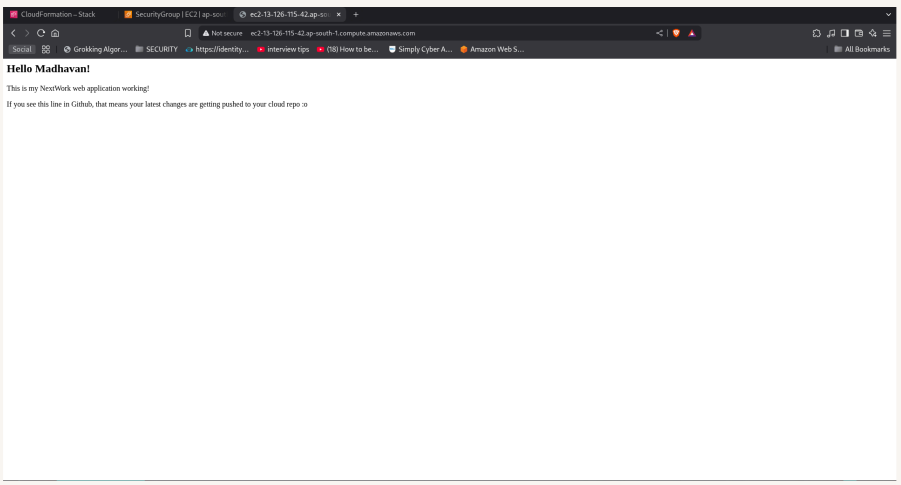
nextwork.org

Deploy a Web App with CodeDeploy

MA Harish madhavan J U



MA
Harish madhavan J U
NextWork Student
nextwork.org

Introducing Today's Project!

In this project, I will demonstrate how to setup CodeDeploy a continuous deployment service for our web app. I'm doing this project to learn about CodeDeploy and devolep muscle memory in using it.

Key tools and concepts

Services I used where CodeBuild, CodeDeploy, EC2, CloudFormation and S3. Key concepts I learnt include rollbacks, infrastructure as code, deploying application usind CodeDeploy, automating infrastructre using CloudFormation

Project reflection

This project took me approximately 3 hrs. The most challenging part was deploying using CodeDeploy and it was most rewarding to watch the website go live after modifying the inbound rules of the instance.

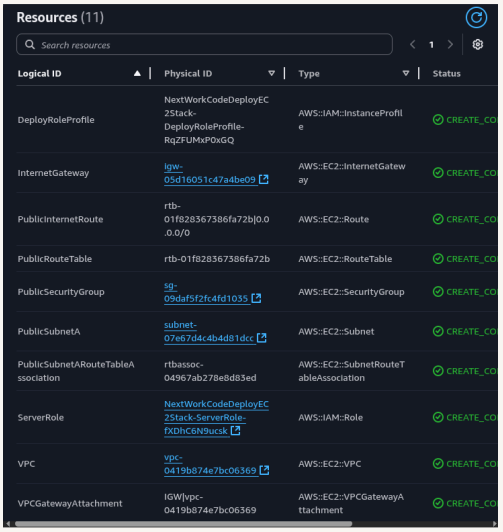
This project is part five of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow.

Deployment Environment

To set up for CodeDeploy, I launched an EC2 instance and VPC because EC2 instance acts as a production environment where we get code from CodeDeploy and execute it. VPC is used because it provides a secure infrastructure to control the access to instance and monitor both the deployment and production environment.With the setup I am creating a secure, reliable and scalable infrastructure for automating deployments.

Instead of launching these resources manually, I used CloudFormation which helps in creating infrastructure using IaC. Since this bundles all the connected resources into a stack When I need to delete these resources with one click I can delete all.

Other resources created in this template include Subnet, Internet gateway, EC2 instance, Security group and Route table. They're also in the template because they are responsible creating a secure, scalable infrastructure that is very useful in connecting different databases to the webapp and monitoring public and private traffic.


