

Trabajo primera evaluación

Despliegue de aplicaciones web



Noelia Ruiz López
2º DAW
07/12/2020

Apache Virtual Hosting	3
1. Introducción e instalación	3
2. Crear la Estructura del Directorio	3
3. Otorgar Permisos	4
4. Crear Páginas de Prueba para cada Virtual Host	5
5. Creamos un archivo para el virtual host	6
6. Habilitar los Nuevos Archivos Virtual Host	7
7. Configure su Archivo Hosts Local	8
8. Pruebe sus Resultados	9
Apache Mapeo URL	12
1. Opciones de directorios	12
2. Trabajando con alias	13
3. Negociación de contenidos	13
4. Redirecciones	14
5. Páginas de errores personalizadas	14
Introducción a Git	16
1. Comandos iniciales e instalación de git	16
2. Iniciamos git	17
3. Creamos un archivo txt	17
4. Hacemos un git status	18
5. Cambio en el archivo ejercicio.txt	18
6. Creamos otro archivo "ejercicio 2.txt"	19
7. git log --oneline --color	19
8. git rm ejercicio 2.txt	19
9. mover archivos entre carpetas	20
10. Creamos una key ssh	20
11. copiamos en github	21
12. github hacemos un pull	22
Instalar vsftpd	23
1. Instalar Vsftpd	23
2. Permitir el tráfico FTP desde el firewall	24
3. Crear el directorio de usuarios	25
4. Configurar vsftpd	26
5. Hacer que el FTP sea seguro	28
6. Prueba de conexiones con FileZilla	29
Apache control de acceso, autenticación	33
1. Instalar el paquete de utilidades de Apache	33
2. Crear el archivo de contraseña	34
3. Paso 3: Configurar la autenticación de contraseña de Apache	35
4. Confirmar la autenticación con contraseña	39

Apache Virtual Hosting

1. Introducción e instalación

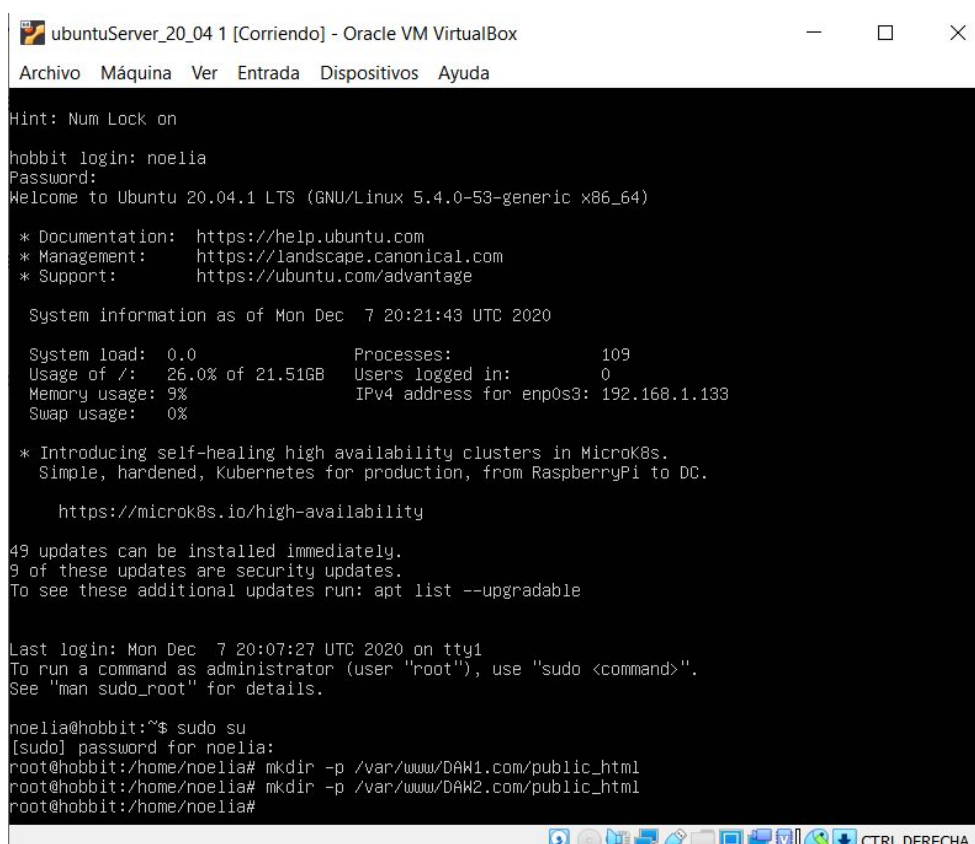
```
sudo apt-get update
sudo apt-get install apache2
```

2. Crear la Estructura del Directorio

El primer paso será crear una estructura de directorios que alojará los datos del sitio que vamos a proporcionar a nuestros visitantes.

Nuestro documento root (ó documento raíz, es el directorio más alto en el que Apache buscará contenido para mostrar) será configurado en directorios individuales bajo el directorio `/var/www/`. Crearemos un directorio aquí para cada uno de los virtual hosts que pretendemos crear.

Dentro de cada uno de *estos* directorios, crearemos una carpeta `public_html` que mantendrá los archivos. Esto nos dará algo de flexibilidad en nuestro hosting.



```
ubuntuServer_20_04 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Hint: Num Lock on
hobbit login: noelia
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Dec  7 20:21:43 UTC 2020

System load:  0.0               Processes:    109
Usage of /:   26.0% of 21.51GB   Users logged in:  0
Memory usage: 9%               IPv4 address for enp0s3: 192.168.1.133
Swap usage:   0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

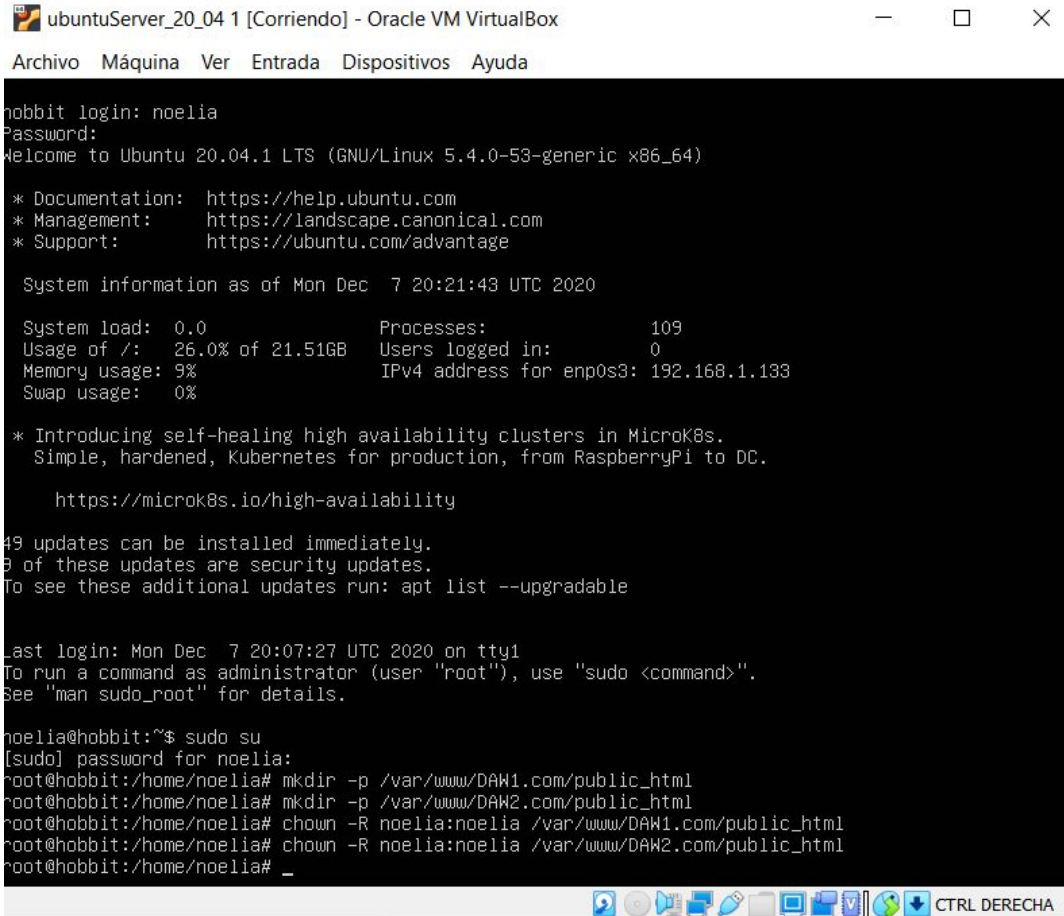
49 updates can be installed immediately.
9 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Mon Dec  7 20:07:27 UTC 2020 on tty1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

noelia@hobbit:~$ sudo su
[sudo] password for noelia:
root@hobbit:/home/noelia# mkdir -p /var/www/DAW1.com/public_html
root@hobbit:/home/noelia# mkdir -p /var/www/DAW2.com/public_html
root@hobbit:/home/noelia#
```

