

(SCTP) Cloud Infrastructure Engineering

Capstone Project
Cohort 3
Group 1

Table of Contents

01

Introduction

Use Case
Who We Are

02

System Architecture Technology Stack

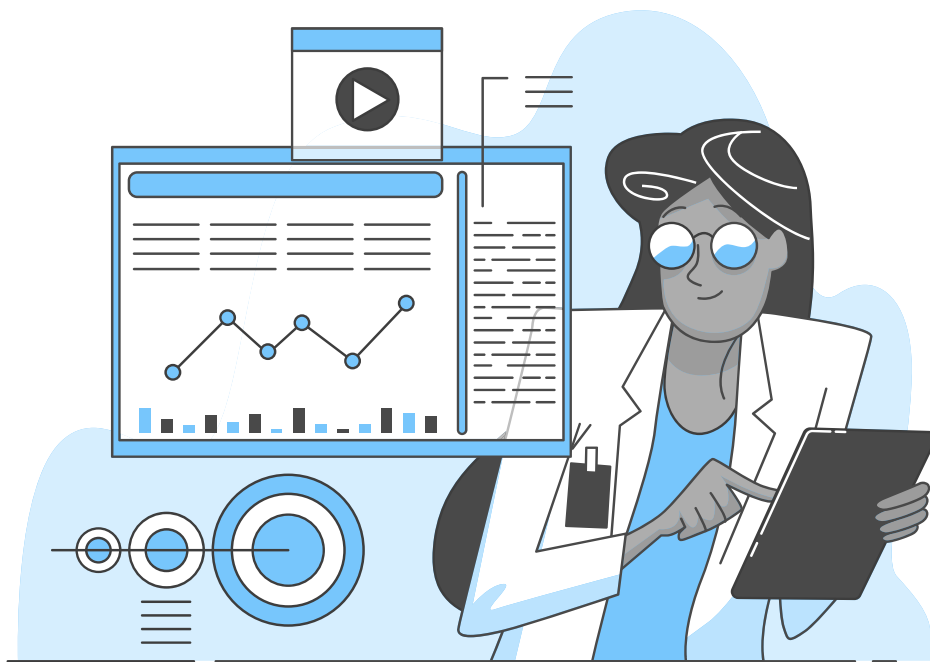
03

Branching Strategy

Workflow
Branch Protections

04

Outputs and Conclusions






01

Introduction

Use Case
Who We Are



Use Case 1 – CI/CD Pipeline Focused (DevOps)

Scenario:

You are working in a new startup.

You are tasked to **automate the deployment process** so that each release cycle can be released quickly from the development environment, to staging and to production environment.

As the company is going at a fast paced growth rhythm, there will be more engineers onboarding in the next few quarters. The tech lead needs to ensure that as the team size grows, **the release pipeline is well controlled**.





Our Company

From blueprint to blast-off in record time. We're the DevOps alchemists, transforming your ideas into scalable websites, enabling rapid deployment that leave your competition in the dust.

...



