

Tabla de Contenidos

- [Pre-requisitos](#)
 - [Multipass](#)
- [Instalando la maquina virtual](#)
 - [Creando el servidor](#)
 - [Instalando paquetes en el servidor](#)
 - [Configurando Squid en el servidor](#)
 - [Habilitando el firewall](#)
- [Configurando tu navegador para usar proxy](#)
 - [Firefox](#)
- [Autores](#)
- [Fuentes](#)

Pre-requisitos

► [Click aqui para ver los pre-requistos](#)

Multipass



[Multipass](#) proporciona una interfaz de línea de comandos para iniciar, administrar y, en general, jugar con instancias de Linux. La descarga de una imagen fresca lleva unos segundos, y en cuestión de minutos una VM puede estar en funcionamiento.

[Multipass](#) es un software gratuito desarrollado por Ubuntu que permite instalar maquinas virtuales de manera similar a que si las tuvieramos en un [container](#). Las ventajas de Multipass a comparacion de las classicas maquinas virtuales es basicamente que no se necesita un gran equipo para poder instancias de ubuntu. Cuando se habla de una instancia se hace referencia a una imagen que contiene cierto sistema operativo.

Instalando Multipass

El siguiente link lleva a la descarga de un ejecutable desde el sitio oficial:

-  [Link de descarga](#)

Habilitando Hyper-V en Windows

Para poder correr cualquier maquina virtual en Windows tenemos que habilitar [Hyper-V](#), que por default viene deshabilitado.

Para poder habilitarlo basta con abrir Windows Power Shell (como Administrador), escribir el siguiente comando y reiniciar:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

Si no sabe como abrir Windows Power Shell en Windows vea el siguiente video [como abrir Windows Power Shell en Windows](#)

Instalando la maquina virtual

Para instalar la maquina virtual lo haremos a travez de **multipass**, lo cual nos creara una instancia de ubuntu de manera muy sencilla a la cual podemos acceder muy facilmente y sin necesitar gran cantidad de recursos de nuestro sistema.

Creando el servidor

Con el siguiente comando crearemos el servidor en una VM de ubuntu con version 18.04

```
multipass launch bionic --name servidor
```

► Click aqui para ver la salida del comando anterior

```
bash-5.0$ multipass launch bionic --name servidor
Launched: servidor
```

Instalando paquetes en el servidor

Una vez creado el servidor, procederemos a instalar paquetes para poder crear nuestro servidor proxy, para eso tendremos que estar dentro del servidor antes de poder instalar cualquier paquete.

Inserta el siguiente comando en la terminal para poder entrar al servidor

```
multipass shell servidor
```

► Click aqui para ver la salida del comando anterior

```
bash-5.0$ multipass shell servidor
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-106-generic x86_64)

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

System information as of Thu Jun 18 01:22:21 CDT 2020

System load:  0.0               Processes:            90
Usage of /:   21.0% of 4.67GB   Users logged in:     0
Memory usage: 11%              IP address for enp0s2: 192.168.64.22
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@servidor:~$
```

Una vez dentro del servidor procederemos a instalar los paquetes necesarios

Inserta el siguiente comando para actualizar los paquetes que ya contiene nuestro servidor

```
sudo apt update -y && sudo apt upgrade -y
```