3. Otorgar Permisos

Ahora tenemos la estructura de directorios para nuestros archivos, pero son propiedad de nuestro usuario root. Si queremos que nuestro usuario regular sea capaz de modificar archivos dentro de nuestros directorios web, debemos cambiar la propiedad haciendo lo siguiente:



```
ubuntuServer_20_04 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

noebit login: noelia
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Dec  7 20:21:43 UTC 2020

System load:  0.0               Processes:            109
Usage of /:   26.0% of 21.51GB   Users logged in:     0
Memory usage: 9%               IPv4 address for enp0s3: 192.168.1.133
Swap usage:   0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

9 updates can be installed immediately.
3 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Mon Dec  7 20:07:27 UTC 2020 on tty1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

noelia@hobbit:~$ sudo su
[sudo] password for noelia:
root@hobbit:/home/noelia# mkdir -p /var/www/DAW1.com/public_html
root@hobbit:/home/noelia# mkdir -p /var/www/DAW2.com/public_html
root@hobbit:/home/noelia# chown -R noelia:noelia /var/www/DAW1.com/public_html
root@hobbit:/home/noelia# chown -R noelia:noelia /var/www/DAW2.com/public_html
root@hobbit:/home/noelia# _
```

Debemos además, modificar un poco nuestros permisos para asegurarnos de que el acceso de lectura esté habilitado en el directorio web general y todos los archivos y directorios en él para que todas las páginas puedan ser servidas correctamente:

```
ubuntuServer_20_04 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
hobbit login: noelia
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Dec  7 20:21:43 UTC 2020

System load:  0.0               Processes:            109
Usage of /:   26.0% of 21.51GB  Users logged in:     0
Memory usage: 9%               IPv4 address for enp0s3: 192.168.1.133
Swap usage:   0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

49 updates can be installed immediately.
9 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Mon Dec  7 20:07:27 UTC 2020 on tty1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

noelia@hobbit:~$ sudo su
[sudo] password for noelia:
root@hobbit:/home/noelia# mkdir -p /var/www/DAW1.com/public_html
root@hobbit:/home/noelia# mkdir -p /var/www/DAW2.com/public_html
root@hobbit:/home/noelia# chown -R noelia:noelia /var/www/DAW1.com/public_html
root@hobbit:/home/noelia# chown -R noelia:noelia /var/www/DAW2.com/public_html
root@hobbit:/home/noelia# chmod -R 755 /var/www
root@hobbit:/home/noelia#
```

Su servidor web ahora debe tener los permisos que requiere para servir el contenido, y su usuario deberá ser capaz de crear contenido entre las carpetas necesarias.

4. Crear Páginas de Prueba para cada Virtual Host

Tenemos nuestra propia estructura de directorios en forma. Vamos a crear algo de contenido en los archivos HTML.

-Creamos los dos archivos HTML en sus carpetas correspondientes:

```
-/var/www/DAW1.com/public_html/index.html
```

```
-/var/www/DAW2.com/public_html/index.html
```

5. Creamos un archivo para el virtual host

Copiamos el archivo `/etc/apache2/sites-available/000-default.conf`

y creamos un archivo que se llame `DAW1.com.conf` con el mismo contenido, dejándolo así:

```
GNU nano 4.8 /etc/apache2/sites-available/DAW1.com.conf Modified
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

ServerName DAW1.com
ServerAlias www.DAW1.com
DocumentRoot /var/www/DAW1.com/public_html

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^N Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo
```

y el de DAW2, así:


```
GNU nano 4.8 /etc/apache2/sites-available/DAW2.com.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

ServerName DAW2.com
ServerAlias www.DAW2.com
DocumentRoot /var/www/DAW1.com/public_html

[ Read 36 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File ^N Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo
```

6. Habilitar los Nuevos Archivos Virtual Host

Ahora que hemos creado nuestros archivos de virtual host, debemos habilitarlos.

Apache incluye algunas herramientas que nos permiten hacer esto.

Podemos usar la herramienta a2ensite para habilitar cada uno de nuestros sitios así:

Posteriormente, deshabilite el sitio poder defecto definido en 000-default.conf, Cuando concluyas, deberá reiniciar Apache para hacer que estos cambios sean efectivos

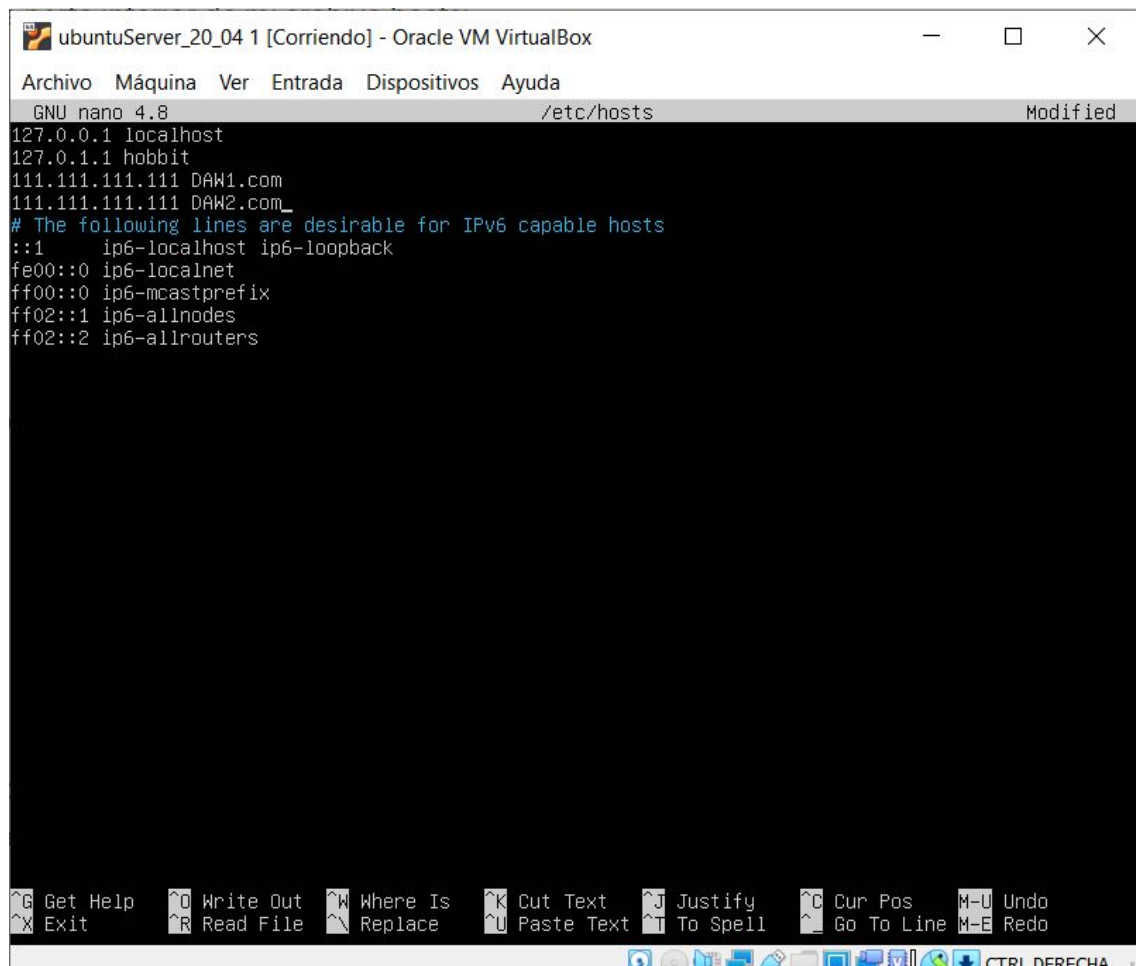
```

-t -D DUMP_RUN_CFG : show parsed run settings
-S                  : a synonym for -t -D DUMP_VHOSTS -D DUMP_RUN_CFG
-t -D DUMP_MODULES : show all loaded modules
-M                  : a synonym for -t -D DUMP_MODULES
-t -D DUMP_INCLUDES: show all included configuration files
-t                  : run syntax check for config files
-T                  : start without DocumentRoot(s) check
-X                  : debug mode (only one worker, do not detach)
noelia@hobbit:~$ sudo a2ensite DAW1.com.conf
Site DAW1.com already enabled
noelia@hobbit:~$ sudo a2ensite DAW2.com.conf
Enabling site DAW2.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
noelia@hobbit:~$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'apache2.service'.
Multiple identities can be used for authentication:
 1. usuario
 2. noelia ruiz,, (noelia)
Choose identity to authenticate as (1-2): 2
Password:
==== AUTHENTICATION COMPLETE ====
noelia@hobbit:~$ sudo a2disssite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
noelia@hobbit:~$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'apache2.service'.
Multiple identities can be used for authentication:
 1. usuario
 2. noelia ruiz,, (noelia)
Choose identity to authenticate as (1-2): 2
Password:
==== AUTHENTICATION COMPLETE ====
noelia@hobbit:~$ _

```

7. Configure su Archivo Hosts Local

Si aún no está utilizando un dominio real para probar estos procedimientos y ha utilizado un dominio ejemplo para ello, entonces puede al menos probar la funcionalidad de este proceso modificando temporalmente el archivo hosts en su computadora local.

