

Developers Tools

CodeCommit, CodeBuild, CodeDeploy & CodePipeline

Building a Angular Web Application [Cycling Experience] and Hosting at AWS Cloud EC2

Presented by Wong Teck Choy, 31 August 2024

AWS Developer Tools

AWS Developer Tools is a set of services designed to enable developers and IT operations professionals to rapidly and safely deliver software.

This suite includes **CodeCommit**, **CodeBuild**, **CodeDeploy** and **CodePipeline**, which together facilitate continuous integration and continuous delivery (CI/CD) of applications.



What does this AWS product do?

- **CodeCommit:** is a fully **managed source control service** that hosts Git repositories. It allows teams to store code securely and manage the lifecycle of their software development.
- **CodeBuild:** A fully **managed build service** that compiles source code, runs tests, and produces software packages ready for deployment. It scales continuously and processes multiple builds concurrently.
- **CodeDeploy:** A **deployment service** that automates the process of deploying applications to various compute services like Amazon EC2, AWS Lambda, and on-premises servers. It ensures deployments are safe and repeatable.
- **CodePipeline:** A **continuous integration and delivery service (CI/CD)** for fast and reliable application and infrastructure updates. CodePipeline **automates** the build, test, and deploy phases of your release process every time there is a code change.



Who will use this AWS Developer Tools

- **Software Developers** : Who need tools to build, test, and deploy code efficiently.
- **DevOps Engineers** : Who automate the software delivery process to improve deployment speed and reliability.
- **IT Operations Teams** : Who manage infrastructure and application updates.
- **QA Teams** : Who integrate testing into the CI/CD pipeline to ensure code quality

