

# Usuarios y permisos

## 1. ¿Cómo manejo los usuarios de mi VPS?

### 1.1. Modificar la clave de mi usuario

Lo primero que podemos hacer es cambiar la contraseña de nuestro propio usuario. Para ello, una vez entremos a nuestro servidor por SSH, usaremos este comando:

```
$ passwd
```

Si queremos modificar la contraseña de otro usuario, basta con señalarlo luego del comando anterior:

```
$ passwd usuario
```

### 1.2. Agregar un usuario nuevo

Para ello utilizaremos el comando:

```
$ useradd usuario
```

Luego le agregamos una contraseña:

```
$ passwd usuario
```

Te será solicitada ingresar dos veces la contraseña.

### 1.3. Cambiar de usuario

Supongamos que accedemos al servidor vía SSH con el usuario *guru* y queremos cambiar a otro usuario que hayamos creado previamente. Para ello debemos utilizar:

```
$ sudo su - usuario
```

O podríamos utilizar el siguiente comando y nos será solicitada la contraseña del usuario *usuario*:

```
$ su usuario
```

Para salir del usuario al que acabamos de entrar utilizamos:

```
$ exit
```

## 1.4. Eliminar un usuario

Si queremos eliminar un usuario, tipeamos:

```
$ sudo userdel usuario
```

## 2. Permisos sobre archivos

Para entender cómo Linux maneja los permisos, se deben comprender algunos conceptos:

- **Owner** (dueño): Los permisos de dueño aplican sólo al dueño del archivo o directorio en cuestión. No tendrán influencia sobre las acciones de otros usuarios.
- **Group** (grupo): estos permisos aplicarán sólo al grupo al que han sido asignados los archivos o directorio. No afectarán a las acciones de los demás.
- **All users** (todos): estos permisos aplican al resto de los usuarios del sistema. Debemos tener especial cuidado con este grupo.
- **Read** (leer): se refiere a la capacidad del usuario de leer el contenido del archivo.
- **Write** (escribir): el permiso de escribir hace referencia a la capacidad de escribir o modificar el archivo o directorio específico.
- **Execute** (ejecutar): afecta la capacidad de ejecutar el archivo o ver el contenido de un directorio.

Para ver los permisos de los archivos dentro de un directorio podemos ejecutar:

```
$ ls -l
```

Esto nos dará una lista de archivos y nos presentará los permisos al lado izquierdo:

```
[guru@ubuntu-16:/tmp$ ls -l
total 4
-rw-rw-r-- 1 guru guru    0 Apr 30 13:46 archivo1.txt
-rw-rw-r-- 1 guru guru    0 Apr 30 13:47 archivo2.txt
drwx----- 3 root root 4096 Apr 30 12:55 systemd-private
```

Para el primer archivo la tercera columna nos dice que pertenece al **usuario** *guru* y la cuarta nos dice que pertenece al **grupo** *guru*.

En la primera columna veremos los permisos separados por los guiones, donde el primero se refiere al **dueño** del archivo, luego al **grupo** y luego a **todos** los demás. En este caso concreto (*archivo1.txt*), el usuario *guru* puede escribir y leer, el grupo *guru* puede escribir y leer, y los demás solo pueden leer.

NOTA: si el primer caracter de la sección de permisos es una *d*, nos indica que es un directorio. Si es un guión, nos indica que es un archivo.

## 2.1. Cambiando permisos de archivos y directorios con *chmod* y *chown*

*chmod* funciona de dos maneras. En la primera, 3 números representan los permisos de dueño, grupo y todos los usuarios. Cada número corresponde a una suma de 3 posibles valores: 1 es permiso de ejecutar, 2 es permiso de escribir y 4 de leer.

La segunda manera es utilizando las letras *u* (**u**suario dueño), *g* (**g**ruppo) y *o* (**o**tros), luego un signo de suma o resta (agregar o quitar permiso respectivamente) y el tipo de permiso (**r**ead, **w**rite, **e**xecute).

En el siguiente ejemplo, tanto a usuario, grupo y otros le agregamos (+) los permisos de *rw*x (lectura, escritura y ejecución):

```
$ chmod 777 archivo.txt$ chmod ugo+rw x archivo.txt
```

A continuación otro ejemplo donde asignamos todos los permisos al dueño (u), y al resto escritura y lectura:

```
$ chmod 755 archivo1.txt    # Primer método  
# Segundo método (equivalente)  
$ chmod u=rwx filename$ chmod go=rx filename
```

Para cambiar el owner (dueño) del archivo a *usuario*, tipearemos:

```
$ chown usuario archivo.txt
```

Para cambiar el grupo del archivo a *administradores*:

```
$ chown :administradores archivo1.txt
```

Y para hacerlo de manera simultánea, en un solo comando, usaremos:

```
$ chown usuario:administradores archivo1.txt
```

---

Revision #6

Created 7 years ago by [Admin](#)

Updated 6 years ago by [José Reyes](#)