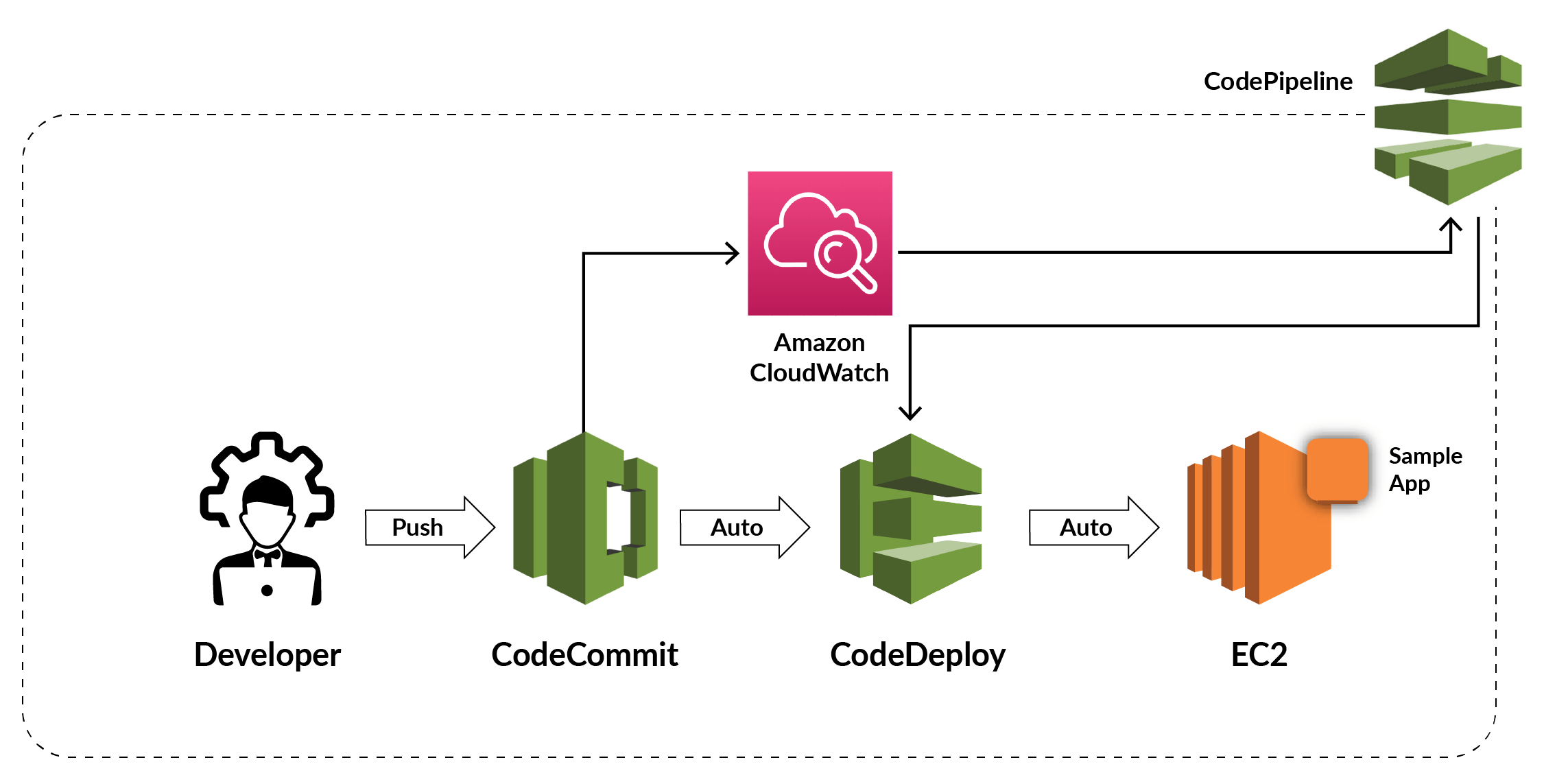
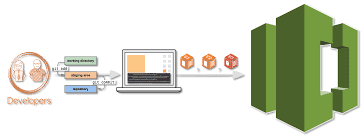


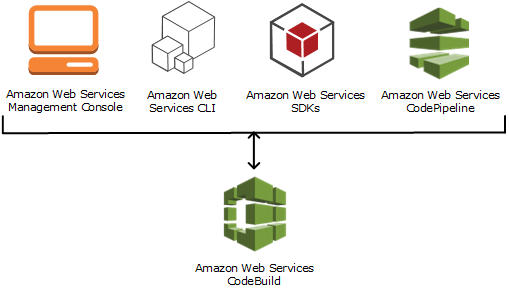
**Code Deployment on AWS using Code Pipeline: Code Commit, Code Deploy, CloudWatch**

****

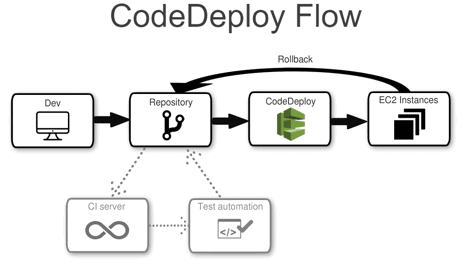
**AWS Code Commit** – It is a fully managed source control service that hosts secure Git-based repositories. Code Commit makes it easy for teams to collaborate on code in a secure and highly scalable ecosystem.

****

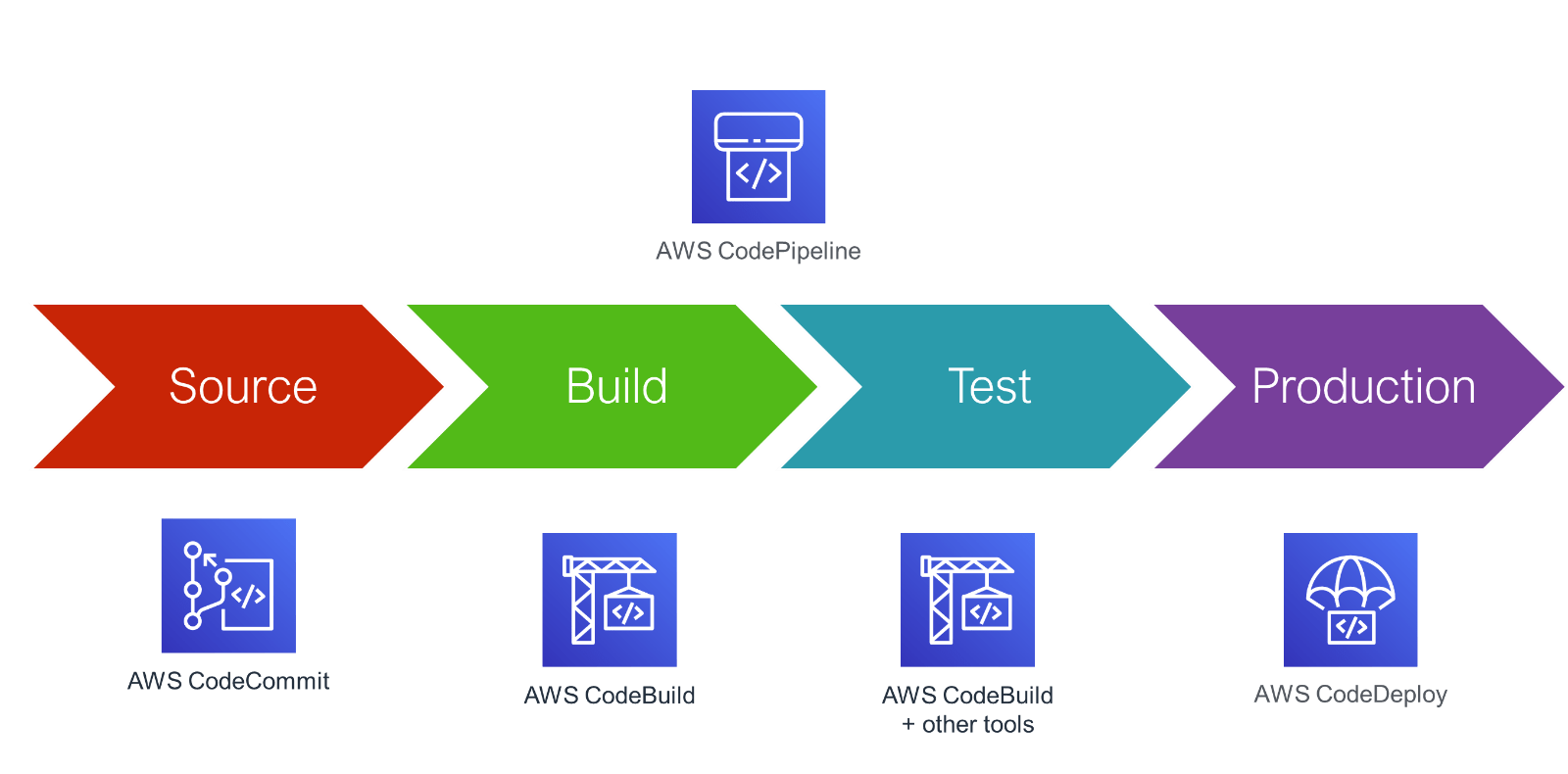
**AWS Code Build** – A fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy, on a dynamically created build server.