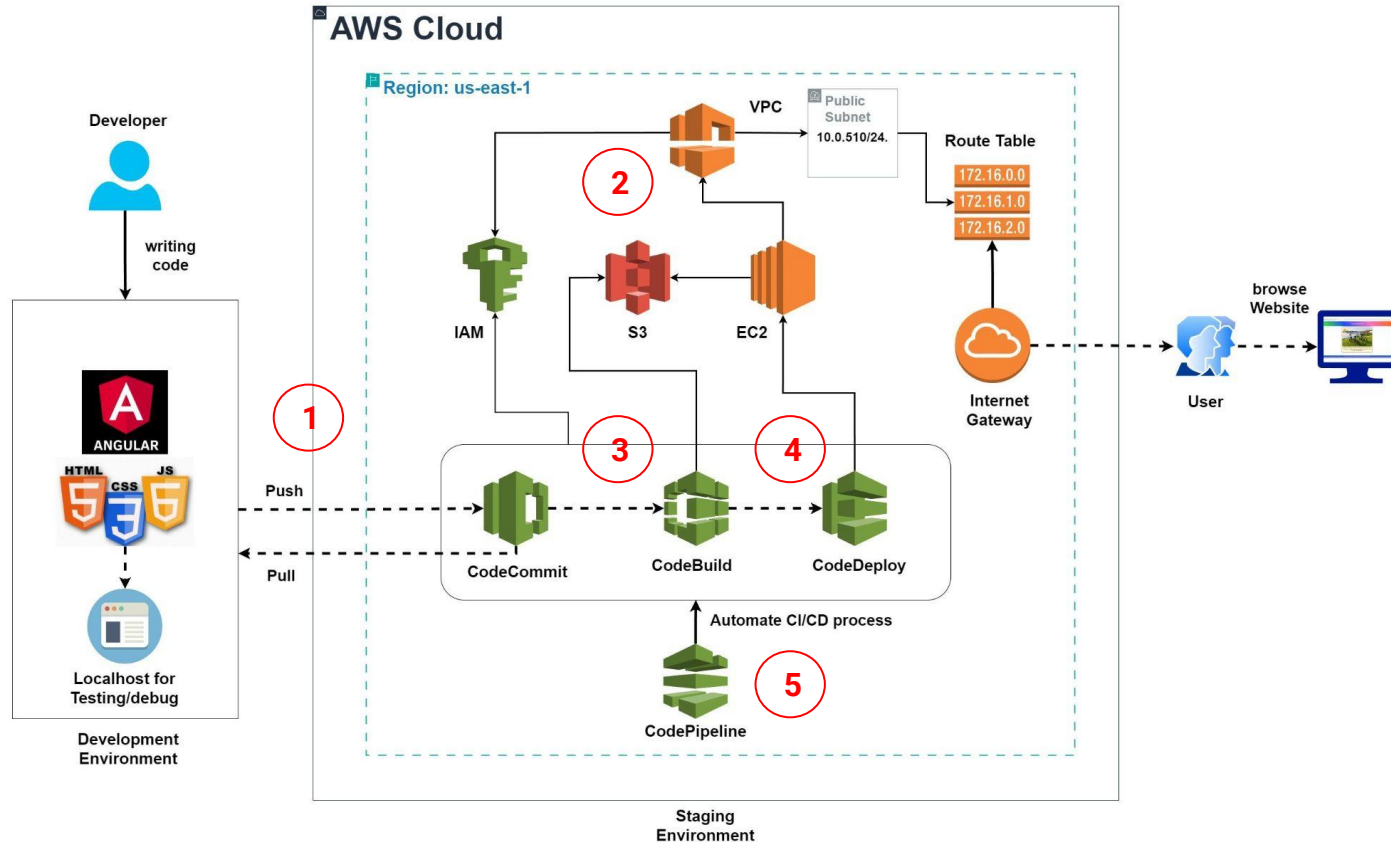


Agenda

1. Setup **CodeCommit** repository and upload the source code.
2. Setup **VPC, S3 bucket** and **EC2**
3. Setup **CodeBuild**
4. Setup **CodeDeploy**
5. Setup **CodePipeline**



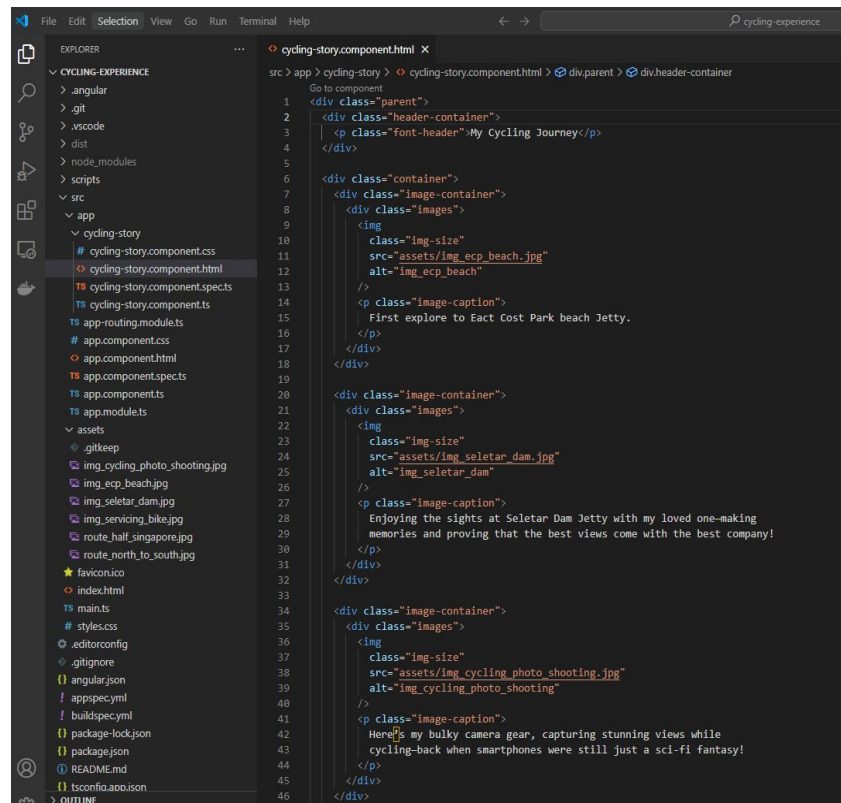
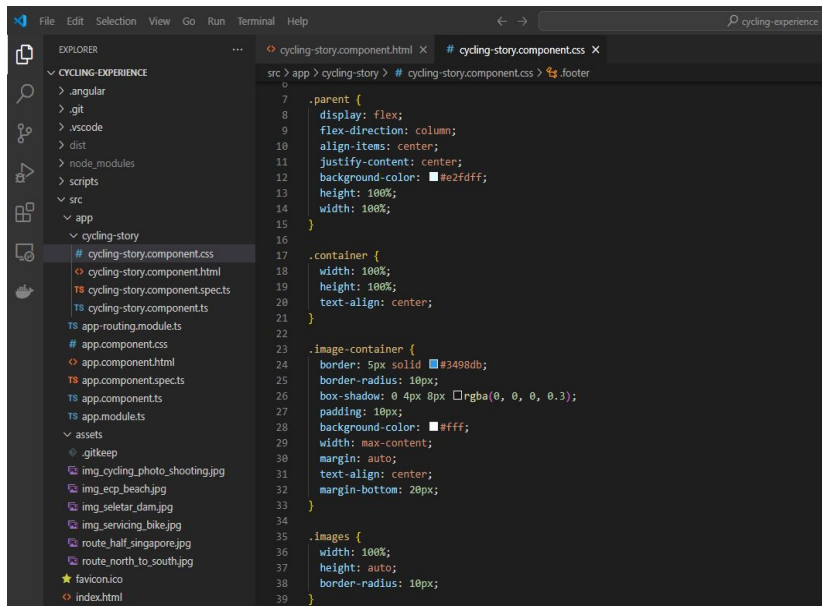
Architecture to Host Web Application in AWS Cloud