► Click aquí para ver la salida del comando anterior

```

ubuntu@servidor:~$ sudo apt update -y && sudo apt upgrade -y
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  dbus libdbus-1-3 libdrm-common libdrm2 libnetplan0 netplan.io nplan
7 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 457 kB of archives.
After this operation, 1024 B of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 dbus amd64 1.12.2-1ubuntu1.2 [150 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdbus-1-3 amd64 1.12.2-1ubuntu1.2 [175 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnetplan0 amd64 0.99-0ubuntu3~18.04.3 [22.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 netplan.io amd64 0.99-0ubuntu3~18.04.3 [70.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 nplan all 0.99-0ubuntu3~18.04.3 [1800 B]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdrm-common all 2.4.101-2~18.04.1 [5560 B]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdrm2 amd64 2.4.101-2~18.04.1 [32.3 kB]
Fetched 457 kB in 21s (22.2 kB/s)
(Reading database ... 60098 files and directories currently installed.)
Preparing to unpack .../0-dbus_1.12.2-1ubuntu1.2_amd64.deb ...
Unpacking dbus (1.12.2-1ubuntu1.2) over (1.12.2-1ubuntu1.1) ...
Preparing to unpack .../1-libdbus-1-3_1.12.2-1ubuntu1.2_amd64.deb ...
Unpacking libdbus-1-3:amd64 (1.12.2-1ubuntu1.2) over (1.12.2-1ubuntu1.1) ...
Preparing to unpack .../2-libnetplan0_0.99-0ubuntu3~18.04.3_amd64.deb ...
Unpacking libnetplan0:amd64 (0.99-0ubuntu3~18.04.3) over (0.99-0ubuntu3~18.04.2) ...
Preparing to unpack .../3-netplan.io_0.99-0ubuntu3~18.04.3_amd64.deb ...
Unpacking netplan.io (0.99-0ubuntu3~18.04.3) over (0.99-0ubuntu3~18.04.2) ...
Preparing to unpack .../4-nplan_0.99-0ubuntu3~18.04.3_all.deb ...
Unpacking nplan (0.99-0ubuntu3~18.04.3) over (0.99-0ubuntu3~18.04.2) ...
Preparing to unpack .../5-libdrm-common_2.4.101-2~18.04.1_all.deb ...
Unpacking libdrm-common (2.4.101-2~18.04.1) over (2.4.99-1ubuntu1~18.04.2) ...
Preparing to unpack .../6-libdrm2_2.4.101-2~18.04.1_amd64.deb ...
Unpacking libdrm2:amd64 (2.4.101-2~18.04.1) over (2.4.99-1ubuntu1~18.04.2) ...
Setting up libdrm-common (2.4.101-2~18.04.1) ...
Setting up libnetplan0:amd64 (0.99-0ubuntu3~18.04.3) ...
Setting up libdbus-1-3:amd64 (1.12.2-1ubuntu1.2) ...
Setting up netplan.io (0.99-0ubuntu3~18.04.3) ...
Setting up libdrm2:amd64 (2.4.101-2~18.04.1) ...
Setting up nplan (0.99-0ubuntu3~18.04.3) ...
Setting up dbus (1.12.2-1ubuntu1.2) ...
A reboot is required to replace the running dbus-daemon.
Please reboot the system when convenient.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.41) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
ubuntu@servidor:~$

```

Inserta el siguiente comando para instalar el servidor proxy **squid**

```
sudo apt install vim squid -y
```

► Click aquí para ver la salida del comando anterior

```

vim set to manually installed.
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libdbi-perl libecap3 libltdl7 squid-common squid-langpack ssl-cert
Suggested packages:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl squidclient squid-cgi squid-purge resolvconf smbclient winbind openssl-blacklist
The following NEW packages will be installed:
  libdbi-perl libecap3 libltdl7 squid squid-common squid-langpack ssl-cert
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 3329 kB of archives.
After this operation, 13.0 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libecap3 amd64 1.0.1-3.2 [16.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 libltdl7 amd64 2.4.6-2 [38.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 squid-langpack all 20170901-1 [137 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 squid-common all 3.5.27-1ubuntu1.6 [177 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 libdbi-perl amd64 1.640-1 [724 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 ssl-cert all 1.0.39 [17.0 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 squid amd64 3.5.27-1ubuntu1.6 [2218 kB]
Fetched 3329 kB in 2s (1658 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libecap3:amd64.
(Reading database ... 60098 files and directories currently installed.)
Preparing to unpack .../0-libecap3_1.0.1-3.2_amd64.deb ...
Unpacking libecap3:amd64 (1.0.1-3.2) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../1-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Selecting previously unselected package squid-langpack.
Preparing to unpack .../2-squid-langpack_20170901-1_all.deb ...
Unpacking squid-langpack (20170901-1) ...
Selecting previously unselected package squid-common.
Preparing to unpack .../3-squid-common_3.5.27-1ubuntu1.6_all.deb ...
Unpacking squid-common (3.5.27-1ubuntu1.6) ...
Selecting previously unselected package libdbi-perl.
Preparing to unpack .../4-libdbi-perl_1.640-1_amd64.deb ...
Unpacking libdbi-perl (1.640-1) ...
Selecting previously unselected package ssl-cert.
Preparing to unpack .../5-ssl-cert_1.0.39_all.deb ...
Unpacking ssl-cert (1.0.39) ...
Selecting previously unselected package squid.
Preparing to unpack .../6-squid_3.5.27-1ubuntu1.6_amd64.deb ...
Unpacking squid (3.5.27-1ubuntu1.6) ...
Setting up ssl-cert (1.0.39) ...
Setting up libecap3:amd64 (1.0.1-3.2) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Setting up squid-langpack (20170901-1) ...
Setting up squid-common (3.5.27-1ubuntu1.6) ...
Setting up libdbi-perl (1.640-1) ...
Setting up squid (3.5.27-1ubuntu1.6) ...
Setcap worked! /usr/lib/squid/pinger is not suid!
Skipping profile in /etc/apparmor.d/disable: usr.sbin.squid
Processing triggers for systemd (237-3ubuntu10.41) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
ubuntu@servidor:~$

