

Lesson 03 Demo 04

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

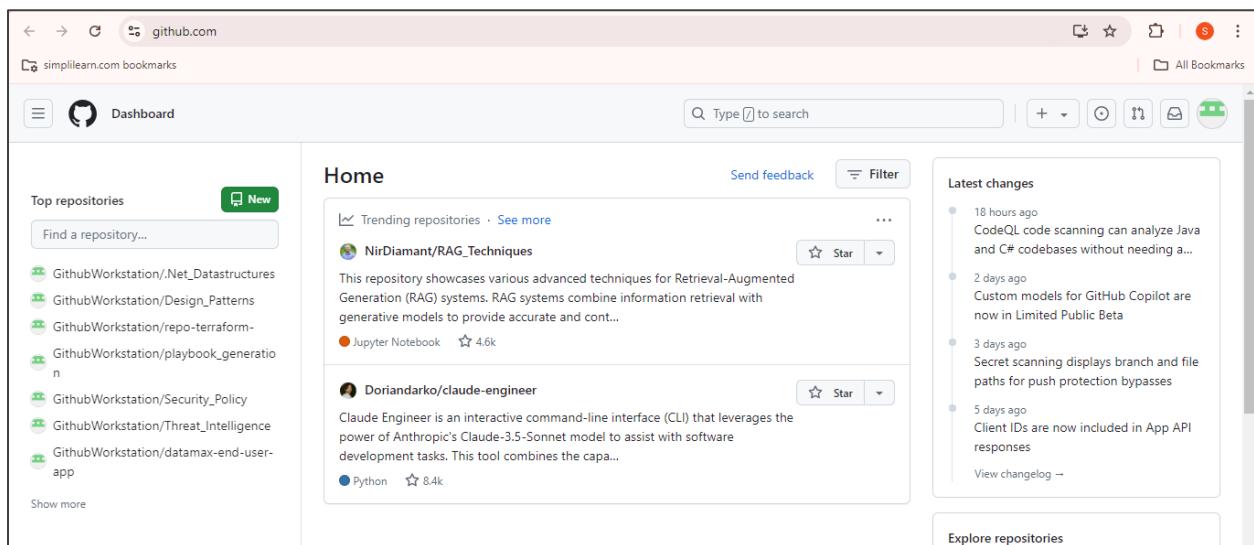
Steps to be followed:

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

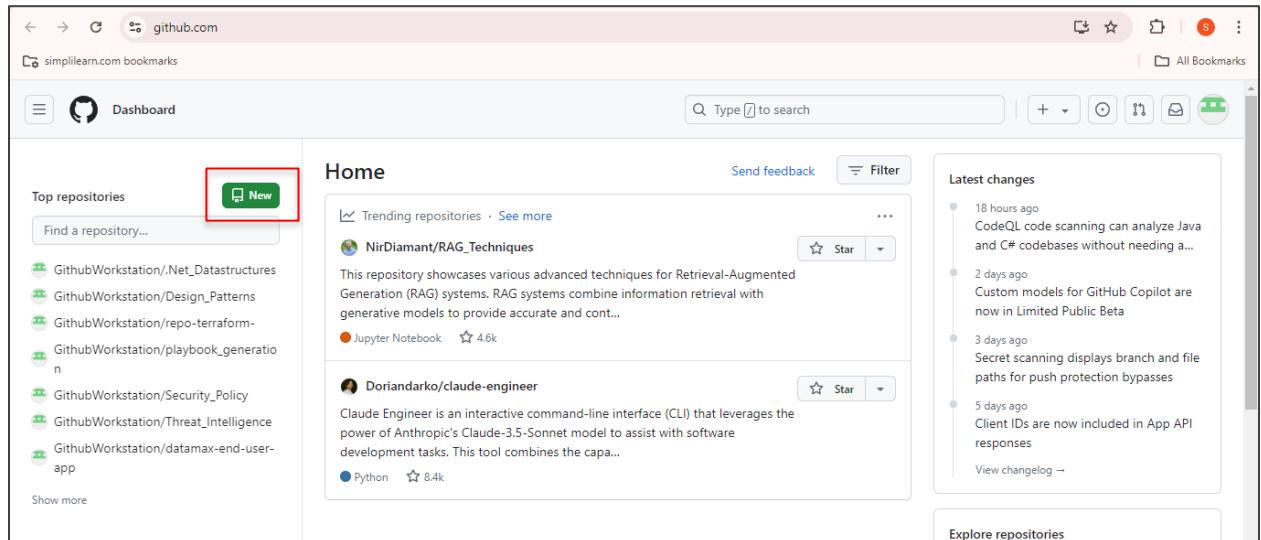
Step 1: Create a React GitHub repository

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

<https://github.com/>

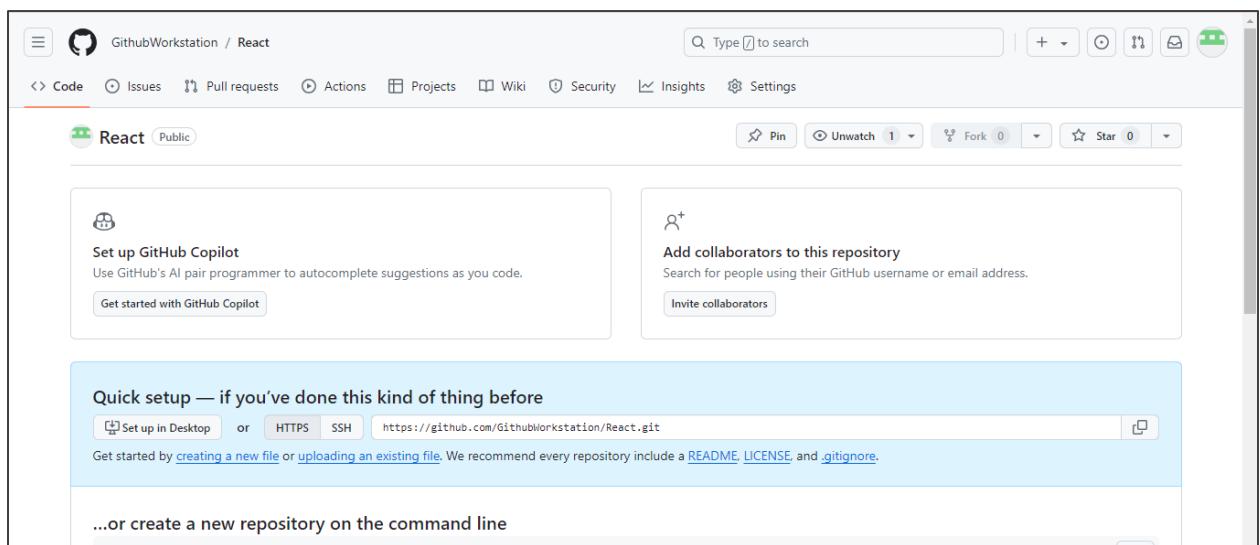


1.2 Click on the New button to create a new repository



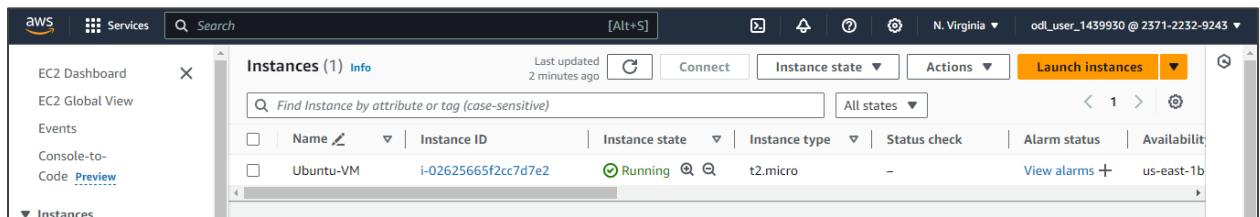
1.3 Name the repository as React, scroll down, and click on Create repository

