Module 2: SDLC Automation

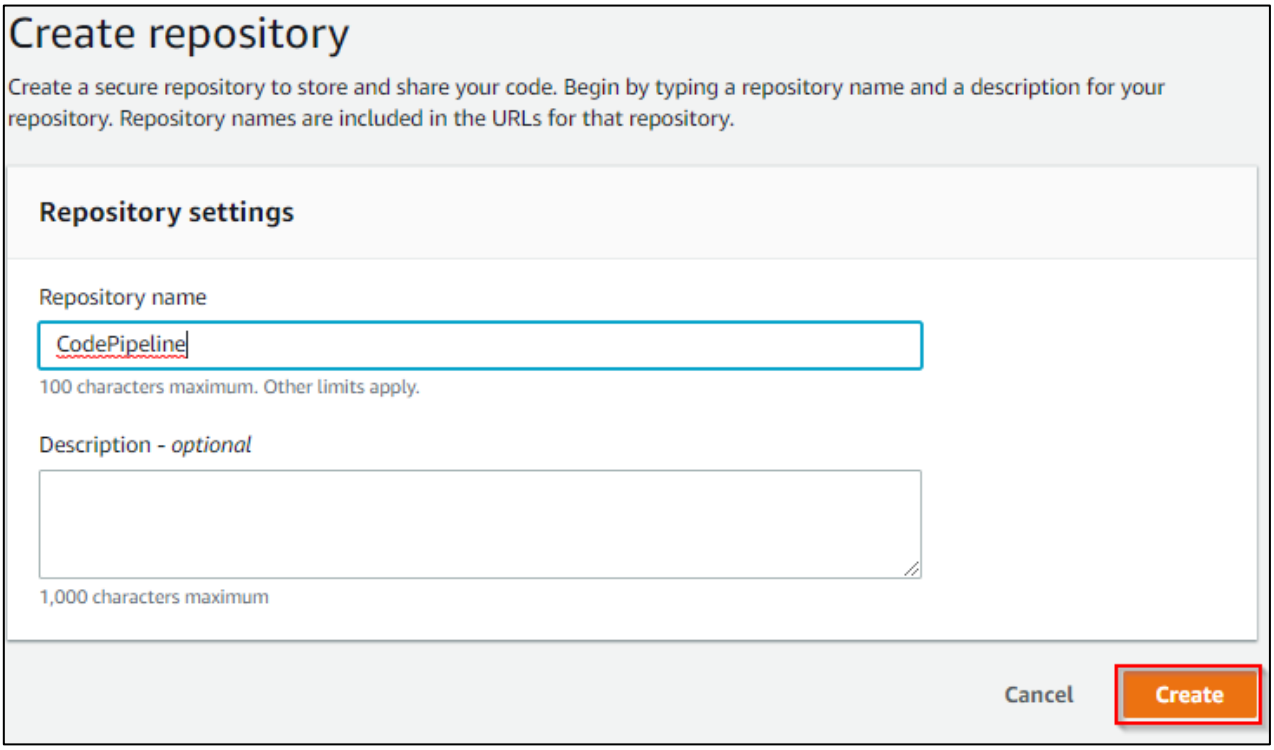
Demo Document 2

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



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
- 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