Our Story



01

Client

Online Education
Training Provider

02

Static Website

Develop and deploy a newly
revamped website in AWS S3

03

DevOps Principles

Streamline workflow
using GitHub Actions

04

Infrastructure as Code

Terraform automates
deployments to AWS



02

System Architecture & Technology Stack

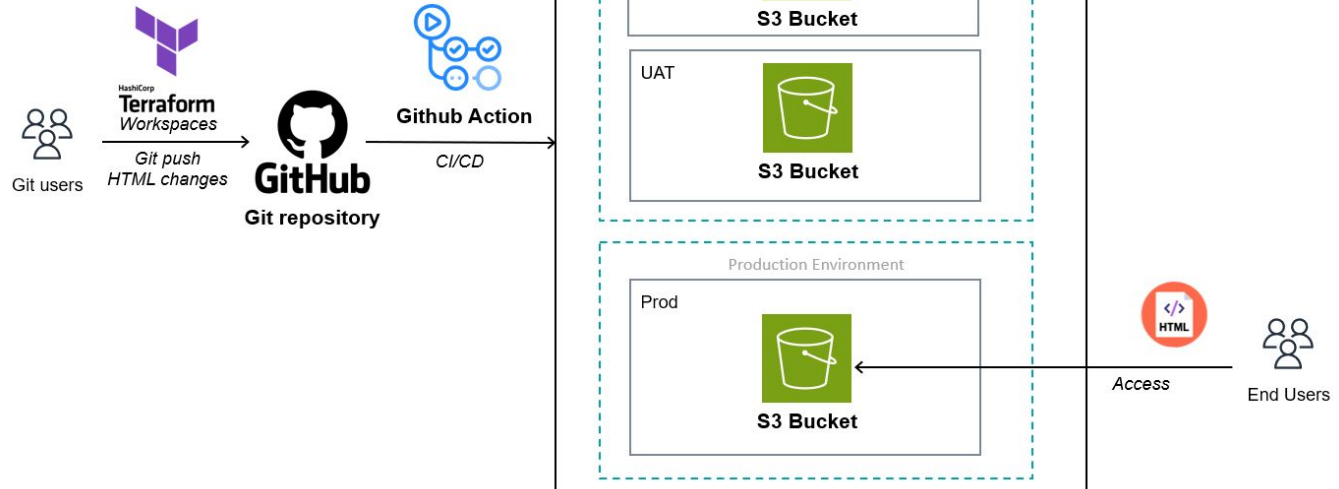


AWS Cloud Architecture

3

AWS S3 Buckets

- Dev
- UAT
- Prod



Technology Stack



Terraform

Resources Created
through Terraform



Terraform Workspaces

Infrastructure Environment
Setup/Management



GitHub Actions

CI/CD



Snyk & Checkov

Package/Vulnerability Scans



Technology Stack



Terraform

Resources Created
through Terraform



Terraform Workspaces

Infrastructure Environment
Setup/Management



GitHub Actions

CI/CD



Github Secrets

Secrets Management and Access
Control



03

Branching Strategy

Workflow
Branch Protection

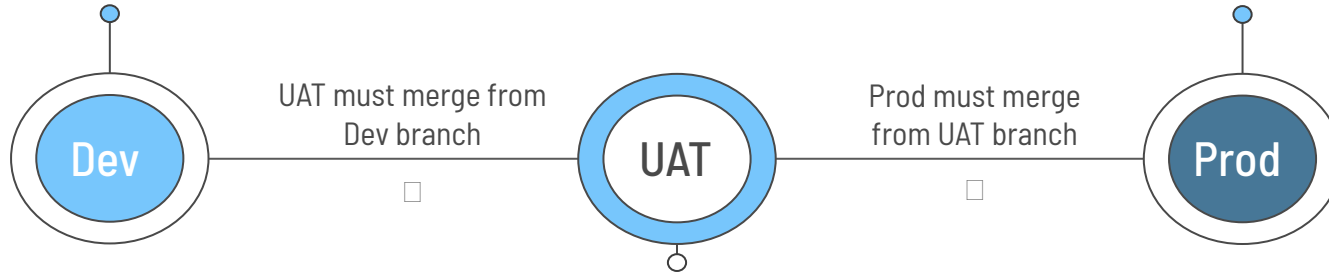
Workflow – 3 branches

1. Development Branch

- Primary integration branch - feature branches merge here for integration testing.
- Continuous integration implemented to allow automated testing and verification of code changes.

3. Production Branch

- Contains stable and thoroughly tested code
- Only fully reviewed and approved code changes can be merged
- Direct commits or modifications are restricted



2. User Acceptance Testing (UAT)

- Serves UAT for ongoing testing.
- It acts as a staging area for testing before deployment into Production

CI/CD

Single CI/CD yml file

Utilising **Terraform Workspaces** and **Variable Definition Files (.tfvars)** to manage multiple environment

✓ .github/workflows

📄 cicd.yml

> static-website

📄 .gitignore

📄 README.md

📄 backend.tf

📄 bucket-policy.tf

📄 dev.tfvars

📄 main.tf

📄 output.tf

📄 prod.tfvars

📄 provider.tf

📄 uat.tfvars

📄 variables.tf

```
CI:
  runs-on: ubuntu-latest
  steps:
```

```
  - name: Terraform init
    run: terraform init

  - name: Terraform workspace list
    run: terraform workspace list

  - name: Terraform workspace new environment and select
    run: |
      terraform workspace select ${github.base_ref || github.ref_name} || terraform workspace new ${github.base_ref || github.ref_name}
      terraform workspace list

  - name: Terraform plan
    run: terraform plan -var-file=${github.base_ref || github.ref_name}.tfvars
```

Terraform State Files

Amazon S3 > Buckets > sctp-ce3-tfstate-bucket-1 > env:/



Copy S3 URI

Copy URL

Download

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to grant them permissions. [Learn more](#)

