Laboratory Practice Report

Version control (git) and continuous deployment (CI/CD)

# 02/04/2025

Departamento de Electrónica, Sistemas e Informática (DESI)

Cloud Architecture *(Arquitectura en la Nube)*

Mtro. Rodolfo Luthe Ríos

# Daniel García Hernández

# 715989

# Ciencia de datos

*Keywords: pipeline, versión control, entrega continua, DevOps.*

## Introduction

Aplicar usos prácticos de los servicios de la nube pública.

Objetivos:

* Usar un cliente de control de versiones.
* Configurar servicio de control de versiones centralizado.
* Control de versiones de un documento.
* Implementar un ambiente de entrega continua DevOps.

Actividades generales:

* Configurar un repositorio local.
* Crear un repositorio de GitHub.
* Control de versiones de un documento.
* Configurar un ambiente de entrega continua de GitHub a ElasticBeanstalk.

## Theoretical Framework

Como nos menciona Yépez [1] las ventajas ofrecidas por la aequitectura serverless, ha dado lugar a un creciente desarrollo de este tipo de aplicaciones en Cloud Computing, por lo que surge la necesidad de contar con una referencia para la implementación de CI/CD que apoyen este tipo de desarrollo de aplicaciones, y por ende constituyan un aporte al ciclo DevOps.

Dentro del entorno AWS, el desarrollador de aplicaciones sin servidor, puede hacer uso de la combinación de servicios como AWS CodeCommit, AWS CodePipeline, AWS CodeBuild, AWS CodeDeploy y AWS CloudFormation para automatizar la construcción, pruebas y despliegue de aplicaciones serverless que están expresadas en plantillas construidas con AWS SAM (Serverless Application Model).

De igual manera Yépez [1] define la integración continua (CI) y la entrega continua/despliegue continuo (CD) como prácticas de desarrollo de software para producir software en ciclos cortos, en la que la integración de cambios en el código fuente y la actualización de aplicaciones es bastante frecuente. El objetivo final de estas prácticas es reducir los costos, el tiempo y los riesgos mediante la entrega de software en piezas pequeñas.

CI/CD constituye la base para la metodología DevOps, donde desarrolladores y operaciones trabajan de manera colaborativa para asegurar el despliegue ágil y continuo de sofware de calidad, enfocados en agregar mejas continuas al proceso en cada iteración.

Por otro lado, Moreno [2] menciona que una de las partes mas relevantes cuando se esta codificando, si no la que más, es el repositorio de código y el uso que hacemos de este. Tenemos que tener en cuenta las funcionalidades que nos proporciona el sistema de gestión de versiones de código, la herramienta git es open source que a día de hoy es el sistema de control de versiones más usado por la comunidad y el standard de facto en cualquier organización. Además debemos de considerar las características intrínsecas cuando se está definiendo programáticamente la infraestructura.

## Architectural diagram

Diagrama

El contenido generado por IA puede ser incorrecto.

## Practice Development

Version control

Lo primero que se hace es instalar git en la computadora local y configurar nombre y correo electrónico.

Texto

El contenido generado por IA puede ser incorrecto.

Se realizan 3 versiones del archivo versions y se realiza el commit en cada una de ellas.

Texto

El contenido generado por IA puede ser incorrecto.

En la imagen podemos observar las versiones del repositorio y vemos que efectivamente se realizaron los 3 commits.

Después de esto, se crea un nuevo repositorio en github, y dentro de nuestro repositorio local se conecta por medio de Add Hub y se suben nuestros archivos por medio del push.

Texto

El contenido generado por IA puede ser incorrecto.

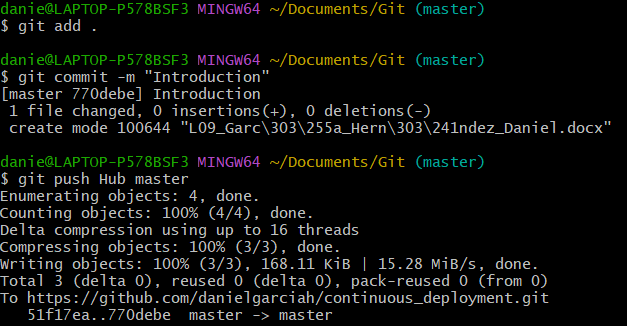
Para corroborar que efectivamente se subieron los cambios a Github abrimos en el navegador el repositorio.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Y observamos que efectivamente tenemos los 3 commits que se realizaron.

Después, para cada sección del presente documento se subirá un commit.



Y observamos en GitHub si nuestro documento se subió correctamente.

