COMP4964 Data Pipeline CI/CD Automation

A Demonstration of Seamless Integration and Deployment for Data Workflows

Group 5

Neriyel Reyes

Dipenvir Kaur

Automated Data Pipeline: Project Overview

This project showcases a robust CI/CD pipeline designed for data processing, ensuring reliability and efficiency from code commit to deployment.

Automated Jenkins Pipeline

Hosted securely on DigitalOcean, orchestrating the entire CI/CD workflow.

GitHub Webhook Integration

Triggers automatic builds upon every code push, maintaining continuous integration.

AWS Lambda Processing

Serverless function designed to efficiently process CSV files from S3 buckets.

Automated Unit Tests

Ensures code quality and prevents deployment of broken or faulty code.

Key Technologies Driving Our Automation

Our solution leverages industry-leading tools to create a resilient and scalable data pipeline.

CI/CD Orchestration: Jenkins

The heart of our continuous integration and continuous delivery system, Jenkins automates the build, test, and deployment phases.

Version Control: GitHub

Manages our codebase and integrates webhooks for real-time pipeline triggers.

Serverless Compute: AWS Lambda

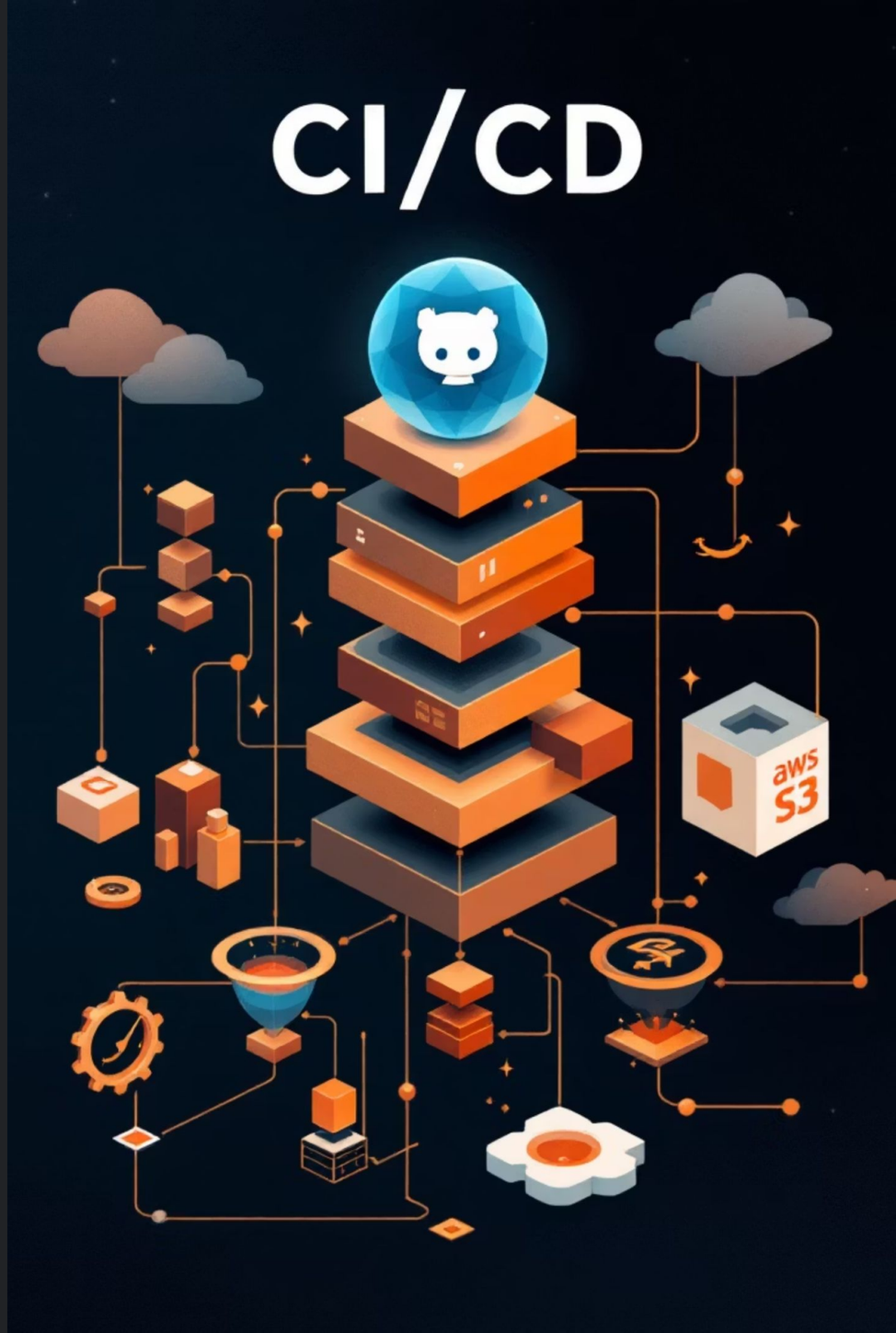
Executes our data processing logic without server provisioning or management.

