

# Module 2: SDLC Automation

---

## Demo Document 2

edureka!

**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

## Deploy an application using Codepipeline

### Demo steps:

#### Step 1: Create an AWS CodeCommit Repository

- In your AWS Management Console, search for CodeCommit and select it
- Click on create repository
- Give a name and description for your repository
- Click on create repository

### Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

#### Repository settings

Repository name

100 characters maximum. Other limits apply.

Description - *optional*

1,000 characters maximum

CancelCreate

#### Step 2: Add a file to your repository

- Download the file from [https://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp\\_Linux.zip](https://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip)
- Extract the file and save it in the directory where you have cloned the repository in your local system

- Upload the extracted file in

```
|my-demo-repo
|-- appspec.yml
|-- index.html
|-- LICENSE.txt
`-- scripts
    |-- install_dependencies
    |-- start_server
    `-- stop_server
```

### Step 3: Create an Amazon EC2 Instance and Install the AWS CodeDeploy Agent

- In your EC2 services, click on Launch Instance
- Choose an Amazon Linux AMI
- In instance type choose the t2.micro
- In configure instance details, under IAM role and click on create a new IAM role
- Create a new IAM role with the policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

- In advanced details of configure instance details, copy the below code

```
#!/bin/bash
yum -y update
yum install -y ruby
yum install -y aws-cli
yum install wget
cd /home/ec2-user
wget https://bucket-name.s3.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto
```

- To know the name of the bucket, which is in your region  
<https://docs.aws.amazon.com/codedeploy/latest/userguide/resource-kit.html#resource-kit-bucket-names>
- Add storage and attach security groups to it
- Click on review and launch

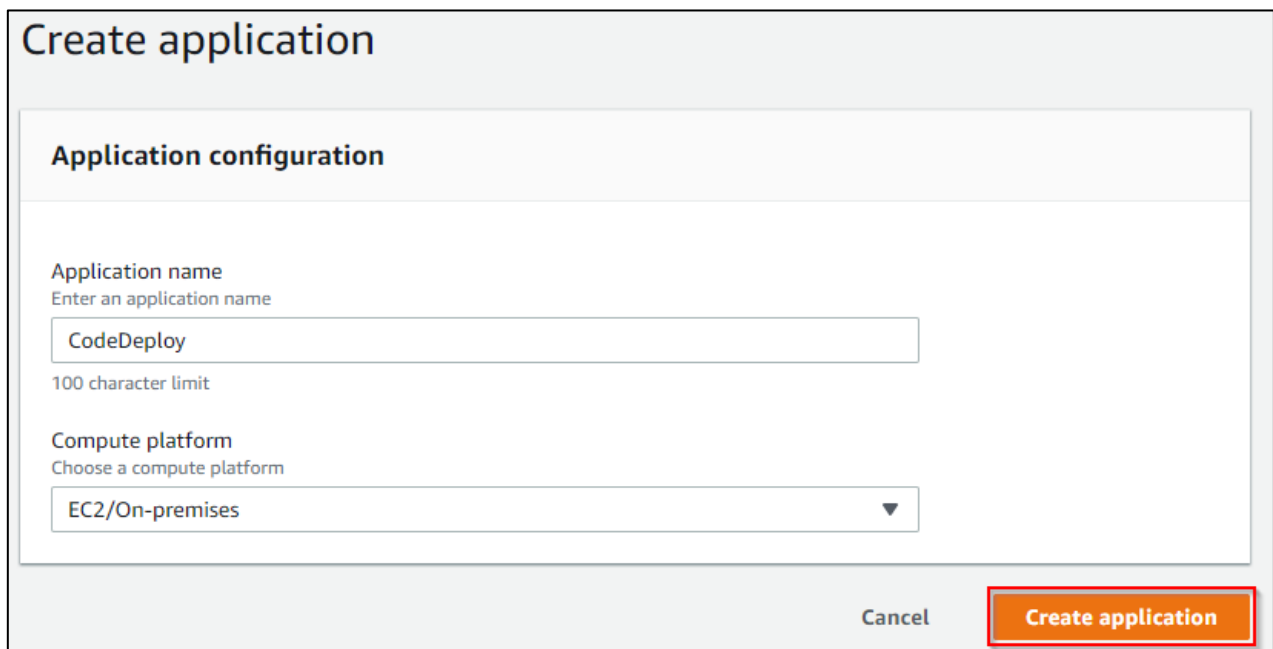
#### Step 4: Create an IAM role for CodeDeploy

- In IAM dashboard, click on roles
- Click on create roles
- Choose the services to be CodeDeploy and click on next permission
- In Attached permissions policy, choose AWSCodeDeployRole policy and click on review
- Give a name for your role and click on create role
- Once the role is created go to trusted relationship tab, click on edit trust relationship

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codedeploy.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

### Step 5: Create an Code deploy application

- In your AWS management console, search for CodeDeploy and select it
- Click on create applications
- Give a name for your application and deployment group



The screenshot shows the 'Create application' page in the AWS CodeDeploy console. The page has a header 'Create application' and a section 'Application configuration'. Under 'Application configuration', there are two fields: 'Application name' with a text input containing 'CodeDeploy' and a note '100 character limit', and 'Compute platform' with a dropdown menu showing 'EC2/On-premises'. At the bottom right, there are two buttons: 'Cancel' and 'Create application' (which is highlighted with a red border).

