**Lesson 11 Demo 3**

**CodeDeploy**

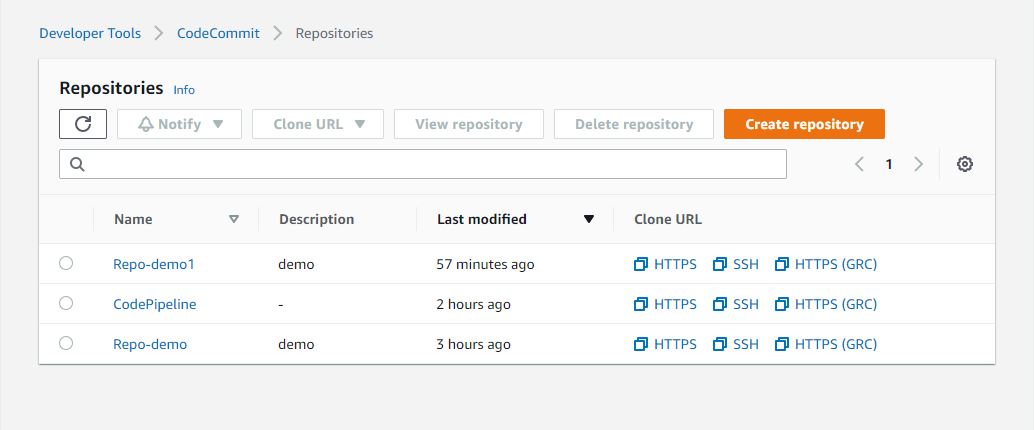


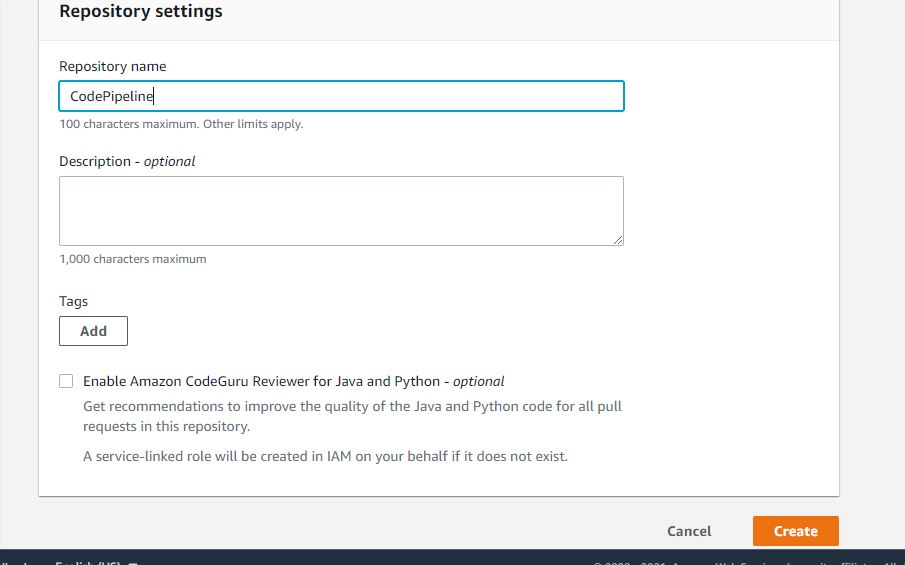
Steps to be followed:

1. Create an AWS CodeCommit Repository
2. Add a file to your repository
3. Create an Amazon EC2 Instance and Install the AWS CodeDeploy Agent
4. To know the name of the bucket, which is in your region
5. Create an IAM role for CodeDeploy
6. Create a Code deploy application
7. Create a deployment group in CodeDeploy

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

1. In your AWS Management Console, search for and CodeCommit. Click on **Create repository**
2. Write a name and description for your repository. Click on **Create repository**