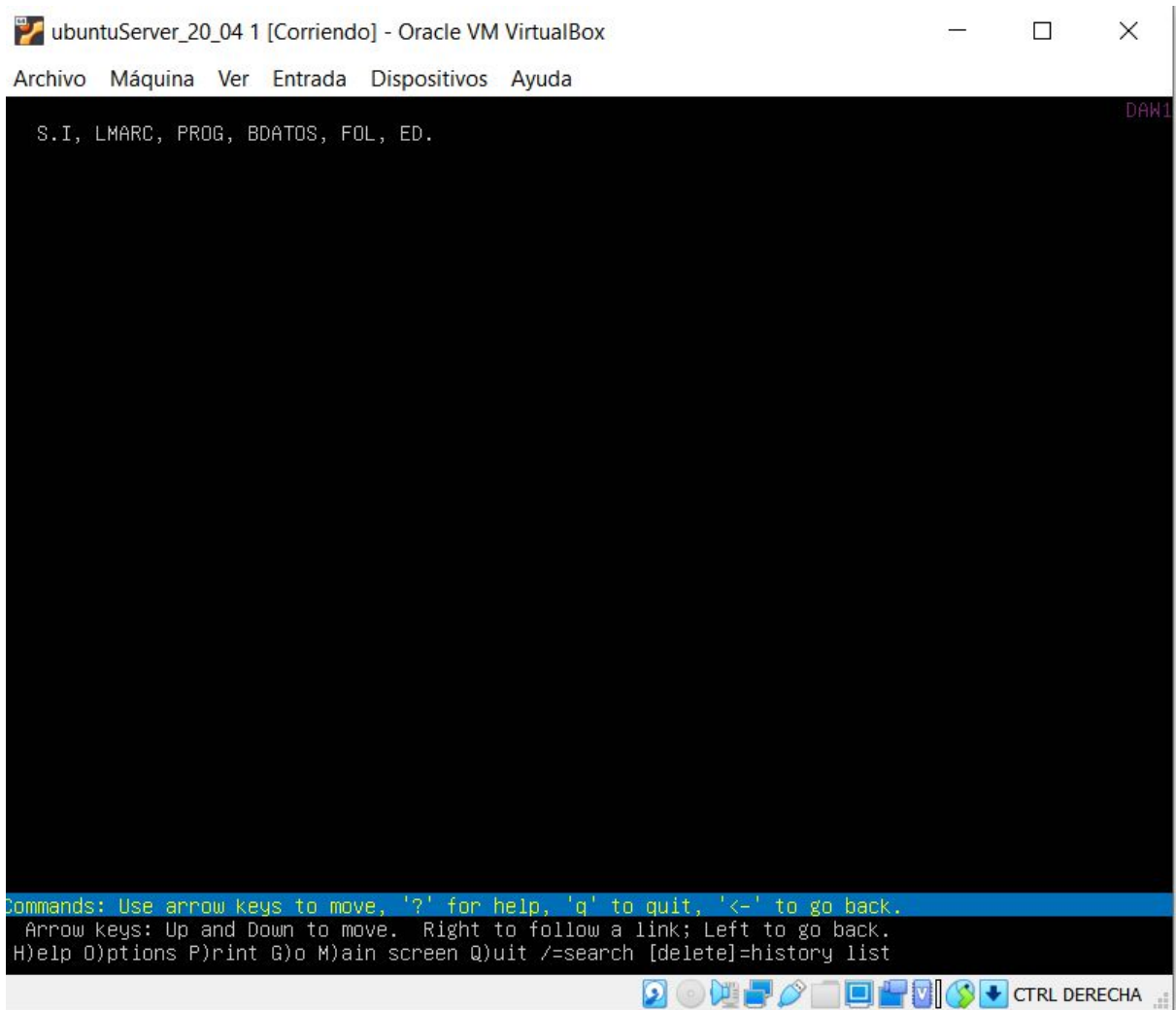


```
ubuntuServer_20_04 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/hosts Modified
127.0.0.1 localhost
127.0.1.1 hobbit
111.111.111.111 DAW1.com
111.111.111.111 DAW2.com
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

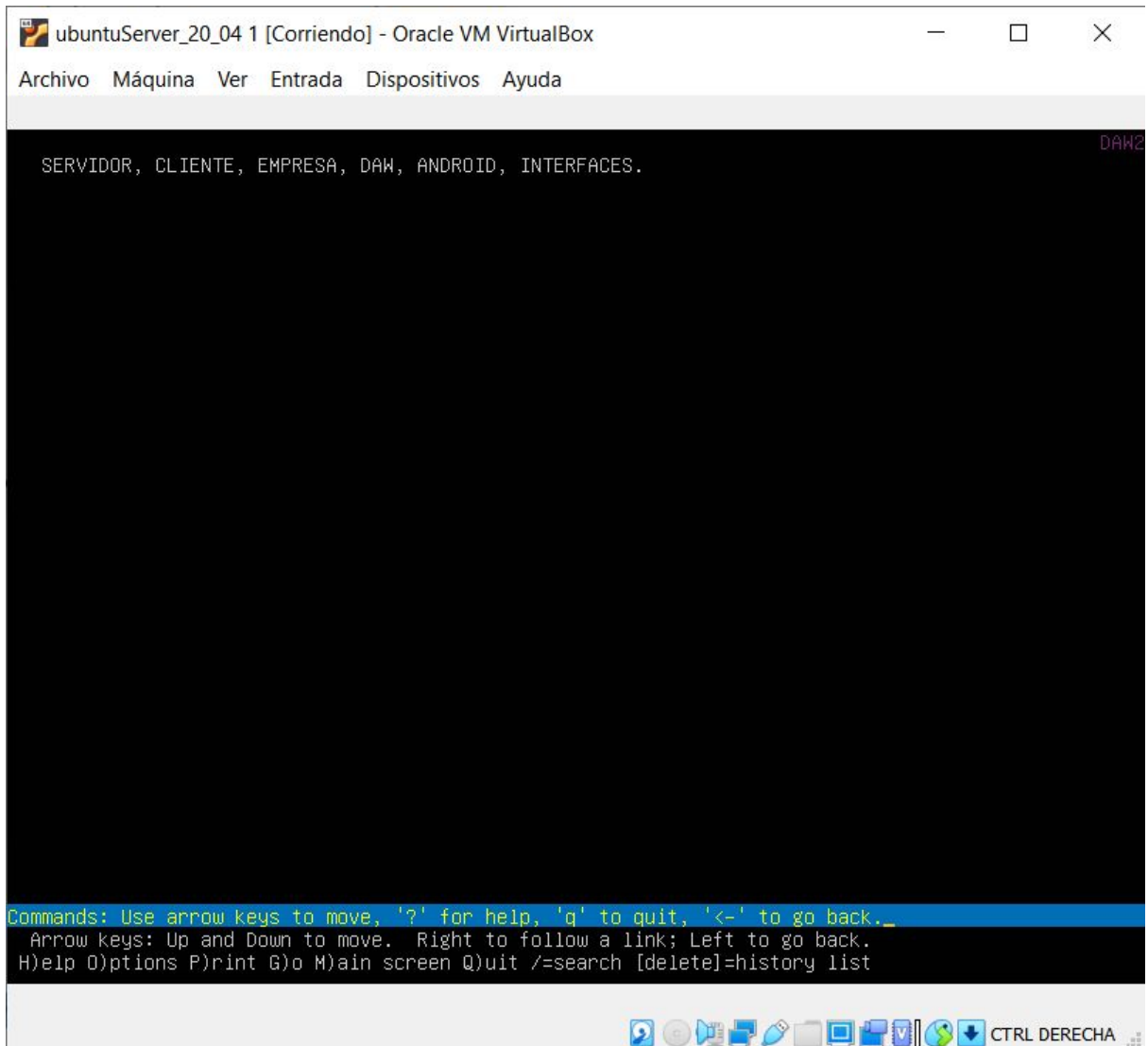
8. Pruebe sus Resultados

Ahora que cuenta con sus virtual hosts configurados, puede probar su configuración fácilmente dirigiéndose a los dominios que ha configurado directamente desde su navegador web:

Usamos navegador lynx para acceder a nuestras páginas web con el comando `lynx DAW1.com`



y lynx DAW2.com



Apache Mapeo URL

Opciones de directorios

=====

2) Explica la directiva "Options" (valores: All, FollowSymLinks, Indexes, MultiViews, SymLinksWithOwnerMatch, ExecCGI)

Controla qué cualidades están disponibles en un directorio concreto.

-All: todas las opciones menos multiViews, es el valor por defecto.

-FollowSymLinks: el servidor seguirá links simbólicos en este directorio.

-Indexes: Si una URL que mapea a un directorio es solicitada y no hay directoryindex en ese directorio se devolverá un listado formateado del directorio.

-MultiViews: El contenido negociado por multiviews se permite mediante mod_negotiation.

-SymLinksWithOwnerMatch: Permite seguir los enlaces simbólicos del directorio.

- ExecCGI: Admite la ejecución de scripts CGI.

3) Explica las opciones de configuración de "<Directory /var/www/>" (de tu instalación)

1. Options Indexes FollowSymLinks MultiViews
2. AllowOverride All
3. Order allow,deny
4. allow from all

4) ¿Para qué sirve el archivo dir.conf?

Directivas de configuración en el fichero httpd.conf

5) ¿Qué indican las siguientes opciones:

a) Options -Indexes

En el caso de que no haya una página index.html, Apache devolverá un listado con los ficheros correspondientes a la ruta

b) Options -FollowSymLinks

Permite seguir los enlaces simbólicos del directorio.

Un enlace simbólico se crea con el comando ln -s.

Apache tiene únicamente un tratamiento especial de los enlaces simbólicos. Los enlaces "duros" que son creados con el comando ln sin el parámetro -s son a todos los efectos iguales que los ficheros normales.

Trabajando con alias

=====

6) ¿Qué significa la siguiente directiva?:

Aquí es cuando alias entra en uso para poder asignar una abreviación que ejecute el mismo comando, llamando así de forma más resumida a la ruta de nuestro proyecto.