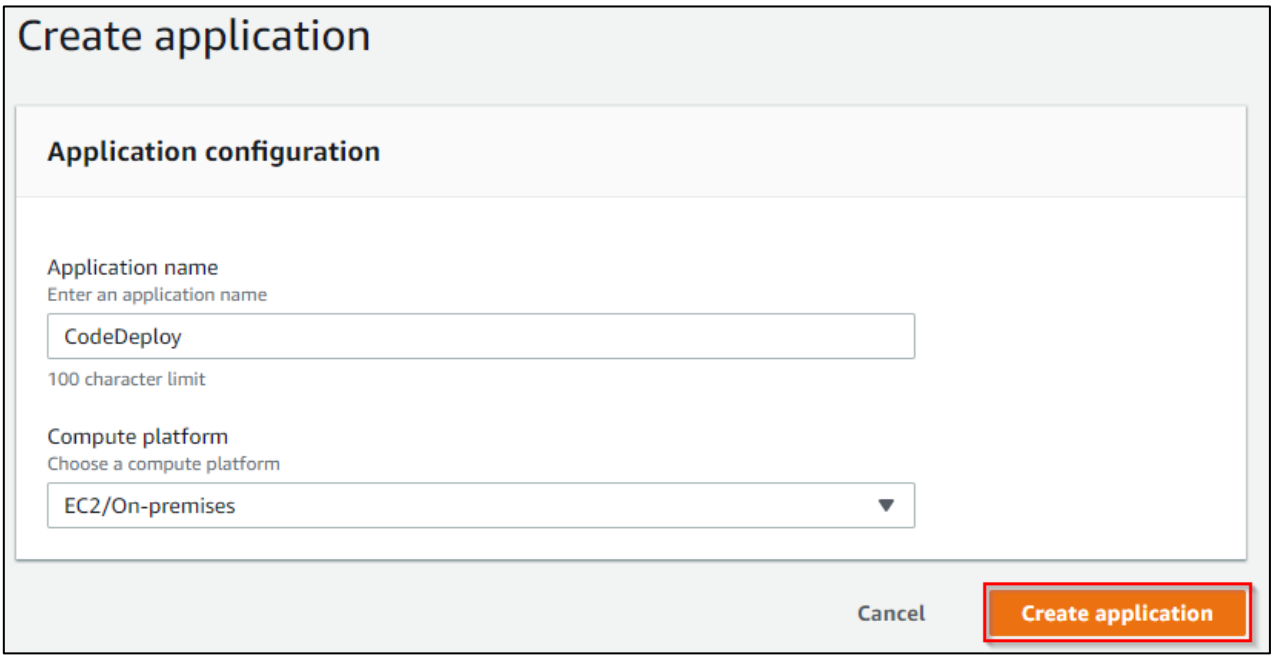
Step 4: Create an IAM role for CodeDeploy

- 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

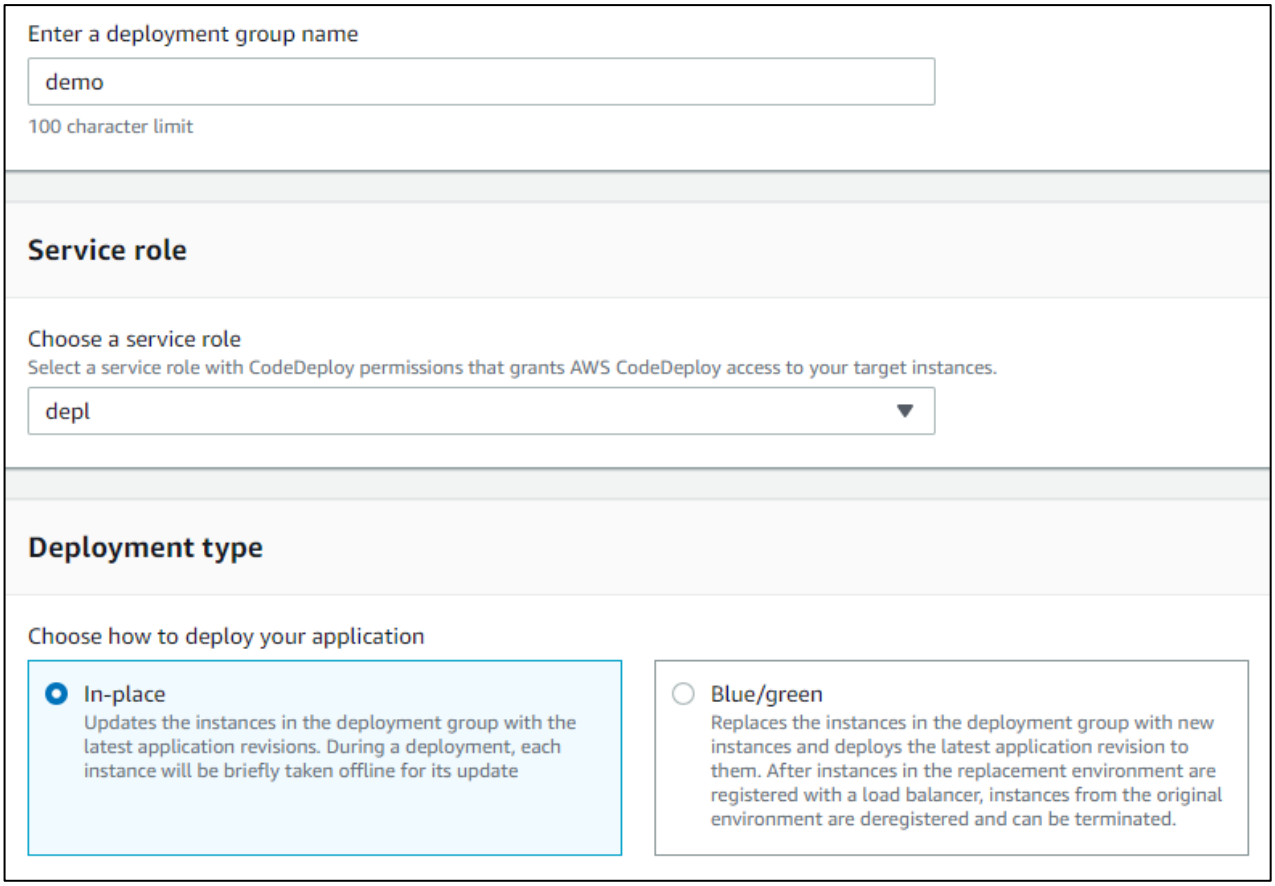
- In your AWS management console, search for CodeDeploy and select it
- Click on create applications
- Give a name for your application and deployment group