The screenshot shows the 'Create a new repository' form. At the top, there's a header 'Create a new repository' and a note about project files. Below that, there's a 'Required fields are marked with an asterisk (*).'. The 'Owner' dropdown is set to 'GithubWorkstation' and the 'Repository name *' input field contains 'React', which is also highlighted with a red rectangle. There's a note below the input field stating 'React is available.' The 'Description (optional)' field is empty. Under 'Visibility', the 'Public' radio button is selected, with a note that anyone can see the repository. The 'Private' radio button is also present. The bottom section contains fields for 'Add .gitignore' (with a dropdown for '.gitignore template: None'), 'Choose which files not to track from a list of templates.', 'Choose a license' (with a dropdown for 'License: None'), and a note that a license tells others what they can and can't do with your code. At the very bottom, a note says '(1) You are creating a public repository in your personal account.' and the 'Create repository' button is highlighted with a red rectangle.



The repository is created successfully.

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



1.5 Create a directory using the following command:

mkdir demo

```
ubuntu@ip-172-31-86-89:~$ mkdir demo
ubuntu@ip-172-31-86-89:~$
```

1.6 Navigate inside the created directory using the following command:

cd demo

```
ubuntu@ip-172-31-86-89:~$ cd demo
ubuntu@ip-172-31-86-89:~/demo$
```

1.7 Initialize Git using the following command:

```
git init
```

```
ubuntu@ip-172-31-86-89:~/demo$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ubuntu/demo/.git/
ubuntu@ip-172-31-86-89:~/demo$ █
```

1.8 Clone the GitHub repository using the following command:

```
git clone --mirror https://github.com/anujdevopslearn/SonarQubeNodeJS
```

```
ubuntu@ip-172-31-86-89:~/demo$ git clone --mirror https://github.com/anujdevopslearn/SonarQubeNodeJS
Cloning into bare repository 'SonarQubeNodeJS.git'...
remote: Enumerating objects: 205, done.
remote: Counting objects: 100% (104/104), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 205 (delta 59), reused 69 (delta 31), pack-reused 101 (from 1)
Receiving objects: 100% (205/205), 212.16 KiB | 12.48 MiB/s, done.
Resolving deltas: 100% (67/67), done.
ubuntu@ip-172-31-86-89:~/demo$ █
```

1.9 Switch to the cloned repository using the following command:

```
cd SonarQubeNodeJS.git
```

```
ubuntu@ip-172-31-86-89:~/demo$ cd SonarQubeNodeJS.git
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ █
```

1.10 List all files present in the repository using the following command:

ls -lart

```
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ ls -lart
total 44
drwxrwxr-x 2 ubuntu ubuntu 4096 Aug 29 12:13 info
drwxrwxr-x 2 ubuntu ubuntu 4096 Aug 29 12:13 hooks
-rw-rw-r-- 1 ubuntu ubuntu    73 Aug 29 12:13 description
drwxrwxr-x 2 ubuntu ubuntu 4096 Aug 29 12:13 branches
drwxrwxr-x 4 ubuntu ubuntu 4096 Aug 29 12:13 ..
drwxrwxr-x 4 ubuntu ubuntu 4096 Aug 29 12:13 refs
drwxrwxr-x 4 ubuntu ubuntu 4096 Aug 29 12:13 objects
-rw-rw-r-- 1 ubuntu ubuntu   181 Aug 29 12:13 config
-rw-rw-r-- 1 ubuntu ubuntu   165 Aug 29 12:13 packed-refs
-rw-rw-r-- 1 ubuntu ubuntu    23 Aug 29 12:13 HEAD
drwxrwxr-x 7 ubuntu ubuntu 4096 Aug 29 12:13 .
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$
```

1.11 List all remote repositories using the following command:

git remote -v

```
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ git remote -v
origin https://github.com/anujdevopslearn/SonarQubeNodeJS (fetch)
origin https://github.com/anujdevopslearn/SonarQubeNodeJS (push)
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$
```

1.12 Remove the existing remote configuration using the following command:

git remote remove origin

```
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ git remote remove origin
Note: Some branches outside the refs/remotes/ hierarchy were not removed;
      to delete them, use:
        git branch -d develop
        git branch -d master
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$
```

1.13 Add a new remote repository using the following command:

git remote add origin Repository-link

```
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ git remote add origin https://github.com/GithubWorkstation/React.git
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$
```

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

- 1.14 Verify if the new remote repository is correctly configured by using the following command:

```
git remote -v
```

```
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ git remote -v
origin  https://github.com/GithubWorkstation/React.git (fetch)
origin  https://github.com/GithubWorkstation/React.git (push)
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ █
```

- 1.15 Upload your local repository content to the remote repository using the following command:

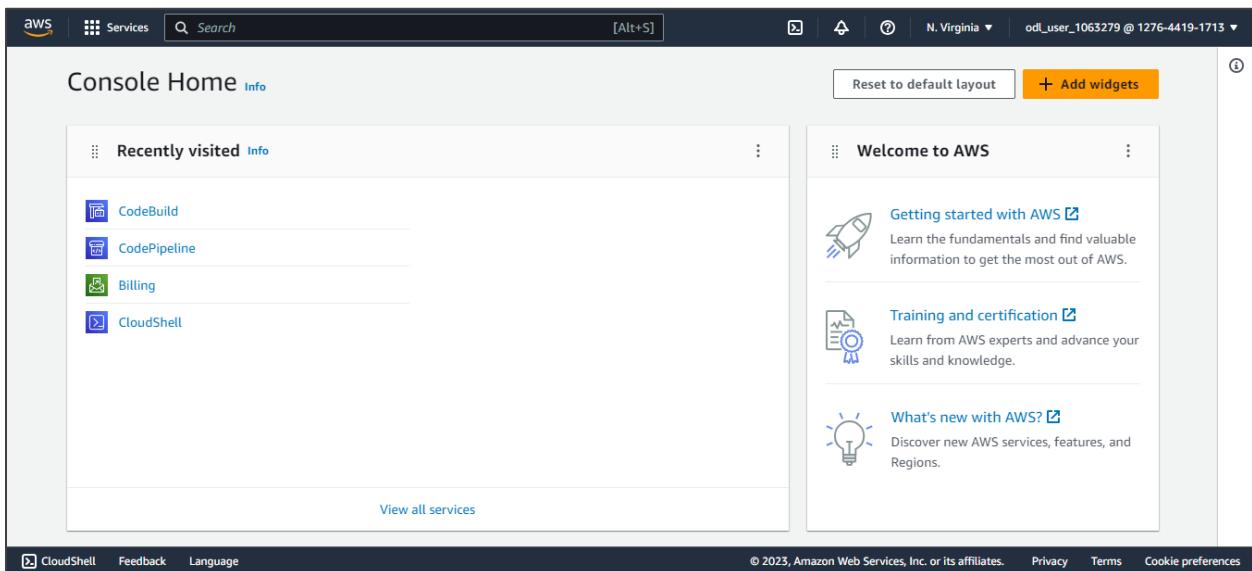
```
git push --all origin
```

```
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ git push --all origin
Username for 'https://github.com': GithubWorkstation
Password for 'https://GithubWorkstation@github.com':
Everything up-to-date
ubuntu@ip-172-31-86-89:~/demo/SonarQubeNodeJS.git$ █
```

