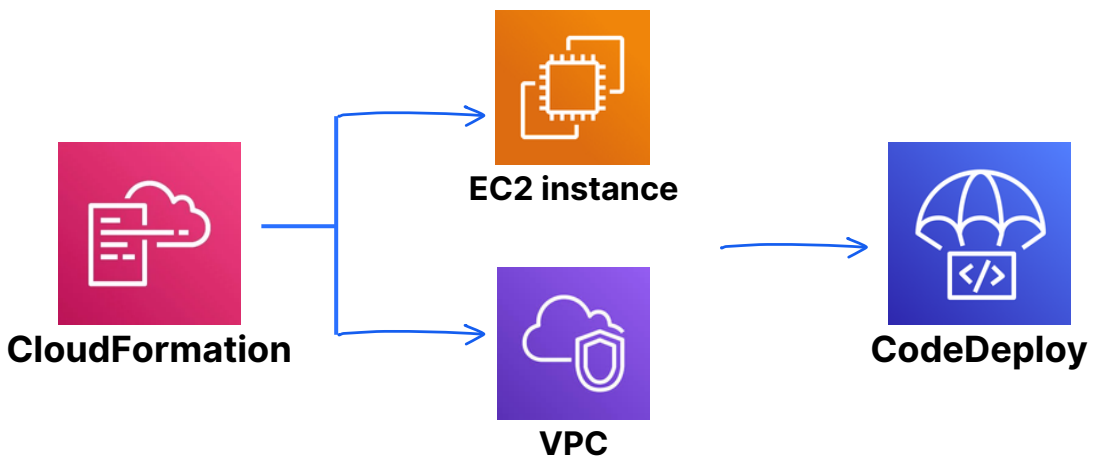
Set up an EC2 instance

- I set up an EC2 instance and VPC because deploying my web app requires a separate environment (my production environment).

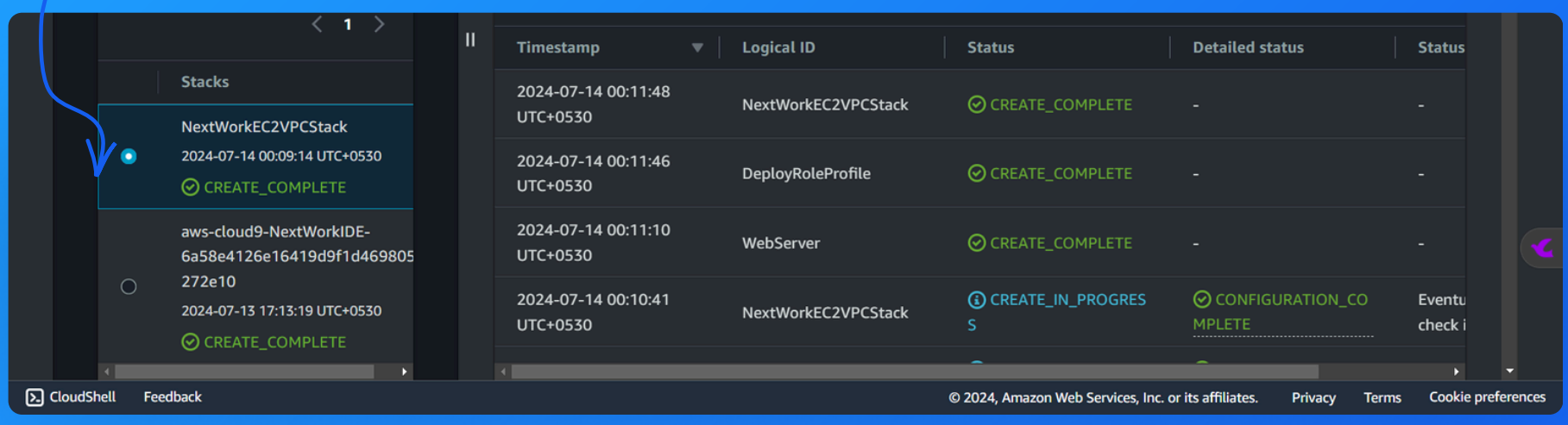
- The production environment is different to the build environment because deploying my web app will require different tools/dependencies in order to function properly. it is also best practice to separate environment for developers to test code without interfering with what users see.

- 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!