```
Alias /web /home/debian/directorio
<Directory /home/debian/directorio>
    Require all granted
</Directory>
```

Negociación de contenidos

=====

7) Sigue el caso práctico que realiza el ponente y haz una captura de las dos formas de hacer la negociación de contenidos. Hazlo primero con la opción "Options +Multiviews".

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName apache1.openwebinars.net

ServerAdmin webmaster@localhost
DocumentRoot /var/www/apache1
<Directory /var/www/apache1/internacional>
    Options +Multiviews
</Directory>
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error_apache1.log
CustomLog ${APACHE_LOG_DIR}/access_apache1.log combined

[ Read 33 lines ]
G Get Help      ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page
X Exit          ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^ Go To Line  ^V Next Page
```

```

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName apache1.openwebinars.net

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/apache1
    <Directory /var/www/apache1/internacional>
        DirectoryIndex index.var
        AddHandler type-map .var
    </Directory>
    # Available loglevels: trace8, ..., tracel, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error_apache1.log

```

[Read 34 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
 ^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page

Redirecciones

=====

8) Haz dos ejemplos del uso de la directiva "Redirect" (captura el archivo de configuración modificado)

```

VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName apache1.openwebinars.net

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/apache1

    Redirect "/traducir" "/internacional"

    # Available loglevels: trace8, ..., tracel, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error_apache1.log

```

[Read 34 lines]

G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
 X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page

Páginas de errores personalizadas

=====

9) Modifica el archivo "localized-error-pages.conf" para que utilice el manejo de errores que viene implementado por Apache (haz una captura de lo más relevante)

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName apache1.openwebinars.net

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/apache1

    ErrorDocument 404 /error/index.html

    # Available loglevels: trace8, ..., tracel, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error_apache1.log
    CustomLog ${APACHE_LOG_DIR}/access_apache1.log combined

```

[Read 33 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page

404... No se encuentra el recurso...

```
# Customizable error responses come in three flavors:
# 1) plain text
# 2) local redirects
# 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
#
# Putting this all together, we can internationalize error responses.
#
# We use Alias to redirect any /error/HTTP_<error>.html.var response to
# our collection of by-error message multi-language collections. We use
# includes to substitute the appropriate text.
#
# You can modify the messages' appearance without changing any of the
# default HTTP_<error>.html.var files by adding the line:
#
#Alias /error/include/ "/your/include/path/"

```

[Read 81 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page

```
# which allows you to create your own set of files by starting with the
# /usr/share/apache2/error/include/ files and copying them to /your/include/path/,
# even on a per-VirtualHost basis. If you include the Alias in the global server
# context, it has to come before the 'Alias /error/ ...' line.
#
# The default include files will display your Apache version number and your
# ServerAdmin email address regardless of the setting of ServerSignature.
#
# WARNING: The configuration below will NOT work out of the box if you have a
#           SetHandler directive in a <Location /> context somewhere. Adding
#           the following three lines AFTER the <Location /> context should
#           make it work in most cases:
#           <Location /error/>
#               SetHandler none
#           </Location>
#
# The internationalized error documents require mod_alias, mod_include
# and mod_negotiation. To activate them, uncomment the following 37 lines.
#<IfModule mod_negotiation.c>
#    <IfModule mod_include.c>
#        <IfModule mod_alias.c>
#
#G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page
#X Exit        ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line  ^V Next Page
```

Object not found!

The requested URL was not found on this server. If you entered the URL manually please check your spelling and try again.

If you think this is a server error, please contact the [webmaster](#).

Error 404

apache1.openwebinars.net
 Apache/2.4.25 (Debian)

Introducción a Git

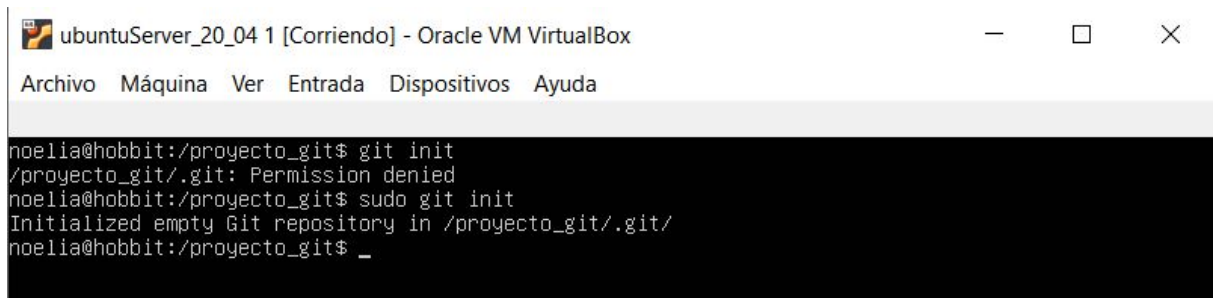
1. Comandos iniciales e instalación de git

```

noelia@hobbit:/$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Fetched 324 kB in 2s (199 kB/s)
sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
41 packages can be upgraded. Run 'apt list --upgradable' to see them.
noelia@hobbit:/$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 41 not upgraded.
noelia@hobbit:/$ git --version
git version 2.25.1
noelia@hobbit:/$
noelia@hobbit:/$ git config --global user.name "noelia"
noelia@hobbit:/$ git config --global user.email "noelia@gmail.com"
noelia@hobbit:/$ git config --global --list
user.name=noelia
user.email=noelia@gmail.com
noelia@hobbit:/$ mkdir proyecto_git
mkdir: cannot create directory 'proyecto_git': Permission denied
noelia@hobbit:/$ sudo mkdir proyecto_git

```

2. Iniciamos git



```

noelia@hobbit:/proyecto_git$ git init
/proyecto_git/.git: Permission denied
noelia@hobbit:/proyecto_git$ sudo git init
Initialized empty Git repository in /proyecto_git/.git/
noelia@hobbit:/proyecto_git$ _

```

3. Creamos un archivo txt

Creamos un archivo txt con texto dentro, en mi caso, ejercicio.txt

```

noelia@hobbit:/proyecto_git$ ls -la
total 16
drwxr-xr-x  3 root root 4096 Dec  7 23:05 .
drwxr-xr-x 21 root root 4096 Dec  7 22:54 ..
drwxr-xr-x  7 root root 4096 Dec  7 23:03 .git
-rw-r--r--  1 root root   31 Dec  7 23:05 ejercicio.txt
noelia@hobbit:/proyecto_git$ _

```

4. Hacemos un git status

Hacemos un git status para saber el estado en el que se encuentra el archivo, aparece en color rojo lo que nos indica que está en la rama master y el archivo a un no ha pasado a la zona de indexación.

Luego hacemos un git add, para pasarlo a la zona de indexación y volvemos a hacer un git status, en este caso aparece en color verde.

```
noelia@hobbit:/proyecto_git$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ejercicio.txt

nothing added to commit but untracked files present (use "git add" to track)
noelia@hobbit:/proyecto_git$ git add ejercicio.txt
fatal: Unable to create '/proyecto_git/.git/index.lock': Permission denied
noelia@hobbit:/proyecto_git$ sudo git add ejercicio.txt
noelia@hobbit:/proyecto_git$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   ejercicio.txt
```

5. Cambio en el archivo ejercicio.txt

Hacemos un cambio en el archivo ejercicio.txt, luego hacemos un git status y nos sale el estado del archivo, que ha sido modificado.

Luego, hacemos un git add para que el archivo vuelva a pasar con el cambio a la zona de indexación.

```
noelia@hobbit:/proyecto_git$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   ejercicio.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ejercicio.txt

noelia@hobbit:/proyecto_git$ git add ejercicio.txt
```

6. Creamos otro archivo “ejercicio 2.txt”

Creamos otro archivo “ejercicio 2.txt”, hacemos git add y git status y por último hacemos un commit -m con nombre “cambios” para ambos archivos.

```
root@hobbit:/proyecto_git# git commit -m "cambios"
[master (root-commit) 072461e] cambios
 2 files changed, 4 insertions(+)
 create mode 100644 ejercicio.txt
 create mode 100644 ejercicio2.txt
root@hobbit:/proyecto_git# git status
On branch master
nothing to commit, working tree clean
root@hobbit:/proyecto_git#
```

7. git log --oneline --color

Hacemos git log --oneline --color para visualizar el historial de cambios.

```
root@hobbit:/proyecto_git# git log --oneline --color
072461e (HEAD -> master) cambios
root@hobbit:/proyecto_git# _
```

8. git rm ejercicio 2.txt

Hacemos un git rm ejercicio 2.txt para eliminar el ejercicio 2 del repositorio, luego hacemos un status para que nos muestre el estado del archivo

```

root@hobbit:/proyecto_git# git rm ejercicio2.txt
rm 'ejercicio2.txt'
root@hobbit:/proyecto_git# git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    ejercicio2.txt

```

9. mover archivos entre carpetas

Para mover archivos entre carpetas, creamos la carpeta con `mkdir`, y luego usamos el comando `mv archivo.txt /ruta/de/destino`, al final hacemos un `git status` y el archivo aparecerá como delete. Si hacemos un `ls -la` en ambas podemos ver el archivo.