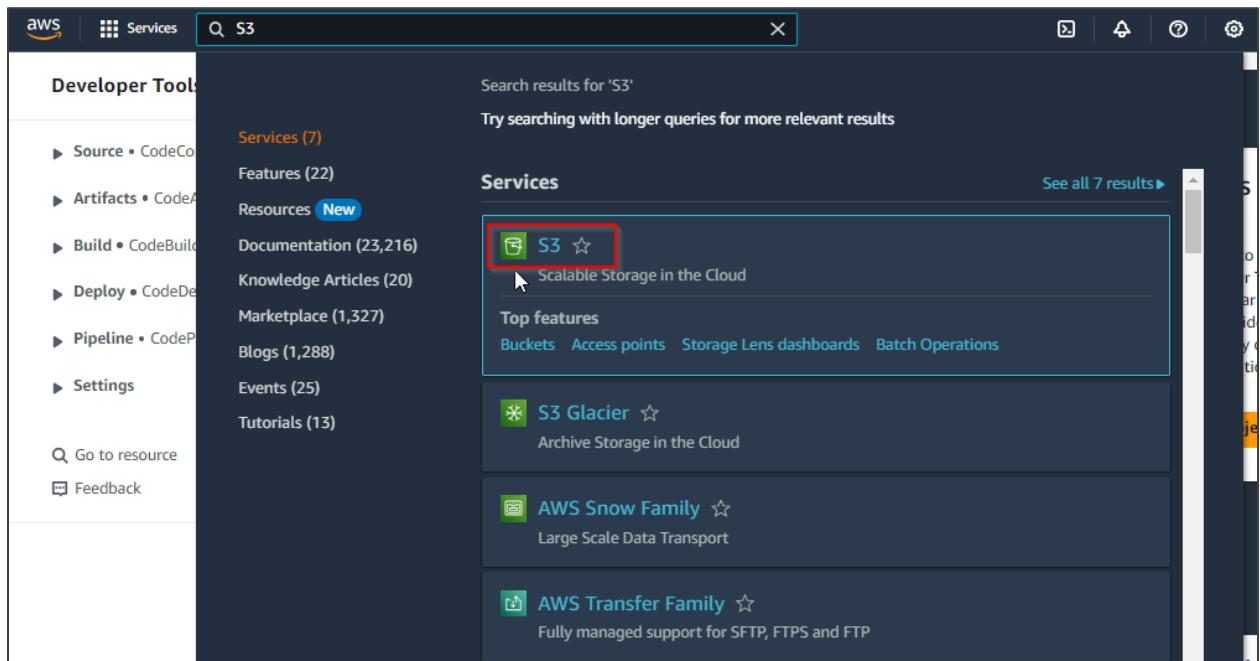
Note: When prompted, enter your GitHub credentials.

Step 2: Create an S3 bucket

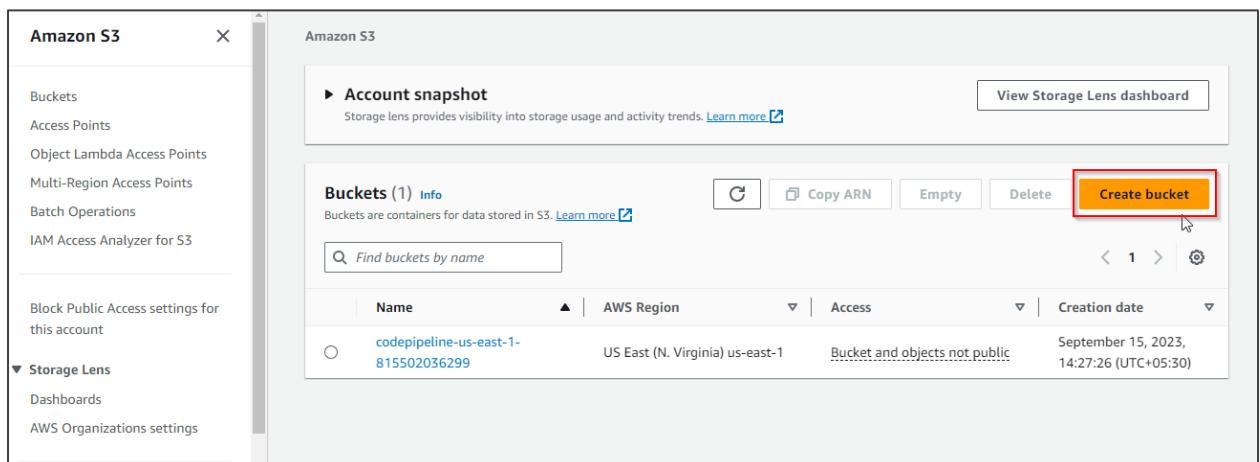
2.1 Go to the AWS Console



2.2 Search for S3 and click on it



2.3 Click on the Create bucket button



2.4 Enter the name of the bucket

Create bucket Info

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

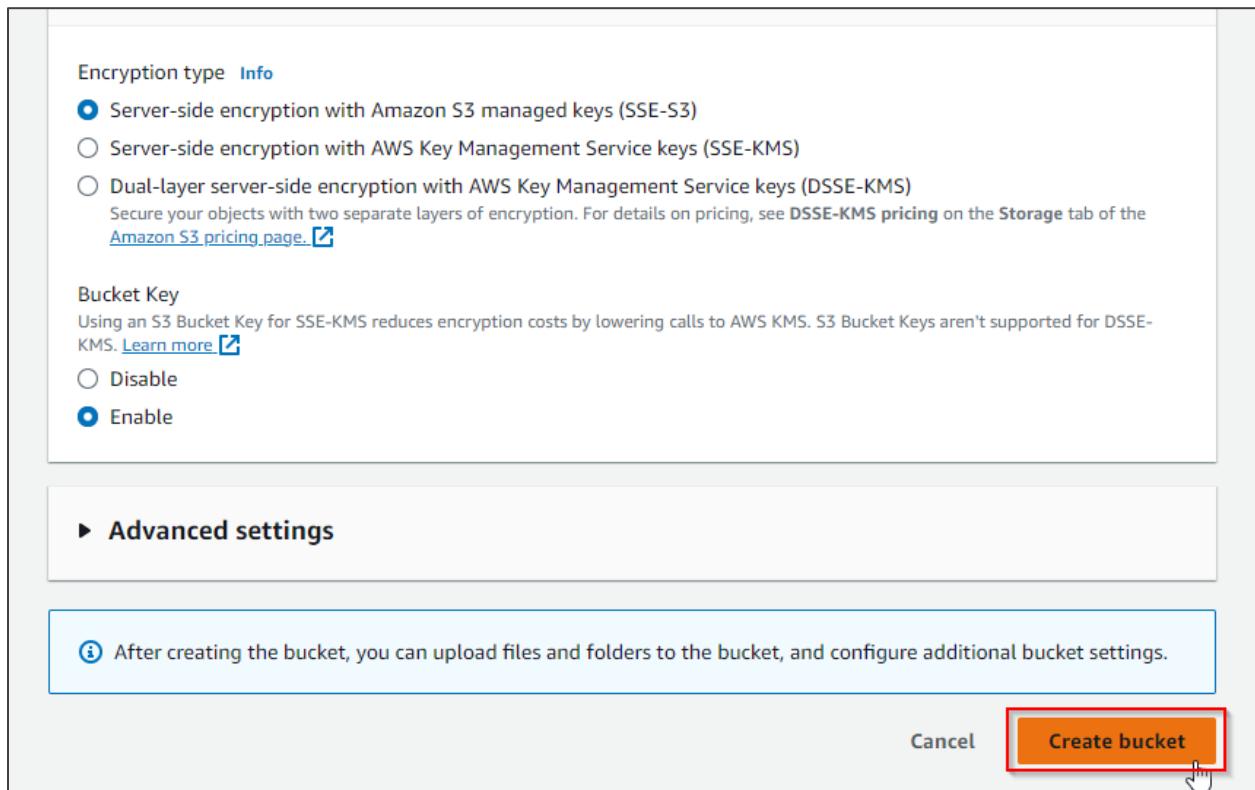
Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