Sadeesha Perera
linkedin.com/sadeesha-perera

NextWork.org

Write bash scripts

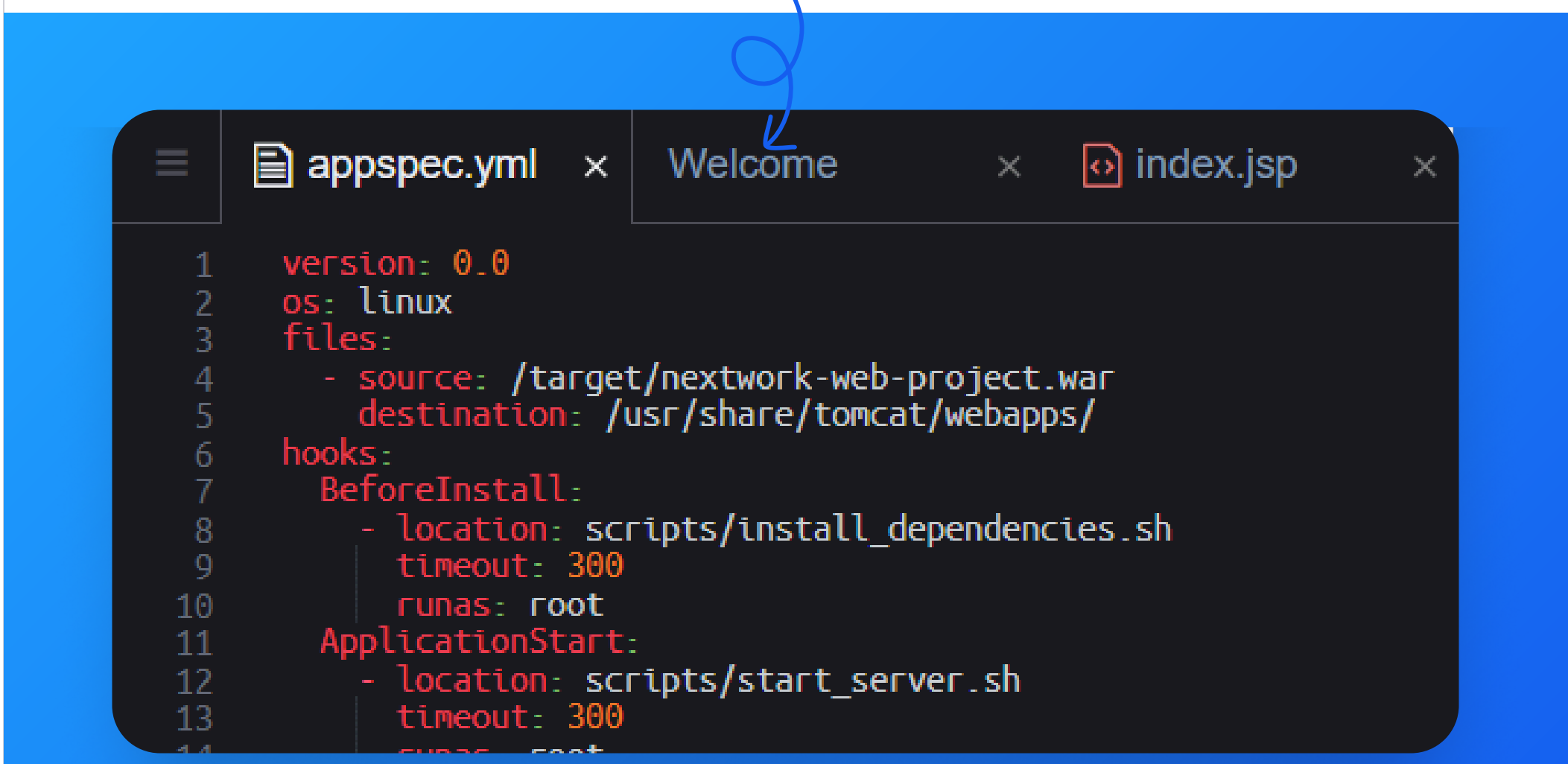
- Scripts are collections of commands in a file i.e. when you run a script, you are telling the terminal to run (line by line) what i've written in the script

- 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 serves.
c.stop_server.sh stops the web servers.

A peek into appspec.yml!

