

Web hosting

Instalando un servidor de web hosting

- Lo primero que necesitamos hacer es instalar el servidor apache2. Para ello utilizamos los siguientes comandos:

```
sudo apt update && sudo apt upgrade  
sudo apt install apache2
```

- Una vez realizado esto podemos verificar que el puerto 80 se encuentre aceptando peticiones ya sea utilizando un browser, el comando curl o cualquier otro medio. Si lo hacemos desde el navegador visualizaremos la página por defecto de apache.

A continuación podemos determinar la estrategia de implementación del servidor. Podemos seleccionar fork, event o worker. En este caso seleccionamos worker, para ello ejecutamos:

```
a2enmod mpm_worker #para fork mpm_prefork, y para event mpm_event
```

Toda la configuración de apache queda en el directorio **/etc/apache2**

- Con el archivo **/etc/apache2/ports.conf** podemos configurar los puertos y direcciones IP en los que apache recibirá solicitudes. Por defecto escuchará los puertos 80 y 443 (SSL).

Alojando sitios en nuestro servidor web

Podemos adoptar multiples técnicas, en nuestro caso crearemos un directorio en el home de nuestro usuario llamada **web-server**. Dentro de ella creamos dos directorios más:

- **sites**: En este directorio alojaremos los sitios o aplicaciones web. Esto no se limita a sitios o apps, también puede tratarse de un directorio.
- **conf**: En este directorio se almacenarán los archivos de configuración de los respectivos sitios o aplicaciones.

En el directorio **/etc/apache2/sites-available** crearemos un archivo de configuración principal que contendrá lo siguiente:

```
NameVirtualHost *:80  
  
Include [path]/web-server/conf/*.conf
```

A para efectivamente almacenar los diferentes sitios deberemos crear un directorio dentro de **sites** con el contenido que se servirá del sitio y, a la vez, un archivo de configuración para cada sitio que se alojará en **conf**. En nuestro caso se utiliza la siguiente plantilla:

```
<Directory "[path]/web-server/sites/[your_site_dir]">  
  Options Indexes FollowSymLinks  
  AllowOverride none  
  Require all granted  
</Directory>  
  
<VirtualHost *:80>
```

```

ServerAdmin sl2018@localhost
ServerName [your_site_name eg: www.slwebhosting.com] #Este nombre se utilizará para determinar que sitio servir

#Root del sitio alojado
DocumentRoot [path]/web-server/sites/[your_site_dir]

#Directorio del log de errores
ErrorLog ${APACHE_LOG_DIR}/[your_site_name eg: www.slwebhosting.com]-error.log

#Nivel al cual deseamos efectuar el logging
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/[your_site_name eg: www.slwebhosting.com]-access.log combined

# Disables all caching (use for development)
<FilesMatch "\.(html|htm|js|css|json)$">
    FileETag None
    <IfModule mod_headers.c>
        Header unset ETag
        Header set Cache-Control "max-age=0, no-cache, no-store, must-revalidate"
        Header set Pragma "no-cache"
        Header set Note "CACHING IS DISABLED ON LOCALHOST"
        Header set Expires "Web, 11 Jan 1984 05:00:00 GMT"
    </IfModule>
</FilesMatch>

</VirtualHost>

```

Para que los cambios realizados tengan efecto, debemos desmontar y montar nuevamente el archivo de configuración principal. En nuestro caso lo logramos de la siguiente forma.

```

a2disssite slwebhosting.conf
a2ensite slwebhosting.conf
service apache2 reload

```

Nota: Como para este ejercicio no disponemos de un dominio DNS debemos alterar el archivo **/etc/hosts** para que la resolución de nombres apunte al servidor que hayamos levantado. Por ejemplo si tenemos el servidor montado en la IP 192.168.0.239, y estamos alojando los sitios www.pepe1.com y www.pepe2.com, deberemos agregar una línea que diga ****192.168.0.239 www.pepe1.com www.pepe2.com****.

Creando un servidor de correo electronico

Instalaremos un servidor de correo electronico utilizando el **servidor smtp postfix** y el servidor de imap/pop3 **dovecot**. A continuación se explica su instalación y configuración.

- En primer lugar es aconsejable actualizar el sistema para ello utilizamos los comandos **apt update** y **apt upgrade**.
- A continuación modificaremos el archivo **/etc/host.conf** de modo de modificar la resolución DNS, el mismo deberá quedar como se indica a continuación:

```

order hosts,bind
multi on

```

- Esto es opcional pero aconsejable, podemos cambiar el nombre de nuestro host a través del archivo **/etc/hosts**. Para ello utilizamos los siguientes comandos:

```
hostnamectl set-hostname mail.slwebhosting.com
echo "192.168.0.239 slwebhosting.com mail.slwebhosting.com" >> /etc/hosts
init 6
```

Luego del reinicio, nuestra línea de comando se debería ver como [user]@[host_name], siendo host_name el hostname seleccionado anteriormente. Del mismo modo deberíamos ser capaces de hacer ping a nuestro host, por ejemplo:

```
ping slwebhosting.com
ping mail.slwebhosting.com
```

- Ahora estamos en condiciones de instalar **postfix** el cual será nuestro servidor SMTP es decir, aquel encargado de enviar los mails salientes. Para ello utilizamos el comando **apt install postfix**. Durante la instalación es posible que se nos hagan algunas preguntas acerca de la configuración deseada. Si es así seleccionar la opción **internet site** y luego insertar el nombre de dominio.
- Finalizada la instalación deberemos configurar postfix. Deberemos aplicar la configuración al archivo **/etc/postfix/main.cf**, es aconsejable hacer un backup del archivo y luego aplicar nuestro archivo. En este caso se adjunta el archivo de configuración utilizado. Cabe destacar que debemos reemplazar la asignación de las variables **myhostname**, **mydomain** y **mynetworks** para que representen nuestra configuración. Finalmente, podemos utilizar el comando **postconf -n** para verificar eventuales errores de configuración.
- A continuación, reiniciamos el servicio de postfix y verificamos tanto su estado como si esta escuchando el puerto 25. Esto lo realizamos mediante el comando **netstat -tln**.
- Una vez instalado el servidor SMTP, debemos instalar el servidor IMAP, es decir, aquel que actuará como la casilla de correos. En este caso utilizaremos **dovecot**. Para instalarlo ejecutamos el siguiente comando: "apt install dovecot-core dovecot-imapd".
- Ahora deberemos configurarlo para ello realizaremos una serie de modificaciones a los siguientes archivos:

```
# --> /etc/dovecot/dovecot.conf (descomentar)
listen = *, ::

# --> /etc/dovecot/conf.d/10-auth.conf (editar las siguientes lineas, descomentar si estan coment
adas)
disable_plaintext = no
auth_mechanisms = plain login

# --> /etc/dovecot/conf.d/10-mail.conf (append -- especifica la ubicación del directorio de mail
de cada usuario)
mail_location = maildir:~/Maildir

# --> /etc/dovecot/conf.d/10-master.conf (modificamos el bloque smtp-auth)
unox_listener = /var/spool/postfix/private/auth {
    mode = 0666
    user = postfix
    group = postfix
```

```
}
```

- Finalmente reiniciamos el servidor y del mismo modo que como hicimos con el servidor de SMTP, verificamos su estado.

Opcional: Instalar webmail

Aprovechando que ya tenemos configurado el servidor apache, podemos agregar un dominio dentro de nuestro servidor web para que los usuarios puedan loguear y ver sus mails.

Para ello seguimos el procedimiento antes indicado para agregar un sitio a nuestro servidor web. A continuación instalamos los siguientes paquetes:

```
apt install php7.0 libapache2-mod-php7.0 php7.0-curl php7.0-xml
cd [path-to-apache-webmail-dir]
wget http://repository.rainloop.net/v1/rainloop-1.6.3.715-f96ed936916b7f3d9039819323c591b9.zip
unzip rainloop-1.6.3.715-f96ed936916b7f3d9039819323c591b9.zip
rm rainloop-*.zip
```

Ahora reiniciamos apache, y podremos acceder al administrador de webmail desde el browser. En primera instancia accederemos con **[url]?admin**, con lo cual accederemos al panel de administración. El usuario por defecto es admin - 12345. Una vez dentro agregaremos nuestro dominio, para ello nos dirigiremos al menu correspondiente en la sidebar. Allí completaremos los datos requeridos, lo mas importante es especificar en el en el servidor SMTP e IMAP nuestro dominio.

Una vez realizado esto accedemos normalmente al servicio de mail por la url designada y tendremos una interfaz para administrar nuestra casilla.

Instalando un servidor de bases de datos

Instalaremos un servidor de **my-sql**. Para ello comenzaremos instalando el servidor con el comando **apt install mysql-server**.

El server quedará instalado, si no nos lo solicita durante la instalación podemos administrar el server para agregar usuarios y/o cambiar la contraseña con el comando:

```
mysqladmin -u <user> -p password newpass
```

Ahora habilitaremos el acceso remoto. En primer lugar debemos modificar el archivo de configuración para poder establecer un acceso externo. En este caso debemos editar el archivo de configuración **/etc/mysql/mariadb.conf.d/50-server.cnf** simplemente comentando la linea que dice **bind-address = 127.0.0.1**.

Finalmente se debe dar privilegios de acceso al usuario externo, para ello podemos acceder a la consola de mysql y ejecutar los siguientes comandos.

```
mysql #accedemos a la consola
GRANT ALL PRIVILEGES ON [db].[table] TO '[user]'@[host] IDENTIFIED BY '[password]' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Instalando un servidor de archivos

En primer lugar debemos crear un grupo y un usuario para este grupo. Para ello utilizamos los siguientes comandos:

```
groupadd sftp
useradd -c [user_description] -g sftp -m -s /bin/nologin [user_name]
```

Muy importante: El hecho de especificar como shell no login, implica que el usuario no tendrá acceso al sistema, simplemente se podrá conectar al servidor de archivos.

Ahora creamos un directorio en la carpeta **/home** que llamaremos **sfpt**. Este directorio lo crearemos con el usuario root, por ende los usuario no tendrán permisos sobre este directorio. Aquí dentro creamos otro directorio llamado **usuarios_sftp**. Finalmente asignamos como dueño del directorio al usuario que creamos antes para el grupo que creamos antes y le damos **permisos 755**

Ahora debemos configurar el servidor ssh. En primer lugar buscamos el servidor sfpt por medio del comando **find / -name sftp-server**. Debemos recordar la ubicación de dicho archivo. A continuación editamos el archivo **/etc/ssh/sshd_config**. Modificamos la línea que dice **Subsystem sftp...** colocando el path que buscamos anteriormente y al final del archivo agregamos:

```
Match group sftp
ChrootDirectory /home/sftp
ForceCommand internal-sftp
```

Finalmente para conectarnos desde un host externo, utilizamos el comando (usualmente el puerto es el 22, a no ser que se haya cambiado explícitamente la configuración):

```
sftp -P [PORT] [USER]@[HOST_IP]
```

Administración de usuarios

Para administrar los usuarios se desarrollaron 5 scripts, cada uno con una función particular.

Creación de usuarios

Este script (**createUser.sh**) en primera instancia verifica que no exista un usuario con el nombre indicado. Si esto es así, realiza la siguiente tarea:

- **Crear el usuario en el sistema y agregarlo al grupo slwebhosting.** Este usuario no contará con login al sistema.
- **Crear el archivo de configuración del dominio virtual.** Se crea el archivo de configuración y se lo agrega en el directorio de archivos de configuración. Para el mismo se utiliza la plantilla que se mostro para la configuración del servidor web. El dominio del usuario sera [\[username\].slwebhosting.com](#)
- **Crear el directorio de dominio virtual en el directorio SFTP.** Se crea un directorio con permisos de escritura unicamente para el usuario creado y de lectura/ejecución para todos. El mismo se ubica como se dijo antes en el directorio raíz de SFTP y es apuntado por el archivo de configuración de dominio virtual.
- **Crear la cuenta de email.** Simplemente se crea el directorio Maildir/ en el directorio home del usuario inicado y se le otorgan los permisos correspondientes.
- **Crear usuario mysql:** Se crea el usuario en la tabla de usuarios del servidor MariaDB, a continuación se crea la base de datos para el usuario la cual será nombrada por su user_name y

se le otorgan todos los privilegios sobre la misma. El usuario se crea como 'username'@'%' para dar independencia para el logueo remoto.

- **Generar archivo de configuración y enviarlos al usuario.** Se crea un archivo con todos los datos e instrucciones necesarias para la utilización de los servicios y se envía a la casilla de correo recientemente creada para el usuario.

- **Reinicio de todos los servicios.** Se reinician todos los servicios para que los cambios tengan efecto (mejorable).

Listar usuarios

Este script (**listUsers.sh**) simplemente se aprovecha que todos los usuarios del servidor de webhosting están en el mismo grupo. Se utiliza el comando **groupmems -g \$WEBHOSTING_GROUP -l**.

Borrado de usuarios

Se plantearon dos alternativas: deshabilitar (**disableUser.sh** y **enableUser**) y borrar completamente (**purgeUser.sh**).

En el primer caso se sigue el siguiente procedimiento:

disableUser:

- Se utiliza el comando **passwd -l** para deshabilitar el login del usuario.
- Se quitan absolutamente todos los privilegios al directorio del dominio del usuario (**chmod 000**).
- Deshabilitamos el login al servidor de mysql, para ellos se edita la entrada del usuario en la tabla de usuarios, reemplazando el campo host '%' -> '!'.
- Reiniciamos todos los servicios (mejorable).

enableUser:

- Se utiliza el comando **passwd -u** para habilitar nuevamente el login del usuario.
- Se otorga nuevamente permisos al directorio del dominio del usuario (**chmod 755**).
- Se vuelve a habilitar el login al usuario mediante la modificación del campo host en la tabla de usuarios de mysql '!' -> '%'
- Reiniciamos todos los servicios (mejorable).

En el segundo caso se sigue el siguiente procedimiento:

purgeUser:

- Se elimina el archivo de configuración de virtual host correspondiente al usuario.
- Se elimina el directorio de dominio del usuario dentro del directorio SFTP.
- Se elimina el directorio home del usuario.
- Se elimina la base de datos del usuario y posteriormente el usuario.
- Finalmente eliminamos el usuario del sistema.
- Reiniciamos todos los servicios (mejorable).