1. Setup CodeCommit repository and upload the source code

CodeCommit

- Creating **Angular** web application project using **Visual Studio Code** with **Angular Framework**.
- Writing code, build and testing at **local environment**.



CodeCommit

- Create “**buildspec.yml**” file.
- This is a configuration file used by **AWS CodeBuild** to define the **build process** for your project. It provides **instructions** on how to build, test, and package your code.

```
! buildspec.yml
1  version: 0.2
2  env:
3    variables:
4      APP_NAME: "cycling-experience"
5  phases:
6    install:
7      runtime-versions:
8        nodejs: 20
9      commands:
10       - echo install process started
11       - npm install && npm install -g @angular/cli
12    build:
13      commands:
14       - echo build process started now
15       - ng build
16    post_build:
17      commands:
18       - echo build process finished, we should upload to S3 now
19       - cp appspec.yml dist/$APP_NAME
20       - cp -r scripts dist/$APP_NAME
21       - ls -la dist/$APP_NAME
22    artifacts:
23      files:
24       - '**/*'
25       - appspec.yml
26       - scripts/**
27    base-directory: 'dist/$APP_NAME'
```

CodeCommit

- Create “**appspec.yml**” file and **scripts**
- This file is crucial for **AWS CodeDeploy** to handle the deployment process on EC2 instance, such as **executing scripts** to install the Apache server and **managing services**.

```
scripts > $ install_dependencies.sh
1  #!/bin/bash
2  sudo yum install httpd -y
```

```
scripts > $ start_server.sh
1  #!/bin/bash
2  sudo service httpd start
```

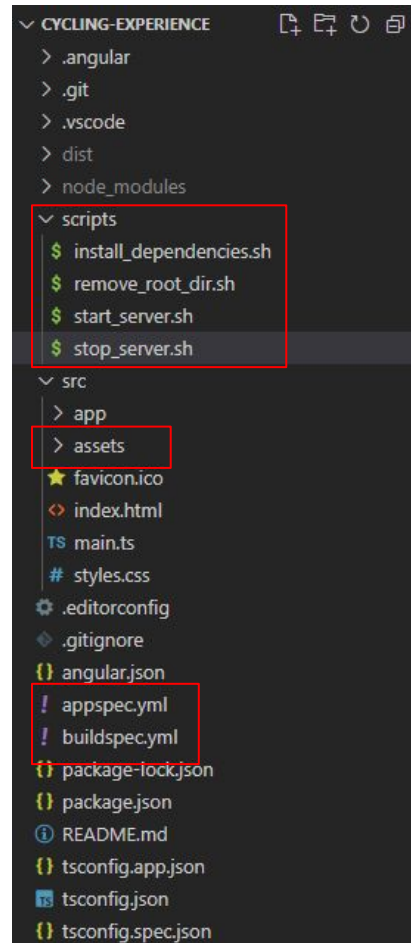
```
scripts > $ stop_server.sh
1  #!/bin/bash
2  isExistApp = `pgrep httpd`
3  if [[ -n $isExistApp ]]; then
4  |   sudo service httpd stop
5  fi
```

```
! appspec.yml
1  version: 0.0
2  os: linux
3  files:
4  | - source: /
5  |   destination: /var/www/html
6  hooks:
7  | BeforeInstall:
8  |   - location: install_dependencies.sh
9  |     timeout: 300
10 |     runas: root
11 | ApplicationStart:
12 |   - location: start_server.sh
13 |     timeout: 300
14 |     runas: root
15 | ApplicationStop:
16 |   - location: stop_server.sh
17 |     timeout: 300
18 |     runas: root
```