****



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

1. Download the file from https://s3.amazonaws.com/aws-codedeploy-us-east- 1/samples/latest/SampleApp\_Linux.zip
2. Extract the file and save it in the directory where you have cloned the repository in your local system
3. 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**

1. In your EC2 services, click on **Launch Instance**
2. Choose an **Amazon Linux AMI**
3. Under instance type, choose the **t2.micro**
4. Go to configure instance details, under IAM role, and click on **Create a new IAM role**
5. Create a new IAM role with the policy

***{***

***"Version": "2012-10-17",***

***"Statement": [***

***{***

***"Action": "codedeploy:\*",***

***"Effect": "Allow",***

***"Resource": "\*"***

***},***

***{***

***"Sid": "CodeStarNotificationsReadWriteAccess",***

***"Effect": "Allow",***

***"Action": [***

***"codestar-notifications:CreateNotificationRule",***

***"codestar-notifications:DescribeNotificationRule",***

***"codestar-notifications:UpdateNotificationRule",***

***"codestar-notifications:DeleteNotificationRule",***

***"codestar-notifications:Subscribe",***

***"codestar-notifications:Unsubscribe"***

***],***

***"Resource": "\*",***

***"Condition": {***

***"StringLike": {***

***"codestar-notifications:NotificationsForResource": "arn:aws:codedeploy:\*"***

***}***

***}***

***},***

***{***

***"Sid": "CodeStarNotificationsListAccess",***

***"Effect": "Allow",***

***"Action": [***

***"codestar-notifications:ListNotificationRules",***

***"codestar-notifications:ListTargets",***

***"codestar-notifications:ListTagsforResource",***

***"codestar-notifications:ListEventTypes"***

***],***

***"Resource": "\*"***

***},***

***{***

***"Sid": "CodeStarNotificationsSNSTopicCreateAccess",***

***"Effect": "Allow",***

***"Action": [***

***"sns:CreateTopic",***

***"sns:SetTopicAttributes"***

***],***

***"Resource": "arn:aws:sns:\*:\*:codestar-notifications\*"***

***},***

***{***

***"Sid": "CodeStarNotificationsChatbotAccess",***

***"Effect": "Allow",***

***"Action": [***

***"chatbot:DescribeSlackChannelConfigurations"***

***],***

***"Resource": "\*"***

***},***

***{***

***"Sid": "SNSTopicListAccess",***

***"Effect": "Allow",***

***"Action": [***

***"sns:ListTopics"***

***],***

***"Resource": "\*"***

***}***

***]***

***}***

1. 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

**Step 4: To know the name of the bucket, which is in your region**

1. Add storage and attach security groups to it
2. Click on **Review and launch**

https://docs.aws.amazon.com/codedeploy/latest/userguide/resource-kit.html#resource- kit-bucket-names

**Step 5: Create an IAM role for CodeDeploy**

1. On the IAM dashboard, click on **Roles**
2. Click on **Create roles**
3. Choose the services as **CodeDeploy** and click on **Next permission**
4. In **Attached permissions policy**, choose **AWSCodeDeployRole policy** and click on **Review** Give a name for your role and click on **Create role**
5. Once the role is created, go to the trusted relationship tab, and click on **Edit trust relationship**

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "",

"Effect": "Allow",

"Principal": {

"Service": [

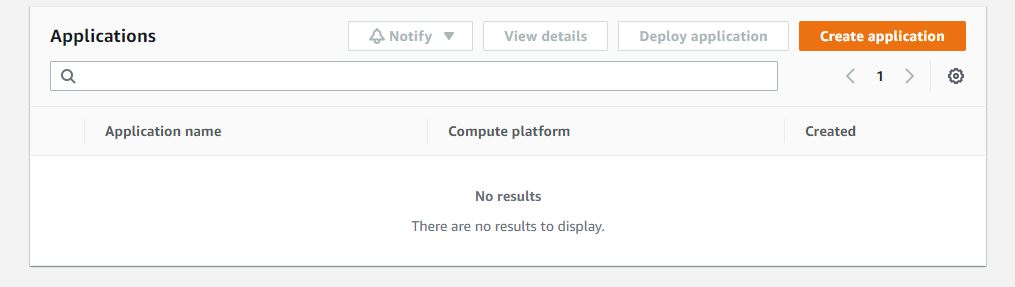
"[codedeploy.amazonaws.com](http://codedeploy.amazonaws.com/)"

