Reporte de práctica de laboratorio

Control de versiones (CodeCommit)

4 abril 2020

Maestría en Sistemas Computacionales

*Aplicaciones en la Nube*

Prof. Mtro. Rodolfo Luthe Ríos

Mawrer Amed Ramírez Martínez

ms723737@iteso.mx

# Introducción

Actividades para realizar:

• Configurar un repositorio local

• Crear un repositorio en GitHub

• Crear un repositorio en CodeCommit

• Controlar las versiones de un documento wor.

En esta práctica se utilizará el servicio de CodeCommit de AWS. CodeCommit es un servicio de manejo de versiones administrado, seguro, escalable que aloja repositorios privados de Git.

Se generarán diferentes versiones de un documento y se trabajara con GitHub en un inicio, esto con el objetivo de poder entender el funcionamiento de una herramienta open source y entender las funcionalidades que tiene para después generar las mismas actividades o ejercicios con la herramienta de Amazon en este caso AWS CodeCommit.

# Marco Teórico

Las herramientas de control de versiones permiten registrar los cambios a un archivo o conjunto de archivos a lo largo del tiempo de forma que es posible volver a versiones específicas de manera posterior. El flujo funciona en una variedad de aplicaciones y tipos de archivos.

En un sistema de control de versiones Distribuido (DVCS) como Git, los clientes no solo hacen check-out de la fotografía (snapshot) más reciente, sino que replican el repositorio en su totalidad, incluyendo su historia. De tal forma que, si un servidor se cae y estos sistemas colaboraban por medio de este servidor, cualquier repositorio de los clientes puede ser copiado para restaurar el servidor.

La diferencia más importante entre Git y otros DVCSs es que Git no maneja los datos como una serie de deltas de archivos a lo largo del tiempo, sino como una secuencia de snapshots de un sistema de archivos en miniatura. En Git, cada vez que se realiza un commit se guarda el estado de nuestro proyecto, Git toma la fotografía de cómo se ve nuestro disco en dicho momento y almacena una referencia a dicho snapshot.

Dentro de las herramientas de integración continua Git es una de las mas importantes, la cual de acuerdo con su documentación [1], es un sistema que registra los cambios realizados a un archivo o conjunto de archivos a lo largo del tiempo, de modo que pueden ser recuperadas las versiones especificas mas adelante.

Git utiliza sistemas de control de versiones centralizados (CVCS) el cual permite colaborar con desarrolladores de otros sistemas, tienen un único servidor que contiene todo los archivos versionados y varios clientes que descargan los archivos desde ese lugar central. Por otro lado están los sistemas de control de versiones distribuidos (DVCS) en el cual, los clientes no solo descargan la última copia instantánea de los archivos sino que se replica completamente el repositorio. De esta manera si un servidor deja de funcionar y estos estaban colaborando a través de él, cualquiera de los repositorios disponibles puede ser copiado con el fin de ser restaurado.

Git tiene muchas funcionalidades entre ellas es el hecho de utilizar branches (ramas) en las cuales como lo dice en su documentación [2], las branches es una de las funciones mas poderosas de GIT, es una herramienta muy potente que ayuda para no mezclar líneas diferente de código, se recomienda utilizar branches para nuevos proyectos, funcionalidades, ideas o componentes.

Por su parte Amazon ofrece una herramienta de control de versiones llamada CodeCommit: De acuerdo con la documentación de AWS [3], AWS CodeCommit es servicio de control de versiones generado por Amazon Web Servicies, el cual se puede utilizar para administrar archivos y almacenaje de documentos, código fuentes, archivos binarios, etc.

Con CodeCommit se puede obtener los beneficios de un servicio completamente administrado y alojado por AWS. CodeCommit provee alta disponibilidad y durabilidad del servicio y elimina el overhead de administrar nuestro propio software o hardware. Tambien se puede almacenar código de manera segura. Los repositorios de CC están encriptados tanto en reposo (rest) como en tránsito (transit).

**HTTPS, SSH y HTTPS (GRC)**

Para el protocolo de **HTTPS** es un protocolo el cual, se necesita agregar las credenciales generadas en IAM para poder acceder al repositorio, modificar o crear archivos.

**SSH** es otro protocolo con el cual podemos descargar el repositorio de AWS CodeCommit, en este caso es necesario las configuraciones también de la modificación y el acceso en las políticas adjuntas al usuario de IAM, pero por otro lado también se necesita registrar una llave SSH publica y de esta forma poder hacer las modificaciones pertinentes al repositorio en cuestión.

**HTTPS** (GRC) es otro protocolo que permite conectarse con AWS CodeCommit a través de git-remote-codecommit por sus siglas (GRC). Esta es una utilería que provee un método simple para hacer push y pull del código desde los repositorios de AWS CodeCommit. Este es un método recomendado para generar conexiones de soporte hechas para accesos federados, para entidades o credenciales temporales.