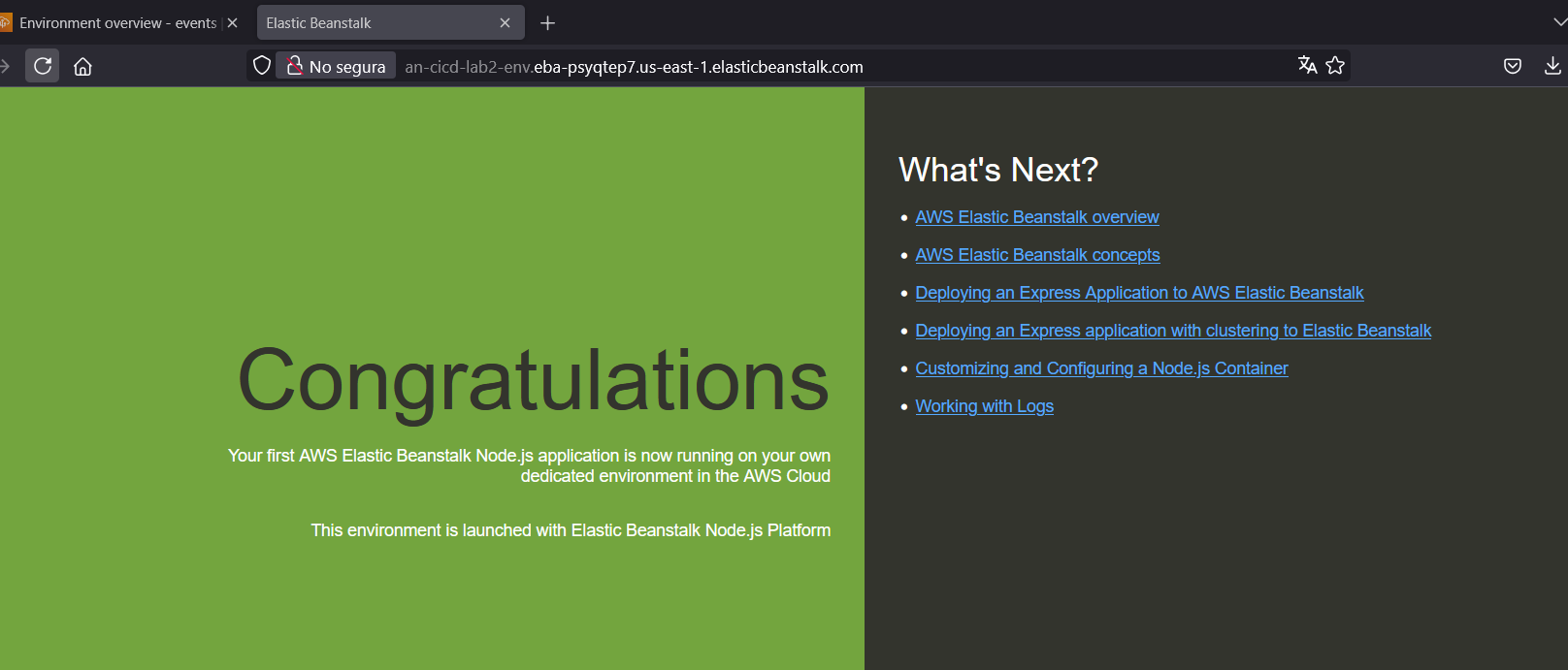
Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Continuous deployment environment

Lo primero que se hace es crear el environment y aplicación dentro de Amazon ElasticBeanstalk.

En donde se crea la aplicación dentro de Nodejs.



En la imagen podemos observar la aplicación de prueba, el cual nos confirma que efectivamente se creó el ambiente y la aplicación de manera correcta.

Después se procede a crear un nuevo repositorio en donde se suben los archivos de la aplicación Nodejs.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Una vez teniendo esto se crea el codepipeline.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Dicho pipeline extrae la información de Github y despliégala nueva aplicación por medio de Elastic Beanstalk.

Comprobamos la aplicación para asegurarnos que se desplegó de manera correcta.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Una vez teniendo el todos nuestros servicios configurados, se edita el código desde github para cambiar el fondo de la aplicación a azul.

Refrescamos la ventana de Pipeline y observamos que se está ejecutando un nuevo deploy.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Volvemos a lanzar la aplicación por medio de beanstalk y observamos que efectivamente se cambió el fondo de pantalla a azul.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

## Problems and Solutions

Al momento de crear el Pipeline no pude conectarme a github por usar la cuenta de estudiante.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Para solucionarlo me tuve que cambiar a mi cuenta personal para poder hacer uso de este servicio.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Experiments and Results

El primer resultado obtenido en el presente laboratorio fue utilizar de manera exitosa la consola de git de manera local, así como subir 3 versiones de un archivo a GitHub.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

En la imagen podemos observar que se subieron de manera exitosa las 3 versiones del archivo de prueba.

El segundo resultado obtenido fue realizar una actualización al código de una aplicación la cual su código está almacenado dentro de GitHub, esto se logró conectando dicho código con Elasticbeanstalk y aplicando las actualizaciones por medio de Pipeline.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

En la imagen podemos observar como se ejecutó de manera correcta el cambio del código en el dominio de nuestra aplicación de Beanstalk.

## Cost analysis

Todos los servicios fueron cotizados dentro de la región US east (N. Virginia).

Code pipeline 1 type V1.

Mensual: 1 USD.

Anual: 12 USD.

Instancia EC2 Linux t3.micro US-East (N. Virginia) On Demand 100% de uso para hostear la aplicación.

Mensual: 7.59 USD.

Anual: 91.08 USD.

## Conclusions

Esta práctica me pareció de una gran utilidad, debido a que el propósito de la misma es muy común y muy usado. El poder desplegar una aplicación, así como sus actualizaciones dentro de la nube, es muy usado, sobretodo por páginas web o aplicaciones. Lo que mas me gustó fue que se linkeara con la github, el cual es uno de los servicios de control de versiones mas utilizado. Por lo tanto, me pareció un caso práctico muy apegado a la realidad y muy utilizable.

## Bibliography

[1] Yépez, C. “Implementación de un ciclo de CI/CD de una API serverless usando AWS Lambda”, trabajo de grado, Universidad Internacional de La Rioja.

[2] Moreno, J. “CI/CD en Infraestructura como código (IaC). Caso real en Amazon Web Services (AWS).”, trabajo de grado, Universitat Oberta de Catalunya, España.