```

Inserta el siguiente comando para eliminar los paquetes innecesarios del servidor

```
sudo apt autoremove -y
```

► Click aquí para ver la salida del comando anterior

```

ubuntu@servidor:~$ sudo apt autoremove -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  grub-pc-bin
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 2987 kB disk space will be freed.
(Reading database ... 62602 files and directories currently installed.)
Removing grub-pc-bin (2.02-2ubuntu8.15) ...
ubuntu@servidor:~$

```

Inserta el siguiente comando para reiniciar el servidor

```
sudo reboot
```

► Click aqui para ver la salida del comando anterior

```
ubuntu@servidor:~$ sudo reboot
bash-5.0$
```

Inserta el siguiente comando para forzar el inicio del servidor con **multipass**

```
multipass start servidor
```

► Click aqui para ver la salida del comando anterior

```
bash-5.0$ multipass start servidor
bash-5.0$
```

Configurando Squid en el servidor

Primeramente hay que entrar al servidor con el siguiente comando:

```
multipass shell servidor
```

► Click aqui para ver la salida del comando anterior

```
bash-5.0$ multipass shell servidor
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-106-generic x86_64)

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

System information as of Thu Jun 18 01:22:21 CDT 2020

System load:  0.0               Processes:            90
Usage of /:   21.0% of 4.67GB   Users logged in:     0
Memory usage: 11%              IP address for enp0s2: 192.168.64.22
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@servidor:~$
```

Una vez dentro procederemos a escribir el siguiente comando en terminal para ver el estado del servidor proxy **Squid**


```
sudo systemctl status squid
```

Es muy importante que la salida del comando anterior sea algo muy similar a la imagen de abajo, de lo contrario no podras proseguir con los pasos siguientes.

► Click aqui para ver la salida del comando anterior

```
ubuntu@servidor:~$ sudo systemctl status squid
● squid.service - LSB: Squid HTTP Proxy version 3.x
   Loaded: loaded (/etc/init.d/squid; generated)
   Active: active (running) since Thu 2020-06-18 01:30:46 CDT; 6min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 784 ExecStart=/etc/init.d/squid start (code=exited, status=0/SUCCESS)
    Tasks: 4 (limit: 1152)
   CGroup: /system.slice/squid.service
           └─888 /usr/sbin/squid -YC -f /etc/squid/squid.conf
             └─890 (squid-1) -YC -f /etc/squid/squid.conf
               └─892 (logfile-daemon) /var/log/squid/access.log
                 └─895 (pinger)

Jun 18 01:30:44 servidor systemd[1]: Starting LSB: Squid HTTP Proxy version 3.x...
Jun 18 01:30:45 servidor squid[784]: * Starting Squid HTTP Proxy squid
Jun 18 01:30:46 servidor squid[784]: ...done.
Jun 18 01:30:46 servidor systemd[1]: Started LSB: Squid HTTP Proxy version 3.x.
Jun 18 01:30:46 servidor squid[888]: Squid Parent: will start 1 kids
Jun 18 01:30:46 servidor squid[888]: Squid Parent: (squid-1) process 890 started
ubuntu@servidor:~$
```

Escribe el siguiente comando para copiar el archivo original de configuracion de **Squid**, en pocas palabras hacer un respaldo

```
sudo cp /etc/squid/squid.conf{,.original}
```

