

ADV DEVOPS CASE STUDY PRACTICAL

AIM:

Cloud Deployment with Automation

- Concepts Used: AWS CodePipeline, EC2, and S3.
- Problem Statement: "Build a simple web application using AWS CodeBuild and deploy it to an S3 bucket. Then, automate the deployment process using AWS CodePipeline, ensuring the application is deployed on an EC2 instance. Use a sample index.html page for demonstration."
- Tasks:
 - Set up AWS CodeBuild for the web app.
 - Create a pipeline that deploys to an S3 bucket.
 - Use AWS CodeDeploy to push updates to an EC2 instance.

THEORY:

1. Overview of AWS Services Involved

Before diving into the technical process, it is essential to understand the AWS services used in this experiment and how they interact with each other. The experiment will use the following AWS services:

- **Amazon EC2 (Elastic Compute Cloud):** EC2 is a cloud-based service that provides scalable compute capacity. It enables users to rent virtual machines (VMs) to host web applications and services. These virtual machines can be configured with various operating systems and software environments.
- **Amazon S3 (Simple Storage Service):** S3 is an object storage service that allows users to store and retrieve large amounts of data over the web. It can be used to store build artifacts, such as code, static files, and other resources required by the application.
- **AWS CodePipeline:** CodePipeline is a fully managed continuous integration and delivery service that helps automate the process of releasing applications. It builds, tests, and deploys code whenever there is a change in the source code repository (such as GitHub).
- **AWS CodeBuild:** CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software artifacts that are ready for deployment.
- **AWS CodeDeploy:** CodeDeploy automates the process of deploying applications to various compute platforms, such as EC2 instances, Lambda functions, or on-premises servers.

Each of these services plays a crucial role in the automation pipeline, and understanding their function and how they interact is key to building a successful deployment system.

2. Detailed Explanation of AWS CodePipeline

AWS CodePipeline acts as the backbone of this experiment, automating the movement of code from a version control repository (like GitHub or S3) through the stages of building, testing, and deployment. Each step in the pipeline is responsible for specific actions such as pulling the code from the repository, compiling it using CodeBuild, and deploying it using CodeDeploy.

Pipeline Stages:

1. **Source Stage:** The pipeline begins with the source stage, which pulls the latest version of the code from the specified repository (GitHub, Bitbucket, or S3). In this case, the source is a simple HTML file (`index.html`) that needs to be updated and deployed onto an EC2 instance.
2. **Build Stage:** This stage uses AWS CodeBuild to compile and package the application. For static websites or simple HTML files, this stage may not perform heavy compilation, but it will still produce an output artifact (such as an updated `index.html` file) that will be deployed to an EC2 instance.
3. **Deploy Stage:** In the final stage, AWS CodeDeploy pushes the build artifacts (e.g., `index.html`) to the EC2 instance. During this stage, the application is installed, and the EC2 instance is updated to reflect the latest changes in the code.

PROCEDURE & SCREENSHOTS:

Create a Simple Web App

1. First, create a simple web app with an `index.html` file:

A screenshot of a Visual Studio Code editor window. The top pane shows the 'index.html' file with the following content:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Simple AWS web app for case study</title>
5 </head>
6 <body>
7   <h1>Welcome to my Adv devops case study!</h1>
8 </body>
9 </html>
10
```

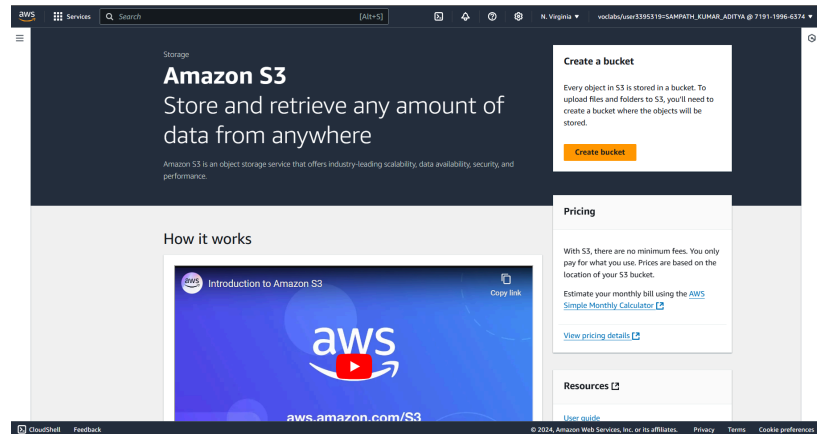
The bottom pane shows the 'TERMINAL' tab with a PowerShell prompt: `PS D:\Aditya_Work\VS_Code\IP all exps - sem 5\Case study>`. The terminal also displays a table with file metadata:

This file will serve as the web page deployed to your S3 bucket and later to the EC2 instance.

2. Set Up S3 Bucket for Web App Hosting

1. Go to the AWS S3 Console:

- Open the [S3 console](#).



- Create a new S3 bucket, giving it a unique name (e.g., **my-s3-web-bucket**).

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type [Info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

my-s3-web-app-bucket

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

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

Choose bucket

Format: s3://bucket/prefix

- Under **Permissions**, uncheck the "Block all public access" option, allowing public access for web hosting.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐

Block public access to buckets and objects granted through *new* access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐


Block public access to buckets and objects granted through *any* access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.

☐

Block public access to buckets and objects granted through *new* public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐

Block public and cross-account access to buckets and objects through *any* public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming


2. Configure the Bucket for Website Hosting:

- Go to the **Properties** tab of your S3 bucket.

my-s3-web-app-bucket [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN)  arnaws:s3:::my-s3-web-app-bucket	Creation date October 21, 2024, 16:52:43 (UTC+05:30)
-----------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------

Bucket Versioning [Edit](#)

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning
Disabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)
Disabled

- Scroll down to **Static website hosting**.

Static website hosting [Edit](#)

Use this bucket to host a website or redirect requests. [Learn more](#)

S3 static website hosting
Disabled

- Enable it, and set the **Index document** as `index.html`.

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable

☒ Enable

Hosting type

☒ Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

☐ Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

Index document
Specify the home or default page of the website.

`index.html`

Error document - optional
This is returned when an error occurs.

`error.html`

- Copy the bucket website URL for testing the web app later.

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#) Edit

S3 static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

`http://my-s3-web-app-bucket.s3-website-us-east-1.amazonaws.com`

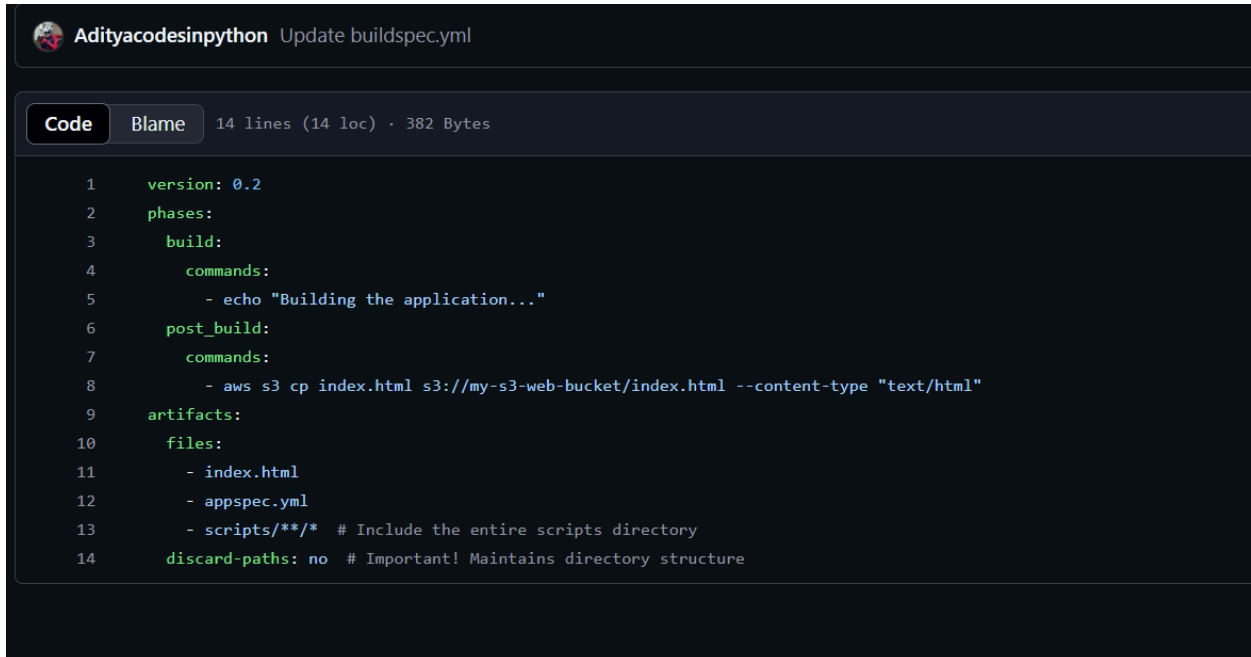
in my case it is:

<http://my-s3-web-bucket.s3-website-ap-southeast-2.amazonaws.com>

- if s3 website shows 403 forbidden, its a IAM permission issue.

3. Set Up CodeBuild for Your Web App

Create a Buildspec File: In your project directory (where `index.html` resides), create a `buildspec.yml` file. This file tells AWS CodeBuild what to do during the build.

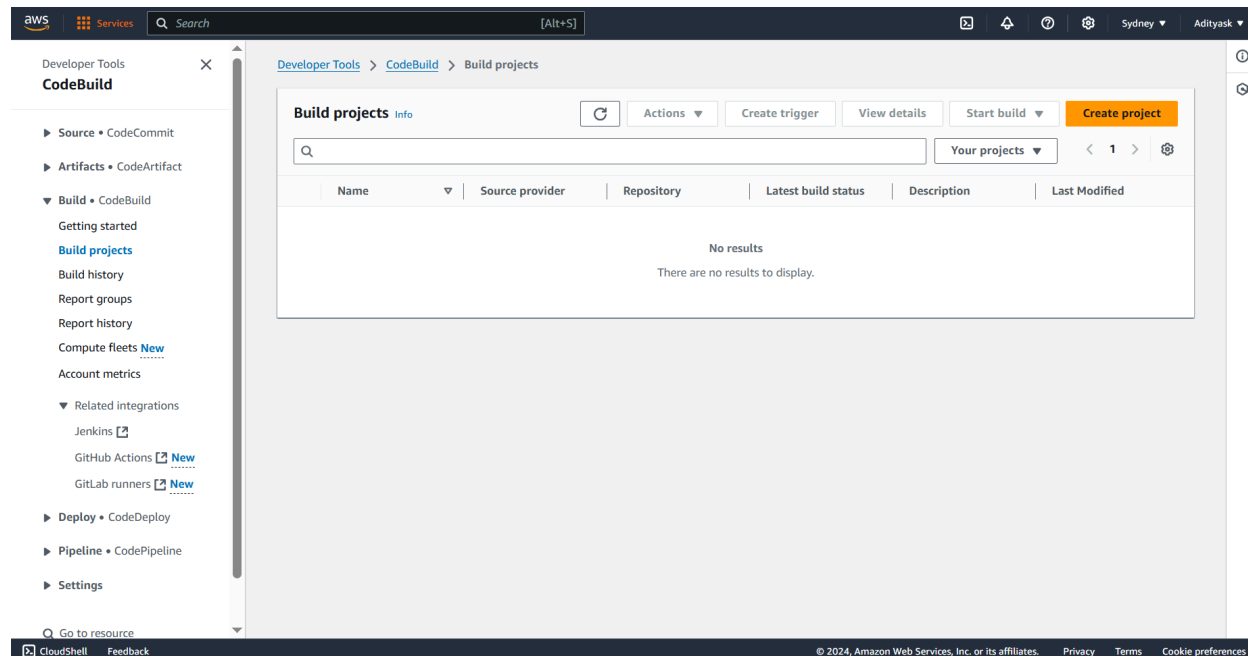


The screenshot shows a GitHub repository for 'Adityacodesinpython' with the file 'Update buildspec.yml' selected. The file contains a YAML configuration for an AWS CodeBuild buildspec. The configuration includes a version of 0.2, a build phase with a command to echo 'Building the application...', a post_build phase with a command to upload the index.html file to an S3 bucket, and artifacts that include index.html, appspec.yml, and the scripts directory. The discard-paths are set to no.

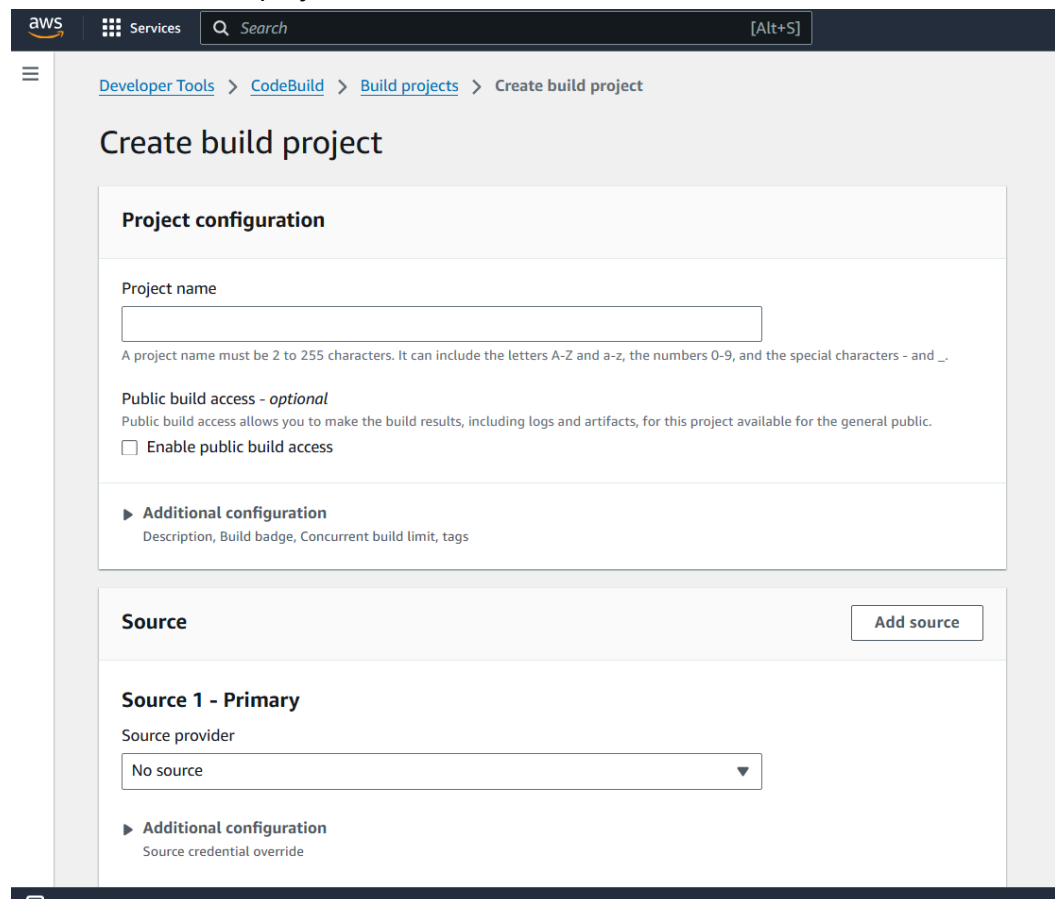
```
1  version: 0.2
2  phases:
3    build:
4      commands:
5        - echo "Building the application..."
6    post_build:
7      commands:
8        - aws s3 cp index.html s3://my-s3-web-bucket/index.html --content-type "text/html"
9  artifacts:
10   files:
11     - index.html
12     - appspec.yml
13     - scripts/**/* # Include the entire scripts directory
14   discard-paths: no # Important! Maintains directory structure
```

Go to AWS CodeBuild:

- Open the [AWS CodeBuild console](#).

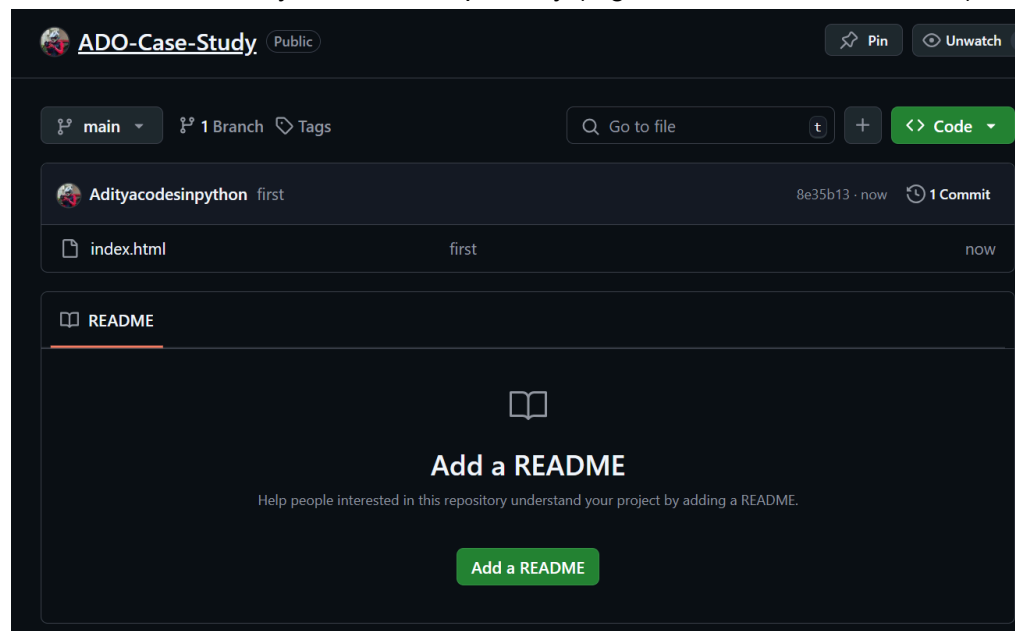


- Create a new build project.



The screenshot shows the AWS CodeBuild 'Create build project' page. The breadcrumb navigation is 'Developer Tools > CodeBuild > Build projects > Create build project'. The main heading is 'Create build project'. Below this is the 'Project configuration' section, which includes a 'Project name' input field with a note: '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 _.' There is also a checkbox for 'Enable public build access' under the heading 'Public build access - optional'. Below this is an 'Additional configuration' section with a link to 'Description, Build badge, Concurrent build limit, tags'. The 'Source' section is below, with an 'Add source' button. Under 'Source 1 - Primary', there is a 'Source provider' dropdown menu currently set to 'No source', and an 'Additional configuration' link for 'Source credential override'.

- For **Source**, choose your source repository (e.g., GitHub, Bitbucket, or S3).



Source 1 - Primary

Source provider
 GitHub

Credential
☐ Default source credential
 Use your account's default source credential to apply to all projects

☒ Custom source credential
 Use a custom source credential to override your account's default settings

Credential type
☒ GitHub App
 Connect project to GitHub using an AWS managed GitHub App

☐ OAuth app
 Connect project to GitHub using an OAuth app

☐ Personal access token
 Connect project to GitHub using a personal access token

Connection
 You can [create a new GitHub connection](#) by using an AWS managed GitHub App

arn:aws:codeconnections:ap-southeast-2:010928205803:conne

Repository
☒ Repository in my GitHub account

☐ Public repository

☐ GitHub scoped webhook

Repository
 https://github.com/Adityacodesinpython/ADO-Case-Study

Source version - *optional* [Info](#)
 Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

Additional configuration

- Under **Environment**, choose **Managed Image** with **Ubuntu**, and select **Runtime: Standard**.

Operating system
 Ubuntu

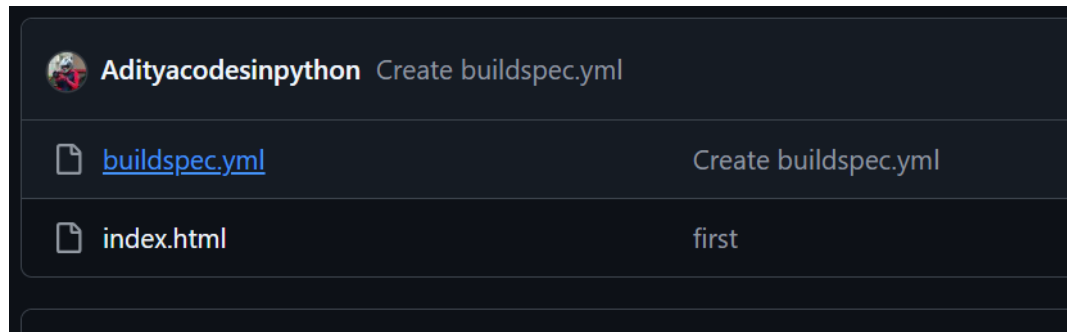
Runtime(s)
 Standard

Image
 aws/codebuild/standard:5.0

Image version
 Always use the latest image for this runtime version

☐ Use GPU-enhanced compute

- specify the `buildspec.yml` file you created. before this add it to your git repo



Buildspec

Build specifications

☐ Insert build commands
Store build commands as build project configuration

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

Buildspec name - optional
By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

- Set **Artifacts** to "S3", and choose the bucket you created earlier.

Artifacts

Add artifact

Artifact 1 - Primary

Type

Amazon S3

You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.

Bucket name

Q my-s3-web-bucket X

Name

The name of the folder or compressed file in the bucket that will contain your output artifacts. Use Artifacts packaging under Additional configuration to choose whether to use a folder or compressed file. If the name is not provided, defaults to project name.

☐ Enable semantic versioning

Use the artifact name specified in the buildspec file

Path - optional

The path to the build output ZIP file or folder.

Example: MyPath/MyArtifact.zip.

Namespace type - optional

- Create the build project and start the build to ensure it uploads `index.html` to the S3 bucket.

Project created

You have successfully created the following project: case-study-code-build

Create a notification rule for this project

Developer Tools > CodeBuild > Build projects > case-study-code-build

case-study-code-build

Actions Create trigger Edit Clone Debug build Start build with overrides Start build

Configuration

Source provider GitHub	Primary repository Adityacodesinpython/ADO-Case-Study	Artifacts upload location my-s3-web-bucket	Service role arn:aws:iam::010928205803:role/service-role/codebuild-case-study-code-build-service-role
