

9 septiembre, 2014 • por Jan Sanchez





Las claves SSH también conocidas como llaves SSH, son una manera de identificar las computadoras de confianza, sin tener que ingresar una contraseña.

Las claves SSH se deben generar para cada usuario. Luego de realizar la generación obtendremos una clave privada y una clave pública.

En este artículo te mostraremos como generar claves SSH para tu usuario, siguiendo estos sencillos pasos:

### VALIDAMOS SI TENEMOS UNA CLAVE SSH EXISTENTE

Para ello escribimos lo siguiente en nuestro terminal:

```
ls -al ~/.ssh | grep ".pub"
```

Si ya tenemos una clave SSH nos aparecerá algo similar a esto:

```
-rw-r--r-- 1 frontend frontend 399 Aug 22 17:04 id_rsa.pub
```

Pero si aún no la hemos generado entonces nos pueden salir este mensaje:

```
ls: cannot access /home/frontend/.ssh: No such file or directory
```

Sí, nos puede salir vacío o en blanco, ya que puede que tengamos la carpeta .ssh pero no haya ningún archivo dentro de esta.

### **GENERANDO UNA NUEVA CLAVE SSH**

Si aún no tenemos una clave .ssh entonces la procederemos a generar escribiendo en nuestro terminal lo siguiente:

Claro, colocando tu correo electrónico en vez de tu\_correo@tu\_dominio.com .

```
ssh-keygen -t rsa -C "tu_correo@tu_dominio.com"
```

De preferencia dejemoslo con los ajustes por defecto, por lo que cuando nos haga las 3 preguntas, simplemente pulsamos Enter.

Luego del proceso nos quedará algo parecido a esto:

Esto quiere decir que ya generamos nuestras claves SSH correctamente. Como podemos apreciar, la clave privada a sido guardada en:

```
/home/frontend/.ssh/id_rsa y la clave pública en /home/frontend/.ssh/id_rsa.pub .
```

Entonces ya podemos agregar nuestra nueva clave SSH al ssh-agent, escribiendo lo siguiente en nuestro terminal:

```
eval "$(ssh-agent -s)" && ssh-add ~/.ssh/id_rsa
```

Con esto hemos agregado nuestra clave al pequeño programa ssh-agent para que administre nuestras llaves por nosotros. Lo que quiere decir es que tu sólo ingresaras tu clave una sola vez, y después de eso, ssh-agent mantendrá la clave en memoria y cada vez que sea requerida ssh-agent la entregará por nosotros y si todo nos fue bien nos saldrá algo como esto:

```
Agent pid 6026

Identity added: /home/frontend/.ssh/id_rsa (/home/frontend/.ssh/id_rsa)
```

### AGREGANDO NUESTRA CLAVE PÚBLICA A GITHUB

Primero instalamos xclip, para poder copiar el contenido de nuestra clave pública por consola.

sudo apt-get install xclip

Luego seleccionamos y copiamos el contenido de nuestra clave pública con la siguiente línea de código:

xclip -sel clip < ~/.ssh/id\_rsa.pub</pre>

Con esto tenemos nuestra clave pública en el portapapeles, entonces ingresamos a Github, y En la sección Settings elegimos la opción SSH Keys e ingresamos.

Hacemos click en el botón Add SSH Key (Agrega tu clave SSH).

En el campo **Title** escribimos algo descriptivo sobre esta clave y en el campo **Key** pegamos el contenido del archivo ~/.ssh/id\_rsa.pub (contenido que tenemos en nuestro portapapeles listo para pegar).

Y luego hacemos click en el bóton Add Key , nos pedirá nuestra contraseña de Github para validar y listo.

### VALIDANDO NUESTRA CLAVE SSH CON GITHUB

Para comprobar que todo anda bien, simplemente escribimos lo siguiente en nuestra terminal

ssh -T git@github.com

Si todo anda bien, debemos obtener lo siguiente:

Warning: Permanently added the RSA host key for IP address '178.10.123.50' to the list of known hosts.

