



Deploy an App with CodeDeploy



Prerana P

```
File Edit Selection View Go Run ... <- > SSH: ec2-15-207-89-214.ap-south-1.compute.amazonaws.com [SSH] | Find Aa No results ↑ ↓ ⌂ ...
```

```
NEXTWORK-WEB-PROJECT... / appspec.yml
```

```
scripts:
  install_dependencies:
    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
```

```
DIRECTORY
```

```
No symbols found in document "appspec.yml"
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
bash - nextwork-web-project + v ⌂ ...
```

```
SSH: ec2-15-207-89-214.ap-south-1.compute.amazonaws.com [SSH] | Find Aa No results ↑ ↓ ⌂ ...
```

```
git clone https://github.com/PreranaCI/CI_CD.git
cd CI_CD
git checkout main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-172-31-13-99 nextwork-web-project]$
```

In 19 Col 1 Spaces: 2 UTF-8 LF YAML ⌂



Introducing today's project!

What is AWS CodeDeploy?

AWS CodeDeploy is a fully managed service that automates application deployment to EC2 instances, Lambda, or on-prem servers. It ensures consistent, scalable, and reliable deployments, reduces manual errors, and enables rollback in case of failure, e

How I'm using AWS CodeDeploy in this project

In today's project, I used AWS CodeDeploy to automate the deployment of an application to EC2 instances. I set up a CodeDeploy application and deployment group, created an IAM role with the necessary permissions, and specified the revision location i

One thing I didn't expect...

One thing I didn't expect in this project was that CodeDeploy wouldn't recognize the EC2 instance. Despite having the correct IAM roles and configurations, the deployment group didn't find the instance, likely due to missing or incorrect tags or the

This project took me...

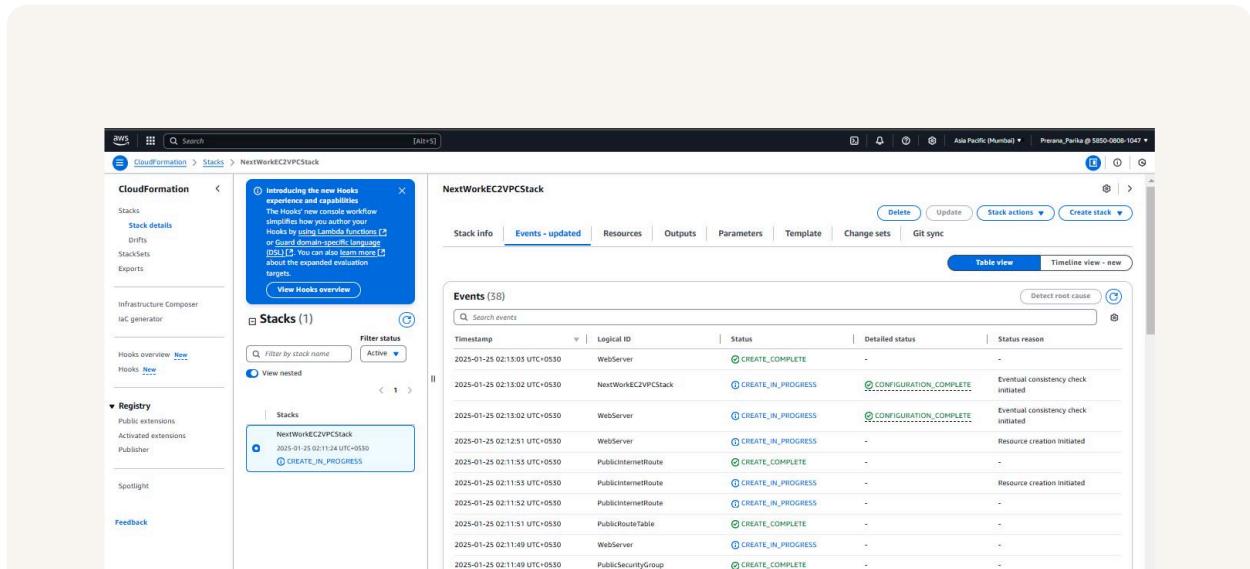
Around 1hour

Set up an EC2 instance

I set up an EC2 instance and VPC because it provides a secure and scalable environment for deploying applications. The EC2 instance allows for computing power, while the VPC ensures isolated network configuration and control over resources within AWS.