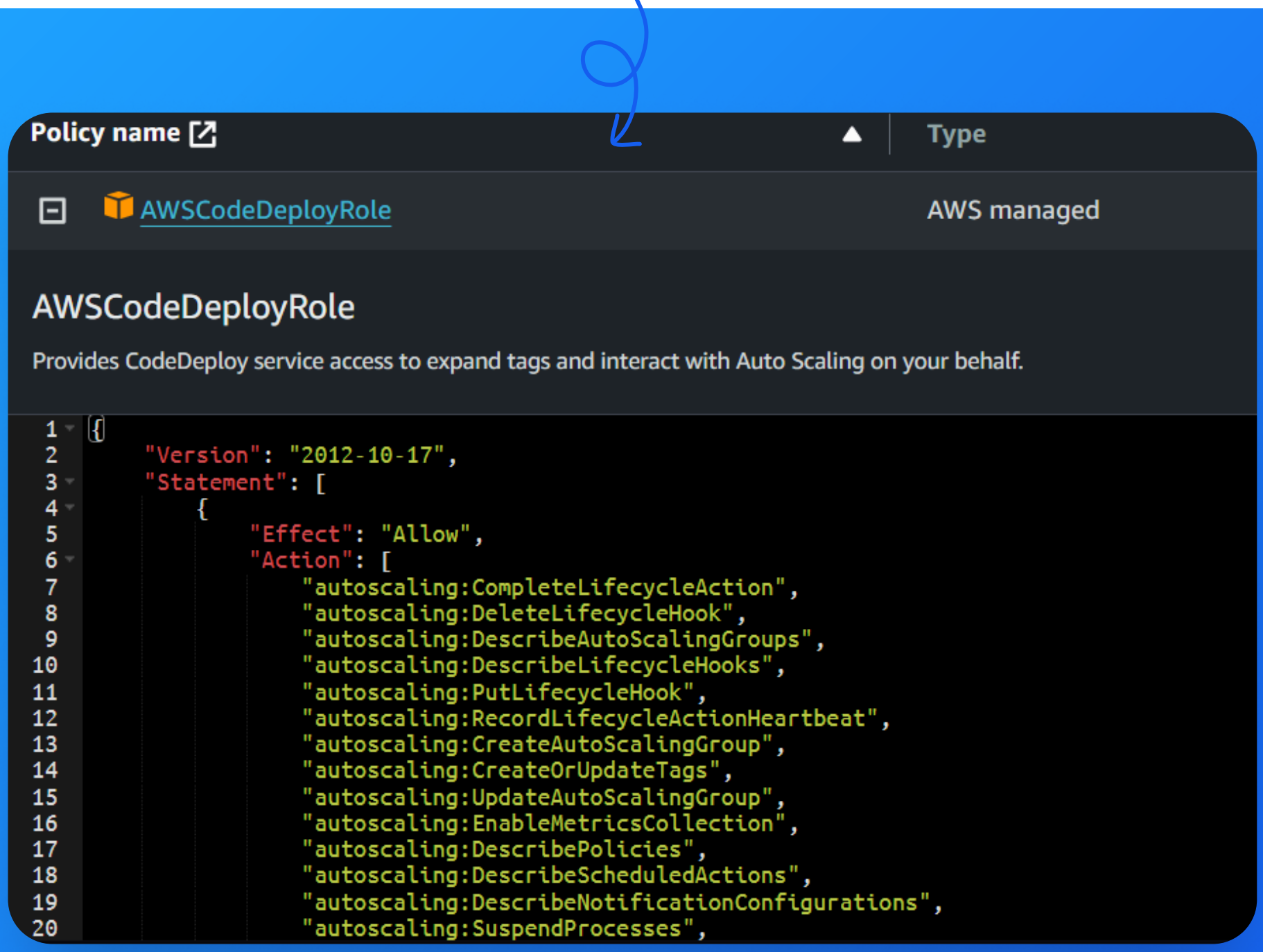



Sadeesha Perera
linkedin.com/sadeesha-perera

CodeDeploy’s IAM Role

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

The permissions granted by my CodeDeploy IAM role.




