

«Cómo configurar un servidor Git privado en un VPS» por Brian Rogers

Introducción

Este tutorial le mostrará cómo configurar un servidor Git completamente funcional utilizando claves SSH para la autenticación. No tendrá una interfaz web, esto sólo cubrirá la instalación de Git y la configuración de su acceso a ella. Aquí usaremos el anfitrión "git.droplet.com" en lugar del dominio que usará para su VPS.

- This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).
- **English language:** This article is a translation from English into Spanish, published under license [«Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)»](https://creativecommons.org/licenses/by-nc-sa/4.0/), written by Brian Rogers, published on line by the company for leasing virtual machines [DigitalOcean](https://www.digitalocean.com/). The title is [«How To Set Up a Private Git Server on a VPS»](https://www.digitalocean.com/articles/how-to-set-up-a-private-git-server-on-a-vps) and [we](https://www.digitalocean.com/)

[created a copy at Wayback Machine for prevent in future a broken link](#). This work is licensed under the mentioned license but, of course, in castilian language (AKA *spanish*): « [Atribución - NoComercial - Compartirlgual 4.0 Internacional \(CC BY-NC-SA 4.0\)](#) ».

•

- Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-Compartirlgual 4.0 Internacional](#).
- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «[Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](#) » escrito por Brian Rogers, publicado en línea por la empresa de alojamiento de máquinas virtuales [DigitalOcean](#). El título original en idioma inglés es «[How To Set Up a Private Git Server on a VPS](#)» y [hemos creado una copia en Wayback Machine](#) para prevenir un posible enlace roto a futuro.

Nota del traductor:

los comandos han sido adaptados al ambiente Debian/Ubuntu.

Esta puede ser una gran opción si usted quiere mantener su código de manera privado mientras trabaja en él. Mientras que el *código abierto* tiende a ser el *status quo*, hay momentos en los que no se quiere tener el código disponible gratuitamente. Un ejemplo sería si está desarrollando una aplicación móvil, especialmente una de pago. ***Tenga en cuenta que este código puede ser leído por cualquier persona que conozca la dirección URL y con ella clonar el código, pero sólo si la conoce.***

Hay una gran preocupación para muchos y esta es la interfaz web para sus repositorios. GitHub logra esto increíblemente bien. Hay aplicaciones que puedes instalar como [Gitis](#), [GitList](#) y [Goblet](#). No los revisaremos en este tutorial, pero si usted depende en gran medida de una interfaz

gráfica, es posible que quiera revisarlos y pensar en instalar uno de ellos tan pronto como termine de instalar su servidor Git privado.

Creando el par de Llaves SSH

Primero, [necesitamos generar un par de claves SSH](#). Si usted utiliza el sistema operativo Mac o GNU/Linux, puede emitir el siguiente comando en un terminal, pero sustituya la dirección de correo electrónico por la suya propia:

```
ssh-keygen -C "youremail@mailprovider.com" Generating public/private rsa
key pair. Enter file in which to save the key (/home/flynn/.ssh/id_rsa)
: Enter passphrase (empty for no passphrase): Enter same passphrase aga
in: Your identification has been saved in foo_rsa. Your public key has
been saved in foo_rsa.pub. The key fingerprint is: ab:cd:ef:01:23:45:6
7:89:0a:bc:de:f0:12:34:56:78 flynn@en.com The key's randomart image is:
+--[ RSA 2048]-----+ |   o+--+ ..   | | E o           | |   . ++.
o..   | |   o o H .   | |   . .   =   | |   . =o.o=       |
| o .   | |   .   | |   = o .   | |   +-----+
-----+

```

Recomiendo encarecidamente [establecer una contraseña en los ficheros clave](#), lo cual agrega una capa más de seguridad y tiene un impacto muy pequeño en el cálculo numérico. Si está utilizando un sistema operativo basado en Windows, existen herramientas disponibles para generar pares de claves, como [PuTTY Gen](#), aunque viene con una cláusula de exención de responsabilidad que debe consultar con sus leyes locales antes de utilizarlo, ya que algunos países han prohibido su uso. Si ese no es el caso, puede iniciar sesión en su VPS, crear el par de claves y descargar **id_rsa** e **id_rsa.pub** para su uso.

A continuación, el VPS necesitará un usuario específico para Git. La mayoría de la gente simplemente creará un usuario llamado "Git", y eso es lo que haremos para este tutorial, pero siéntase libre de nombrar a este usuario como usted quiera.