Public builds Disabled			

Build history

Batch history Project details Build triggers Metrics

Build history

Stop build View artifacts View logs Delete builds Retry build

< 1 >

Build run	Status	Build number	Source version	Submitter	Duration	Completed
-----------	--------	--------------	----------------	-----------	----------	-----------

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

case-study-code-build:7345b99d-7bb3-4e94-82ad-09b8e0cd8f4e

Build status

Status

In progress

Initiator

root

Start time

Oct 21, 2024 5:37 PM (UTC+5:30)

End time

-

Build logs

Phase details

Reports

Environment variables

Build details

case-study-code-build:7345b99d-7bb3-4e94-82ad-09b8e0cd8f4e

Stop build

Retry build

Build status

Status

Succeeded

Initiator

root

Build ARN

arn:aws:codebuild:ap-southeast-2:010928205803:build/case-study-code-build:7345b99d-7bb3-4e94-82ad-09b8e0cd8f4e

Resolved source version

8e35b133cbff173042ca2a2e1fcd25693dd3f12f

Start time

Oct 21, 2024 5:37 PM (UTC+5:30)

End time

Oct 21, 2024 5:37 PM (UTC+5:30)

Build number

2

Build logs

Phase details

Reports

Environment variables

Build details

Resource utilization

Showing the last 51 lines of the build log. [View entire log](#)

Tail logs

No previous logs

1 [Container] 2024/10/21 12:07:28.161414 Running on CodeBuild On-demand

2 [Container] 2024/10/21 12:07:28.161430 Waiting for agent ping

3 [Container] 2024/10/21 12:07:28.663932 Waiting for DOWNLOAD_SOURCE

4 [Container] 2024/10/21 12:07:31.481942 Phase is DOWNLOAD_SOURCE

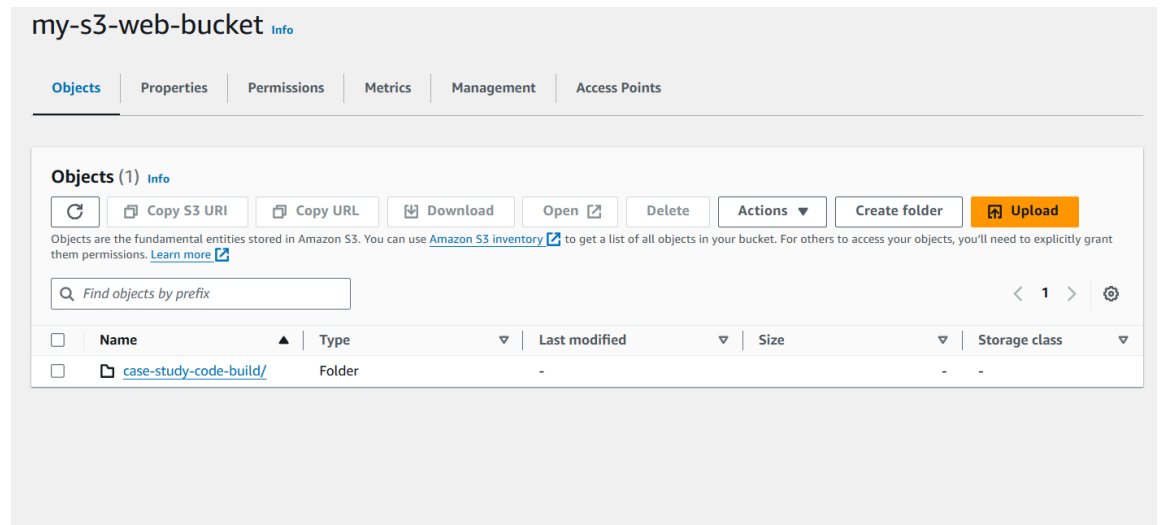
© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

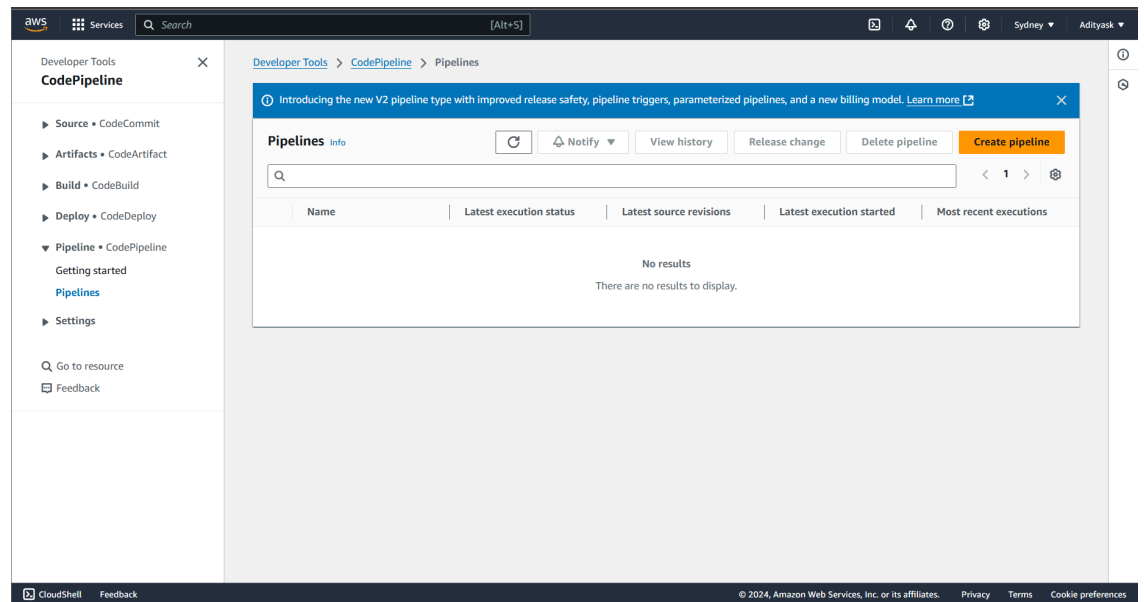
Cookie preferences

s3 bucket was updated:



4. Set Up AWS CodePipeline

1. **Go to AWS CodePipeline:**
 - Open the [CodePipeline console](#).



- Create a new pipeline.

The screenshot shows the 'Choose creation option' dialog, which is Step 1 of 4. It has a title 'Choose creation option' with an 'Info' link. Below the title is a section 'Creation options' with the instruction 'Choose one of the following options to create your pipeline.' There are two radio button options: 'Create pipeline from template' (selected) with the description 'Create a pipeline from a pre-built template for common scenarios.', and 'Build custom pipeline' with the description 'Build a pipeline from scratch to meet your specific needs.' At the bottom right are 'Cancel' and 'Next' buttons.

- For **Source**:
 - Select your repository (e.g., GitHub, Bitbucket).

The screenshot shows the 'Source' configuration page. It has a title 'Source'. Under 'Source provider', there is a dropdown menu set to 'GitHub (Version 2)'. Below this is a blue box with an information icon and text: 'New GitHub version 2 (app-based) action. To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)'. Under 'Connection', there is a text input with 'arn:aws:codeconnections:ap-southeast-2:010928205803:connection/b1923c...' and a 'Connect to GitHub' button. Under 'Repository name', there is a text input with 'Adityacodesinpython/ADO-Case-Study'. Under 'Default branch', there is a text input with 'main' and a dropdown menu showing 'main'. At the bottom, there are two radio button options for 'Choose the output artifact format': 'CodePipeline default' (selected) and 'Full clone'.

- Connect and choose the appropriate branch where the `index.html` and `buildspec.yml` files are.

The screenshot shows the 'Default branch' section with a search bar containing 'main' and a dropdown menu showing 'main'. Below this is the 'Choose the output artifact format' section with two radio button options: 'CodePipeline default' (selected) and 'Full clone'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

main

Choose the output artifact format.

☒ CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

☐ Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only

2. Add Build Stage:

- In the **Build stage**, choose **AWS CodeBuild** as the build provider.

The screenshot shows the 'Build - optional' section. Under 'Build provider', the 'Other build providers' radio button is selected. Below this, a dropdown menu shows 'AWS CodeBuild'.

Build - optional

Build provider
Choose the tool you want to use to run build commands and specify artifacts for your build action.

☐ Commands ☒ Other build providers

AWS CodeBuild

- Select the CodeBuild project you created earlier.

The screenshot shows the 'Project name' section with a search bar containing 'case-study-code-build' and a 'Create project' button. Below this is the 'Environment variables - optional' section with an 'Add environment variable' button. At the bottom is the 'Build type' section with two radio button options: 'Single build' (selected) and 'Batch build'.

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.

case-study-code-build or Create project

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](#)

Add environment variable

Build type

☒ Single build
Triggers a single build.

☐ Batch build
Triggers multiple builds as a single execution.

3. Deploy to S3:

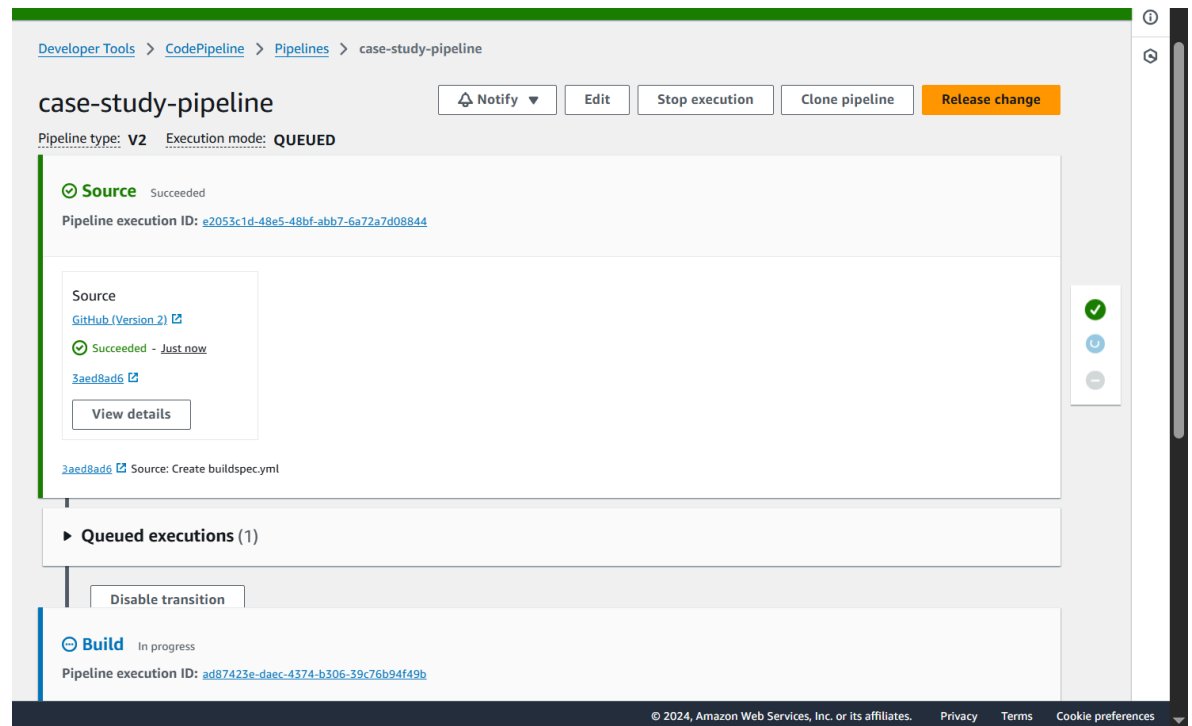
- In the next stage, choose **Deploy**. Select **Amazon S3**. Choose your S3 bucket (**my-s3-bucket**) where the **index.html** file will be deployed.

The screenshot displays the AWS CodePipeline console interface for configuring a new stage. On the left, a sidebar lists the steps: Step 2 (Choose pipeline settings), Step 3 (Add source stage), Step 4 (Add build stage), Step 5 (Add deploy stage), Step 6, and Review. The main panel is titled 'Deploy - optional' and contains the following configuration fields:

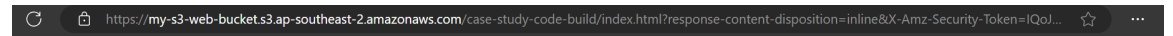
- Deploy provider:** A dropdown menu set to 'Amazon S3'.
- Region:** A dropdown menu set to 'Asia Pacific (Sydney)'.
- Input artifacts:** A section with a 'Choose an input artifact for this action. [Learn more](#)' link and an empty dropdown menu.
- BuildArtifact:** A button labeled 'BuildArtifact' with a close icon, indicating it is 'Defined by: Build'.
- Bucket:** A search input field containing 'my-s3-web-bucket' and a close icon.
- S3 object key:** An input field containing '/case-study-code-build/index.html'.
- Extract file before deploy:** An unchecked checkbox with the text 'The deployed artifact will be unzipped before deployment.'
- Additional configuration:** A section with a checked checkbox labeled 'Configure automatic rollback on stage failure'.

4. Test the Pipeline:

- Once the pipeline is set up, click **Release Change** to start the pipeline. This should fetch the latest code, build it, and upload `index.html` to the S3 bucket.



- Visit the S3 bucket's website URL to verify that the `index.html` page is live.

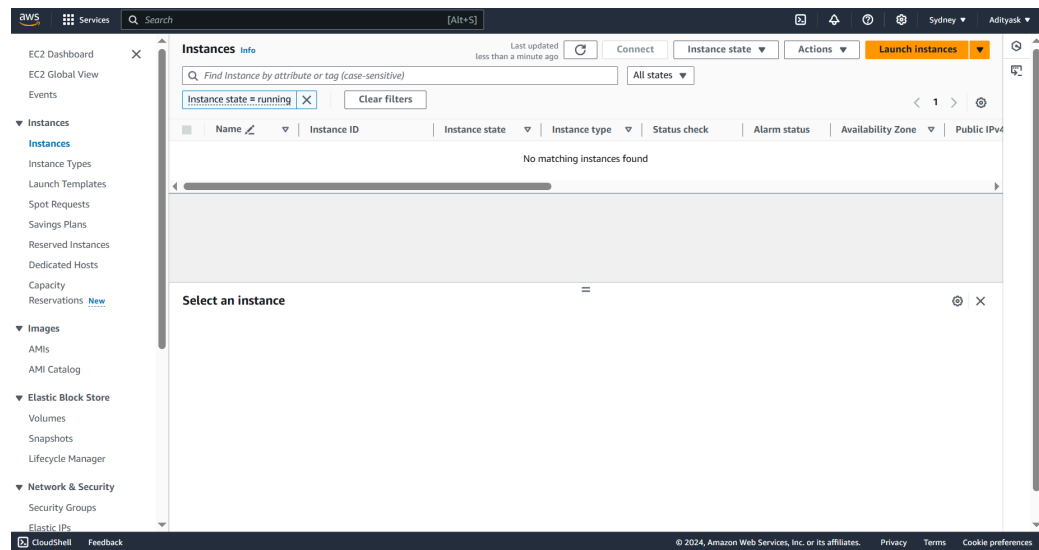


Welcome to my Adv devops case study!

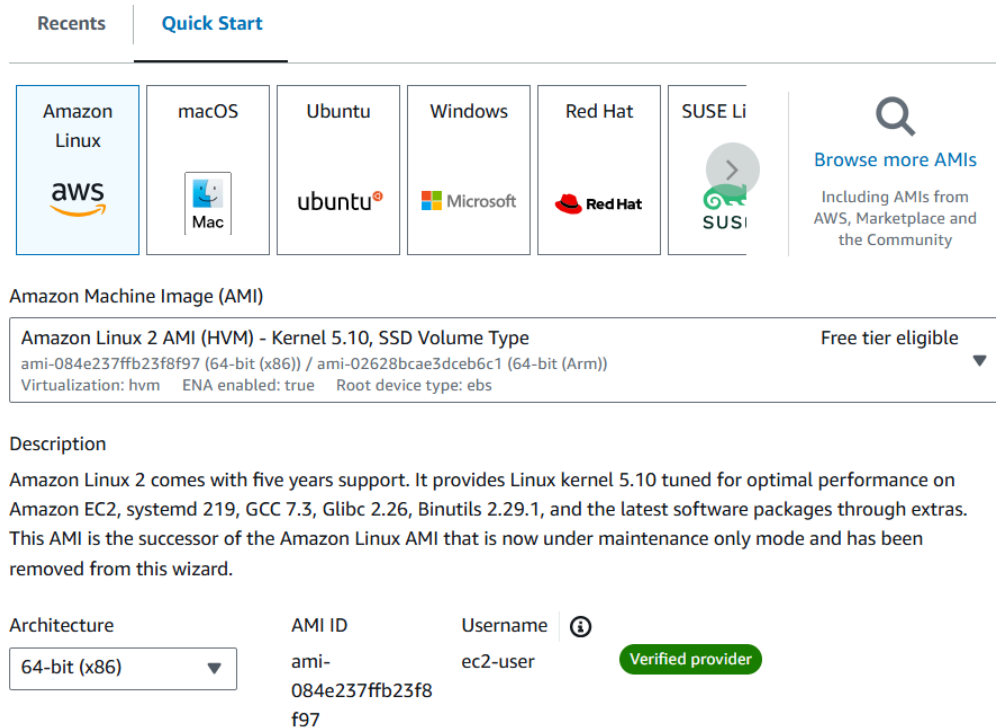
5. Set Up EC2 Instance for Web Hosting

1. Launch an EC2 Instance:

- Open the [EC2 console](#).



- Launch a new instance, selecting an Amazon Linux 2 AMI.



- Choose the default t2.micro instance type.
- Configure instance settings and storage (use defaults for now).

- In **Configure Security Group**, allow HTTP traffic by adding a rule to open port 80.

The screenshot displays the 'Configure Security Group' interface in the AWS Management Console. At the top, there's a navigation bar with 'Services', a search bar, and a keyboard shortcut '[Alt+S]'. The main section is titled 'Network settings' with an 'Info' link and an 'Edit' button. Under 'Network', it shows the VPC ID 'vpc-030c6ebc1841ce4bb' and the Subnet ID 'No preference (Default subnet in any availability zone)'. The 'Auto-assign public IP' option is set to 'Enable'. Below this, it mentions 'Additional charges apply when outside of free tier allowance'. The 'Firewall (security groups)' section includes an 'Info' link and a description: 'A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.' There are two radio buttons: 'Create security group' (selected) and 'Select existing security group'. A message states: 'We'll create a new security group called 'launch-wizard-1' with the following rules:'. Three rules are listed: 'Allow SSH traffic from' (checked, with a dropdown set to 'Anywhere' and IP '0.0.0.0/0'), 'Allow HTTPS traffic from the internet' (unchecked), and 'Allow HTTP traffic from the internet' (checked). A yellow warning box at the bottom says: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

2. Connect to the EC2 Instance:

- Once the instance is running, connect via SSH.

Install the required web server (Apache) on your instance:

```
sudo yum update -y
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
```

```

Running transaction
Installing : apr-1.7.2-1.amzn2.x86_64 1/9
Installing : apr-util-1.6.3-1.amzn2.0.1.x86_64 2/9
Installing : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64 3/9
Installing : httpd-tools-2.4.62-1.amzn2.0.2.x86_64 4/9
Installing : httpd filesystem-2.4.62-1.amzn2.0.2.noarch 5/9
Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch 6/9
Installing : mailcap-2.1.41-2.amzn2.noarch 7/9
Installing : mod_http2-1.15.19-1.amzn2.0.2.x86_64 8/9
Installing : httpd-2.4.62-1.amzn2.0.2.x86_64 9/9
Verifying : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64 1/9
Verifying : httpd-2.4.62-1.amzn2.0.2.x86_64 2/9
Verifying : apr-1.7.2-1.amzn2.x86_64 3/9
Verifying : mod_http2-1.15.19-1.amzn2.0.2.x86_64 4/9
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64 5/9
Verifying : mailcap-2.1.41-2.amzn2.noarch 6/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 7/9
Verifying : httpd-tools-2.4.62-1.amzn2.0.2.x86_64 8/9
Verifying : httpd filesystem-2.4.62-1.amzn2.0.2.noarch 9/9