```

root@hobbit:/# mkdir otra_carpeta
root@hobbit:/# cd otra_carpeta
root@hobbit:/otra_carpeta# cd ..
root@hobbit:/# cd proyecto_git
root@hobbit:/proyecto_git# ls -la
total 16
drwxr-xr-x  3 root root 4096 Dec  8 00:21 .
drwxr-xr-x 22 root root 4096 Dec  8 00:24 ..
drwxr-xr-x  8 root root 4096 Dec  8 00:21 .git
-rw-r--r--  1 root root   64 Dec  7 23:14 ejercicio.txt
root@hobbit:/proyecto_git# mv ejercicio.txt
mv: missing destination file operand after 'ejercicio.txt'
Try 'mv --help' for more information.
root@hobbit:/proyecto_git# mv ejercicio.txt /otra_carpeta
root@hobbit:/proyecto_git# ls -la
total 12
drwxr-xr-x  3 root root 4096 Dec  8 00:26 .
drwxr-xr-x 22 root root 4096 Dec  8 00:24 ..
drwxr-xr-x  8 root root 4096 Dec  8 00:21 .git
root@hobbit:/proyecto_git# cd ..
root@hobbit:/# cd otra_carpeta
root@hobbit:/otra_carpeta# ls -la
total 12
drwxr-xr-x  2 root root 4096 Dec  8 00:26 .
drwxr-xr-x 22 root root 4096 Dec  8 00:24 ..
-rw-r--r--  1 root root   64 Dec  7 23:14 ejercicio.txt

```

10. Creamos una key ssh

Creamos una key ssh para subir el archivo a un repositorio de github

```

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDJrG/9nv/i0DiW7PISCqrRdP5izxn64bK+VgeX2PrTsBANu7y7DAozaBG6BU4c
aPg93QB1jzMuTDwesP3V/Es054XhA473M+et0nCNqymxjD1/uJCBICrTEMmyRDTBWC7PrLzHppusms04ToGugfwAo4YU8QpG0i5
G3/Vd3XeUtgRssLWzn9+r6csq8J9w8bI99U92Kvcykk+KSeppaKZ+hR5SSxDR29t9z8S1RRcfSYhbr6pzL2007B1snPayIac6+np
D/rK1NqiNBXEVSsfTg/9Vp9NT3LT1cXqr0IU44o4Xiya163UpJsaHCzn7N3jQC3iA7n9Nt1Q0ALXiQWPrZVnBmYuS2//VcE3S9Dh
gUBheu5NUf6Ej7C4z1Awnt+7Wx10+bs06naeZCE19r1EBvdK+QnXSxL1QFmKagtORpg/BdXtN7VkSmDQVaU805bDpi/TfmJJUXEn
Br0Zjfw/iY4gzszPdL15HIRuFu2RVdeTpL2fpbF0Sc0gMWdDEQk= ruizlopeznoelia@gmail.com

```


11. copiamos en github

Lo copiamos en github en el apartado SSH key.

introducimos nuestro usuario de git en la virtualbox y hacemos un push de la rama master.


El archivo nos aparecerá como eliminado.

```
noelia@hobbit:/proyecto_git$ git push -u origin master
Username for 'https://github.com': Notelodigo98
Password for 'https://Notelodigo98@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 324 bytes | 324.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/Notelodigo98/Introduccion-a-git.git
 * [new branch]      master -> master
error: could not lock config file .git/config: Permission denied
error: Unable to write upstream branch configuration
hint:
hint: After fixing the error cause you may try to fix up
hint: the remote tracking information by invoking
hint: "git branch --set-upstream-to=origin/master".
error: update_ref failed for ref 'refs/remotes/origin/master': cannot lock ref 'refs/r
master': unable to create directory for .git/refs/remotes/origin/master
noelia@hobbit:/proyecto_git$ sudo git push -u origin master
Username for 'https://github.com': Notelodigo98
Password for 'https://Notelodigo98@github.com':
Branch 'master' set up to track remote branch 'master' from 'origin'.
Everything up-to-date
noelia@hobbit:/proyecto_git$ git status
On branch master
Your branch is up to date with 'origin/master'.


Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    ejercicio2.txt



noelia@hobbit:/proyecto_git$ _
```

los archivos se han subido a mi repositorio de github.


master
1 branch
0 tags

[Go to file](#)
[Add file](#)
[Code](#)


Notelodigo98 cambios
072461e 2 hours ago
1 commits

 ejercicio.txt	cambios	2 hours ago
 ejercicio2.txt	cambios	2 hours ago

Help people interested in this repository understand your project by adding a README.

[Add a README](#)

12. github hacemos un pull

Para hacer cambios en el archivo de github hacemos un pull para bajarnos el archivo, lo modificamos, hacemos git add, luego hacemos un commit -m "cambios" y por último hacemos un push.

Yo he hecho un status para saber el estado del archivo en cada momento.

```
noelia@hobbit:/proyecto_git$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   ejercicio.txt
        deleted:    ejercicio2.txt

noelia@hobbit:/proyecto_git$ sudo git commit -m "cambios realizados"
[master 9856734] cambios realizados
 2 files changed, 1 insertion(+), 2 deletions(-)
 delete mode 100644 ejercicio2.txt
noelia@hobbit:/proyecto_git$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
noelia@hobbit:/proyecto_git$ git push origin master
Username for 'https://github.com': Notelodigo98
Password for 'https://Notelodigo98@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/Notelodigo98/Introduccion-a-git.git/'
noelia@hobbit:/proyecto_git$ git push origin master
Username for 'https://github.com': Notelodigo98
Password for 'https://Notelodigo98@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Notelodigo98/Introduccion-a-git.git
   072461e..9856734  master -> master
error: update_ref failed for ref 'refs/remotes/origin/master': cannot lock ref 'refs/remotes/origin/master': Unable to create '/proyecto_git/.git/refs/remotes/origin/master.lock': Permission denied
noelia@hobbit:/proyecto_git$
```

3 lines (3 sloc) | 96 Bytes

```
1  Hola esto es una prueba de git
2  Esta es la segunda prueba de git
3  Esta es una modificacion de git
```

Instalar vsftpd

Instalar Vsftpd

Primero lo primero, obtengamos las actualizaciones de nuestros paquetes antes de continuar con la instalación del daemon vsftpd. Para comenzar, ejecuta el siguiente comando:

```
sudo apt-get update
```

Espera a que se completen todos los procesos y verás una confirmación tan pronto como finalice la actualización.

```
Reading package lists... Done
```

Cuando termines con esto, instala el daemon vsftpd usando el siguiente comando:

```
sudo apt-get install vsftpd
```

Ahora verás un mensaje de confirmación en el que tendrás que escribir Y y presionar Enter para continuar con la instalación.

```
root@kindly-lawyer:~# apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libmemcached11 libmemcachedutil2
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  proftpd-basic
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 1 to remove and 118 not upgraded.
Need to get 115 kB of archives.
After this operation, 4140 kB disk space will be freed.
Do you want to continue? [Y/n] Y
```

Una vez completada la instalación, haz una copia de seguridad del archivo original para que podamos comenzar nuestro trabajo con un archivo de configuración en blanco:

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
```

Ahora estamos listos para dar el siguiente paso y configurar el firewall.

Permitir el tráfico FTP desde el firewall

Para permitir que el servidor FTP de Ubuntu se comunice con el mundo exterior, tiene que abrirse paso a través del firewall. Primero veamos si el firewall está habilitado en la máquina o no. Ejecuta el siguiente comando para verificar el estado:

```
sudo ufw status
```

Si ves el siguiente mensaje:

```
ufw: command not found
```

Significa que el firewall no está instalado y puedes continuar con el siguiente paso.

Sin embargo, si el resultado muestra algunas reglas definidas o un mensaje de que el estado del firewall es activo, deberás verificar si el tráfico FTP funcionará. Avancemos y abramos los puertos 20 y 21 para el tráfico FTP; los puertos 40000-50000 serán los reservados para el rango de puertos pasivos que eventualmente se establecerán en el archivo de configuración y el puerto 990 se usará cuando se habilite el TLS. Ejecuta los siguientes comandos para hacerlo:

```
sudo ufw allow 20/tcp
```

```
sudo ufw allow 21/tcp
```

```
sudo ufw allow 990/tcp
```

```
sudo ufw allow 40000:50000/tcp
```

Ahora veamos el estado de nuevo:

```
sudo ufw status
```

El resultado debería ser algo así:

Output

Status: active

To	Action	From
--	-----	----
990/tcp	ALLOW	Anywhere
20/tcp	ALLOW	Anywhere
21/tcp	ALLOW	Anywhere
40000:50000/tcp	ALLOW	Anywhere
20/tcp (v6)	ALLOW	Anywhere (v6)
21/tcp (v6)	ALLOW	Anywhere (v6)
990/tcp (v6)	ALLOW	Anywhere (v6)
40000:50000/tcp (v6)	ALLOW	Anywhere (v6)

Ahora que tenemos todos los puertos necesarios abiertos y disponibles para nosotros, podemos pasar al siguiente paso.