Resources (11)
Search resources
1
Logical ID
Physical ID
Type
Status
DeployRoleProfile
NextWorkCodeDeployEC2Stack-DeployRoleProfile-RqZFUMxPOxGQ
AWS::IAM::InstanceProfile
InternetGateway
igw-05d16051c47a4be09
AWS::EC2::InternetGateway
PublicInternetRoute
rtb-01f828367386fa72bj0.0.0.0/0
AWS::EC2::Route
PublicRouteTable
rtb-01f828367386fa72b
AWS::EC2::RouteTable
PublicSecurityGroup
sg-09daf5f2fc4fd1035
AWS::EC2::SecurityGroup
PublicSubnetA
subnet-07e67d4c4b4d81dcc
AWS::EC2::Subnet
PublicSubnetARouteTableAssociation
rtbassoc-04967ab278e8d83ed
AWS::EC2::SubnetRouteTableAssociation
ServerRole
NextWorkCodeDeployEC2Stack-ServerRole-fXDhC6N9ucsk
AWS::IAM::Role
VPC
vpc-0419b874e7bc06369
AWS::EC2::VPC
VPCGatewayAttachment
IGW|vpc-0419b874e7bc06369
AWS::EC2::VPCGatewayAttachment

Deployment Scripts

Scripts are small pieces of code used to automate small tasks by bundling all relevant commands in the order to be executed. To set up CodeDeploy I also wrote scripts to install dependencies, start the server and stop the same.

install_dependencies installs Apache HTTP Server, which is used to host the web server, and Tomcat, which runs the Java web application. Apache acts as a reverse proxy, forwarding incoming HTTP requests to Tomcat, which processes the application logic and responses.

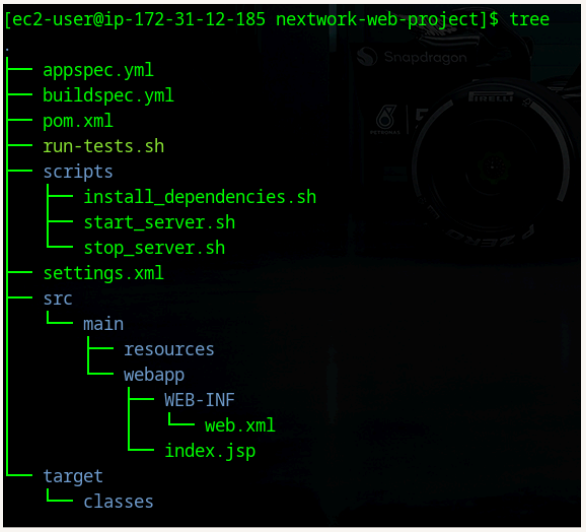
start_server.sh starts both Tomcat (Java application server) and Apache HTTP Server (web server). It also ensures these services automatically restart when the instance reboots. Service management and control is handled using systemctl, a command-line tool in Linux that interacts with the systemd init system.

stop_server.sh stops the Apache and Tomcat services if they are currently running. It's important to check whether the services are active before attempting to stop them—otherwise, attempting to stop a service that isn't running can produce errors, which may disrupt the deployment process.

appspec.yml

Then, I wrote an appspec.yml file to guide codedeploy on how to deploy the web app. The key sections in appspec.yml are BeforeInstall - helps with dependencies to install, AfterInstall - helps with starting the server and ApplicationStart - used to stop the server.

I also updated buildspec.yml because it gives instructions to Codebuild to create a deployable package, the section I modified was artifacts which tells codebuild include the mentioned files along with the package which is required for deployment.