2.5 Scroll down and click on the **Create bucket** button



Successfully created bucket "codepipeline-deploy-bucket-simplilearn"
To upload files and folders, or to configure additional bucket settings choose [View details](#).

[Amazon S3](#) > Buckets

Account snapshot
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

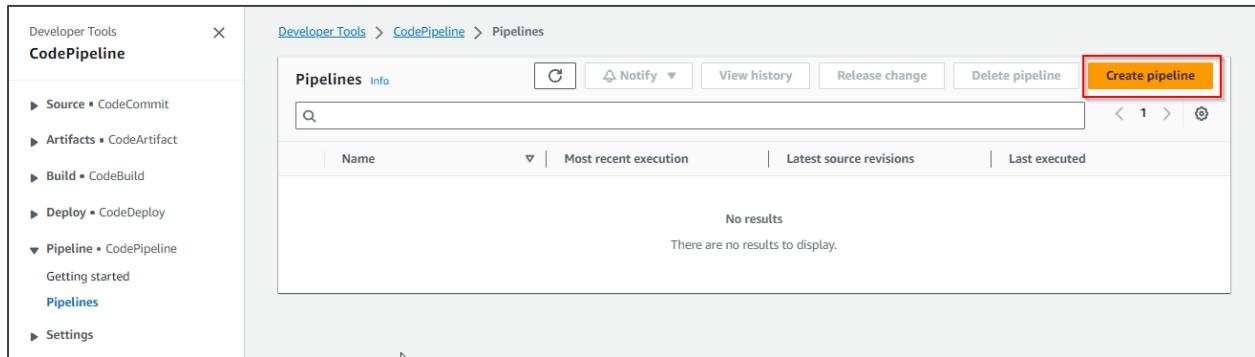
Buckets (2) [Info](#)
Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
codepipeline-deploy-bucket-simplilearn	US East (N. Virginia) us-east-1	Bucket and objects not public	September 15, 2023, 16:15:51 (UTC+05:30)
codepipeline-us-east-1-815502036299	US East (N. Virginia) us-east-1	Bucket and objects not public	September 15, 2023, 14:27:26 (UTC+05:30)

The S3 bucket is created successfully.

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

3.1 Click on the **Create pipeline** button



3.2 Enter the **Pipeline name**, and click the **Next** button

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
 No more than 100 characters

Service role
 New service role Create a service role in your account Existing service role Choose an existing service role from your account

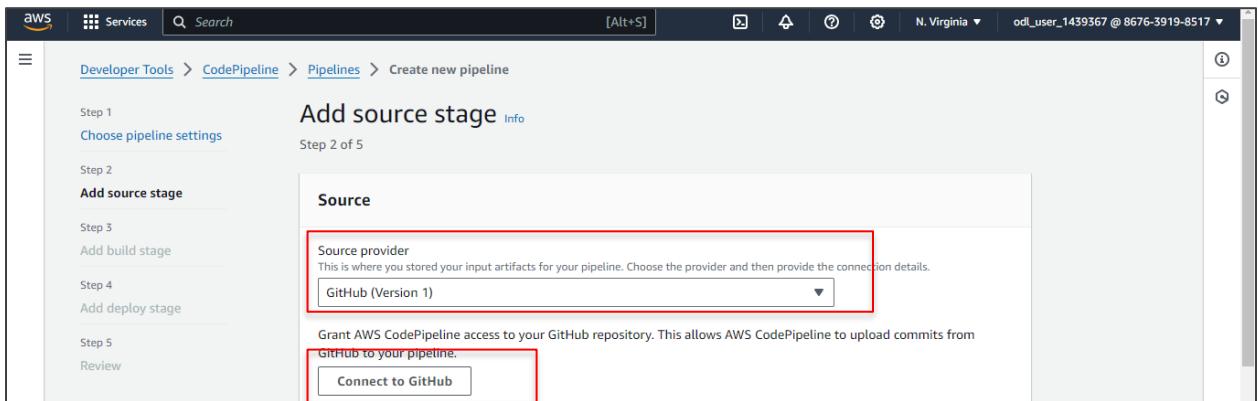
Role name

Type your service role name
 Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