Crear el directorio de usuarios

En el tercer paso para crear un servidor FTP de Ubuntu, tendremos que seleccionar el usuario que va a utilizar el acceso FTP. Solo para mostrar cómo se hace, agregaremos un nuevo usuario. Para crearlo, usa el siguiente comando:

```
sudo adduser alex
```

Cuando el sistema te pregunte, ingresa una contraseña para el usuario y completa todos los demás detalles. Lo ideal es que el FTP se restrinja a un directorio específico por motivos de seguridad. Vsftpd usa jaulas chroot para lograr esto. Con chroot habilitado, un usuario local

está restringido a su directorio de inicio (por defecto). Sin embargo, es posible que debido a la seguridad de vsftpd, un usuario no pueda escribir en el directorio. No eliminaremos los privilegios de escritura de la carpeta de inicio; en su lugar, crearemos un directorio ftp que actuará como chroot junto con un directorio de archivos modificables que será responsable de mantener los archivos pertinentes. Usa el siguiente comando para crear la carpeta FTP:

```
sudo mkdir /home/alex/ftp
```

Establece la propiedad usando:

```
sudo chown nobody:nogroup /home/alex/ftp
```

Finalmente, elimina los permisos de escritura:

```
sudo chmod a-w /home/alex/ftp
```

Ahora, usa el siguiente comando para verificar los permisos:

```
sudo ls -la /home/alex/ftp
```

El resultado debería ser algo así:

```
total 8
dr-xr-xr-x 2 nobody nogroup 4096 Jun 29 11:32 .
drwxr-xr-x 3 alex alex 4096 Jun 29 11:32 ..
```

Como paso siguiente, crearemos el directorio contenedor de archivos y asignaremos la propiedad:

```
sudo mkdir /home/alex/ftp/files
```

```
sudo chown alex:alex /home/alex/ftp/files
```

Finalmente, agrega un archivo de prueba al directorio el cual se usará cuando probemos todo más adelante:

```
echo "vsftpd sample file" | sudo tee /home/alex/ftp/files/sample.txt
```

Configurar vsftpd

El siguiente paso en nuestra apuesta por configurar un servidor FTP en Ubuntu VPS, es configurar vsftpd y nuestro acceso FTP. En este tutorial, permitiremos que un solo usuario se conecte con FTP utilizando una cuenta shell local. Las dos configuraciones clave requeridas para esto ya están establecidas en el archivo de configuración (vsftpd.conf). En

primer lugar, verifica que el archivo de configuración tenga una configuración que coincida con las mencionadas a continuación utilizando el comando nano:

```
sudo nano /etc/vsftpd.conf
```

```
...
```

```
# Allow anonymous FTP? (Disabled by default).
```

```
anonymous_enable=NO
```

```
#
```

```
# Uncomment this to allow local users to log in.
```

```
local_enable=YES
```

```
...
```

En el mismo archivo, procederemos a eliminar # y a habilitar el write_enable:

```
...
```

```
write_enable=YES
```

```
...
```

Chroot tampoco se comentará para garantizar que el usuario conectado a través de FTP solo acceda a los archivos dentro del directorio permitido:

```
...
```

```
chroot_local_user=YES
```

```
...
```

También se deben agregar manualmente algunos valores nuevos. Simplemente puedes pegarlos en la parte inferior del archivo. En primer lugar, se agregará un user_sub_token en la ruta del directorio local_root. Esto permitirá que la configuración funcione con el usuario actual y con cualquier otro usuario que se agregue posteriormente:

```
user_sub_token=$USER
```

```
local_root=/home/$USER/ftp
```

Para garantizar que haya una cantidad considerable de conexiones disponibles, limitaremos la cantidad de puertos utilizados en el archivo de configuración:

```
pasv_min_port = 40000
```

```
pasv_max_port = 50000
```

En este tutorial, planeamos permitir el acceso caso por caso, así que ajustemos la configuración de forma tal que el acceso sólo se otorgue a los usuarios que se hayan agregado explícitamente a una lista:

```
userlist_enable=YES
```

```
userlist_file=/etc/vsftpd.userlist
```

```
userlist_deny=NO
```

El flag `userlist_deny` es el responsable de alternar la lógica; cuando se establece en «NO», solo se permitirá el acceso a los usuarios especificados en la lista. Una vez hecho esto, haz clic en CTRL+X y confirma los cambios del archivo.

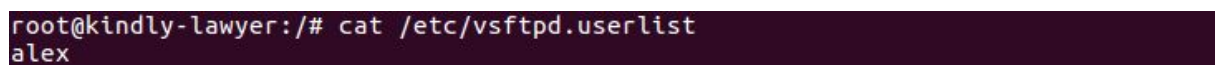
Por último, procederemos con la creación y adición de nuestro usuario al archivo:

```
echo "alex" | sudo tee -a /etc/vsftpd.userlist
```

Verifica que el usuario esté realmente activo ejecutando el siguiente comando:

```
cat /etc/vsftpd.userlist
```

El resultado debe ser «alex» como se muestra en esta captura de pantalla:



```
root@kindly-lawyer:/# cat /etc/vsftpd.userlist
alex
```

Reinicia el daemon utilizando el siguiente comando para cargar los cambios de configuración:

```
sudo systemctl restart vsftpd
```

Hacer que el FTP sea seguro

Por defecto, FTP no hace ninguna encriptación de datos, por eso utilizaremos TLS/SSL para garantizar la seguridad. En primer lugar, debemos crear el certificado SSL y usarlo para proteger el servidor FTP de Ubuntu. Para comenzar, usa el siguiente comando:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```

El flag `-days` hace que el certificado sea válido por un año y hemos incluido una clave privada RSA de 2048 bits en el mismo comando. Una vez sean solicitados, ingresa los datos personales correspondientes en el campo provisto.

Cuando termines de crear el certificado, abre nuevamente el archivo de configuración:

```
sudo nano /etc/vsftpd.conf
```

El final del archivo debe contener dos líneas que comiencen con «`_rsa`». Comenta ambas líneas así:

```
# rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
```

```
# rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

En lugar de eso, apuntemos el archivo de configuración al certificado que acabamos de crear. Agrega las siguientes líneas:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
```

```
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

Ahora habilitaremos SSL y nos aseguraremos de que solo los clientes que tengan SSL habilitados nos puedan contactar. Cambia el valor de `ssl_enable` a YES:

```
ssl_enable=YES
```

Ahora agrega las siguientes líneas para mayor protección: (Esto no permitirá conexiones anónimas a través de SSL)

```
allow_anon_ssl=NO
```

```
force_local_data_ssl=YES
```

```
force_local_logins_ssl=YES
```

Configura el servidor para usar TLS usando:

```
ssl_tlsv1=YES
```

```
ssl_sslv2=NO
```

```
ssl_sslv3=NO
```

Aquí agregaremos 2 opciones más. En primer lugar, no será necesario reutilizar SSL porque puede ocasionar que muchos clientes de FTP se averíen. En segundo lugar, utilizaremos suites de encriptación de alto cifrado, lo que significa que las longitudes de claves son iguales o superiores a 128 bits.