Sadeesha Perera
linkedin.com/sadeesha-perera

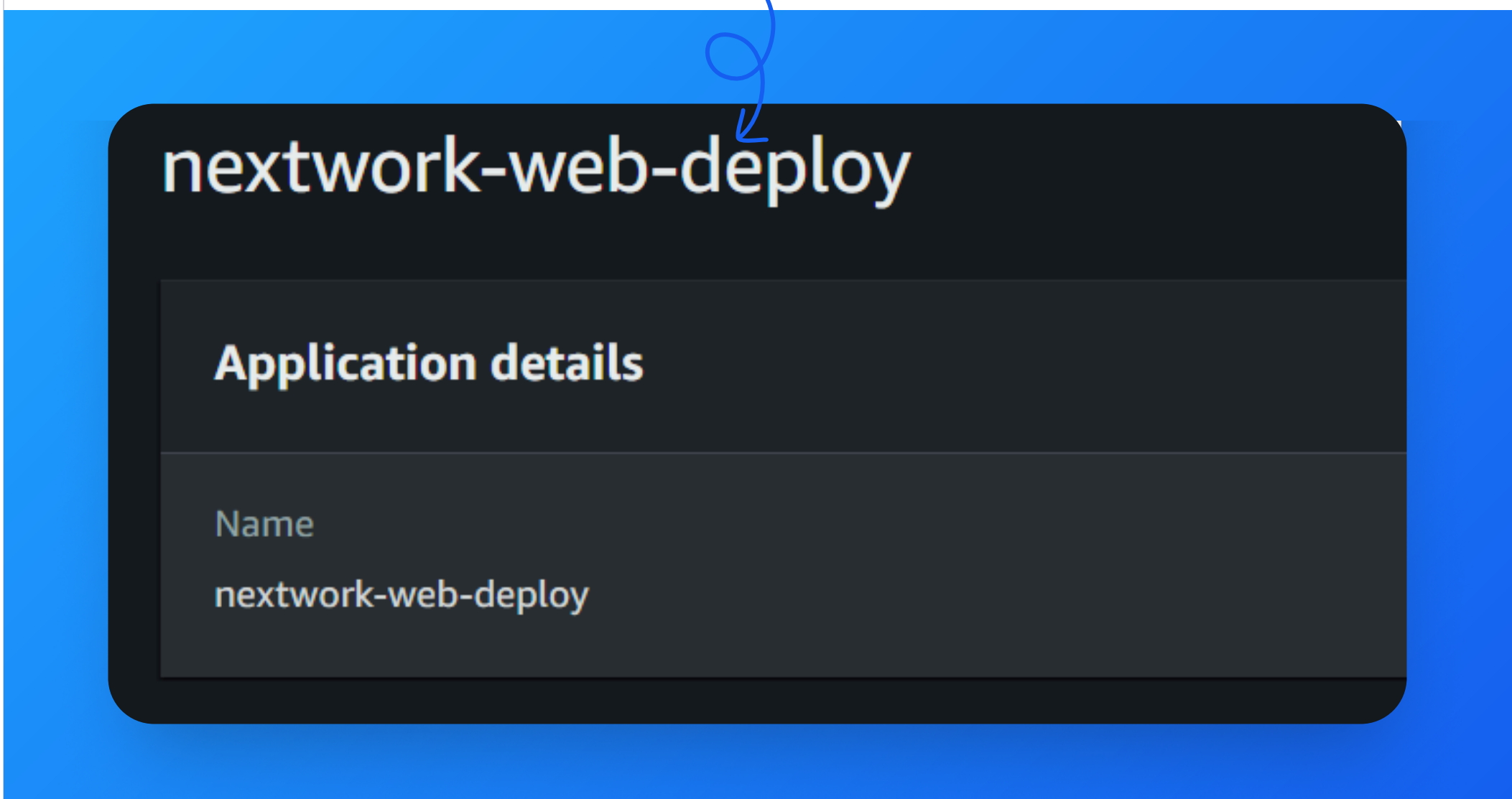
Create a CodeDeploy app

- 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 wed 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!




Sadeesha Perera
linkedin.com/sadeesha-perera

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

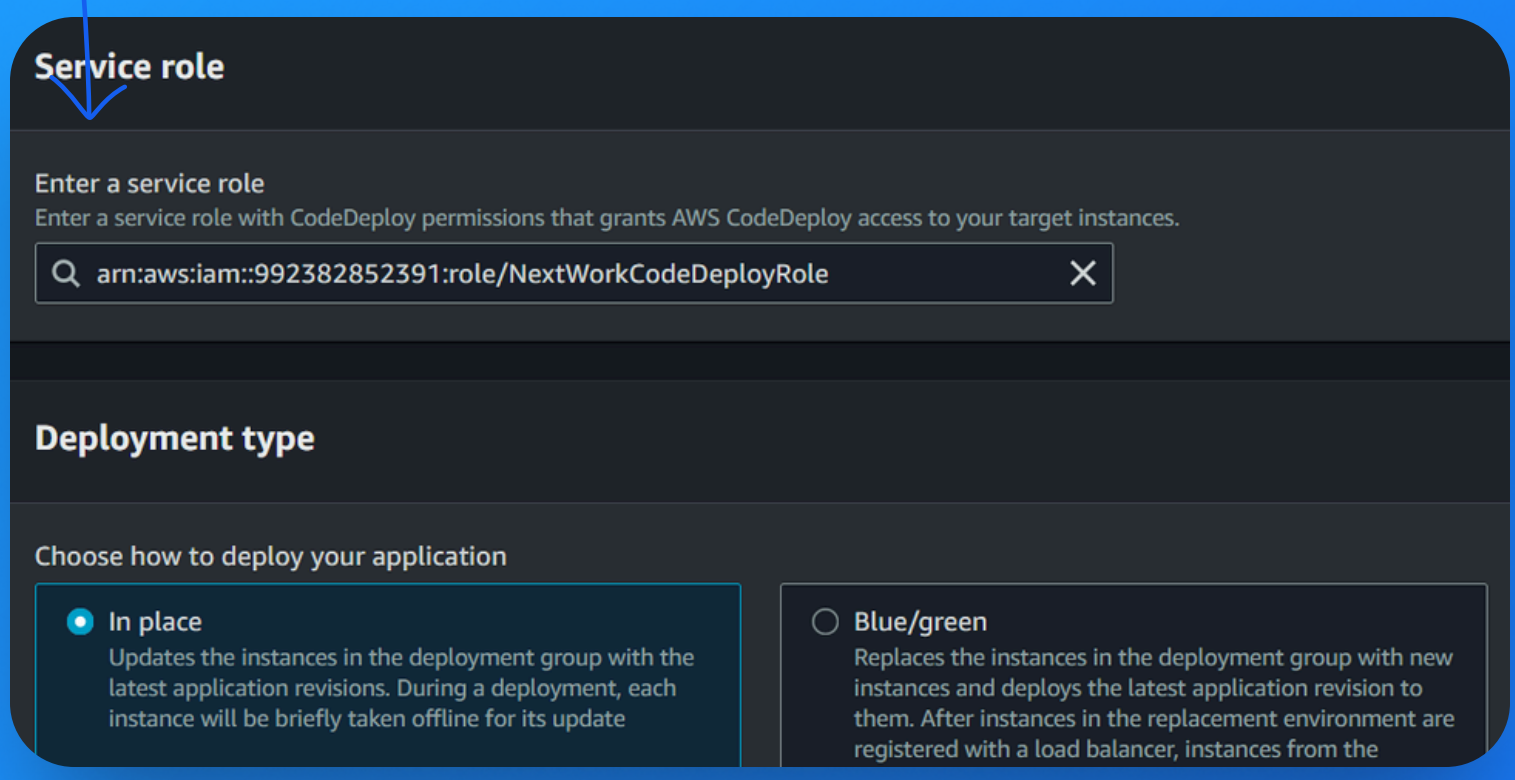
        - **CodeDeploy Agent,** which is a helper for communicating with my EC2 web server and making sure they carry out the instructions for deployment.