CodeCommit

Your project development **files and folder structure**.

- Angular project source code and **app** folder
- All **yml** files
- **scripts** folder - all scripts file
- **assets** folder - all documents or image files



CodeCommit

Create new CodeCommit repository

1. Repository name : **wtc-cycling-repo**
2. Enter the **description** and click “**Create**” button.

Developer Tools > CodeCommit > Repositories > Create repository

ⓘ AWS CodeCommit is no longer available to new customers. Existing customers of AWS CodeCommit can continue to use the service as normal. [Learn more](#)

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name

100 characters maximum. Other limits apply.

Description - *optional*

1,000 characters maximum

Tags

Add tag

► **Additional configuration**

AWS KMS key

☐ Enable Amazon CodeGuru Reviewer for Java and Python - *optional*

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

Cancel Create

CodeCommit

Generate credentials for **HTTPS Git credentials** for AWS CodeCommit

1. Go to **IAM**->**Users**->**HTTPS Git credentials** for AWS CodeCommit->Click on “**Generate credential**” button.
2. Download the credentials **csv file** on your local computer, it contain your **username** and **password**.

The screenshot shows the AWS IAM console interface. On the left is a sidebar with the 'Identity and Access Management (IAM)' header and a search bar. Below the search bar is a navigation menu with 'Dashboard' and 'Access management' expanded, showing options like 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', and 'Account settings'. The main content area is titled 'HTTPS Git credentials for AWS CodeCommit (1)'. It includes a description: 'Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can have a maximum of 2 sets of credentials (active or inactive) at a time. [Learn more](#)'. Below this is a table with columns 'User name', 'Created', and 'Status'. The table contains one entry with the username 'tcwong99-at-255945442255', created 'Now', and status 'Active'. To the right of the table is an 'Actions' dropdown menu, and the 'Generate credentials' option is highlighted with a red rectangular box.

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings

User SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time. [Learn more](#)

SSH Key ID	Uploaded	Status
No SSH public keys		
Upload SSH public key		

HTTPS Git credentials for AWS CodeCommit (1)

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can have a maximum of 2 sets of credentials (active or inactive) at a time. [Learn more](#)

User name	Created	Status
<input type="radio"/> tcwong99-at-255945442255	Now	Active

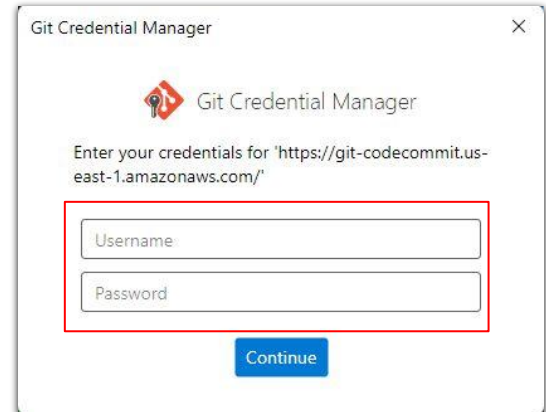
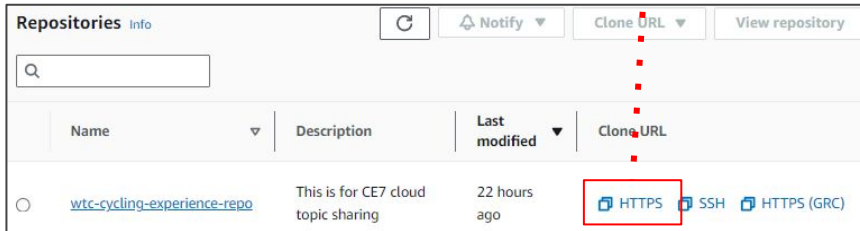
Actions ▼

