

Proyecto Final ASO

Nicolás A. Ortega Froysa

9 de febrero de 2023

Índice

1. Resumen de Topología	2
2. Configuración Inicial Windows Server	3
3. Configuración Inicial Linux	3
3.1. Instalación	4
3.2. Post-Instalación	4
4. Active Directory	7
4.1. Instalación	7
4.2. Configuración en Linux	10
5. Terminal de Administración	11
6. Base de Datos	11
7. Servidor Web	13
7.1. Nginx	13
7.2. Certificado SSL	15
7.3. MediaWiki	17
8. Derechos de Autor y Licencia	19

Índice de figuras

1. Mapa de intrenet.	2
2. Selección de <i>software</i> adicional.	5
3. Proceso instalación de Active Directory.	8
4. Active Directory instalado.	9
5. Cambiar contraseña del usuario «Administrator».	9
6. Habilitando la opción de crear una carpeta home el entrar. . .	11
7. Página de Nginx por defecto.	14
8. Instalación de MediaWiki.	18
9. Página inicial de nuestro Wiki.	19

Índice de cuadros

1. Tabla de direcciones IP.	3
2. Tabla de particiones lógicas en los sistemas Debian.	4

1. Resumen de Topología

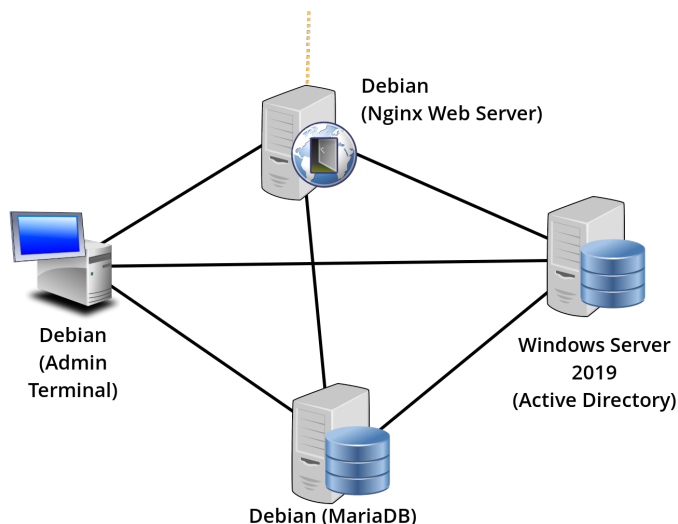


Figura 1: Mapa de intrenet.

El objetivo será montar una red interna que se asemeja a lo visto en la figura 1. Se ha decidido montar el servidor LDAP con Windows Server 2019, ya que ya tuvimos una máquina de este tipo, y nuestro superior aconsejaba el uso de Active Directory para servir de LDAP. Todo lo demás hemos decidido usar Debian, ya que es una distribución estable, y así evitamos que las actualizaciones rompan nuestras máquinas. También tiene una comunidad muy grande para proveer soporte, aunque éste no sea de carácter profesional. Para servidor web hemos decidido por Nginx por familiaridad y también por facilidad de uso. Para nuestro servidor de bases de datos se ha elegido a MariaDB, también por familiaridad, pero también por facilidad de instalación en Debian respecto a otros más comerciales como Oracle. Finalmente nuestro terminal administrador, que también corre Debian, servirá para conectarse y administrar los demás dispositivos por medio de SSH, en el caso de los servidores Debian, o VNC en el caso del Windows Server.

En cuanto a la identificación de cada dispositivo, usaremos las direcciones IP y los nombres de *host* que aparecen en la tabla 1.

Para nuestra red, usaremos el dominio de **nortega.local**. Todos nuestros dispositivos tendrán un nombre por debajo de este dominio de acuerdo a su nombre de *host*.

A todos nuestros dispositivos tendremos al menos una tarjeta de red para conectarse a la red local «ASO». Esto se hará en forma de una Red NAT.

Nombre	Servicio	Dirección IP
ad-serv	Active Directory	10.0.2.50
web0	Servidor Web	10.0.2.5
db-serv	Base de Datos	10.0.2.10
admin	Administración	10.0.2.2

Cuadro 1: Tabla de direcciones IP.

El servidor web (**web0**) en particular tendrá otra tarjeta red configurado como Adaptador Puente Promiscuo, que habilitará la conexión a él desde el exterior.

2. Configuración Inicial Windows Server

El servidor Windows necesita más recursos, ya que consume más por causa de todo lo que tiene instalado por defecto – no es una instalación mínima, por así decirlo. Así que asignaremos a él las especificaciones siguientes:

- Procesadores: 2
- Memoria: 4096 MB
- Disco: 50 GB

Instalaremos Windows Server utilizando las opciones por defecto. No hay mucho misterio. Es darle para adelante en cada paso.

3. Configuración Inicial Linux

Nuestros servidores Debian no ocupan mucho espacio en sí, como lo hace Windows Server, por eso nos podemos permitir reducir los recursos alojados a éstos. A estas máquinas alojaremos los recursos siguientes a todos:

- Procesadores: 1
- Memoria: 1024 MB
- Disco: 10 GB

3.1. Instalación

En la instalación de Debian, queremos modificar varias cosas de forma particular. Primero, el dominio que queremos configurar es, como se mencionó anteriormente, `nortega.local`. También, a la hora de especificar un *hostname*, usamos el especificado anteriormente para cada servidor. Esto se hará en la configuración de red.