        - **Deployment settings**, which is whether deployment is staggered out or done all out once.

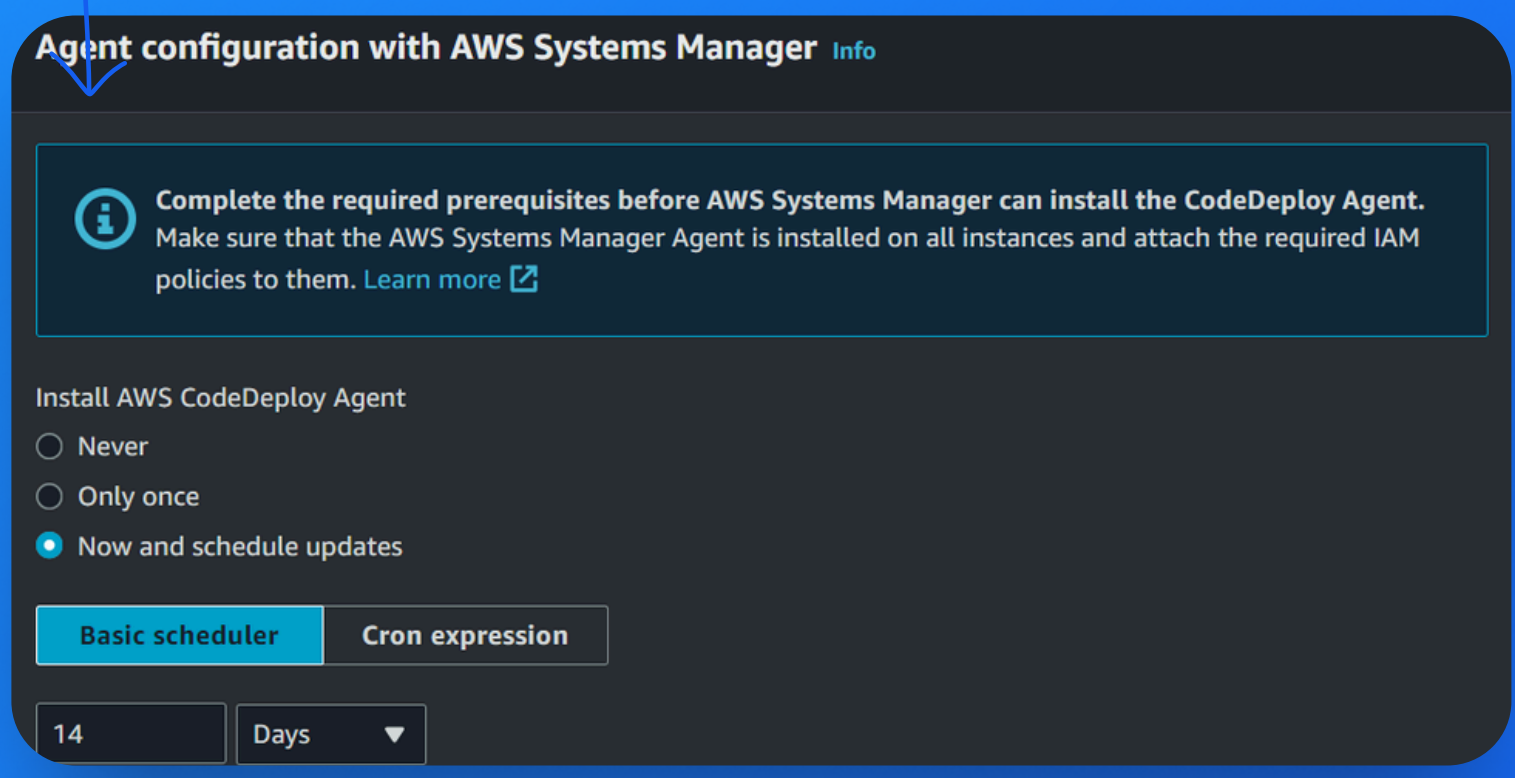
- For the load balancer setting, I disabled the load balancer.
    - This was because I only have one web server deploying my web app.

