Laboratory Practice Report

Version control (CodeCommit) and continuous

deployment (CI/CD)

4/17/2024

 Departamento de Electrónica, Sistemas e Informática (DESI)

Cloud Architecture *(Arquitectura en la Nube)*

Mtro. Rodolfo Luthe Ríos

Victor Alberto Lopez Cardona

747175

MSC Computer Science

# Introduction

# The report focuses on using real-world examples with public cloud services, particularly in version control and continuous deployment. The main goals are to use a version control client to manage and control document versions, as well as set up a centralized version control service. Additionally, the report aims to implement a continuous deployment DevOps environment to streamline the software delivery process.

# The activities include setting up local repositories, creating repositories on both GitHub and AWS CodeCommit, and controlling document versions. Furthermore, the practice involves setting up a continuous delivery environment, integrating repositories from GitHub to AWS Elastic Beanstalk to automate deployment and improve efficiency. This practice demonstrates the capabilities and benefits of leveraging public cloud services for modern software development and deployment practices.

The practice­ showcases integrating repositorie­s from GitHub to AWS Elastic Beanstalk. This automates deployme­nt, enhancing efficiency. Stre­amlining workflows allows developers to focus on cre­ating value and innovation. Cloud-based service­s handle technical deployme­nt intricacies. The report highlights utilizing public cloud se­rvices' advantages and potential. It improve­s modern software deve­lopment and deployment practice­s, leading to agile, scalable, e­fficient workflows.

# Theoretical Framework

Managing changes to files or documents is handled by version control systems. They track edits, revert to past versions, and allow collaborative work. Git is a popular version control tool for software development. It offers distributed revision control and uses local or remote repositories.

CI (continuous inte­gration) and CD (continuous deployment) are practices for automation. CI automatically tests and merges code changes from multiple contributors. CD extends this by automatically deploying verified updates to production environments. These streamline development cycles and enhance software quality through constant testing and deployment.GitHub serves as a collaborative platform for managing code repositories, enabling programmers to store code and work together seamlessly. It offers tools for checking code changes, discussing code, and tracking issues, making it a cornerstone of modern software development and project management.

# The integration of GitHub with AWS Elastic Beanstalk creates a seamless workflow that allows direct deployment from GitHub code repositories to Elastic Beanstalk environments. AWS Elastic Beanstalk simplifies the deployment and management of applications by providing a Platform as a Service (PaaS) solution. By linking GitHub repositories to Elastic Beanstalk, developers can benefit from automated deployments, taking advantage of monitoring, load balancing, and autoscaling features to optimize performance and resource utilization. This integration streamlines the process of deploying and managing applications, enhancing overall productivity and efficiency.

# Architectural diagram

Diagram of the implemented architecture.

# Practice Development

First I started by creating …

Texto, Chat o mensaje de texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

After the first upload:

Captura de pantalla de un celular

Descripción generada automáticamente

# Problems and Solutions

List all the problems encountered during the development of the practice and how they were resolved.

# Experiments and Results

This section should explain each of the experiments conducted and their results. The results should be properly interpreted to understand their significance and why they are considered good or bad. Diagrams, photographs, or images can be included in this section to help represent the results.

# Cost analysis

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

# Conclusions

The conclusions should be a reflective work presenting the knowledge gained from the experiments, results, and the theoretical framework presented. Invalid conclusions include: "I learned a lot!", "I really liked the practice", "everything worked correctly." (1 or 2 paragraphs)

# Bibliography

In IEEE format.