Version control (CodeCommit) and continuous development (CI/CD)

Infrastructure as a Service

Date: April 10th, 2024

 Departamento de Electrónica, Sistemas e Informática (DESI)

Cloud Architecture *(Arquitectura en la Nube)*

Mtro. Rodolfo Luthe Ríos

Isaac Gabriel Gómez Flores

703027

Data Science Master

Introduction

The learning objective is to apply practical cases using public cloud services, to achieve this:

Objectives:

* Use a version control client.
* Configure centralized version control service.
* Control versions of a document.
* Implement a continuous deployment DevOps environment.

Practice Objectives:

* Set up a local repository.
* Create a repository on GitHub.
* Create a repository in CodeCommit.
* Set up a continuous delivery environment from GitHub to Elasticbeanstalk

# Theoretical Framework

Continuous integration and continuous delivery (CI/CD) is a set of practices that enables software development teams to deliver high-quality code faster and more reliably. CI/CD involves automating the processes of building, testing, and deploying software, as well as monitoring and improving its performance and security. By using CI/CD, developers can reduce errors, improve collaboration, and accelerate feedback loops [1].

There are many advantages to using CI/CD. Teams can improve their productivity, lower the chances of errors, speed up the delivery of products, and easily revert if problems occur. Furthermore, CI/CD allows for better planning, more efficient testing and monitoring, and can even lead to cost savings due to the shorter time it takes to deliver a product. In general, CI/CD is a valuable tool that can help teams to produce high-quality software more dependably and effectively. [2]

CI/CD tools integrate Git to enable developers to collaborate on code and integrate them into a shared repository. It is a distributed version control system that allows developers to track changes in their code and collaborate with others. Git enables developers to create branches, merge changes, and revert to previous versions of their code. Git also supports remote repositories, which are hosted on servers and can be accessed by multiple users. [3]

There are several AWS services that enable the creation of CI/CD pipeline, such as:

* AWS CodeCommit: A fully managed source control service that hosts secure and scalable Git repositories. CodeCommit integrates with other AWS services, such as CodeBuild, CodeDeploy, CodePipeline, and Lambda, to enable automated and continuous delivery of code changes [4].
* AWS CodeBuild: A fully managed service that compiles, tests, and packages code in a consistent and reproducible environment. CodeBuild can run custom build commands or use predefined build environments based on popular programming languages and frameworks. [5]
* AWS CodeDeploy: A fully managed service that automates the deployment of code to various compute services, such as EC2, ECS, EKS, Lambda, and Fargate. CodeDeploy can perform rolling updates, blue/green deployments, and canary deployments, as well as monitor and rollback deployments in case of errors. CodeDeploy can also integrate with CodeBuild, CodePipeline, and other sources to trigger deployments automatically [6].
* AWS CodePipeline: A fully managed service that orchestrates and automates the stages of a CI/CD pipeline, such as source, build, test, and deploy. CodePipeline can integrate with various AWS and third-party services, such as CodeCommit, CodeBuild, CodeDeploy, GitHub, Jenkins, and CloudFormation, to create customized and flexible pipelines [7].
* AWS Elastic Beanstalk: A service that simplifies the deployment and management of web applications using various platforms, such as Java, .NET, PHP, Ruby, Node.js, Python, and Docker. Elastic Beanstalk automatically handles the provisioning, scaling, load balancing, and monitoring of the underlying resources, while allowing developers to retain full control over the configuration and code [8].

# Architectural diagram

Figure 1. Architectural Diagram of the bucket and IA rekognition service

# Practice Development

The report is based on *Laboratory 2* of the *Laboratory Notebook from Cloud Applications and Services course.* Figures 2 to 4 show the steps required to follow to achieve the objectives. Each process has a flowchart describing the process.

Figure 2 IA Rekognition process.

# Problems and Solutions

# Experiments and Results

Figure 3 AWS IA Rekognition Face Comparison

Figure 4 Celebrity Rekognition using Dwayne Johnson the Rock

Figure 5 Detect Faces Service Example

Figure 6. Label detection example

# Cost analysis

Explain the cost of the implemented solution, justifying the chosen solution based on costs. Should detail monthly and annual costs.

Table 1 presents a cost analysis for a company that

Table 1 Windows Bastion and Oracle Database Cost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Oracle Database and Instance | | | | |
| Service | Hourly Cost | Hours Month | Monthly Cost | Yearly Cost |
| S3 Object | -- | -- | 2.30 | 27.60 |
| IA Rekognition Api Calls |  |  | 0.31 | 3.72 |
| Sub total | | | | 31.32 |
| Total | | | | 31.32 |

# Conclusion

# Bibliography

**There are no sources in the current document.**

A screenshot of a computer program

Description automatically generatedA screenshot of a computer screen

Description automatically generated