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

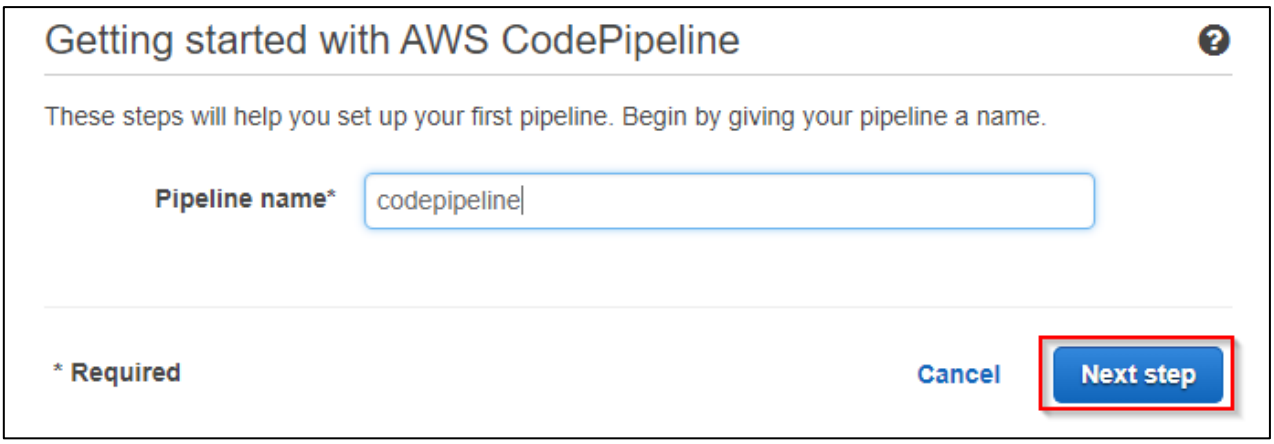


- In environment configuration, choose Amazon EC2 instance
- Choose the key as Name and value as the name of the instances you have given
- In service role, select the IAM role you have created for code deploy
- Click on create application