Configurando la Autenticacion

Squid cuenta con varios metodos de autenticacion, a continuacion configuraremos el tipo de autenticacion basica que consta de un usuario y un password

Escribiremos el siguiente comando en la terminal, el cual creara un usuario llamado **test** con un password **123** que estara encriptado en el archivo **/etc/squid/htpasswd**

```
printf "test:${(openssl passwd -crypt '123')}\n" | sudo tee -a
/etc/squid/htpasswd
```

► Click aqui para ver la salida del comando anterior

```
ubuntu@servidor:~$ printf "test:${(openssl passwd -crypt '123')}\n" | sudo tee -a /etc/squid/htpasswd
test:VJ0/ukhzxDHto
ubuntu@servidor:~$
```

Ahora procederemos a buscar la línea en la cual está ubicada el siguiente párrafo `#http_access allow localnet` en el archivo de configuración de `Squid` ubicado en `/etc/squid/squid.conf`

```
cat -n /etc/squid/squid.conf | grep "#http_access allow localnet"
```

► Click aquí para ver la salida del comando anterior

```
ubuntu@servidor:~$ cat -n /etc/squid/squid.conf | grep "#http_access allow localnet"
1190 #http_access allow localnet
ubuntu@servidor:~$
```

Teniendo en cuenta la línea del comando anterior, abrimos el archivo de configuración `/etc/squid/squid.conf` con el editor `vim` y buscaremos esa línea en particular

```
sudo vim /etc/squid/squid.conf
```

► Click aquí para ver la salida del comando anterior


```
#Default:
# Deny, unless rules exist in squid.conf.
#
#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
#http_access allow localnet
http_access allow localhost

# And finally deny all other access to this proxy
http_access deny all
```

Teniendo el archivo abierto insertaremos las siguientes líneas de código inmediatamente debajo del texto `# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS`

```
acl web_prohibidas dstdomain "/etc/squid/web_prohibidas.txt"
acl palabras_prohibidas url_regex "/etc/squid/palabras_prohibidas.txt"
auth_param basic program /usr/lib/squid3/basic_ncsa_auth
/etc/squid/htpasswd
auth_param basic realm proxy
acl authenticated proxy_auth REQUIRED
```

De igual manera inmediatamente después del texto `http_access allow localhost` insertamos las siguientes líneas

```
http_access deny web_prohibidas
http_access deny palabras_prohibidas
http_access allow authenticated
```

► Click aqui para ver la salida del comando anterior

```
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
acl web_prohibidas dstdomain "/etc/squid/web_prohibidas.txt"
acl palabras_prohibidas url_regex "/etc/squid/palabras_prohibidas.txt"
auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/squid/htpasswd
auth_param basic realm proxy
acl authenticated proxy_auth REQUIRED
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
#http_access allow localnet
http_access allow localhost
http_access deny web_prohibidas
http_access deny palabras_prohibidas
http_access allow authenticated
# And finally deny all other access to this proxy
http_access deny all
# TAG: adapted_http_access
```

Nota: el orden es muy importante, de manera contrario no funcionara el servidor proxy adecuadamente!

Ahora crearemos los archivos `web_prohibidas.txt` y `palabras_prohibidas.txt` que como su nombre lo infiere, aqui se pondran todas aquellas webs a las que no queremos que los usuarios tengan acceso asi mismo como palabras especificas para bloquear sitios indeseados.

Ejemplo para bloquear `facebook.com` y `youtube.com`

```
echo -ne ".facebook.com\n.youtube.com\n" | sudo tee
/etc/squid/web_prohibidas.txt
```

► Click aqui para ver la salida del comando anterior

```
ubuntu@servidor:~$ echo -ne ".facebook.com\n.youtube.com\n" | sudo tee /etc/squid/web_prohibidas.txt
.facebook.com
.youtube.com
ubuntu@servidor:~$
```

Nota: los sitios a bloquear deben empezar con un punto al principio

Ejemplo para bloquear palabras claves en las URLs:

```
echo -ne "play\nsteam\n" | sudo tee /etc/squid/palabras_prohibidas.txt
```

► Click aqui para ver la salida del comando anterior

```
ubuntu@servidor:~$ echo -ne "play\nsteam\n" | sudo tee /etc/squid/palabras_prohibidas.txt
play
steam
ubuntu@servidor:~$
```