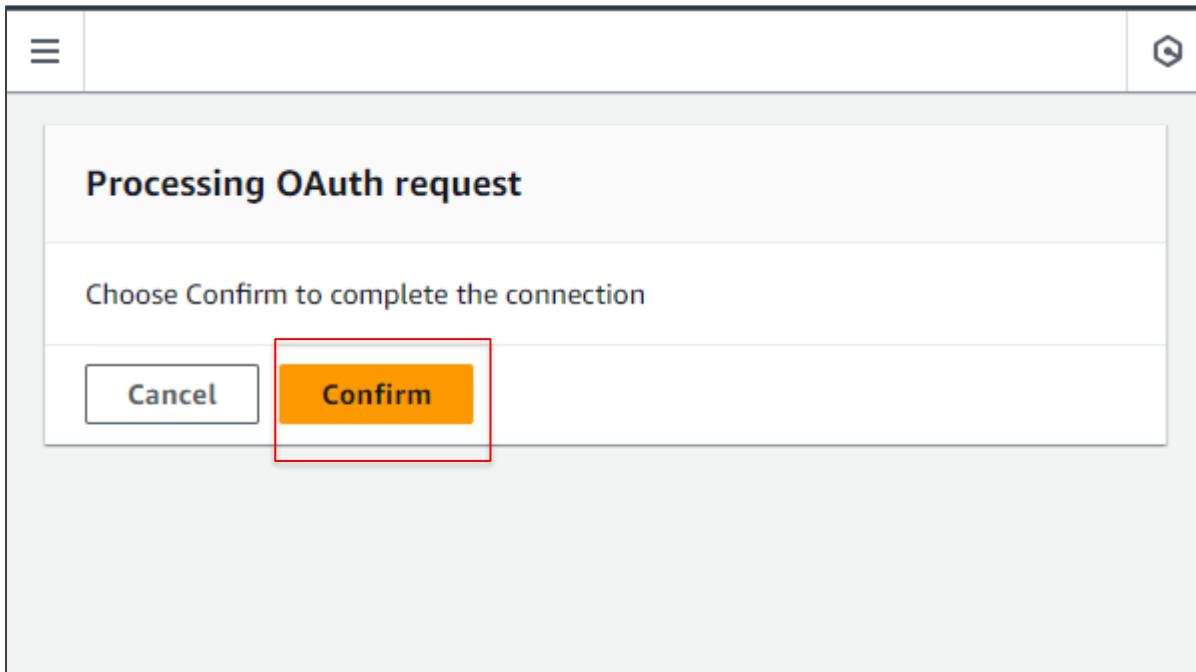
Advanced settings

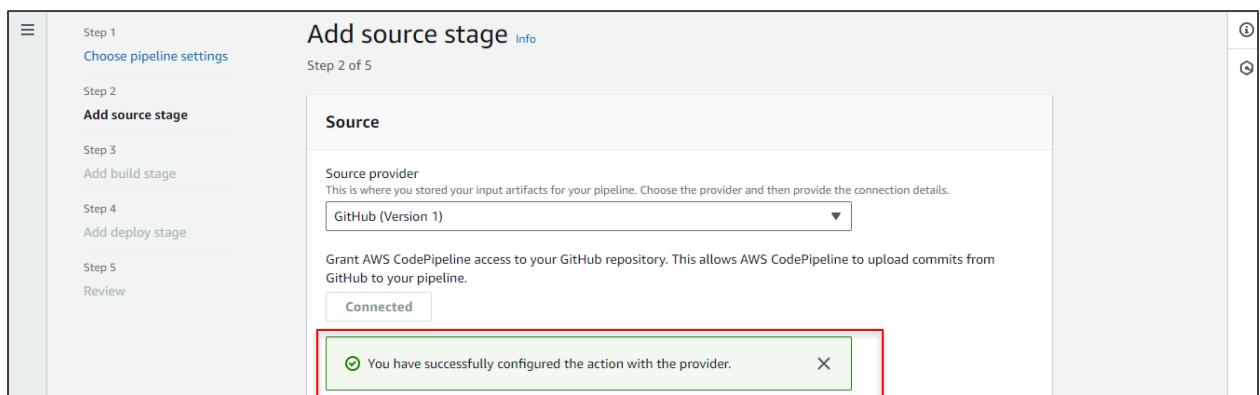
Cancel **Next**

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



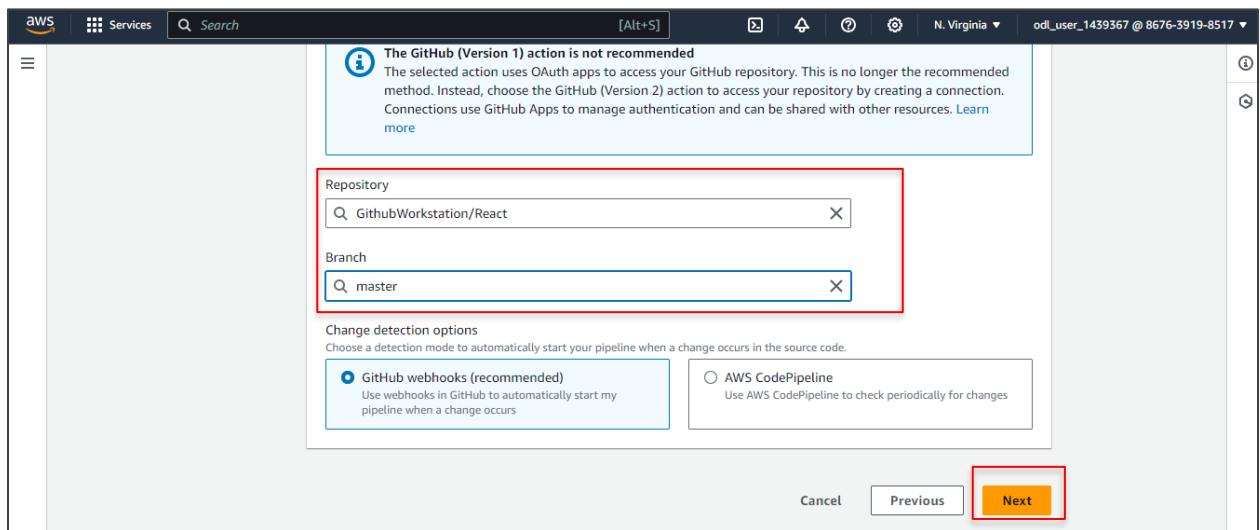
3.4 When prompted, click on Confirm



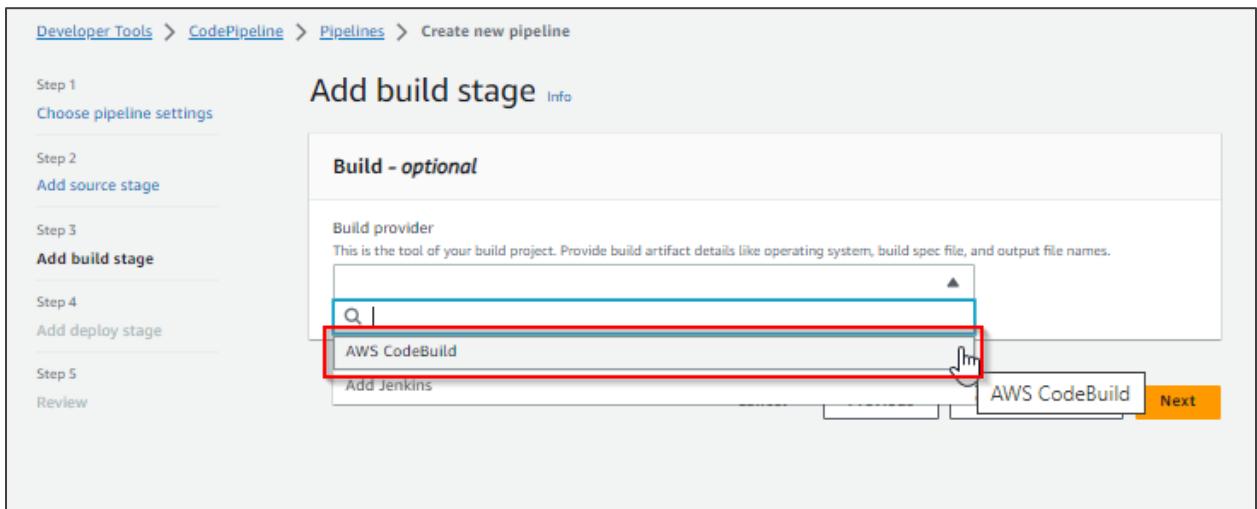


GitHub is configured successfully.