Configurar a un usuario Git e instalar Git en su VPS

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

Inicie sesión en su VPS y logre acceso como *usuario raíz* o *root*.*

```
su -
```

O, en el caso de Ubuntu:

```
sudo su
```

*Nota: muchas personas se sienten incómodas trabajando directamente como usuario raíz o root, si su VPS está configurado para tener usuarios que utilicen **sudo** ([lo cual también es llamado *sudoers*](#), una contracción de **sudo** y **users**) puede trabajar de esa manera, anteponiendo el comando **sudo** a los comandos que utilizaremos.

Agregue el usuario en Unix o Linux llamado **git** (no necesariamente tiene que ser ese nombre, el parámetro **-m** creará una carpeta *home* para el usuario **git**):

```
useradd -m git
```

Entonces establezca una contraseña para el usuario **git**:

```
passwd git
```

Ahora instalar Git es tan fácil como:

- CentOS/Fedora: yum install git

- Debian/Ubuntu: apt-get install git

Agregar su llave SSH a la lista de acceso

En este punto necesitamos estar conectado como usuario de Git, el usuario **git**. Si aún no ha iniciado sesión en ese usuario, utilice este comando para cambiar a él:

```
su git
```

Ahora usted necesita subir su archivo generado llamado **id_rsa.pub** al directorio raíz de tu usuario de Git, el usuario **git**. Una vez hecho esto, necesitamos que el demonio SSH sepa qué claves SSH aceptar. Esto se hace usando el archivo de claves autorizadas llamado **authorized_keys**, y que reside en la carpeta de puntos "ssh". Para crear esto, introduzca:

```
mkdir ~/.ssh && touch ~/.ssh/authorized_keys
```

Nota: Usando el doble **et '&'** en sus cadenas de comando, indica al sistema que ejecute el primer comando y luego el segundo. Usando la *virgulilla* al principio de la vía le indicará al sistema que use su directorio *home*, así que '~' se convierte en **/home/git/** para su VPS. El comando **touch** creará un archivo llamado **authorized_keys** con longitud de cero bytes.

A continuación abriremos en nuestra máquina local una nueva ventana terminal de comandos. Vamos a usar el comando **cat**, que tomará el contenido del archivo **id_rsa.pub** (nuestra llave pública) y lo devolverá a la línea de comandos. Luego usamos el modificador '>>' para hacer algo distinto con esa salida en lugar de simplemente imprimirla en su consola. Tenga cuidado con esto, ya que un solo ">" sobrescribirá todo el contenido del segundo archivo que especifique. Un doble ">" lo agregará al final del fichero, así que asegúrese de conocer bien lo que quiere hacer; en la mayoría de los casos será más fácil usar ">>" para que siempre pueda borrar lo que añade en lugar de buscar restaurar lo que ha sobrescrito.

Cada línea de este archivo es una entrada para una clave a la que desea tener acceso a esta

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

cuenta. Para añadir la clave que acaba de cargar, escriba lo siguiente, sustituyendo -textos en color rojo- su nombre de usuario y la dirección IP por la dirección IP de su máquina local (su contraseña de usuario **git** que acabamos de crear le será solicitada):

```
cat ~/.ssh/id_rsa.pub | ssh git@git.droplet.com  
"cat >> ~/.ssh/authorized_keys"
```

Ahora vuelva a la ventana donde tiene la sesión abierta del usuario **git** y usted podrá ver la llave allí si utiliza el comando **cat** en el archivo de llaves autorizadas:

```
cat ~/.ssh/authorized_keys
```

Si desea añadir a otras personas a su lista de acceso, ellos o ellas solo tienen que darle sus respectivas clave **id_rsa.pub** y usted las añadirá al archivo de claves autorizadas.

Configurar un repositorio nuevo

(Nota del traductor: en [este blog tenemos un tema](#) donde se explica en detalle el trabajar con repositorios en Git).

Este es un proceso bastante simple, sólo tiene usted que llamar al comando Git e inicializar un repositorio inicial en el directorio que quiera. Digamos que quiero usar "Mi proyecto" como título del proyecto. Al crear la carpeta, usaría todas las minúsculas, reemplazaría cualquier espacio con guiones y agregaría ".git" al nombre. Así, "Mi proyecto" se convierte en "mi-proyecto.git".

Estando conectados al VPS con sesión abierta del usuario **git**:

```
git init --bare mi-proyecto.git
```

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

¡Y eso es todo! Ahora tienes un repositorio Git configurado en tu VPS. Pasemos a cómo utilizarlo con su ordenador local.