Installed:
  httpd.x86_64 0:2.4.62-1.amzn2.0.2

Dependency Installed:
  apr.x86_64 0:1.7.2-1.amzn2          apr-util.x86_64 0:1.6.3-1.amzn2.0.1  apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
  generic-logos-httpd.noarch 0:18.0.0-4.amzn2  httpd filesystem.noarch 0:2.4.62-1.amzn2.0.2  httpd-tools.x86_64 0:2.4.62-1.amzn2.0.2
  mailcap.noarch 0:2.1.41-2.amzn2          mod_http2.x86_64 0:1.15.19-1.amzn2.0.2

Complete!
[ec2-user@ip-172-31-15-234 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-15-234 ~]$ sudo systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-15-234 ~]$

```

i-Ofa23c71f1e0e0980 (case-study)

PublicIPs: 13.210.200.175 PrivateIPs: 172.31.15.234

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

6. Set Up AWS CodeDeploy to Push Updates to EC2

1. Install CodeDeploy Agent on EC2:

- Connect to the EC2 instance and Install the CodeDeploy agent:

```

sudo yum update -y
sudo yum install -y ruby wget
wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
chmod +x ./install

```

`sudo ./install auto`

```

AWS CloudShell [Alt+S] Sydney Aditya
Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together. Close permanently

Installing:
codedeploy-agent          noarch          1.7.0-92          /codedeploy-agent-1.7.0-92.noarch.tmp-20241022-3938-lazqggd 13 M

Transaction Summary
-----
Install 1 Package

Total size: 13 M
Installed size: 13 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction

pre hook : 1
Checking if there is already a process named codedeploy-agent running.
Installing : codedeploy-agent-1.7.0-92.noarch 1/1

post hook : 1
Check if there is a codedeployagent config file.
Start codedeploy-agent in post hook if this is a first install.
Verifying : codedeploy-agent-1.7.0-92.noarch 1/1

Installed:
codedeploy-agent.noarch 0:1.7.0-92

Complete!
[2024-10-22T10:55:57.547877 #3938] INFO -- : Update check complete.
[2024-10-22T10:55:57.547928 #3938] INFO -- : Stopping updater.
[ec2-user@ip-172-31-15-234 ~]$
[ec2-user@ip-172-31-15-234 ~]$
[ec2-user@ip-172-31-15-234 ~]$

i-Ofa23c71f1e0e0980 (case-study)
PublicIPs: 13.210.200.175 PrivateIPs: 172.31.15.234

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```

2. Set Up CodeDeploy Application:

3.1 Create Application

1. Using AWS CLI from Local Terminal on pc (not ec2, use your own pc's command line):

```
# First ensure AWS CLI is installed
aws --version
```

if not installed, install from <https://awscli.amazonaws.com/AWSCLIV2.msi>

```
aws configure
```

2. Create application:

```
aws deploy create-application --application-name my-webapp
```

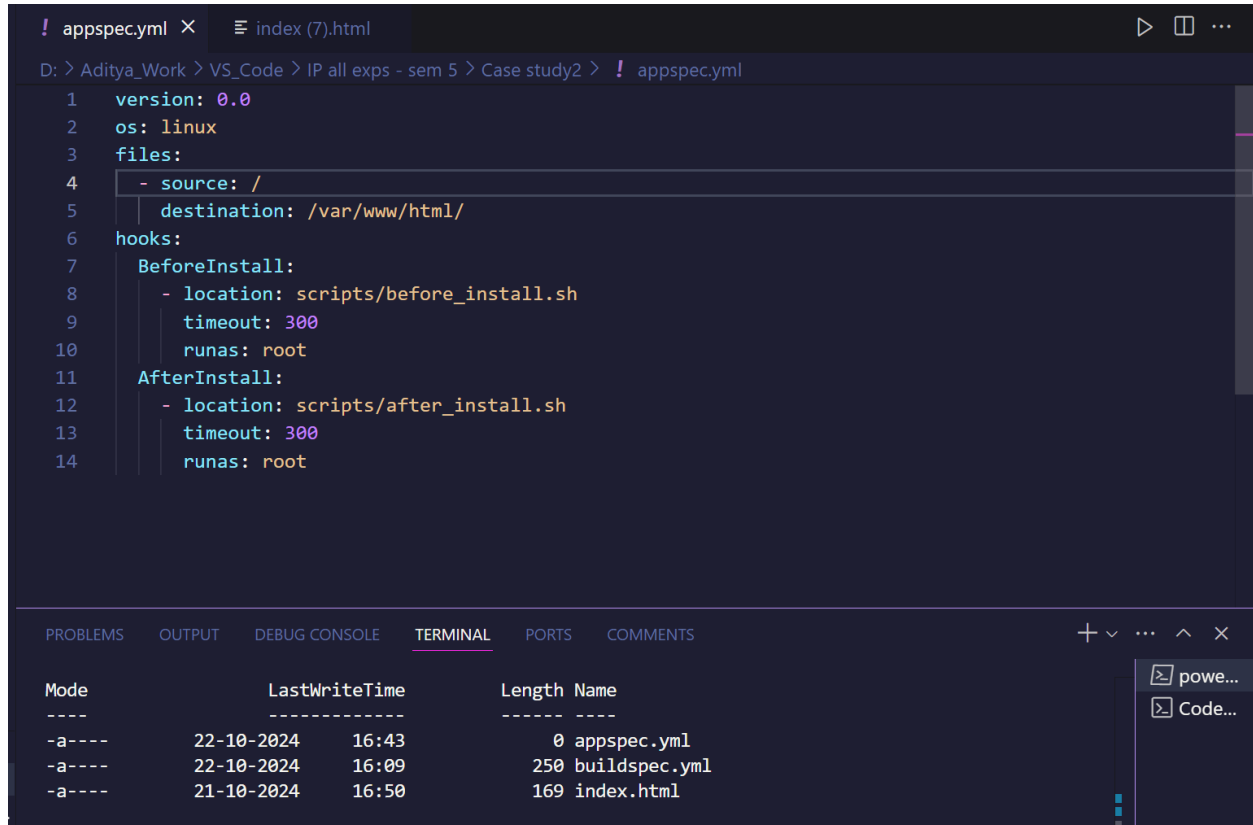
3.2 Create Deployment Group

- Create deployment group:

```
aws deploy create-deployment-group --application-name
my-webapp --deployment-group-name my-webapp-group
```

- ```
--service-role-arn
arn:aws:iam::ACCOUNT_ID:role/CodeDeployServiceRole
```
- enter your ACCOUNT\_ID above!!

**Create `appspec.yml` for CodeDeploy:** In your project folder, create an `appspec.yml` file to specify how CodeDeploy should handle the deployment:

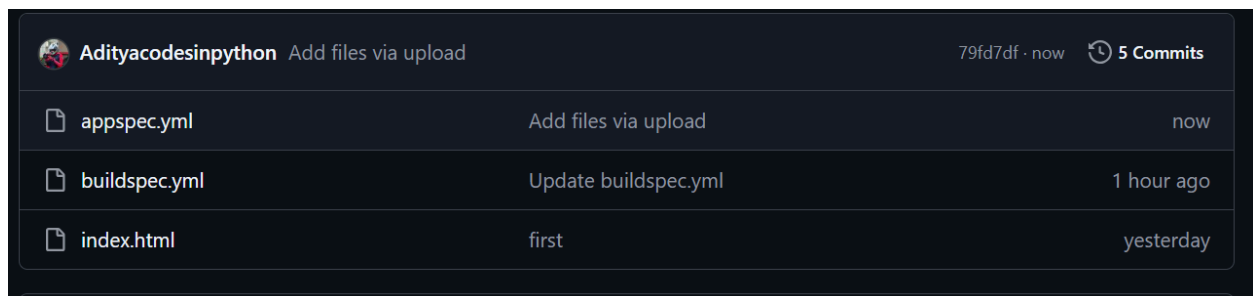


The screenshot shows the VS Code editor with the `appspec.yml` file open. The file content is as follows:

```
1 version: 0.0
2 os: linux
3 files:
4 - source: /
5 destination: /var/www/html/
6 hooks:
7 BeforeInstall:
8 - location: scripts/before_install.sh
9 timeout: 300
10 runas: root
11 AfterInstall:
12 - location: scripts/after_install.sh
13 timeout: 300
14 runas: root
```

The Explorer sidebar on the right shows the file structure:


| Mode   | LastWriteTime    | Length | Name          |
|--------|------------------|--------|---------------|
| -a---- | 22-10-2024 16:43 | 0      | appspec.yml   |
| -a---- | 22-10-2024 16:09 | 250    | buildspec.yml |
| -a---- | 21-10-2024 16:50 | 169    | index.html    |



The screenshot shows the GitHub repository interface for the `Adityacodesinpython` repository. The commit history for the `appspec.yml` file is displayed:

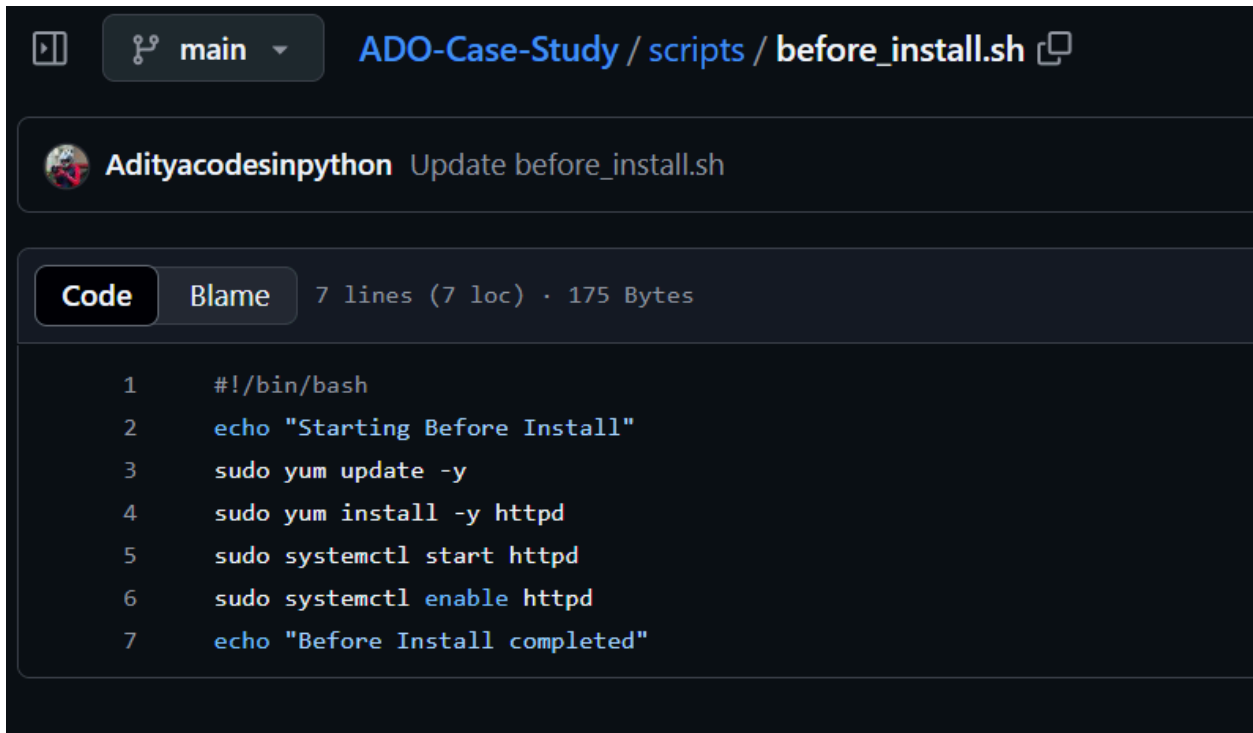
| File                       | Commit Message       | Time       |
|----------------------------|----------------------|------------|
| <code>appspec.yml</code>   | Add files via upload | now        |
| <code>buildspec.yml</code> | Update buildspec.yml | 1 hour ago |
| <code>index.html</code>    | first                | yesterday  |

Now create a scripts folder inside your repo which will contain 2 files: `before_install.sh` and `after_install.sh`

|                                                                                                       |                         |                        |              |
|-------------------------------------------------------------------------------------------------------|-------------------------|------------------------|--------------|
|  Adityacodesinpython | Update index.html       | c1ca273 · 18 hours ago | 🕒 13 Commits |
| 📁 scripts                                                                                             | Update after_install.sh | 19 hours ago           |              |
| 📄 appspec.yml                                                                                         | sss                     | yesterday              |              |
| 📄 buildspec.yml                                                                                       | Update buildspec.yml    | 19 hours ago           |              |
| 📄 index.html                                                                                          | Update index.html       | 18 hours ago           |              |
| 📖 README                                                                                              |                         |                        |              |

|                                 |                          |                         |
|---------------------------------|--------------------------|-------------------------|
| main ADO-Case-Study / scripts / |                          | Go to file              |
| Adityacodesinpython             |                          | Update after_install.sh |
| Name                            | Last commit message      |                         |
| 📁 ..                            |                          |                         |
| 📄 after_install.sh              | Update after_install.sh  |                         |
| 📄 before_install.sh             | Update before_install.sh |                         |

|                                                                                                                                                                                                                                                                                                        |       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| main ADO-Case-Study / scripts / after_install.sh                                                                                                                                                                                                                                                       |       |
| Adityacodesinpython                                                                                                                                                                                                                                                                                    |       |
| Update after_install.sh                                                                                                                                                                                                                                                                                |       |
| Code                                                                                                                                                                                                                                                                                                   | Blame |
| 7 lines (7 loc) · 263 Bytes                                                                                                                                                                                                                                                                            |       |
| <pre>1  #!/bin/bash 2  echo "Starting After Install" 3  sudo cp -r /opt/codedeploy-agent/deployment-root/*/deployment-archive/* /var/www/html/ 4  sudo chmod -R 755 /var/www/html 5  sudo chown -R apache:apache /var/www/html 6  sudo systemctl restart httpd 7  echo "After Install completed"</pre> |       |

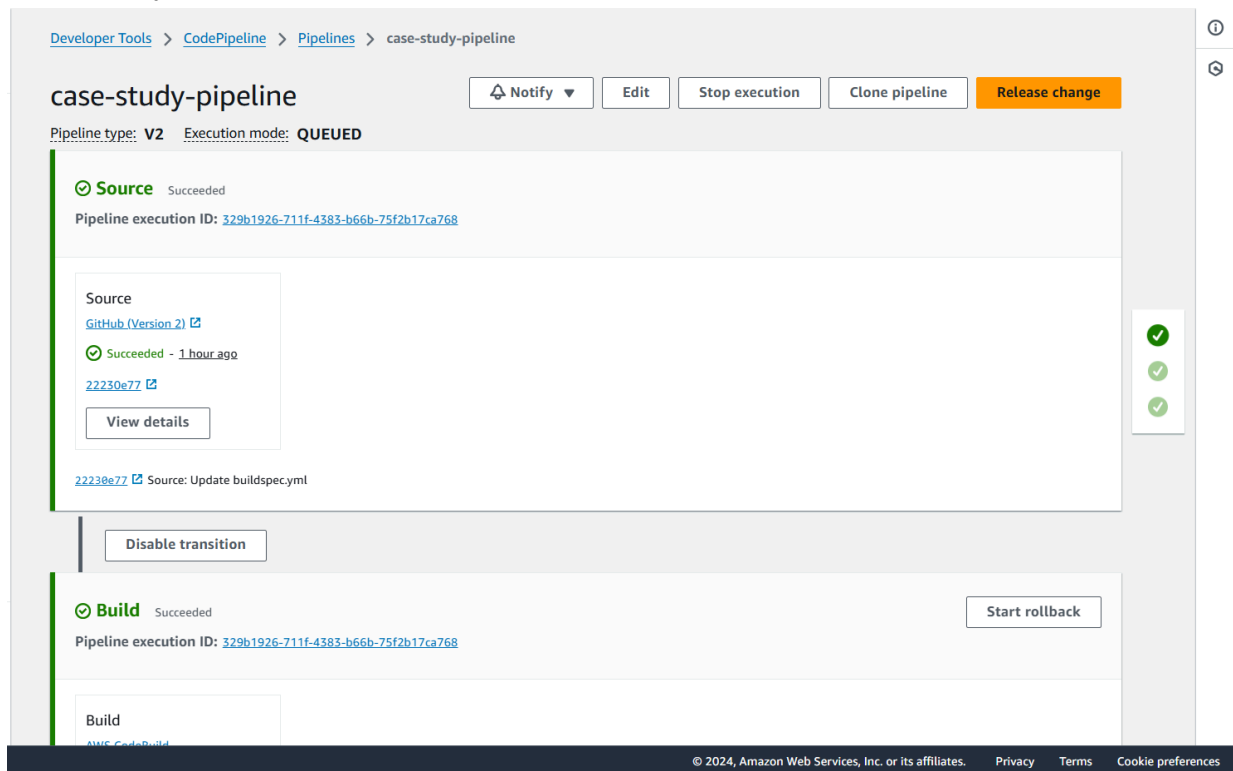


The screenshot shows a GitHub repository interface. At the top, the repository name is 'ADO-Case-Study' and the file path is 'scripts / before\_install.sh'. The user 'Adityacodesinpython' has updated the file. Below the file name, there are tabs for 'Code' and 'Blame', and statistics showing '7 lines (7 loc) · 175 Bytes'. The code content is as follows:

```
1 #!/bin/bash
2 echo "Starting Before Install"
3 sudo yum update -y
4 sudo yum install -y httpd
5 sudo systemctl start httpd
6 sudo systemctl enable httpd
7 echo "Before Install completed"
```

### 3. Add Deployment Stage to CodePipeline:

- Go back to your CodePipeline.



The screenshot shows the AWS CodePipeline console for a pipeline named 'case-study-pipeline'. The pipeline type is 'V2' and the execution mode is 'QUEUED'. The pipeline has two stages: 'Source' and 'Build'. The 'Source' stage is 'Succeeded' and the 'Build' stage is 'Succeeded'. The pipeline execution ID is '329b1926-711f-4383-b66b-75f2b17ca768'. The 'Source' stage details show it used 'GitHub (Version 2)' and 'Succeeded - 1 hour ago'. The 'Build' stage details show it used 'AWS CodeBuild'. The pipeline has a 'Release change' button and a 'Start rollback' button.

- Add a new stage for deployment.

The screenshot shows the 'Edit: Deploy' console window. At the top right are buttons for 'Cancel', 'Delete', and 'Done'. Below these are three buttons: 'Add entry condition', 'Add success condition', and 'Add failure condition'. A '+ Add action group' button is also present. The main area contains two action groups. The first group is named 'Deploy' and contains an action 'Amazon S3'. The second group is named 'Deploy-to-AWS-Code...' and contains an action 'AWS CodeDeploy'. Each action has edit and delete icons. A '+ Add action' button is to the right of the second group. At the bottom, there is an 'Automated stage configuration' section with a dropdown menu set to 'Enable automatic rollback on stage failure'.