Sadeesha Perera
linkedin.com/sadeesha-perera

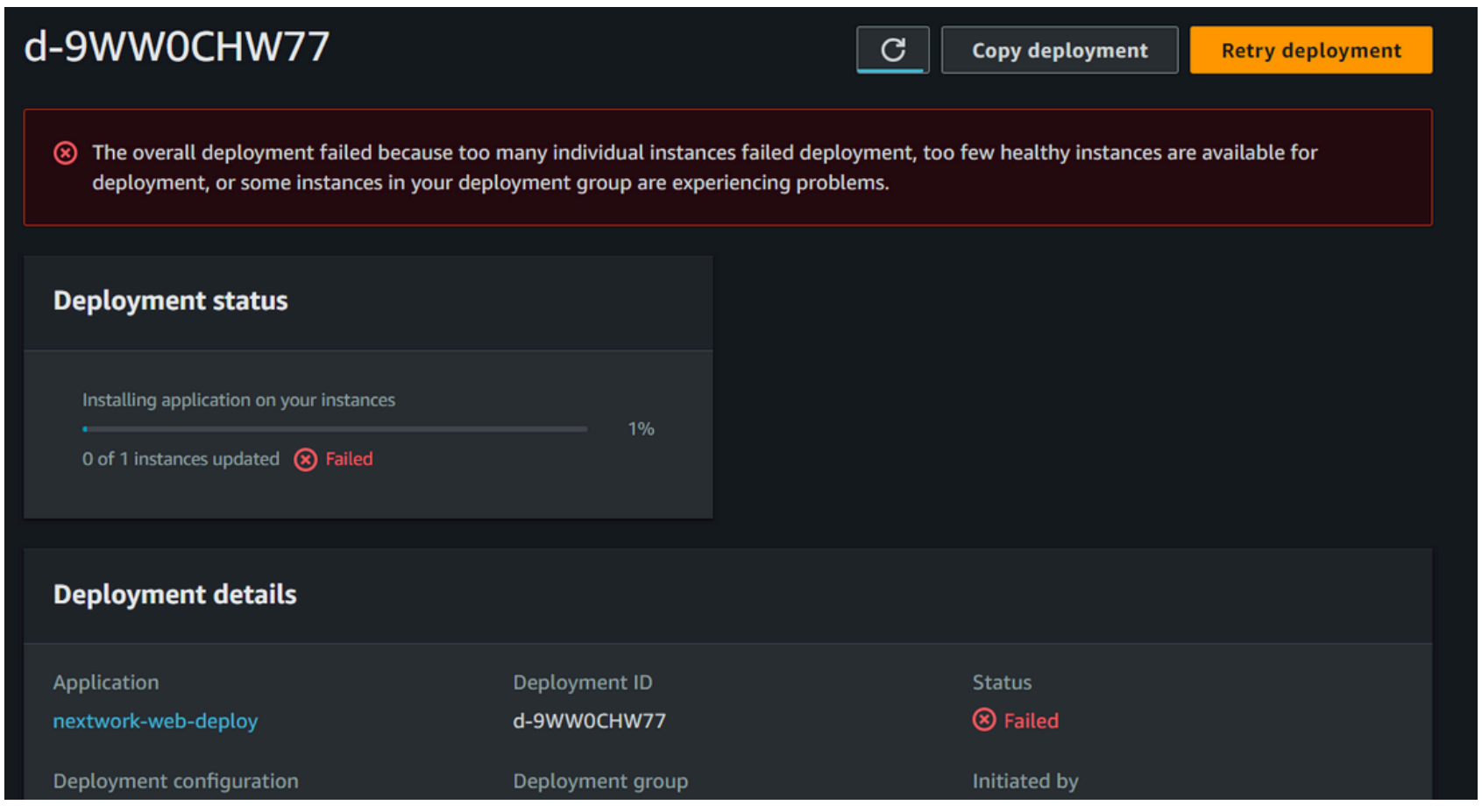
Setting up the service role + deployment type.