Ahora procederemos a reiniciar el servicio de **Squid** para que nuestros cambios realizados tomen efecto inmediato

```
sudo systemctl restart squid
```

► Click aqui para ver la salida del comando anterior

```
ubuntu@servidor:~$ sudo systemctl restart squid
ubuntu@servidor:~$
```

Nota: este comando puede tardar un poco

Y para corroborar que todos nuestros cambios surtieron el efecto deseado, insertaremos el siguiente comando

```
sudo systemctl status squid
```

► Click aqui para ver la salida del comando anterior

```
ubuntu@servidor:~$ sudo systemctl status squid
● squid.service - LSB: Squid HTTP Proxy version 3.x
   Loaded: loaded (/etc/init.d/squid; generated)
   Active: active (running) since Thu 2020-06-18 01:30:46 CDT; 6min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 784 ExecStart=/etc/init.d/squid start (code=exited, status=0/SUCCESS)
    Tasks: 4 (limit: 1152)
   CGroup: /system.slice/squid.service
           └─888 /usr/sbin/squid -YC -f /etc/squid/squid.conf
             └─890 (squid-1) -YC -f /etc/squid/squid.conf
               └─892 (logfile-daemon) /var/log/squid/access.log
                 └─895 (pinger)

Jun 18 01:30:44 servidor systemd[1]: Starting LSB: Squid HTTP Proxy version 3.x...
Jun 18 01:30:45 servidor squid[784]: * Starting Squid HTTP Proxy squid
Jun 18 01:30:46 servidor squid[784]: ...done.
Jun 18 01:30:46 servidor systemd[1]: Started LSB: Squid HTTP Proxy version 3.x.
Jun 18 01:30:46 servidor squid[888]: Squid Parent: will start 1 kids
Jun 18 01:30:46 servidor squid[888]: Squid Parent: (squid-1) process 890 started
ubuntu@servidor:~$
```

Nota: si la salida del comando anterior no es igual a la imagen, significa que algo tienes mal en la configuracion y no podras avanzar hasta que lo resuelvas

Habilitando el firewall

Por defecto el firewall de Ubuntu viene deshabilitado, lo tendremos que habilitar para que trabaje en conjunto con nuestro servidor proxy **Squid**

Insertamos el siguiente comando para habilitar el firewall

```
sudo ufw enable
```

► [Click aqui para ver la salida del comando anterior](#)

```
ubuntu@servidor:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ubuntu@servidor:~$
```

Insertamos el siguiente comando para permitir el uso de **Squid** en el firewall

```
sudo ufw allow 'Squid'
```

► [Click aqui para ver la salida del comando anterior](#)

```
ubuntu@servidor:~$ sudo ufw allow 'Squid'
Rule added
Rule added (v6)
ubuntu@servidor:~$
```

Insertamos el siguiente comando para ver si **Squid** fue correctamente habilitado

```
sudo ufw status
```

► [Click aqui para ver la salida del comando anterior](#)

```
ubuntu@servidor:~$ sudo ufw status
Status: active

To Action From
--
Squid ALLOW Anywhere
Squid (v6) ALLOW Anywhere (v6)

ubuntu@servidor:~$
```

Nota: si no tienes habilitado **Squid** en el firewall de Ubuntu no podras continuar hasta que lo habilites!

Configurando tu navegador para usar proxy

En este punto **Squid** esta funcionando y solo queda configurar nuestro navegador para usar el servidor proxy,

Como pre-requisito tendras que descubrir cual es la ip del servidor con el siguiente comando

```
multipass ls
```

► Click aqui para ver la salida del comando anterior

```
bash-5.0$ multipass ls
Name          State      IPv4          Image
servidor      Running   192.168.64.22 Ubuntu 18.04 LTS
bash-5.0$
```