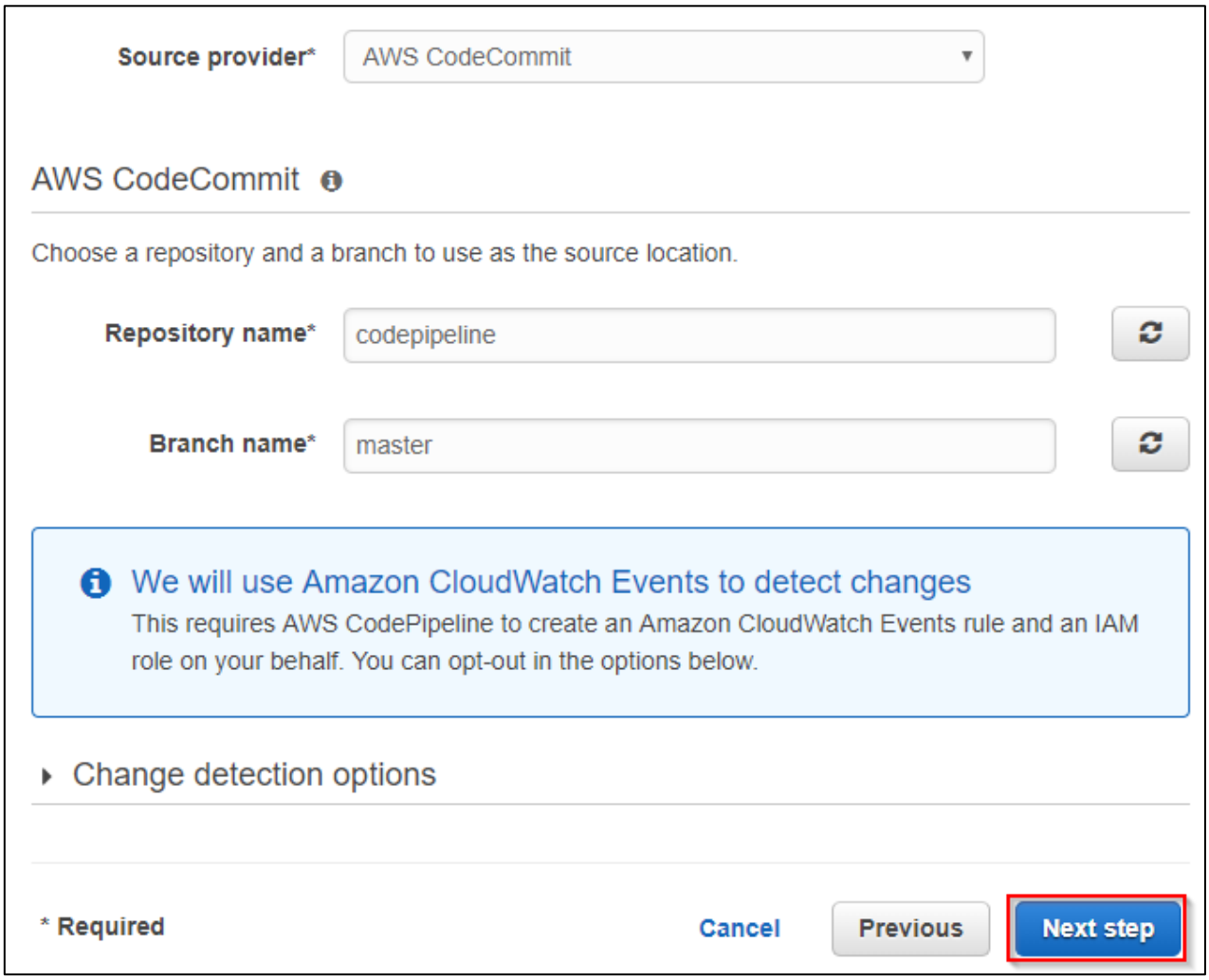
Step 7: Create a pipeline

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

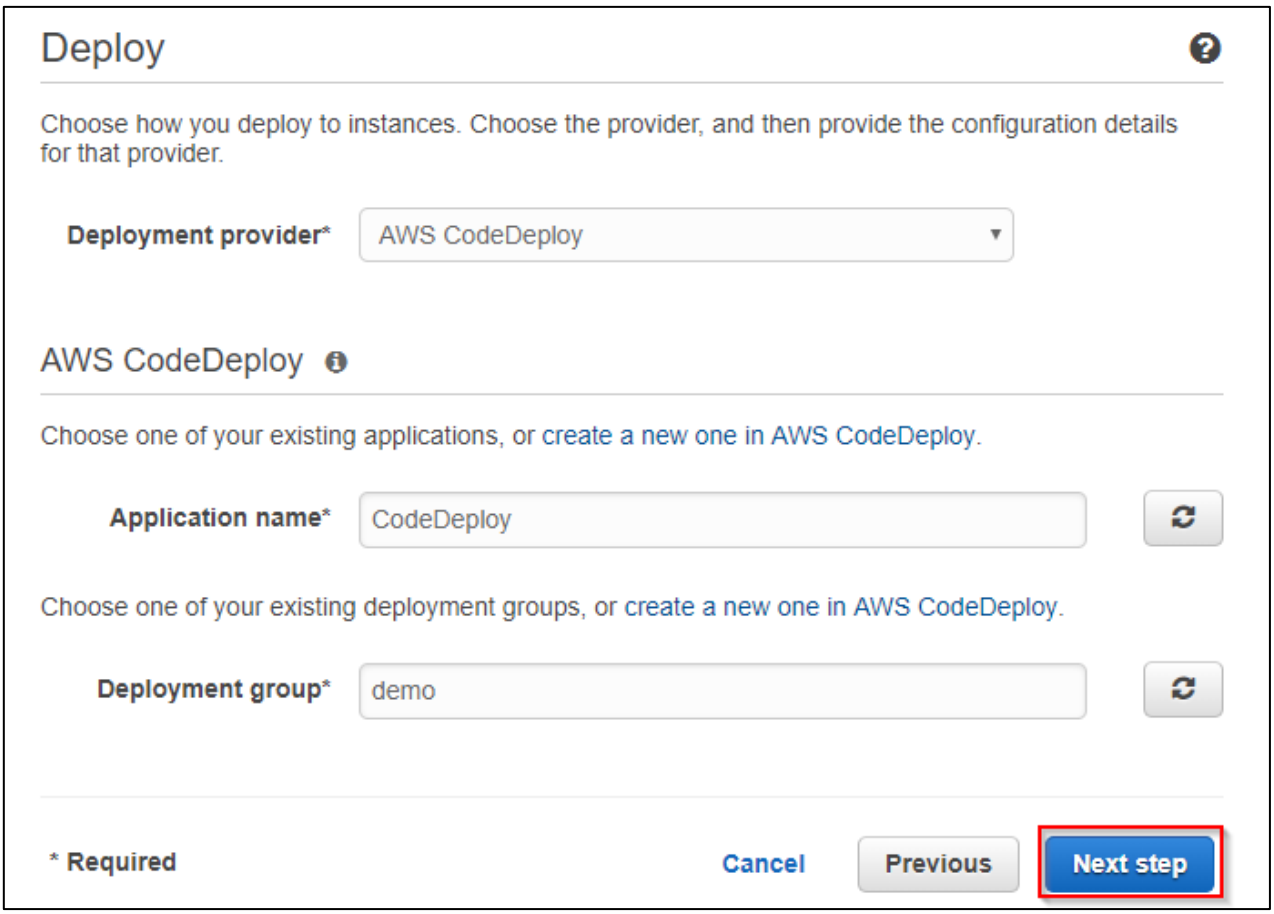
- Give a name for your pipeline and click on next step



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