File Storage: AWS S3

Provides secure, scalable, and highly available storage for our raw and processed data files.

Infrastructure as Code: CloudFormation

Defines and provisions all AWS infrastructure resources using a declarative template, ensuring consistency and repeatability.



End-to-End Pipeline Architecture

Code Push

Commit and push changes to GitHub

Webhook

GitHub webhook notifies CI

Jenkins Trigger

Jenkins starts pipeline automatically

Pipeline

Checkout, Test, Build, Package, Deploy, Validate

Jenkins File Explained

```groovy
pipeline {

    environment {
        AWS_REGION = 'us-west-2'
        STACK_NAME = 'data-pipeline-stack'
        ENVIRONMENT = 'dev'
        S3_DEPLOY_BUCKET = 'jenkins-sam-artifacts-195275680578-20251125123539'
        AWS_DEFAULT_REGION = 'us-west-2'
    }

    stages {
        stage('Checkout') {
            steps {
                echo 'Checking out code from GitHub...'
                checkout scm
            }
        }

        stage('Test') {
            steps {
                echo 'Running unit tests for Lambda function...'
                sh '''
                    cd jenkins-data-pipeline
                    python3 -m pytest tests/test_lambda.py -v
                '''
            }
        }

        stage('Build') {
            steps {
                echo 'Validating SAM template...'
                sh '''
                    echo "Checking template.yaml exists..."
                    ls -la jenkins-data-pipeline/template.yaml
                    echo "Template validated!"
                '''
            }
```

```groovy
        stage('Package') {
            steps {
                echo ' Preparing for deployment...'
                sh '''
                    echo "Using pre-built template..."
                    cp jenkins-data-pipeline/template.yaml ./packaged.yaml
                    echo "Ready to deploy!"
                '''
            }
        }

        stage('Deploy') {
            steps {
                echo ' Deploying to AWS...'
                sh '''
                    echo "========================================"
                    echo "CloudFormation Stack Deployment"
                    echo "========================================"
                    echo "Stack Name: ${STACK_NAME}"
                    echo "Region: ${AWS_REGION}"
                    echo "Environment: ${ENVIRONMENT}"
                    echo "Template: packaged.yaml"
                    echo ""
                    echo "Template size: $(wc -c < packaged.yaml) bytes"
```

```groovy
    post {
        success {
            echo ' Pipeline completed successfully!'
            echo 'Your data pipeline is ready for use.'
        }
        failure {
            echo ' Pipeline failed. Check logs above.'
        }
    }
}
```

Demo 1: Webhook Automation in Action

The Scenario:

Add a minor comment to `lambda_handler.py`.

- Commit and push the changes to our GitHub repository.
- Observe the Jenkins dashboard: a new build automatically commences.

Verify the Jenkins logs show **"Started by GitHub push"**.

Behind the Scenes:

GitHub dispatches a webhook event to our Jenkins instance at `jenkin.neriyelreyes.org/github-webhook/`.

- Jenkins receives the notification, detects the code change, and immediately triggers the configured pipeline.
- This automation eliminates manual triggers, streamlining the development workflow.

Demo 2: CSV File Automated Cleanup

The Flow:

Upload `demo_data_messy.csv` to S3

- An S3 event notification automatically triggers the AWS Lambda function.
- The Lambda function cleans the CSV data (trims whitespace, removes duplicates).

The processed file appears in a separate S3.

- This demonstrates a second layer of robust automation, complementing the CI/CD pipeline.

Demo 3: Catching Bugs with Test Stage Validation

This demonstration highlights the critical role of automated testing, ensuring that faulty code is identified and halted before deployment, maintaining system integrity.

1

Simulating a Bug

Intentionally remove a key function (e.g., `process_csv_data`) from `lambda_handler.py` to create a critical error.

2

Automatic Trigger & Failure

Commit and push the "broken" code. Jenkins will automatically trigger, but the **Test stage will FAIL** due to `pytest` not being able to import the missing function.

3

The Safety Net

This failure prevents the deployment of non-functional code, showcasing the effectiveness of the CI/CD pipeline's testing phase as a crucial safety net.

Key Takeaways: The Power of Automation

GitHub Webhook

Enables instant pipeline triggers on every code push for continuous integration.

Jenkins Pipeline

Provides multi-stage automated testing and deployment for reliable releases.

Unit Tests

Crucial for catching bugs early in the **Test** stage, preventing broken code deployment.

S3 Triggers & Lambda

Automates data processing upon file upload, ensuring immediate and clean data availability.

salamat

ਧੰਨਵਾਦ

Any Questions?