

# Mirror repository en GitLab.com

*12 de Diciembre de 2016 a las 18:08*

Soy usuario de [bitbucket](#) desde hace un tiempo y quería probar *GitLab.com*, copiando algunos repositorio de un servicio al otro para ver como funcionaba, pero quería seguir utilizando Bitbucket. Por diversos motivos quería utilizar los repositorios en los dos servicio y necesitaba una forma de replicarlos los cambios entre los dos servicios de una forma cómoda.

Para esto, existe una opción en *GitLab* llamada *mirror repository* que permite replicar los cambios(commit,ramas y etiquetas) de un servicio a otro servicio.

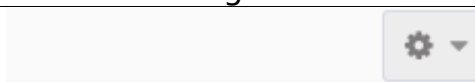
Por ejemplo podemos replicar los cambios de un repositorio en GitLab a otro en *Bitbuckets* y viceversa, incluso podemos encadenar repositorios de varios servicios, por ejemplo, *Bitbuckets*→*GitLab*→*GitHub*. Esta característica, evita tener que realizar varios push a esos repositorios desde nuestro repositorio local, también facilita el trabajar con varios servicios de este tipo.

## Activar y configurar mirror repository

Como ejemplo tenemos un repositorio en *Bitbuckets* y queremos tener el mismo repositorio en *GitLab*. Podemos empezar utilizando la opción de importación de GitLab con el repositorio en Bitbucket, y despues activar la opción *mirror repository*.

Como requisito para utilizar la opción de *mirror*, los repositorios deben ser públicos porque en caso negativo la replicación no funcionara.

Accedemos con una cuenta en *GitLab.com* a nuestro proyecto creado a partir del repositorio de Bitbucket, en la pagina principal del proyecto tenemos la opción de configuración arriba a la derecha, pulsamos en ella y despliega una lista de opciones.



Members

Groups

Deploy Keys

Webhooks

Services

Protected Branches

Runners

Variables

Triggers

CI/CD Pipelines

Push Rules

Mirror Repository

Pages

Audit Events

Edit Project

En la lista desplegable veremos la opción de *Mirror Repository*. Pulsamos y aparecerá la configuración de esta opción.

Project Activity Repository Pipelines Registry Graphs Issues Merge Requests Wiki

**Pull from a remote repository**  
Set up your project to automatically have its branches, tags, and commits updated from an upstream repository every hour.

**Set up mirror repository**  
 Mirror repository  
Automatically update this project's branches, tags, and commits from the upstream repository every hour.

Git repository URL

- The repository must be accessible over `http://`, `https://` or `git://`.
- If your HTTP repository is not publicly accessible, add authentication information to the URL:  
`https://username:password@gitlab.company.com/group/project.git`.
- The update action will time out after 10 minutes. For big repositories, use a clone/push combination.

Mirror user  
Jose Jimenez

This user will be the author of all events in the activity feed that are the result of an update, like new branches being created or new commits being pushed to existing branches. They need to have at least master access to this project.

Trigger builds for mirror updates  
Trigger builds when branches or tags are updated from the upstream repository. Depending on the activity of the upstream repository, this may greatly increase the load on your CI runners. Only enable this if you know they can handle the load.

**Push to a remote repository**  
Set up the remote repository that you want to update with the content of the current repository every hour.

Remote mirror repository  
Automatically update the remote mirror's branches, tags, and commits from this repository every hour.

Git repository URL

- The repository must be accessible over `http://`, `https://` or `git://`.
- If your HTTP repository is not publicly accessible, add authentication information to the URL:  
`https://username:password@gitlab.company.com/group/project.git`.
- The update action will time out after 10 minutes. For big repositories, use a clone/push combination.

Blog de J.A. Jimenez Toro, [rooteando.com](https://rooteando.com)

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Tenemos dos secciones:

- *Pull from a remote repository*: Configurara el repositorio para recibir los cambios de otro repositorio. En este caso, *Bitbucket*->*GitLab*.
- *Remote mirror repository*: Configurara el repositorio para enviar los cambios a otro repositorio, En este caso, *GitLab*->*Bitbucket*.

En las dos secciones se debe especificar la dirección del repositorio (*Git repository URL*), la dirección del repositorio en *Bitbucket*.

Si se ha utilizado opción de importar para crear el proyecto, veremos que en la sección de *Pull* se ha insertado la dirección del repositorio en *Bitbucket*. En esta sección podemos escoger con que usuario sera el autor de los eventos producidos, muestra una lista con los usuario disponibles.

Por último, en la sección (*Pull*) podemos crear *Trigger*( activadores) que permiten crear acciones que se ejecutan cuando se realiza un mirror, hay que tener cuidado porque un *trigger* puede generar mucha carga al servidor.

Dependiendo de lo queremos hacer debemos marcar la opción correspondiente *Mirror repository* / *Remote mirror repository* y pulsar en *Update*.

La acción de *mirror* esta programa para ejecutarse cada hora, existe un botón *Update Now* para ejecutar la acción cuando queramos.