We manage production and development environments separately because it ensures stability and security in production while allowing for experimentation and testing in development. This separation helps prevent disruptions and ensures that changes are

To set up my EC2 instance and VPC, I used AWS CloudFormation. This service allows me to define the infrastructure as code, ensuring consistent, repeatable deployments of both the EC2 instance and VPC with minimal manual configuration.





Bash scripts

Scripts are files that contain a series of commands or instructions written in a programming or scripting language, such as Bash. Bash is a command-line interface and scripting language used in Unix-like operating systems to automate tasks and manage

I used three scripts for my project's deployment

The first script I created was to install and configure Tomcat and Apache HTTP server. It installs Tomcat and HTTPD, then creates a custom Apache configuration file (tomcat_manager.conf) to set up a reverse proxy to Tomcat on port 8080 for the Nexttwo

The second script I created was to start and enable the Tomcat and Apache HTTPD services. It ensures that both services are started and automatically begin running on system boot by using systemctl to manage the services.

The third script I created was to stop the Apache HTTPD and Tomcat services if they are running. It checks if each service is active using pgrep, and if found, it stops the corresponding service with systemctl. This helps in safely shutting down the



Prerana P
NextWork Student

NextWork.org

Bash scripts

```
File Edit Selection View Go Run ... ← → / nextwork-web-project [SSH: ec2-15-207-89-214.ap-south-1.compute.amazonaws.com]
EXPLORER index.jsp settings.xml buildspec.yml install_dependencies.sh start_server.sh stop_server.sh pom.xml appspec.yml ...
NEXTWORK-WEB-PROJECT...
scripts
install_dependencies.sh
start_server.sh
stop_server.sh
src/main
resources
webapp
WEB-INF
index.jsp
target
appspec.yml
buildspec.yml
pom.xml
settings.xml

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4) 12 bytes | 992.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/prerana/AWS_CI_CD.git
   6979fc..a08393  main -> main
branch 'main' set up to track 'origin/main'.
SSH: ec2-15-207-89-214.ap-south-1.compute.amazonaws.com [ec2-user@ip-172-31-13-99 nextwork-web-project]$ 
```



Prerana P

NextWork Student

NextWork.org

CodeDeploy's IAM Role

I created an IAM service role for CodeDeploy because CodeDeploy needs permission to interact with EC2 instances to deploy applications. The role, using the AWS Managed AWSCodeDeployRole policy, grants the necessary access for seamless deployment to E

To set up CodeDeploy's IAM role, I created a new IAM role in the IAM console and selected the "CodeDeploy" service as the trusted entity. Then, I attached the AWS managed policy `AWSCodeDeployRole` to the role to grant the necessary permissions for dep

The screenshot shows the AWS IAM Policy Editor interface. The left sidebar navigation includes 'Identity and Access Management (IAM)', 'Access management' (with 'User groups', 'Users', 'Roles', and 'Policies' selected), 'Identity providers', 'Account settings', 'Root access management', and 'Access reports'. The main content area displays a JSON policy titled 'AWSCodeDeployRole' with the following content:

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": [
7          "autoscaling:CompleteLifecycleAction",
8          "autoscaling:DeleteLifecycleHook",
9          "autoscaling:DescribeLifecycleHooks",
10         "autoscaling:DescribeLifecycleHooks",
11         "autoscaling:PutLifecycleHook",
12         "autoscaling:PutScalingPolicyHeartbeat",
13         "autoscaling:CreateAutoScalingGroup",
14         "autoscaling:DeleteAutoScalingGroup",
15         "autoscaling:UpdateAutoScalingGroup",
16         "autoscaling:EnableMetricsCollection",
17         "autoscaling:PutScalingPolicy",
18         "autoscaling:DescribeScheduledActions",
19         "autoscaling:DescribeNotificationConfigurations",
20         "autoscaling:PutNotificationConfiguration",
21         "autoscaling:ResumeProcesses",
22         "autoscaling:AttachLoadBalancerTargetGroups",
23         "autoscaling:PutScalingPolicy",
24         "autoscaling:PutScalingPolicy",
25         "autoscaling:PutNotificationConfiguration",
26         "autoscaling:PutNotificationConfiguration",
27         "autoscaling:PutWarmPool",
28         "autoscaling:DescribeAutoScalingGroups",
29         "autoscaling:DeleteAutoScalingGroup",
30         "ec2:DescribeInstances",
31         "ec2:TerminateInstances",
32         "ec2:TerminateInstances",
33         "tag:GetTags",
34         "sns:Publish",
35         "cloudwatch:DescribeMetrics",
36         "cloudwatch:DescribeMetrics",
37         "elasticloadbalancing:DescribeLoadBalancerAttributes",
38         "elasticloadbalancing:DescribeTargetGroupAttributes",
39         "elasticloadbalancing:DescribeTargetGroupAttributes",
40         "elasticloadbalancing:DescribeInstanceHealth",
41       ],
42     }
43   ]
44 }
```

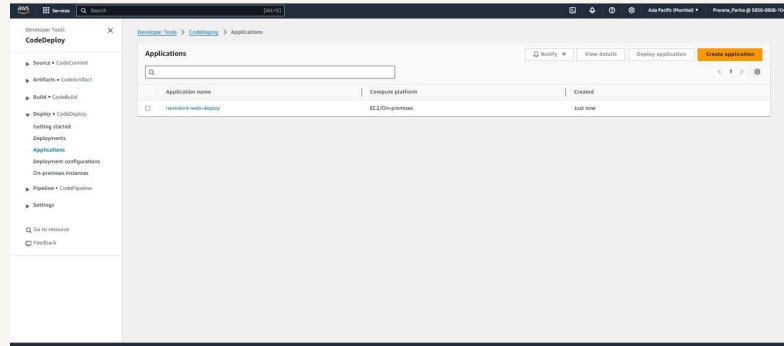
The top right of the editor has buttons for 'Copy', 'Summary', and 'JSON'. The bottom right of the page has links for 'CloudShell', 'Feedback', and copyright information.

CodeDeploy application

A CodeDeploy application means a logical entity within AWS CodeDeploy that represents the deployment of code to one or more compute resources, such as EC2 instances or Lambda functions. It defines the deployment settings, configurations, and target e