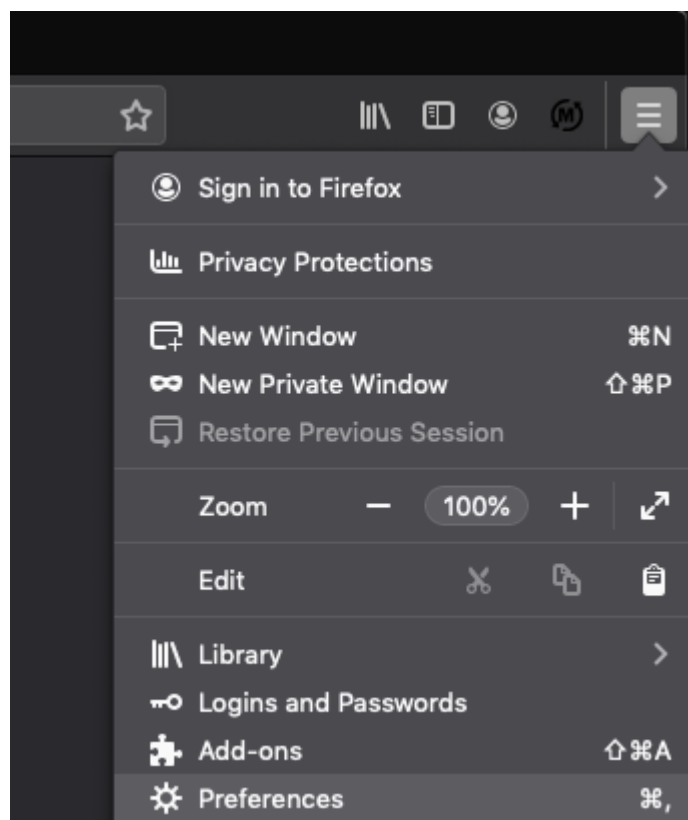
Nota: la ip del servidor la necesitaremos para configurarlas en los navegadores