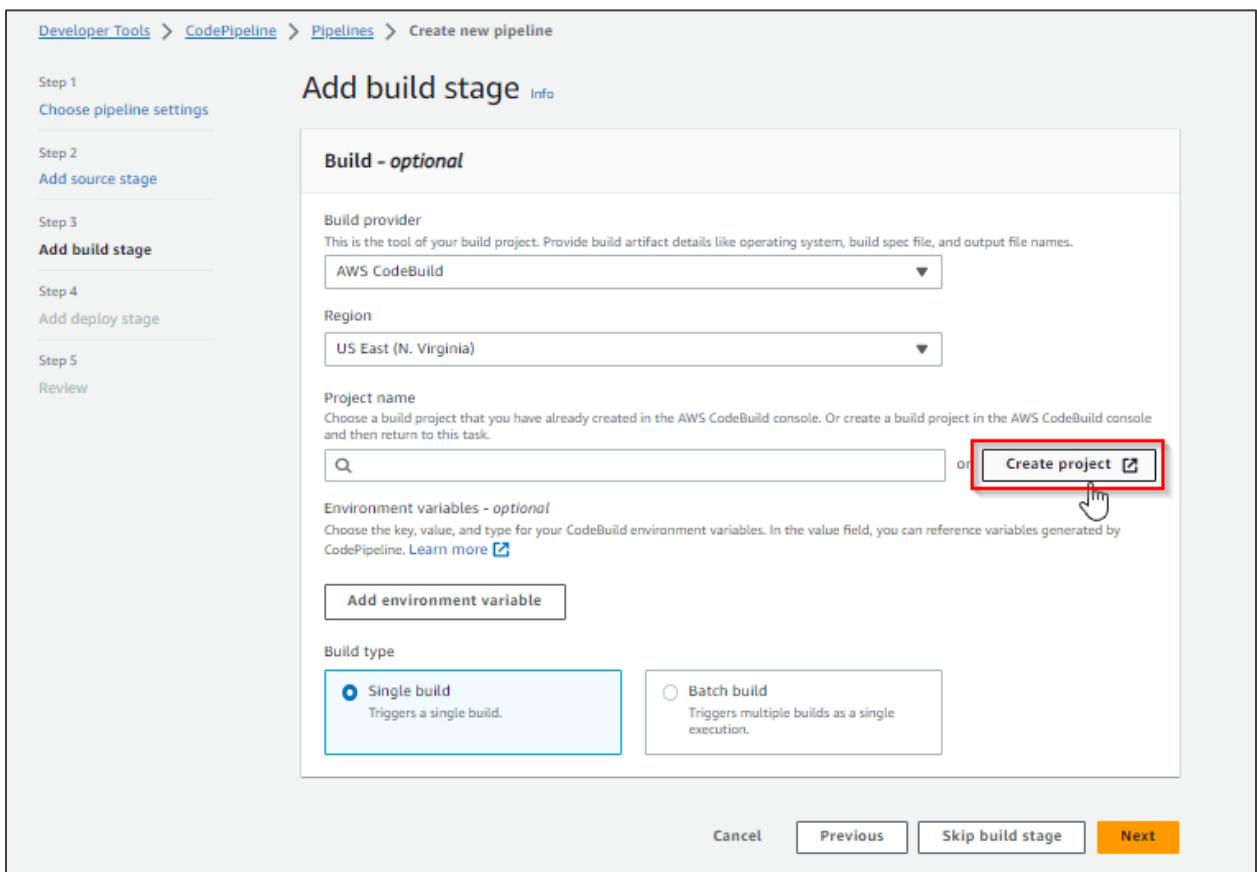
3.5 Enter the created repository name, select **master** under the branch section, and click on **Next**



3.6 Select AWS CodeBuild as the Build provider for building and automating tests



3.7 Then, click on the Create project button, and a new pop-up window will appear



3.8 In the new window, enter a name for the project

Create build project

Project configuration

Project name

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Description - optional

Enable concurrent build limit - optional

Limit the number of allowed concurrent builds for this project.

Restrict number of concurrent builds this project can start

▶ **Additional configuration**

tags

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

Environment

Environment image

Managed image
Use an image managed by AWS CodeBuild

Custom image
Specify a Docker image

Operating system

Ubuntu

Runtime(s)

Standard

Image

aws/codebuild/standard:7.0

Image version

Always use the latest image for this runtime version

Environment type

Linux EC2

Privileged

Enable this flag if you want to build Docker images or want your builds to get elevated privileges

Service role

New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

Role name

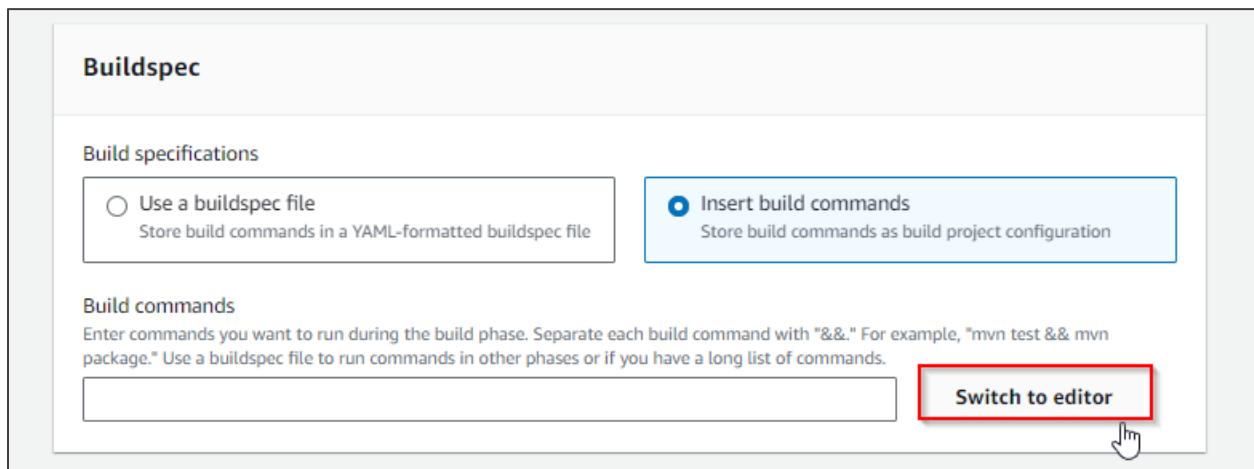
codebuild-ReactAppBuild-service-role

Type your service role name

▶ **Additional configuration**

Timeout, certificate, VPC, compute type, environment variables, file systems

- 3.10 In the **Buildspec** section, select the **Insert build commands** option and click on the **Switch to editor** button



- 3.11 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: .

Buildspec

Build specifications

Use a buildspec file
Store build commands in a YAML-formatted buildspec file

Insert build commands
Store build commands as build project configuration

Build commands

```
1 version: 0.2
2
3 ▼ phases:
4 ▼   build:
5 ▼     commands:
6       - "ls -alrt"
7       - "npm install"
8       - "npm run build"
9       - "npm run test"
10      - "npm run test:e2e"
11      - "zip -r dist.zip dist"
12 ▼ artifacts:
13 ▼   files:
14     - "dist.zip"
15
16   name: $(date +%Y-%m-%d)
17   discard-paths: yes
18   base-directory: .
19
```

YAML Ln 19, Col 1 ✘ Errors: 0 ⚠ Warnings: 0

3.12 Scroll down to the bottom of the page and click on the **Continue to CodePipeline** button

Batch configuration
You can run a group of builds as a single execution. Batch configuration is also available in advanced option when starting build.

Define batch configuration - *optional*
You can also define or override batch configuration when starting a build batch.

Logs

CloudWatch

CloudWatch logs - *optional*
Checking this option will upload build output logs to CloudWatch.

Group name

Stream name

S3

S3 logs - *optional*
Checking this option will upload build output logs to S3.

[Cancel](#) [Continue to CodePipeline](#)

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

[X](#) or [Create project](#) [!\[\]\(c6a6c4a832d644a43ecb21a17103874e_img.jpg\)](#)

[!\[\]\(7dc23964f7983046548514e9e3df4175_img.jpg\) Successfully created ReactAppBuild in CodeBuild. \[!\\[\\]\\(fe0fedaf9abd4a15c2c1a026a16ab198_img.jpg\\)\]\(#\)](#) [X](#)

The project is successfully created.