**AWS Code Deploy** – A fully managed deployment service that automates software deployments to a variety of computing services such as Amazon EC2, AWS Fargate, AWS Lambda, and you’re on-premises servers.

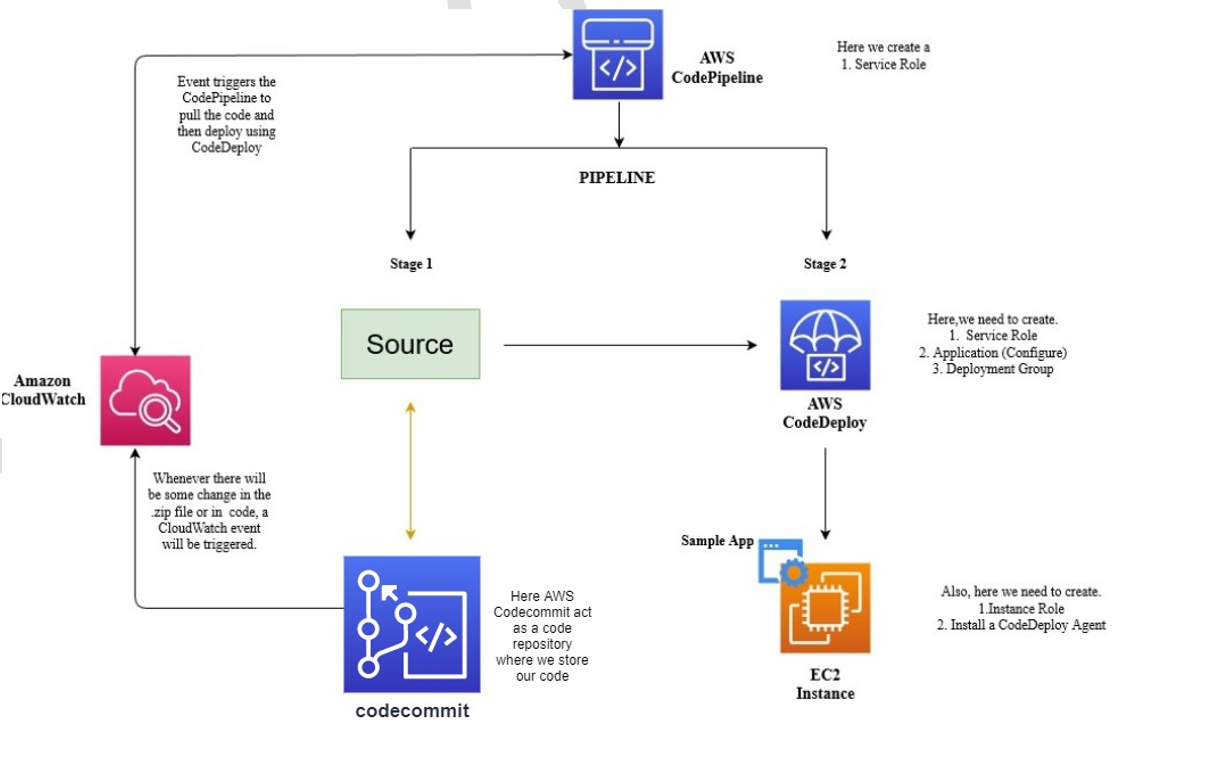


**AWS Code Pipeline** – A fully configured continuous delivery service that helps the user to automate their released pipelines for fast and reliable application and infrastructure updates.

****

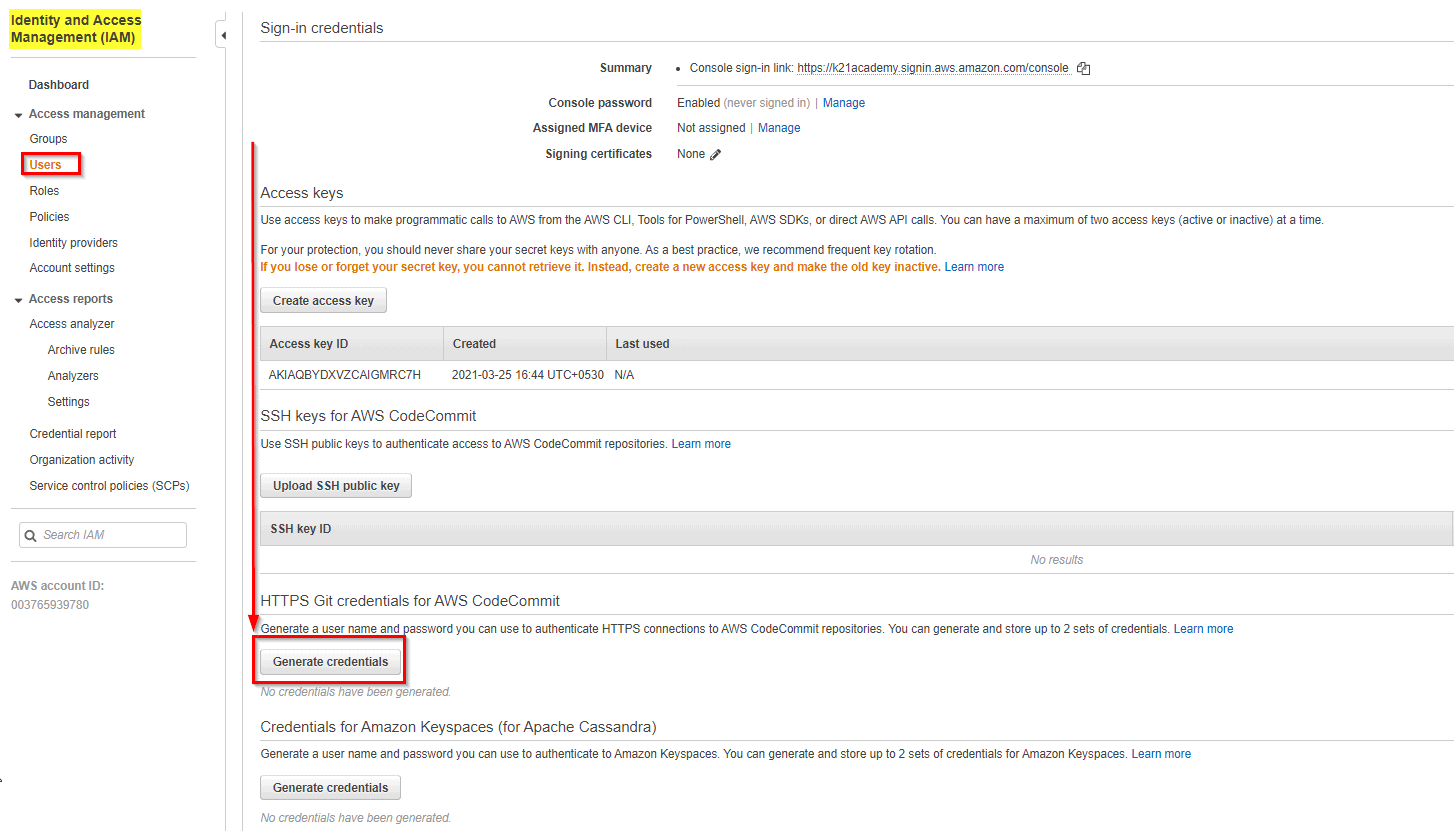
**The pipeline has two stages:**

* A source stage **(Source)** for your Code Commit source action.
* A deployment stage **(Deploy)** for your Code Deploy deployment action.

****

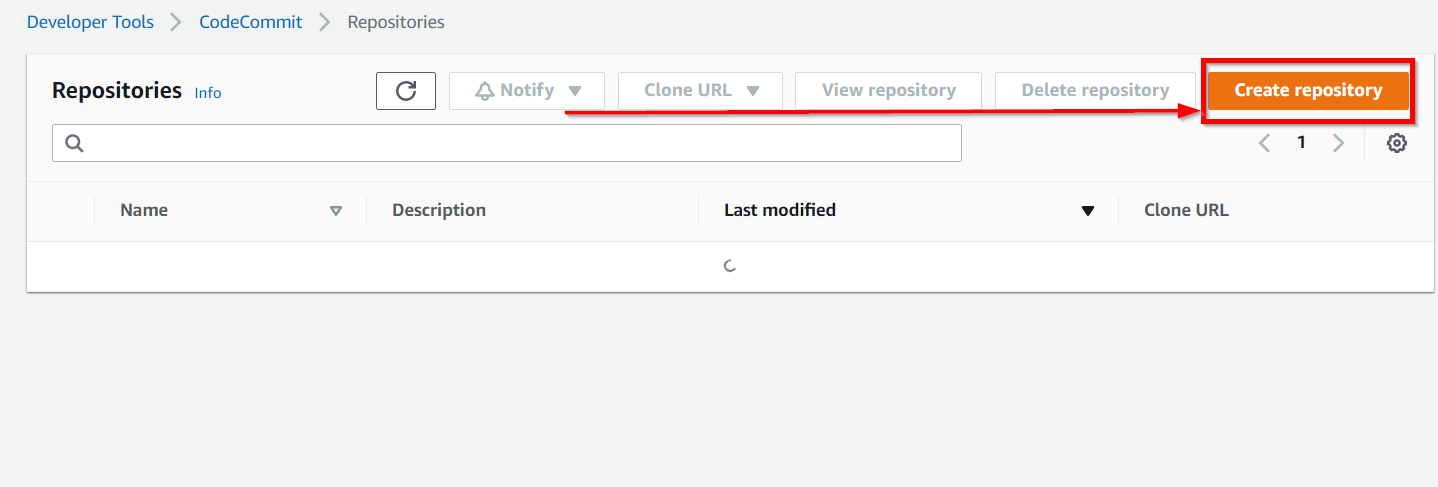
### Step 1: Download the HTTPS Git credentials for AWS Code Commit using the IAM console