Usando el servidor Git desde su máquina local

En GNU/Linux o Mac OS, debemos cambiar el "*origin*" remoto a su servidor recién creado. Si ya dispone de un repositorio local que desea enviar al servidor, cambie el "*origin*" remoto utilizando este comando:

```
git remote set-url origin git@git.droplet.com:mi-proyecto.git
```

Atención: el nombre por defecto del servidor que aloja cualquier repositorios es "***origin***" por lo que es buena idea asignar un nombre distinto que permita identificarlo claramente.

Esto se logra con el comando «git remote rename nombre_viejo nombre_nuevo»

Consejo del traductor

Si este es un nuevo repositorio el que está configurando, utilice esto:

```
git init && git remote add origin git@git.droplet.com:mi-proyecto.git
```

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

Ahora usted puede agregar, subir las modificaciones, extraer versiones específicas e incluso clonar todo el proyecto en otra máquina, sabiendo que su código solamente es accesible para usted.

Pero, ¿qué sucede si desea que algunas personas de confianza tengan acceso a este servidor y desea simplificar las cosas ordenándolas por los nombres de sus usuarios? Una manera simple y efectiva de hacerlo es crear una carpeta con el nombre de cada persona, por lo que en la carpeta de inicio de su lista de usuarios de Git, ingrese:

```
mkdir usuario1 usuario2
```

Ahora, cuando especifique el repositorio remoto, se vería así:

```
git remote add origin git@git.droplet.com:usuario1/proyecto-usuario1.git
```


- This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#).
- **English language:** This article is a translation from English into Spanish, published under license «[Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](#) », written by Brian Rogers, published on line by the company for leasing virtual machines [DigitalOcean](#). The title is «[How To Set Up a Private Git Server on a VPS](#)» and [we created a copy at Wayback Machine for prevent in future a broken link](#). This work is licensed under the mentioned license but, of course, in castilian language (AKA *spanish*): «[Atribución - NoComercial - Compartirlgual 4.0 Internacional \(CC BY-NC-SA 4.0\)](#) ».

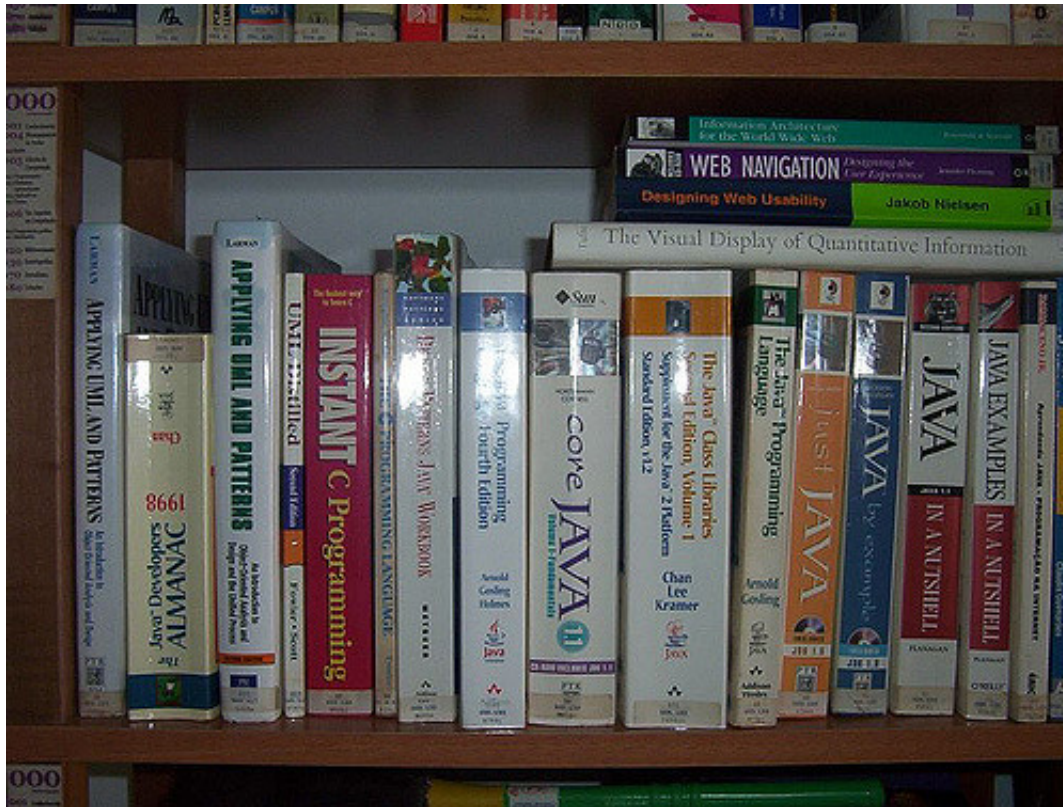
-

- Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-Compartirlgual 4.0 Internacional](#).
- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «[Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](#) » escrito por Brian Rogers, publicado en línea por la empresa de alojamiento de máquinas virtuales [DigitalOcean](#). El título original en idioma inglés es «[How To Set Up a Private Git Server on a VPS](#)» y [hemos creado una copia en Wayback Machine](#) para prevenir un posible enlace roto a futuro.