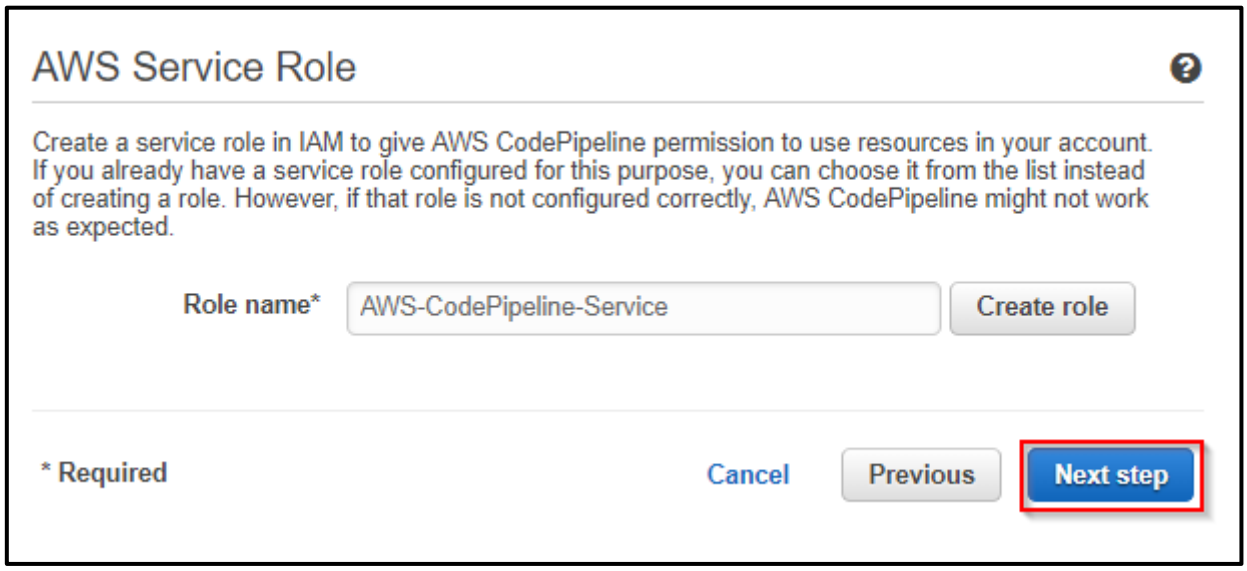
- Click on next step
- In build provider, select no build and click on Next step
- In Deploy, choose the deployment provider to be AWS CodeDeploy
- Select the Application name and deployment group, which you gave while creating CodeDeploy