3.13 Now, click the **Next** button

Region
US East (N. Virginia)

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
ReactAppBuild or

Successfully created ReactAppBuild in CodeBuild.

Environment variables - *optional*
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Build type
 Single build
Triggers a single build. Batch build
Triggers multiple builds as a single execution.

3.14 In the deploy stage, select the **Amazon S3** option

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Add deploy stage Info

Deploy - optional

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

AWS CloudFormation Stack Set
AWS CodeDeploy
AWS Elastic Beanstalk
AWS OpsWorks Stacks
AWS S3PublishDev
AWS Service Catalog
Alexa Skills Kit
Amazon ECS
Amazon ECS (Blue/Green)
Amazon S3

skip deploy stage **Next**

3.15 Select the **codepipeline-deploy-bucket-simplilearn** option under Bucket

Add deploy stage Info

Deploy - optional

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

Amazon S3

Region

US East (N. Virginia)

Bucket

Q **codepipeline-deploy-bucket-simplilearn**

codepipeline-us-east-1-815502036299

Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

3.16 Enter dist.zip in the S3 object key section

Deploy - optional

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

Amazon S3

Region
US East (N. Virginia)

Bucket
codepipeline-deploy-bucket-simplilearn

S3 object key
dist.zip

Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

Extract file before deploy
The deployed artifact will be unzipped before deployment.

3.17 Click on the Next button

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

Amazon S3

Region
US East (N. Virginia)

Bucket
codepipeline-deploy-bucket-simplilearn

S3 object key
dist.zip

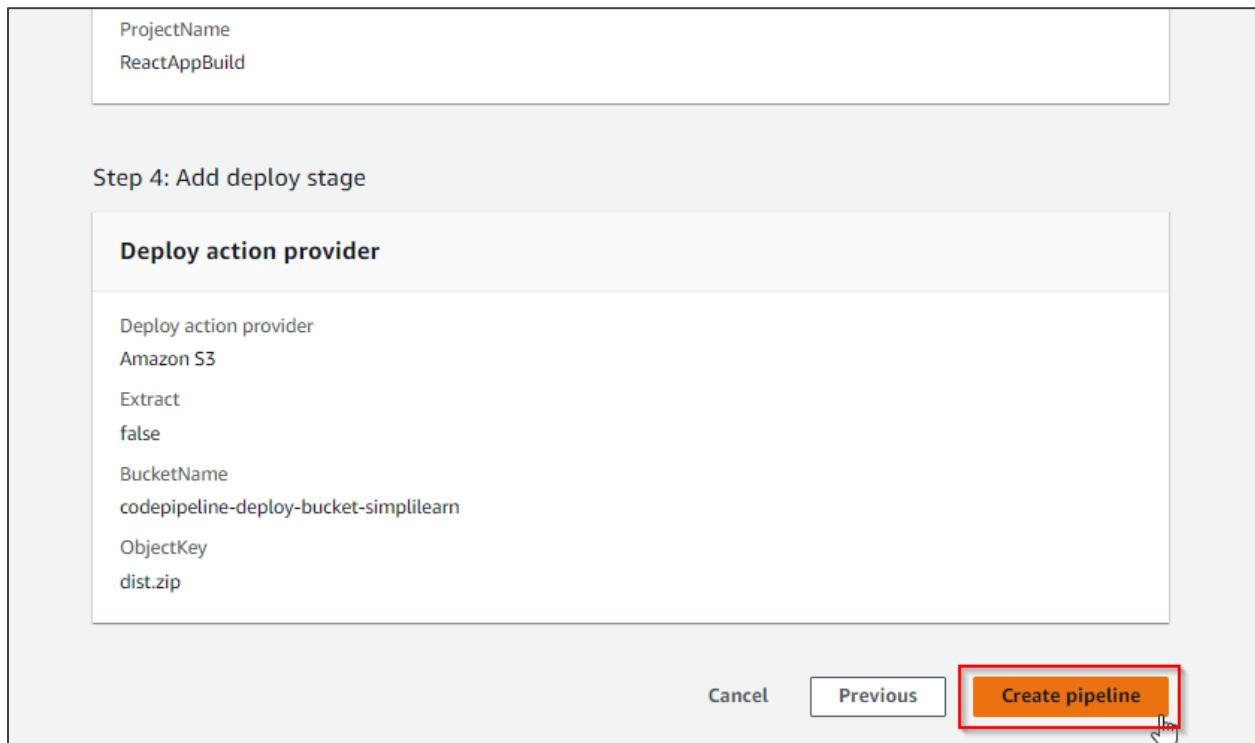
Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

Extract file before deploy
The deployed artifact will be unzipped before deployment.