Ve a la seccion acorde al navegador que tengas instalado en tu sistema

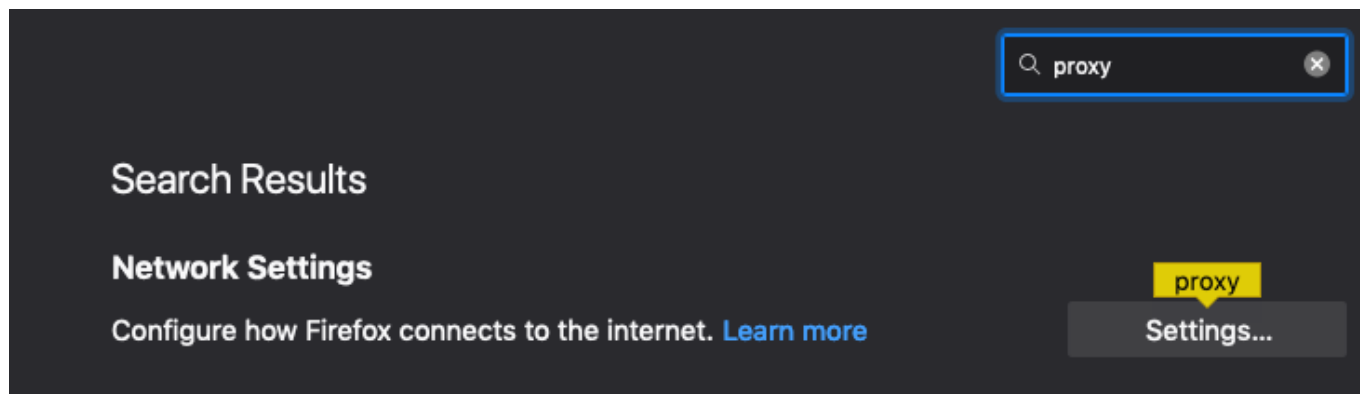
Firefox

► Click aqui para ver la configuracion de Firefox

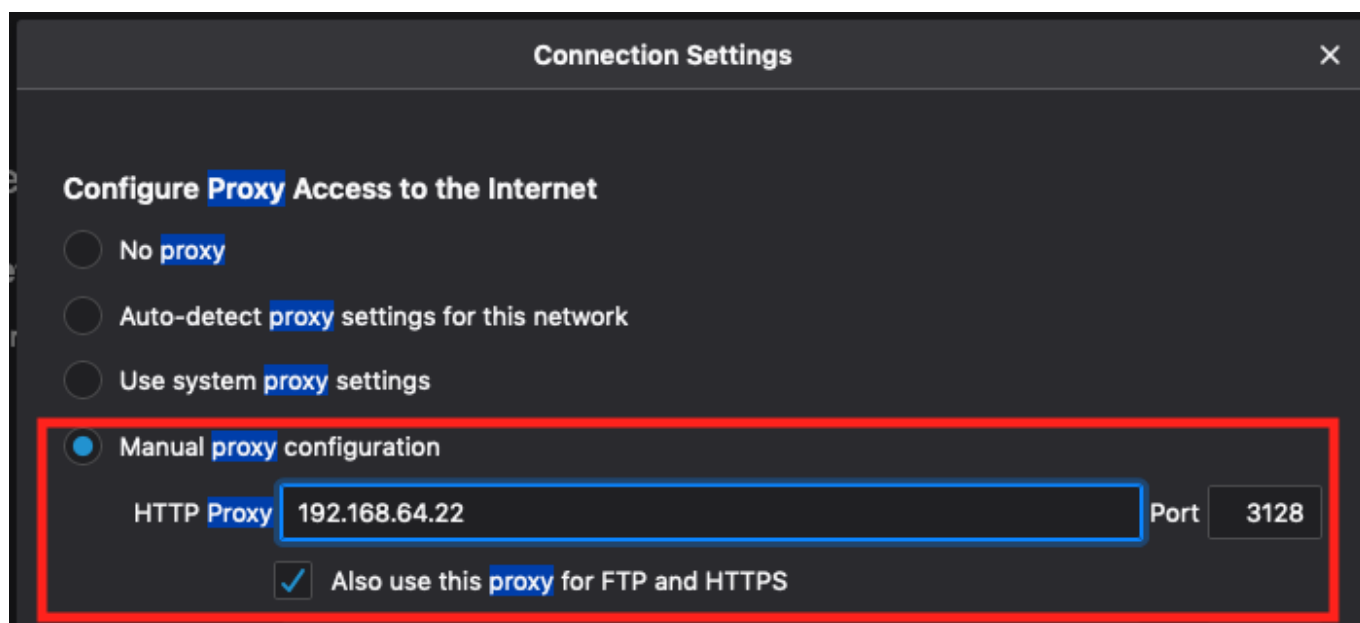
1. Abrir las preferencias de **Firefox** como se muestra en la imagen a continuacion



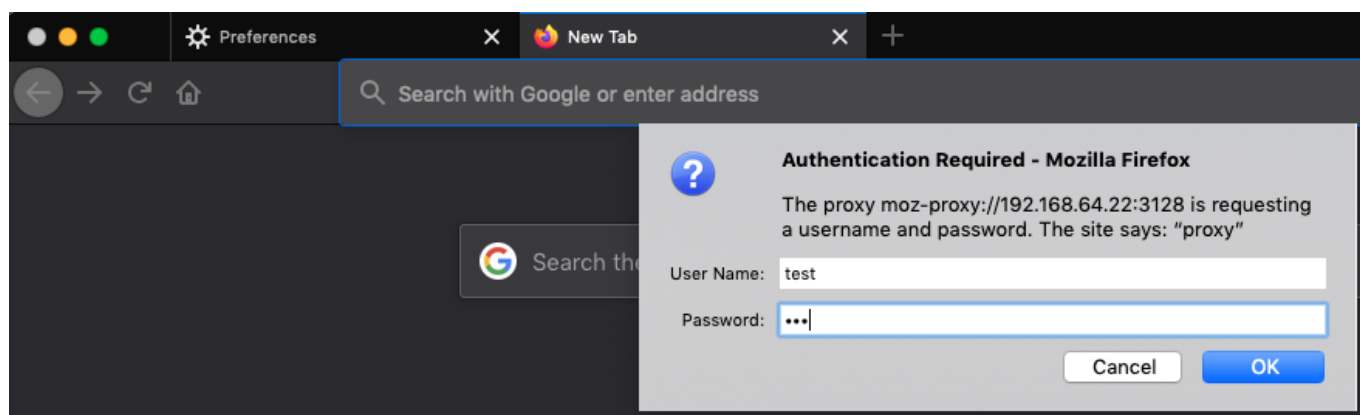
2. En el buscador insertar la palabra **proxy** y hacer click en la opcion de **Settings...**