Hi jansanchez! You've successfully authenticated.

Con esto ya tenemos registrada nuestra clave pública en github, lo cual nos permitirá subir cambios al servidor de Github creando una conexión segura.

Si deseas realizar la configuración básica de git te recomendamos leer el siguiente artículo:

Comandos básicos Git.

# EXTRA: CÓMO "INSTALAR" TU CLAVE PÚBLICA EN UNA MAQUINA REMOTA

Para no estar poniendo siempre la clave del usuario de la pc a la que nos queremos conectar por ssh dejamos esta linea de código para que cuando te intentes conectar por ssh a una pc nunca más te pida la clave del usuario ya que la pc remota sabra que eres un usuario autorizado gracias a tu clave publica.

Bueno eso es todo, espero que les haya servido de algo este pequeño artículo, no se olviden de enviar sus comentarios, compartir los posts en las redes sociales, Gracias.

Twittear me gusta 8 Compartir 8

#### RELACIONADO

Git: Comandos básicos que no debes dejar de usar 30 junio, 2014 En "Git" Tip: Ignorar un archivo en git que ya ha sido commiteado antes 13 noviembre, 2014 En "Git" Aplicar un commit a otra rama con git cherry-pick 13 octubre, 2015 En "Git"



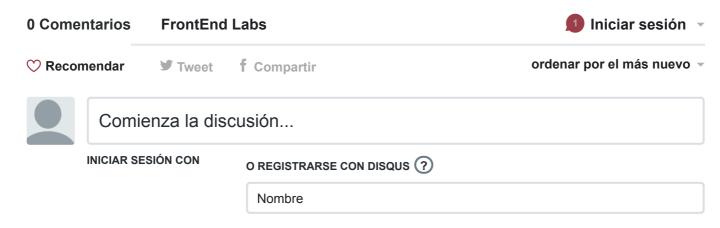
Tags: Debian, Git, Github, Linux, Ubuntu /

Category: Git

**SHARE ON** 

**TWITTER** 

**FACEBOOK** 

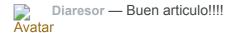


Sé el primero en comentar.

#### TAMBIÉN EN FRONTEND LABS

## ¿Qué es YARN? y ¿Cómo usar este administrador de dependencias

4 comentarios • hace 2 años



# Aplicar un commit a otra rama con git cherry-pick

2 comentarios • hace 3 años

Mario — Si da conflictos si haces esa

Avataropereación, tal vez ya no se necesite hacer
merge

# ¿Cómo clonar un repositorio git en pocos segundos?

2 comentarios • hace 3 años

Renán Romero Ruiz — Hola, esto lo puedes
Avatarefectuar en el cmd solo tienes que bajarte la
versión git para windows https://git-

# Tip: Ignorar un archivo en git que ya ha sido commiteado antes

5 comentarios • hace 3 años

Daniel Espinosa — Hola, Muy interesante

Avatareste aporte. Tengo una pregunta. Que hacer
cuando el commit ya ha sido enviado a github

### **ENTRADAS RECIENTES**

Buscar ... Q

Soy un Desarrollador Frontend ¿Cómo me mantengo actualizado con las últimas tecnologías?

¿Por qué debería aprender Fastlane?

¿Qué es YARN? y ¿Cómo usar este administrador de dependencias

Talleres Web React JS – Taller 1 – Demo básica con React JS

¿Cómo crear un API REST falso en unos cuantos minutos con json-server?

#### **DONAR**

#### **COMENTARIOS RECIENTES**

Ayúdanos a seguir compartiendo más contenido!

Funciones Flecha (Arrow)
Fernando González Frías en La importancia del comando "Git stash" en un proyecto
PipBoy en Aprendiendo ECMAScript 6:
Funciones Flecha (Arrow)
Anders Angel en React js en español – tutorial básico y primeros pasos
Josue Mente en Instalar un ipa en un iPhone o iPad

Javier Moran en Aprendiendo ECMAScript 6:

Copyright © 2019 — Frontend Labs.