### Step 6: Create a deployment group in CodeDeploy

- Give a deployment group name
- Choose the role you have created for codedeploy

- In deployment type, choose In-place deployment

Enter a deployment group name

100 character limit

### Service role

Choose a service role

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

### 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 original environment are deregistered and can be terminated.

- In environment configuration, choose Amazon EC2 instance
- Choose the key as Name and value as the name of the instances you have given
- In service role, select the IAM role you have created for code deploy
- Click on create application

### Step 7: Create a pipeline

- In your AWS management console, search for pipeline and select it
- Click on create pipeline

- Give a name for your pipeline and click on next step

## Getting started with AWS CodePipeline

These steps will help you set up your first pipeline. Begin by giving your pipeline a name.

**Pipeline name\***


\* Required Cancel Next step


- Choose the source provider to be your AWS CodeCommit
- Select your repository name
- In branch, select the master branch

**Source provider\***

**AWS CodeCommit**

Choose a repository and a branch to use as the source location.

**Repository name\***  

**Branch name\***  

**i We will use Amazon CloudWatch Events to detect changes**

This requires AWS CodePipeline to create an Amazon CloudWatch Events rule and an IAM role on your behalf. You can opt-out in the options below.

► Change detection options

\* Required Cancel Previous Next step

- Click on next step
- In build provider, select no build and click on Next step
- In Deploy, choose the deployment provider to be AWS CodeDeploy
- Select the Application name and deployment group, which you gave while creating CodeDeploy

## Deploy ?

Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

**Deployment provider\*** AWS CodeDeploy

---

**AWS CodeDeploy** ⓘ

Choose one of your existing applications, or [create a new one in AWS CodeDeploy](#).

**Application name\*** CodeDeploy ↻

Choose one of your existing deployment groups, or [create a new one in AWS CodeDeploy](#).

**Deployment group\*** demo ↻

---

\* Required

[Cancel](#) [Previous](#) [Next step](#)

- In service role, click on create role
- It would create a role for you, then click on Allow
- The policy would be created but add an inline policy to it



```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": [  
        "codecommit:GetBranch",  
        "codecommit:GetCommit",  
        "codecommit:UploadArchive",  
        "codecommit:GetUploadArchiveStatus",  
        "codecommit:CancelUploadArchive"  
      ],  
      "Effect": "Allow",  
      "Resource": "*"   
    }  
  ]  
}
```

- Click on next

## AWS Service Role ?

Create a service role in IAM to give AWS CodePipeline permission to use resources in your account. If you already have a service role configured for this purpose, you can choose it from the list instead of creating a role. However, if that role is not configured correctly, AWS CodePipeline might not work as expected.

Role name\*

AWS-CodePipeline-Service

Create role

\* Required

Cancel

Previous

Next step

- Now review it and click on create pipeline

## Pipeline settings

Pipeline name

pipeline

Artifact location

s3://codepipeline-us-west-1-436608493143/  
AWS CodePipeline will use this existing S3 bucket to store artifacts for this pipeline. Depending on the size of your artifacts, you might be charged for storage costs. For more information, see [Amazon S3 storage pricing](#).

Role name

AWS-CodePipeline-Service

To save this configuration with these resources, choose Create pipeline.

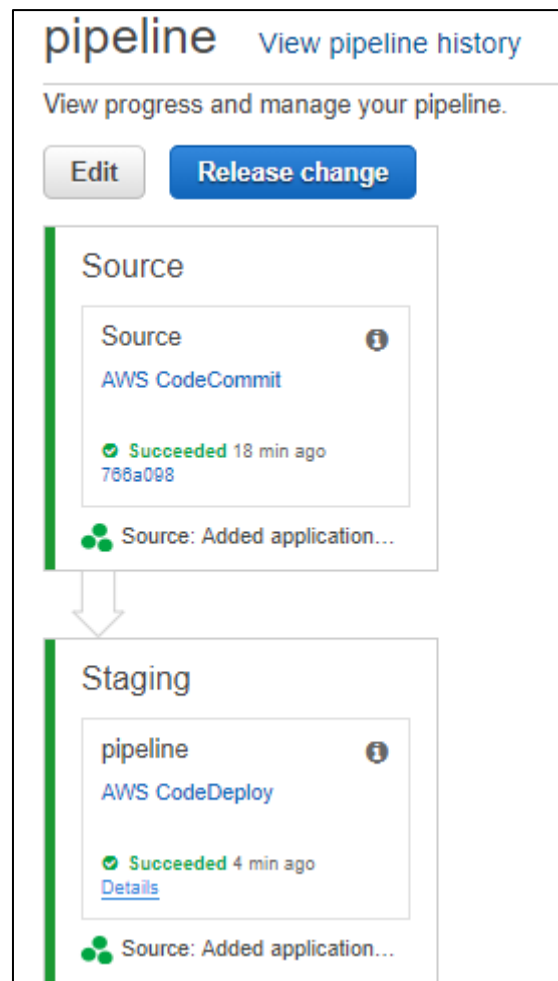
Would you like to create this pipeline?

