Laboratory Practice Report

Laboratory 9: Version Control (CodeCommit) and continuous deployment (CI/CD)

April 6th,2024

 Departamento de Electrónica, Sistemas e Informática (DESI)

Cloud Architecture *(Arquitectura en la Nube)*

Mtro. Rodolfo Luthe Ríos

Jonathan Valencia

745959

Msc. Computer Science

# Introduction

In today's rapidly evolving technological landscape, businesses are increasingly embracing cloud-based solutions to streamline their software development processes. This practice aims to demonstrate the utilization of public cloud services for version control and continuous deployment, essential components of modern DevOps practices.

Learning Objectives

* **Use a Version Control Client:** Learn how to manage and track changes to project files using version control software.
* **Configure Centralized Version Control Service:** Set up a centralized repository to store and collaborate on code efficiently.
* **Control Versions of a Document:** Implement versioning for project documentation to track changes and maintain a history of edits.
* **Implement a Continuous Deployment DevOps Environment:** Explore the automation of software delivery processes to achieve rapid and reliable deployments.

General Activities

1. **Set Up a Local Repository:** Begin by establishing a local repository on your development machine to initiate version control.
2. **Create a Repository on GitHub:** Utilize GitHub, a popular cloud-based version control platform, to host and manage your project repository remotely.
3. **Create a Repository in AWS CodeCommit:** Experience AWS's version control service, CodeCommit, to explore cloud-native repository management.
4. **Control Versions of a Document:** Practice versioning by managing changes to a sample document, demonstrating the importance of tracking edits.
5. **Set Up Continuous Delivery from GitHub to Elastic Beanstalk:** Implement a continuous deployment pipeline using AWS Elastic Beanstalk, automating the deployment of your application based on changes committed to GitHub.

Through hands-on exercises and practical demonstrations, we will gain valuable insights into leveraging cloud services for efficient version control and streamlined deployment workflows. By the end of this practice, I will be equipped with essential skills to navigate and leverage public cloud platforms effectively in real-world development scenarios.

# Theoretical Framework

**1. Version Control Concepts**

* **Definition:** Version control is a system that records changes to files over time, allowing multiple contributors to collaborate on a project efficiently.
* **Key Concepts:**
  + **Repositories:** Centralized (e.g., Git, GitHub) vs. Distributed (e.g., Git).
  + **Branching and Merging:** Techniques for managing parallel development efforts.
  + **Commits and History:** Tracking changes and maintaining a chronological history of modifications.
  + **Collaboration:** Enabling multiple team members to work on the same codebase seamlessly.

**2. Public Cloud Services for Version Control**

* **GitHub:**
  + Cloud-based platform for hosting Git repositories.
  + Facilitates collaboration, code review, and integration with CI/CD pipelines.
* **AWS CodeCommit:**
  + Fully managed source control service by AWS.
  + Integrates seamlessly with other AWS services for scalable and secure version control.

**3. Continuous Deployment (CD) and DevOps**

* **Definition:** Continuous Deployment is the automated process of deploying code changes to production environments after passing automated tests.
* **Key Concepts:**
  + **Continuous Integration (CI):** Automating the build and testing of code upon each commit.
  + **Deployment Pipelines:** Defining stages for testing, approval, and deployment.
  + **Infrastructure as Code (IaC):** Treating infrastructure configurations as code for consistency and reproducibility.
  + **Monitoring and Feedback Loops:** Collecting metrics and user feedback to drive iterative improvements.

**4. Integration with Public Cloud Services**

* **GitHub and AWS Elastic Beanstalk:**
  + Leveraging GitHub Actions or AWS CodePipeline to automate deployment to Elastic Beanstalk.
  + Configuration management and scaling of applications in the cloud environment.

**5. DevOps Principles**

* **Automation:** Emphasizing the automation of software development, testing, and deployment processes.
* **Continuous Improvement:** Iterative refinement of workflows based on feedback and performance metrics.
* **Collaboration and Communication:** Fostering a culture of collaboration between development and operations teams.

**6. Benefits and Challenges**

* **Benefits:**
  + Improved code quality and reliability.
  + Faster release cycles and time-to-market.
  + Scalability and flexibility offered by cloud-based services.
* **Challenges:**
  + Adoption of new tools and practices.
  + Ensuring security and compliance in cloud environments.
  + Overcoming organizational resistance to change.

**7. Future Trends and Innovations**

* **Serverless Architectures:** Embracing serverless computing for scalable and cost-effective deployments.
* **Artificial Intelligence (AI) in DevOps:** Leveraging AI for automated testing and optimization.
* **Multi-cloud and Hybrid Deployments:** Managing applications across diverse cloud platforms for resilience and flexibility.

# Architectural diagram

Diagram of the implemented architecture.

\*I placed AWS CodeCommit here as reference but the instructions state it doesn’t work otherwise it would’ve been used instead of github

# A screenshot of a computer Description automatically generated

# Practice Development

# Problems and Solutions

# Experiments and Results

# Cost analysis

# Conclusions

In conclusion

# Bibliography

1. **Version Control Concepts**
   * Loeliger, J., & McCullough, M. (2012). *Version Control with Git: Powerful tools and techniques for collaborative software development*. O'Reilly Media.
2. **Public Cloud Services for Version Control**
   * GitHub. (n.d.). GitHub. [Online]. Available: <https://github.com/>
   * AWS CodeCommit. (n.d.). AWS CodeCommit. [Online]. Available: <https://aws.amazon.com/codecommit/>
3. **Continuous Deployment (CD) and DevOps**
   * Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional.
   * Kim, G., Debois, P., Willis, J., & Humble, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press.
4. **Integration with Public Cloud Services**
   * GitHub Actions. (n.d.). GitHub Actions. [Online]. Available: <https://github.com/features/actions>
   * AWS CodePipeline. (n.d.). AWS CodePipeline. [Online]. Available: <https://aws.amazon.com/codepipeline/>
   * AWS Elastic Beanstalk. (n.d.). AWS Elastic Beanstalk. [Online]. Available: <https://aws.amazon.com/elasticbeanstalk/>
5. **DevOps Principles**
   * Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional.
6. **Benefits and Challenges**
   * Pahl, C., & Jamshidi, P. (Eds.). (2017). *Advanced Software Engineering: Expanding the Frontiers of Software Technology*. Springer.
7. **Future Trends and Innovations**
   * Leitner, P., & Cito, J. (Eds.). (2019). *Microservices, IoT and Azure: Leveraging DevOps and Microservice Architecture to deliver SaaS Solutions*. Apress.