```
require_ssl_reuse=NO
```

```
ssl_ciphers=HIGH
```

Comencemos una vez más para aplicar las nuevas configuraciones:

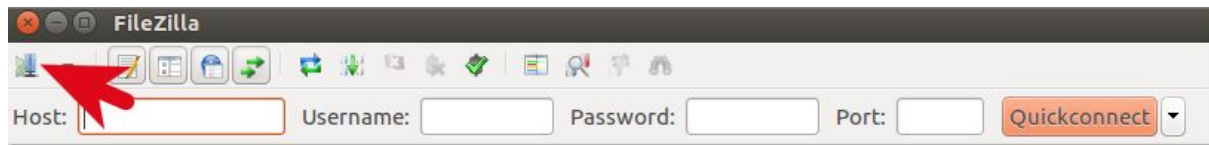
```
sudo systemctl restart vsftpd
```

¡Buen trabajo! Has configurado el servidor FTP en tu VPS de Ubuntu para que funcione con el protocolo SSL/TLS.

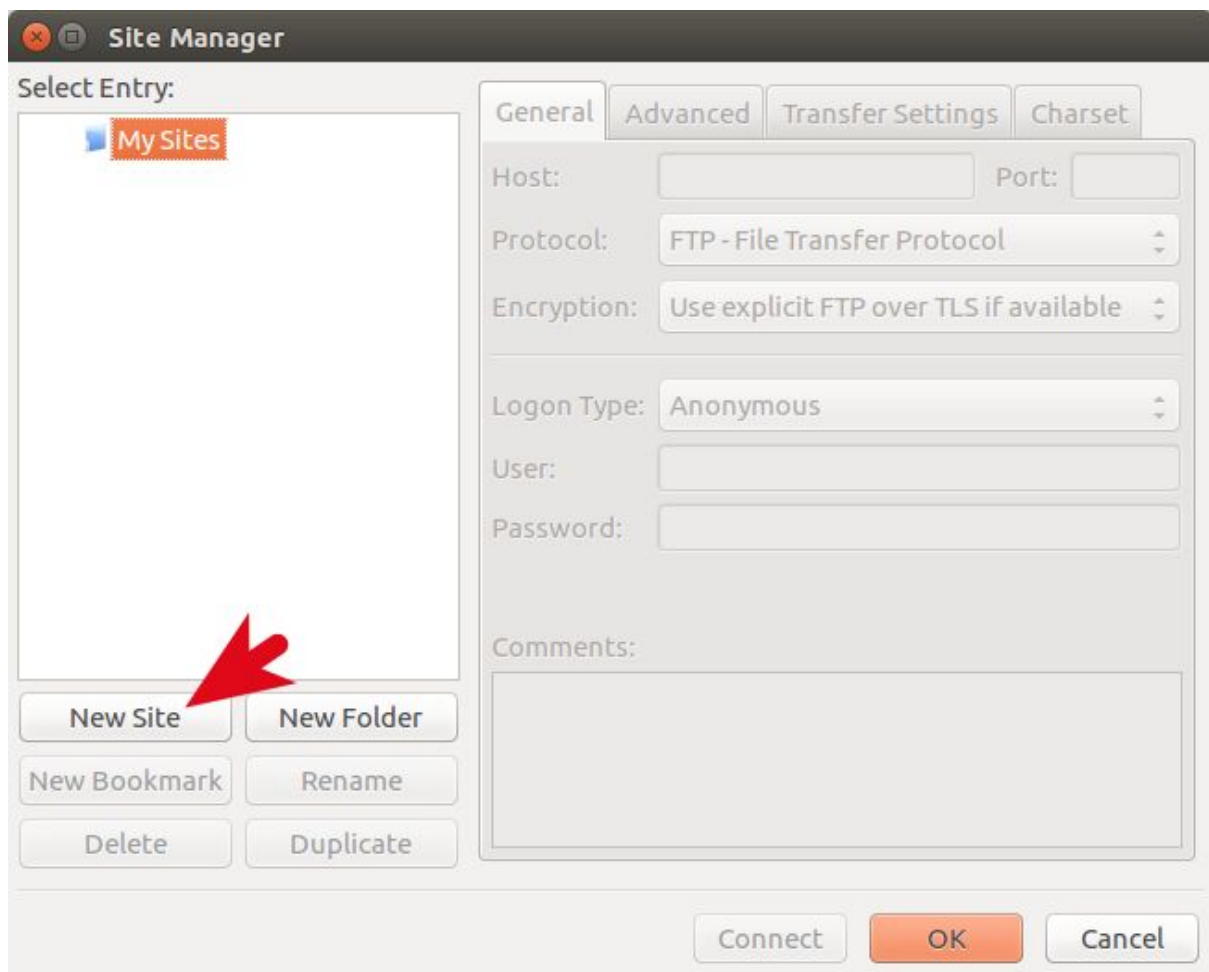
Prueba de conexiones con FileZilla

Hoy en día, la mayoría de los clientes de FTP admiten configuraciones de cifrado TLS, por lo que es una excelente manera de comprobar si tu servidor FTP de Ubuntu funciona según

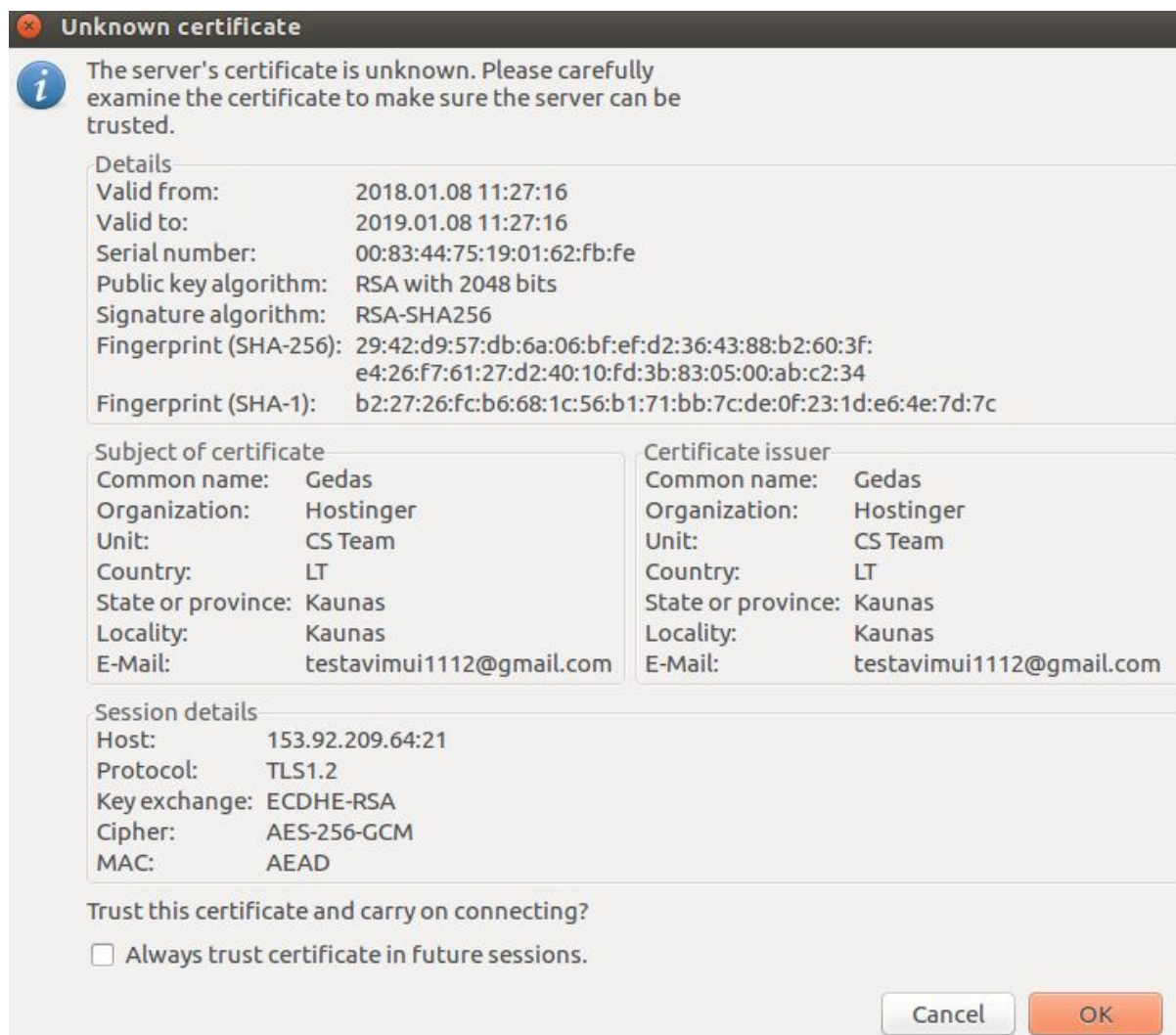
lo previsto. Para probar la conexión, utilizaremos un cliente FTP de FileZilla. Para comenzar inicia FileZilla y haz clic en el icono de Site Manager.



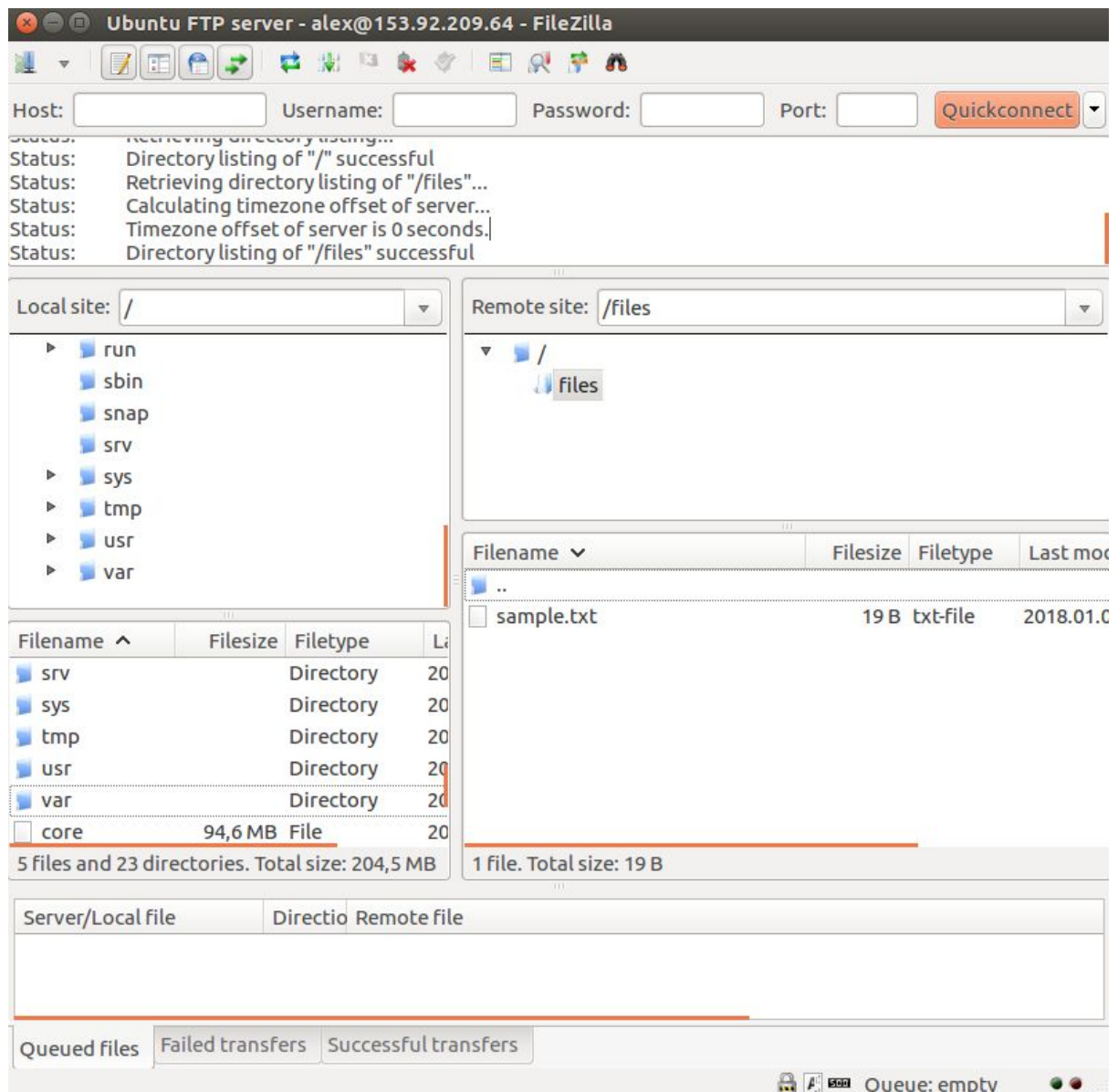
Haz clic en el botón New Site en la ventana que aparece para comenzar a ingresar los detalles del servidor FTP de Ubuntu.



Completa todos los detalles requeridos con tu información del servidor FTP de Ubuntu recién creada. Como lo configuramos para usar TLS, también podemos marcar el cifrado para que sea explícitamente FTP sobre TLS.



Después de confirmar, el directorio raíz con el archivo de prueba debería aparecer en tu pantalla.



Apache control de acceso, autenticación

Instalar el paquete de utilidades de Apache

Comenzaremos por actualizar nuestro servidor e instalar un paquete que necesitaremos.

Para completar este tutorial, usaremos una herramienta llamada htpasswd, que forma parte

del paquete apache2-utils, a fin de crear el archivo y administrar el nombre de usuario

y las contraseñas que se necesitarán para acceder a contenido restringido.

