Unit - 5 DevOPS

Implementing Git-based DevOps for Project Deployment

Submitted By:

2021UCA1801 Kumar Gaurav

2021UCA1833 Utkarsh Gupta

2021UCA1862 Aditya Singh

Implementation of a Git-based DevOps approach for project deployment using GitHub Actions. The main objective was to automate the deployment process of static content to GitHub Pages using a CI/CD pipeline.

Code:

https://github.com/utkarshh05/GAS-GUARD/blob/main/gasguard.yaml

Approach

We embraced the GitOps philosophy, leveraging GitHub Actions as our CI/CD tool to automate the deployment pipeline. The workflow (static.yml) was meticulously crafted to ensure seamless deployment while adhering to best practices in DevOps.

Workflow Description

The workflow is named "Deploy static content to Pages" and is comprehensively structured:

Permissions

The permissions section plays a crucial role in securing the deployment process. It carefully assigns permissions to the GitHub token, granting read access to repository contents and write access to Pages. This ensures that the deployment process is authorized to access and modify the required resources.

Concurrency

Concurrency settings are configured to maintain a streamlined deployment process. By allowing only one deployment to run at a time (`group: "pages"`), we prevent conflicts and ensure that deployments are executed in an orderly fashion. Additionally, we've chosen not to cancel in-progress runs, prioritizing the completion of ongoing deployments to maintain system stability.

Jobs

The workflow defines a single job (`deploy`) dedicated to orchestrating the deployment process. This job encompasses the following key elements:

- **Environment**: The GitHub Pages environment is defined within the job configuration, specifying its name and URL. This ensures that the deployment occurs within the appropriate environment context.
- **Runs-On**: The job is configured to run on the latest version of the Ubuntu operating system, providing a reliable and consistent execution environment.
- **Steps**: A series of meticulously crafted steps are executed within the job to facilitate the deployment process. These steps include:
    - **Checkout**: The repository's code is checked out, ensuring that the latest changes are incorporated into the deployment artifact.
    - **Setup Pages**: GitHub Pages settings are configured to facilitate the deployment of static content.
    - **Upload Artifact**: The entire repository, along with its contents, is packaged and uploaded as an artifact. This ensures that all necessary files and configurations are included in the deployment.

Usage

Utilizing this workflow is straightforward and can be accomplished through the following steps:

1. Configuration: Ensure that the static.ym file is present within the github/workflows directory of your repository.
2. Trigger Deployment: Push changes to the default branch ( )nmitao automatically trigger the deployment workflow. Alternatively, manually initiate the workflow execution from the GitHub Actions tab.

CODE:

```
# @format

name: Gas Guard Automation CI/CD

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

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

      - name: Set up Node.js
        uses: actions/setup-node@v2
        with:
          node-version: "14"

      - name: Install dependencies
        run: npm install

      - name: Run tests
        run: npm test

  deploy:
    needs: build
    runs-on: ubuntu-latest
```

```
steps:
  - name: Checkout code
    uses: actions/checkout@v2

  - name: Set up Node.js
    uses: actions/setup-node@v2
    with:
      node-version: "14"

  - name: Install dependencies
    run: npm install

  - name: Build and deploy
    run: |
      npm run build
      # Add commands for deploying the application
```

Conclusion

In conclusion, the implementation of a Git-based DevOps approach for project deployment using GitHub Actions has proven to be highly effective. By automating the deployment pipeline, we've significantly reduced manual intervention, minimized deployment errors, and enhanced overall efficiency. Moving forward, we remain committed to refining and optimizing our DevOps practices to further streamline our development and deployment processes.