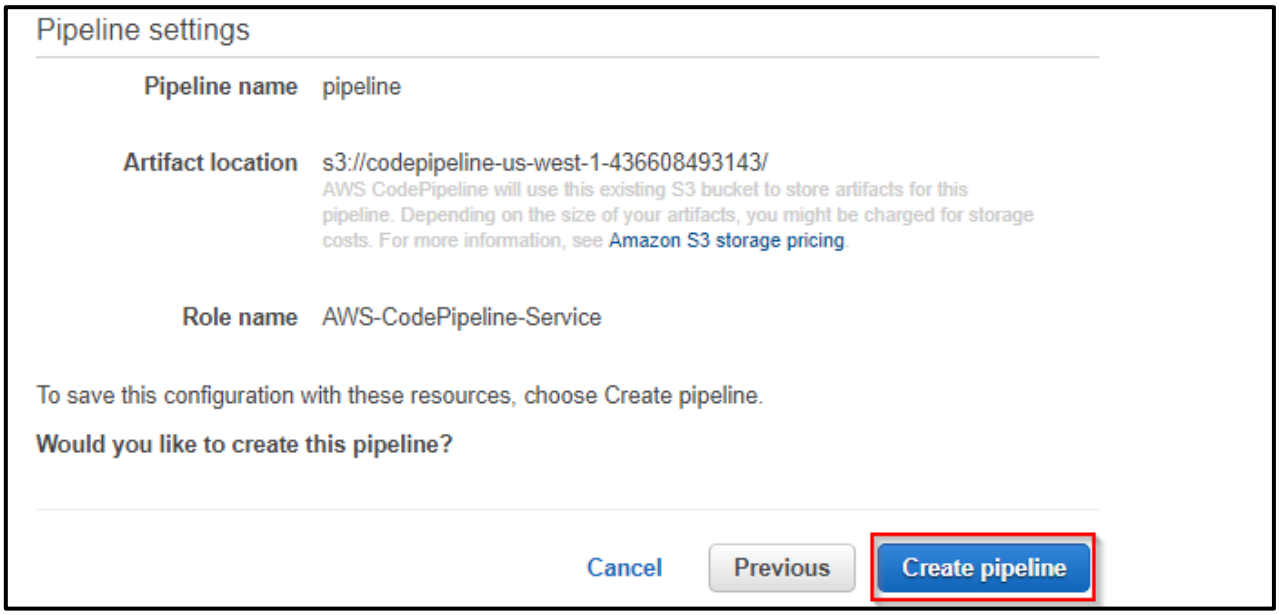
- In service role, click on create role
- It would create a role for you, then click on Allow
- The policy would be created but add an inline policy to it