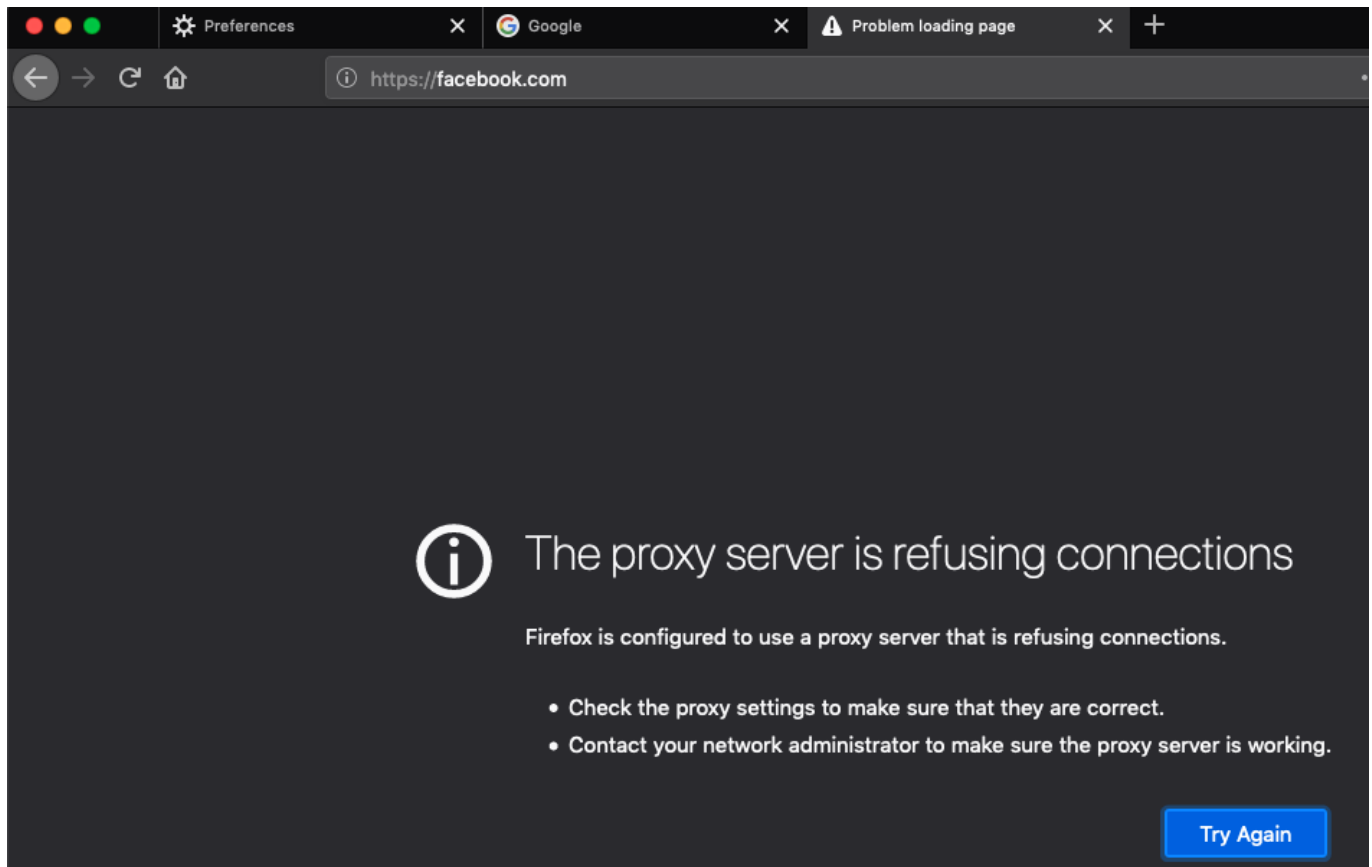
3. Selecciona "Manual proxy configuration" y pon la ip del servidor proxy (**Squid**) asi como el puerto como se muestra a continuacion



4. Abre una nueva pestaña de firefox y notarás que te pedirá credenciales, inserta las credenciales creadas anteriormente (User Name = test) (Password = 123)



5. Trata de acceder a las web prohibidas o inserta una palabra prohibida para verificar el funcionamiento del servidor proxy



Autores

- Humberto Israel Perez Rodriguez

Fuentes:

- [How to install and configure squid proxy on ubuntu 18.04](#)
- [Instalación y configuración básica de Squid en Ubuntu 18.04](#)