|  |  |
| --- | --- |
| Assignment 2 Report | |
|  | |
| DevOpsCS423 | 2020005 - 2020214M. Sajid Ali |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Introduction**

Project Overview:

The assignment aimed at hands-on implementation of Continuous Integration and Continuous Delivery (CI/CD) methodologies, emphasizing the automation of software development processes. The key objective was to establish a robust CI/CD pipeline using popular tools to facilitate efficient software deployment across different environments—Development, Testing, Staging, and Production.

**Challenges Faced During Deployment**

1. Tool Selection:

Identifying and selecting appropriate CI/CD tools compatible with the project's tech stack (ReactJS and NodeJS) posed an initial challenge. The choice required consideration of integration ease and suitability for automating the development-to-deployment process.

1. Environment Configuration:

Configuring and synchronizing dependencies across AWS EC2 instances (Testing and Staging environments) to ensure consistency and functionality. Adapting the environments to the project's requirements required meticulous setup and troubleshooting.

1. Workflow Automation:

Developing automated workflows using GitHub Actions was challenging, involving the creation of intricate workflows triggering deployment on code changes. Configuring these workflows to compile, test, analyze, and deploy code seamlessly was a significant task.

1. Security Configuration:

Setting up secure access controls and managing security groups across instances and repositories presented challenges. Ensuring appropriate access while preventing unauthorized commits to critical branches demanded careful configuration.

1. Error Handling and Notifications:

Managing error handling within the automated workflows and setting up effective email notifications for workflow status updates, especially on deployment success or failure, required attention to detail and debugging.

**Detailed Workflow of the Assignment Tasks**

Task 1: Selection and Deployment of Open-Source Application

* Cloned a simple open-source application from GitHub, utilizing ReactJS and NodeJS technologies.
* Followed the instructions provided in the Readme file to deploy the application on a localhost environment for demonstration purposes.
* Initialized the cloned repository code to a new repository on GitHub and added collaborators, facilitating collaboration with team members.
* Made the repository public and configured branch protection settings to prevent direct commits to the main branch.

Task 2: Creation of EC2 Instances on AWS for Testing and Staging Environment

* Created two distinct EC2 instances on AWS, assigned appropriate names representing the Testing and Staging environments.
* Employed Ubuntu Server 22.04 LTS (HVM) Amazon Machine Images (AMIs) for both instances, ensuring consistency and compatibility.
* Established a common security group shared among instances to streamline rule management and access control.
* Configured essential project dependencies for the React and Node.js application on the respective Testing and Staging environments.
* Adjusted security group rules based on project requirements for optimal functionality and security measures.

Task 3: Automating Application from Development to Deployment

* Developed an automated workflow triggered by pull requests to initiate deployment for QA testing on the AWS Testing server.
* The workflow encompassed sequential steps:
* Building the project: Compiling source code and building the application.
* Executing unit tests: Verifying individual components or functions' functionality.
* Conducting code analysis/linting: Ensuring code quality and adherence to standards.
* Sending notification emails on workflow status: Providing updates to stakeholders on successful or failed deployments.
* Deployed changes from the main branch to the AWS Staging server upon confirmation by QA or merge into the main branch.
* Provided accessible links for team members and clients to access the deployed project on the Staging environment.