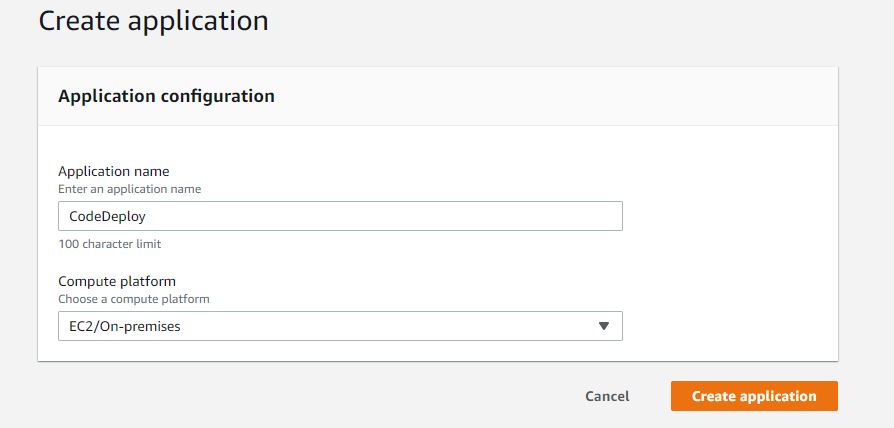
] },

"Action": "sts:AssumeRole" } ]}

**Step 6: Create a Code deploy application**

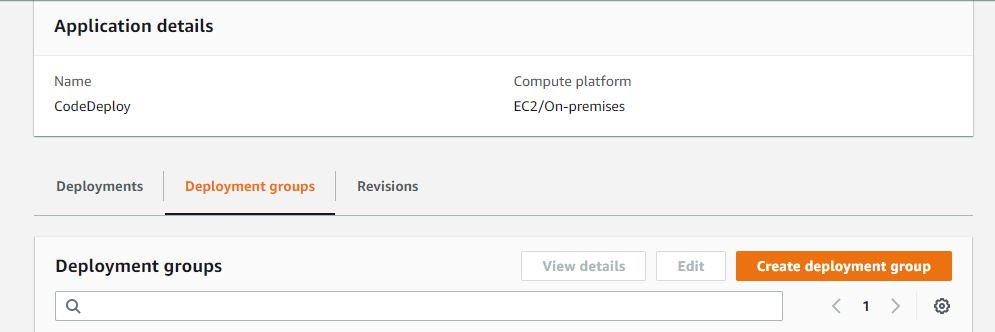
1. In your AWS management console, search for and select CodeDeploy
2. Click on **Create applications**
3. Give a name for your application and deployment group