To create a CodeDeploy application, I had to select a compute platform, which means choosing the type of resource where the application will be deployed. This could be EC2 instances, on-premises servers, or AWS Lambda functions, depending on the appl

The compute platform I chose was EC2 because I am deploying the application to EC2 instances, which provides the necessary compute resources and scalability for the application. EC2 instances offer flexibility and control over the environment for run



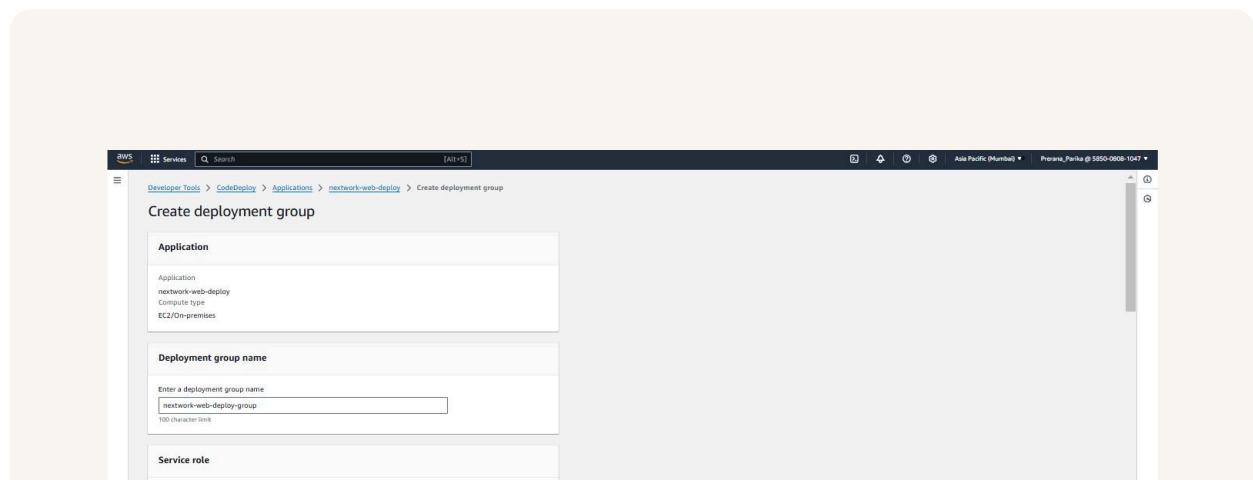
Deployment group

A deployment group means a set of EC2 instances or on-premises servers that CodeDeploy targets for deployment. It allows you to organize instances based on tags, Auto Scaling groups, or manually defined groups, ensuring that the correct instances rec

Two key configurations for a deployment group

Environment means the specific settings and configurations that define the target deployment infrastructure for an application, such as the compute platform, deployment group, and other resources. It ensures the proper setup for where and how the ap

A CodeDeploy Agent is a software component installed on the target EC2 instances or on-premises servers. It communicates with the CodeDeploy service, enabling the deployment of application revisions to those instances. The agent ensures the proper in



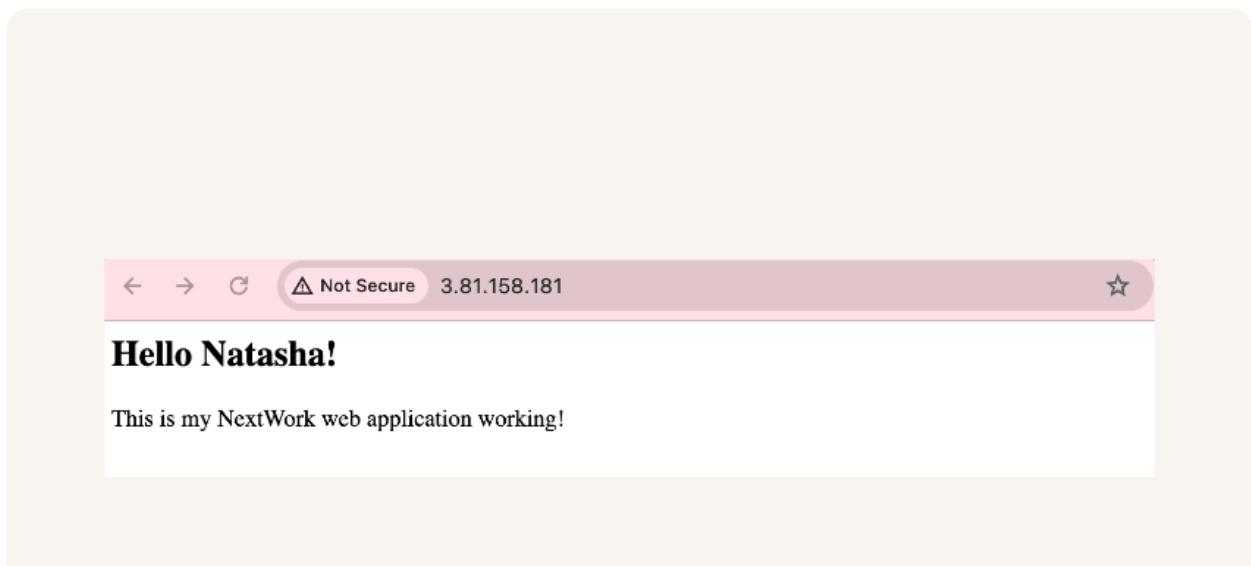
A circular profile picture of a young woman with dark hair, wearing a white lace-trimmed top.

CodeDeploy application

To create my deployment, I had to set up a revision location, which means specifying the location of the application's source code or deployment package. This could be a file stored in an Amazon S3 bucket or a GitHub repository that CodeDeploy will u

My revision location was my Amazon S3 bucket where I uploaded the application's deployment package. The S3 URI pointed to the specific location of the zipped application code or artifact that CodeDeploy would retrieve and deploy to the target EC2 ins

To visit my web app, I had to visit the public IP address or DNS name of the EC2 instance that was running the application. This address, combined with the appropriate port (e.g., 80 for HTTP), provided the link to access the deployed web app.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