**1)**Click on **users**from the left menu and click on your user then click on **security credentials**Scroll down to **HTTPS Git credentials** for AWS Code Commit and click on **Generate Credential**

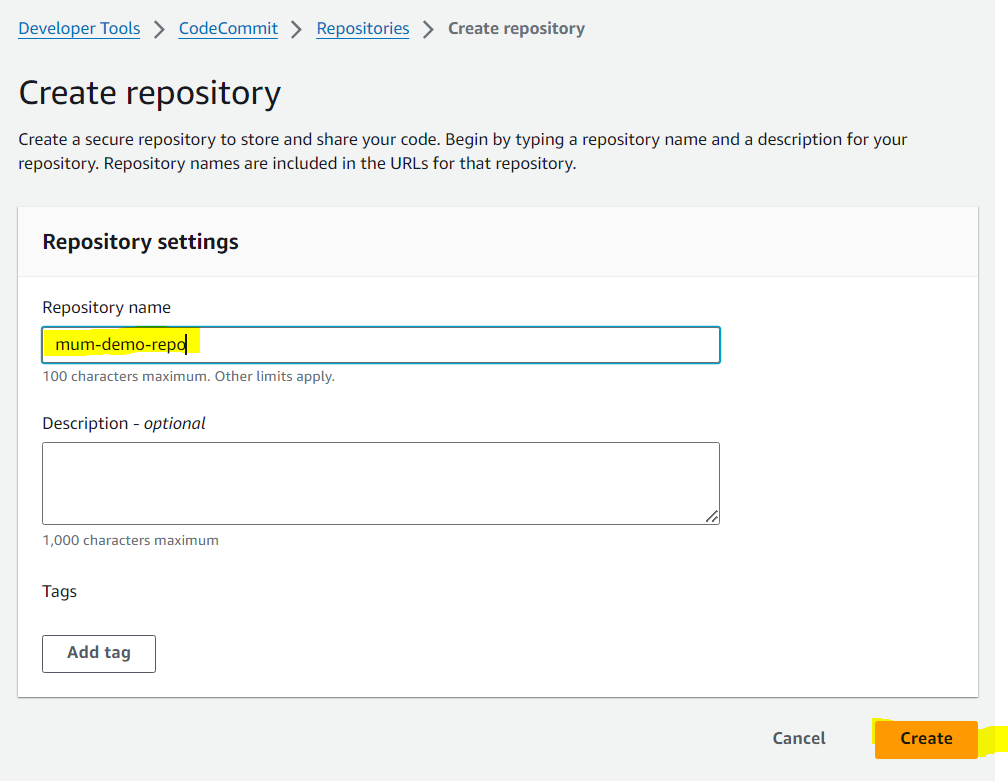


### **Step 2: Create a Code Commit Repository**

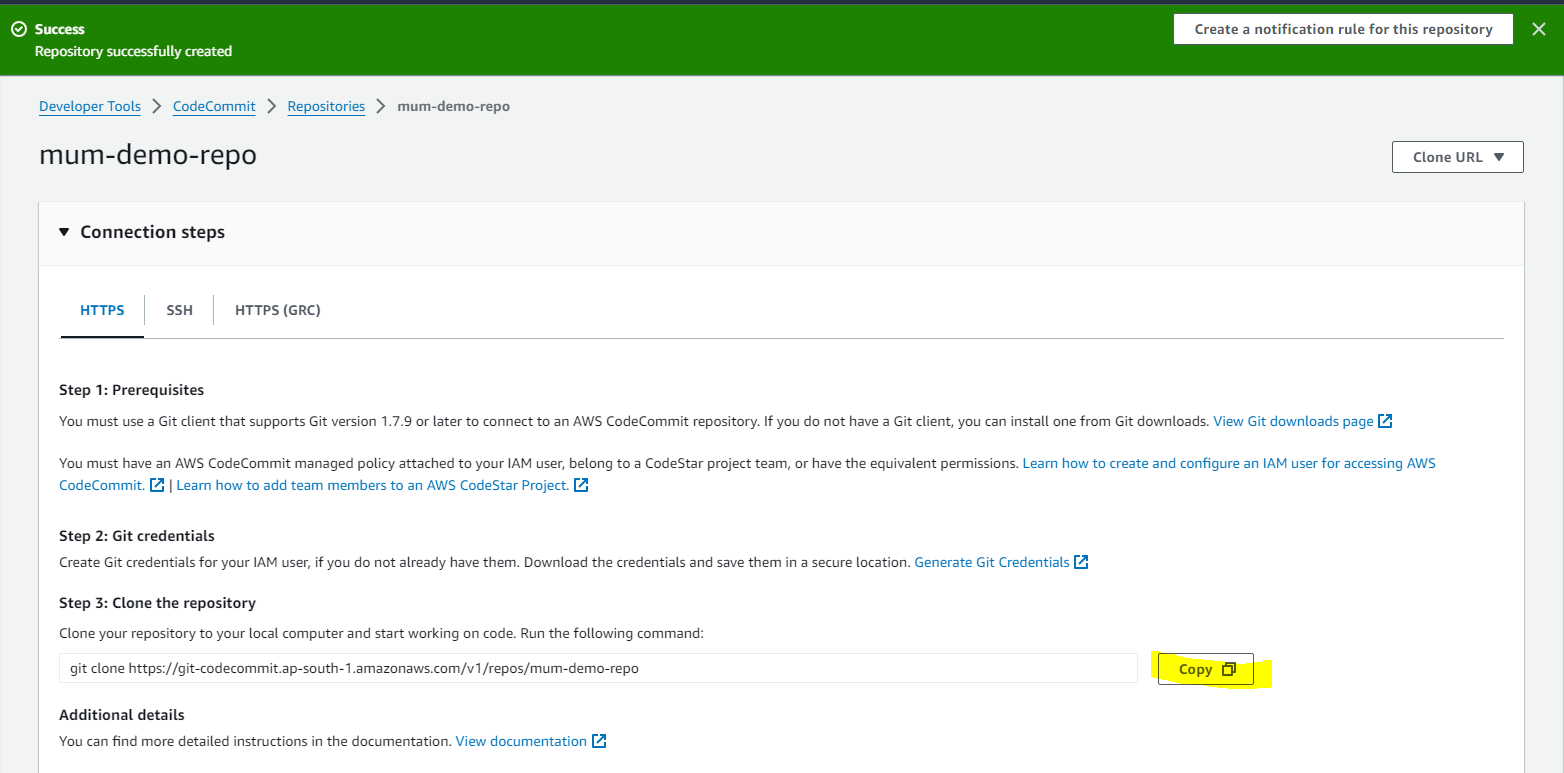
**1)** Open the Amazon Code commit console and choose **Create repository.**



**2)** On the **Create repository page**, in the Repository name, enter a name for your repository (for example, **Demo-repo**). Then Choose **Create.**

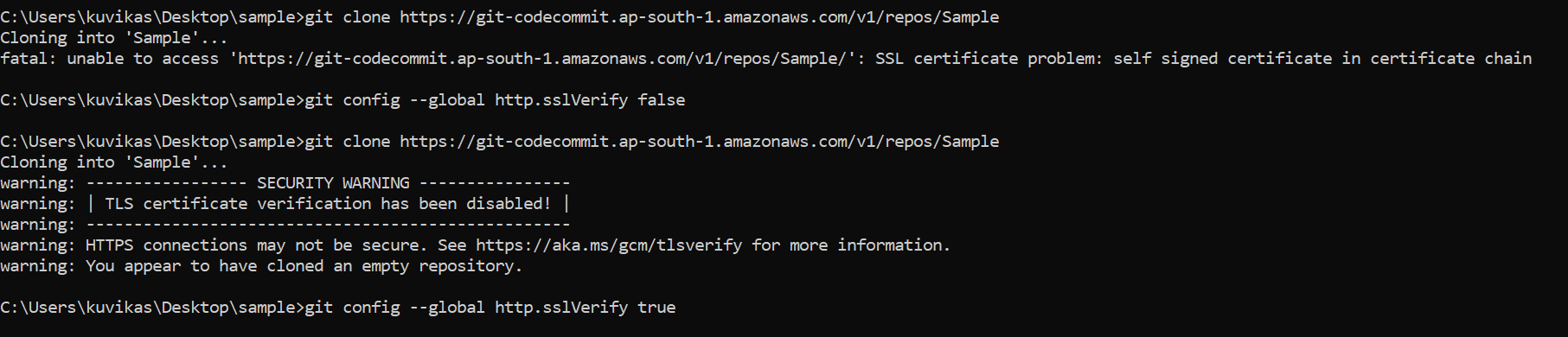


**3)**On the next screen, click on Copy next to Step 3: Clone the Repository (We’ll use this to clone this blank repository in the next step).