```
sudo apt-get update
```

```
sudo apt-get install apache2-utils
```

Una vez instalado esto, tendremos acceso al comando htpasswd.

Crear el archivo de contraseña

El comando htpasswd nos permitirá crear un archivo de contraseña que Apache puede usar para

autenticar usuarios. Crearemos un archivo oculto para este propósito, llamado .htpasswd dentro

de nuestro directorio de configuración /etc/apache2.

La primera vez que se usa esta utilidad, se debe añadir la opción -c para crear el _

passwdfile _especificado. Especificamos un nombre de usuario (en este ejemplo, sammy)

al final del comando para crear una entrada nueva dentro del archivo:

```
sudo htpasswd -c /etc/apache2/.htpasswd sammy
```

Se le solicitará proporcionar y confirmar una contraseña para el usuario.

Deje el argumento -c para cualquier usuario adicional que desee añadir, a fin de no sobrescribir

el archivo:

```
sudo htpasswd /etc/apache2/.htpasswd another_user
```

Si vemos el contenido del archivo, podemos ver el nombre de usuario y la contraseña cifrada para

cada registro:

```
cat /etc/apache2/.htpasswd
```

Output

```
sammy:$apr1$.0CAabqX$rb8lueIORA/p8UzGPYtGs/  
another_user:$apr1$fqH7UG8a$SrUxurp/Atfq6j7GL/VEC1
```

Ahora, nuestros usuarios y nuestras contraseñas se encuentran en un formato que Apache puede leer.

Paso 3: Configurar la autenticación de contraseña de Apache

En este paso, debemos configurar Apache para que verifique este archivo antes de proporcionar

nuestro contenido protegido. Podemos hacerlo de una de estas dos formas: ya sea directamente en

el archivo de host virtual de un sitio o mediante la disposición de archivos `.htaccess` en los directorios que requieren restricción. Generalmente, es mejor usar el archivo de host virtual, pero si necesita que los

usuarios no root puedan administrar sus propias restricciones de acceso, verificar las restricciones de

control de versiones junto con el sitio web o contar con una aplicación web que ya use

archivos `.htaccess` para otros fines, consulte la segunda opción.

Seleccione la opción que mejor satisfaga sus necesidades.

Configurar el control de acceso dentro de la definición de host virtual (preferida)

La primera opción es editar la configuración de Apache y añadir la protección con contraseña al

archivo de host virtual. Esto generalmente proporcionará un mejor rendimiento porque evita el

esfuerzo de leer archivos de configuración distribuida. Esta opción requiere acceso a la configuración,

que no siempre está disponible, pero si tiene acceso es la que se recomienda.

Comience por abrir el archivo de host virtual al que desea añadir una restricción. Para nuestro ejemplo,

usaremos el archivo default-ssl.conf que contiene el host virtual predeterminado instalado a través del paquete de apache de Ubuntu. Abra el archivo con un editor de texto de línea de comandos como nano:

```
sudo nano /etc/apache2/sites-enabled/default-ssl.conf
```

Dentro de este, sin los comentarios vaciados, el archivo debería tener un aspecto similar al siguiente:

```
/etc/apache2/sites-enabled/default-ssl.conf
```

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Copy

La autenticación se realiza por directorio. Para configurar la autenticación, necesitará apuntar al

directorio que desea restringir con un bloque `<Directory ____>`. En nuestro ejemplo, restringimos

el root de todo el documento, pero puede modificar este listado para apuntar solo a un directorio

específico dentro del espacio web:

```
/etc/apache2/sites-enabled/default-ssl.conf
```

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined

  <Directory "/var/www/html">
  </Directory>
</VirtualHost>
```

Dentro de este bloque de directorio, especificaremos que configuraremos la autenticación Basic. Para el AuthName, seleccione un nombre de territorio que el usuario verá cuando se le solicite ingresar

las credenciales. Utilice la directiva AuthUserFile para orientar a Apache al archivo de contraseña

que creamos. Por último, cree un requisito para que solo un valid-user pueda tener acceso a este

recurso. Esto significa que podrá ingresar quien sea capaz que verificar su identidad con una contraseña:

```
/etc/apache2/sites-enabled/default-ssl.conf
```

```
<VirtualHost *:80>
```

```
ServerAdmin webmaster@localhost
```

```
DocumentRoot /var/www/html
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
<Directory "/var/www/html">
```

```
AuthType Basic
```

```
AuthName "Restricted Content"
```

```
AuthUserFile /etc/apache2/.htpasswd
```

```
Require valid-user
```

```
</Directory>
```

```
</VirtualHost>
```

Copy

Guarde y cierre el archivo cuando termine. Si utiliza nano, puede hacerlo presionando CTRL+X seguido de ENTER.

Antes de reiniciar el servidor web, puede verificar la configuración con el siguiente comando:

```
sudo apache2ctl configtest
```

Si todo está comprobado y el resultado es Syntax OK, puede reiniciar el servidor para

implementar su política de contraseñas. Debido a que systemctl no muestra el resultado de

todos los comandos de gestión de servicios, usaremos status para asegurarnos de que el servidor

esté en ejecución:

```
sudo systemctl restart apache2
```

```
sudo systemctl status apache2
```

Ahora, el directorio que especifica debe estar protegido con contraseña.

Para habilitar la protección con contraseña usando archivos .htaccess, abra el archivo de configuración principal de Apache con un editor de texto de línea de comandos como nano:

```
sudo nano /etc/apache2/apache2.conf
```

Encuentre el bloque <Directory> del directorio /var/www que contiene la root del documento. Active el procesamiento de .htaccess cambiando la directiva AllowOverride dentro de ese bloque de None a All:

```

                                     /etc/apache2/apache2.conf
...

<Directory /var/www/>
  Options Indexes FollowSymLinks
  AllowOverride All
  Require all granted
</Directory>

...
```

Guarde y cierre el archivo cuando termine. Si utiliza nano, puede hacerlo presionando CTRL+X seguido de Y y ENTER.

A continuación, debemos añadir un archivo .htaccess al directorio que deseamos restringir

. En nuestra prueba, restringimos la root de todo el documento (todo el sitio web), que tiene base en /var/www pero puede disponer este archivo en cualquier directorio en que desee restringir

el acceso:

```
sudo nano /var/www/html/.htaccess
```

Dentro de este archivo, especifique que deseamos establecer la autenticación Basic. Para el AuthName, seleccione un nombre de territorio que el usuario verá cuando se solicitan credenciales.

Utilice la directiva AuthUserFile para orientar a Apache al archivo de contraseña que creamos.

Por último, solicitaremos que este recurso sea accesible para un valid-user, lo cual significa que podrá ingresar quien sea capaz de verificar su identidad con una contraseña:

```
/var/www/html/.htaccess
```

```
AuthType Basic
```

```
AuthName "Restricted Content"
```

```
AuthUserFile /etc/apache2/.htpasswd
```

```
Require valid-user
```

Copy

Guarde y cierre el archivo. Reinicie el servidor web para que proteja con contraseña todo

el contenido que se encuentre en el directorio con el archivo .htaccess, o debajo de este, y

utilice `systemctl status` para verificar el éxito del reinicio:

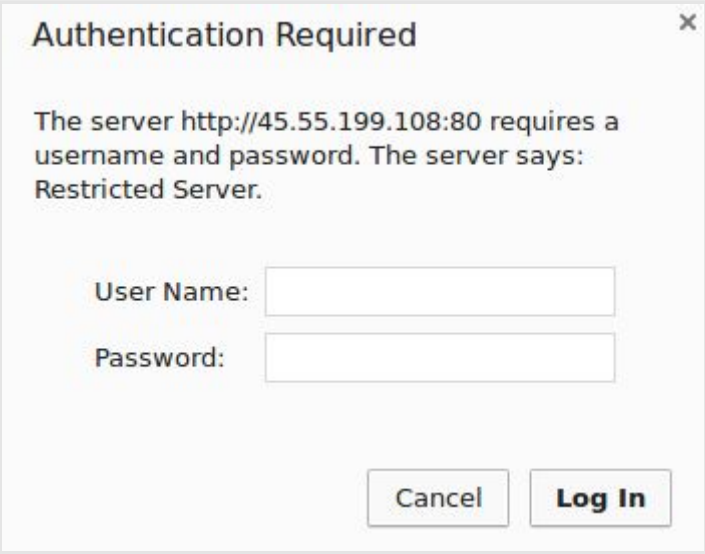
```
sudo systemctl restart apache2
```

```
sudo systemctl status apache2
```

El directorio que especifica debe estar protegido con contraseña.

Confirmar la autenticación con contraseña

Para confirmar que su contenido esté protegido, intente acceder a la parte restringida de este desde un navegador web. Debería ver una solicitud de ingreso de nombre de usuario y contraseña con el siguiente aspecto:



Authentication Required [X]

The server `http://45.55.199.108:80` requires a username and password. The server says:
Restricted Server.

User Name:

Password:

Si introduce las credenciales correctas, se le permitirá acceder al contenido.

Si escribe credenciales incorrectas o presiona “Cancel”, verá la página de error “Unauthorized”:

Unauthorized

This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.

Apache/2.4.7 (Ubuntu) Server at 45.55.199.108 Port 80