

Reporte de práctica de laboratorio

Control de versiones (CodeCommit)

Marzo 28 de 2020

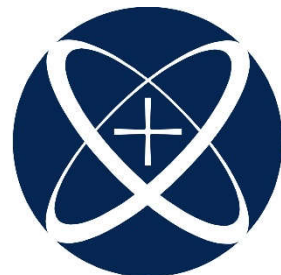
Maestría en Sistemas Computacionales

Aplicaciones y Servicios en la Nube

Prof. Mtro. Rodolfo Luthe Ríos

Alumna: Sarahi Partida Ochoa

ms710662@iteso.mx



ITESO

Universidad Jesuita
de Guadalajara

Introducción

Para esta práctica haremos uso de las herramientas de control de versiones; una de ellas es github, donde solo necesitamos crear una cuenta con nuestro correo electrónico, instalar localmente git donde configuraremos dicho correo para luego hacer los commits que necesitemos, los cuales nos ayudarán a tener la verificación de los cambios que le hemos hecho al proyecto que hayamos compartido; luego usaremos CodeCommit que es parte de AWS, donde también es otra herramienta para control de versiones; para este servicio también haremos uso de git, cuando generemos nuestro proyecto en CodeCommit, el link que se nos genera se lo pasaremos a git para luego sincronizarlo y poder monitorear los cambios que realicemos ya sea a un archivo o a varios del folder que tengamos compartido.

Marco Teórico

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Entre las opciones que tenemos está GitHub el cual crea un entorno que permite almacenar código en un servidor remoto, brinda la capacidad de compartir código con otras personas y facilita que más de una persona agregue, modifique o elimine el código en el mismo archivo y proyecto, manteniendo una fuente de verdad para ese archivo. [1]

Los sistemas de control de versiones (también conocidos como gestión de control de origen o SCM) son software que realiza un seguimiento de cada versión de cada archivo en un proyecto de codificación, una marca de tiempo de cuándo se creó esa versión y el autor de esos cambios. El flujo de trabajo básico de codificación con soporte de sistema de control de versiones es el siguiente:

- Crear un proyecto, generalmente en una carpeta en su computadora.
- Decir al sistema de control de versiones que realice un seguimiento de los cambios del proyecto/carpeta.
- Cada vez que el proyecto esté funcionando, decirle al sistema de control de versiones que lo guardará como la próxima versión.
- Si alguna vez se necesita volver a una versión anterior, se puede solicitar al sistema de control de versiones que vuelva a la versión anterior que se necesite.

Un punto de partida importante para cualquier canalización de CI/CD es un repositorio de control de fuente simple pero funcional. Tradicionalmente, esto se configuraría en uno o más servidores físicos en forma de un repositorio Git o SVN que los desarrolladores usarían para enviar su código y actualizaciones; sin embargo, mantener dichos repositorios de código y escalarlos siempre sería un desafío. AWS CodeCommit es un servicio de control de origen administrado que permite a los desarrolladores almacenar de forma segura su código en la nube de AWS. Ofrece muchas de las características que necesitaría y utilizaría al trabajar con diferentes repositorios de control de código fuente, como ramificaciones, confirmaciones, reversiones y mucho más.[2]

- **AWS CodeBuild:** AWS CodeBuild es un servicio de creación de código que los desarrolladores pueden aprovechar para automatizar sus compilaciones de código fuente, pruebas, ejecuciones y empaque de código para implementaciones. Al igual que sus otros servicios de contraparte en Code Suite, CodeBuild también es administrado completamente por AWS, lo que elimina cualquier sobrecarga administrativa innecesaria, como parchear o escalar el software de creación de código. CodeBuild es altamente extensible y también se integra fácilmente con sus flujos de trabajo de CI/CD existentes.
- **AWS CodeDeploy:** con el código de la aplicación almacenado de forma segura y compilado, el paso final requiere que el código se implemente en una flota de instancias EC2. Usando CodeDeploy, un desarrollador puede automatizar las implementaciones de código en cualquier entorno que se ejecute fuera de las instancias de EC2, así como de los servidores que se ejecutan en un centro de datos local. CodeDeploy esencialmente elimina las complejidades de implementación al permitirle automatizar la entrega de su código en miles de instancias sin tener que pasar por tiempos de inactividad importantes.
- **AWS CodePipeline:** AWS CodePipeline es un servicio completo de CI/CD provisto por AWS que los desarrolladores pueden aprovechar para construir sus pipelines de CI/CD de extremo a extremo, ya sea mediante el uso del conjunto de servicios AWS Code Suite o incluso con otras herramientas populares de terceros, como GitHub, Jenkins, etc. Usando CodePipeline, también puede crear y definir modelos de lanzamiento de software personalizados mediante los cuales su aplicación se actualiza con el último conjunto de actualizaciones, probado y empaquetado para el próximo conjunto iterativo de implementaciones.

Diagrama

En el diagrama se muestran los dos tipos de control de versiones que se utilizaron para la practicas que fue: CodeCommit y GitHub