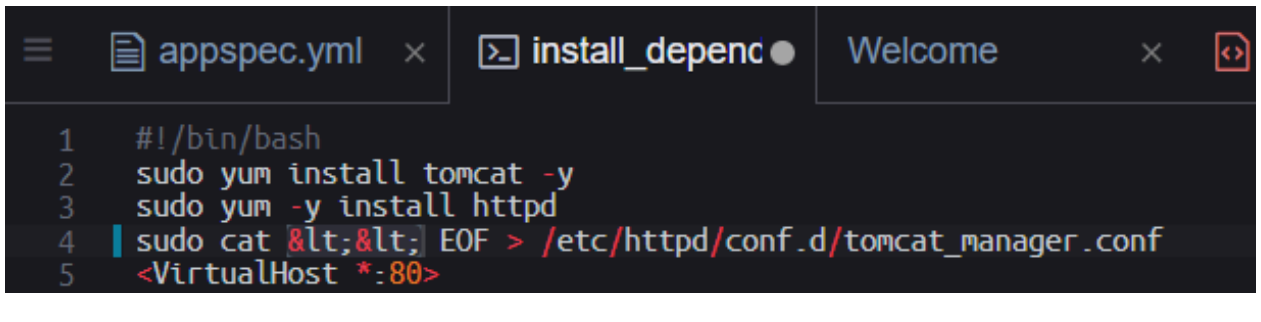


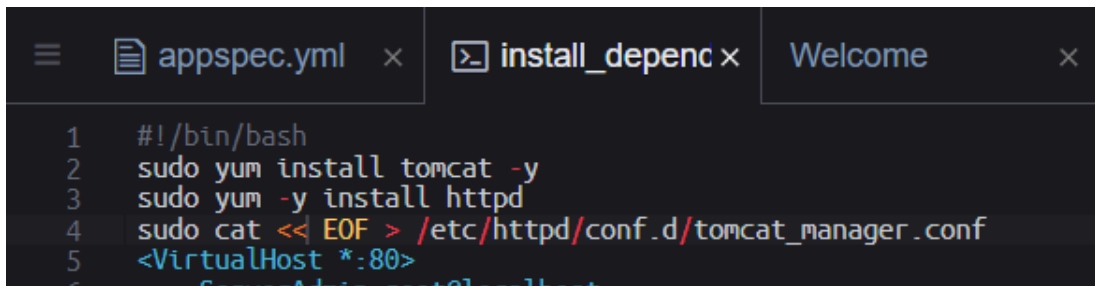
Setting up the CodeDeploy Agent.



First deployment failed 😐







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

Sadeesha Perera
linkedin.com/sadeesha-perera

Deployment success! 🚀

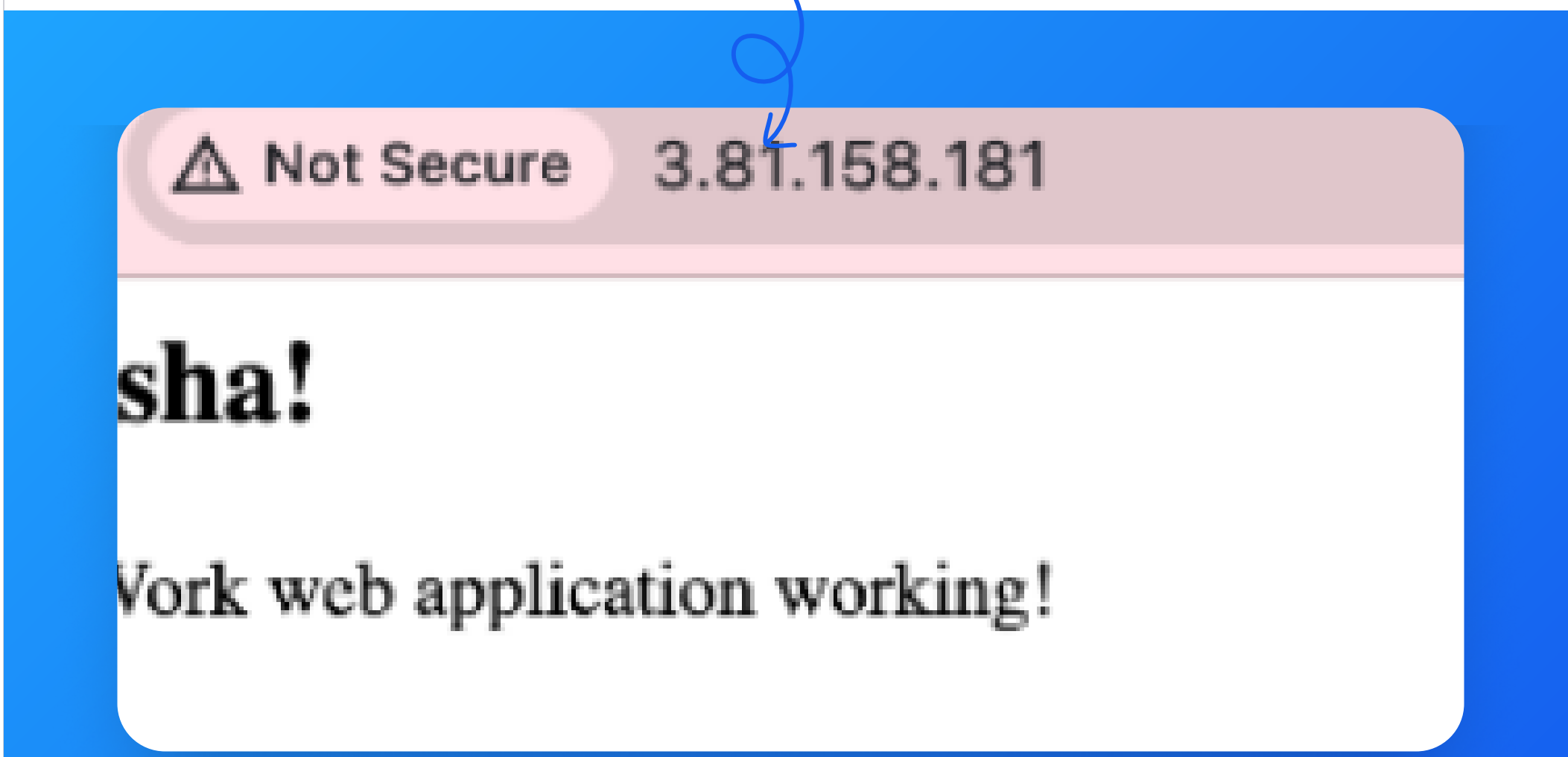
- After setting up my deployment group, I was ready to kick off my first deployment!

