**Lesson 02 Demo 02**

**Building and Deploying a React Application with AWS CodeBuild and S3**

**Objective:** To automate the continuous integration process by implementing CodePipeline to perform build automation for React source code and deploying it to an S3 bucket

**Tools required:** AWS CodeBuild, AWS CodePipeline, and S3 bucket

**Prerequisites:** EC2 instance creation

Steps to be followed:

1. Create a React GitHub repository
2. Create an S3 bucket
3. Configure CodeBuild and CodePipeline to perform build and test automation

**Step 1: Create a React GitHub repository**

1. Go to the following URL and sign in to your GitHub account:

**https://github.com/**

A screenshot of a chat

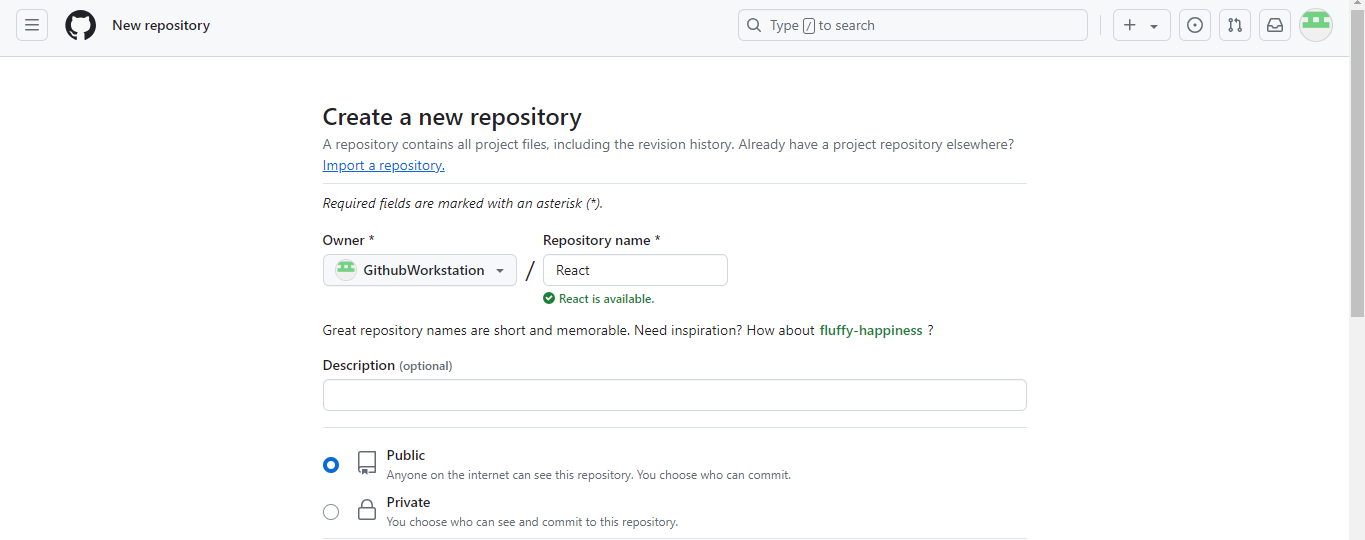
Description automatically generated

1. Click on the **New** button to create a new repository

A screenshot of a chat

Description automatically generated

1. Name the repository as **React**, scroll down, and click on **Create repository**



A screenshot of a computer

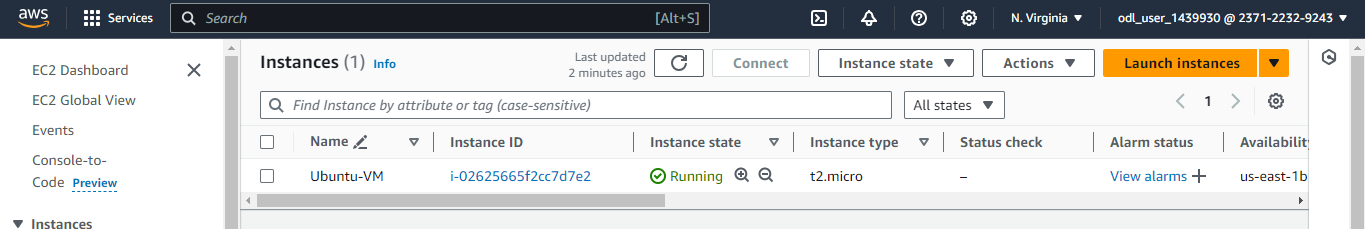
Description automatically generated

A screenshot of a computer

Description automatically generated

The repository is created successfully.

1. Navigate to your AWS Console, create an EC2 instance, and connect to it



1. Create a directory using the following command:

**mkdir demo**



1. Navigate inside the created directory using the following command:

**cd demo**



1. Initialize Git using the following command:

**git init**

**A screenshot of a computer program

Description automatically generated**

1. Clone the GitHub repository using the following command:

**git clone --mirror** [**https://github.com/anujdevopslearn/SonarQubeNodeJS**](https://github.com/anujdevopslearn/SonarQubeNodeJS)

**A computer screen with white text

Description automatically generated**

1. Switch to the cloned repository using the following command:

**cd SonarQubeNodeJS.git**



1. List all files present in the repository using the following command:

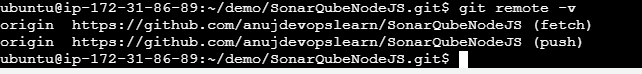
**ls -lart**

A screen shot of a computer

Description automatically generated

1. List all remote repositories using the following command:

**git remote -v**



1. Remove the existing remote configuration using the following command:

**git remote remove origin**

A black screen with white text

Description automatically generated

1. Add a new remote repository using the following command:

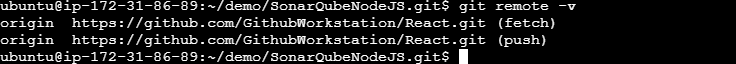
**git remote add origin Repository-link**



|  |
| --- |
| Note: Replace **Repository-link** with the link to the repository you created in Step 1.3 |

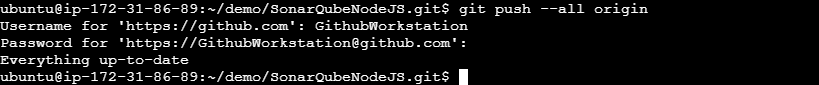
1. Verify if the new remote repository is correctly configured by using the following command:

**git remote -v**



1. Upload your local repository content to the remote repository using the following command:

**git push --all origin**



|  |
| --- |
| Note: When prompted, enter your GitHub credentials. |

**Step 2: Create an S3 bucket**

1. Go to the AWS Console

A screenshot of a computer

Description automatically generated

1. Search for **S3** and click on it  
     
   A screenshot of a computer

   Description automatically generated
2. Click on the **Create bucket** button  
     
   A screenshot of a computer

   Description automatically generated
3. Enter the name of the bucket  
     
   A screenshot of a computer

   Description automatically generated
4. Scroll down and click on the **Create** **bucket** button  
     
   A screenshot of a computer

   Description automatically generated  
     
   A screenshot of a computer

   Description automatically generated

The S3 bucket is created successfully.

**Step 3: Configure CodeBuild and CodePipeline to perform build and test**

**automation**

1. Click on the **Create pipeline** button

**A screenshot of a computer

Description automatically generated**

1. Enter the **Pipeline name**, and click the **Next** button

A screenshot of a computer

Description automatically generated

1. Select **GitHub (Version 1)** as the **Source provider** and click on **Connect to GitHub**

**A screenshot of a computer

Description automatically generated**

1. When prompted, click on **Confirm**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

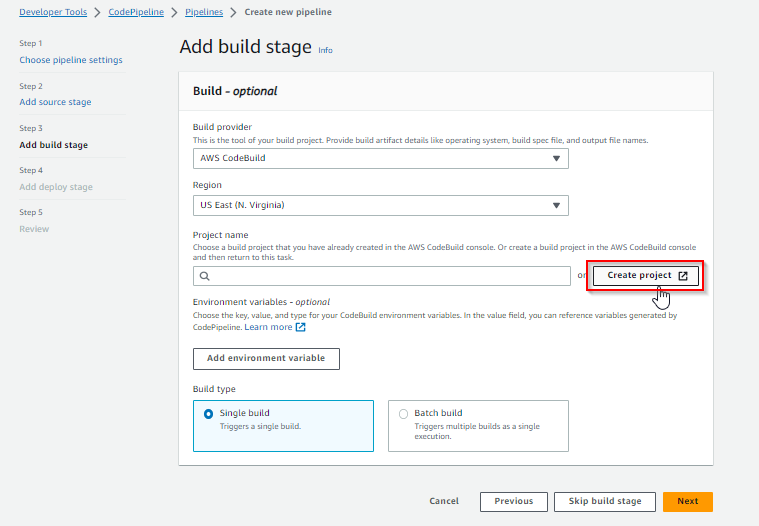
GitHub is configured successfully.

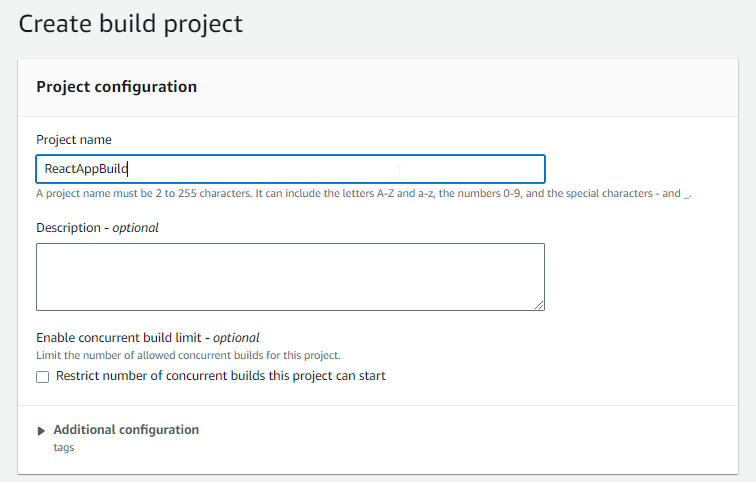
1. Enter the created repository name, select **master** under the branch section, and click on **Next**

**A screenshot of a computer

Description automatically generated**

1. Select **AWS CodeBuild** as the **Build provider** for building and automating tests  
     
   A screenshot of a computer

   Description automatically generated
2. Then, click on the **Create project** button, and a new pop-up window will appear  
     
   
3. In the new window, enter a name for the project



1. In the **Environment** section, add the details as shown in the following screenshots:

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. In the **Buildspec** section, select the **Insert build commands** option and click on the **Switch to editor** button  
     
   A screenshot of a computer

   Description automatically generated
2. Remove the existing build commands from the editor and enter the following YAML code:  
   **version: 0.2**

**phases:**

**build:**

**commands:**

**- "ls -alrt"**

**- "npm install"**

**- "npm run build"**

**- "npm run test"**

**- "npm run test:e2e"**

**- "zip -r dist.zip dist"**

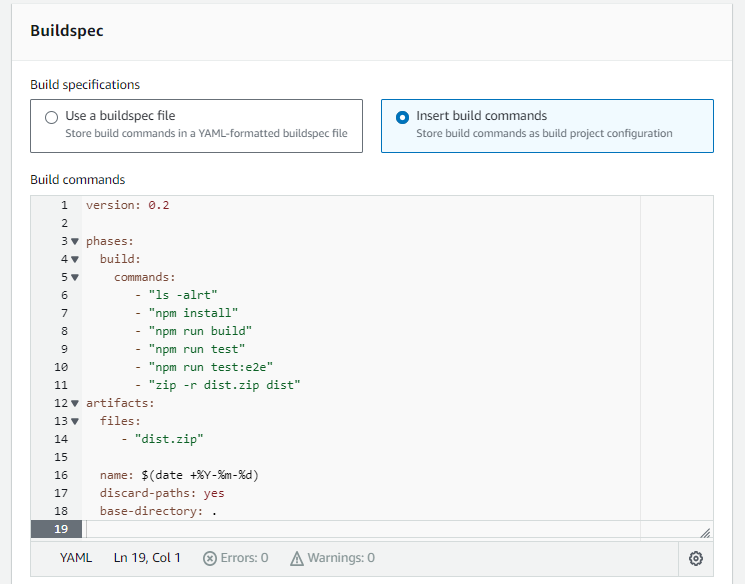
**artifacts:**

**files:**

**- "dist.zip"**

**name: $(date +%Y-%m-%d)**

**discard-paths: yes**

**base-directory: .  
  
  
  
  
  
  
  
  
**

1. Scroll down to the bottom of the page and click on the **Continue to CodePipeline** button **A screenshot of a computer

   Description automatically generated**

**A screenshot of a computer

Description automatically generated**

The project is successfully created.

1. Now, click the **Next** button  
     
   **A screenshot of a computer

   Description automatically generated**
2. In the deploy stage, select the **Amazon S3** option  
     
   **A screenshot of a computer

   Description automatically generated**
3. Select the **codepipeline-deploy-bucket-simplilearn** option under **Bucket**

**A screenshot of a computer

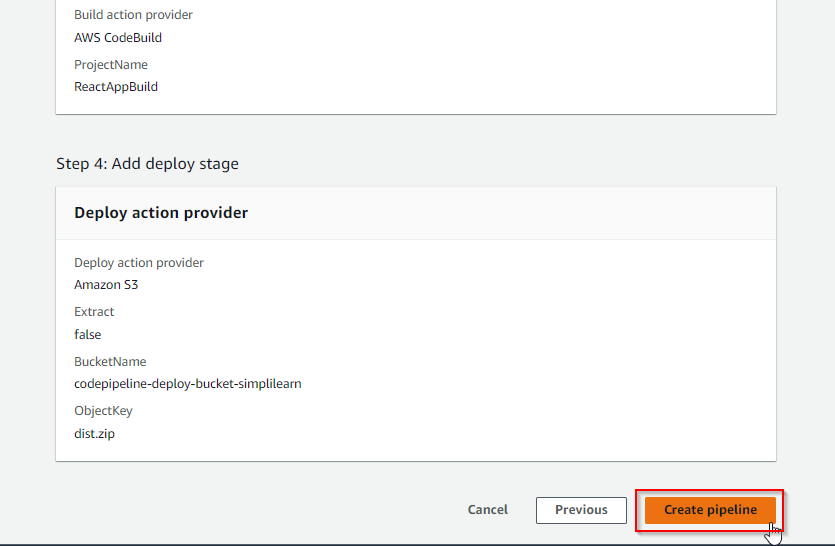
Description automatically generated**

1. Enter **dist.zip** in the **S3 object key** section **A screenshot of a computer

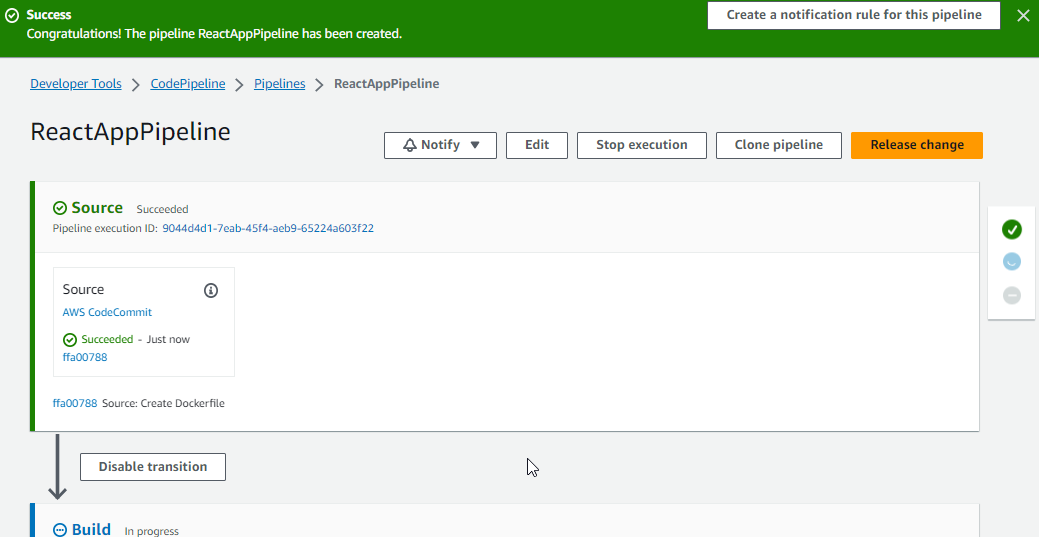
   Description automatically generated**
2. Click on the **Next** button

A screenshot of a computer

Description automatically generated

1. Scroll to the bottom of the page and click on the **Create pipeline** button ****

After creating the pipeline, the execution will start.



**A screenshot of a computer

Description automatically generated**

1. Once the build stage is complete, click on the **View logs** button in the **Build** section to validate the process

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

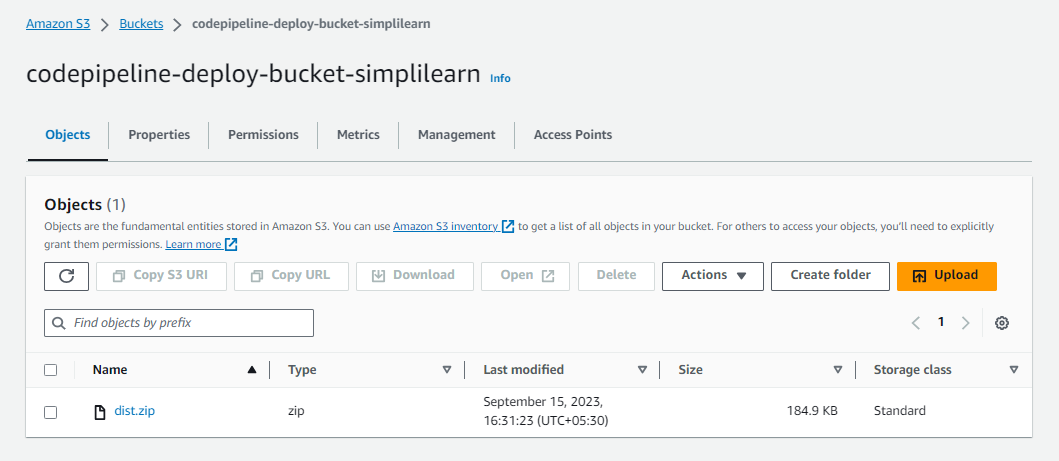
Description automatically generated**

1. Once the **Build** and **Deploy** stages are complete, validate whether the artifact was deployed to the S3 bucket by clicking on **Amazon S3**  
   A screenshot of a computer

   Description automatically generated
2. Now, click on the bucket name **codepipeline-deploy-bucket-simplilearn  
     
   A screenshot of a computer

   Description automatically generated**

The following interface appears after successfully building and deploying a React application with AWS CodeBuild and S3:

****

By following these steps, you have successfully completed the process of building and deploying a React application using AWS services, including CodeBuild and S3.