### **Step:3 Add Sample Code to your Code Commit Repository**

1. Run the following command to clone the repository, **replacing the GIT Clone Address** with the one you copied in the **earlier step**.  
   **git clone <GIT Clone Address>**then enter the **username** and **password** for the **IAM user's HTTPS Git Credentials** that we already downloaded



After that, it will show you a warning message that you cloned an empty repository but it’s ok

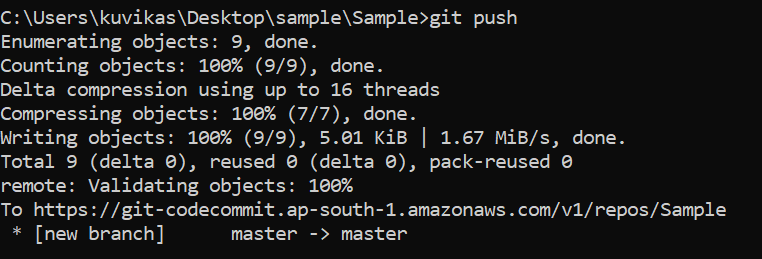
**2)**Next, download the sample code and save it in which we created our local repository.

Unzip the files from SampleApp\_Linux.zip into the local directory you created earlier (for example, /tmp/MyDemoRepo).  
Deploy to Amazon Linux instances using Code Deploy

**3)**Open Command Prompt and make sure you are in the **SampleApp\_Linux** folder

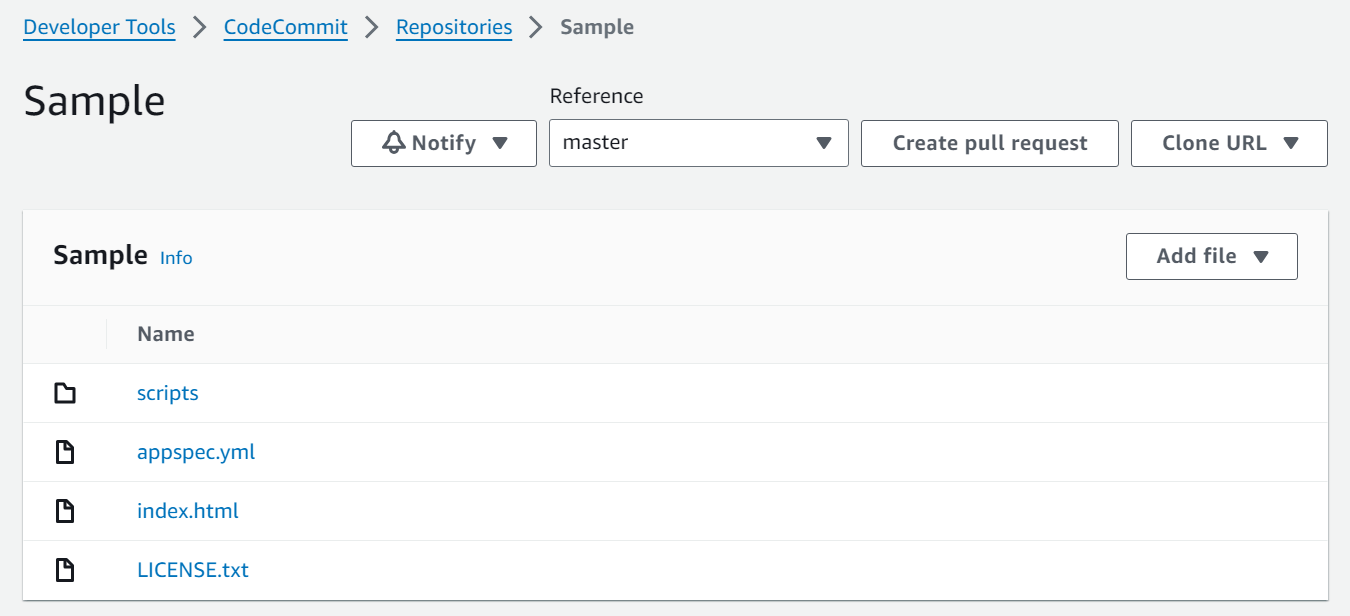
Run the following commands to stage all your files at once and commit the files with a commit message:

**git add .  
git commit -m “Add sample application files”  
git push**



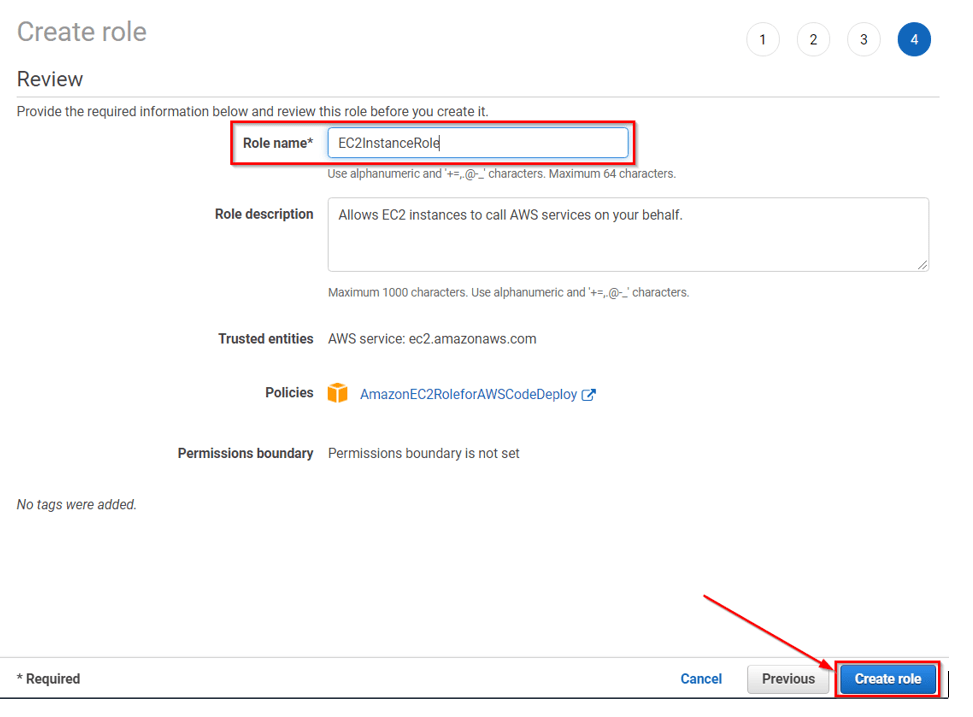
**4)**The files you downloaded and added to your local repo have now been added to the master branch in your Code Commit

Repo repository and are ready to be included in a pipeline.



### **Step 4: Create Amazon EC2 Windows instances and install the Code Deploy agent**

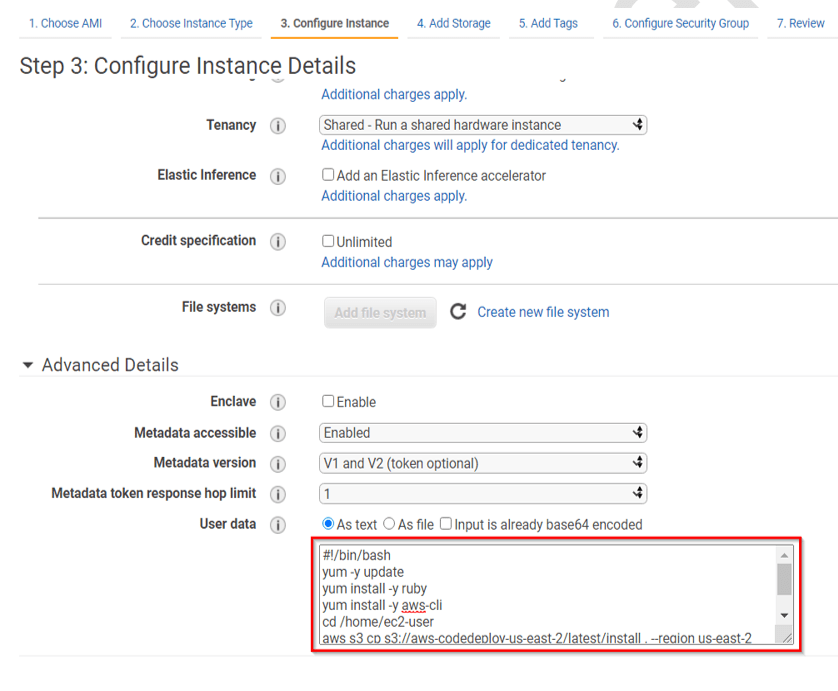
**1) Create an IAM role** that will be required to grant permission to the EC2 instance. Select the policy named **AmazonEC2RoleforAWSCodeDeploy**to create.

****

1. Launch the instance on which our code will be deployed.Just remember to add the **IAM role** that we have created and In Auto-assign Public IP, choose **Enable.**

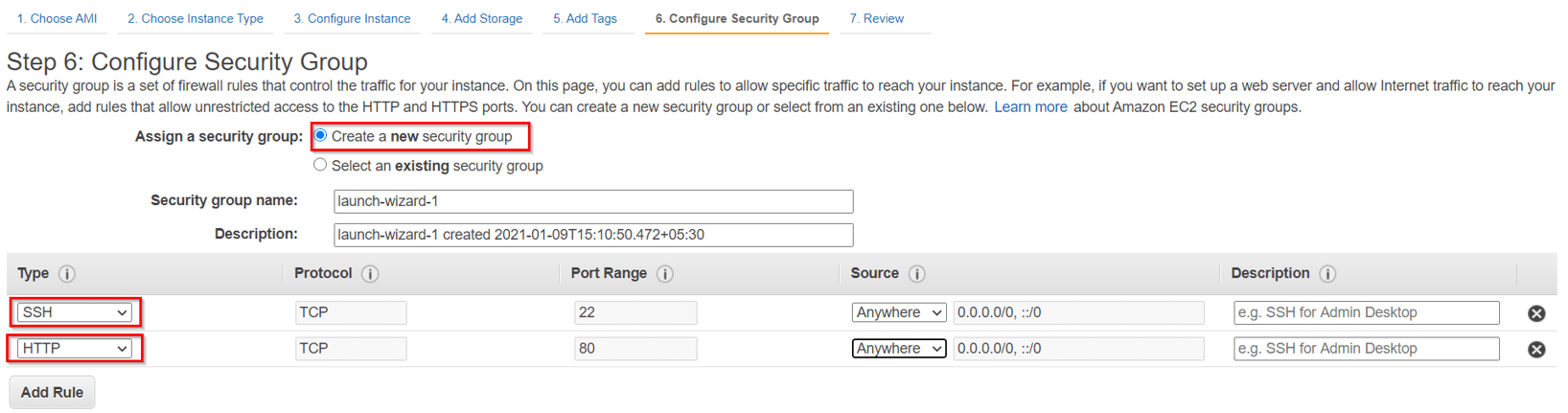
Expand Advanced Details, and in User data, As **text selected**, enter the following:

**#!/bin/bash  
yum -y update  
yum install -y ruby  
yum install -y aws-cli  
cd /home/ec2-user  
aws s3 cp s3://aws-codedeploy-us-east-1/latest/install . --region us-east-1  
chmod +x ./install  
./install auto**

****

**4)**On the Configure Security Group page, do the following:

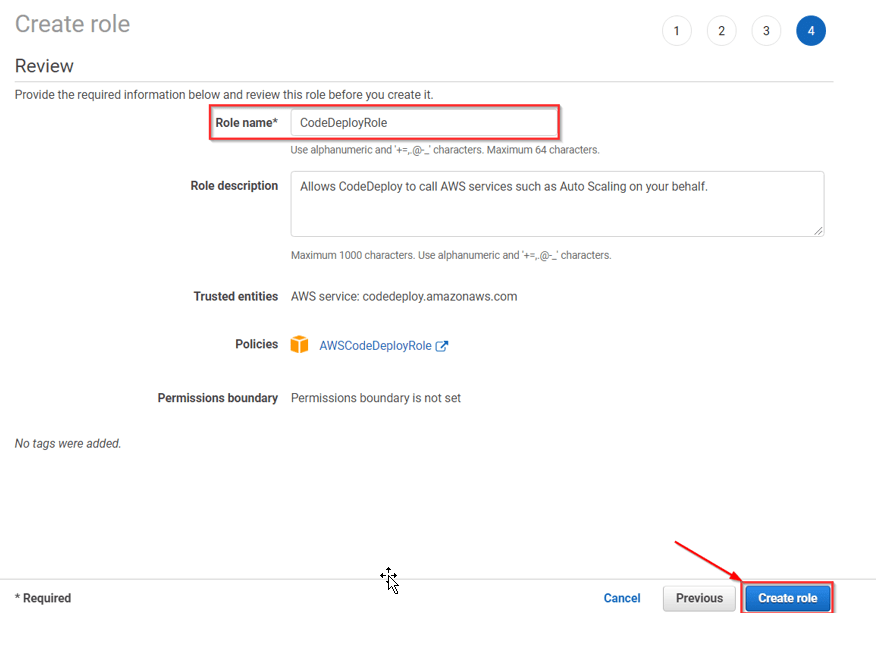
• Next to Assign a security group, choose to create a new security group.  
• In the row for SSH, under Source, choose Anywhere.  
• Choose Add Rule, choose HTTP, and then under Source, choose Anywhere.



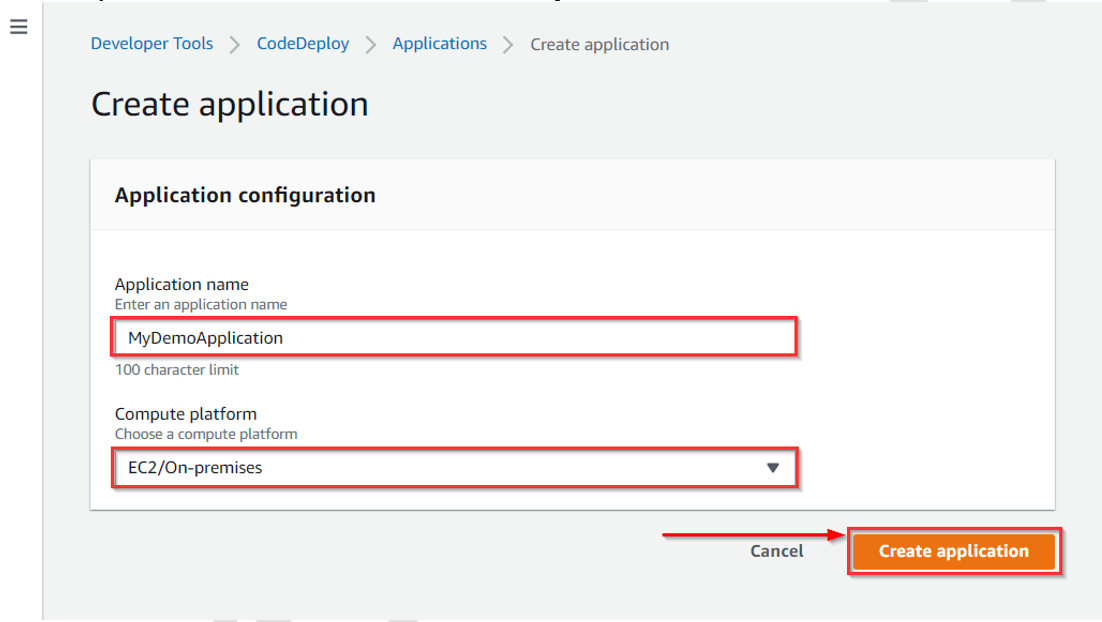
Leave rest settings default and click on **Launch**

### **Step 5: Create an application in Code Deploy**

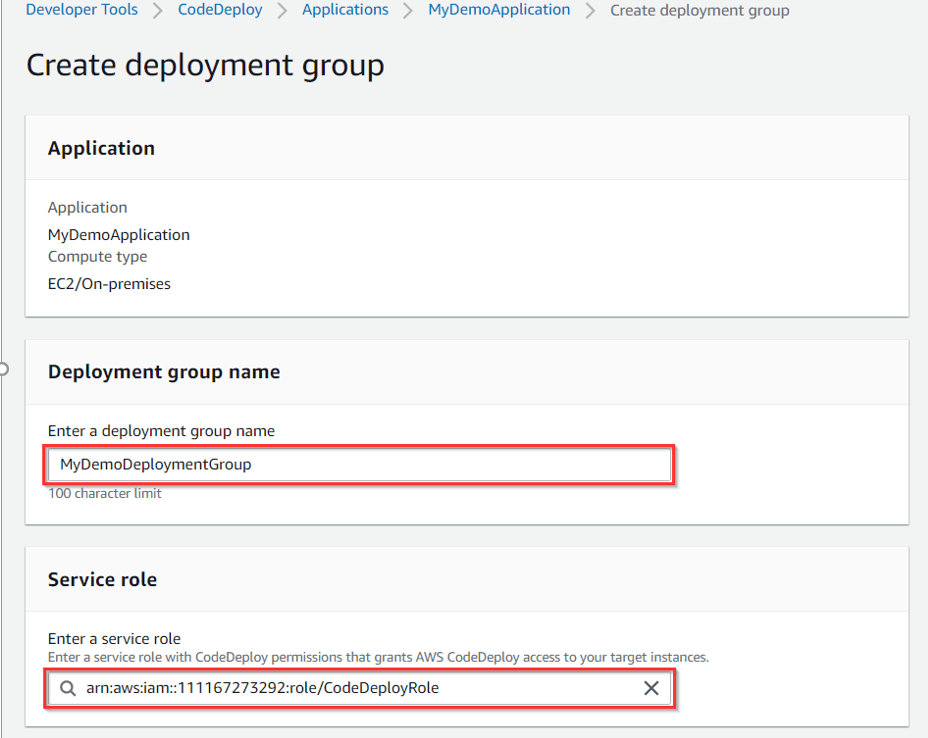
**1)**Create an **AWSCodeDeployRole** that allows Code Deploy to perform deployments



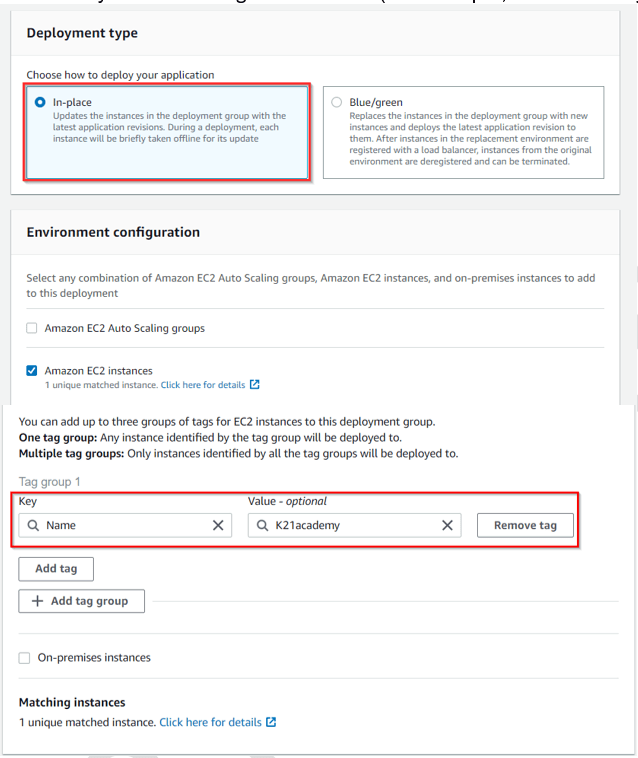
**2)** Initially click on Create an application in **Code Deploy**, and In Compute Platform, choose**EC2/On-premises**. Choose to **Create application.**



1. On the page that displays your application, choose **Create a deployment group.**In service, the role creates an IAM role under the code deploy category.

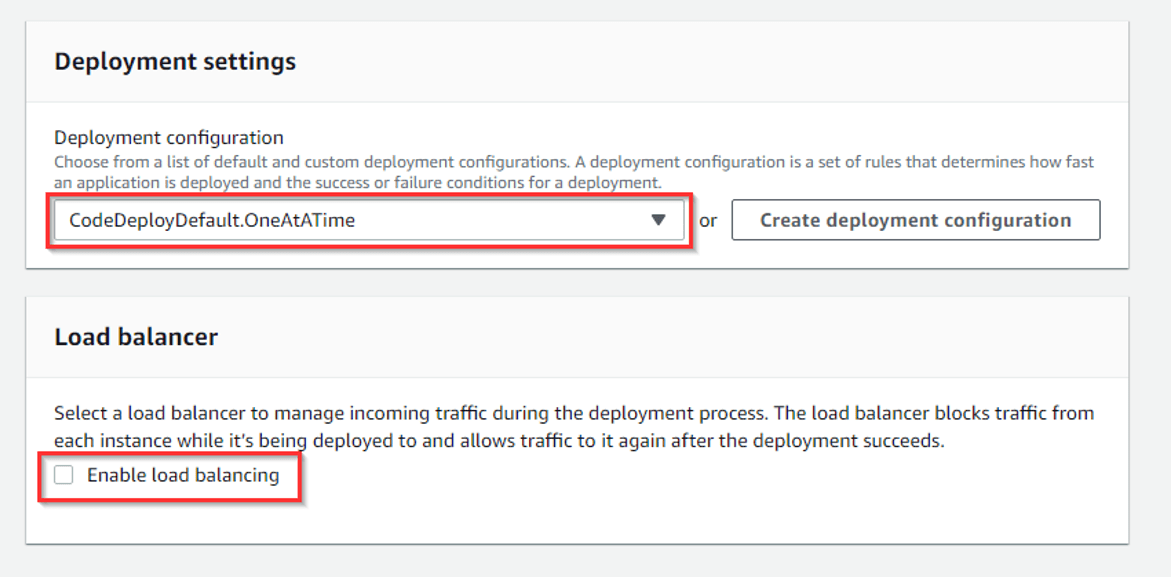


1. Under Deployment type, choose **In-place**. Under the Environment configuration, choose**Amazon EC2 Instances**.



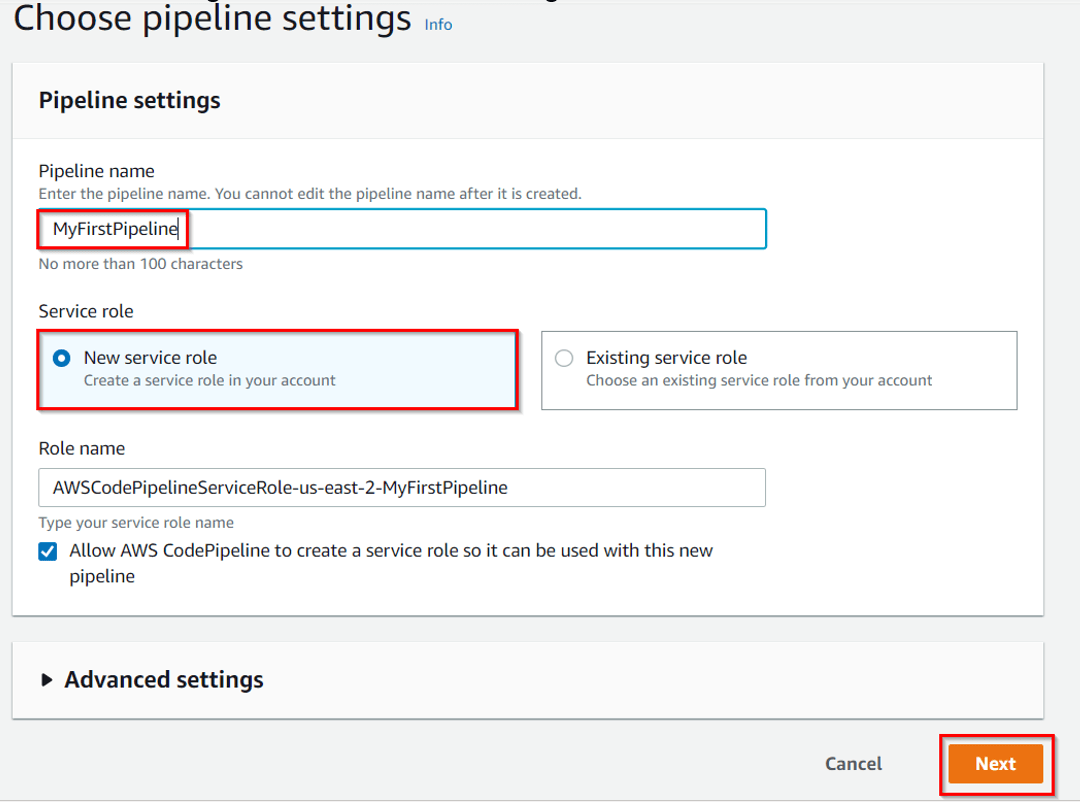
1. Under Deployment configuration, choose**CodeDeployDefault. OneAtaTime.**

Under Load Balancer,**clear Enable load balancing, leave the defaults then choose to Create a deployment group.**

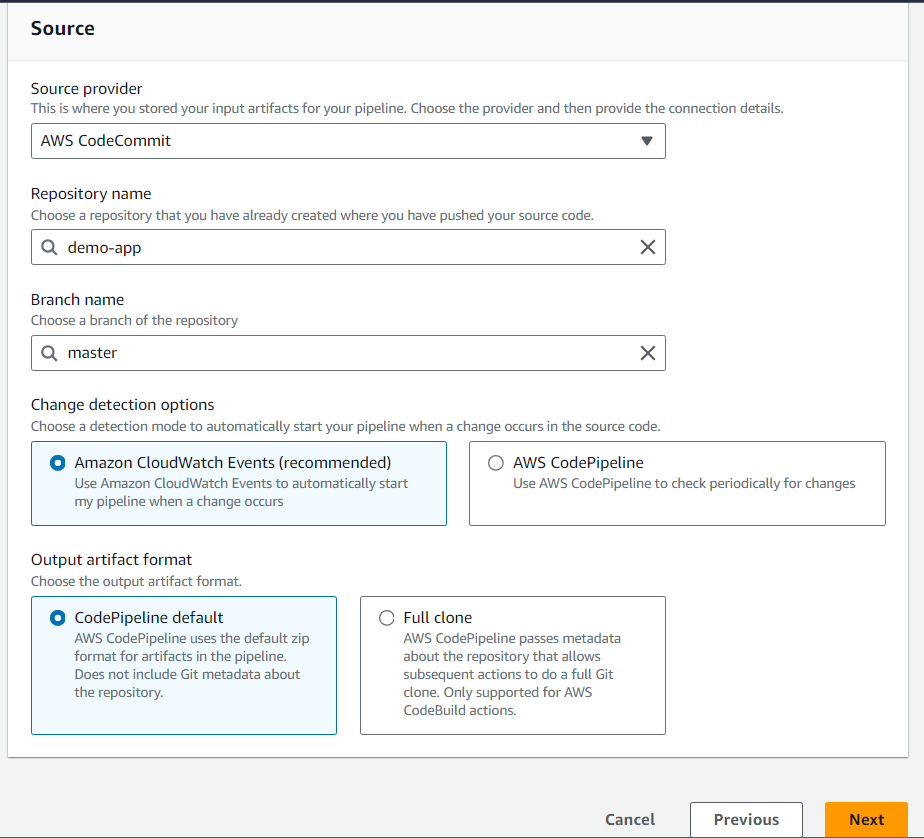


### **Step 6: Create your first pipeline in Code Pipeline**

1. Open the **Code Pipeline console.**Choose pipeline settings, Enter your desired name, and in the **Service role**, Choose New service role to allow Code Pipeline to create a new service role in IAM.