Cancel

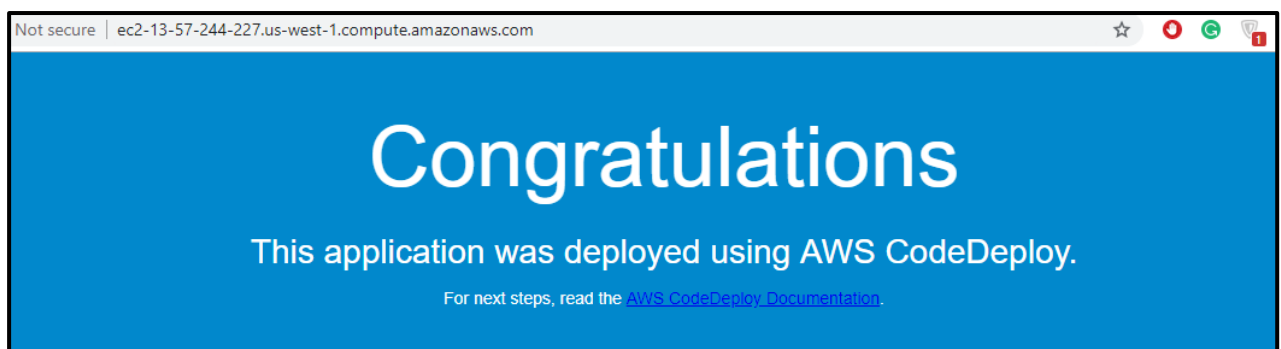
Previous

Create pipeline

- You will be able to see the pipeline you created



- when you copy your DNS of the EC2 instance in your browser you will be able to see your application

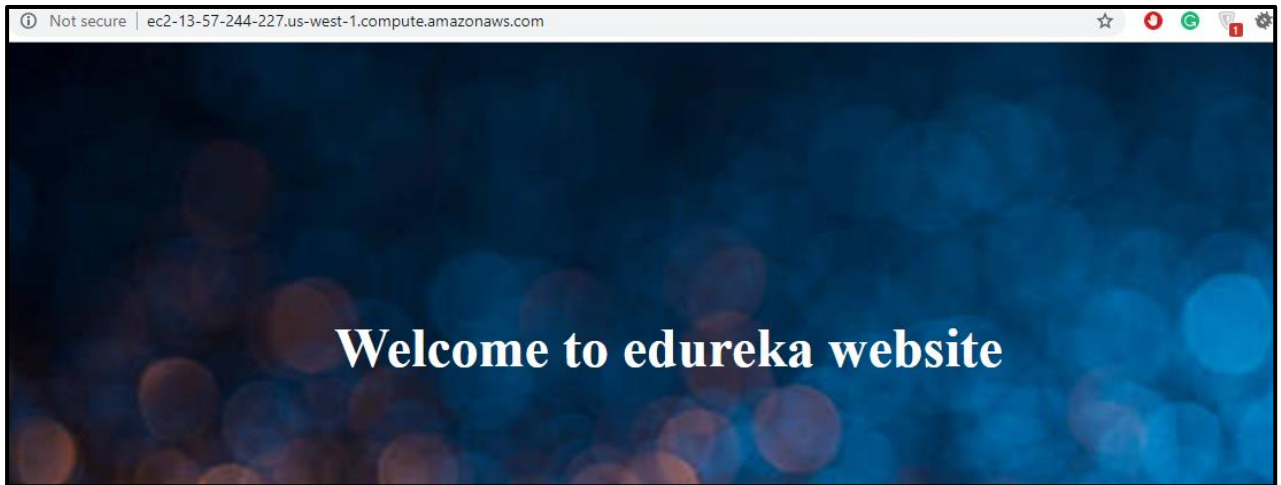


### Step 8: Update the code in CodeCommit

- In your local repo, modify the index.html by typing the below code in it and upload it

```
<!DOCTYPE html>
<html>
<head>
<style>
.a{ background-color: #2471A3;
color: white;
padding: 12px 20px;
border: none;
border-radius: 4px;
cursor: pointer;
float: center;  }
.bg {background-image: url("https://bit.ly/2OEVTYp");
/* Full height */ height: 100%;
background-position: center;
background-repeat: no-repeat;
background-size: cover; }
.label {color: white;
padding: 8px;
font-family: Arial;
} </style>
</head>
<body class="bg" style="padding: 210px 0; background-color: #dbfcf9;">
<center> <h3><font size="24"> <font color="white">Welcome to edureka
website</font></h3>
</center>
</body>
</html>
```

- Now when you reload the web browser you will be able to see



- Thus, you have successfully created a pipeline