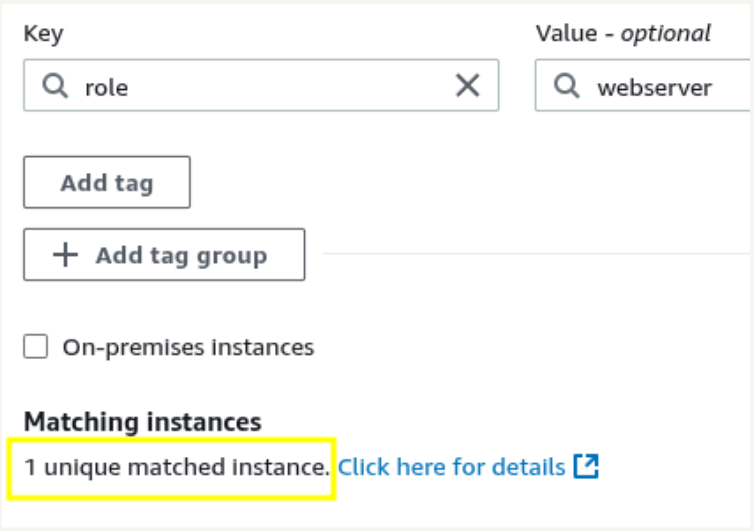


Setting Up CodeDeploy

A deployment group is a collection of EC2 instances that are grouped together to deploy something together and contains info related to where and how to deploy. A codedeploy application is a logical container for a application with different deployment groups with each serving different purpose.

To set up a deployment group, you also need to create an IAM role to grant necessary permission to work with your instances, permissions are strictly confined to working with necessary services.

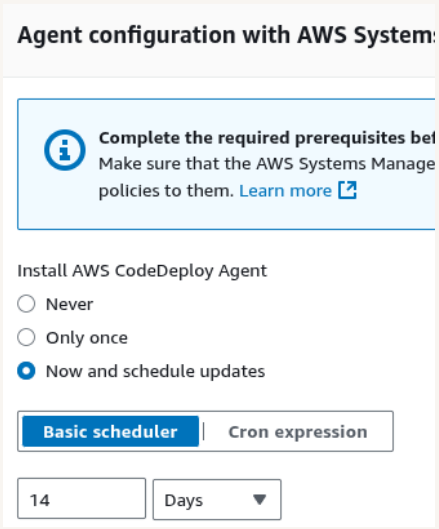
Tags are helpful for CloudDeploy to identify the target instances, we can also deploy an application to multiple instance by adding a tag. I used the tag role: webserver to deploy the web application.


Key
role
Value - optional
webserver
Add tag
Add tag group
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 time taken to roll out an applicatino. I used CodeDeployDefault.AllAtOnce, so my application is deployed in the instance all at once, its riskier to go with this option when you have hundereds of instances because if anything goes wrong, all the instances will be affected.

In order to connect CodeDeploy and EC2 instance a CodeDeploy Agent is also set up to recieve bash scripts(instructions) from codedeploy to deploy the application and the agent is responsible for communication between EC2 and CodeDeploy.

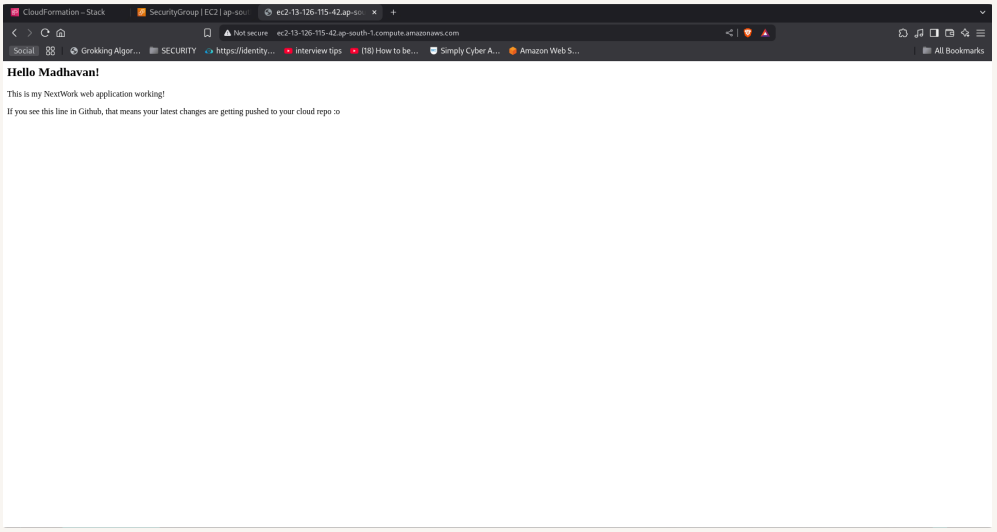


Success!