- 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.



Sadeesha Perera
linkedin.com/sadeesha-perera

My CI/CD pipeline so far...

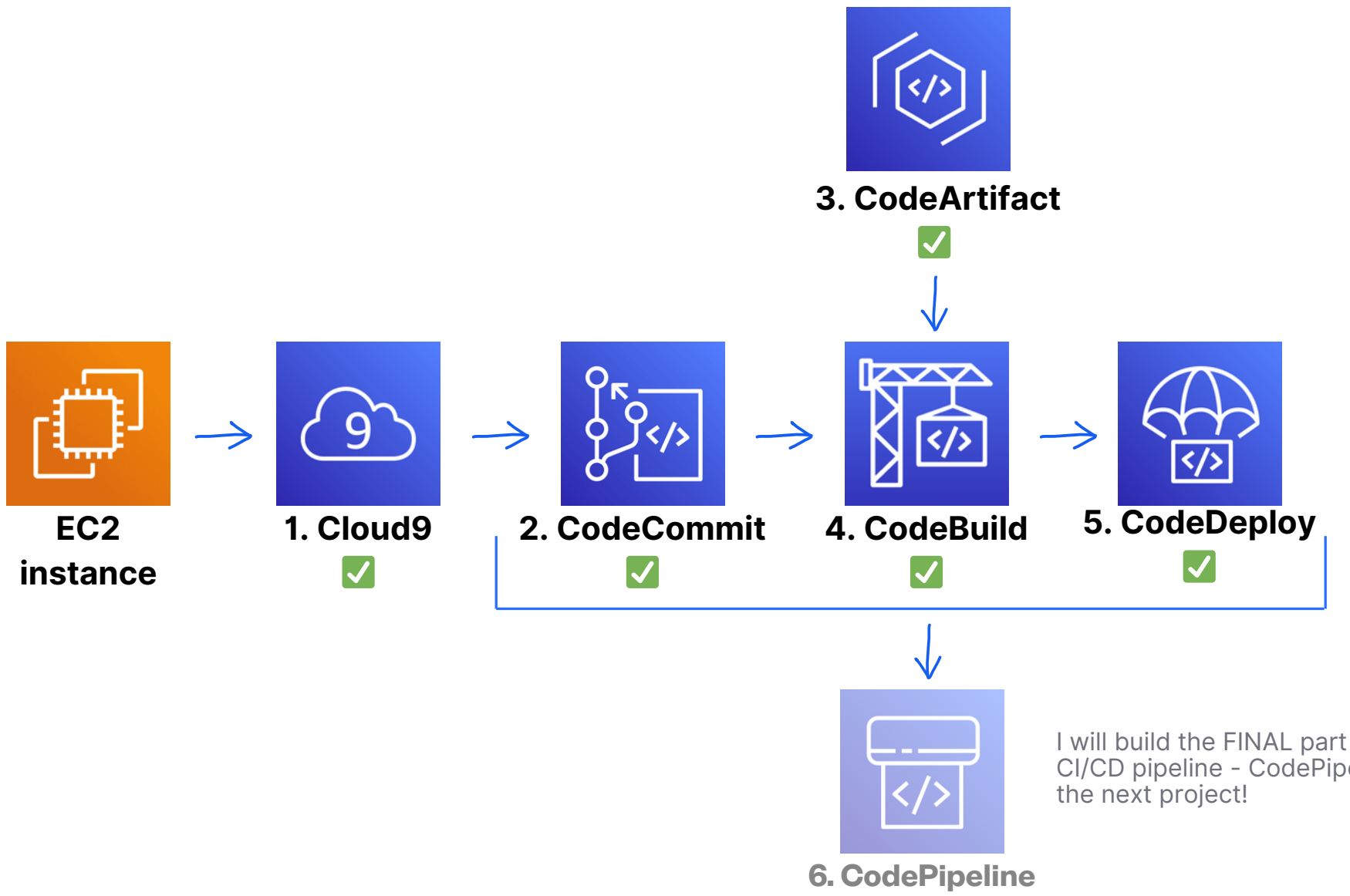
1. **AWS Cloud9** is responsible for writing running and debugging the code for my web app.

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.




Sadeesha Perera
linkedin.com/sadeesha-perera

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.

NEXTWORK

Everyone should be in a job they love. yes!

Check out community.nextwork.org for more free projects