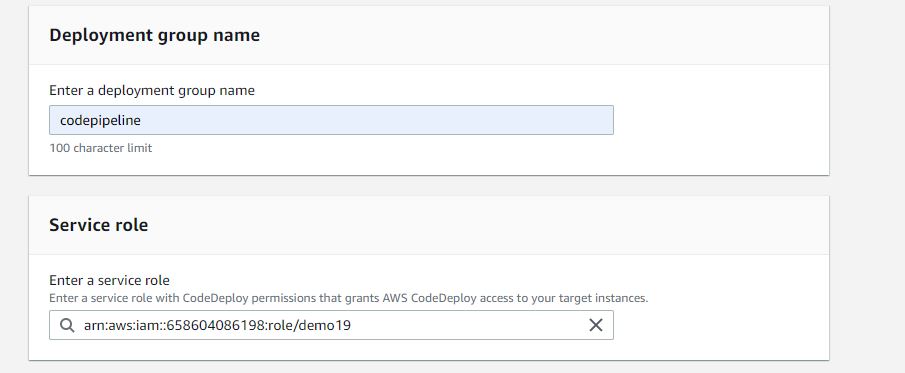


**Step 7: Create a deployment group in CodeDeploy**

1. Give a deployment group name



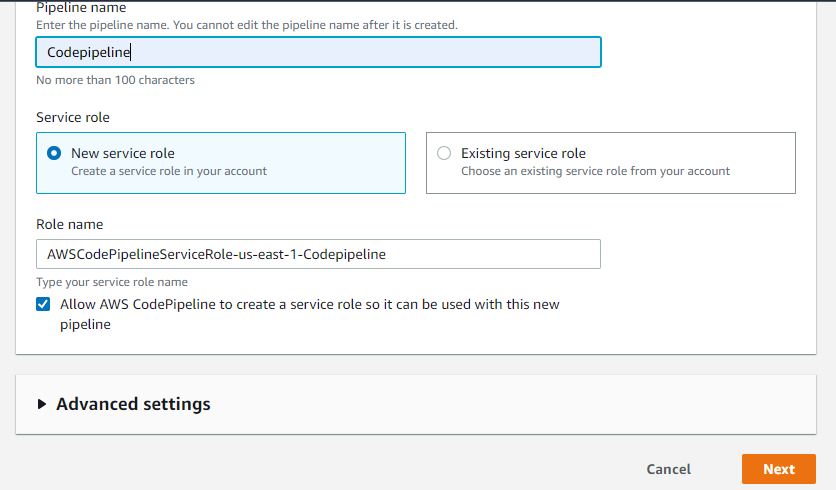
1. Choose the role you have created for CodeDeploy



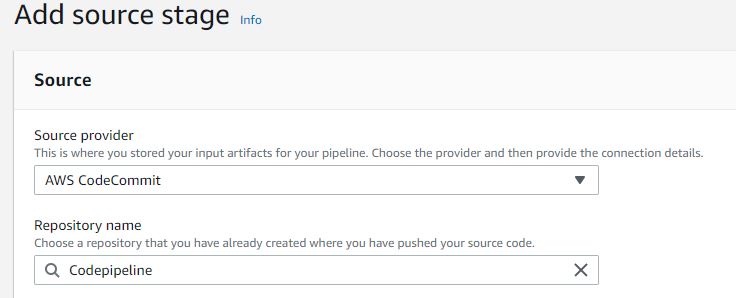
1. In environment configuration, choose **Amazon EC2 instance**
2. Choose the key as **Name** and value as the name of the instances you have given Under service role, select the IAM role you have created for CodeDeploy
3. Click on **Create application**

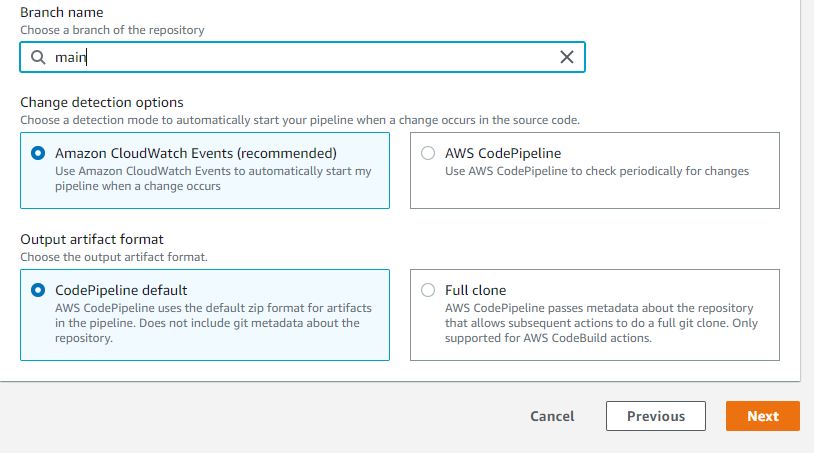
**Create a pipeline**

* In your AWS management console, search for and select pipeline Click on **Create pipeline**
* In deployment type, choose In-place deployment
* Give a name for your pipeline and click on **Next step**