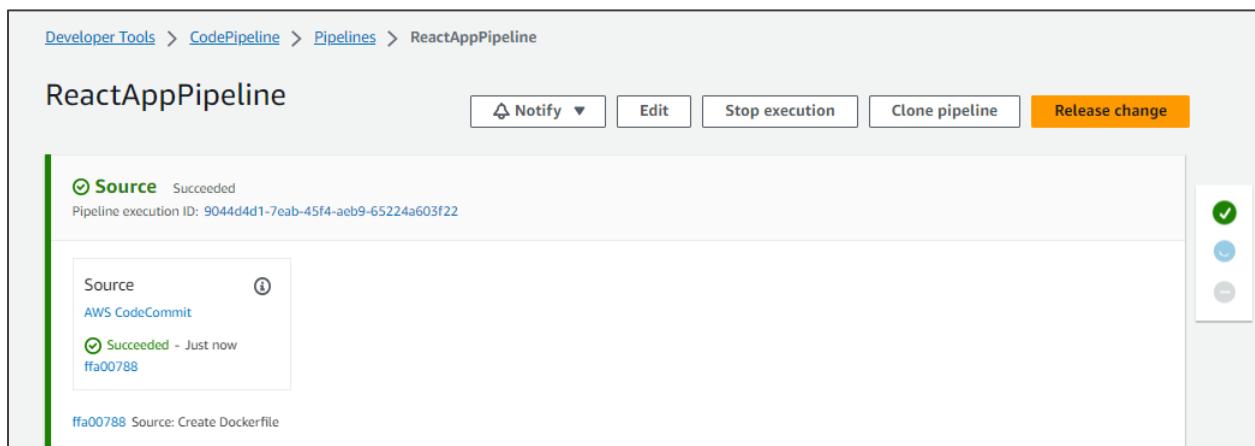
► Additional configuration

Cancel Previous Skip deploy stage **Next**

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

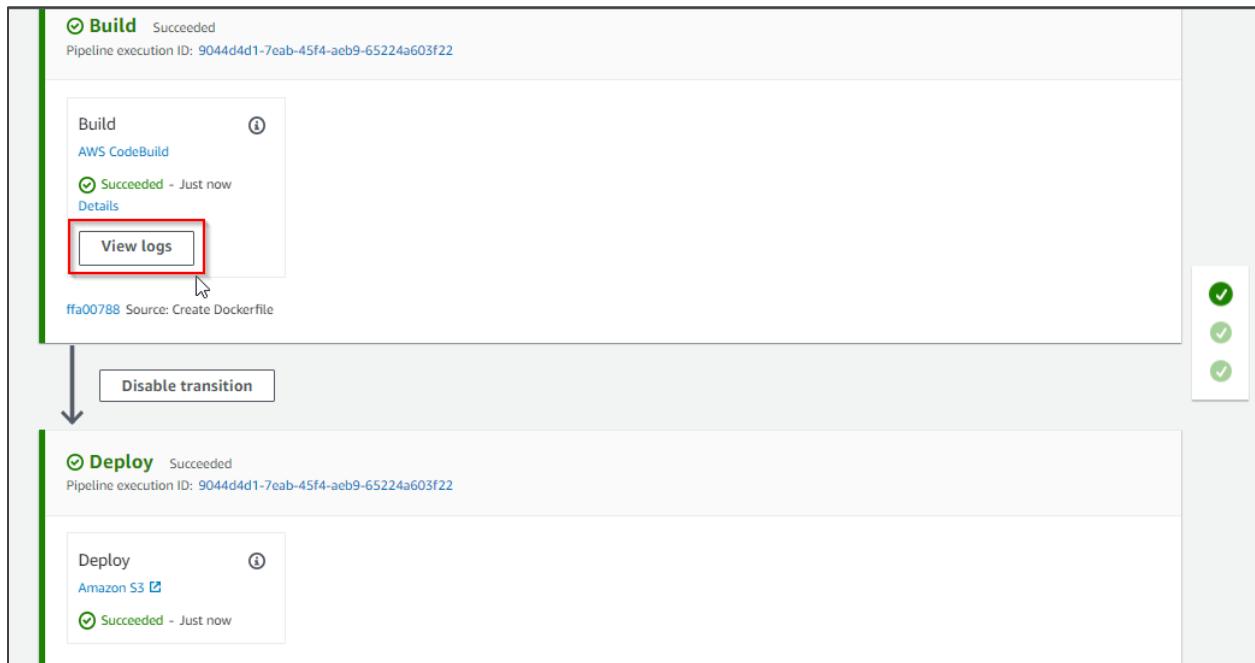


After creating the pipeline, the execution will start.





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



A modal window titled "Logs" is open, displaying the build logs. The logs show the process of building the application, including file additions and the completion of the build phase. The log entries are as follows:

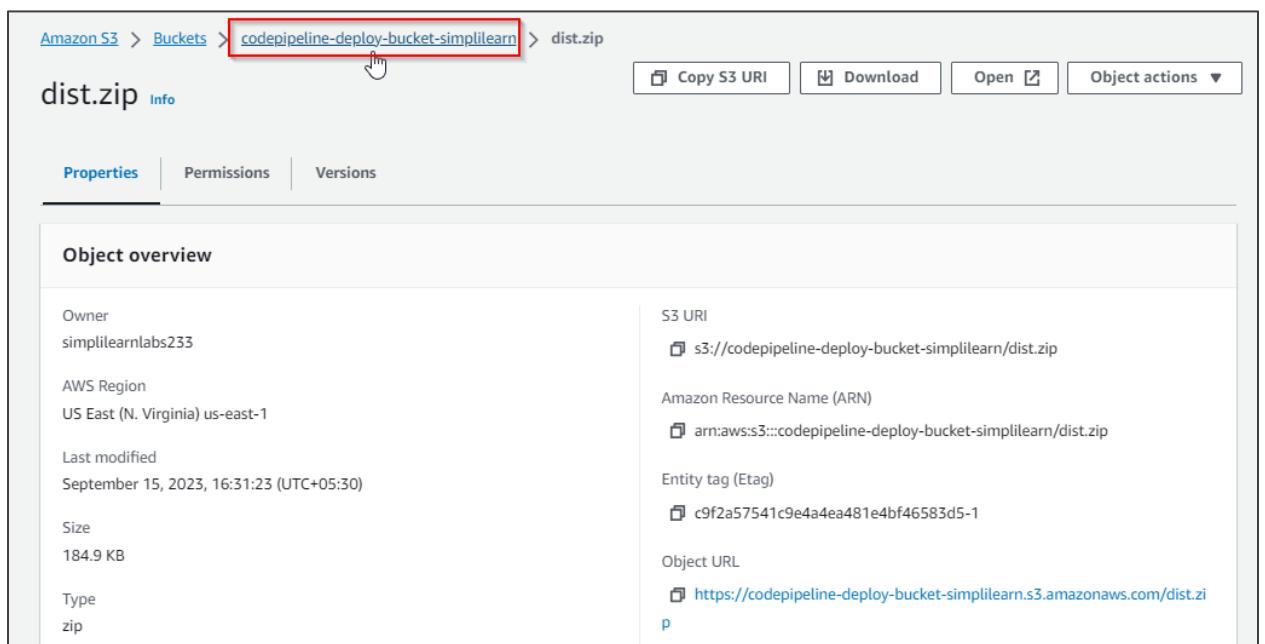
```
215 adding: dist/server.js.map (deflated 58%)
216 adding: dist/server.js (deflated 53%)
217 adding: dist/views/ (stored 0%)
218 adding: dist/views/_footer.html (deflated 53%)
219 adding: dist/views/_header.html (deflated 50%)
220 adding: dist/views/_search_form.html (deflated 57%)
221 adding: dist/views/_search_results.html (deflated 55%)
222 adding: dist/views/index.html (deflated 48%)
223
224 [Container] 2023/09/15 11:00:49 Phase complete: BUILD State: SUCCEEDED
225 [Container] 2023/09/15 11:00:49 Phase context status code: Message:
226 [Container] 2023/09/15 11:00:49 Entering phase POST_BUILD
227 [Container] 2023/09/15 11:00:49 Phase complete: POST_BUILD State: SUCCEEDED
228 [Container] 2023/09/15 11:00:49 Phase context status code: Message:
229 [Container] 2023/09/15 11:00:49 Expanding base directory path: .
230 [Container] 2023/09/15 11:00:49 Assembling file list
231 [Container] 2023/09/15 11:00:49 Expanding .
232 [Container] 2023/09/15 11:00:49 Expanding file paths for base directory .
233 [Container] 2023/09/15 11:00:49 Assembling file list
234 [Container] 2023/09/15 11:00:49 Expanding dist.zip
235 [Container] 2023/09/15 11:00:49 Found 1 file(s)
236 [Container] 2023/09/15 11:00:49 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED
237 [Container] 2023/09/15 11:00:49 Phase context status code: Message:
238
```

The "Close" button is visible at the bottom right of the log viewer.

3.20 Once the **Build** and **Deploy** stages are complete, validate whether the artifact was deployed to the S3 bucket by clicking on **Amazon S3**



3.21 Now, click on the bucket name **codepipeline-deploy-bucket-simplilearn**



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

The screenshot shows the Amazon S3 console interface. At the top, the path is shown as 'Amazon S3 > Buckets > codepipeline-deploy-bucket-simplilearn'. The bucket name 'codepipeline-deploy-bucket-simplilearn' is displayed with an 'Info' link. Below the bucket name, there are tabs for 'Objects' (which is selected), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' section title 'Objects (1)' is shown, followed by a description of objects in S3. A search bar 'Find objects by prefix' is present. Below the search bar is a table with one row, showing the details of the single object 'dist.zip'. The table columns are 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The object 'dist.zip' is a zip file, last modified on September 15, 2023, at 16:31:23 (UTC+05:30), with a size of 184.9 KB and a storage class of Standard.

Name	Type	Last modified	Size	Storage class
dist.zip	zip	September 15, 2023, 16:31:23 (UTC+05:30)	184.9 KB	Standard

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