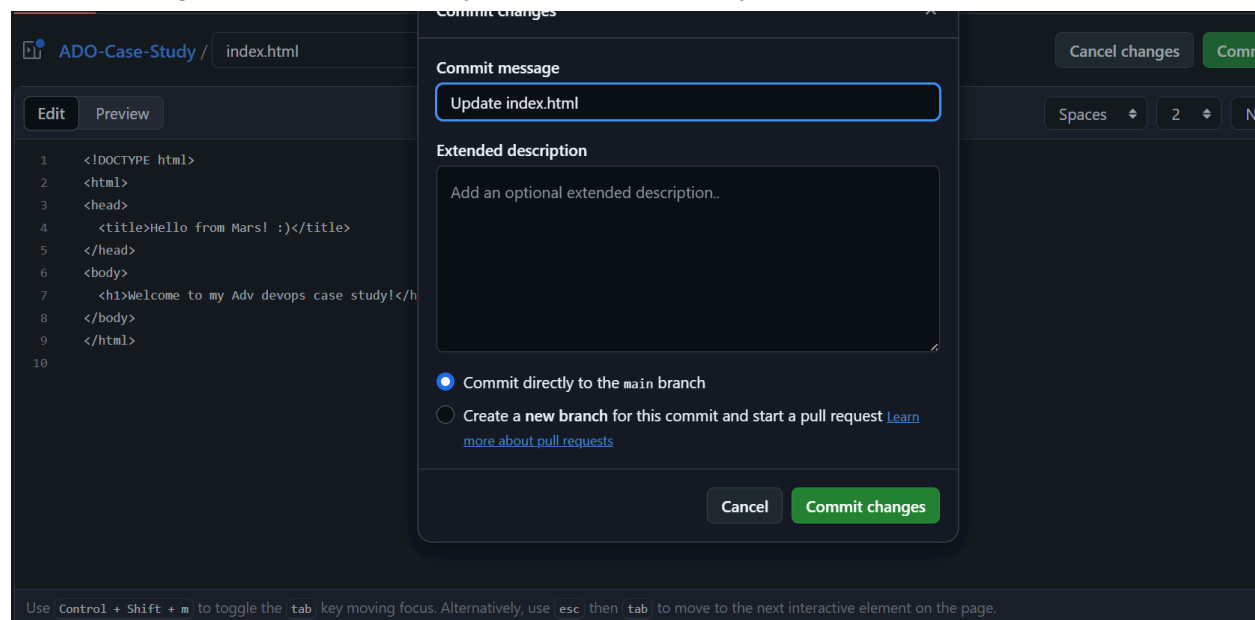
- Select **AWS CodeDeploy** and choose the application and deployment group you created earlier.

The screenshot shows the 'Edit action' console window. It has a title bar with a close button. The form contains several fields: 'Action name' (set to 'Deploy-to-AWS-Code-deploy'), 'Action provider' (set to 'AWS CodeDeploy'), 'Region' (set to 'Asia Pacific (Sydney)'), 'Input artifacts' (empty), 'Application name' (set to 'my-webapp'), and 'Deployment group' (set to 'my-webapp-group'). There are also buttons for 'BuildArtifact' and 'Variable namespace - optional'.

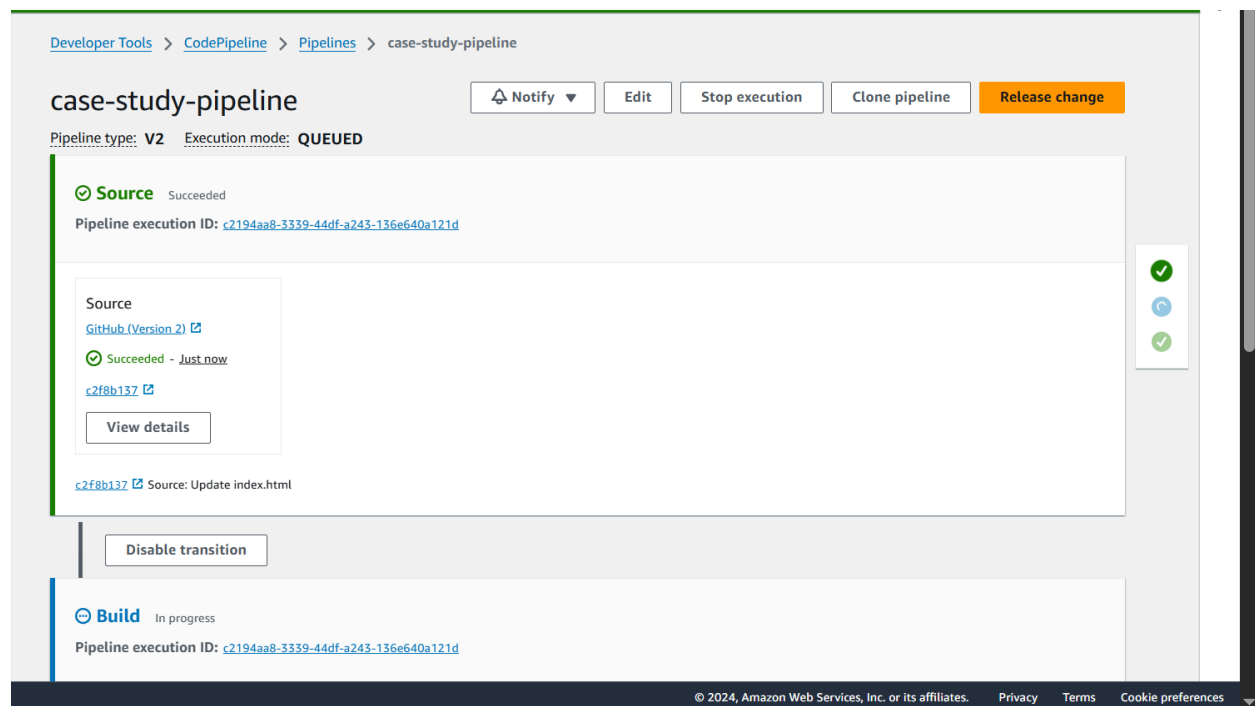
#### 4. Test Deployment:

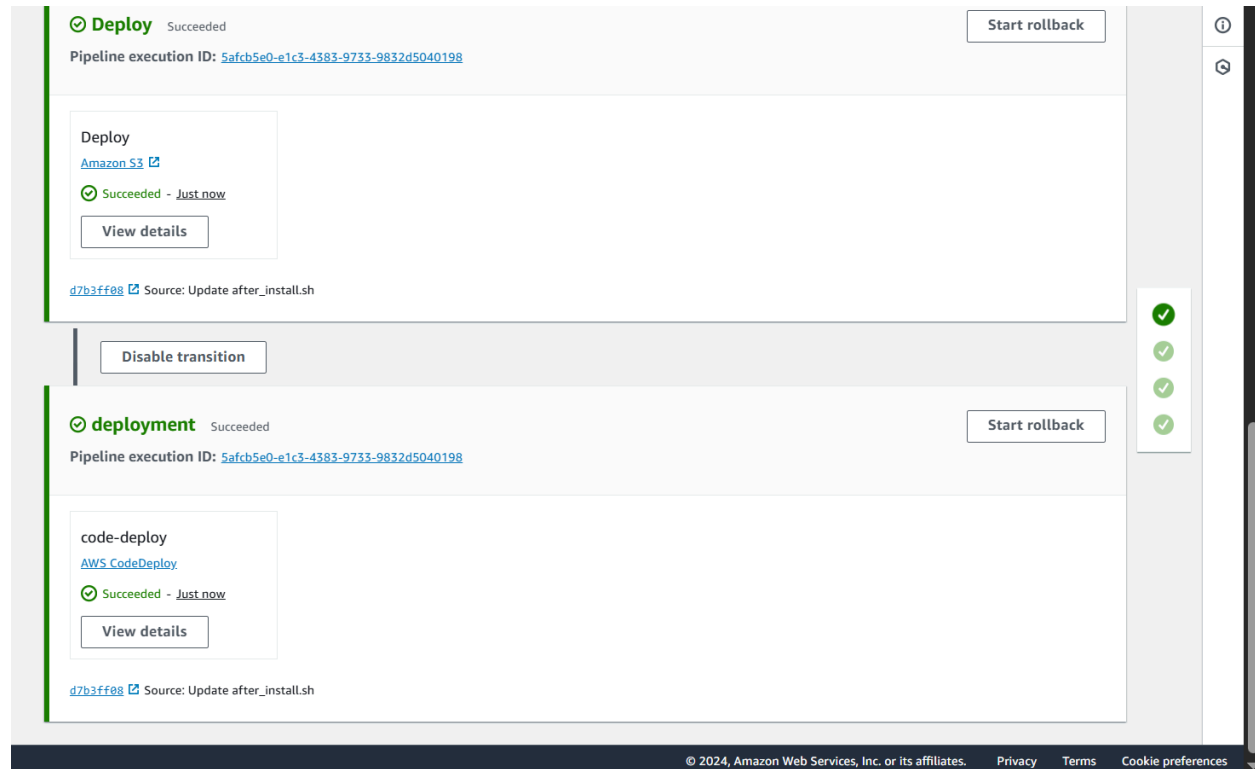


- Make a change to `index.html` in your source repository and push it.

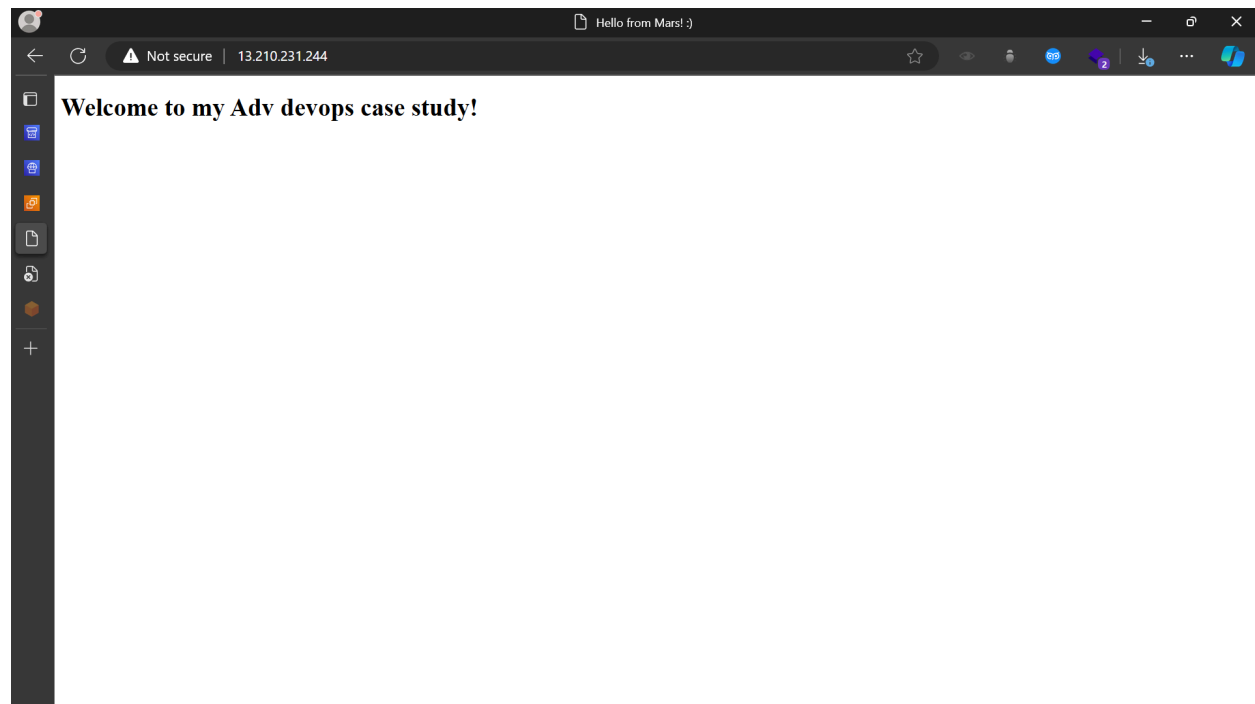


- This should trigger the pipeline, rebuild the app, push it to S3, and deploy it to the EC2 instance.