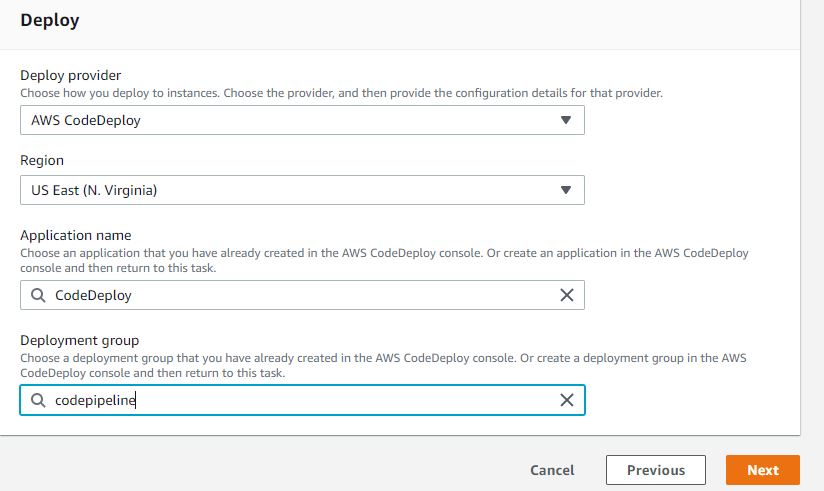
* Choose the source provider as **AWS CodeCommit**
* Select your repository name
* Under the Branch name, select the main branch





* Click on Next step
* Under the build provider, select No build and click on Next step
* In Deploy, choose the deployment provider as AWS CodeDeploy
* Select the Application name and Deployment group that you drafted while creating

**CodeDeploy**



* In service role, click on **Create role**
* When it creates a role for you, click on **Allow**
* Add an inline policy to it upon creation of the policy

{  
"Version": "2012-10-17", "Statement": [

{  
"Action": [  
"codecommit:GetBranch", "codecommit:GetCommit", "codecommit:UploadArchive", "codecommit:GetUploadArchiveStatus", "codecommit:CancelUploadArchive"  
],  
"Effect": "Allow",  
"Resource": "\*"

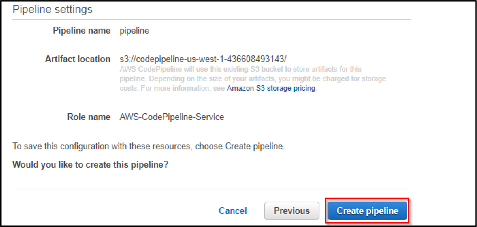
} ]

}

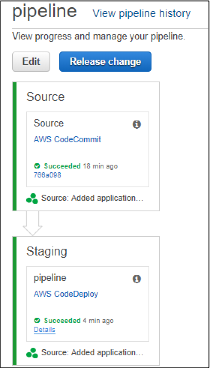
• Click on **Next**



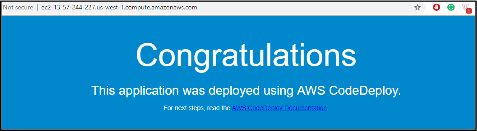
• Now, please review it and click on **Create pipeline**



• 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 view your application



**Step 8: Update the code in CodeCommit**

1. In your local repository, modify the **index.html** by typing the below code 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 Sample website</font></h3>

</center> </body> </html>

• Now, when you reload the web browser, you can view the following result: