No 2
MAR 2019
DISTRIBUCION
ONLINE
GRATUITA

USA, ESTUDIA, DISTRIBUYE Y MEJORA

MAGAZINE

SOLOLINUX

VISITA NUESTRO SITIO SOLOLINUX.ES

MANUALES
SCRIPTS
SOFTWARE
HARDWARE
DISTROS LINUX
SEGURIDAD
REDES
Y MUCHO MAS EN LA WEB

INSTALAR Y CONFIGURAR CENTOS WEB PANEL – CWP

NUEVA DISTRIBUCIÓN TAILS 3.13

SoloLinux

"Si compila esta bien, si arranca es perfecto."

LINUX TORVALDS

SIGUENOS EN:

SoloLinux

VISITANOS

Visítanos en www.sololinux.es

DISEÑO

Adrián A. A.
adrian@sololinux.es

INFO

Solo Linux Magazine
No. 2
Marzo 2019
108 Paginas

PUBLICIDAD

Quieres aparecer en la revista, escríbenos a
adrian@sololinux.es

REDACCIÓN

Sergio G. B.
info@sololinux.es

COLABORA

Quieres colaborar con algún articulo, mándalo a
adrian@sololinux.es
Y si gusta será publicado.

AGRADECIMIENTOS

Quiero agradecer a **Sergio G.B. Admin de SOLOLINUX.ES** que me dejara usar sus recursos para poder crear esta revista, gracias a el estamos en el segundo numero de muchos, o eso espero :P

Quiero agradecer los ánimos que mucha gente del foro **GNULINUXVAGOS** para que esto siga adelante.
Muchas gracias a todos y nos vemos en el tercer numero. Disfruten con este segundo número de marzo.

Adrián A. A.

Revista de distribución gratuita, comparte conocimientos.

SOLOLINUX

INSTITUTO LINUX

¿Quieres trabajar en las mejores empresas del sector IT?

Certifica LINUX

CURSO Linux Foundation + VOUCHER Linux Foundation
+ Curso LSA CLA (EN VIVO - inicia 20 DE ABRIL)
+ COCHING y MENTORÍA de Fabián Ampalio + TUTORÍAS on line

CURSO LINUX FOUNDATION + **VOUCHER** LINUX FOUNDATION
+ **CURSO** LSA CLA (EN VIVO - INICIA 20 DE ABRIL)
+ COCHING Y MENTORÍA DE FABIÁN AMPALIO + TUTORÍAS ON LINE

50% OFF

TODO POR SOLO
U$S **249.50**
Valor real de combo ~~U$S 998~~

50% OFF
CUPÓN

¿TE LO VAS A PERDER?

Encontrá toda la información aquí:
www.linuxadistancia.com

@exameneslinux @aprender_linux

INDICE

INDICE

THANKS!

TU PUBLICIDAD AQUI

QUIERES APARECER EN LA REVISTA, GANAR CON ELLO MAS VENTAS EN TU WEB, MAS SEGUIDORES EN TUS REDES SOCIALES...

SOLO TIENES QUE MANDAR UN CORREO A

adrian@sololinux.es

Y TE EXPLICAMOS COMO

SoloLinux

www.sololinux.es

Modificar el propietario y el grupo con chown y chgrp

Modificar el propietario y el grupo de un archivo o carpeta con **chown** y **chgrp**.

Como continuación de la serie de artículos "**Permisos de archivos en Linux**", hoy veremos como modificar el propietario y el grupo de un archivo o carpeta con **chown** y **chgrp**.

Recordemos todos los artículos de la saga:

- **Permisos de archivos en Linux** (presentación).
- **Uso del comando chmod**.
- **Uso de los comandos chown y chgrp**.
- **Uso del comando chattr**.

El uso de estos dos **comandos** es bastante simple, así que no vamos a explayarnos en explicaciones.

Modificar el propietario de un archivo con chown

El comando **chown** nos permite alterar el propietario de un archivo o carpeta (directorio).

La sintaxis de "**chown**" es la siguiente:

chown <USUARIO>[:<GRUPO>] [ARCHIVO]

Insertamos un par de ejemplos para que lo tengas más claro.

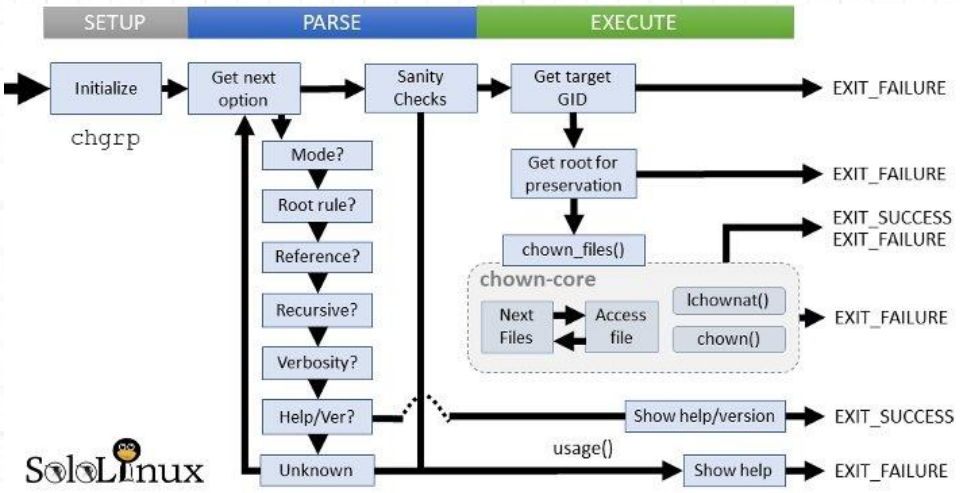
En el primer ejemplo tenemos el archivo "**linux-para-todos.txt**", y queremos que su nuevo propietario sea el usuario "**sololinux**", hacemos lo siguiente:

chown sololinux linux-para-todos.txt

En el segundo ejemplo establecemos el propietario del archivo "**linux-para-todos.txt**" al usuario "**sololinux**", y configuramos el propietario del grupo en "**root**".

chown sololinux:root linux-para-todos.txt

Como ves es muy fácil. Pasamos al siguiente comando.



Modificar el grupo de usuarios de un archivo con chqrp

Con el comando **chgrp** podemos cambiar el grupo de usuarios de un archivo o directorio (carpeta).

La sintaxis de "**chgrp**" es la siguiente:

chgrp <GRUPO> [ARCHIVO]

En el siguiente ejemplo establecemos el grupo de usuarios "**staff**" al archivo "**linux-para-todos.txt**". Debes tener presente que a partir de ahora todos los usuarios del grupo "**staff**", tendrán privilegios en el archivo "**linux-para-todos.txt**".

chgrp staff linux-para-todos.txt

Si especificamos la opción "**-R**", modificamos el grupo de forma **recursiva** en todos los archivos del directorio o carpeta que hayas especificado.

Por ejemplo, configuramos a todos los usuarios del grupo "**staff**", en todos los archivos del directorio "**/home/sololinux**".

chgrp -R staff /home/sololinux

Si te resulto útil el articulo, entra en la WEB y compártelo.

Proteger archivos y carpetas con el comando Chattr

Proteger archivos y carpetas con el **comando Chattr**.

Como punto final a la serie de artículos "Permisos de archivos en Linux", vamos a ver el comando "**Chattr**", que lleva el hecho de "**proteger archivos y carpetas**" a una seguridad extrema.

Chattr al igual que otros comandos, tiene la propiedad de modificar los atributos de un archivo o directorio en un sistema Linux.

Pero tiene una particularidad y es que brinda el mayor nivel de seguridad en archivos y directorios.

También puedes aprovechar esta seguridad extrema para evitar que archivos importantes se eliminen accidentalmente, incluso siendo **root**.

Antes de comenzar con el comando **Chattr**, recordamos los artículos de la serie:

- **Permisos de archivos en Linux** (presentación).
- **Uso del comando chmod**.
- **Uso de los comandos chown y chgrp**.
- **Uso del comando chattr**.

Proteger archivos y carpetas con el comando Chattr

La sintaxis de "**chattr**" es la siguiente:

chattr [-RVf] [-v version] [mode] archivo

Continuamos…

Antes de ver algunos ejemplos de uso, es importante conocer los parámetros del comando.

Parámetros de Chattr

Los dividimos en tres partes:

1. Opcionales.
2. Operadores.
3. Atributos.

Parámetros de opción:

- **V –** Imprime en pantalla, información y detalles ampliados del comando.
- **R –** Indica una forma **recursiva** a los atributos de las carpetas seleccionadas y a su contenido.
- **v –** Imprime en pantalla la versión utilizada.

Parámetros de operadores:

- **+ –** Agrega atributos a los que ya existen.
- **– –** Reduce o elimina los permisos existentes.
- **= –** Sustituye los atributos que tiene el archivo por los que le indiques.

Parámetros de los atributos (los más usados)**:**

- **a** – Solo permite agregar contenido en el archivo.
- **A** – Que la fecha del último acceso no sea modificada.
- **c** – El archivo se comprimirá automáticamente.
- **d** – No se permite **backups** del archivo o carpeta con **dump**.
- **i** – Este atributo hará el **archivo inmutable**, para que me entiendas bien, no se podrá eliminar, tampoco renombrar, ni apuntar a enlaces simbólicos, y mucho menos insertar datos en el archivo o carpeta. **Ojo!!! con su uso**, el **root** tampoco tendrá permisos (al final del articulo vemos como anular esta protección).
- **S** – Las modificaciones del archivo se escribirán en el disco de forma sincrónica.
- **s** – Los bloques que usa el archivo serán re-escritos con ceros (0), el fichero será irrecuperable, es como si ese espacio hubiese sido **formateado a bajo nivel** (de fabrica).
- **u** – Al eliminar un archivo, su contenido quedara guardado por si lo quieres recuperar después con alguna herramienta especializada.

Ejemplos de uso del comando Chattr

Ejemplos de como hacer que un archivo o carpeta sea "**inmutable**", con "**+ i** ".

chattr +i sololinux.txt

chattr +i carpeta

Usaremos "**+ a**" para que solo permita adjuntar contenido.

chattr +a sololinux.txt

Con la opción "**–**" podemos eliminar atributos, incluso los del archivo inmutable.

chattr -i sololinux.txt

chattr -a linuxparatodos.txt

***En una carpeta de forma recursiva.

chattr -R -ai carpeta

Y con el **comando "chattr"** damos por concluida la serie de artículos "**Permisos de archivos en Linux**". Si crees que son de utilidad, entra en la WEB y compártelos.

Instalar Let's Encrypt en Nginx con Ubuntu 18.04

Instalar **Let's Encrypt en Nginx** con **Ubuntu 18.04**.

Let's Encrypt es una entidad emisora de certificados gratuitos ademas de **open source** (Github). El certificado fue desarrollado por la **Internet Security Research Group** (**ISRG**).

La inmensa mayoría de navegadores web, confían en los certificados emitidos por Let's Encrypt.

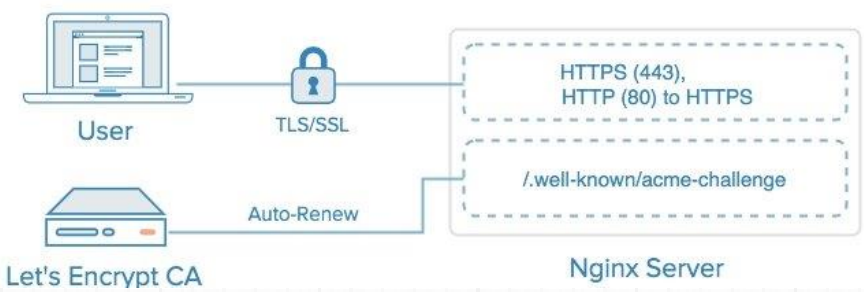
En este articulo veremos paso a paso cómo asegurar / instalar **Let's Encrypt en Nginx** con Ubuntu 18.04. Para conseguir una integración perfecta utilizaremos la herramienta "**Cerbot**".

Instalar Let's Encrypt en Nginx con Ubuntu 18.04
Instalar Cerbot
Lo primero que debemos hacer es **instalar Certbot**, que es una herramienta muy completa y fácil de usar.
Con **Certbot** podemos automatizar las tareas de obtener y renovar los **certificados SSL Let's Encrypt**, además de configurar los servidores web para su uso. La **herramienta Certbot** está incluida en los repositorios oficiales de **Ubuntu 18.04**.
Actualizamos el sistema e instalamos **Cerbot**.

```
sudo apt update
sudo apt install certbot
```

Una vez instalado, generamos un conjunto de **parámetros DH** (Diffie–Hellman key exchange) de 2048 bits para fortalecer la seguridad:

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```



Obtener el certificado Let's Encrypt

Para obtener el **certificado SSL** de nuestro dominio, vamos a utilizar el plugin **Webroot**, la función del plugin es crear un archivo temporal para validar el dominio solicitado, en: ${webroot-path}/.well-known/acme-challenge

El **servidor Let's Encrypt** hará las solicitudes HTTP directamente en el archivo temporal, de esta forma se podrá confirmar que el dominio solicitado resuelve en el servidor que ejecuta **Certbot**.
Para que no haya equívocos, asignamos todas las solicitudes http .well-known/acme-challenge a un solo directorio, /var/lib/letsencrypt.

```
mkdir -p /var/lib/letsencrypt/.well-known
chgrp www-data /var/lib/letsencrypt
chmod g+s /var/lib/letsencrypt
```

Para evitar duplicación de código creamos dos archivos.

```
sudo nano /etc/nginx/snippets/letsencrypt.conf
```

Copia y pega lo siguiente:

```
location ^~ /.well-known/acme-challenge/ {
  allow all;
  root /var/lib/letsencrypt/;
  default_type "text/plain";
  try_files $uri =404;
}
```

Guarda el archivo y cierra el editor.
Creamos el segundo archivo.

```
sudo nano /etc/nginx/snippets/ssl.conf
```

Copia y pega lo siguiente:

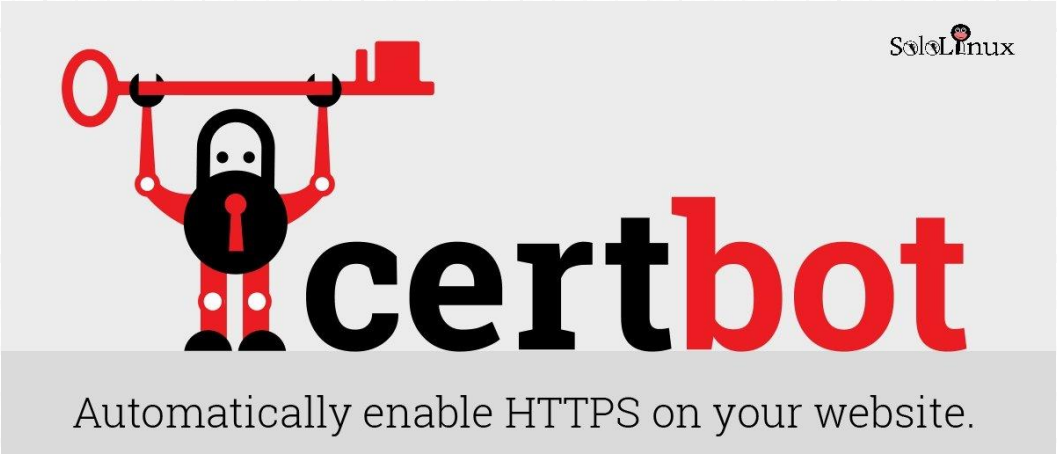
```
ssl_dhparam /etc/ssl/certs/dhparam.pem;

ssl_session_timeout 1d;
ssl_session_cache shared:SSL:50m;
ssl_session_tickets off;

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:DES-CBC3-SHA:!DSS';
ssl_prefer_server_ciphers on;

ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 30s;

add_header Strict-Transport-Security "max-age=15768000; includeSubdomains; preload";
add_header X-Frame-Options SAMEORIGIN;
add_header X-Content-Type-Options nosniff;
```

Guarda el archivo y cierra el editor.
Lo agregamos a nuestro dominio.
sudo nano /etc/nginx/sites-available/tudominio.com

```
server {
  listen 80;
  server_name tudominio.com www.tudominio.com;

  include snippets/letsencrypt.conf;
}
```

Guarda y cierra el editor, porque ahora debemos crear un **enlace simbólico** para que al iniciar "**Nginx**" lo pueda leer.
sudo ln -s /etc/nginx/sites-available/tudominio.com /etc/nginx/sites-enabled/

Reiniciamos Nginx.
sudo systemctl restart nginx



Ejecutamos **Certbot** con el **plugin webroot** para obtener los archivos del **certificado SSL** (introduce tus datos).
ejemplo de respuesta valida…

IMPORTANT NOTES:
– Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/example.com/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/example.com/privkey.pem
Your cert will expire on 2018-07-28. To obtain a new or tweaked
version of this certificate in the future, simply run certbot
*again. To non-interactively renew *all* of your certificates, run*
"certbot renew"
– Your account credentials have been saved in your Certbot
configuration directory at /etc/letsencrypt. You should make a
secure backup of this folder now. This configuration directory will
also contain certificates and private keys obtained by Certbot so
making regular backups of this folder is ideal.
– If you like Certbot, please consider supporting our work by:
Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
Donating to EFF: https://eff.org/donate-le

Ya tenemos los archivos del certificado, solo nos falta agregarlos al dominio.
sudo nano /etc/nginx/sites-available/tudominio.com

Edita el archivo (con tus datos), como te indico en el siguiente código:

```
server {
    listen 80;
    server_name www.tudominio.com tudominio.com;

    include snippets/letsencrypt.conf;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl http2;
    server_name www.tudominio.com;

    ssl_certificate /etc/letsencrypt/live/tudominio.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/tudominio.com/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/tudominio.com/chain.pem;
    include snippets/ssl.conf;
    include snippets/letsencrypt.conf;

    return 301 https://tudominio.com$request_uri;
}

server {
    listen 443 ssl http2;
    server_name tudominio.com;

    ssl_certificate /etc/letsencrypt/live/tudominio.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/tudominio.com/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/tudominio.com/chain.pem;
    include snippets/ssl.conf;
    include snippets/letsencrypt.conf;

    # . . . el resto del archivo déjalo como está.
}
```

Guarda el archivo y cierra el editor.
Recargamos "**Nginx**".

```
sudo systemctl reload nginx
```

Renovar automáticamente Let's Encrypt

Los certificados **Let's Encrypt** tienen una validez de 90 días. Para renovar automáticamente los certificados antes de que caduquen, usando la herramienta **Certbot** creamos una **tarea cron** para que se ejecute dos veces al día, así, siempre mantendremos el o los certificados renovados.

```
sudo nano /etc/cron.d/certbot
```

Copia y pega:

```
0 */12 * * * root test -x /usr/bin/certbot -a \! -d /run/systemd/system && perl -e 'sleep int(rand(3600))' && certbot -q renew --renew-hook "systemctl reload nginx"
```

Podemos comprobar si renueva ejecutando lo siguiente:

```
sudo certbot renew --dry-run
```

Si no lanza ningún error, el proceso fue exitoso.

Felicidades!!! ya tienes instalado **Let's Encrypt en Nginx** con **Ubuntu 18.04**.
Comparte el articulo "**Instalar Let's Encrypt en Nginx con Ubuntu 18.04**".

THANKS!

TU PUBLICIDAD AQUI

QUIERES APARECER EN LA REVISTA, GANAR CON ELLO MAS VENTAS EN TU WEB, MAS SEGUIDORES EN TUS REDES SOCIALES...

SOLO TIENES QUE MANDAR UN CORREO A

adrian@sololinux.es

Y TE EXPLICAMOS COMO

SoloLinux

www.sololinux.es

Uso del comando Systemctl con ejemplos

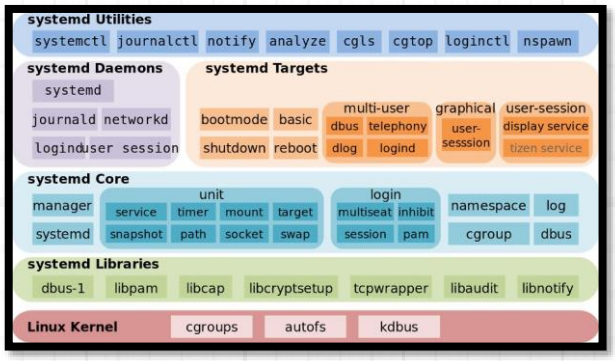
Uso del **comando Systemctl** con ejemplos.

El **comando systemctl** es una herramienta que sirve para poder controlar el sistema y sus servicios. **Systemctl** es el reemplazo natural de la obsoleta herramienta de administración que conocíamos como **SysVinit**.

La gran mayoría de los **sistemas operativos Linux** modernos ya usan este comando, que realmente es una herramienta que maneja un conjunto de demonios

Red Hat, CentOS, Ubuntu, Debian, Fedora, OpenSuse, Arch, Mageia, etc..., todos usan "systemd" en sus ultimas versiones.

Como anécdota... si estas utilizando **CentOS 6**, o **Debian7**, aun no esta integrada. Por tanto si no conoces muy bien su uso, hoy lo aprenderás.



Uso del comando Systemctl

El servidor que usamos para los ejemplos monta **CentOS7**, por tanto hace uso de **MariaDB** que es el servicio que tomamos para la presentación (exceptuando una imagen donde también vemos el status de **memcached**).

La sintaxis es la siguiente:

systemctl [OPCIÓN] [SERVICIO]

Ahora veremos los usos más comunes en tus tareas diarias, recuerda que debes sustituir el servicio "**mariadb**", por el que necesites tu.

Iniciar o detener el servicio

Iniciar: systemctl start mariadb.service

Detener: systemctl stop mariadb.service

```
[root@[redacted] ~]# systemctl stop mariadb.service
[root@[redacted] ~]# systemctl start mariadb.service
[root@[redacted] ~]#
```

Reiniciar o recargar el servicio

Reiniciar: systemctl restart mariadb.service

Recargar (reload): systemctl reload mariadb.service

Recargar o reiniciar: En este comando combinamos las dos opciones, primero intentara recargar el servicio y si no es posible lo reiniciara.

systemctl reload-or-restart mariadb.service

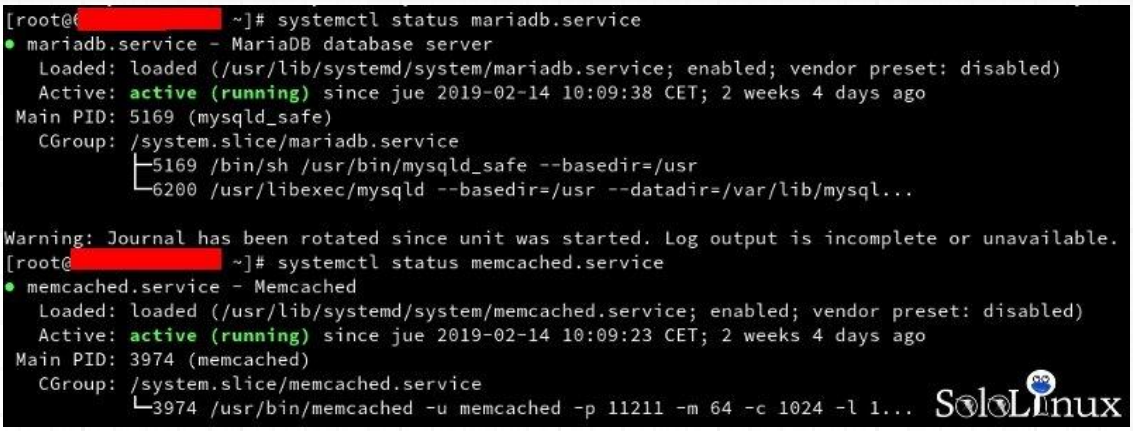
```
[root@[redacted] ~]# systemctl restart mariadb.service
[root@[redacted] ~]# systemctl reload-or-restart mariadb.service
[root@[redacted] ~]#
```

Estado del servicio

Estado del servicio: Con la opción estado del servicio (status), verificamos si un servicio esta corriendo o no. En el ejemplo comprobamos "**MariDB**" y "**Memcached**".

systemctl status mariadb.service

systemctl status memcached.service



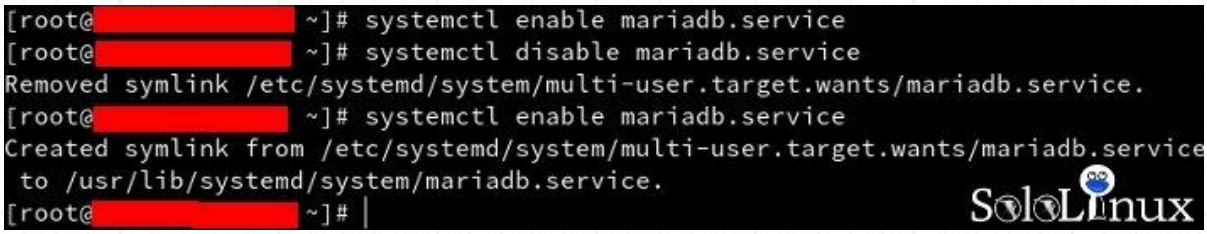
Habilitar o inhabilitar un servicio al iniciar el sistema

- **Habilitar:** Permitir que el servicio se inicie en el arranque del sistema.

  systemctl enable mariadb.service

- **Inhabilitar:** No permite que el servicio inicie en el arranque del sistema.

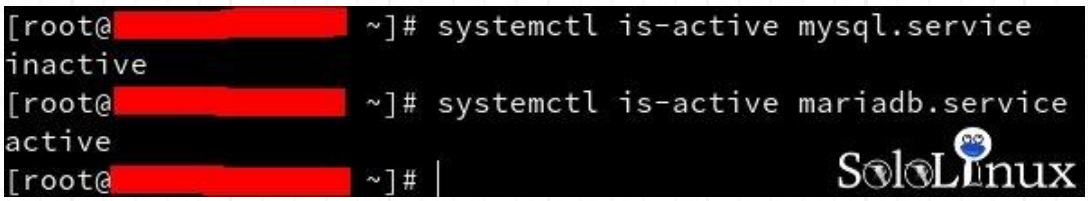
  systemctl disable mariadb.service



Comprobar si un servicio está activado o habilitado

- **Activado:** Verifica que un servicio este activado.
  - systemctl is-active mariadb.service
- **Habilitado:** Verifica que un servicio esta habilitado y arrancara con el sistema.
  - systemctl is-enabled mariadb.service

En el ejemplo y a modo de prueba comprobamos si tenemos activado "**MySQL**". Nosotros usamos "**MariaDB**" por tanto aparece como no activado.



Si este articulo te ayudo, entra en la WEB y compártelo.

Configurar tareas cron con crontab

Configurar tareas cron con crontab.

Puede parecer algo complicado, pero en realidad es todo lo contrario, **configurar tareas cron** es muy sencillo si sigues los pasos que te indicare en este articulo.

Crontab es un programador de tareas basado en tiempo. Cron es el archivo de configuración de **crontab** que especifica los **comandos shell** que se deben ejecutar según el periodo programado.

Los archivos de crontab se almacenan en la misma localización que las listas de tareas y otras instrucciones del **daemon cron**.

Normalmente los puedes encontrar en: /etc/crontab

Configurar tareas cron

Nosotros usaremos a modo de ejemplo un servidor con **CentOS7**, en otras distribuciones Linux el proceso es prácticamente el mismo.

Para hacer uso de esta herramienta necesitamos tener instalado el paquete "**cronie**", podemos comprobar si lo tenemos instalado con el siguiente comando:

sudo rpm -q cronie

Si está instalado recibirás una respuesta similar a... "***cronie-1.4.11-20.el7_6.x86_64***".

Si la respuesta no es valida, lo instalamos.

sudo yum install cronie

Una vez instalado comprobamos que se ejecuta, usamos el comando "**systemctl**".

sudo systemctl status crond.service

Ejemplo de que se ejecuta correctamente...

```
[[redacted]]# sudo systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since sáb 2019-03-02 07:12:33 CET; 3 days ago
 Main PID: 3736 (crond)
   CGroup: /system.slice/crond.service
           └─3736 /usr/sbin/crond -n

mar 05 06:07:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 06:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 07:07:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 07:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 08:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 09:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 10:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 11:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 12:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
mar 05 13:44:01 host.adminserver.es crond[3736]: (root) RELOAD (/var/spool/c...)
Hint: Some lines were ellipsized, use -l to show in full.
```

Todo correcto... verificamos la configuración actual de la herramienta.

sudo cat /etc/crontab

Como puedes ver en la siguiente imagen **crontab** ya contiene una explicación sobre cómo definir tus propias **tareas cron**.

```
[root@[redacted] etc]# cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

*/1 * * * * root /usr/local/rtm/bin/rtm 4 > /dev/null 2> /dev/null
```

La sintaxis que deber crear es (**ver los comandos útiles al final del articulo**):

minuto / hora / día / mes / día de la semana / usuario / comando

Se permite el uso de asteriscos (*) para especificar todos los valores válidos, por ejemplo si quieres que el comando o script seleccionado se ejecute todos los días a medianoche, agregas lo siguiente:

0 0 * * * root /libera.sh

Puedes usar las palabras reservadas en vez de números.

- @reboot: se ejecuta una única vez al inicio.
- @yearly/@annually: ejecutar cada año.
- @monthly: ejecutar una vez al mes.
- @weekly: una vez a la semana.
- @daily/@midnight: una vez al día.
- @hourly: cada hora.

Ejemplo…

@hourly /bin/libera.sh

También puedes crear tareas para usuarios específicos, en ese caso las tareas las localizaras en:

/var/spool/cron/usuario

Debes tener en cuenta que si creas una tarea para un usuario especifico, no debes colocar el nombre de usuario en la sintaxis.

minuto / hora / día / mes / día de la semana / comando

Una vez guardada la tarea debes reiniciar el servicio.

sudo systemctl restart crond.service

Comandos útiles

crontab -e: Editar o crear un archivo.
crontab -l: Lista el contenido de crontab.
crontab -r: Eliminar archivo.
man cron: Manual.
man crontab: Manual.

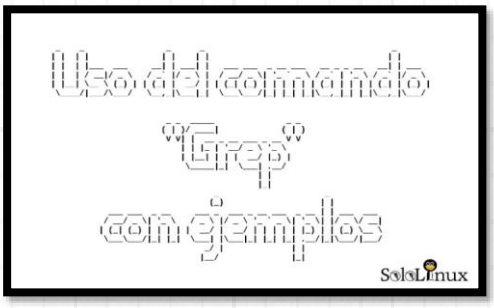
Si este articulo te ayudo, entra en la WEB y compártelo. Gracias por compartir. ☺☺☺

Uso del comando Grep con ejemplos

Uso del **comando Grep** con ejemplos.

El **comando grep** (impresión de la expresión regular global), se usa para buscar cadenas de texto y expresiones regulares línea por línea que coincidan con un patrón definido en uno o más archivos.

En este artículo, veremos con ejemplos, como utilizar eficazmente el **comando grep** en Linux, de la misma forma trataremos sus variantes: **egrep** y **fgrep**.



El comando Grep con ejemplos

Buscar en un archivo especifico:

Buscamos la línea que contiene '**DB_USER**' en el archivo de configuración "**wp-config.php**" de **WordPress**.

```
grep 'DB_USER' wp-config.php
```

Ejemplo de salida...

define('DB_USER', 'usuariobasededatos');

Buscar en todos los archivos:

Buscamos en el directorio "**wp-admin**" y solo en los archivos "**php**", las líneas que incluyan el texto '**str_replace**'.

```
grep "str_replace" admin*.php
```

Ejemplo de salida...

admin-ajax.php: add_action('wp_ajax_' . $_GET['action'], 'wp_ajax_' . str_replace('-', '_', $_GET['action']), 1);
admin-ajax.php: add_action('wp_ajax_' . $_POST['action'], 'wp_ajax_' . str_replace('-', '_', $_POST['action']), 1);
admin-header.php:$admin_body_class .= ' branch-' . str_replace(array('.', ','), '-', floatval($wp_version));
admin-header.php:$admin_body_class .= ' version-' . str_replace('.', '-', preg_replace('/^([.0-9]+).*/', '$1', $wp_version));
admin-header.php:$admin_body_class .= ' locale-' . sanitize_html_class(strtolower(str_replace('_', '-', get_locale())));

Buscar en todos los archivos identificando la linea:

Buscamos en el directorio "**wp-admin**" y solo en los archivos "**php**", las lineas que incluyan el texto '**str_replace**'. Este comando es similar al anterior, con la particularidad que nos indica el numero de linea donde se encuentra la expresión.

```
grep -n "str_replace" admin*.php
```

Ejemplo de salida...

admin-ajax.php:73: add_action('wp_ajax_' . $_GET['action'], 'wp_ajax_' . str_replace('-', '_', $_GET['action']), 1);
admin-ajax.php:76: add_action('wp_ajax_' . $_POST['action'], 'wp_ajax_' . str_replace('-', '_', $_POST['action']), 1);
admin-header.php:157:$admin_body_class .= ' branch-' . str_replace(array('.', ','), '-', floatval($wp_version));
admin-header.php:158:$admin_body_class .= ' version-' . str_replace('.', '-', preg_replace('/^([.0-9]+).*/', '$1', $wp_version));
admin-header.php:160:$admin_body_class .= ' locale-' . sanitize_html_class(strtolower(str_replace('_', '-', get_locale())));

Buscar archivos recursivamente que contengan un texto especifico:
Este comando es similar a los anteriores, pero solo nos imprimirá en pantalla el nombre de los archivos que contienen '**str_replace**'.

```
grep -ril "str_replace" admin*.php
```

Ejemplo de salida…
admin-ajax.php
admin-header.php

El comando grep permite la combinación con otros comandos, vemos algunos ejemplos.

Buscar archivos con un texto definido en la carpeta actual:
Hacemos una búsqueda de los archivos "**php**" en la carpeta donde nos encontremos en ese momento, que contengan el texto '**eval**'. Combinamos "**grep**" con "**find**".
Ejemplo de salida…
./wp-admin/includes/image.php
./wp-admin/includes/class-wp-upgrader.php
./wp-admin/includes/class-pclzip.php
./wp-admin/includes/media.php
./wp-admin/includes/update-core.php
./wp-admin/includes/ajax-actions.php
./wp-admin/includes/class-wp-automatic-updater.php
./wp-admin/includes/class-wp-posts-list-table.php
./wp-admin/user-edit.php
./wp-admin/edit-tag-form.php

Listar procesos de Apache:
Combinando "**grep**" con "**ps aux**", listamos los procesos de **Apache**.

```
ps aux | grep http
```

Ejemplo de salida…

```
root     19357 0.0 0.1 433928 22380 ? Ss mar05 0:01 /usr/sbin/httpd -DFOREGROUND
apache   25539 0.0 0.0 276728 9140 ? S 03:37 0:00 /usr/sbin/httpd -DFOREGROUND
apache   25541 0.0 0.0 436012 11092 ? S 03:37 0:00 /usr/sbin/httpd -DFOREGROUND
apache   25542 0.2 0.4 2361240 69560 ? Sl 03:37 0:29 /usr/sbin/httpd -DFOREGROUND
apache   25543 0.0 0.4 2361240 70536 ? Sl 03:37 0:09 /usr/sbin/httpd -DFOREGROUND
apache   25546 0.0 0.4 2361240 67648 ? Sl 03:37 0:09 /usr/sbin/httpd -DFOREGROUND
apache   27003 0.0 0.4 2361240 72088 ? Sl 03:40 0:11 /usr/sbin/httpd -DFOREGROUND
root     56885 0.0 0.0 112732 980 pts/0 S+ 06:57 0:00 grep –color=auto http
```

Ver el socket del ID de un proceso:
En este caso probaremos el "id 25546".

```
lsof -p 25546 | grep -Ei 'cwd|unix|sock'
```

Ejemplo de salida…

```
httpd 25546 apache cwd DIR 8,2 4096 2 /
httpd 25546 apache mem REG 8,2 91528 23729450 /usr/lib64/php-zts/modules/sockets.so
httpd 25546 apache mem REG 8,2 15392 23205555 /usr/lib64/httpd/modules/mod_unixd.so
httpd 25546 apache 1u unix 0xffff8bb764960000 0t0 5922520 socket
httpd 25546 apache 3u sock 0,7 0t0 5923227 protocol: TCP
httpd 25546 apache 5u sock 0,7 0t0 5923235 protocol: TCP
```

Ver el numero de conexiones a un puerto:
Combinando "**grep**" con el comando "**netstat**" podemos imprimir en pantalla el numero de conexiones de un puerto (en el ejemplo el 80).

```
netstat -an | grep :80 |wc -l
```

Ejemplo de salida…
1827

Eliminar mensajes congelados de Exim:
Eliminamos todos los mensajes congelados de la cola de correo de **Exim**.

```
exim -bpr | grep frozen | awk {'print $3'} | xargs exim -Mrm
```

Ejemplo de salida…
Message 1dChLr-0000y6-D1 has been removed
Message 1dPinU-0000Ld-S3 has been removed
Message 1dPaox-0000OC-A8 has been removed

Buscar archivos con un patrón de texto:
Buscamos archivos que contengan el patrón "**sololinux**".

```
find . -iname "*.txt" -exec grep -l "sololinux" {} +
```

Con el comando "**egrep**" podemos buscar varios patrones a la vez (en el ejemplo buscamos sololinux y adminserver).

```
egrep 'sololinux|adminserver' /etc/yum.conf
```

Ejemplo…
sololinux=/var/cache/yum/$basearch/$releasever
*adminserver=lm_sensors**

Por ultimo usaremos el comando "**fgrep**", que buscara un archivo o lista de archivos de una cadena de patrón fija.

```
fgrep 'sololinuxes' /etc/yum.conf
```

Ejemplo…

```
sololinuxes=/var/cache/yum/$basearch/$releasever
```

Damos por concluido el articulo "**Uso del comando Grep con ejemplos**", compártelo.

Instalar GitLab Runner en linux

Instalar GitLab Runner en **linux**.
GitLab Runner es una herramienta open source que te ayudara a ejecutar tus trabajos y enviar el resultado final a **GitLab**.
Su uso es conjunto con **GitLab CI**, que es el "**servicio de integración continua open source**" incluido en **GitLab** para coordinar los trabajos.
Sus principales características:

- Permite ejecutar:
  - Varios trabajos al mismo tiempo.
  - Uso de múltiples tokens en varios servidores.
  - Limitar el número de trabajos simultáneos por token.
- Los trabajos se pueden ejecutar:
  - En la zona.
  - Utilizando contenedores Docker.
  - Usando contenedores Docker y ejecutando trabajos sobre SSH.
  - Uso de contenedores Docker con autoescalado en diferentes nubes e hipervisores de virtualización.
  - Conexión al servidor SSH remoto.
- Está escrito en **Go** y se distribuye como binario único.
- Es compatible con **Bash**, **Windows Batch y Windows PowerShell**.
- Compatible con sistemas GNU / Linux, macOS y Windows.
- Permite personalizar el entorno de ejecución de trabajos.
- Configuración automática de recarga sin tener que reiniciar.
- La configuración es sencilla y admite múltiples entornos de ejecución (Docker, Docker-SSH, Parallels o SSH).
- Es posible almacenar en la caché de los **contenedores Docker**.
- Servidor HTTP con **monitorización Prometheus** incluida.

Las opciones más comunes de instalación son dos:

- Instalar desde binarios.
- Instalar desde repositorios oficiales.

En este articulo usaremos la opción de instalación desde binarios, considero que es la más efectiva (paquetes deb y rpm al final del articulo).

Vemos como **instalar GitLab Runner** en nuestra **distribución linux**.

Instalar GitLab Runner desde binarios

Dependiendo de la arquitectura de tu sistema, descargamos el archivo que corresponda con "**wget**".

64 Bits:

```
sudo wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64
```

32 Bits:

```
sudo wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-386
```

ARM:

```
sudo wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-arm
```

Establecemos los **permisos** requeridos.

```
sudo chmod +x /usr/local/bin/gitlab-runner
```

Opcional: Si piensas usar **"Docker"**, lo instalas ahora.

```
curl -sSL https://get.docker.com/ | sh
```

Creamos el usuario en **GitLab CI**.

```
sudo useradd --comment 'GitLab Runner' --create-home gitlab-runner --shell /bin/bash
```

Procedemos a instalar **GitLab Runner**.

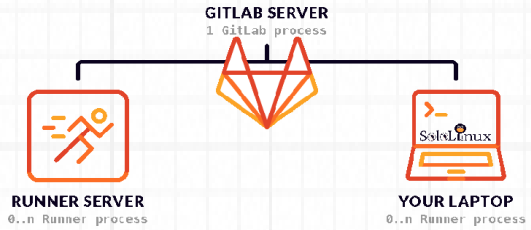
```
sudo gitlab-runner install --user=gitlab-runner --working-directory=/home/gitlab-runner
```

Lo iniciamos.

```
sudo gitlab-runner start
```

Aun no hemos concluido, debemos registrarnos en el servicio.

Podemos registrarnos en el servicio usando una línea de comandos similar a está (con tus datos):

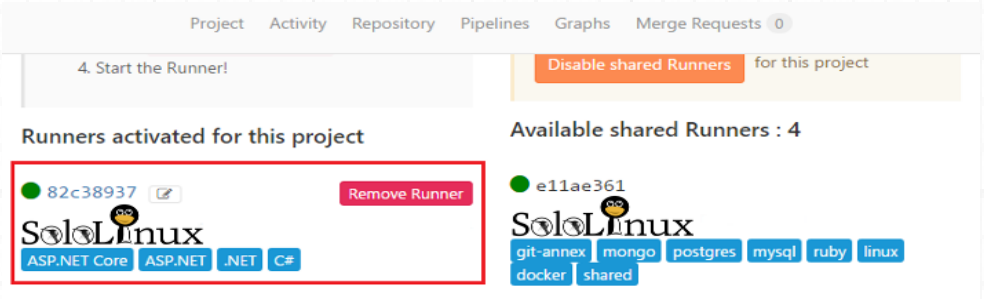



```
gitlab-runner register --url=http://gitlab.local/ --registration-token=xyLWaXwd15xy6VxU7HUV --non-interactive=true --locked=false --name=gitlab-runner-sololinux --executor=docker --docker-image=docker:stable --docker-volumes=/var/run/docker.sock:/var/run/docker.sock
```

Explicamos los datos de la linea de comandos:

- **gitlab-runner** – Herramienta.
- **register** – Opción para registrar un gitlab-runner.
- **–url=** – http: //gitlab.local (Url de la instancia GitLab).
- **–registration-token=** – xyLWaXwd15xy6VxU7HUV (Para registrar el "runner" inserta el Token. Lo tienes en el área de administración de tu GitLab.
- **–non-interactive=true** – No mostrar la salida.
- **–locked=false** – No bloquear el "runner".
- **–name=gitlab-runner-sololinux** – El nombre con el que se registrará el "gitlab-runner".
- **–executor=docker** – Debes seleccionar un ejecutor (ssh, docker+machine, docker-ssh+machine, kubernetes, docker, parallels, virtualbox, docker-ssh).
- **–docker-image=docker:stable** – Selecciona una imagen de docker (en caso de que uses docker).
- **–docker-volume=/var/run/docker.sock:/var/run/docker.sock** – Enlace del socket docker (en caso de que uses docker).

Si los datos introducidos son validos, tu servicio ya está operativo.



Actualizar GitLab Runner
Detenemos el servicio.

```
sudo gitlab-runner stop
```

Descargamos el archivo que corresponda con nuestra arquitectura.
64 Bits:

```
sudo wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64
```

32 Bits:

```
sudo wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-386
```

ARM:

```
sudo wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-arm
```

Establecemos los **permisos** requeridos.

```
sudo chmod +x /usr/local/bin/gitlab-runner
```

Iniciamos de nuevo el servicio.

```
sudo gitlab-runner start
```

Ya lo tenemos actualizado e iniciado.

Pd: Si no quieres descargar los binarios, puedes realizar la instalación desde paquetes “**deb**” o “**rpm**” que puedes encontrar en la **“zona de descargas oficial”**.

Comparte el articulo “**Instalar GitLab Runner en linux**“.

Instalar un servidor LEMP en Ubuntu 18.04

Instalar un **servidor LEMP en Ubuntu 18.04**.
LEMP es una plataforma de desarrollo muy útil para alojar sitios web estáticos o dinámicos.
A diferencia del paquete **LAMP** (usa apache), **LEMP** viene con el servidor "**Nginx**". El acrónimo usado para describir el paquete de herramientas, es el siguiente:

- **L** – Sistema operativo Linux
- **E** – Servidor web Nginx
- **M** – Base de datos MySQL, MariaDB, etc…
- **P** – Lenguajes de programación: PHP, Perl y Python

La **pila LEMP** es muy popular en ciertos entornos de trabajo, nos ofrece un rendimiento muy superior a **LAMP**.
Debes tener presente un detalle importante, las reglas de "**.htaccess**" no son validas en **Nginx** (existen conversores).
En este articulo veremos como instalar un **servidor LEMP en Ubuntu 18.04**.

Instalar un servidor Lemp en Ubuntu 18.04

Iniciamos sesión en consola / terminal y actualizamos el sistema.

```
sudo apt-get update && apt upgrade
```

Una vez actualizado el sistema, instalamos **Nginx**.

```
sudo apt-get install nginx
```

Iniciamos Nginx.

```
systemctl start nginx
```

Lo habilitamos para que inicie con el sistema.

```
systemctl enable nginx
```

Verificamos que Nginx esta activo y corriendo.

```
systemctl status nginx
```

Ejemplo de salida correcta…

Desde tu **navegador web** favorito, escribe la ip de tu servidor o vps en el campo de la url. La salida correcta será similar a la siguiente imagen de ejemplo.

Ahora instalamos **MySQL**.

```
apt install mysql-server
```

Una vez instalado, iniciamos MySQL.

```
systemctl start mysql
```

Lo habilitamos para que inicie con el sistema.

```
systemctl enable mysql
```

Aseguramos MySQL (responde a todo que si (Y)).

```
mysql_secure_installation
```

Ahora instalaremos php (Ubuntu 18.04 viene con php 7.2).
Antes de instalar el php es necesario que sepas que Nginx no procesa PHP de forma nativa. Por ese motivo debemos instalar **PHP-FPM** (FastCGI Process Manager), que es una implementación alternativa de **PHP FastCGI** que cuenta con unas características adicionales que ayudan a manejar sitios con una alta carga.
Instalamos php-fpm.

```
apt-get install php-fpm php-mysql
```

Verificamos la instalación.

```
php -v
```

Ejemplo de salida…

PHP 7.2.16-0ubuntu0.18.04.1 (cli) (built: Mar 12 2019 09:56:21) (NTS)
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
with Zend OPcache v7.2.15-0ubuntu0.18.04.1, Copyright (c) 1999-2018, by Zend Technologies

Como ultimo paso, solo nos falta completar el archivo de configuración de "**Nginx**" con nuestro **php** y sitio web.

```
cd /etc/nginx/sites-available/
nano /etc/nginx/sites-available/your_domain.com.conf
```

Copia y pega lo siguiente:

```
server {
listen 80;
root /var/www/html;
index index.php index.html index.htm index.nginx-debian.html;
server_name your_domain.com;

location / {
try_files $uri $uri/ =404;
}

location ~ \.php$ {
include snippets/fastcgi-php.conf;
fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
}

location ~ /\.ht {
deny all;
}
}
```

Guarda el archivo y cierra el editor.
Habilitamos la configuración.

```
ln -s /etc/nginx/sites-available/your_domain.com.conf /etc/nginx/sites-enabled/your_domain.com.conf
```

Para asegurarnos de que no existe ningún error de sintaxis en el archivo de configuración.

Ejecuta lo siguiente:

```
nginx -t
```

Ejemplo de salida que indica que todo es correcto…

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

```
service nginx reload
systemctl restart nginx
```

Ya tienes un **servidor LEMP en Ubuntu**, no te olvides que la ruta para alojar tu sitio, es:

/var/www/html/

Si te fue útil este articulo, compártelo "**Instalar un servidor Lemp en Ubuntu 18.04**".

Comandos de “nano”

Comandos de nano, el editor por excelencia en consola.
En un articulo anterior ya tratamos en profundidad al editor “**vi / vin**“, hoy hablamos del que nosotros siempre recomendamos “**nano**“.
El **editor nano** no es el mejor, tampoco destaca en ningún apartado, sencillamente es el más fácil de usar.

Por eso es altamente recomendado para los usuarios noveles.

Hoy veremos los comandos mas habituales en este fabuloso editor.

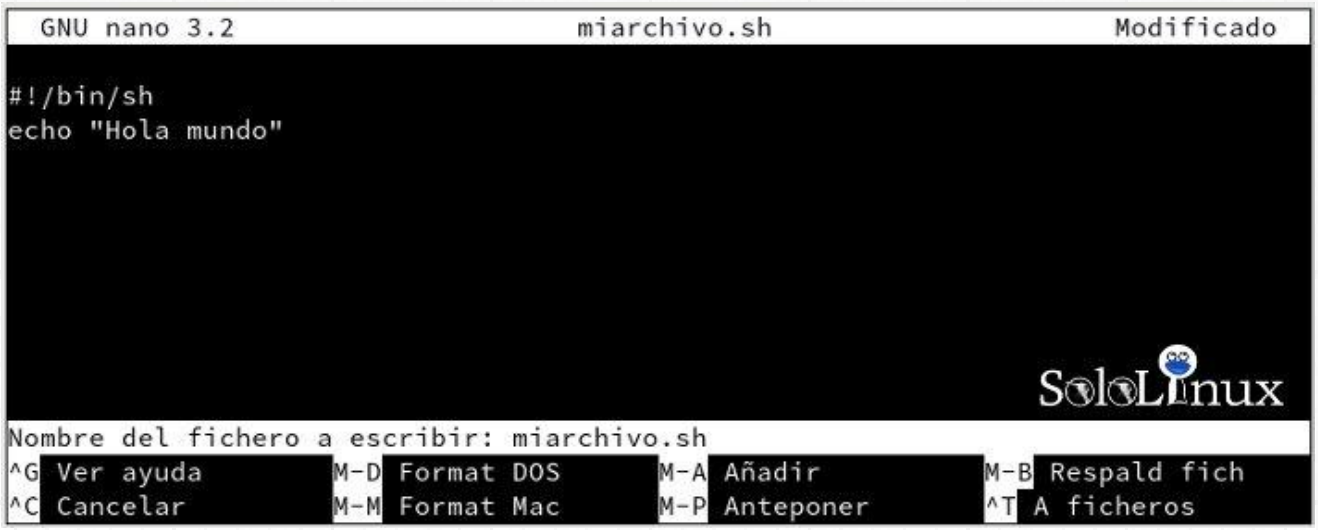
Comandos de nano
Crear un nuevo archivo en **nano** es tan fácil como escribir en la consola “nano” seguido del archivo a crear, por ejemplo:

```
nano miarchivo.sh
```

Ahora creamos cualquier script, en este caso un “**Hola mundo**“.

```
#!/bin/sh
echo "Hola mundo“
```

Para guardar el **script** pulsas la tecla “**Ctrl**” y la tecla “**o**“, el editor te preguntara si quieres guardar, pulsas **enter** y ya lo tienes guardado.
Para salir de **nano**, pulsas la tecla “**Ctrl**” y la tecla “**w**“.



Los comandos, o mejor dicho, los atajos del **editor nano** más habituales son los siguientes:

- **Ctrl + g** ———- Abrir la ayuda de nano.
- **Ctrl + x** ———- Salir de nano.
- **Ctrl + o** ———- Guardar el archivo actual.
- **Ctrl + r** ———- Insertar otro fichero en el actual.
- **Ctrl + w** ———- Buscar un texto en el archivo que tienes abierto.
- **Ctrl + y** ———- Volver a la página anterior.
- **Ctrl + v** ———- Saltar a la página siguiente.
- **Ctrl + k** ———- Cortar la línea o región seleccionada y guardarla en el cutbuffer.
- **Ctrl + u** ———- Pegar lo guardado en el cutbuffer en la línea actual.
- **Ctrl + l** ———- Recargar la pantalla actual.
- **Ctrl + j** ———- Justificar el párrafo actual.
- **Ctrl + m** ———- Insertar un retorno del carro en la posición del cursor.
- **Ctrl + _** ———- Saltar a un número de línea en concreto.
- **M + g** ———- Ir a un número de línea en concreto.
- **M + i** ———- Auto indentar ademas de habilitar y deshabilitar.
- **M + x** ———- Habilitar/deshabilitar el modo ayuda.
- **M + p** ———- Habilitar/deshabilitar el modo pico.
- **M + m** ———- Habilitar/deshabilitar el soporte del mouse.
- **M + r** ———- Reemplazar texto.
- **M + e** ———- Habilitar/deshabilitar las expresiones regulares.
- **M + b** ———- Habilitar/deshabilitar el respaldo de ficheros.
- **M + s** ———- Habilitar/deshabilitar el desplazamiento suave.
- **M + h** ———- Habilitar/deshabilitar la tecla ‘smart home’.
- **M + y** ———- Habilitar/deshabilitar el coloreado de la sintaxis.
- **M + p** ———- Habilitar/deshabilitar mostrar blancos.

Para ampliar información puedes revisar la **documentación oficial**.

Si te gusto el articulo, compartelo. “**Comandos de “NANO”**”.

Aumentar el valor en max_allowed_packet o wait_timeout

Aumentar el valor en max_allowed_packet o wait_timeout.

Cuando un servidor o un cliente "**MySQL /MariaDB**" recibe un paquete con un tamaño superior a lo que está fijado en **"max_allowed_packet"** (por defecto 16M), automáticamente lanza el "**error Packet too large**" o similar, y cierra la conexión.

Debes saber que cada servidor y cliente tienen su propia variable "**max_allowed_packet**", por tanto si queremos trabajar con paquetes grandes, debemos aumentar el tamaño definido en la variable tanto en el servidor como en el cliente.

Existen multitud de aplicaciones que te exigen que aumentes el tamaño de la variable, o por lo menos te lo recomiendan.

Un ejemplo claro es con "Matomo".

Si tienes un servidor de **analíticas Matomo** y dado los grandes archivos de datos que maneja, es normal que te solicite que aumentes la variable "**max_allowed_packet**".

Tamaño máximo del paquete

Es recomendable configurar un tamaño de al menos 64MBen 'max_allowed_packet' en su base de datos MySQL. Actualmente está configurada en 16MB.

Aumentar el valor en max_allowed_packet

Dependiendo de tu sistema, puedes localizar el archivo de configuración en alguno de los siguientes archivos:

- **/etc/my.cnf**
- **/etc/mysql/my.cnf**
- **lampp/etc/my.cnf**

Nota: wait_timeout= 31536000 es el valor máximo admitido en un servidor MySQL.

Ejemplo de uso:

nano /etc/my.cnf

Modificas las siguientes lineas según tus necesidades (si no están, las puedes crear tu mismo):

```
[mysqld]
wait_timeout = 31536000
max_allowed_packet=80M
....
[mysqldump]
max_allowed_packet=80M
```

Una vez guardes el archivo y cierres el editor, solo falta reiniciar.

service mysqld restart

Uso del comando last

Uso del **comando last** en **Linux**.

El comando "**last**" es una sencilla pero útil herramienta que nos lista los accesos al sistema, las salidas, si reiniciaron la maquina, accesos a un archivo en particular, etc…, y todo ello indicando fecha y hora.

Si administras un sistema es indispensable que lo conozcas en profundidad.

Toma la información del archivo /var/log/wtmp

En este articulo veremos algunos ejemplos de uso, así como las opciones y reglas del comando.

Uso del comando last

La sintaxis del comando es la siguiente:

last [options] [username...] [tty...]

Las opciones:

-f archivo	Indica cuando se uso por ultima vez un archivo especifico.
–(numero de lineas)	Puedes indicar el máximo numero de lineas a visualizar (en numérico).
-n (numero de lineas)	Lo mismo que la opción anterior pero numerándolas.
-t (yyyymmddhhmmss)	Muestra los inicios de sesión desde una fecha especifica, incluyendo la hora, los minutos y segundos (útil para localizar intromisiones ajenas).
-R	Con está opción no se visualiza el hostname en el listado.
-a	Muestra el host al final de cada linea, es muy útil combinada con -d.
-d	Traduce ip's a host si el acceso es remoto, si es local hace lo contrario.
-F	Lista los inicios y cierres de sesión especificando la fecha y horario al completo.
-i	Similar a la opción -d.
-o	Lee los archivos de wtmp que escribieron aplicaciones linux-libc5.
-w	Muestra el usuario al completo, incluyendo sus salidas.
-x	Muestra los apagados del sistema y cambios de estado.

Ejemplos de uso del comando last

Uso del comando “last” sin opciones. last	*sergio console :0 Thu Mar 21 10:23 still logged in* *reboot system boot 4.4.175-89-defau Thu Mar 21 10:22 – 12:46 (02:23)* *sergio console :0 Thu Mar 21 09:42 – 10:07 (00:25)* *reboot system boot 4.4.175-89-defau Thu Mar 21 09:41 – 10:07 (00:25)* *sergio console :0 Wed Mar 20 22:24 – 08:40 (10:16)* *reboot system boot 4.4.175-89-defau Wed Mar 20 22:24 – 08:40 (10:16)*
Imprime cuando se reinicio el sistema. last reboot | less	*reboot system boot 4.4.175-89-defau Thu Mar 21 10:22 – 12:48 (02:25)* *reboot system boot 4.4.175-89-defau Thu Mar 21 09:41 – 10:07 (00:25)* *reboot system boot 4.4.175-89-defau Wed Mar 20 22:24 – 08:40 (10:16)* *reboot system boot 4.4.175-89-defau Wed Mar 20 14:09 – 08:40 (18:31)* *reboot system boot 4.4.175-89-defau Wed Mar 20 10:13 – 13:08 (02:54)* *reboot system boot 4.4.175-89-defau Wed Mar 20 09:43 – 10:13 (00:29)*
Ahora también imprime los apagados. last -x | less	*sergio console :0 Wed Mar 20 09:44 – 10:13 (00:28)* *runlevel (to lvl 5) 4.4.175-89-defau Wed Mar 20 09:44 – 10:13 (00:28)* *reboot system boot 4.4.175-89-defau Wed Mar 20 09:43 – 10:13 (00:29)* *shutdown system down 4.4.175-89-defau Wed Mar 20 05:13 – 09:43 (04:29)* *sergio console :0 Tue Mar 19 09:40 – 05:12 (19:31)* *runlevel (to lvl 5) 4.4.175-89-defau Tue Mar 19 09:40 – 05:13 (19:33)*
Imprime el host al final de la linea. last -a	*sergio console Thu Mar 21 10:23 still logged in :0* *reboot system boot Thu Mar 21 10:22 – 12:57 (02:35) 4.4.175-89-default* *sergio console Thu Mar 21 09:42 – 10:07 (00:25) :0* *reboot system boot Thu Mar 21 09:41 – 10:07 (00:25) 4.4.175-89-default* *sergio console Wed Mar 20 22:24 – 08:40 (10:16) :0* *reboot system boot Wed Mar 20 22:24 – 08:40 (10:16) 4.4.175-89-default*

Como puedes ver el uso de las opciones nos aportara mucha información que espero que te sea de utilidad. Compártelo.

Instalar y configurar CentOS Web Panel – CWP 1/3

Instalar y **configurar CentOS Web Panel** – **CWP** 1/3.

"CentOS Web Panel" más conocido como "**CWP**", es un popular **panel de control web** totalmente gratuito, y especialmente diseñado para que puedas administrar de una forma rápida y sencilla tus **servidores dedicados** o **VPS** .

En esta serie (que constara de tres partes), aprenderemos a **instalar y configurar CentOS Web Panel** de principio a fin. Todo el proceso lo realizaremos en un **Cloud VPS** cedido por **"Clouding.io"**, y debo decir que estoy realmente sorprendido del servicio que ofrecen, simplemente…

Las características principales, de **Clouding.io** hablan por si solas.

- **Son potentes**: Procesadores **Intel Xeon**, discos **SSD Ceph**, tres niveles de **cache en RAM**, ademas te ofrecen una conexión de 500 Mbps, con 2Tb de transferencia.
- **Son estables**: **Anti-DDos**, copias en triple réplica, calidad empresarial en un **datacenter Tier IV**.
- **Son Flexibles**: **Cloud Linux/Windows**, múltiples imágenes autoinstall, configuración a medida, amplia/reduce, apaga/enciende, pagas por horas de uso, y ademas puedes lanzar tu nuevo **Cloud VPS** en segundos.
- **Uso muy fácil**: El uso de esta plataforma es simple, no requiere ningún tipo de conocimiento extra (en este mismo articulo lo podrás comprobar). Ademas ofrecen soporte telefónico, por email, y lo más importante… en **castellano**.
- **Demo**: Ofrecen 5€ de saldo gratis para que compruebes tu mismo su rendimiento y lo sencillo que es usar el servicio.

El datacenter, al igual que las oficinas se encuentran en Barcelona (España), te recomiendo que lo pruebes.

Una vez explicadas las bondades del servicio donde haremos estás pruebas, y antes de comenzar con la primera de las tres partes del tutorial, quería comentar unos pequeños detalles sobre **CentOS Web Panel**.

- Tiene su propia **consola ssh**.
- Permite asignar diferentes recursos a los usuarios.
- **Firewall CSF** preinstalado.
- Requisitos de hardware mínimos.

clouding.io = **Impresionante**

Instalar y configurar CentOS Web Panel
Nos registramos en "**clouding.io**".

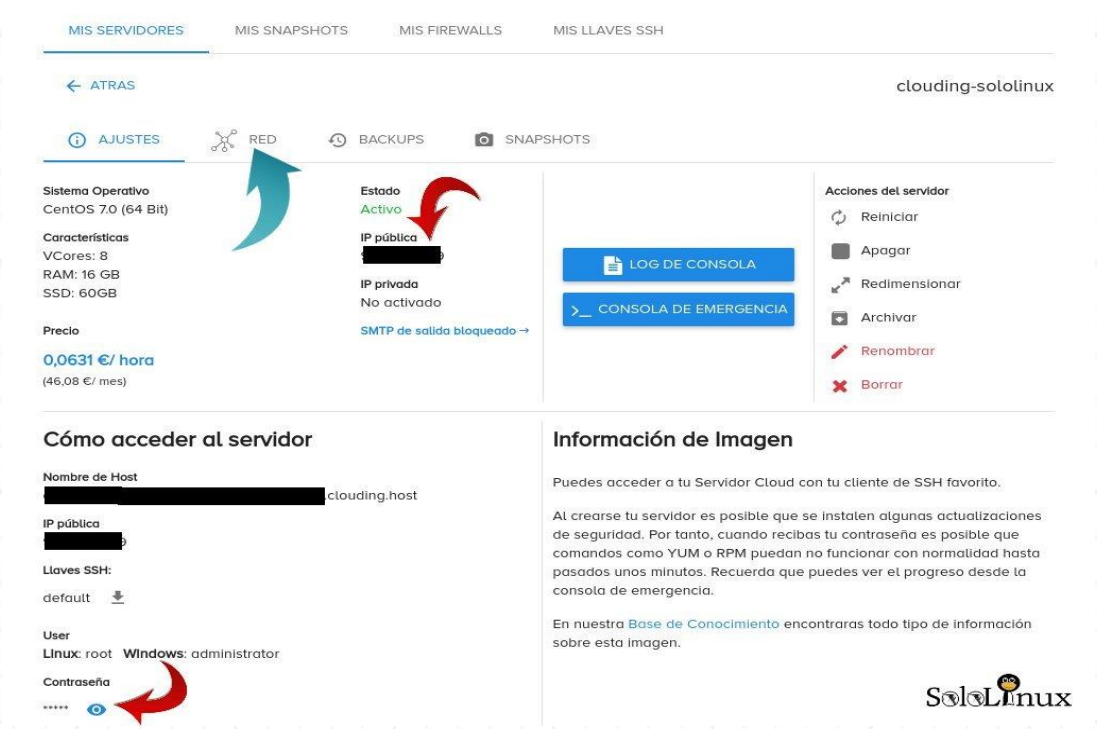
Accedemos al panel y creamos nuestro servidor.

Seleccionamos un nombre para el **Cloud VPS** (será el **hostname**), elige tu configuración y haz click en enviar (reproduce el vídeo de demo).

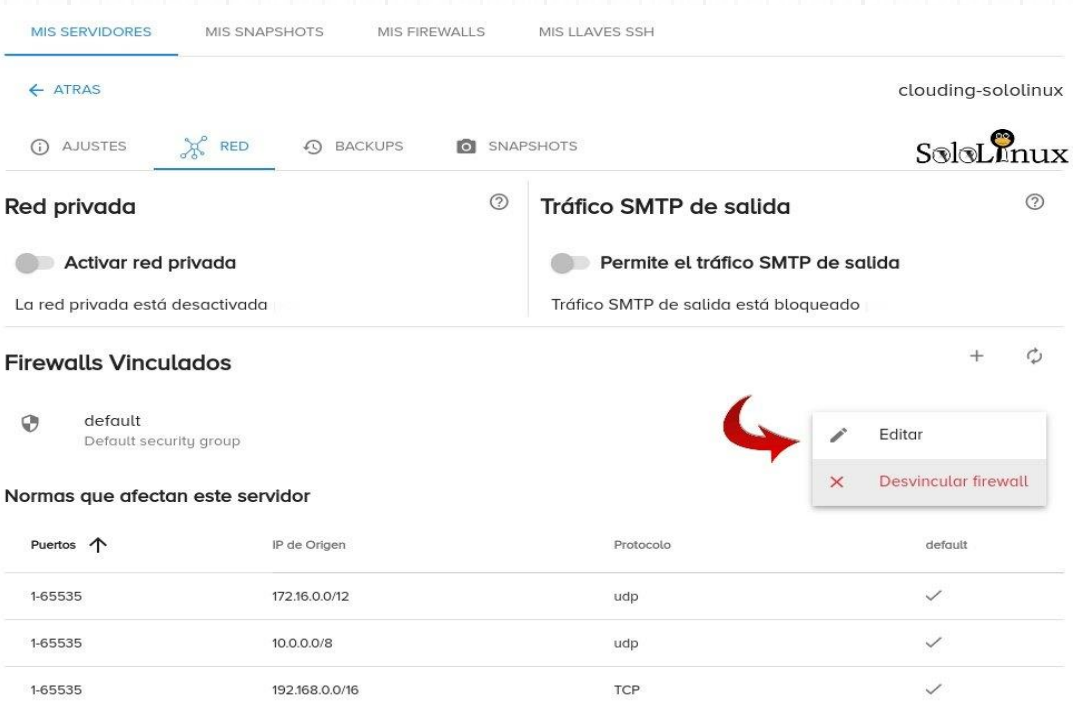
Seleccionamos un nombre para el **Cloud VPS** (será el **hostname**), elige tu configuración y haz click en enviar (reproduce el vídeo de demo).

En segundos (es muy rápido) el sistema ya esta listo, para continuar pulsa en el nombre que le dimos a nuestra maquina.

En la pantalla del panel veras los datos del servidor, donde te indican las flechas rojas podrás localizar los datos para acceder mediante la consola o terminal **ssh** al servidor. Pero antes debemos modificar el **firewall** que por defecto viene con nuestro servidor, pulsamos en la casilla "**RED**".



En "**Firewalls Vinculados**", a la derecha tienes la opción que te permite editar el **firewall**. Como estamos preparando nuestra maquina para **instalar y configurar CentOS Web Panel**, debemos abrir el **puerto 2030** para el usuario root del panel, y el 2082 para los usuarios normales, los dos puertos en **TCP**. En la opción editar tienes la opción de agregar reglas, las creas y ya tenemos listo el sistema.



Llego el momento de instalar "CWP" en nuestro CloudVPS de "Clouding.io".
Con los datos que nos aporto el panel que vimos anteriormente, accedemos a la maquina por ssh y comenzamos.
Actualizamos el sistema e instalamos **wget**.
yum update
yum install wget

Abrimos el directorio "**scr**".
cd /usr/local/src

Descargamos el **script bash** de instalación
wget http://centos-webpanel.com/cwp-el7-latest

Ejecutamos.
sh cwp-el7-latest

La instalación comienza de forma automática, puede tardar unos minutos en concluir.

Una vez instalado el panel de control en el **CloudVPS** de **Clouding.io**, aparecerá una pantalla similar a está.

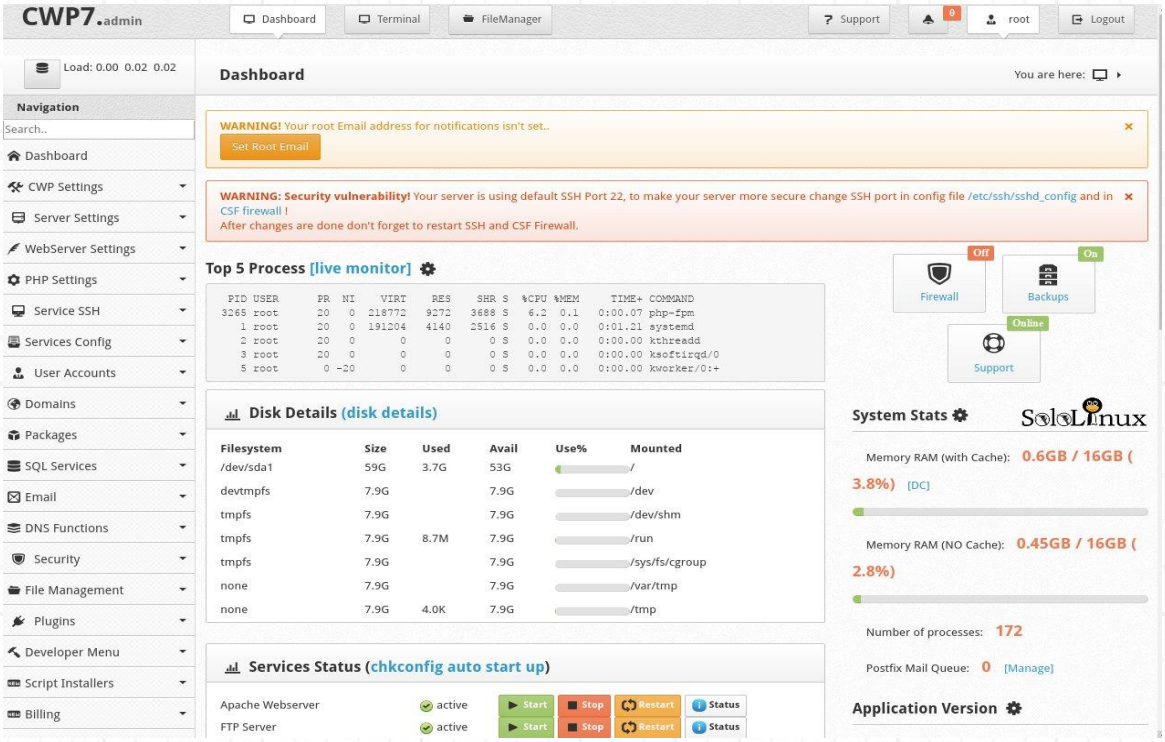
```
##############################
#       CWP Installed        #
##############################

go to CentOS WebPanel Admin GUI at http://SERVER_IP:2030/

http://[redacted]:2030
SSL: https://[redacted]:2031
--------------------
Username: root
Password: ssh server root password
MySQL root Password: YrhuTSkDBVgT

############################################################
          CentOS Web Panel MailServer Installer
############################################################
SSL Cert name (hostname): clouding-sololinux
SSL Cert file location /etc/pki/tls/ private|certs
############################################################

visit for help: www.centos-webpanel.com
Write down login details and press ENTER for server reboot!
Press ENTER for server reboot!
```

Como puedes observar en la anterior imagen, ya tienes la url de acceso y sus datos.

Accedemos al panel de control, pero recuerda que para acceder debes usar:

- **root**
- **password de root**

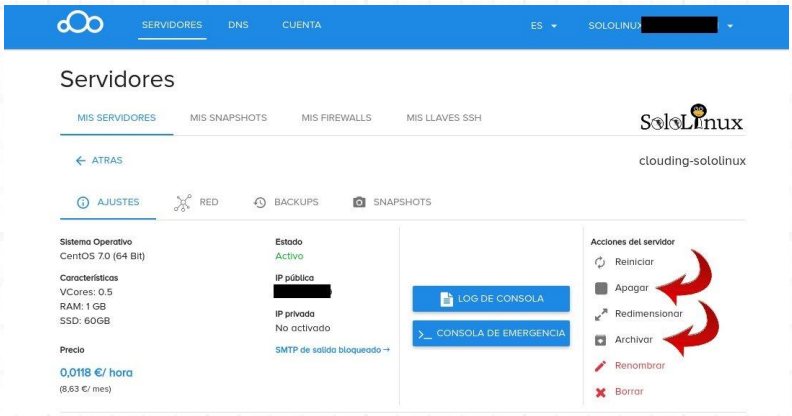
Se abre **CentOS Web Panel** por primera vez.



Como ya explique al comenzar este articulo, el tutorial es en tres partes.

Si consideras que este articulo puede servir de ayuda, compártelo.

Instalar y configurar CentOS Web Panel – CWP 2/3



Instalar y configurar CentOS Web Panel – CWP 2/3.

En este articulo continuamos con la serie "**Instalar y configurar CentOS Web Panel – CWP**" (parte1–parte2-parte3), pero antes… quería comentaros unos detalles sobre el **Cloud VPS** que amablemente nos a cedido "**Clouding.io**" para realizar nuestras pruebas.

Como ya sabes, en el articulo anterior instalamos **CentOS Web Panel**, y para ello creamos un **Cloud VPS** de 8 cores con 16 GB de ram.

Hoy quería comprobar el rendimiento de "CWP" limitando los recursos que nos ofrece "**Clouding.io**" al mínimo, y la verdad es que estoy realmente sorprendido, después de bajar a 0.5 cores y 1 GB de ram, apenas se aprecia la drástica reducción en la usabilidad del **panel CWP**.

Otro detalle interesante del Cloud VPS, es que se permite apagar o archivar el servidor (al archivar solo se paga por el SSD y el backup).

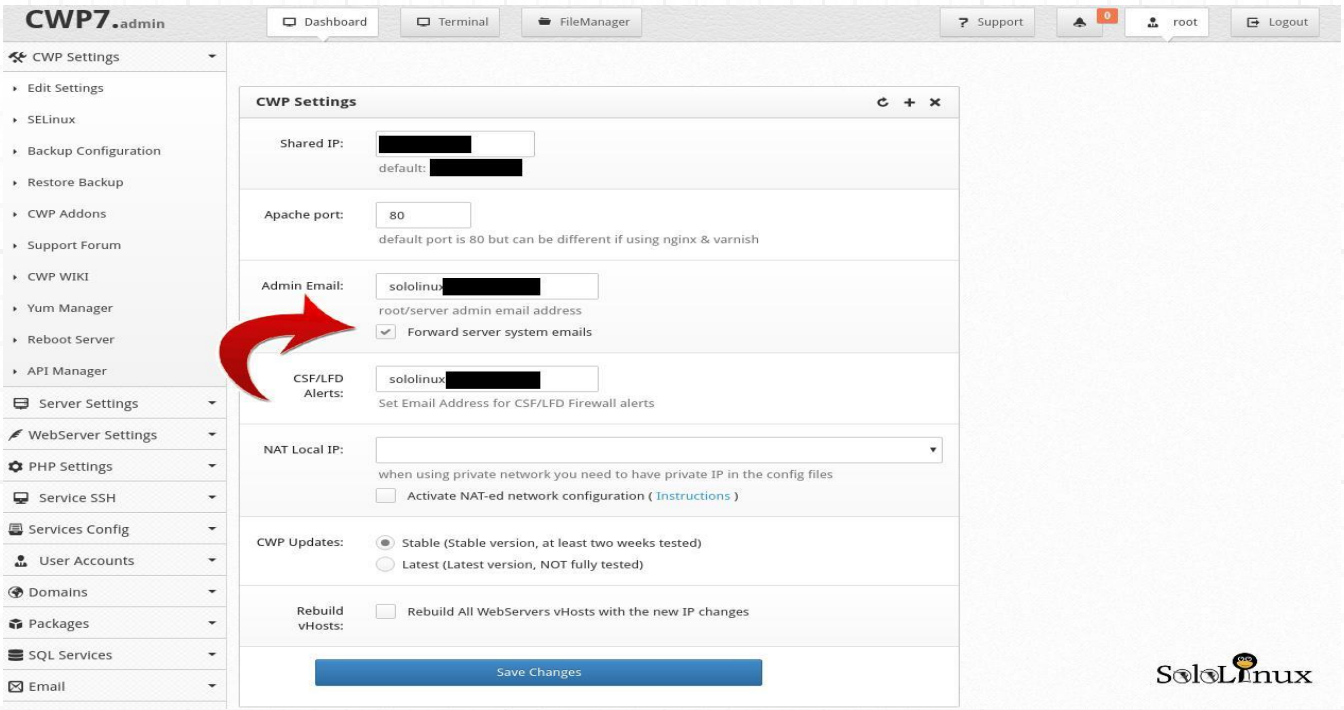
No debes olvidar que **Clouding.io** nos ofrece las distribuciones: CentOS, Debian, Ubuntu, más las imágenes de WordPress, Odoo, Prestashop y Ghost (también Windows server 2003, 2008, 2012, 2016, 2019 y Windows 10), pero nosotros instalamos CentOS 7 para poder utilizar CentOS Web Panel.

Instalar y configurar CentOS Web Panel

Tal como terminamos el articulo anterior, al iniciar por primera vez el panel de control web "**CWP**". Lo primero que nos encontramos es un panel bien estructurado, información necesaria a la vista, un menú visible, y dos advertencias de seguridad que vamos a corregir.

E-mail del root

En la primera advertencia de la imagen anterior pulsamos en "**Set root email**". Introducimos nuestros datos (te recomiendo que marques la opción señalada con una flecha) y guardamos los cambios.

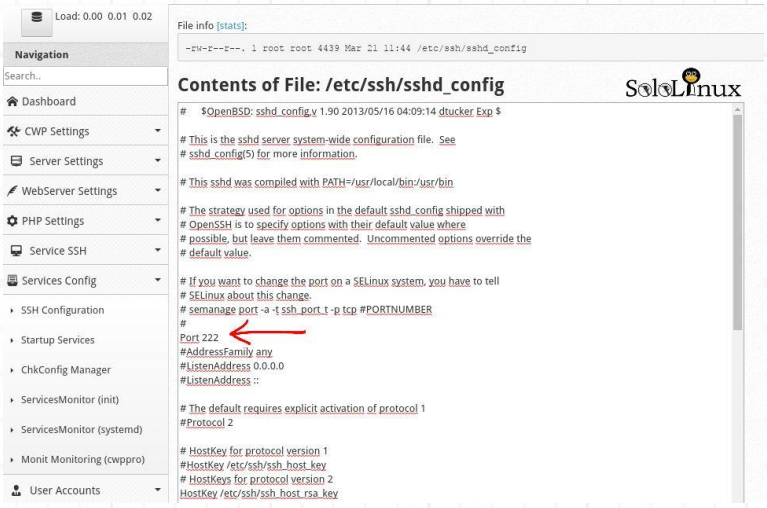


Configurar puerto SSH (opcional)

Por seguridad es altamente recomendable que modifiques el **puerto SSH**: Services Config / SSH Configuration. Descomentas la línea "Port" y cambias el puerto por defecto (22), en el ejemplo colocamos el "222". Recuerda que al modificar el puerto **SSH**, debes abrir el 222 en el firewall de **Clouding.io** (revisa el articulo anterior).
Si quieres acceder vía SSH deberás ejecutar:

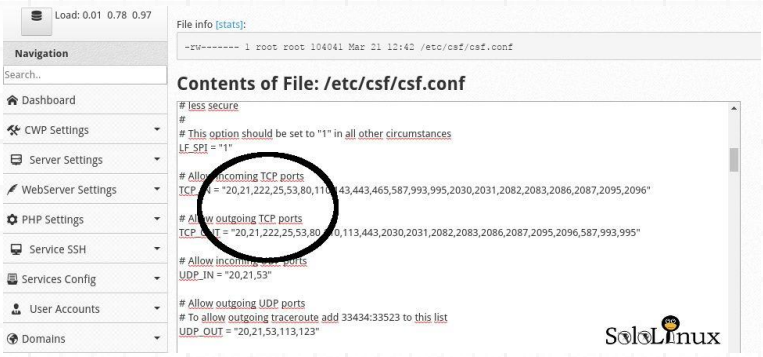
ssh -p 222 root@ip-vps-clouding

Es necesario agregar el puerto 222 al **Firewall**.
Abrimos “Security / Firewall Manager” y en “Configuration”, pulsamos en “Main configuration”.

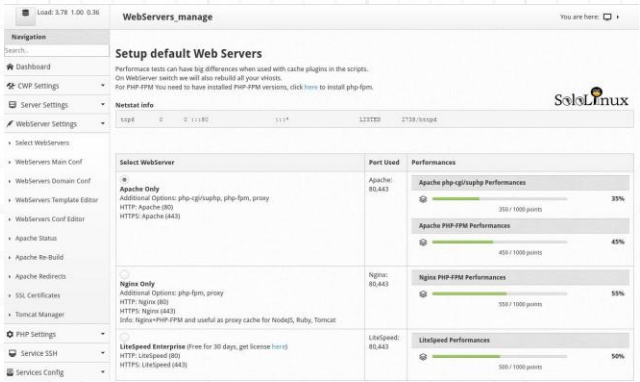


En las líneas 139, 142 y 2478, tienes el puerto 22 por defecto, en las tres líneas debes sustituir “22” por “222”.
Ejemplo…..

138 # Allow incoming TCP ports
139 TCP_IN =
“20,21,222,25,53,80,110,143,443,465,587,993,995,2030,2031,2082,2083,2086,2087,2095,2096”
140
141 # Allow outgoing TCP ports
142 TCP_OUT = “20,21,222,25,53,80,110,113,443,2030,2031,2082,2083,2086,2087,2095,2096,587,993,995”
…….
2477 # /etc/ssh/sshd_config
2478 PORTS_sshd = “222”



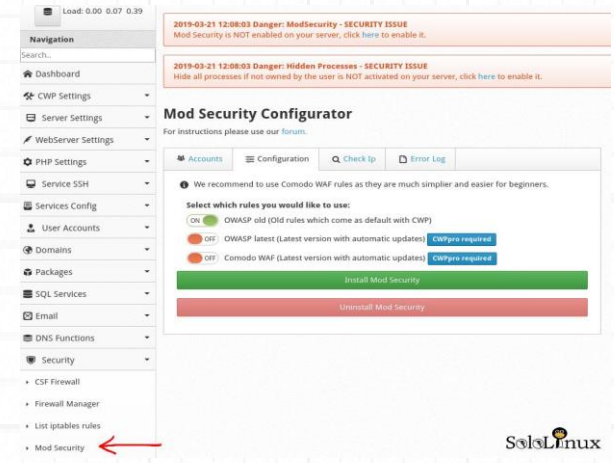
Selecciona el servidor web
Si quieres modificar el servidor web, ahora es el momento.



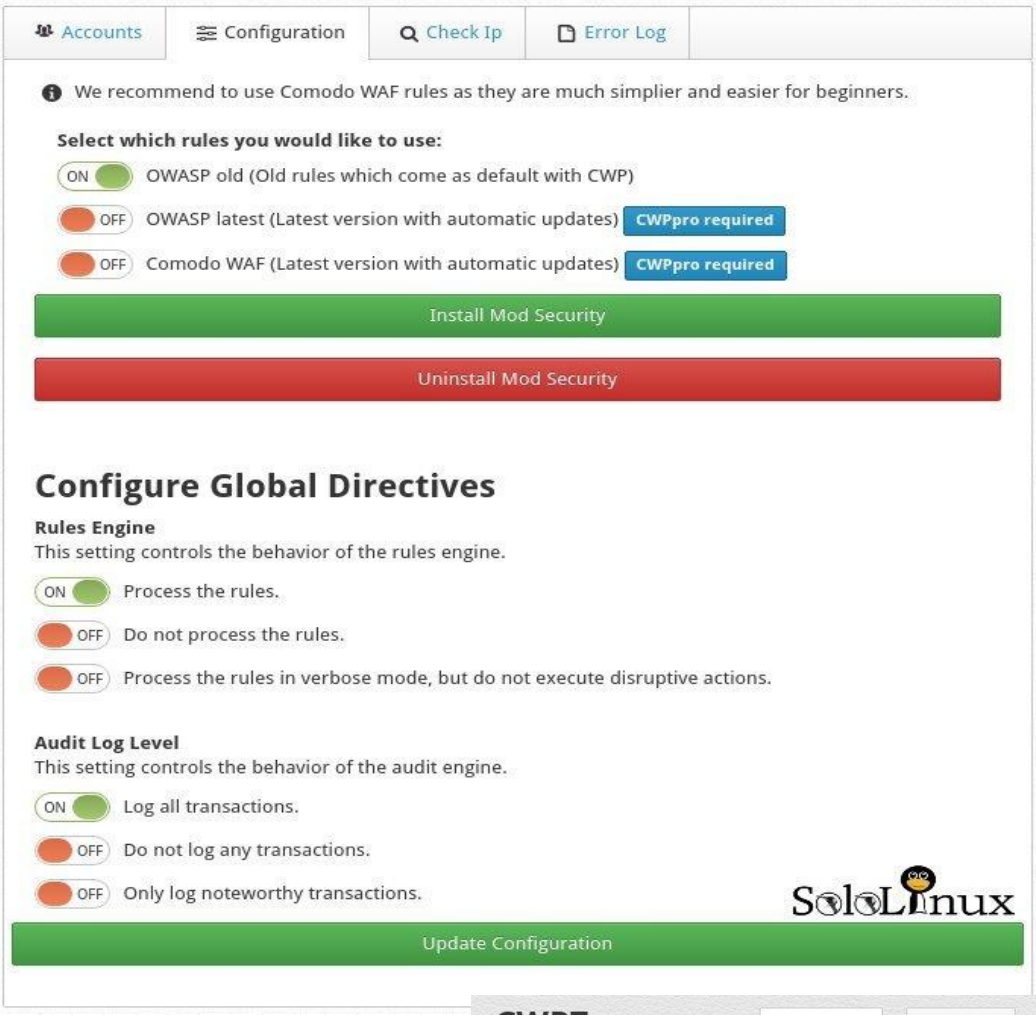
Activar el Firewall
Asegúrate de tener el Firewall activado.
Abrimos “Security / Firewall Manager” y pulsa en “**Enable Firewall**“.

Reiniciar servicios
Una vez activado el Firewall, volvemos al **Dashboard** (pantalla principal) y reiniciamos todos los servicios (pulsando en Restart uno por uno).

Configurar Mod Security
Ahora nos aparecen dos advertencias de seguridad.
La segunda advertencia es para que otros usuarios no puedan visualizar los procesos, pulsas en “here” y desactivas la opción.
La primera advertencia te avisa de que no tienes activado el **Mod_Security**, pulsas en “here”, o en Security / Mod Security. Continua en “Configuration” e “Install Mod Security”.



Configuramos las directivas de seguridad y actualizamos la configuración.



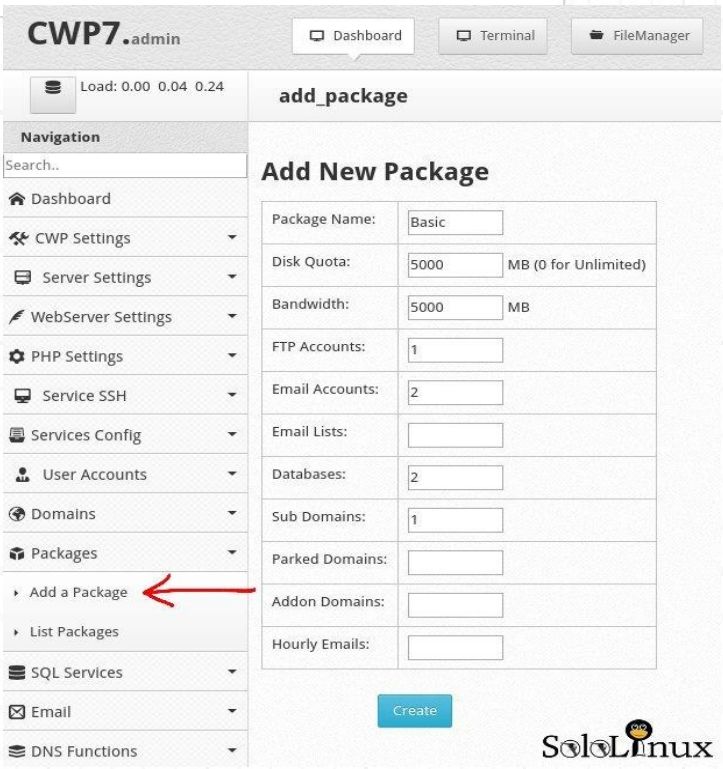
Crear paquetes de alojamiento

Si queremos alojar múltiples usuarios o clientes que tengan sus propios sitios web, lo mejor es que crees paquetes de alojamiento.

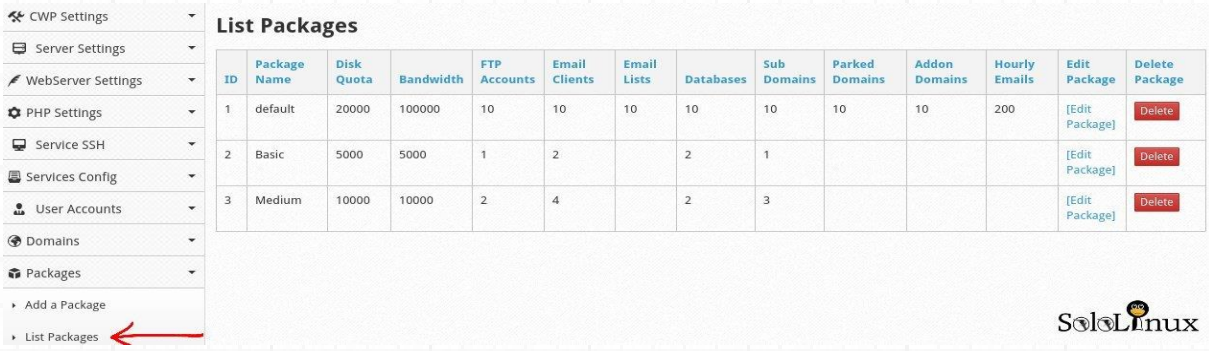
Los paquetes permiten al administrador controlar los recursos del servidor, ya que permite limitar el espacio en disco, el ancho de banda, numero de dominios, etc…

Para crear un nuevo paquete, desde la barra lateral, hacemos click en Packages / Add a Package.

Rellenas los campos que necesites y pulsas en “Create”.

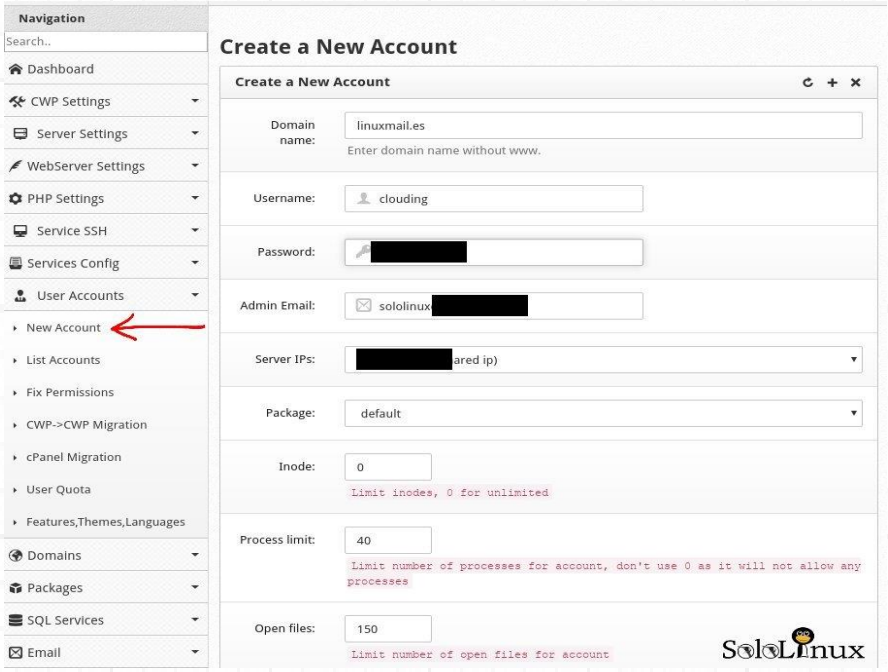


En "**List a Packages**", ademas de listar los paquetes de alojamiento creados puedes editar o borrar cualquiera de ellos.



Crear cuentas de usuario

Ya tienes creados los paquetes de alojamiento, nos falta nuestro primer usuario. En "User Account", pulsamos en "New Account" e insertamos los datos requeridos (permite limitar los **inodos**, procesos y el numero de archivos totales).



Los datos de la nueva cuenta de usuario.

Acceso al panel de usuario

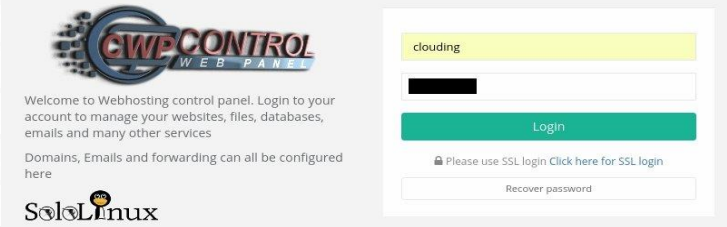
Anteriormente, abrimos en el panel de **Clouding.io** los puertos 2030 y 2082.

•Puerto 2030 – Acceso a CWP como root/admin.

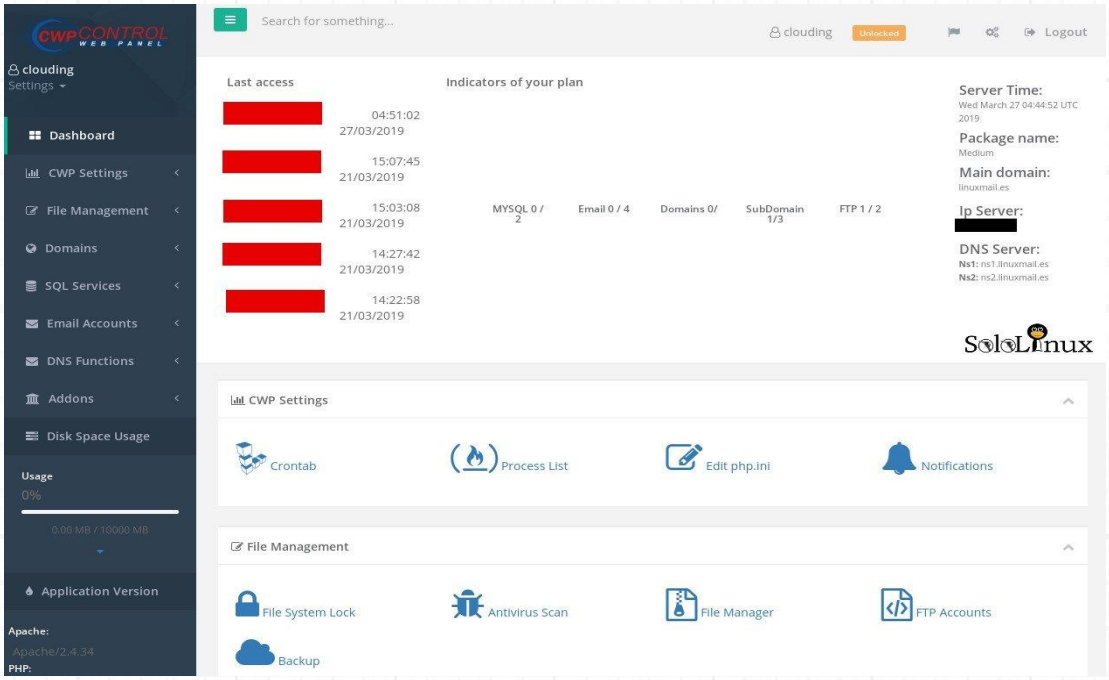
•Puerto 2082 – Acceso a CWP para usuarios sin privilegios.

Accedemos como usuario por primera vez:

http://ipdelservidor:2082



Una vez introducido el nombre de usuario y el password, aparece el “dashboard” del usuario/cliente.



Como puedes observar el panel del usuario tiene menos opciones que el panel del administrador, revísalo.

Comparte el articulo.

Instalar y configurar CentOS Web Panel – CWP 3/3

Instalar y configurar CentOS Web Panel – CWP 3/3 en Clouding.io.
Este es el ultimo articulo de la serie "**Instalar y configurar CentOS Web Panel – CWP**", recordemos que consta de tres partes.

1. Instalar-y-configurar-centos-web-panel-cwp-1-3 – Instalación.
2. Instalar-y-configurar-centos-web-panel-cwp-2-3 – Configuración.
3. Instalar-y-configurar-centos-web-panel-cwp-3-3 – Clientes.

En los anteriores artículos de la serie hablamos de "**Clouding.io**" (recordemos que nos ha cedido un **Cloud VPS** para realizar nuestros tutoriales), que por sus características, rendimiento y versatilidad nos parecen realmente buenos.

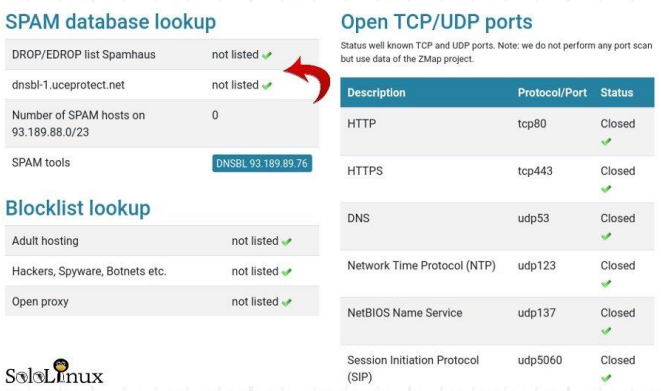
Pero nunca se habla de la calidad de las IP, nunca.

No lo puedo entender, tantos sitios web que realizan "**reviews**" de hostings, VPS, etc…, pero ninguno se preocupa en comprobar la calidad de las IP. Que la IP este limpia es fundamental para un profesional, incluso también si quieres comenzar un nuevo sitio web.

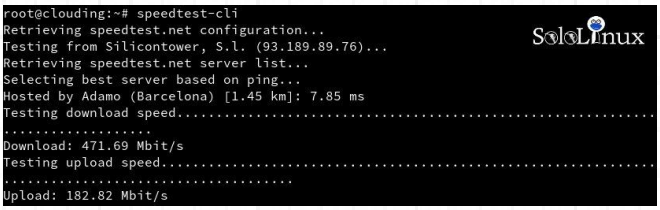
Te imaginas adquirir un VPS y que su ip este en una lista negra, es un desastre. Adiós SEO, adiós autoridad, adiós mails, adiós a todo. Tu IP fue una maquina de enviar spam, bots, etc…, así que olvídate porque limpiar una IP no siempre es fácil.
En la totalidad de pruebas realizadas en **Clouding.io** obtuvimos mas de 15 ip's diferentes, se comprobaron todas, y todas limpias.
Observa las imágenes siguientes:
Como dicen ellos, el "**Datacenter Tier IV**" se localiza en Barcelona.

Clouding.io es una "**Spin-off**" de **SiliconTower** (reconocido prestigio).

Como puedes ver en la siguiente imagen la **ip** está limpia, es más… está limpio el rango completo, y eso denota la calidad de la empresa. Ofrecen un producto de calidad al completo.

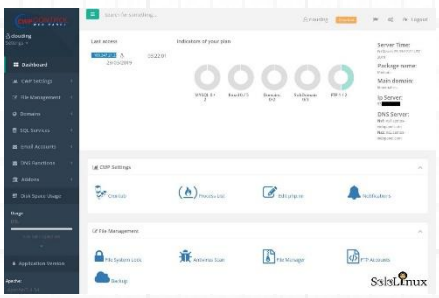


Dejando a un lado el tema de las IP, también comprobamos el ancho de banda. Casi llega a los 500Mbps a la hora de servir datos, la verdad es que no está nada mal (lo que se oferta normalmente son 100).

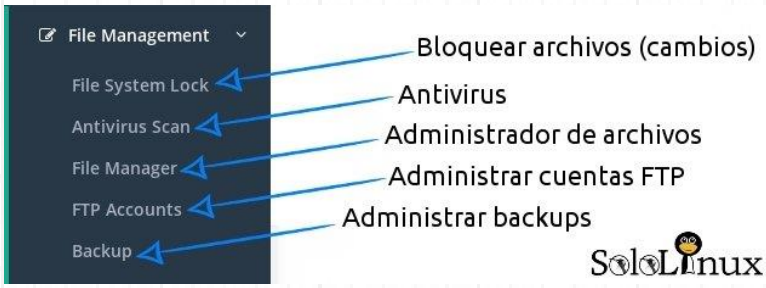


Una vez conocemos al 100% el rendimiento y calidad del servicio que nos ofrece "**Clouding.io**", proseguimos con la tercera y ultima entrega de la serie "**Instalar y configurar CentOS Web Panel – CWP**".

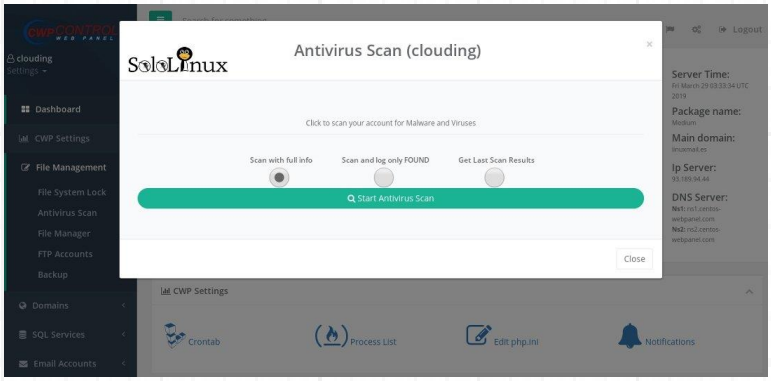
Instalar y configurar CentOS Web Panel
Recordamos que para acceder al panel del cliente se ingresa la ip del servidor con el puerto 2082. Al acceder al panel observamos que es similar a una versión anterior de **CPanel**.
En la zona frontal tienes lanzadores para casi todas las opciones, pero nosotros veremos el menú de la izquierda.
Ojo con la primera opción (**CWP Settings**) no deberías modificar nada, a no ser que sepas lo que haces.



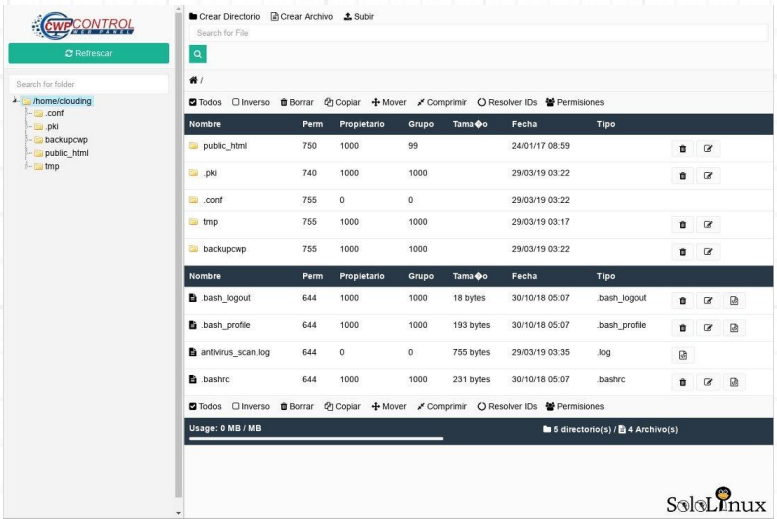
En "**File Manegement**" tenemos lo siguiente…



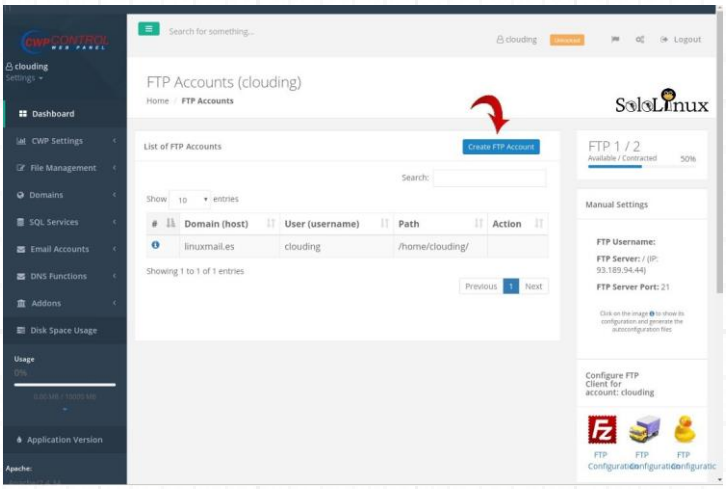
Con la primera opción podemos bloquear la modificación de todos los archivos del sitio, la siguiente es un antivirus que escanea bajo demanda tu espacio.



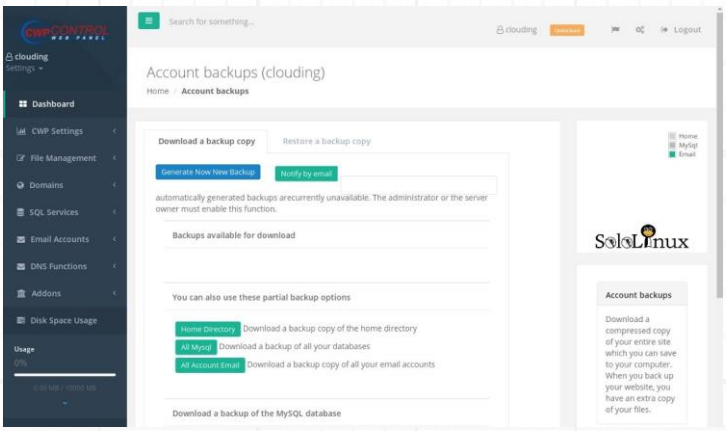
En "**File Manager**", no encontramos con un fantástico administrador de archivos, muy bueno si señor.



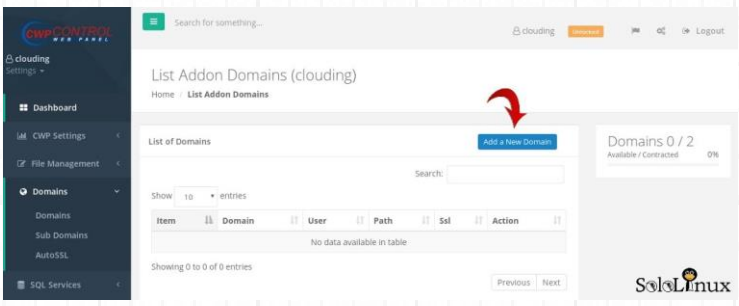
En la siguiente opción nos encontramos con el administrador de cuentas **FTP**, permite crear, eliminar cuentas, etc…



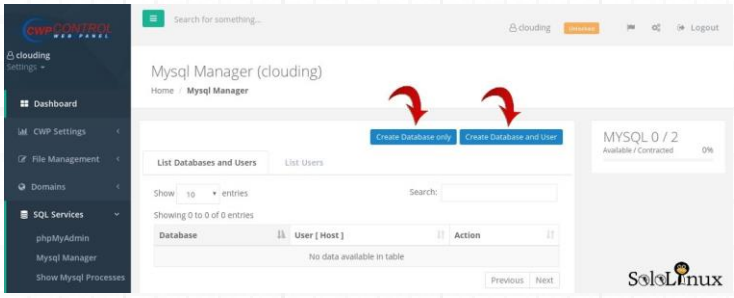
En **Backus** se permite la creación y restauración de los mismos.



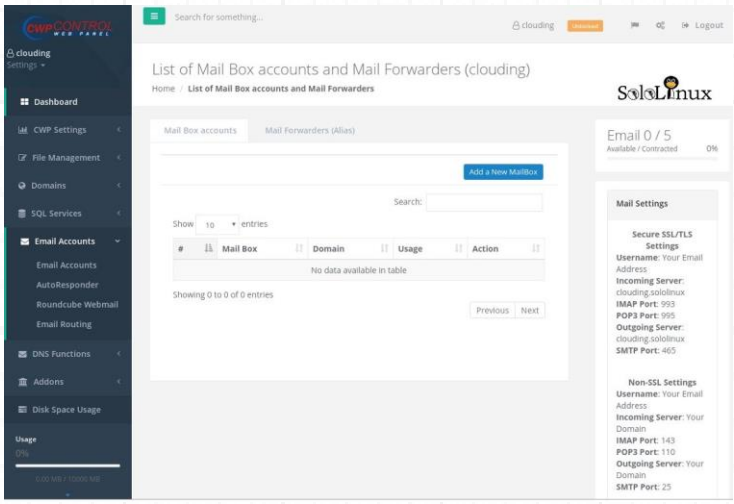
En la pestaña "**Domains**":
Agregamos nuestros dominios, creamos subdominios, y generamos certificados **Let's Encrypt** de dominio.



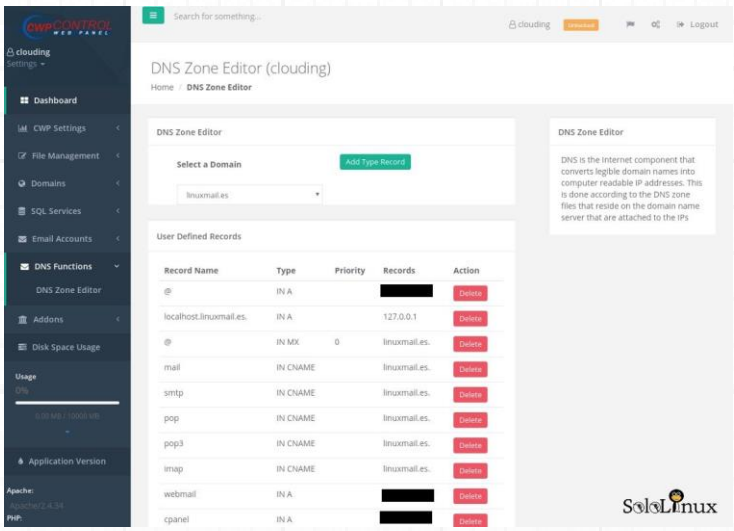
Creamos nuestra/s base de datos, tiene opción de crear una base de datos con un usuario diferente al del sistema.



Evidentemente **CentOS Web Panel Cliente**, admite la gestión de emails.

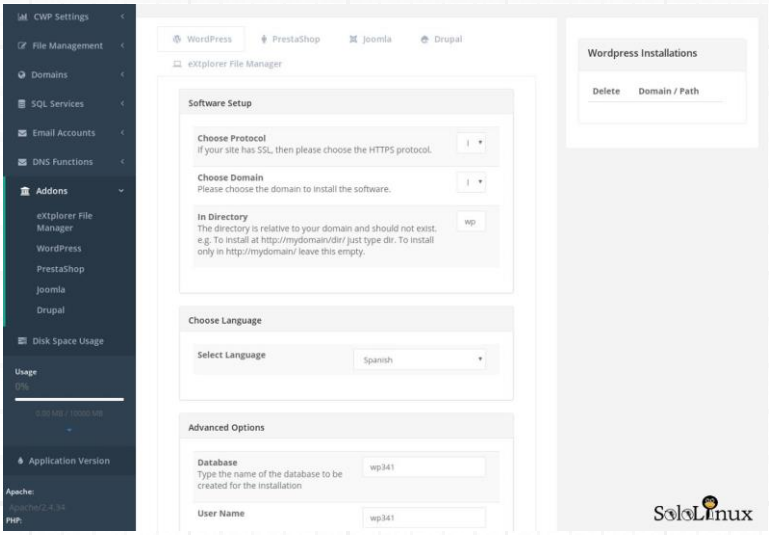


El editor de **DNS** es fácil de usar y muy rápido.



En "**Addons**" encontraras unos prefabricados listos para su instalación (la base de datos se crea automáticamente).

- **eXplorer File Manager**
- **WordPress**
- **PrestaShop**
- **Joomla**
- **Drupal**



Y como final del menú, tenemos barras indicadoras del estado de uso del disco asignado al cliente.

Una vez tengas todo configurado, puedes acceder a la url del dominio desde tu navegador web favorito. Si aun no has subido o instalado nada, te aparecerá una pantalla como esta…

Damos por finalizada la serie de artículos "**Instalar y configurar CentOS Web Panel – CWP**", si crees que es útil, comparte…

Instalar PrestaShop en Ubuntu 18.04 paso a paso

Instalar PrestaShop en **Ubuntu 18.04** paso a paso.

PrestaShop es una afamada plataforma de comercio electrónico **open source**. Se basa en **PHP** y **MySQL**, ademas uno de sus puntos fuertes es la cantidad de complementos y temas gratuitos disponibles.

En el articulo de hoy montaremos un servidor perfecto, especialmente configurado para su uso exclusivo con "**PrestaShop**".

La plataforma de comercio electrónico **PrestShop**, cuenta con unas excelentes características como una interfaz administrativa muy intuitiva, múltiples pasarelas de pago, es multilenguaje, ofrece su propia analítica e informes.

PrestaShop es una de las plataformas preferidas para montar una tienda online.

En este manual, configuraremos un **servidor** o **VPS** con **Ubuntu 18.04** e instalaremos PrestaShop en el. Como servidor web usaremos **Nginx**, **PHP 7.2** y MySQL o MariaDB.
Como único requisito, es evidente que necesitamos un dominio que apunte a un **servidor** o **VPS.**

Instalar PrestaShop en Ubuntu 18.04
Lo primero que debemos hacer es instalar "**Nginx**".

```
sudo apt update
sudo apt install nginx
```

Verificamos que está corriendo.

```
sudo systemctl status nginx
```

ejemplo de salida valida…

```
nginx.service – Startup script for nginx service
Loaded: loaded (/usr/lib/systemd/system/nginx.service;
enabled; vendor preset: enabled)
Active: active (running) since lun 2019-03-25 18:28:13 CET; 11h
ago
Main PID: 33545 (nginx)
CGroup: /system.slice/nginx.service
├─33545 nginx: master process /usr/sbin/nginx
└─33546 nginx: worker process
```

Suponemos que usaras por defecto el firewall de Ubuntu **UFW**, por tanto debemos abrir los puertos 80 y 443 entre otros.

```
sudo ufw allow 'Nginx Full‘
```

Habilitamos el servicio.

```
sudo systemctl enable nginx
```

Actualizamos e instalamos **unzip**

```
sudo apt update && sudo apt upgrade
sudo apt install unzip
```

Si quieres instalar el certificado “**Let’s Encrypt**“, revisa **el siguiente articulo de la revista**.

Instalamos y creamos la base de datos:

ATENCIÓN!!!, muchas imágenes básicas ya incluyen **MySQL** o **MariaDB** instalado, entonces debes saltarte este paso.

```
sudo apt install mysql-server mysql-client
mysql_secure_installation
```

ATENCIÓN!!!, muchas imágenes básicas ya incluyen **MySQL** o **MariaDB** instalado, entonces debes saltarte el paso anterior.

Creamos la base de datos de **PrestaShop**.

```
sudo mysql
```

La nombramos como “prestashop”.

```
CREATE DATABASE prestashop;
```

Creamos la cuenta de usuario “prestashop”.

```
GRANT ALL ON prestashop.* TO 'prestashop'@'localhost' IDENTIFIED BY 'password-de-la-db';
```

Ya tenemos la base de datos con su usuario, salimos de la consola MySQL.

```
EXIT;
```

Instalamos y configuramos PHP 7.2:

```
sudo apt install php7.2-common php7.2-cli php7.2-fpm php7.2-opcache php7.2-gd php7.2-mysql php7.2-curl php7.2-intl php7.2-xsl php7.2-mbstring php7.2-zip php7.2-bcmath php7.2-soap
```

Configuramos las **opciones de php** recomendadas.

```
1 sudo sed -i "s/memory_limit = .*/memory_limit = 1024M/" /etc/php/7.2/fpm/php.ini
2 sudo sed -i "s/upload_max_filesize = .*/upload_max_filesize = 256M/" /etc/php/7.2/fpm/php.ini
3 sudo sed -i "s/zlib.output_compression = .*/zlib.output_compression = on/" /etc/php/7.2/fpm/php.ini
4 sudo sed -i "s/max_execution_time = .*/max_execution_time = 18000/" /etc/php/7.2/fpm/php.ini
5 sudo sed -i "s/;date.timezone.*/date.timezone = UTC/" /etc/php/7.2/fpm/php.ini
6 sudo sed -i "s/;opcache.save_comments.*/opcache.save_comments = 1/" /etc/php/7.2/fpm/php.ini
```

Descargamos PrestaShop:
En la **pagina oficial de descargas**, vemos que la ultima versión estable es la **1.7.5.1**, así que la bajamos con el comando "**wget**".

```
cd /tmp
wget https://download.prestashop.com/download/releases/prestashop_1.7.5.1.zip
```

Creamos la carpeta que contendrá la tienda online (con tu dominio).

```
sudo mkdir -p /var/www/html/tudominio.com
```

Descomprimimos el paquete **PrestaShop**.

```
unzip prestashop_*.zip
```

El paquete **PrestaShop** contiene otro **zip**, lo descomprimimos, y lo movemos a la carpeta de la **tienda online** que creamos antes.

```
sudo unzip prestashop.zip -d /var/www/html/tudominio.com
```

Solo nos falta modificar los permisos.

```
sudo chown -R www-data: /var/www/html
```

Configurar Nginx:
Creamos el archivo de configuración.

```
sudo nano /etc/nginx/sites-available/tudominio.com
```

Copia y pega lo siguiente (con tu dominio).

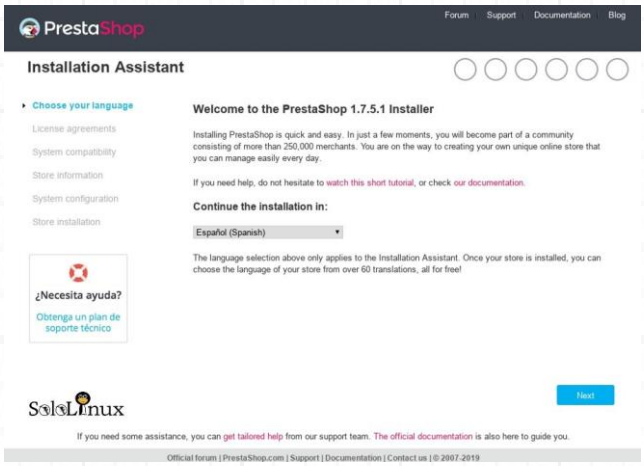
```
# Redirect HTTP -> HTTPS
server {
    listen 80;
    server_name www.tudominio.com tudominio.com;

    include snippets/letsencrypt.conf;
    return 301 https://tudominio.com$request_uri;
}

# Redirect WWW -> NON WWW
server {
    listen 443 ssl http2;
    server_name www.tudominio.com;

    ssl_certificate /etc/letsencrypt/live/tudominio.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/tudominio.com/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/tudominio.com/chain.pem;
    include snippets/ssl.conf;

    return 301 https://tudominio.com$request_uri;
}
```

```
server {
    listen 443 ssl http2;
    server_name tudominio.com;

    root /var/www/html/tudominio.com;
    index index.php;

    # SSL parameters
    ssl_certificate /etc/letsencrypt/live/tudominio.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/example.com/chain.pem;
    include snippets/ssl.conf;
    include snippets/letsencrypt.conf;

    # log files
    access_log /var/log/nginx/tudominio.com.access.log;
    error_log /var/log/nginx/tudominio.com.error.log;

    location = /favicon.ico {
        log_not_found off;
        access_log off;
    }

    location = /robots.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.2-fpm.sock;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg)$ {
        expires max;
        log_not_found off;
    }

}
```

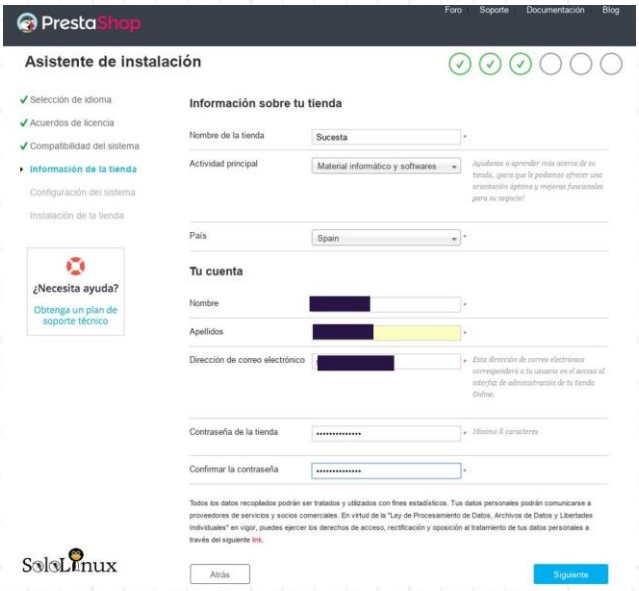
Guarda el archivo y cierra el **editor nano**.
Verificamos que la sintaxis es correcta.
sudo nginx -t

ejemplo de salida valida…
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
Reiniciamos Nginx.
sudo systemctl restart nginx

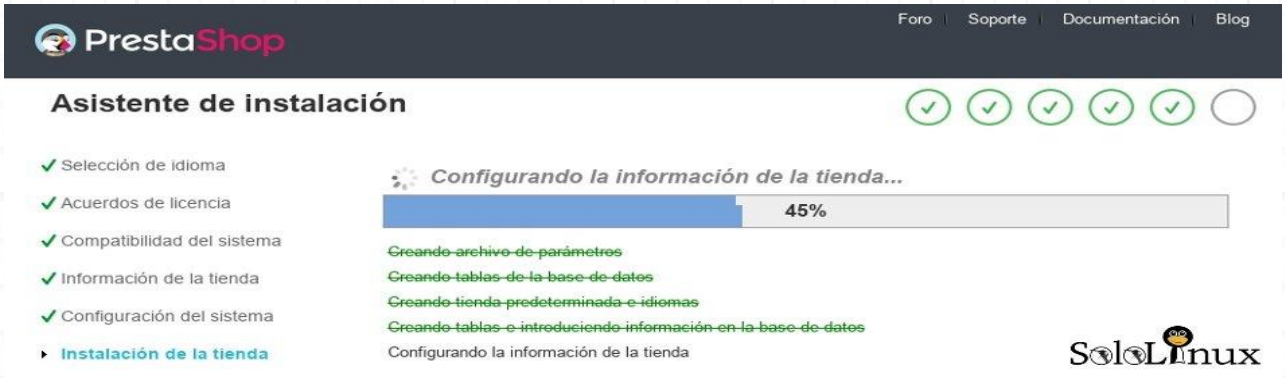
Instalar PrestaShop:
Llego el momento de **instalar PrestaShop** propiamente dicho, para ello abre tu **navegador web** preferido y escribe el dominio de tu **tienda online**.
Comienza la instalación.



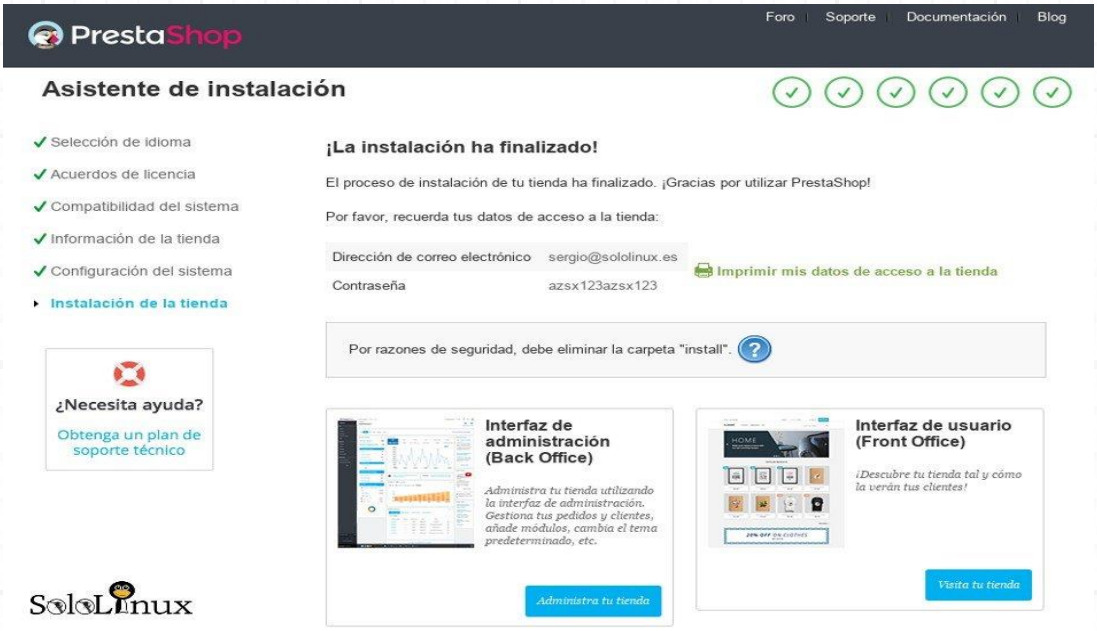
Verifica la compatibilidad del sistema.



Comienza la instalación de PrestaShop.

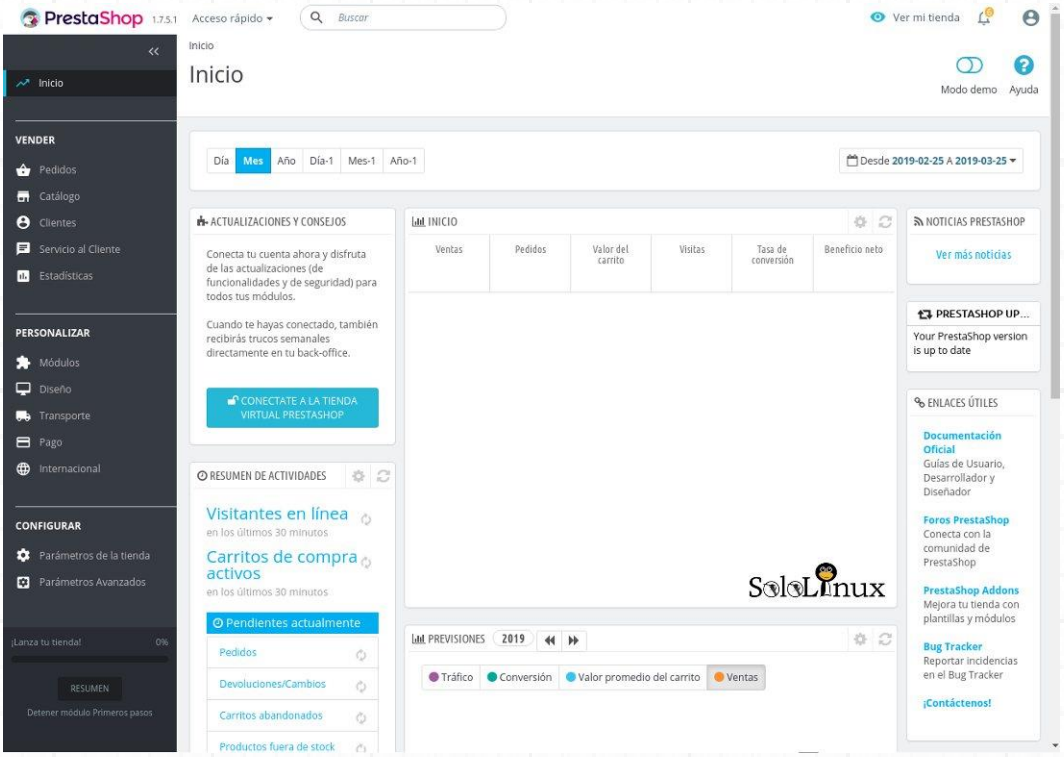


Al concluir la instalación, en la parte inferior de la pantalla podrás ver dos botones, haz click en "**Administra tu tienda**".

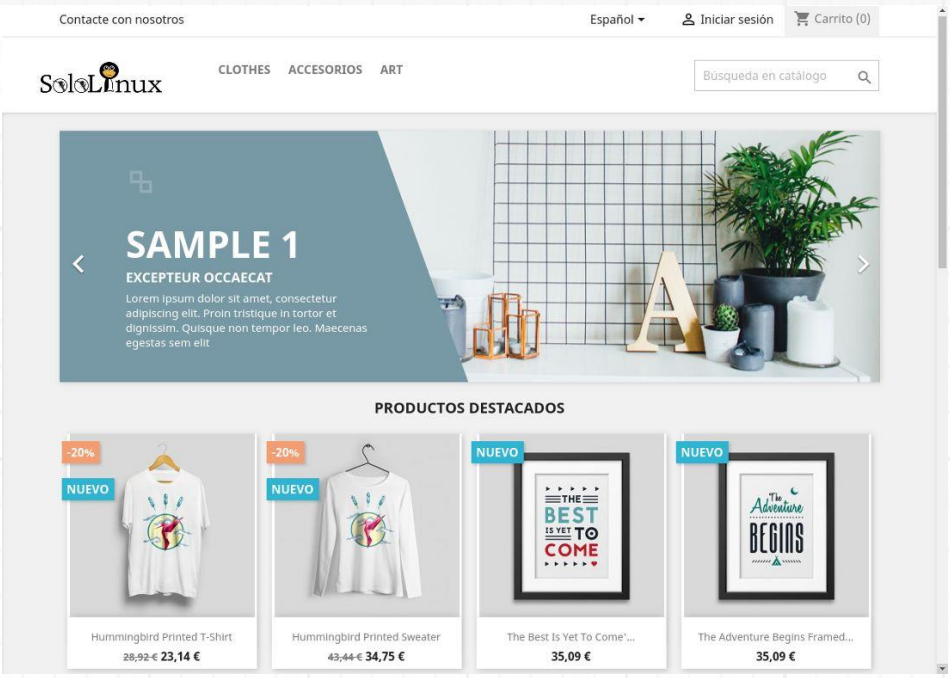


Para acceder escribimos los datos que insertamos en el paso anterior (datos de tu cuenta).

Accedemos a la zona de administración de la tienda. En la zona superior derecha tienes la opción "**Ver mi tienda**".



Como puedes ver la tienda funciona correctamente.



Enhorabuena!!!, ya tienes instalado tu propio "**servidor PrestaShop**".
Comparte el articulo "**Instalar PrestaShop en Ubuntu 18.04 paso a paso**".

Instalar Squid Proxy Server en Ubuntu 18.04

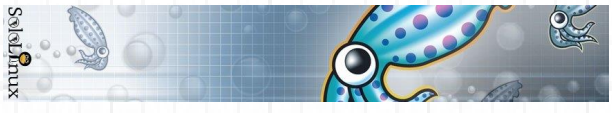
Instalar Squid Proxy Server en **Ubuntu 18.0**

Squid es una aplicación **proxy** basada en **Linux**, que nos aporta una serie de valores añadidos como filtrar el tráfico según nuestras necesidades, implementar una seguridad añadida y la búsqueda de **DNS**.

Independientemente de lo anterior, **Squid** también mejora considerablemente el rendimiento de un servidor web al almacenar en caché los recursos.

Un **servidor Squid** nos permite almacenar en caché las páginas web más visitadas. Su forma de operar es simple, cuando un usuario solicita una página web o un archivo, esa solicitud se transmite al servidor proxy (ejerce de intermediario entre el usuario y el sitio web). El servidor proxy extrae los datos de su cache y los envía al usuario que los solicita.

En este articulo aprenderemos como **instalar Squid Proxy Server** y a configurarlo correctamente, ya sea en **Ubuntu 18.04**, **Ubuntu 16.04** o sus derivados.



Instalar Squid Proxy Server en Ubuntu 18.04
Actualizamos el sistema.
sudo apt-get update
Instalamos el paquete "**squid**".
sudo apt-get install squid
Después de aceptar la confirmación para **instalar Squid**, comienza la instalación…

```
Selecting previously unselected package squid.
Preparing to unpack .../6-squid_3.5.27-1ubuntu1.1_amd64.deb ...
Unpacking squid (3.5.27-1ubuntu1.1) ...
Processing triggers for ufw (0.35-5) ...
Processing triggers for ureadahead (0.100.0-20) ...
Setting up ssl-cert (1.0.39) ...
Setting up libecap3:amd64 (1.0.1-3.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.15) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Setting up squid-langpack (20170901-1) ...
Setting up squid-common (3.5.27-1ubuntu1.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up libdbi-perl (1.640-1) ...
Setting up squid (3.5.27-1ubuntu1.1) ...
Setcap worked! /usr/lib/squid/pinger is not suid!
Skipping profile in /etc/apparmor.d/disable: usr.sbin.squid
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.15) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for ufw (0.35-5) ...
```

Iniciamos el servicio, y le indicamos que arranque con el sistema.
sudo systemctl start squid
sudo systemctl enable squid
Apunte: *Si quieres detener el servicio*.
sudo systemctl stop squid
Apunte: *Si quieres que no arranque con el sistema*.
sudo systemctl disable squid
Apunte: *Si quieres reiniciar Squid*.
sudo systemctl restart squid
Una vez iniciado "**Squid**" verificamos que opera correctamente.
sudo systemctl status squid

Ejemplo de salida correcta…

```
root@squid:~# 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 2019-03-28 04:20:07 UTC; 2min 26s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/squid.service
           ├─2182 /usr/sbin/squid -YC -f /etc/squid/squid.conf
           ├─2184 (squid-1) -YC -f /etc/squid/squid.conf
           ├─2192 (logfile-daemon) /var/log/squid/access.log
           └─2193 (pinger)

Mar 28 04:20:07 squid systemd[1]: Starting LSB: Squid HTTP Proxy version 3.x...
Mar 28 04:20:07 squid squid[2117]:  * Starting Squid HTTP Proxy squid
Mar 28 04:20:07 squid squid[2182]: Squid Parent: will start 1 kids
Mar 28 04:20:07 squid squid[2117]:    ...done.
Mar 28 04:20:07 squid systemd[1]: Started LSB: Squid HTTP Proxy version 3.x.
Mar 28 04:20:07 squid squid[2182]: Squid Parent: (squid-1) process 2184 started
```

Configurar Squid Proxy Server en Ubuntu
La configuración de **Squid Proxy** consta de tres pasos.

- Configuración básica.
- Configuración adicional.
- Bloquear sitios web y palabras clave.

Configuración básica:
El archivo de configuración de **Squid** lo puedes localizar en: **/etc/squid/squid.conf**.
Buscamos la linea "**http_port 3128**", que es el puerto por defecto para el tráfico **TCP**. Si tu red está configurada para el tráfico en otro puerto, lo cambias.
sudo nano /etc/squid/squid.conf

Si quieres evitar que **Squid** modifique las solicitudes y respuestas, activamos el modo proxy transparente.
sudo nano /etc/squid/squid.conf

Modificas…	por…
http_port 3128	http_port 1234 transparent

Por defecto **Squid** no permite el trafico "**http**", debemos permitirlo.
sudo nano /etc/squid/squid.conf

Buscamos la linea "**http_acacess**", y modificamos…
http_acacess deny por http_access allow all

Como ultimo punto de la configuración básica insertamos un **hostname**, escribe lo que quieras en la opción "**visible_hostname**".
Guarda el archivo y cierra el editor.
Reiniciamos **Squid Proxy Server**.
sudo systemctl restart squid

Configuración adicional:
Crear un ACL
Squid nos permite crear un listado de control de acceso, por ejemplo, podemos crear una regla que solo permite la conexión del sistema a una dirección IP.
sudo nano /etc/squid/squid.conf

Agregamos lo siguiente: acl localnet src 192.168.0.42 #Mi_sistema

Se permiten rangos. acl localnet src 192.168.0.42/30 #Mis_sistemas

También es posible abrir
puertos específicos: acl Safe_ports port 123 # Conexión_entre_mis_sistemas

Configurar la autentificación
Para obligar a los usuarios del sistema a autentificarse, debemos instalar **apache2-utils**:
sudo apt-get install apache2-utils

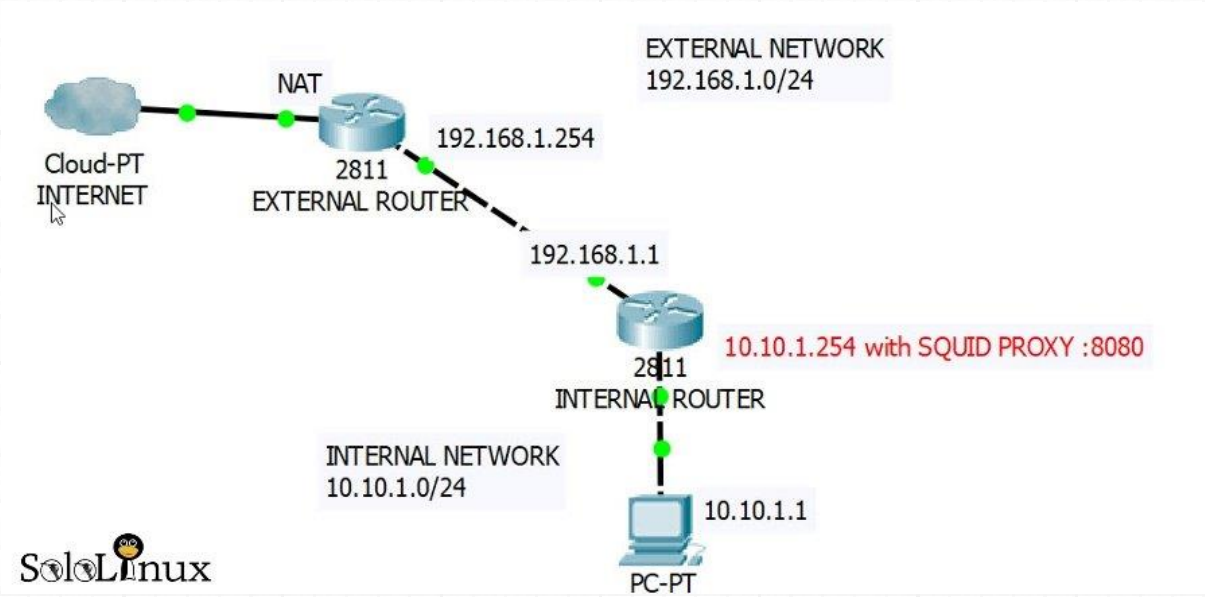
Creamos el archivo "**passwd**" y modificamos el propietario al usuario de **Squid**
sudo touch /etc/squid/passwd
sudo chown proxy: etc/squid/passwd

Ahora agregamos un usuario con contraseña (introduce el usuario y la contraseña).
sudo htpasswd /etc/squid/passwd newuser

Editamos el archivo de configuración otra vez.
sudo nano /etc/squid/squid.conf

Agregamos lo siguiente…
auth_param basic program /usr/lib64/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 2 hours
acl auth_users proxy_auth REQUIRED
http_access allow auth_users

Guarda el archivo y cierra el editor.
Reiniciamos **Squid Proxy Server**.
sudo systemctl restart squid



Bloquear sitios web y palabras clave:
Este paso es bastante sencillo, creamos y editamos el archivo "**blocked.acl**".
sudo nano /etc/squid/blocked.acl

Agregamos los sitios web a bloquear, no te olvides de insertar un punto antes del sitio (si no colocas el punto solo bloqueara el sitio principal, el resto de direcciones del sitio web estarán activas).
por ejemplo…
.facebook.com
.twitter.com
.instagram.com

Guarda el archivo y cierra el editor.
Abrimos el archivo de configuración principal.
sudo nano /etc/squid/squid.conf

Inserta lo siguiente antes de la lista de reglas "**ACL**" que vimos antes.
acl blocked_websites dstdomain "/etc/squid/blocked.acl"
http_access deny blocked_websites

Guarda el archivo y cierra el **editor nano**.
Reiniciamos **Squid**.
sudo systemctl restart squid

Hemos terminado de **Instalar Squid Proxy Server en Ubuntu 18.04**, ya está operativo, ahora bien… si quieres navegar a través de **Squid Proxy** debes configurar el cliente, pero eso en un próximo articulo.

Comparte el articulo "Instalar Squid Proxy Server en Ubuntu 18.04".

Ocultar la IP del sistema en Squid

Ocultar la IP del sistema en **Squid** y otros tips.
Por defecto **Squid Proxy Server** muestra la dirección IP del sistema en los encabezados.
En este mini tutorial veremos como **ocultar la ip** de manera simple y rápida, también aportaremos algún consejo sobre configuraciones de seguridad.

Ocultar la IP del sistema en Squid
Squid tiene una directiva llamada **"forwarded_for"**, que si está configurada en **"on"** incluirá la dirección IP del sistema al responder a las solicitudes HTTP que reenvíes.
ejemplo…

```
7703 #  X-Forwarded-For entries, and place the client IP as the sole entry.
7704 #Default:
7705 forwarded_for on
7706
```

SoloLinux

Se visualiza algo similar a:
X-Forwarded-For: ip-del-sistema

Para solucionarlo editamos el archivo de configuración.
sudo nano /etc/squid/squid.conf

Buscamos la regla **"forwaded_for"**, que localizaras cerca de la linea 7700.
Modificamos…
forwarded_for on
por…
forwarded_for off

Guardamos el archivo y cerramos el editor.
Reiniciamos **Squid**.
sudo systemctl restart squid

Otros Tips de seguridad

Sobre la linea 5114 busca lo siguiente:

 # If set (default), Squid will include a Via header in requests and
 # replies as required by RFC2616.
#Default:
via on

Lo sustituyes por: via off

Para una privacidad optima, asegúrate que los "request_header_access" están como te indico a continuación.

request_header_access Authorization allow all
request_header_access Proxy-Authorization allow all
request_header_access Cache-Control allow all
request_header_access Content-Length allow all
request_header_access Content-Type allow all
request_header_access Date allow all
request_header_access Host allow all
request_header_access If-Modified-Since allow all
request_header_access Pragma allow all
request_header_access Accept allow all
request_header_access Accept-Charset allow all
request_header_access Accept-Encoding allow all
request_header_access Accept-Language allow all
request_header_access Connection allow all
request_header_access All deny all

Guarda el archivo y cierra el editor.

Reiniciamos **Squid**.

sudo systemctl restart squid

Si te gusto el minitutorial. Comparte el articulo "Ocultar la IP del sistema en Squid".

THANKS!

TU PUBLICIDAD AQUI

QUIERES APARECER EN LA REVISTA, GANAR CON ELLO MAS VENTAS EN TU WEB, MAS SEGUIDORES EN TUS REDES SOCIALES...

SOLO TIENES QUE MANDAR UN CORREO A

adrian@sololinux.es

Y TE EXPLICAMOS COMO

SoloLinux

www.sololinux.es

Instalar Aircrack en Android

Instalar Aircrack en Android.

Hijacker es una interfaz gráfica que permite de manera cómoda el uso de las herramientas de penetración **Aircrack-ng, Airodump-ng**, **MDK3** y **Reaver**.
La interfaz de usuario es simple y sencilla de usar, lo mejor es… que no tendrás que escribir los comandos en una consola de **Android**, ni copiar y pegar las **direcciones MAC**.
Esta aplicación requiere un **Android ARM** (la mayoría de los ***smartphones*** modernos ya usan este tipo de CPU)**,** y un adaptador inalámbrico interno que admita el **modo**

Monitor.
Algunos dispositivos admiten el modo monitor (no todos) pero ninguno de ellos de forma nativa, y esto si que supone un inconveniente, pues será necesario cambiar el firmware.
Todos los ***smartphones*** que usen el chipset BCM4339 (MSM8974, como los Nexus 5, Xperia Z1 / Z2, LG G2, LG G Flex, Samsung Galaxy Note 3) funcionarán con el firmware **Nexmon** (acccede al enlace para ver la lista de compatibles). Los dispositivos que usan el chip BCM4330 pueden usar **bcmon**.
Como alternativa, se puede usar un adaptador externo que admita el **modo monitor** en Android, conectado con un **cable OTG**.
Se incluyen las herramientas necesarias para los dispositivos **armv7l** y **aarch64**. También se incluye la utilidad de administración y el controlador **Nexmon** para **BCM4339** y **BCM4358** .
El acceso root es necesario, como es lógico estas herramientas necesitan acceso a la raíz para funcionar correctamente.

Instalar Aircrack-Características
Recopilar información

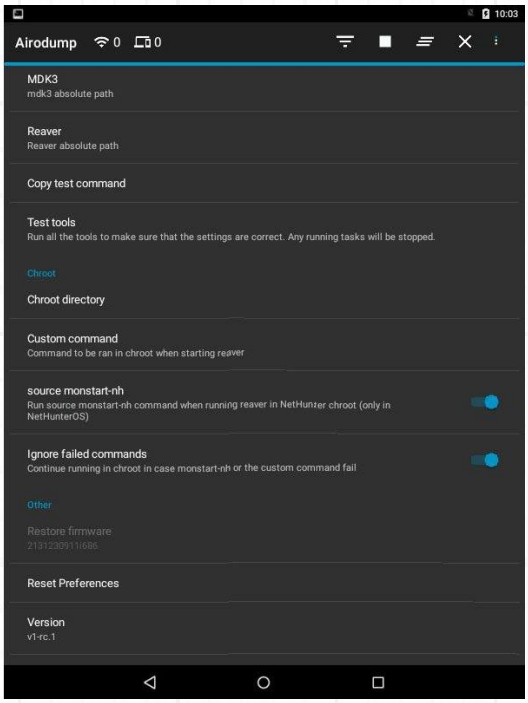
- Ver el listado de puntos de acceso y los clientes (incluso los ocultos).
- Examinar la actividad de una red específica y sus clientes.
- Estadísticas sobre los puntos de acceso y sus clientes.
- Identifica el fabricante de un dispositivo (AP o estación) consultando la base de datos OUI.
- Ver la intensidad de la señal de los dispositivos y filtrar los más cercanos.
- Guarda los paquetes capturados en el archivo .cap.

Tipos de ataques

- **Desautentificar** todos los clientes de una red (ya sea a uno en particular, o sin definir el objetivo).
- Desconectar un cliente específico de la red que estamos conectados.
- **MDK3 Beacon Flooding** con opciones personalizadas y lista SSID.
- Autenticación de **MDK3 DoS** para una red específica o para un AP cercano.
- Capturar el **Handshake WPA.**
- Craqueo de **WPS** con **Reaver.**

Otros detalles

- La aplicación continua ejecutándose en segundo plano, permite una notificación al concluir.
- Copiar comandos o direcciones MAC al portapapeles.
- Incluye las herramientas necesarias, no es necesario una instalacion manual.
- Incluye el controlador Nexmon, la biblioteca necesaria y la utilidad de administración para dispositivos BCM4339 y BCM4358.
- Establece los comandos para habilitar y deshabilitar el modo de monitor de forma automática.
- Crack de los archivos ***.cap** desde una lista de palabras personalizada.
- Crea **acciones personalizadas** y las ejecutas en un AP o un cliente.
- Ordena y filtra puntos de acceso y estaciones de trabajo aportando sus detalles.
- Exporta toda la información recogida a un archivo.
- Agrega un **alias** persistente a un dispositivo (por MAC) para identificarlo rápidamente.



Instalar Aircrack

Requisitos mínimos:

- Android 5+
- Ser root (se requiere SuperSU , si está en CM / LineageOS, instale SuperSU)
- Firmware que admita el modo de monitor en su interfaz inalámbrica, o hacer uso de una externa.

Descarga la ultima versión:

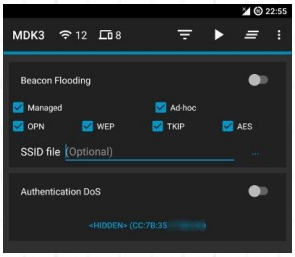
Hijacker

Al ejecutar **Hijacker** por primera vez, te preguntará si quieres instalar el **firmware nexmon** o ir a la pantalla de inicio. Si has instalado el firmware o utilizas un adaptador externo, seleccionas ir a la pantalla de inicio. En caso de que no lo hayas instalado aun (si es compatible), haces clic en '**Instalar Nexmon**'.
Al regresar a la pantalla principal se iniciará **airodump**.

Asegúrate de que tienes habilitada la WiFi y que está en modo de monitor.

Atención: en algunos dispositivos, el cambio de archivos /system puede activar las funciones de seguridad de Android y la partición del sistema se restaurará al reiniciar.

Ejemplos de uso:



Solucionar problemas al instalar Aircrack

Esta aplicación está diseñada y probada para uso exclusivo en dispositivos ARM. Todos los binarios incluidos están compilados para esta arquitectura. Puedes comprobar si tu dispositivo es compatible de manera simple, ves a Configuración: si tienes la opción de instalar Nexmon, tu dispositivo es compatible.
En la configuración, hay una opción para que puedas probar las herramientas. Si algo falla, puedes hacer clic en 'Copiar comando de prueba' y seleccionar la herramienta que falla. Esto copiará un comando de prueba en el portapapeles, que puedes ejecutar manualmente en un shell de **root** y ver qué está fallando.
Si pasas todas las pruebas y el problema persiste, usa la opción 'Enviar comentarios' que tienes en la configuración.
Si la aplicación falla, se iniciará una nueva actividad que generará un informe de error en el almacenamiento externo y te ofrecerá la opción de enviarlo por correo electrónico. El informe se muestra en la actividad para que puedas ver lo que vas a enviar.

No envíes informes sobre errores en dispositivos que no sean compatibles.

Recuerda que **Hijacker** es solo una **GUI** para estas herramientas. La ejecución de las herramientas es la común. Si las pruebas efectuadas son validas y estás en modo monitor, obtendrás los resultados que buscas.

Responsabilidad legal
Debes tener presente que usar esta aplicación para extraer datos de otras redes sin permiso es un delito. Solo se permite su uso en tu propia red u otras en las cuales estas autorizado.
El uso de aplicaciones que hacen uso del modo monitor del adaptador de red, puede considerarse un delito.
Tu eres el responsable de como uses la herramienta y de los daños que puedas causar.

Dispositivo
La aplicación te ofrece la opción de instalar el **firmware Nexmon**, pero a pesar de que realiza una comprobación exhaustiva los errores pueden ocurrir.
La herramienta incluye los firmware para Nexmon **BCM4339** y **BCM4358**. Instalar el firmware incorrecto en un dispositivo wifi puede dejarlo inservible para siempre (a nivel de hardware, sin posibilidad de recuperación).

SOLOLINUX NO ES responsable de ningún daño causado en tu aparato. Si no estas seguro no hagas nada.

Github oficial.

Si este articulo te ayudo, entra en la WEB y compártelo. Comenta tus dudas en el articulo, serán respondidas lo mas rápido posible. ☺

Instalar MyWebSQL en Ubuntu 18.04

Instalar MyWebSQL en **Ubuntu 18.04** y derivados.

MyWebSQL es un cliente WYSIWYG open source basado en web, tiene similitudes con **phpMyAdmin** pero algo más rápido y ágil a la hora de administrar bases de datos de un servidor (**Adminer** sigue siendo el más rápido, es evidente, tan solo es un archivo php).

MyWebSQL nos proporciona una bonita interfaz gráfica, ademas de simple e intuitiva con la apariencia de una aplicación de escritorio.

La aplicación tiene un amplio conjunto de características y un montón de herramientas que te ayudaran en la gestión de tus bases de datos.

Es compatible con MySQL, MariaDB, PostgreSQL y SQLite.

Sus principales características son:

- Sintaxis múltiple con remarcado en los editores de SQL.
- WYSIWYG que también admite crear y editar tablas.
- Edición rápida multi-registro.
- Similar apariencia a una aplicación de escritorio.
- Excelente soporte con los navegadores más populares.
- Instalación y configuración limpia.
- Interfaz multilingüe (incluye el castellano) con soporte de temas.
- Soporta bases de datos MySQL, MariaDB, PostgreSQL y SQLite.
- Importa bases de datos, exporta bases de datos, tablas o diversos resultados a múltiples formatos.
- Soporte para PHP 7.2.

En este articulo vemos como instalar **MyWebSQL** en Ubuntu 18.04 LTS y derivados. Su interfaz web rápida y atractiva, hacen que sea una decente alternativa al popular **phpMyAdmin**.

Instalar MyWebSQL en Ubuntu 18.04

Como único requisito, se necesita que tengas instalado como mínimo alguna de estas dos opciones:

- **LAMP**
- **LEMP**

Antes de **instalar MyWebSQL** actualizamos el sistema.

```
sudo apt update
sudo apt upgrade
```

Nos aseguramos que tenemos instaladas las dependencias de php requeridas.

```
apt install wget zip php-pgsql php-mysql php-bcmath php-curl php-gmp
```

Si utilizas **SQLite** instala lo siguiente…

```
apt-get install php-sqlite3
```

Descargamos e instalamos la herramienta.

```
wget https://newcontinuum.dl.sourceforge.net/project/mywebsql/stable/mywebsql-3.7.zip
unzip mywebsql-3.7.zip -d /var/www/html
```

Aplicamos los permisos necesarios.

```
chown -R www-data:www-data /var/www/html/mywebsql/
chmod -R 775 /var/www/html/mywebsql/
```

Ejecuta desde tu navegador web lo siguiente:

http://tu-ip/mywebsql/install.php

Una vez concluya el proceso, eliminamos el archivo de instalación.

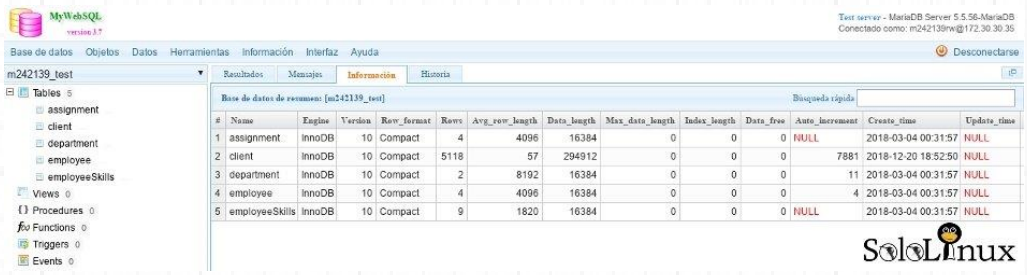
```
rm /var/www/html/mywebsql/install.php
```

La aplicación está instalada.

Para acceder a **MyWebSQL**, tan solo tienes que acceder desde tu **navegador web** preferido a:

http://tu-ip/mywebsql

En la pantalla de inicio te solicitara el nombre de usuario y el **password** de la base de datos (tal vez sea el root y su pass).



Si te fue útil este articulo, compártelo "**Instalar MyWebSQL en Ubuntu 18.04**".

Chrome vs Chromium cual elegir

Chrome vs Chromium cual elegir.
Todos sabemos que **Google Chrome** es actualmente el navegador más popular entre los usuarios de internet, ya que cuenta con prácticamente el 60% de navegantes a nivel mundial en pc's de escritorio.

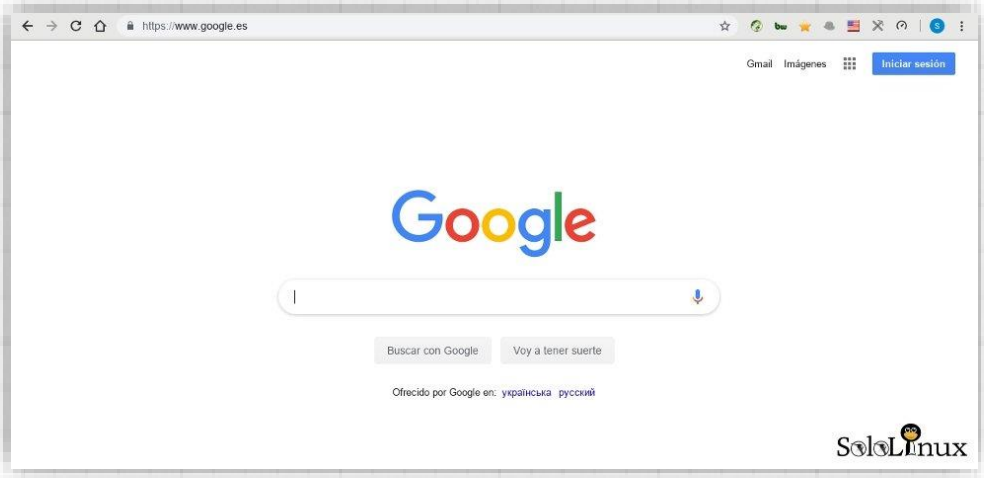
Pero en el mundillo Linux la cosa cambia, **Chrome** no es el más usado, en casi todas las **distribuciones linux** viene incluido **Firefox**, en otras se incluye **Chromium**, jamas **Chome**.
Hoy analizamos las diferencias entre **Chrome y Chromium**, que para mi, este ultimo está un tanto menospreciado simplemente por que su desarrollo lo comenzó **Google**.
A todos los efectos **Chromium** es prácticamente igual a **Google Chrome**. Comparten todo, desde las extensiones, al motor y sus características.

Entonces… que los diferencia?

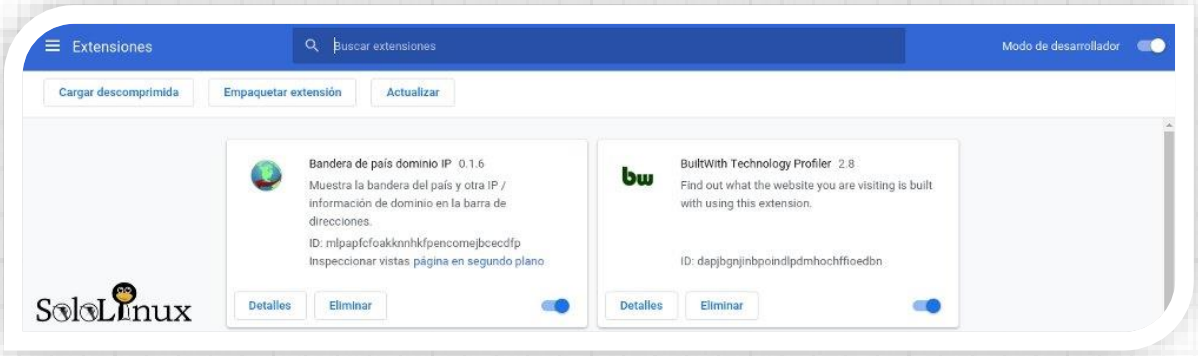
Chrome vs Chromium
En el año 2008 Google presentó el Proyecto Chromium. El Proyecto Chromium es el proyecto **open source** que está detrás del afamado navegador Chrome de Google y de su sistema operativo Google Chrome.
Chromium inicia y desarrolla el navegador. Posteriormente, Google obtiene el código puro del navegador y le agrega sus productos y servicios que no son **open source**.
Por eso la gran mayoría de distribuciones de Linux, no contienen a **Google Chrome** en sus repositorios, rompen el espíritu del "código abierto (ness)".

La interfaz es muy similar, perdón, dije similar.

Es prácticamente la misma, eres capaz de distinguirla?



Si analizas en profundidad podrías encontrar algunas diferencias ínfimas, pero pocas, pocas, muy pocas.
Las mayores diferencias apreciables son los productos o herramientas que no están disponibles en código abierto, como por ejemplo el lector de PDF, Flash y codecs de audio y video (MP3, AAC, H.264), etc.…
Las extensiones también son las mismas:



A titulo personal me quedo con **Chromium**, creo que con su no disponibilidad de ese tipo de herramientas propietarias es mucho más rápido.
Pd: Recuerda que **Chrome** no es de código abierto, por tanto no puedes modificar nada (o solo lo que ellos te permiten).

Tu decides.

Comparte el articulo. "Chrome vs Chromium cual elegir".

Instalar Firefox Beta en Ubuntu, Fedora y derivados

Instalar Firefox Beta en **Ubuntu**, **Fedora** y todos sus derivados.
Como norma general siempre debemos usar en producción la ultima versión de **Firefox estable**. Pero si eres un entusiasta de **Firefox**, o simplemente quieres utilizar o probar lo que se avecina, en este articulo de hoy aprenderás como instalar la ultima versión de **Firefox Beta** de manera limpia y sencilla.
Esta instalación no solo es para Ubuntu y Fedora, también es valida para Debian, Linux Mint, CentOS, OpenSuse, etc..

Instalar Firefox Beta en Ubuntu, Fedora, etc…
Lo primero que haremos es descargar la ultima versión de **Firefox Beta**, lo podemos hacer desde dos sitios diferentes.

- Descargar la ultima versión de tu región.
- Descargar seleccionando idioma y arquitectura.

En nuestro caso la ultima versión disponible para su descarga es…

firefox-67.0b4.tar.bz2

Una vez la tengamos en nuestra carpeta de descargas, descomprimimos el paquete.

```
cd Descargas
tar xjf firefox-67.0b4.tar.bz2
```

Ahora, y solo si ya tenias alguna versión en "**/opt**", debes borrarla.

```
sudo rm -r /opt/Firefox
```

Movemos el nuevo **Firefox Beta** a "/opt" (desde la carpeta de descargas).

```
cd Descargas
sudo mv firefox /opt/Firefox
```

Creamos los enlaces simbólicos de Firefox Beta
Guardamos el antiguo.

```
sudo mv /usr/bin/firefox /usr/bin/firefox-old
```

Creamos uno nuevo.

```
sudo ln -s /opt/firefox/firefox /usr/bin/firefox
```

No es necesario actualizar los iconos y accesos directos, deberían lanzar la nueva versión de Firefox Beta.

Instalar la Beta desde PPA
Si eres usuario de **Ubuntu** o **Linux Mint**, puedes instalar la beta directamente desde sus repositorios oficiales.
Agregamos el ppa.
sudo add-apt-repository ppa:mozillateam/firefox-next

Instalamos **Firefox Beta**.
sudo apt update && sudo apt install firefox

Ya lo tenemos instalado.

Comparte el articulo "**Instalar Firefox Beta en Ubuntu, Fedora y derivados**".

Instalar Wine 4.0 en Ubuntu y derivados

Instalar Wine 4.0 en **Ubuntu**, **Linux Mint** y todos sus derivados.

No hace mucho se lanzo la nueva versión de **Wine**, conocida como **Wine 4.0**, recordemos que esta herramienta nos permite ejecutar aplicaciones y juegos nativos de **Windows** en **Linux** u otros sistemas operativos basados en **UNIX**.

La nueva versión de **Wine** se lanzo como una autentica revolución, más de 6,000 modificaciones que acercan aun más las aplicaciones de la ventana (win) a nuestro Linux.

Las principales características del nuevo **Wine 4.0**, son:

- **Soporte Vulkan.**
- **Soporte Direct3D 12.**
- **Implantación de funciones adicionales de Direct3D 10 / 11.**
- **Soporte para controladores de juegos HID.**
- **Actualizaciones en la base de datos.**
- **Mejoras en el cuadro de diálogo de archivos, incluido el tamaño del archivo, etc.**
- **Soporte del cursor del mouse en Android.**
- **Soporte HiDPI en Android.**
- **Múltiples correcciones de errores.**

Ahora mismo ya son más de **26,000 aplicaciones y juegos de Windows** compatibles con **Wine**, entre las cuales y aunque parezca mentira… aplicaciones tan afamadas como **Photoshop** y **Microsoft Office**, y juegos populares como pueden ser ***StarCraft***, ***Counterstrike*** o ***Team Fortress***.

Probablemente Wine 4.0 estará disponible en el nuevo Ubuntu 19.04 que llegara en Abril de este año 2019.

Si no quieres esperar para actualizar o instalar Wine 4.0 en Ubuntu 18.04 LTS o 18.10, ejecutaremos los siguientes comandos en nuestra consola / terminal.
IMPORTANTE!!!!! Si tienes instalado un Wine desde PPA o repo instalado, debes eliminarlo antes de continuar.

Instalar Wine 4.0 en Ubuntu
Primero descargamos y agregamos la key.
wget -nc https://dl.winehq.org/wine-builds/winehq.key
sudo apt-key add winehq.key

Una vez instalada la key, y dependiendo de tu versión de **Ubuntu** agregaremos un repositorio u otro.
Ubuntu 18.10 (cosmic) y todos sus derivados:
sudo apt-add-repository 'deb https://dl.winehq.org/wine-builds/ubuntu/ cosmic main'

Ubuntu 18.04 (bionic), **Linux Mint 19/19.1** y todos sus derivados:
sudo apt-add-repository 'deb https://dl.winehq.org/wine-builds/ubuntu/ bionic main'

Ubuntu 16.04 (xenial), **Linux Mint 18.x** y todos sus derivados:
sudo apt-add-repository 'deb https://dl.winehq.org/wine-builds/ubuntu/

Ubuntu 14.04 (trusty), **Linux Mint 17.x** y todos sus derivados:
sudo apt-add-repository 'deb https://dl.winehq.org/wine-builds/ubuntu/

Bien, ya tenemos los repos agregados, ahora para **instalar Wine 4.0** tan solo debes ejecutar lo siguiente:
sudo apt install --install-recommends winehq-stable

Si eres de los que les gusta probar cosas nuevas, puedes instalar la versión en desarrollo:
sudo apt install --install-recommends winehq-devel

Espero que disfrutes de tu nuevo **Wine 4.0**, no olvides compartir el articulo.

Instalar Android Studio en Ubuntu 18.04 y derivados

Instalar Android Studio en **Ubuntu 18.04**, **Linux Mint 19** y derivados.

Android Studio es un **IDE** multiplataforma que cuenta con funciones completas, que te ayudara a crear aplicaciones para los dispositivos **Android**.

Está basado en **IntelliJ IDEA** de **JetBrains,** e incluye todo lo que necesita para el desarrollo de excelentes aplicaciones.

El sistema de compilación de **Android Studio** está impulsado por **Gradle**, lo que nos va a permitir crear variantes de compilación para se uso en diferentes dispositivos, esto desde un solo proyecto.

En este articulo vemos cómo **instalar Android Studio** en Ubuntu 18.04, Linux Mint 19 y todos sus derivados.

También es valido para Ubuntu 16.04, incluyendo cualquier distribución compatible basada en Ubuntu, como Kubuntu, Linux Mint y Elementary OS.

Vemos como **instalar Android Studio**.

Instalar Android Studio en Ubuntu 18.04 y derivados

Instalar Java OpenJDK

Android Studio requiere instalar Oracle Java sobre OpenJDK en su versión 8 o superior.

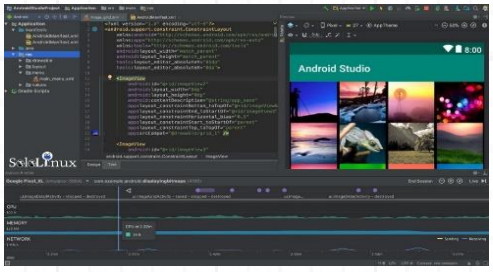
```
sudo apt update
sudo apt install openjdk-8-jdk
```

Verificamos la versión instalada.

```
java –versión
```

Ejemplo de salida valida…

openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-8u191-b12-2ubuntu0.18.04.1-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)



Instalar el IDE de Android
Instalaremos el ide usando "**snap**", si no lo tienes instalado revisa **el anterior articulo**.
sudo snap install android-studio –classic

```
sololinux:/                    # sudo snap install android-studio --classic
Download snap "android-studio" (73) from channel "stable"
Download snap "android-studio" (73) from channel "stable"     22% 2.68MB/s 4m36s
```

Al concluir la instalación te imprimirá el siguiente resucitado.
android-studio 3.3.1.0 from Snapcrafters installed
Android Studio ya está instalado.

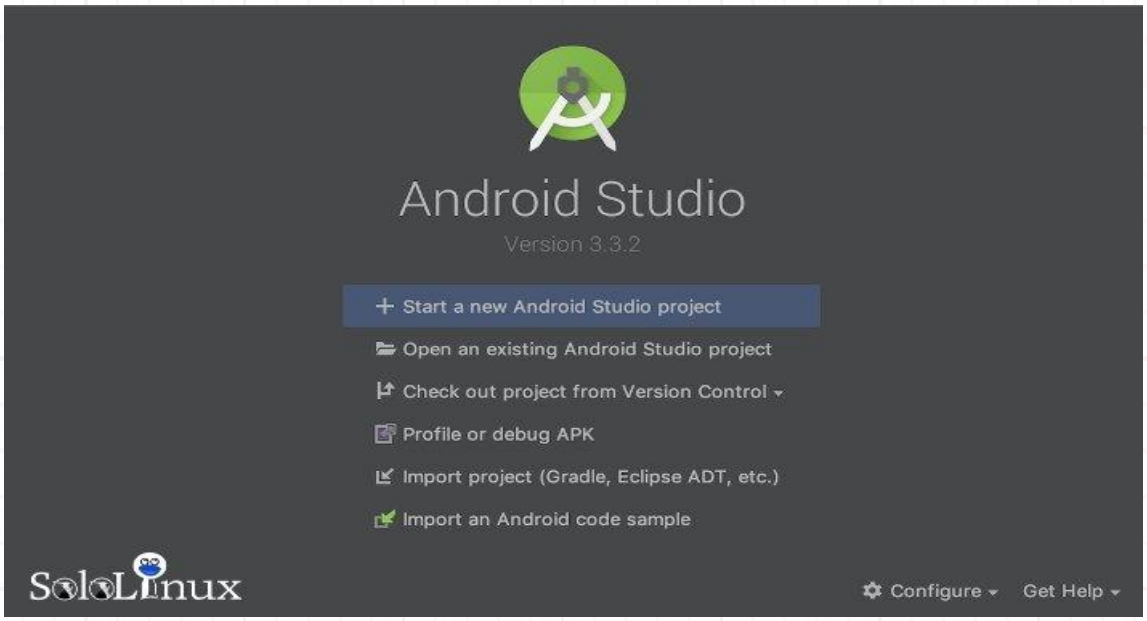
Iniciar Android Studio
Puedes iniciar **Android Studio** desde el icono de tu menú de aplicaciones, o desde la consola / terminal.
android-studio

Al iniciar por primera vez, te preguntara si quieres importar la configuración de una versión anterior de Android Studio.
Al pulsar "OK" aparece el asistente de configuración.
A continuación, nos solicita que elijamos el tipo de configuración que queremos en nuestro **Android studio**.
Lo normal es seleccionar la opción "Estándar".

En el siguiente paso, te permite seleccionar el tema de la IU, el asistente de configuración descargará e instalará automáticamente los componentes requeridos del **SDK**.

OJO!!!, la instalación de los componentes del SDK puede tomar un tiempo, se paciente.

Una vez haya terminado de instalar y cargar el **IDE Android Studio**, veras la siguiente página de bienvenida



Selecciona la opción que te interese, ya puedes empezar a trabajar.

Comparte el articulo.

THANKS!

TU PUBLICIDAD AQUI

QUIERES APARECER EN LA REVISTA, GANAR CON ELLO MAS VENTAS EN TU WEB, MAS SEGUIDORES EN TUS REDES SOCIALES...

SOLO TIENES QUE MANDAR UN CORREO A

adrian@sololinux.es

Y TE EXPLICAMOS COMO

SoloLinux

www.sololinux.es

Instalar una pila LAMP o LEMP en Linux

Instalar una **pila LAMP** o **LEMP** en Linux.

Una **pila LAMP** o **LEMP** (también conocida como "**stack**") es un paquete de aplicaciones y herramientas **open source**, que se instalan conjuntamente con el objetivo de adecuar un servidor para que pueda prestar sus servicios a sitios web dinámicos, y aplicaciones web.

En las "stack / pilas" no es necesario que instales las aplicaciones y herramientas una por una, el paquete lo hará por ti.

Existen muchas variantes, pero sin dudarlo una de las mejores alternativas es la que nos ofrece "**lempstack.com**", y no solo por la multitud de diferentes pilas que ofrece, sino por la calidad de sus **stack**.

Las pilas que se nos ofrecen son las siguientes (ellos escriben "lnmp" en vez de "lemp" por que es realmente como se escribe):

- **LNMP** (Linux + Nginx+ MySQL/MongoDB+ PHP)
- **LAMP** (Linux + Apache+ MySQL/MongoDB+ PHP)
- **LNMPA** (Linux + Nginx+ MySQL/MongoDB+ PHP+ Apache): Nginx handling the static, Apache processing dynamic PHP
- **LNMT** (Linux + Nginx+ MySQL/MongoDB+ Tomcat)
- **LNPP** (Linux + Nginx+ PostgreSQL+ PHP)
- **LAPP** (Linux + Apache+ PostgreSQL+ PHP)
- **LNMH** (Linux + Nginx+ MySQL+ HHVM)

Ademas son compatibles con multitud de SO:

- **CentOS (redhat) 6-7**
- **Debian 7-9**
- **Ubuntu 12-18**
- **Fedora 27-28**
- **Deepin 15**
- **Amazon Linux 2**
- **Aliyun Linux**

Destacamos que las versiones que ofrecen siempre están actualizadas.

Actualmente (14-03-2019) son las siguientes:

```
# Web
nginx_ver=1.14.2
tengine_ver=2.2.3
openresty_ver=1.13.6.2
apache24_ver=2.4.38
apache22_ver=2.2.34
tomcat9_ver=9.0.16
tomcat8_ver=8.5.37
tomcat7_ver=7.0.92
tomcat6_ver=6.0.53

# DB
mysql80_ver=8.0.15
mysql57_ver=5.7.25
mysql56_ver=5.6.43
mysql55_ver=5.5.62
mariadb103_ver=10.3.12
mariadb102_ver=10.2.21
mariadb101_ver=10.1.38
mariadb55_ver=5.5.63
percona80_ver=8.0.13-4
percona57_ver=5.7.24-27
percona56_ver=5.6.43-84.3
percona55_ver=5.5.62-38.14
alisql56_ver=5.6.32-9
pgsql_ver=11.1
mongodb_ver=4.0.6

# PHP
php73_ver=7.3.2
php72_ver=7.2.15
php71_ver=7.1.26
php70_ver=7.0.33
php56_ver=5.6.40
php55_ver=5.5.38
php54_ver=5.4.45
php53_ver=5.3.29

# JDK
jdk110_ver=11.0.2
jdk18_ver=1.8.0_192
jdk17_ver=1.7.0_80
jdk16_ver=1.6.0_45

# phpMyAdmin
phpmyadmin_ver=4.8.5
phpmyadmin_oldver=4.4.15.10

# Jemalloc
jemalloc_ver=5.1.0

# Pure-FTPd
pureftpd_ver=1.0.47

# Redis
redis_ver=5.0.3

# Memcached
memcached_ver=1.5.12
```

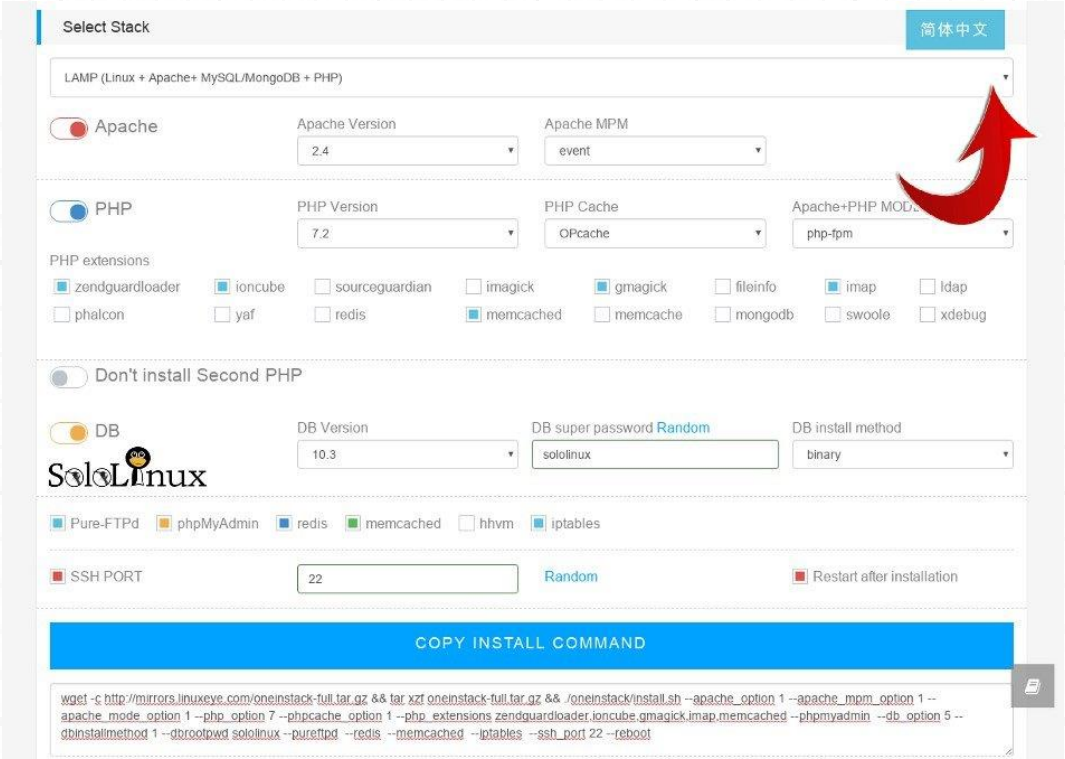
Como ves es bastante completa y actual, así que la instalamos.

Instalar una pila LAMP o LEMP

Para instalar nuestra **stack** debemos seleccionar la pila y sus componentes, lo hacemos desde su pagina oficial.

- **Selecciona tu stack**.

Marca lo que te interese.



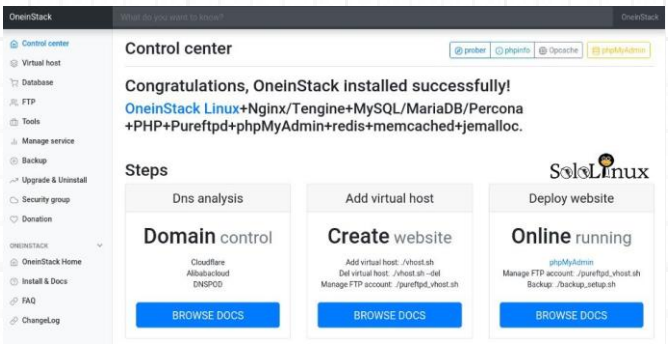
Una vez hayas creado la pila a tu gusto, copias el código que aparece en la parte inferior, lo pegas en la consola / terminal de tu **vps** o **servidor**, y lo ejecutas.

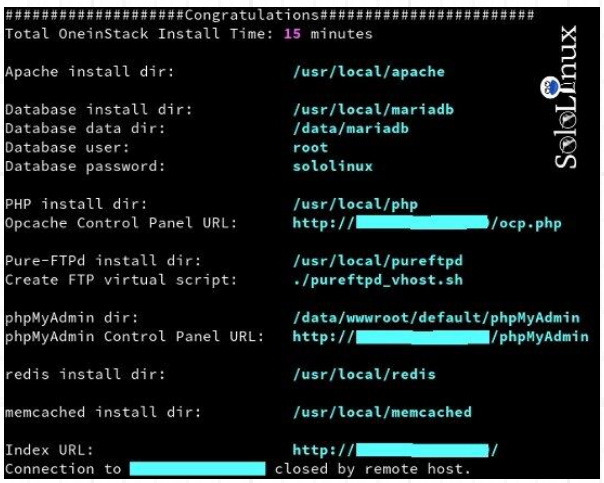
Al finalizar la instalación te aparecerán los datos de acceso y diversas localizaciones.

Para acceder y desde tu **navegador web** favorito,
escribes la ip del server.

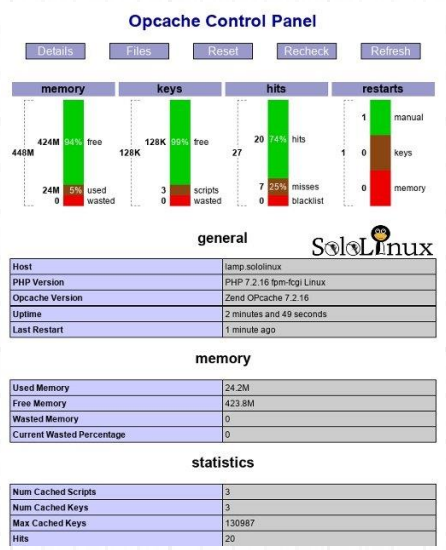
http://tu-ip/

Accedes a la pantalla de control principal.



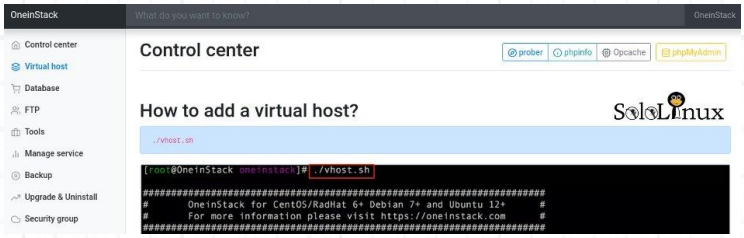


En esta pantalla es donde podrás manejar todo lo que crees en tu "**oneinstack**", los host virtuales, bases de datos, ftp's, etc...
Ademas, en la parte superior derecha tienes unos accesos directos donde podrás consultar datos e información del servidor bastante útiles.
Por ejemplo el consumo de "**opcache**"...



En cada apartado de la columna de la izquierda tienes una explicación de como operar.
Por ejemplo para crear los host virtuales debes escribir en consola:

./vhost.sh



Como punto final... un apunte.
La ejecución de los scripts de creación (en el ejemplo "**./vhost**"), debe ser desde el directorio propio de la pila.

cd oneinstack

Si te gusto este articulo, compártelo "**Instalar una pila LAMP o LEMP**".

Generador de passwords complejas en bash

Generador de passwords complejas en bash.

En este mini tutorial y siguiendo la línea de nuestros artículos sobre "**scripts bash**", vamos a ver un sencillo **script** (uno no, mejor dos) totalmente personalizable, con el que podrás **generar contraseñas complejas.**

A modo de ejemplo crearemos dos scripts:

- **Primer script:** generamos una password que se imprimirá en pantalla.
- **Segundo script:** generamos un script que desglosara la construcción de la contraseña.

Todos conocemos **bash**, así que vamos a ello.

#!/bin/bash

Generador de passwords complejas
Creamos el primer script:

nano mipass.sh

Copia y pega lo siguiente (modifica según los apuntes):

```
#!/bin/bash
#
#Password generator for Bash
#
## Puedes agregar o quitar caracteres.
MATRIX="01234!·$%56789&/ABCDEFGH?¿IJKLMNO
PQR=ÇSTUVWXYZ@#\€abcdefg~¬hijklmnopq€rstuvw
xyz"

##> Modifica 'LENGTH' si quieres cambiar el tamaño
de la pass.
LENGTH="16"

while [ "${n:=1}" -le "$LENGTH" ]
## ":=" es el operador sustituto predeterminado, si
funciona no lo deberías modificar.
do
        PASS="$PASS${MATRIX:$(($RANDOM%$
{#      MATRIX})):1}"
        let n+=1

done

echo "<------------------------------>"
echo "La password es: $PASS"
echo "<------------------------------>"

exit 0
```

Guarda el script y cierra el editor.
Lo ejecutamos: bash mipass.sh

algunos ejemplos de salida…

```
sololinux:/ # bash mipass.sh
<————————————————>
La password es: reN6S4K16dGkujqz
<————————————————>
sololinux:/ # bash mipass.sh
<————————————————>
La password es: M3wHH%xGu&O!VZ#t
<————————————————>
sololinux:/ # bash mipass.sh
<————————————————>
La password es: KiLjJHoLemPI$·?Q
<————————————————>
```

El segundo script, vendría a ser un involución del primero. Conseguiremos llegar a el mismo resultado que el anterior, pero con la diferencia de que nos imprimirá en pantalla la evolución que sigue el script para crear la **password**.
Creamos el segundo: nano mipass2.sh

Copia y pega lo siguiente (modifica según los apuntes):

```
#!/bin/bash
#
#Password generator for Bash
#
#
## Puedes agregar o quitar caracteres.
MATRIX="01234!·$%56789&/ABCDEFGH?¿IJKLMNOPQR=ÇSTUVWXYZ@#\€abcdefg~¬hijklmnopq€rstuvwxyz"

##> Modifica 'LENGTH' si quieres cambiar el tamaño de la pass.
LENGTH="16"


while [ "${n:=1}" -le "$LENGTH" ]
## “:=” es el operador sustituto predeterminado, si funciona no lo deberías modificar.
do
      PASS="$PASS${MATRIX:$(($RANDOM%${#MATRIX})):1}"

      echo "$PASS"

      let n+=1

done

exit 0
```

Guarda el script y cierra el editor.
Lo ejecutamos: bash mipass2.sh

algunos ejemplos de salida…

```
sololinux:/ # bash mipass2.sh
k
kj
kj5
kj5X
kj5Xo
kj5Xo8
kj5Xo8€
kj5Xo8€¿
kj5Xo8€¿N
kj5Xo8€¿Nk
kj5Xo8€¿NkO
kj5Xo8€¿NkO8
kj5Xo8€¿NkO8E
kj5Xo8€¿NkO8E0
kj5Xo8€¿NkO8E0x
kj5Xo8€¿NkO8E0xG
sololinux:/ # bash mipass2.sh
k
kn
knf
knfp
knfps
knfpst
knfpstx
knfpstx~
knfpstx~a
knfpstx~a#
knfpstx~a#!
knfpstx~a#!A
knfpstx~a#!A@
knfpstx~a#!A@z
knfpstx~a#!A@zI
knfpstx~a#!A@zIW
sololinux:/ #
```

Espero te sean útiles para fortalecer tus contraseñas.
Comparte el articulo. “**Generador de passwords complejas en bash**“.

Geolocalizar un servidor con bash

Geolocalizar un servidor con un **script bash**.
Puede parecer una tontería, pero **geolocalizar un servidor** diariamente te ayudara a comprobar que efectivamente se encuentra físicamente donde tu lo contrataste.
Es un buen método para asegurarnos que los **servidores** o **vps** están siempre ubicados en los puntos de acceso regionales que necesitamos.

Conozco un caso de un servidor contratado en **París**, y a los quince días apareció en **Londres**, curioso.

Es imposible que una **dirección ip** te aporte un dato super preciso, pero si que te dirán la ciudad donde esta localizada o por lo menos la región.
Para poder **geolocalizar un servidor o vps** nosotros usaremos dos API abiertas, en este caso nos las proporcionan:

- **ipinfo.c**
- **ipvigilante.com**

Recuerda que los datos recopilados no nos mostraran una **ubicación GPS** exacta, pero… por lo menos veremos el área de situación física.
Vemos como **geolocalizar un servidor**.

Geolocalizar un servidor con bash
Accedemos a la consola / terminal de nuestro servidor, e instalamos las herramientas requeridas.
➢ En Debian, Ubuntu y derivados:
sudo apt-get install curl jq

➢ En Centos, Rhel y derivados:
sudo yum install curl jq

Ahora extraemos la ip publica del servidor desde "**ipinfo**".
curl https://ipinfo.io/ip

Una vez tengas la ip publica del server, llamamos a la API de "**ipvigilante**" para obtener los datos de geolocalización (no te olvides de insertar la ip que obtuvimos con el **comando** anterior).
curl https://ipvigilante.com/la-ip-publica-del-servidor

Ejemplo de respuesta valida…
{"status":"success","data":{"ipv4":"37.187.78.186","continent_name":"Europe","country_name":"France","subdivision_1_name":null,"subdivision_2_name":null,"city_name":null,"latitude":"48.85820","longitude":"2.33870"}}
Como ejemplo, lo convertimos a código para verlo como humano (no aparece la ciudad por una restricción del servidor).

Los datos son validos, así que vamos a crear el **script bash** para automatizar el proceso.
La información se guardara como "**csv**" en el archivo "**geoip_locate.txt**" que localizaras en la carpeta "**/tmp**".
Creamos el archivo.
nano geoiplocate.sh

Copia y pega lo siguiente:

```
#!/bin/sh
OUTPUT_FILE=/tmp/geoip_locate.txt
# Capturar la ip publica del servidor
PUBLIC_IP=`curl -s https://ipinfo.io/ip`
# LLama a la API y captura la respuesta.
curl -s https://ipvigilante.com/${PUBLIC_IP} | \
    jq '.data.latitude, .data.longitude, .data.city_name, .data.country_name' | \
    while read -r LATITUDE; do
        read -r LONGITUDE
        read -r CITY
        read -r COUNTRY
        echo "${LATITUDE},${LONGITUDE},${CITY},${COUNTRY}" | \
            tr --delete \" > \
            ${OUTPUT_FILE}
    done
```

Guarda el archivo y cierra el editor.

Le concedemos permisos al script:
chmod u+x geoiplocate.sh

Lo ejecutamos manualmente y comprobamos el resultado.
./geoiplocate.sh
cat /tmp/geoip_locate.txt

Como punto final puedes crear una **tarea cron** que se ejecute el script una vez al día.
sudo cp geoiplocate.sh /etc/cron.daily

Si crees que este script es útil, compártelo.

Detectar las ip activas en tu red

Detectar las **ip activas** en tu **red**.
Con este **script bash** que te presento, podemos detectar rápidamente intrusiones en nuestra red (de forma simple).
Tan solo nos indicara las ip conectadas (**hostname** o **ip inversa** si se da el caso), pero es más que suficiente para identificar un extraño en nuestra red.
Por ejemplo, supongamos que tenemos dos dispositivos conectados a nuestra **wifi**, ejecutamos el script y nos aparece una ip adicional, ya lo tienes, toma medidas contra el intruso.

El único requisito para ejecutar el script es tener instalado "**nmap**".

```
#!/bin/bash
```

Detectar las ip activas en tu red
Si no tienes instalado nmap…
En Debian, Ubuntu, Linux Mint y derivados:
apt-get install nmap
En CentOS y derivados:
yum install nmap
En Fedora y derivados:
dnf install nmap
En OpenSuse y derivados:
zypper install nmap

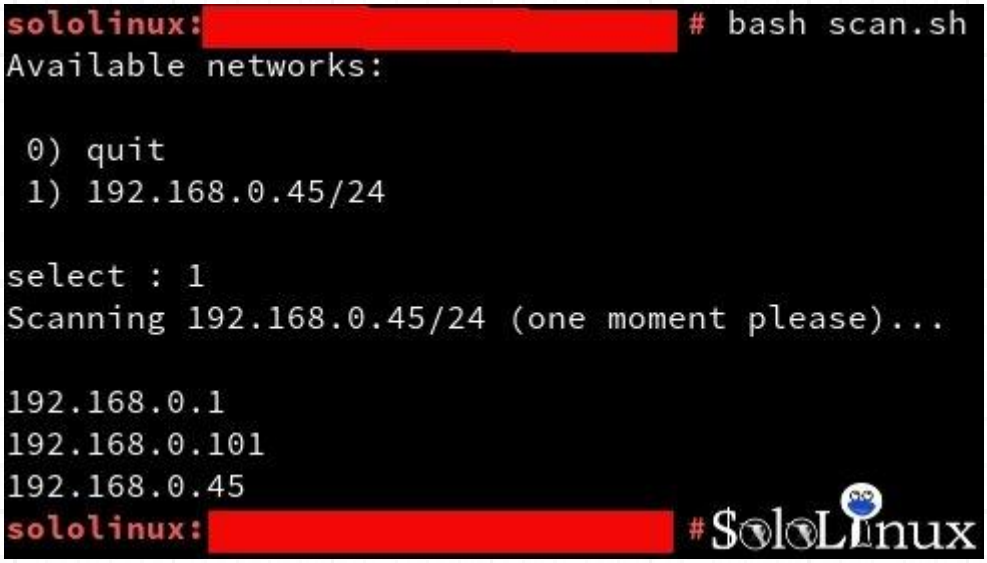
Creamos el **script bash** para detectar las ip activas en la red.
nano scan.sh

```
#!/bin/bash
#
# scan.sh
network=

# Exit if nmap not found, because regular error is ugly
x=`which nmap 1> /dev/null 2>&1`
if [ $? -eq 1 ] ; then
  echo "nmap not installed"
  exit 1
fi

# Find available networks
netlist=`ip addr show | grep inet | awk '{print $2}' | egrep -v ^'fe80|127.0|::|fd0c'`
cnt=1
echo "Available networks:" ; echo
echo " 0) quit"
for net in $netlist; do
  echo " $cnt) $net"
  cnt=`expr $cnt + 1`
done
echo

# select network to scan
netsel=
while [ X"$netsel" = X ] ; do
  echo -n "select : "
  read netsel
done

if [ $netsel -eq 0 ] ; then
  exit 1
fi
cnt=1
for net in $netlist ; do
  if [ $cnt -eq $netsel ] ; then
    network=`echo $net`
    break
  fi
  cnt=`expr $cnt + 1`
done

echo "Scanning $network (one moment please)..." ; echo
network_s=`echo $network | awk -F "." '{print $1"."$2}'`
nmap -sP ${network} | grep ${network_s} | awk -F " " '{print $5 $6}'
```

Lo ejecutamos:
bash scan.sh

Ejemplo….
Available networks:
0) quit
1) 192.168.0.45/24
select : ### 0 para salir – 1 selecciona la red

Respuesta del script:
Scanning 192.168.0.45/24 (one moment please)…
192.168.0.1
192.168.0.101
192.168.0.45
Las ip detectadas son nuestras, así que no tenemos a nadie conectado a nuestra wifi.



Comparte el articulo "**Detectar las ip activas en tu red**".

THANKS!

TU PUBLICIDAD AQUI

QUIERES APARECER EN LA REVISTA, GANAR CON ELLO MAS VENTAS EN TU WEB, MAS SEGUIDORES EN TUS REDES SOCIALES...

SOLO TIENES QUE MANDAR UN CORREO A

adrian@sololinux.es

Y TE EXPLICAMOS COMO

SoloLinux

www.sololinux.es

Generador online de .htaccess

Generador online de .htaccess (htaccess online generator).

Revisando los repositorios de Github, encontré un generador online de .htaccess que quiero compartir con los lectores de "Sololinux".

Al ver el código, me sorprendió el buen diseño y trabajo que el amigo Emirodgar había realizado.

El resultado final está muy elaborado, y cuenta con excelentes características, que pasamos a enumerar:

Opciones de acceso

- Establece el dominio principal.
- Redireccionamiento a www.
- https redirect.

Opciones de configuración

- Página de acceso predeterminada.
- Juego de caracteres predeterminado.
- Forzar la descarga de un tipo de archivo.
- Limitar el tamaño del archivo a subir.
- Configurar el correo del administrador.

Actuación

- Activar la compresión Gzip.
- Habilitar los encabezados de Keep-Alive.
- Vencimiento de los encabezados (headers).

Páginas de error personalizadas

- Error 404
- Error 500

Opciones de seguridad

- Bloquear la navegación por el directorio.
- Deshabilitar la ejecución de CGI.
- Oculta la información sensible del servidor.
- Evitar que inserten tus páginas en otros sitios web.
- Deshabilita la firma del servidor.
- Bloquea los robots de SPAM más conocidos.
- Evita el acceso ilegal o inseguro.
- No permitir la ejecución de scripts (define las extensiones).
- Deniega los métodos de solicitud que definas.
- No permite el hotlinking.

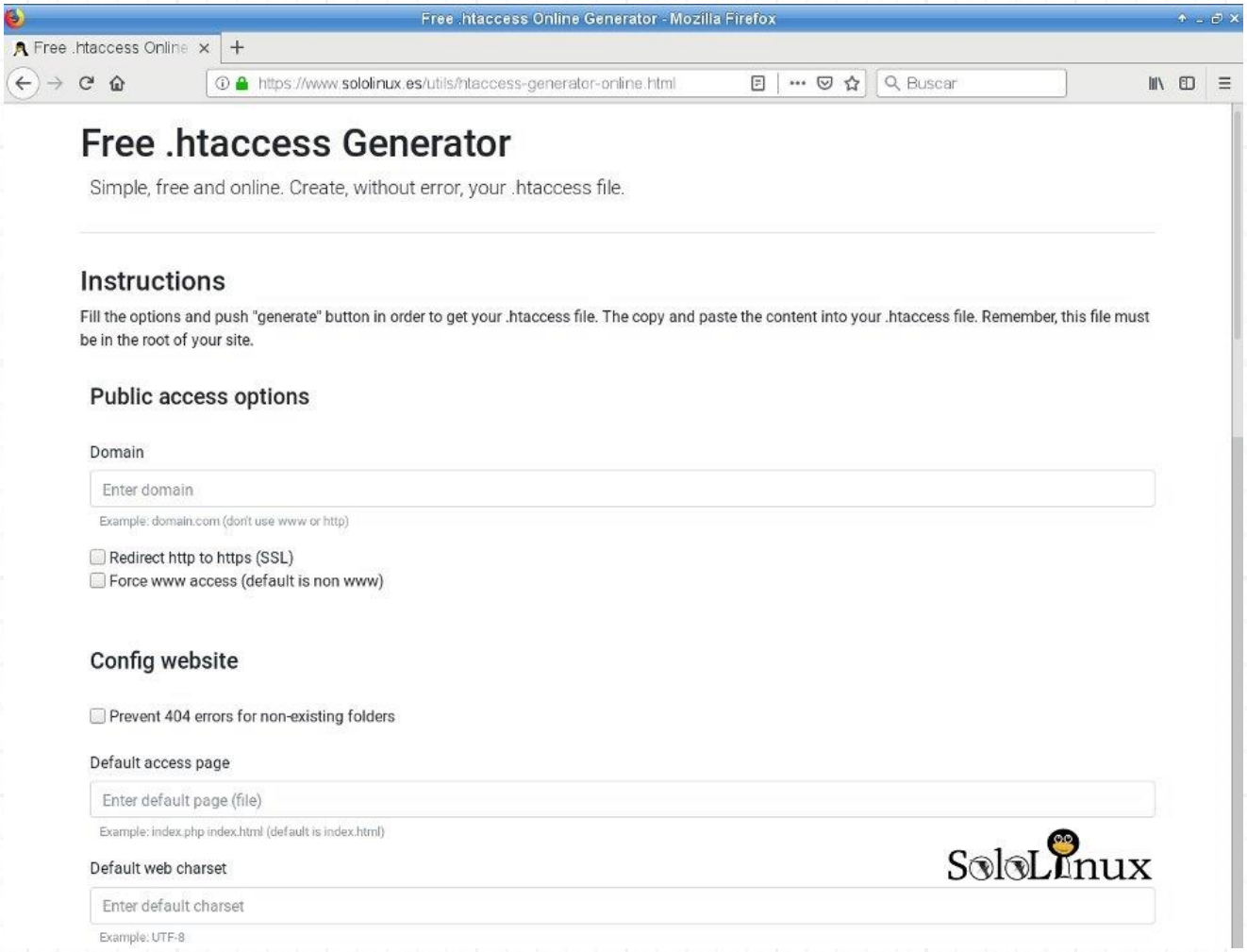
Antes de realizar cualquier modificación en tu archivo **htaccess**, te recomiendo que hagas una copia de seguridad. Es algo típico que por cualquier error en el archivo, el servidor lance un **error 500**.

Generador online de .htaccess
Como considero que es una utilidad altamente recomendable, la hemos subido a **sololinux**, para que puedas generar tu **archivo htaccess**, lo encontraras en la siguiente url.

htaccess online generator

Si prefieres tener tu propio **generador online de .htaccess**, es tan simple como crear un **archivo html** y copiar y pegar el código html que puedes encontrar en la noticia de nuestra web, accediendo en el siguiente enlace.

Guardas el *.html y lo ejecutas en tu **navegador web** favorito.
Ejemplo



Si crees que esta herramienta es útil, compártela.

Lanzan la nueva distribución Tails 3.13

Lanzan la nueva **distribución Tails 3.13**.
Acaban de lanzar la nueva versión de la **distribución Tails 3.13** (***The Amnesic Incognito Live System***), está basada en **Debian** y se presenta con muchas mejoras en el aspecto de actualizaciones y correcciones de bugs.
Tails te ayudara a:

- Navegar por internet de forma anónima y evitar censuras.
- Todas las conexiones pasan por la **red Tor**.
- No deja rastro en la maquina que estés usando, a menos que tu mismo lo solicites.
- Hace uso de herramientas criptográficas de última generación para cifrar los archivos, correos electrónicos y de mensajería instantánea.

Se ofrece en una iso live de solo 1.2Gb.

La nueva versión te garantiza una copia de seguridad de la configuración en un operador permanente en cada cambio.
Herramientas actualizadas:

- **Tor Browser 8.0.7.**
- **Tor 0.3.5.8.**
- **Thunderbird 65.1.0.**
- **Kernel 4.19.28.**

Además se ha actualizado el micro código de **Intel** a la versión *3.20180807a.2*, dado que en esta versión existen soluciones adicionales para las nuevas variantes de las vulnerabilidades **Spectre**, **Meltdown** y **L1TF**.

Si hablamos de bugs, los más destacados son:

- Evitar que software adicional descargue paquetes que ya están guardados en el almacenamiento persistente (# 15957).
- Corregir la localización de **Tor Launcher**, la aplicación que se utiliza para configurar un puente Tor o un proxy local (# 16338).
- Se repara la accesibilidad al abrir el **Navegador Tor** desde una notificación de escritorio (# 16475).
- Se soluciona el bloqueo de WhisperBack al configurar repositorios APT adicionales (# 16563).

Para más detalles, puedes ver el registro de cambios.
Las notas completas de la nueva versión las localizaras aquí.

Descargar la distribución Tails 3.13
Si quieres la iso para un "**usb**" (pendrive), la puedes descargar desde…

- ✓ **USB (pendrive)**

Si la necesitas para grabarla en un **dvd** o maquina virtual:

- ✓ **DVD o Maquina Virtual**.

Recuerda que usar cierto tipo de aplicaciones puede ser un delito, usa lo que se te ofrece para buen fin, y comparte este articulo para promocionar aun más la **live de Tails**.

Si te gusto el articulo, compártelo.

Características de Ubuntu 19.04

Características de Ubuntu 19.04.

Se aproxima la fecha del nuevo lanzamiento de **Canonical**, **"Ubuntu 19.04"**.

El lanzamiento de la versión **Ubuntu 19.04** a la cual se la denomina "**Disco Dingo**", viene con la nueva versión de escritorio **GNOME 3.32** y el **Kernel de Linux 5.0.**

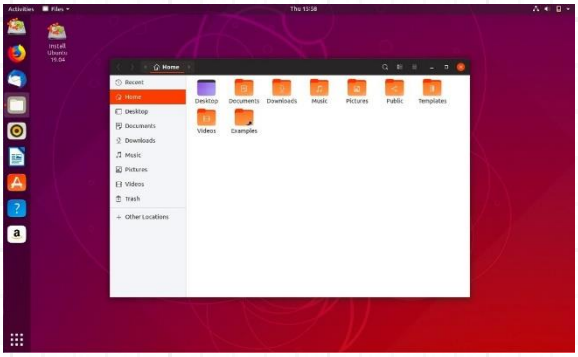
Hoy vemos las grandes mejoras que nos trae **Ubuntu**, estas son sus características.

Características de Ubuntu 19.04
Ya comentamos en este mismo articulo que '**Disco Dingo**' viene con la versión **GNOME 3.32** y el **Kernel de Linux 5.0.**
En **Ubuntu 19.04** se incluye el soporte para poder integrar **Android**, lo consiguen usando **GSConnect** que es una implementación nativa en **JavaScript** del protocolo **KDE Connect**. Esto te permitirá cómodamente conectar un teléfono Android a Ubuntu de forma inalámbrica.
GSconnect no viene instalado de manera predeterminada en Ubuntu 19.04, deberás instalarlo, por defecto trae el soporte.
No esperes grandes cambios en el diseño y banners destacados, **canonical** nos comenta que los lanzaran con las actualizaciones. Aun así observaras unos gráficos más suaves
Por fin ofrecerán **Chromium** aunque sea mediante **Snap**. También se habilita la detección de ubicación de geo-pistas (útil si viajas) y mejora considerablemente el conjunto de **iconos Suru / Yaru**.
Las **características de Ubuntu 19.04** también nos ofrecen una opción oculta. Con esta opción podrás habilitar el **soporte de escalado fraccional** en las pantallas **HiDPI**.
Son muchos los **parches de rendimiento** que aceleraran el escritorio, ya que existían quejas entre los usuarios de la **Shell** de **GNOME** en dispositivos de pantalla táctil.
Finalmente, y como cada nueva versión, veremos un nuevo fondo de pantalla de escritorio predeterminado que evidentemente incluye la mascota del la nueva versión.

Si quieres probar la nueva versión, en el siguiente articulo te lo explico.

Actualizar a Ubuntu 19.04

Actualizar a Ubuntu 19.04 (***Upgrade to Ubuntu 19.04***).
En un **articulo anterior** ya hablamos de las características de la nueva versión de **Ubuntu**, conocida como **Ubuntu 19.04 Disco Dingo**.
Si eres de los que quieres tenerla antes que nadie, puedes actualizar a **Ubuntu 19.04** ahora mismo. El único requisito es que tengas instalado y actualizado **Ubuntu 18.10 Cosmic Cuttlefish**, recuerda que no es posible desde **Ubuntu 18.04** dado que la nueva versión 19.04 no es LTS y la 18.04 si.
El proceso de actualización es simple y rápido, si sigues mi tutorial no tendrás ningún problema.
Actualizamos…



Actualizar a Ubuntu 19.04
Actualizar a Ubuntu 19.04 (***Upgrade to Ubuntu 19.04***) es un proceso rápido y sencillo, accede a tu consola /terminal y ejecuta lo siguiente para actualizar completamente tu **Ubuntu 18.10**.

```
sudo apt update
sudo apt upgrade
sudo apt dist-upgrade
```

Eliminamos los paquetes innecesarios.

```
sudo apt autoremove
```

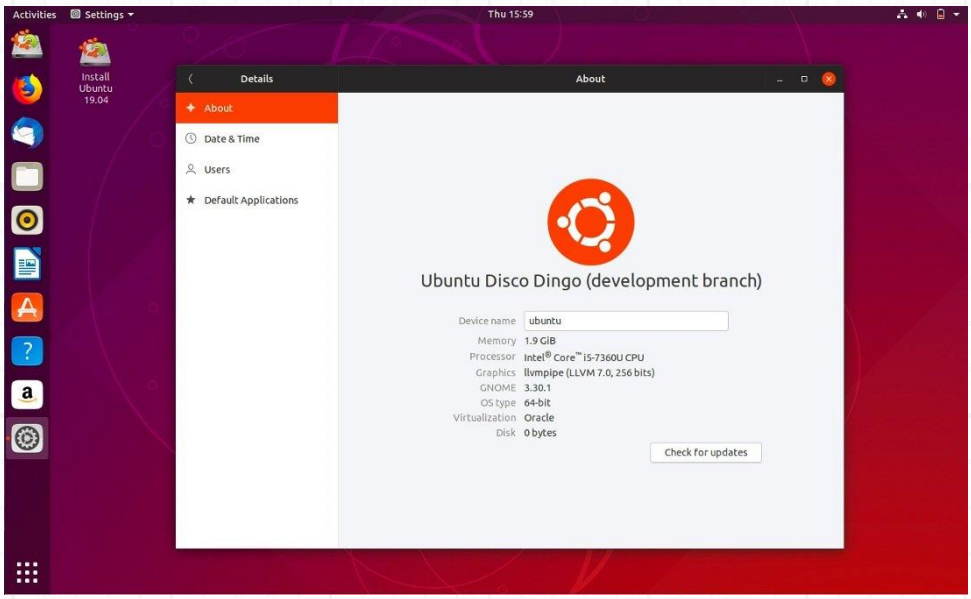
Para poder actualizar debemos configurar la variable "**Prompt**" en normal, la encontraremos en "**/etc/update-manager/release-upgrades**"

```
nano /etc/update-manager/release-upgrades
```

Ejemplo…

```
[DEFAULT]
# Default prompting behavior, valid options:
#
# never – Never check for a new release.
# normal – Check to see if a new release is available. If more than one new
# release is found, the release upgrader will attempt to upgrade to
# the release that immediately succeeds the currently-running
# release.
# lts – Check to see if a new LTS release is available. The upgrader
# will attempt to upgrade to the first LTS release available after
# the currently-running one. Note that this option should not be
# used if the currently-running release is not itself an LTS
# release, since in that case the upgrader won't be able to
# determine if a newer release is available.
Prompt=normal   #####EN NORMAL#####
```

Guarda el archivo y cierra el editor.



Actualizamos.
sudo do-release-upgrade –d

Dependiendo de tu maquina el proceso puede tomar más o menos tiempo.
Una vez concluya la actualización reinicia el sistema.
Al iniciar de nuevo… ya estarás ejecutando el nuevo “**Ubuntu 19.04**“, disfrútalo.

Comparte el articulo “**Actualizar a Ubuntu 19.04” (Upgrade to Ubuntu 19.04).**

THANKS!

TU PUBLICIDAD AQUI

QUIERES APARECER EN LA REVISTA, GANAR CON ELLO MAS VENTAS EN TU WEB, MAS SEGUIDORES EN TUS REDES SOCIALES...

SOLO TIENES QUE MANDAR UN CORREO A

adrian@sololinux.es

Y TE EXPLICAMOS COMO

SoloLinux

www.sololinux.es

GoScan: el escáner de redes interactivo

GoScan: el **escáner de redes** interactivo.

GoScan es un **escáner de redes** interactivo con autocompletado, que nos ofrece abstracción y automatización sobre **nmap**. Realiza las mismas operaciones que "**nmap**", imprimir los host, escaneo de puertos, enumeración de servicios, etc…

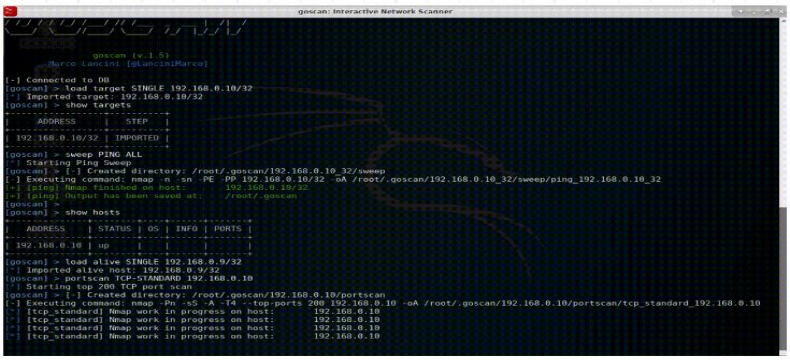
Uno de los puntos fuertes de este escáner de redes es que está especialmente indicado para usarlo en entornos inestables (fallos de conectividad), pues a medida que va realizando los escaneos programados guarda su estado en una base de datos **SQLite**.

Las exploraciones que realiza **GoScan** se ejecutan en segundo plano (separadas del subproceso principal), por lo que incluso si pierde su propia conexión retomara el trabajo. Los resultados se pueden cargar de forma asíncrona, para que me entiendas… los datos se pueden importar a **GoScan** en diferentes etapas del proceso sin que requiera reiniciar todo el proceso desde cero.

El escáner de redes GoScan está escrito en "Go".

Integra una colección de herramientas como: EyeWitness, Hydra, nikto, etc…, cada una adaptada para trabajar con un servicio específico.

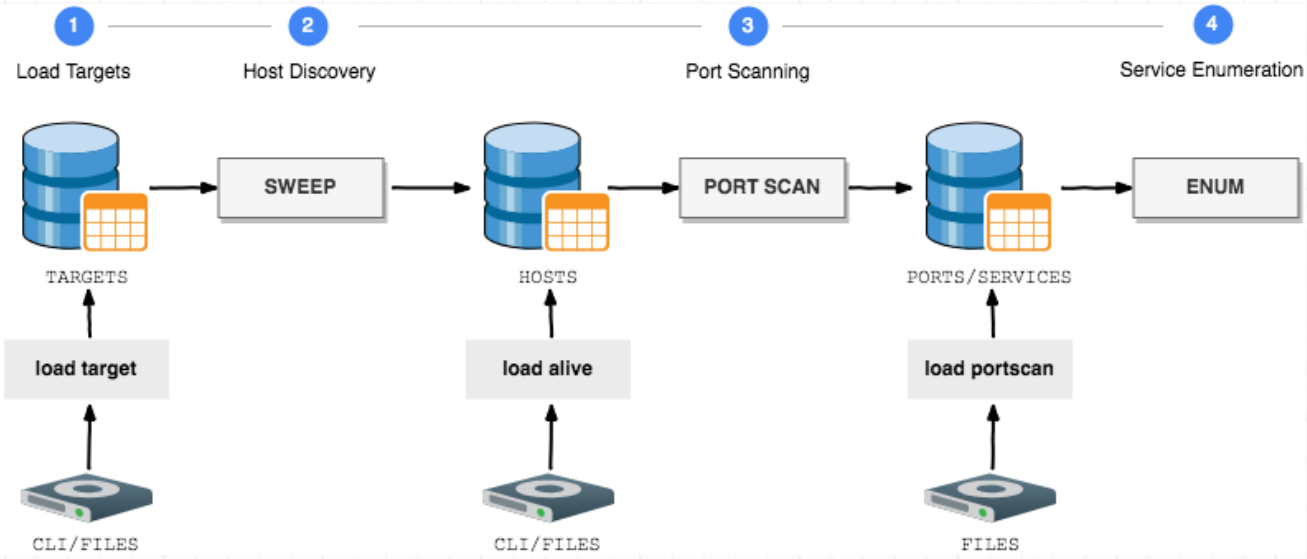
Ejemplo de uso:



Instalar el escáner de redes
En 64bits:
wget https://github.com/marco-lancini/goscan/releases/download/v2.4/goscan_2.4_linux_amd64.zip
unzip goscan_2.4_linux_amd64.zip

En 32bits:
wget https://github.com/marco-lancini/goscan/releases/download/v2.4/goscan_2.4_linux_386.zip
unzip goscan_2.4_linux_386.zip

Concedemos permisos y movemos el ejecutable.
chmod +x goscan
sudo mv ./goscan /usr/local/bin/goscan



Step	Commands
1. Load targets	•Add a single target via the CLI (must be a valid CIDR): load target SINGLE <IP/32> •Upload multiple targets from a text file or folder: load target MULTI <path-to-file>
2. Host Discovery	•Perform a Ping Sweep: sweep <TYPE> <TARGET> •Or load results from a previous discovery: • Add a single alive host via the CLI (must be a /32): load alive SINGLE <IP> • Upload multiple alive hosts from a text file or folder: load alive MULTI <path-to-file>
3. Port Scanning	•Perform a port scan: portscan <TYPE> <TARGET> •Or upload nmap results from XML files or folder: load portscan <path-to-file>
4. Service Enumeration	•Dry Run (only show commands, without performing them): enumerate <TYPE> DRY <TARGET> •Perform enumeration of detected services: enumerate <TYPE> <POLITE/AGGRESSIVE> <TARGET>
5. Special Scans	•EyeWitness • Take screenshots of websites, RDP services, and open VNC servers (KALI ONLY): special eyewitness • EyeWitness.py needs to be in the system path •Extract (Windows) domain information from enumeration data • special domain <users/hosts/servers> •DNS • Enumerate DNS (nmap, dnsrecon, dnsenum): special dns DISCOVERY <domain> • Bruteforce DNS: special dns BRUTEFORCE <domain> • Reverse Bruteforce DNS: special dns BRUTEFORCE_REVERSE <domain> <base_IP>
Utils	•Show results: show <targets/hosts/ports> •Automatically configure settings by loading a config file: set config_file <PATH> •Change the output folder (by default ~/goscan): set output_folder <PATH> •Modify the default nmap switches: set nmap_switches <SWEEP/TCP_FULL/TCP_STANDARD/TCP_VULN/UDP_STANDARD> <SWITCHES> •Modify the default wordlists: set_wordlists <FINGER_USER/FTP_USER/...> <PATH>

Integraciones de GoScan
Se admiten los siguientes servicios de integración.

WHAT	INTEGRATION
ARP	•nmap
DNS	•nmap •dnsrecon •dnsenum •host
FINGER	•nmap •finger-user-enum
FTP	•nmap •ftp-user-enum •hydra [AGGRESSIVE]
HTTP	•nmap •nikto •dirb •EyeWitness •sqlmap [AGGRESSIVE] •fimap [AGGRESSIVE]
RDP	•nmap •EyeWitness
SMB	•nmap •enum4linux •nbtscan •samrdump
SMTP	•nmap •smtp-user-enum
SNMP	•nmap •snmpcheck •onesixtyone •snmpwalk
SSH	•hydra [AGGRESSIVE]
SQL	•nmap
VNC	•EyeWitness

Una buena herramienta para realizar tus pruebas. Puedes visitar el **Githuboficial** de **Marco Lancini** el desarrollador del proyecto.

Si te gusto el articulo, compártelo.

Test de velocidad con SpeedTest-CLI

Test de velocidad con SpeedTest-CLI.
Esta herramienta no solo es útil para realizar el test de velocidad a una conexión a Internet, al ejecutarse en consola es muy valiosa para comprobar el ancho de banda real de nuestro servidor, y que no "**nos den gato por liebre**".
Normalmente cuando queremos comprobar nuestra conexión a internet, nos conectamos a **speedtest** y ejecutamos la prueba.
Es evidente que en la consola no podemos seguir ese proceso, entonces debemos usar **SpeedTest-CLI**.
Su instalación y uso es muy fácil, lo vemos.

Test de velocidad con SpeedTest-CLI
El único requisito es que tengas instalado "**wget**" y "**python**".
Instalar requisitos en:
Debian, Ubuntu, Linux Mint, y derivados.

```
apt-get update
apt-get install wget python
```

Rhel, CentOS, y derivados.

```
yum install epel-release
yum install python wget
```

Fedora y derivados.

```
dnf install python wget
```

OpenSuse, SuSe, y derivados.

```
zypper update
zypper install python wget
```

Ahora descargamos la herramienta.
wget -O speedtest-cli https://raw.githubusercontent.com/sivel/speedtest-cli/master/speedtest.py

Le damos permisos de ejecución.
chmod +x speedtest-cli

Ejecutamos (dos opciones).
/. speedtest-cli
ejecuta cualquiera de los dos.
python speedtest-cli

Ejemplos de salida…

[root@host ~]# ./speedtest-cli
Retrieving speedtest.net configuration…
Testing from ONLINE S.A.S. (IP-SERVIDOR)…
Retrieving speedtest.net server list…
Selecting best server based on ping…
Hosted by Orange (Paris) [1.88 km]: 1.968 ms
Testing download
speed……..
Download: 877.52 Mbit/s
Testing upload
speed……
……
Upload: 778.04 Mbit/s
[root@host ~]#
sololinux: # ./speedtest-cli
Retrieving speedtest.net configuration…
Testing from TENET Scientific Production Enterprise LLC (185.247.21.234)…
Retrieving speedtest.net server list…
Selecting best server based on ping…
Hosted by ISP Black Sea (Одесса) [1.20 km]: 6.97 ms
Testing download
speed……..
Download: 24.83 Mbit/s
Testing upload
speed……
……
Upload: 25.04 Mbit/s
sololinux: #

El comando dispone de opciones, si quieres que nos aporte una url de la imagen del test, ejecuta lo siguiente.
speedtest-cli –share

Ejemplo de salida…
sololinux: # speedtest-cli –share
Retrieving speedtest.net configuration…
Testing from TENET Scientific Production Enterprise LLC (185.247.21.234)…
Retrieving speedtest.net server list…
Selecting best server based on ping…
Hosted by 3ACTABA.NET (Odessa) [1.20 km]: 3.187 ms
Testing download speed……
Download: 24.54 Mbit/s
Testing upload speed……
Upload: 23.93 Mbit/s
Share results: http://www.speedtest.net/result/8149440956.png

Puedes visualizar todas las opciones ejecutando "-h".
./speedtest-cli –h

usage: speedtest-cli [-h] [--no-download] [--no-upload] [--single] [--bytes]
[--share] [--simple] [--csv]
[--csv-delimiter CSV_DELIMITER] [--csv-header] [--json]
[--list] [--server SERVER] [--exclude EXCLUDE]
[--mini MINI] [--source SOURCE] [--timeout TIMEOUT]
[--secure] [--no-pre-allocate] [--version]

Comparte el "**Test de velocidad con SpeedTest-CLI**".

NUESTROS NUMEROS DE UN VISTAZO

No 1 FEBRERO 2019

No 2 MARZO 2019

Visítanos en www.sololinux.es

Revista de distribucion gratuita, comparte conocimientos.

MAGAZINE

SOLOLINUX