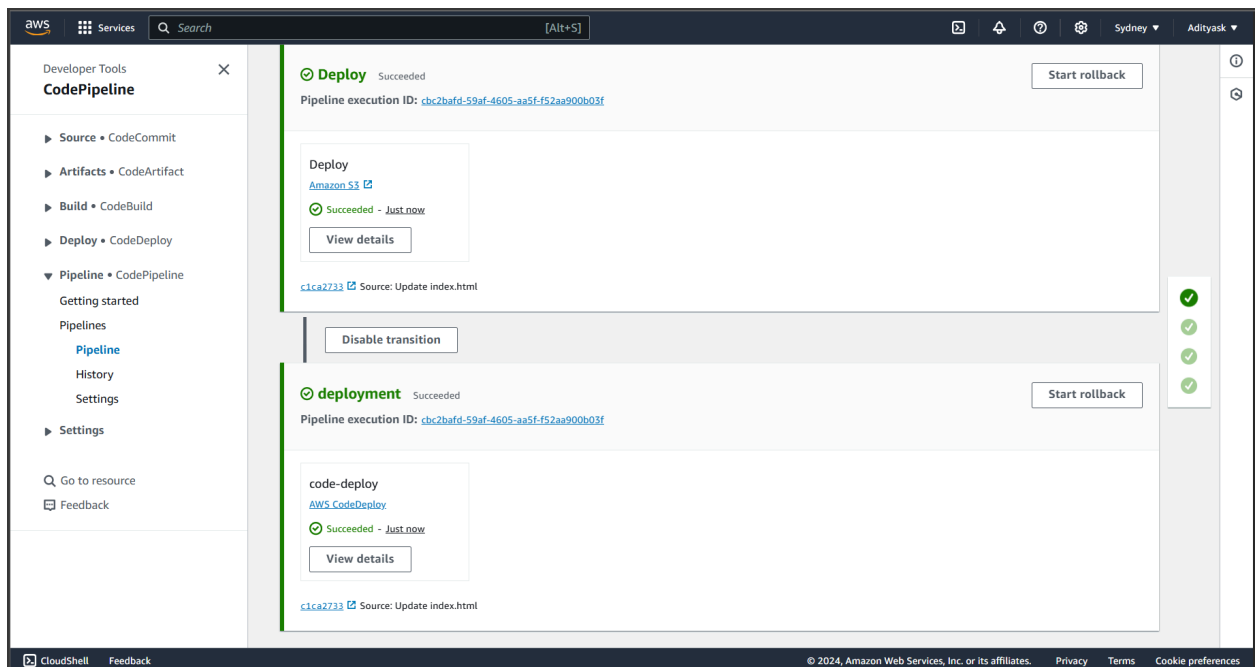
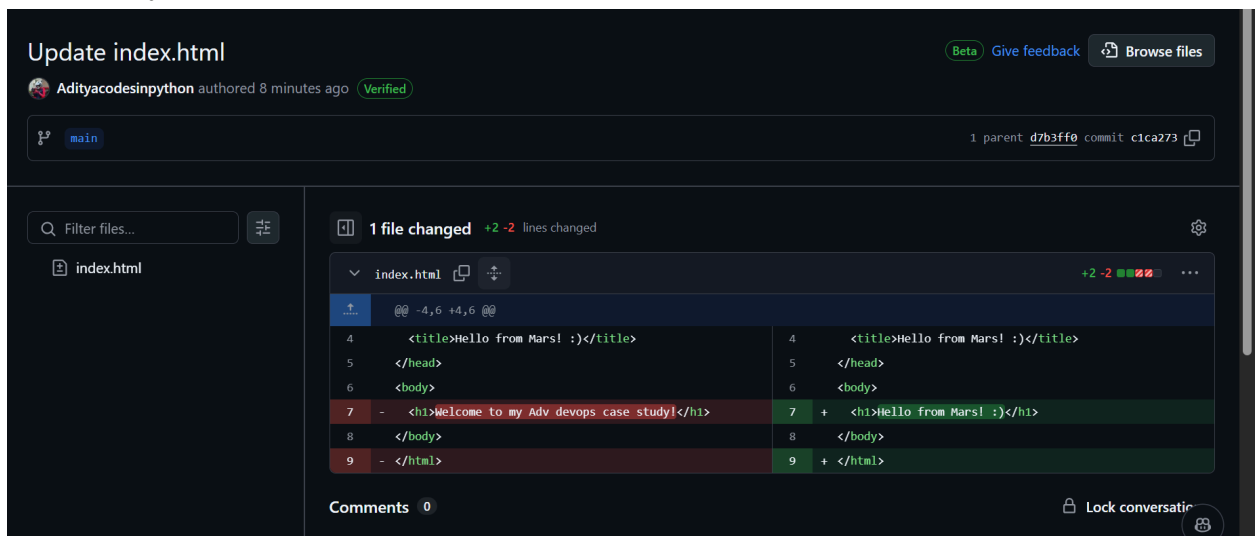
- You can access your EC2 instance via its public IP to view the updated web app.

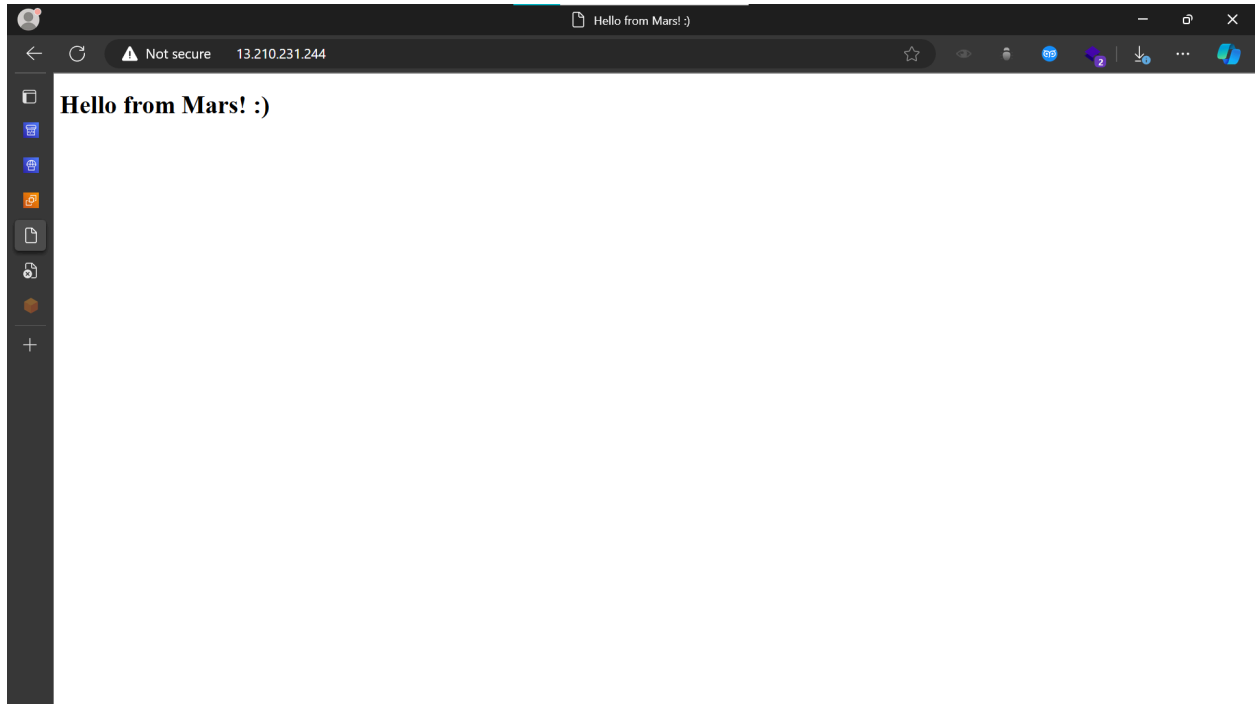


---

## 7. Verify Automation

- Now that your pipeline is set up, any changes to your repository (e.g., modifying the `index.html` file) should automatically trigger the build, deploy it to S3, and push updates to your EC2 instance.





### **CHALLENGES FACED:**

1. **Permission Issues:** One of the main hurdles was setting the correct permissions in IAM roles and the S3 bucket policy. Errors like **403 Forbidden** when accessing S3 were resolved by adjusting the S3 bucket policy.
2. **Missing AppSpec File:** The CodeDeploy process failed initially because the **appspec.yml** file was not placed correctly in the root directory of the build artifacts.
3. **CodeDeploy Agent Issues:** The EC2 instance had issues with the CodeDeploy agent, such as failing to start or showing **Permission Denied** errors. These were resolved by restarting the agent and ensuring the instance had appropriate permissions to access S3.

Despite these challenges, the overall deployment was successful, demonstrating the effectiveness of AWS's automation tools in managing continuous deployment workflows.

### **CONCLUSION:**

This experiment successfully demonstrated the process of building and deploying a simple web application using AWS CodePipeline, CodeBuild, CodeDeploy, and EC2. The pipeline automates the workflow from the moment the code is pushed to a repository until the application is deployed on an EC2 instance.