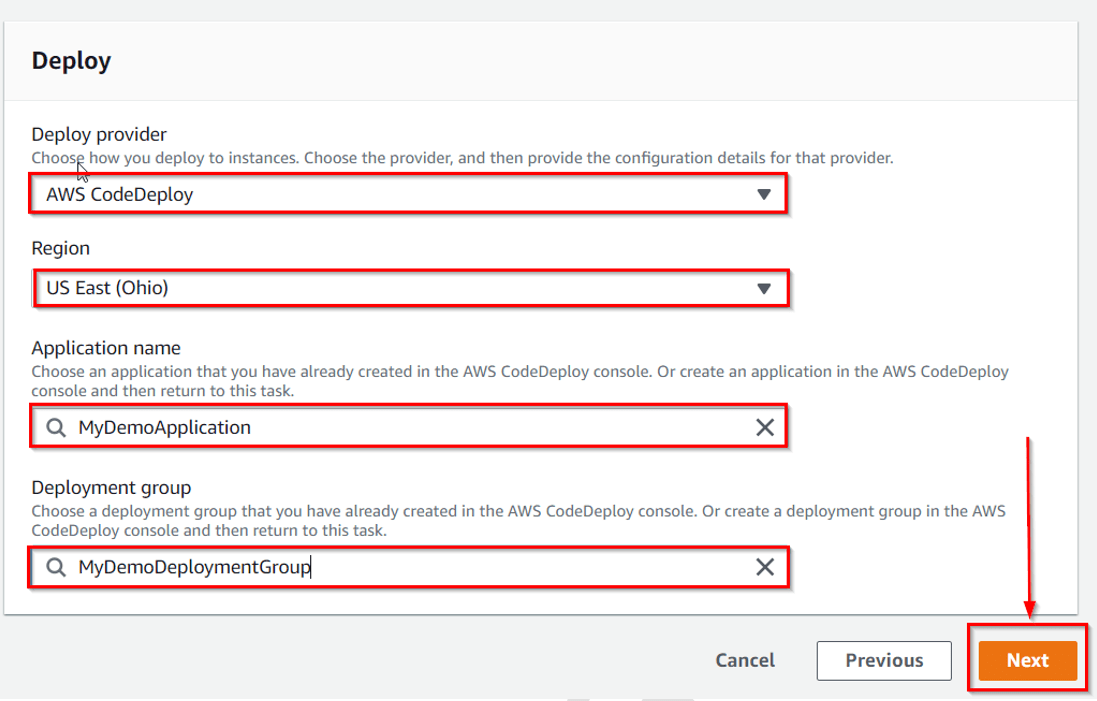


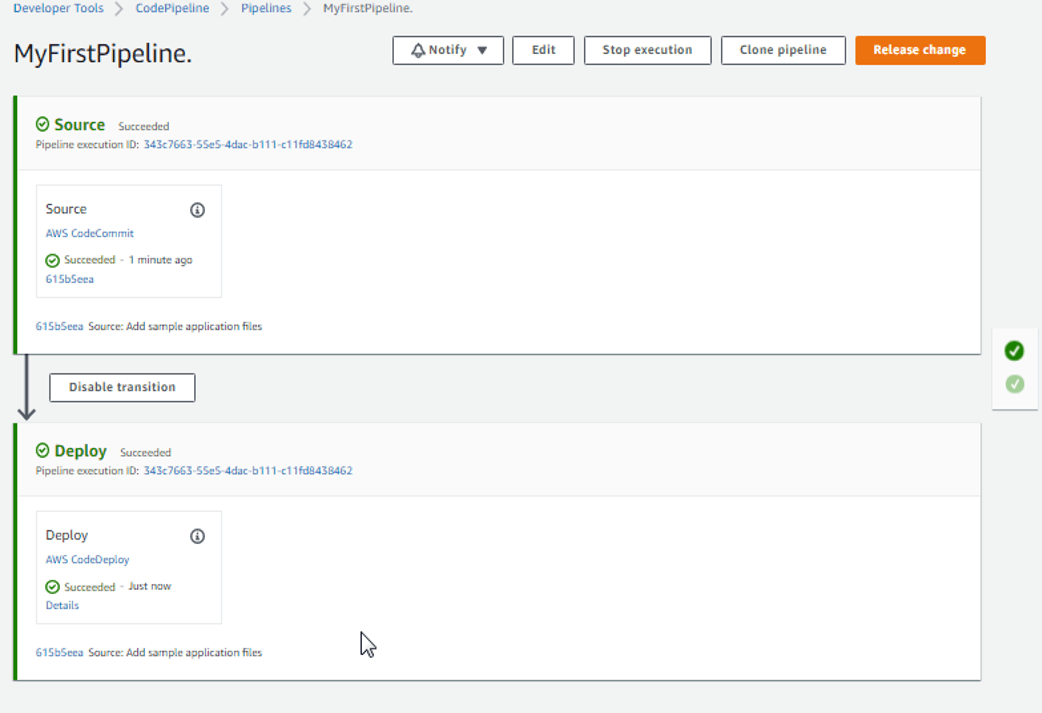
**2)** In the Add source stage, select Source Provider, and choose Amazon S3. Under the S3 object key, enter the object key with or without a file path, and remember to include the file extension.



**3)** In the Add build stage, choose to **Skip build stage**, and then accept the warning message by choosing **Skip again**. Choose **Next**.

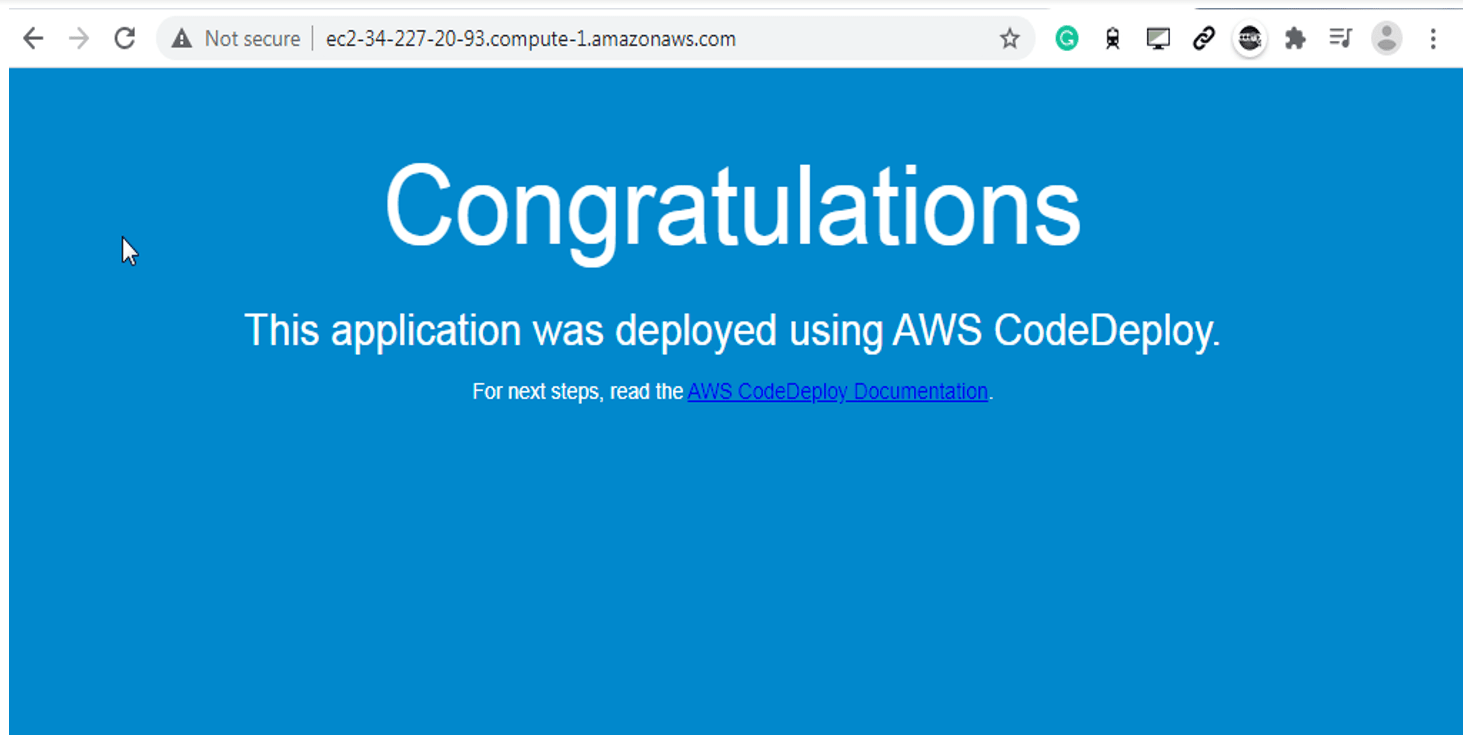
**4)** In the Add deploy stage, in Deploy provider, choose **AWS Code Deploy**. Then enter your application name or choose the application name from the list. In the Deployment group, enter**MyDemoDeploymentGroup**, or choose it from the list, and then choose**Next**.