Find objects by prefix

Show versions

| <input type="checkbox"/> | Name | Type |
|--------------------------|------------------|--------|
| <input type="checkbox"/> | dev/ | Folder |
| <input type="checkbox"/> | feature-test-br/ | Folder |
| <input type="checkbox"/> | feature1/ | Folder |
| <input type="checkbox"/> | main/ | Folder |
| <input type="checkbox"/> | prod/ | Folder |
| <input type="checkbox"/> | uat/ | Folder |

| <input type="checkbox"/> | Name | Type | Last modified |
|--------------------------|--|---------|-------------------------------|
| <input type="checkbox"/> | sctp-ce3-capstone-group1-final.tfstate | tfstate | January 17, 2024, (UTC+08:00) |

| <input type="checkbox"/> | Name | Type | Last modified |
|--------------------------|--|---------|-------------------------------|
| <input type="checkbox"/> | sctp-ce3-capstone-group1-final.tfstate | tfstate | January 17, 2024, (UTC+08:00) |

| <input type="checkbox"/> | Name | Type | Last modified |
|--------------------------|--|---------|-------------------------------|
| <input type="checkbox"/> | sctp-ce3-capstone-group1-final.tfstate | tfstate | January 17, 2024, (UTC+08:00) |



CI/CD – Continuous Integration



```
CI:
  runs-on: ubuntu-latest

  steps:
    - name: Checkout repository
      uses: actions/checkout@v2

    - name: print environment
      run: |
        echo "env choice is ${github.base_ref || github.ref_name}"

    - name: Set up Terraform
      uses: hashicorp/setup-terraform@v3

    - name: Configure AWS credentials
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
        aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
        aws-region: us-east-1

    - name: Terraform init
      run: terraform init

    - name: Terraform workspace list
      run: terraform workspace list

    - name: Terraform workspace new environment and select
      run: |
        terraform workspace select ${github.base_ref || github.ref_name} || terraform workspace new ${github.base_ref || github.ref_name}
        terraform workspace list

    - name: Terraform plan
      run: terraform plan -var-file=${github.base_ref || github.ref_name}.tfvars
```

CI

- Configured to run on Push or Pull Requests
- Main Steps: Terraform Init, Terraform Plan
- Access to AWS configured using Github Secrets

CI/CD – Continuous Delivery

CD:

```
if: ${ github.event_name == 'push' }}
runs-on: ubuntu-latest
steps:
  - name: Checkout repository
    uses: actions/checkout@v2

  - name: Terraform init
    run: terraform init

  - name: Terraform workspace list
    run: terraform workspace list

  - name: Terraform workspace new environment and select
    run: |
      terraform workspace select ${github.base_ref || github.ref_name} || terraform workspace new ${github.ref_name}
      terraform workspace list

  - name: Terraform Apply
    run: terraform apply -var-file=${github.base_ref || github.ref_name}.tfvars -auto-approve

  - name: Deploy to AWS S3
    run: aws s3 sync . s3://${steps.tfout.outputs.s3_bucket_name} --delete
    working-directory: ./static-website
```

CD

- Configured to run only on Push
- Steps : Terraform Init, Terraform workspace select, Terraform Apply, Sync S3 Bucket.

After Pull request is approved, Merge ("Push") will run the CD job to apply infrastructure changes and update index.html file



Branch Protections

All branches are protected

- Github Branch Protection Rules
- Requires Pull request and reviewers
- All checks must pass

☒ Require status checks to pass before merging

Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

☒ Require branches to be up to date before merging

This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

🔍 Search for status checks in the last week for this repository

Status checks that are required.

Branch protection rule



Protect your most important branches

[Branch protection rules](#) define whether collaborators can delete or force push to the branch and set requirements for any pushes to the branch, such as passing status checks or a linear commit history.

[Your GitHub Free plan](#) can only enforce rules on its public repositories, like this one.

Branch name pattern *

dev

Protect matching branches

☒ Require a pull request before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

☒ Require approvals

When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.