Generate credentials

CodeCommit

Upload your code to CodeCommit

1. Please refer to the downloaded **csv file**. You may need to enter **username** and **password** if prompted.
2. Below git commands use to upload source code to AWS CodeCommit.
 - `git clone <Your CodeCommit repository HTTPS url>` or
`git remote add origin <Your CodeCommit repository HTTPS url>`
 - `git status`
 - `git add .`
 - `git commit -m "Initial project code"`
 - `git push -u origin master`
 - `git pull origin master`



CodeCommit

- Once git push successfully execute, your source code will be available at AWS CodeCommit repository.
- To check source code go to CodeCommit -> Repository -> <Choose your repository>

The screenshot displays the AWS CodeCommit console interface. On the left, a sidebar menu under 'Developer Tools' lists various actions for the 'wtc-cycling-experience-repo' repository, including 'Source', 'Pull requests', 'Commits', 'Branches', 'Git tags', 'Settings', and 'Approval rule templates'. The main content area shows the repository's file list. A blue banner at the top of the main area states: 'AWS CodeCommit is no longer available to new customers. Existing customers of AWS CodeCommit can continue to use the service as normal. [Learn more](#)'. Below this, the repository name 'wtc-cycling-experience-repo' is displayed, along with a 'Reference' dropdown set to 'master' and buttons for 'Notify', 'Create pull request', and 'Clone URL'. The file list table has a header 'Name' and contains the following entries: '.vscode', 'src', '.editorconfig', '.gitignore', 'angular.json', 'buildspec.yml', 'package-lock.json', 'package.json', 'README.md', 'tsconfig.app.json', 'tsconfig.json', and 'tsconfig.spec.json'. A red rectangular box highlights the first six items in the file list: '.vscode', 'src', '.editorconfig', '.gitignore', 'angular.json', and 'buildspec.yml'.

Name
.vscode
src
.editorconfig
.gitignore
angular.json
buildspec.yml
package-lock.json
package.json
README.md
tsconfig.app.json
tsconfig.json
tsconfig.spec.json

2. Setup VPC, S3 bucket and EC2

Setup VPC, S3 bucket and EC2

1. Setting up a **VPC** with at least 1 public subnet link to internet gateway.
2. Create a **S3 Bucket** use for CodeBuild.
3. Setting up a **EC2 using Amazon Linux** as OS.
 - Enter below command for EC2 **User data**

```
#!/bin/bash
sudo yum update -y
sudo yum install ruby -y
sudo yum install wget -y
cd /home/ec2-user
wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
sudo chmod +x ./install
sudo ./install auto
systemctl status codedeploy-agent
systemctl start codedeploy-agent
```

Setup VPC, S3 bucket and EC2

The screenshot displays the AWS Management Console interface, divided into three main sections: VPC, S3, and EC2.

VPC Section:

- VPC:** Shows the VPC `wtc-tf-mod-vpc` with CIDR `10.0.0.0/16` and No IPv6.
- Subnets (6):** Lists subnets for `us-east-1a`, `us-east-1b`, and `us-east-1c`. Each region has a public subnet (e.g., `wtc-tf-mod-vpc-public-us-east-1a`) and a private subnet (e.g., `wtc-tf-mod-vpc-private-us-east-1a`).
- Route tables (5):** Lists route tables including `wtc-tf-mod-vpc-private-us-east-1a`, `wtc-tf-mod-vpc-public-route`, `wtc-tf-mod-vpc-private-us-east-1c`, `wtc-tf-mod-vpc-default`, and `wtc-tf-mod-vpc-private-us-east-1b`.
- Network connections (1):** Shows `wtc-tf-mod-igw` (Internet Gateway) with routes to public subnets.

S3 Section:

- Bucket:** `wtc-s3-codebuild-artifacts` (Info icon highlighted).
- Properties:** Shows the bucket's location: `US East (N. Virginia) us-east-1`.
- Amazon Resource Name (ARN):** `arn:aws:s3:::wtc-s3-codebuild-artifacts`.