1. In Step 5: Review, review the information, and then choose **Create pipeline.**The pipeline starts running after it is created. It downloads the code from your Code Commit repository and creates a Code Deploy deployment to your EC2 instance.  
    ****

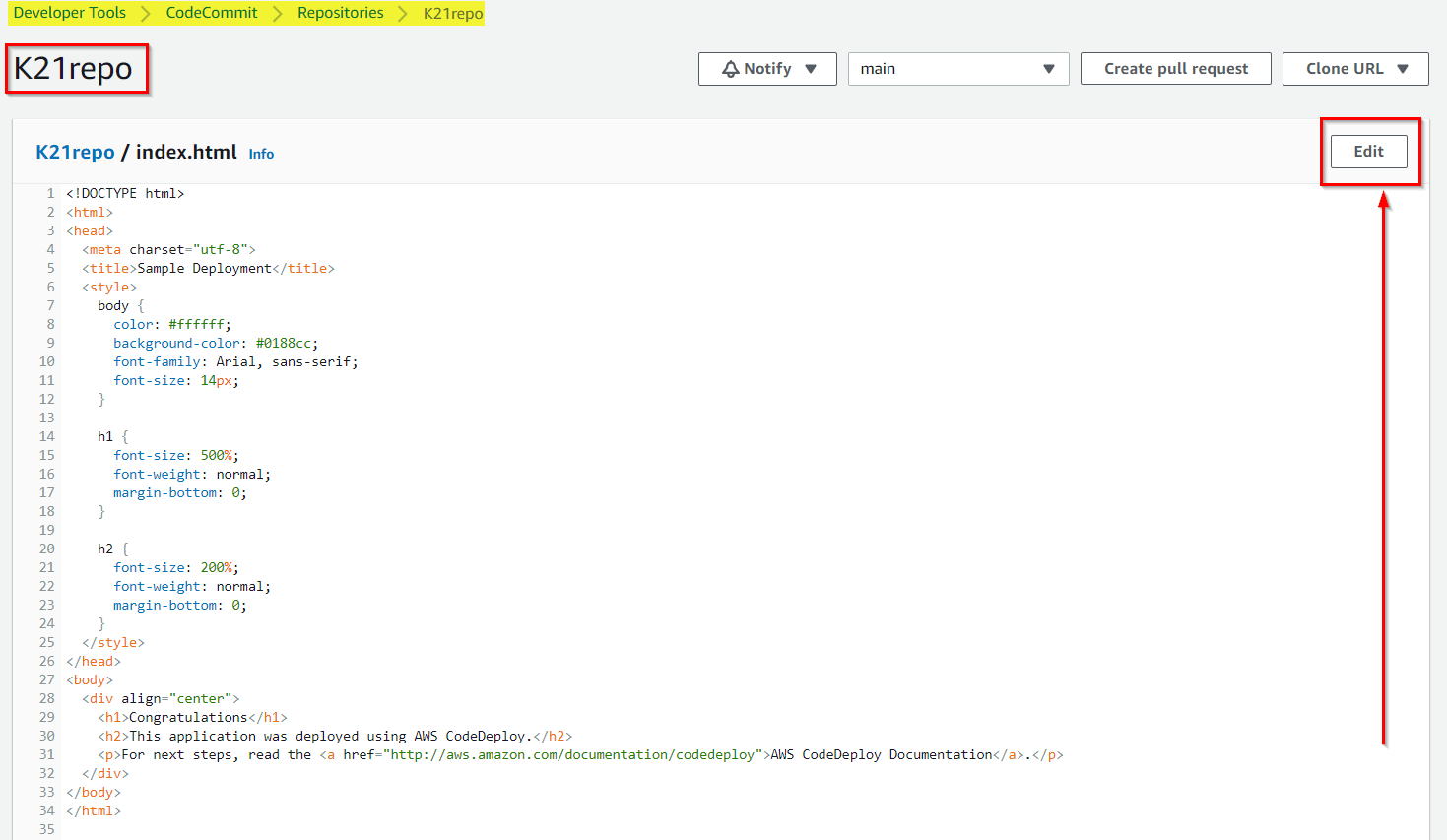
Congratulations! You just created a simple pipeline in CodePipeline.

**you can verify that by coping EC2 Public DNS address and then past it into the address bar of your web browser**

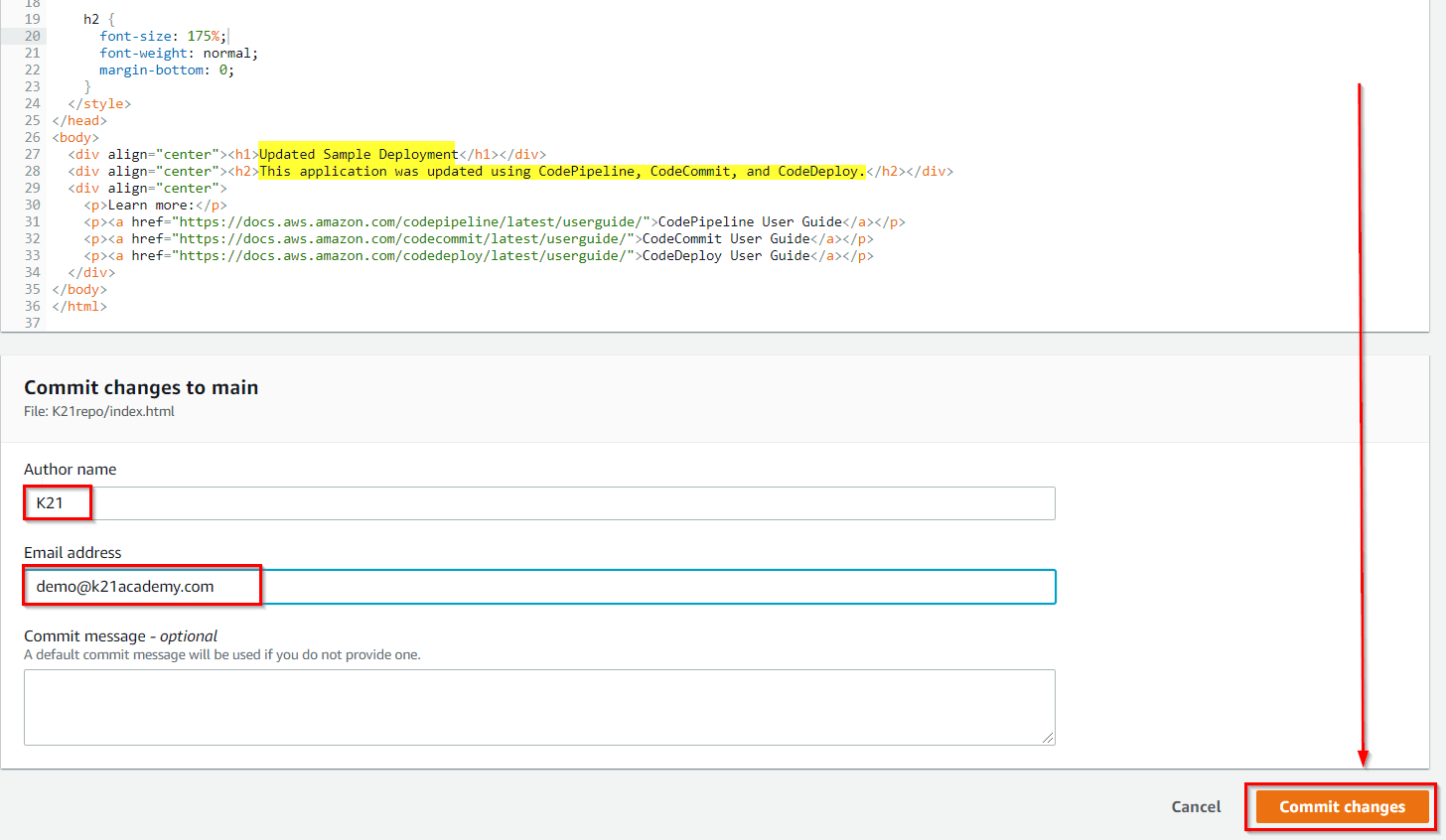
****

### Step 7: Modify Code in your Code Commit Repository

**1)** In your Code Commit repository revise the contents of the **index.html** file to change the background color and some of the text on the webpage, by clicking on the edit



**2)**Under Author name enter **Vikas ,** then enter Email Address, Click on **Commit changes**

****

### Step 8: Verify your Pipeline Ran Successfully

**1)**View the initial progress of the **pipeline**. The status of each stage changes from **No executions yet** to **In Progress**, and then to Succeeded. The running of the pipeline should be complete within a few minutes.

**2)**After Succeeded is displayed for the action status, refresh the demo page you accessed earlier in your browser.