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

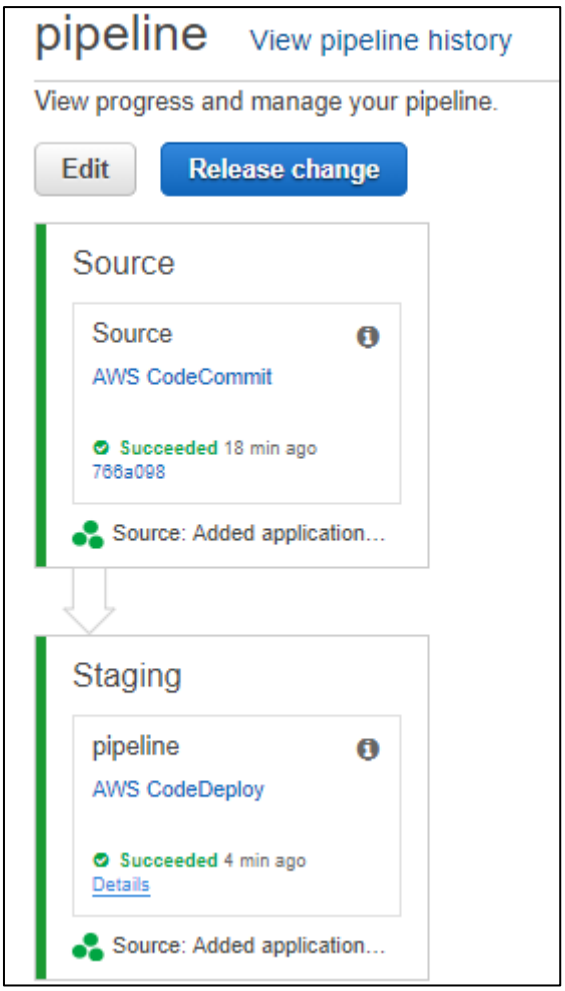
- Click on next



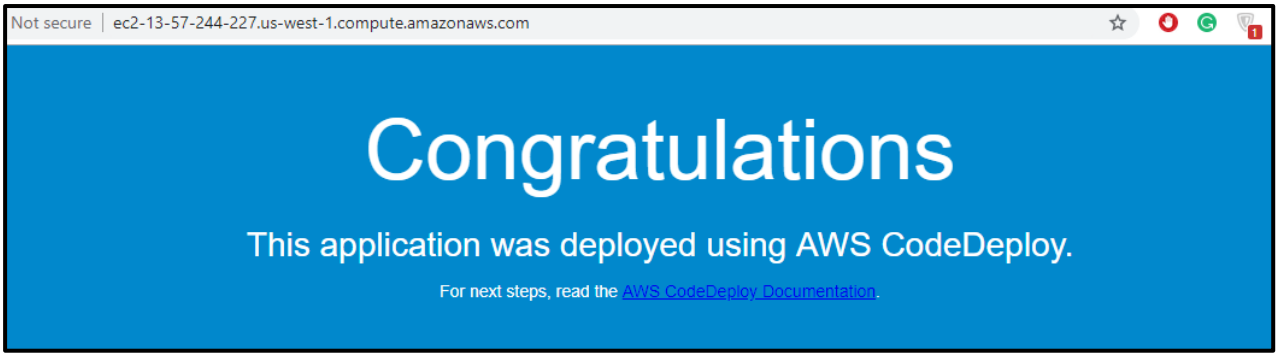
- Now review it and click on create pipeline



- 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

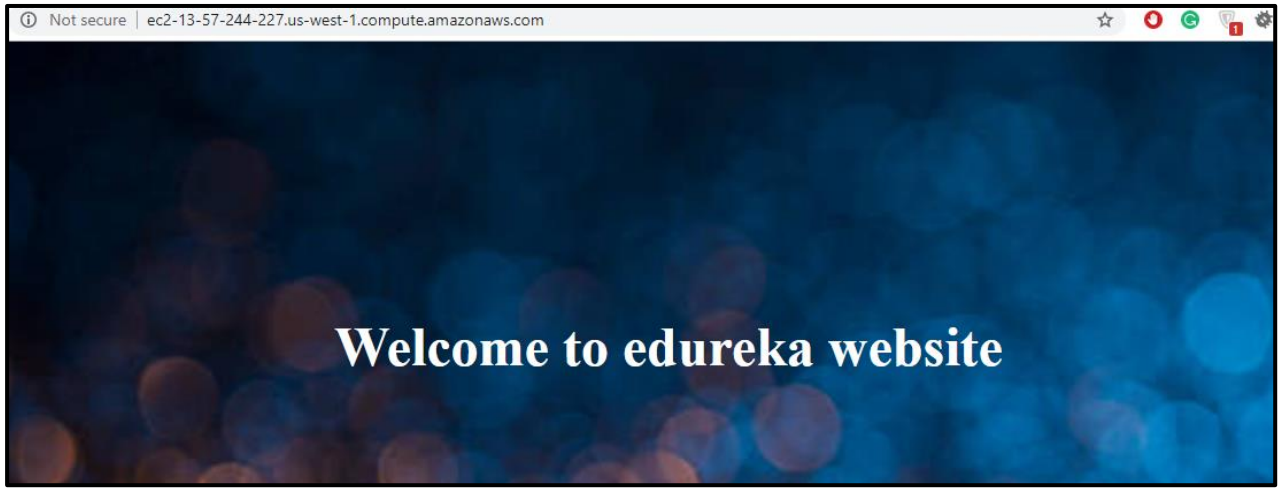


Step 8: Update the code in CodeCommit

- 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