A CodeDeploy deployment is a process through which we can install new version of the application in EC2 instance. The difference to a deployment group is it tells where to install the updates/application whereas deployment includes information related to what, where and how to deploy.

I had to configure a revision location, which means the location of the artifact from which CodeDeploy has to deploy the application. My revision location was an S3 bucket because we used CodeBuild to store the artifact in that bucket.

To check that the deployment was a success, I visited the public IP of my production environment's EC2 instance. I saw a live website.

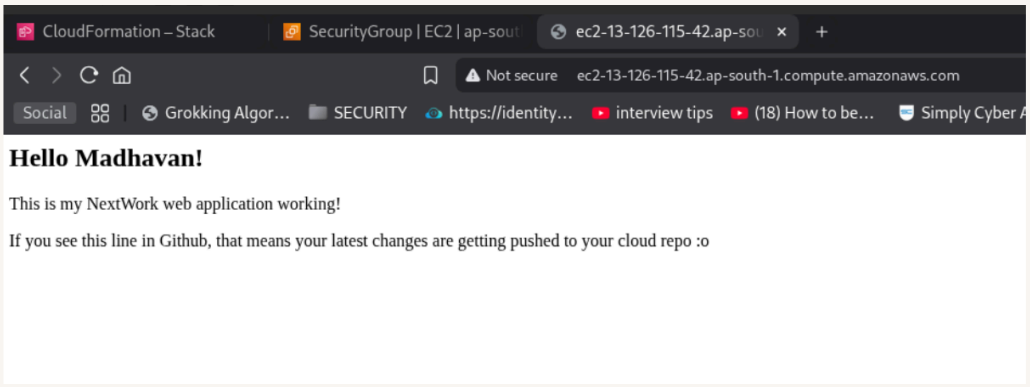

CloudFormation – Stack
SecurityGroup | EC2 | ap-sou
ec2-13-126-115-42.ap-so
Not secure
ec2-13-126-115-42.ap-south-1.compute.amazonaws.com
Social
Grokking Algor...
SECURITY
https://identity...
interview tips
(18) How to be...
Simply Cyber A...
Amazon Web S...
All Bookmarks
Hello Madhavan!
This is my NextWork web application working!
If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :o

Disaster Recovery

In a project extension, I decided to introduce a intentional error so that the deployment could fail. The intentional error I created was to introduce a typo in stop_server.sh file. This will cause the deployment to fail because the typo introduced will result in malformed command which does not exist.

I also enabled rollbacks with this deployment, which means if the deployment fails CodeDeploy will safely rollback the application to the previous version. This is valuable because it reduces downtime when the deployment fails.

When my deployment failed, the automatic rollback initiated a deployment using the previous successful deployment configuration, because that's how rollback works in CodeDeploy. However, to actually recover from the failed deployment, I had to fix the code and create a new version of the artifact by rebuilding it using CodeBuild. In production environments, to implement true automatic rollbacks, we can use more advanced tools like AWS CodePipeline, which can roll back using the previous successful deployment artifact, not just the configuration.


CloudFormation – Stack
SecurityGroup | EC2 | ap-sout
ec2-13-126-115-42.ap-sou
Not secure
ec2-13-126-115-42.ap-south-1.compute.amazonaws.com
Social
Grokking Algor…
SECURITY
https://identity…
interview tips
(18) How to be…
Simply Cyber
Hello Madhavan!
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

The place to learn & showcase your skills

Check out nextwork.org for more projects