EC2 Section:

- Instance:** `i-06cb9598fb2f65a21 (wtc-ec2-topic-sharing-dont-delete)` (ID highlighted).
- Details:** Shows instance details including `Instance ID`, `Public IPv4 address` (`3.91.172.102`), `Private IPv4 addresses` (`10.0.51.212`), and `Instance state` (`Running`).

3. Setup CodeBuild

[Demo in video](#)



4. Setup CodeDeploy

Demo in video

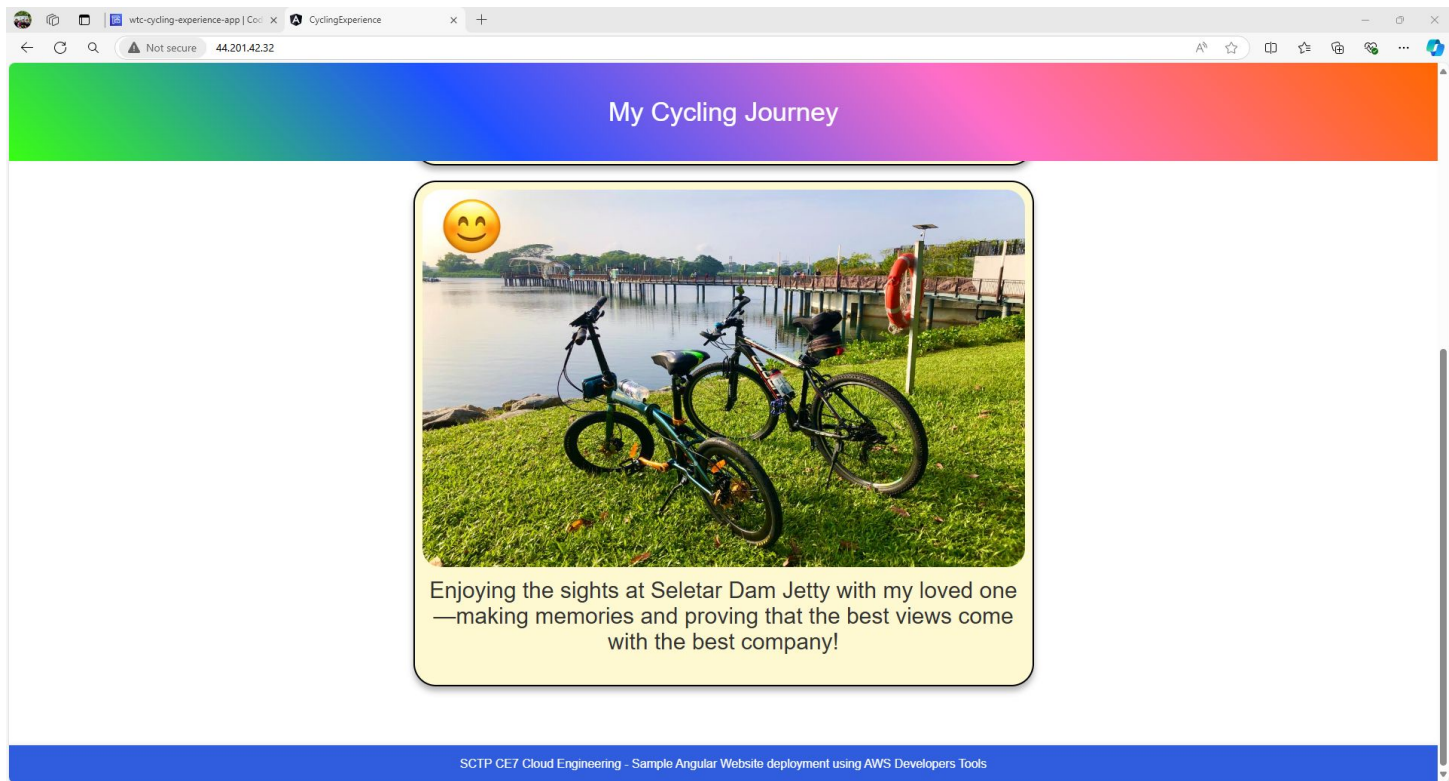


5. Setup CodePipeline

Demo in video



Website Now Running on AWS Cloud



End of Topic Sharing
Thank you!