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>



Fuentes consultadas

En idioma castellano

- [«4.1 Git en el Servidor - Los Protocolos»](#).
- [«Añadir permisos de ejecución a un script en Linux»](#) por davidochobits.

En idioma francés

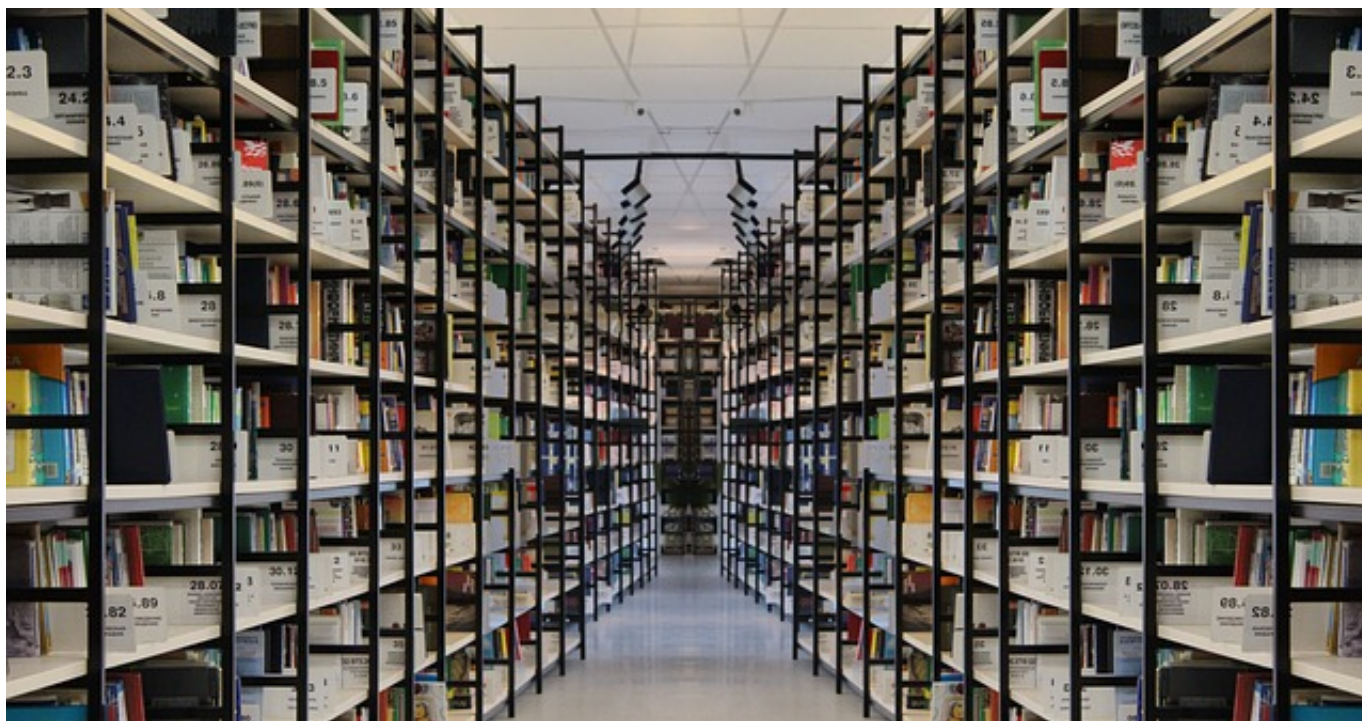
- « ».
- « ».
- « ».

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.
<https://www.ks7000.net.ve>

En idioma inglés

- «[How to Add and Delete Users on Ubuntu 18.04](#)».
- «[How to Create Users in Linux Using the "useradd" Command](#)».
- «[create home directories after create users](#)».
- «[How to Add and Delete Users on Ubuntu 18.04](#)».
- «[How to Run Your Own Git Server](#)» by Swapnil Bhartiya».
- «[Teaching How to push your code into multiple Remote Git repositories](#)».
- «[Testing your SSH connection](#)» with GitHub.
- «[Error: Permission denied \(publickey\)](#)» at GitHub.



Crédito de la imagen [Gerd Altmann](#), [trabajo](#), licencia de uso: [Pixabay](#)

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>