Required number of approvals before merging: 1 ▼

Workflow + CI/CD

Pull Request Submitted

Triggers CI job

Changes Deployed

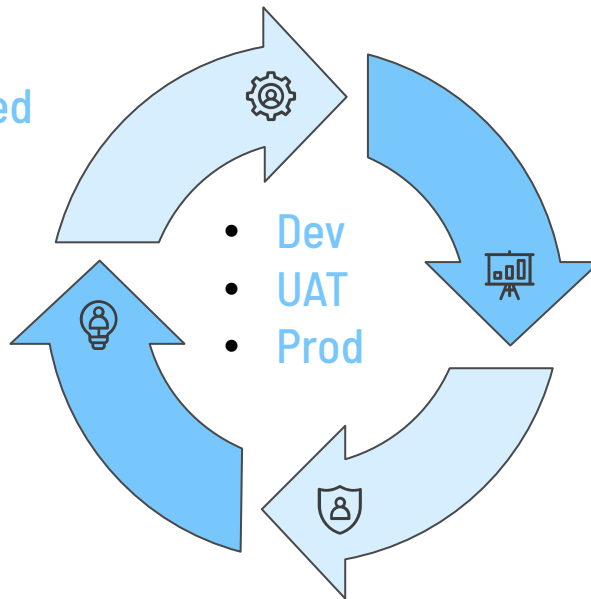
Changes are deployed,
Branch is updated

Review

Reviewer checks code
changes

Reviewer Approves

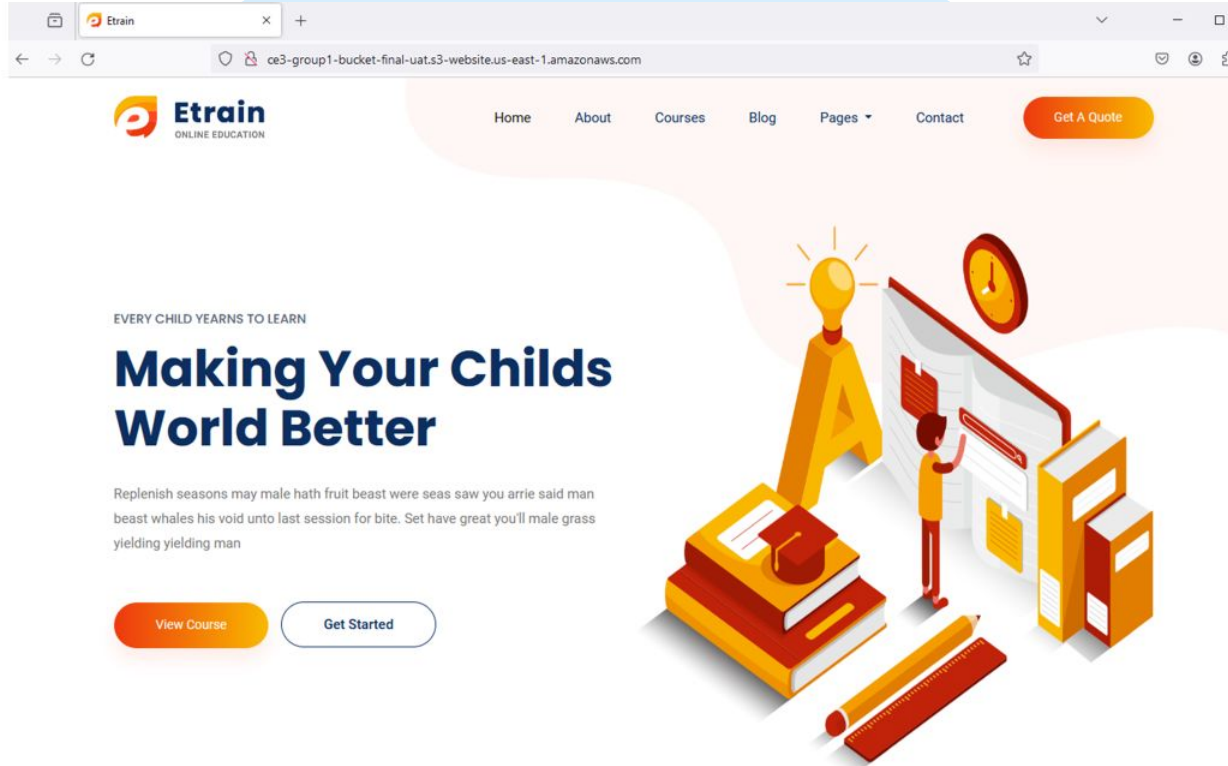
Changes are Merged
Triggers CD job



04

Outputs and Conclusion

Website



Conclusion

- AWS static website hosting
- Infrastructure as a Code – Terraform
- Terraform Workspaces for multi-environment management
- Github Actions for CI/CD
- Branching Strategies



Our Team

...

Caroline Lee

CEO

...

Wei Zhi Min

DevOps Engineer

...

Sharil Reiza

DevOps Engineer

...

Ramesh D

Software
Engineer

...

Dior Ng

Software
Engineer