Cuando llega la hora de escribir las particiones al disco, vamos a usar una tabla de LVM para permitir más flexibilidad en cuanto a modificación del tamaño de las distintas particiones, y sobre todo poder ampliar su tamaño con facilidad, y poder hacerlo *en caliente*. Asignaremos todo el disco a un mismo grupo, y crearemos las particiones de acuerdo a la tabla 2.

Mountpoint	Tamaño
/	1.9 GB
/boot	451 MB
/tmp	463 MB
/usr	3.7 GB
/var	2.8 GB
Swap	496 MB

Cuadro 2: Tabla de particiones lógicas en los sistemas Debian.

En cuanto a los usuarios, crearemos un usuario *proxy*; es decir, sólo sirve para luego pasarte al usuario `root`. En el caso este, se ha creado el usuario `nicolas` con contraseña `nicolas` – en realidad poner otra contraseña más compleja. Para el usuario `root` le asignaremos la contraseña `toor`.

Cuando nos pregunta si hay otros paquetes que queremos instalar queremos especificar los «standard system utilities» y también el «SSH server». No instalamos «web server» porque esto instalará Apache, que no nos interesa. Debería de aparecer como en la figura 2.

El resto de la instalación se realizará de forma normal, eligiendo las opciones correspondientes.

3.2. Post-Instalación

Una vez instalado, debemos de configurar nuestros servidores Debian en cuestión de configuración red, sobre todo nuestro servidor DNS – que será el servidor de Windows con Active Directory – y también se configurará la dirección IP de forma estática.

Primero, para configurar las direcciones IP estáticas. Para esto, debemos de deshabilitar el servicio de NetworkManager que lo tiene Debian por defecto:

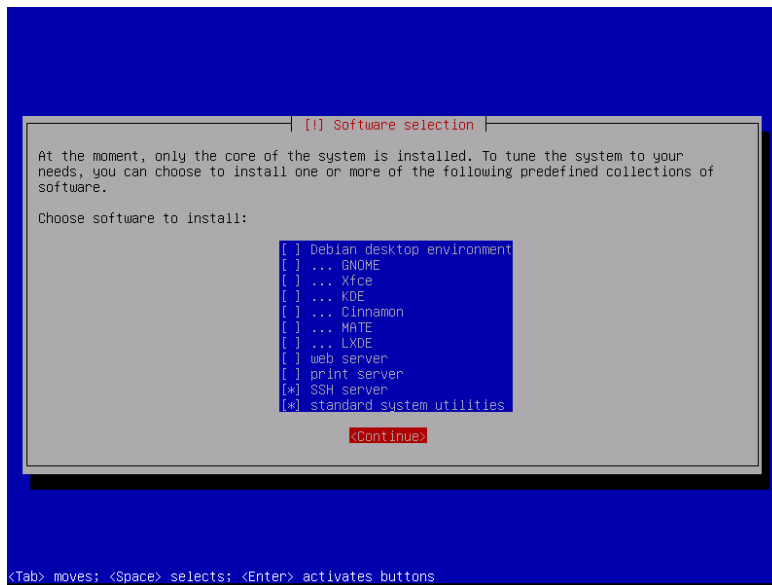


Figura 2: Selección de *software* adicional.

```
# systemctl stop network-manager.service
# systemctl disable network-manager.service
```

Una vez deshabilitado el NetworkManager, ya podemos empezar a modificar los archivos de configuración de red. El primer paso será definir una dirección IP estática. Para empezar, encontramos cuál es el nombre de nuestro interfaz, que se puede hacer utilizando el comando `ip a`.

```
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:31:09:eb brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.2/24 brd 172.16.40.255 scope global dynamic
enp0s3
        valid_lft 595981sec preferred_lft 595981sec
    inet6 fe80::a00:27ff:fe76:1cdf/64 scope link
        valid_lft forever preferred_lft forever
```


Vemos en este ejemplo que queremos hacer uso de la interfaz `enp0s3`, que es la interfaz red de *ethernet*. Luego, en el archivo `/etc/network/interfaces`, vamos a configurarlo de la forma siguiente asumiendo aquel interfaz y que queremos la dirección IP estática 10.0.2.2:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 10.0.2.2
    netmask 255.255.255.0
```

También queremos configurar nuestro servidor de DNS, que será el Windows Server 2019 con Active Directory. Para esto hemos de modificar el archivo `/etc/resolv.conf` con el contenido siguiente:

```
domain nortega.local
search nortega.local
nameserver 10.0.2.50
```

Hemos de evitar que este archivo se sobrescriba, ya que hay muchos servicios de Systemd que buscarán hacer esto precisamente. Así que lo suyo es deshabilitar estos servicios directamente:

```
# systemctl disable systemd-resolved.service
# systemctl stop systemd-resolved.service
```

Ya cuando hayamos configurado esto, podemos reiniciar el servicio de *networking* y veremos como hace uso de la configuración especificada. Esto lo hacemos del modo siguiente:

```
# systemctl restart networking.service
# ifup enp0s3
```


Luego también queremos configurar, particularmente en el servidor de web que estará conectado a *internet*, y será el más vulnerable a ataques. Para esto, hemos de instalar y configurar el programa Firewallld. Usaremos este *firewall* porque es uno de los más conocidos, y es usado (y mantenido) por la empresa Red Hat. Para esto corremos el comando siguiente para instalar:

```
# apt install firewallld
```

Ya cuando se instala, estará activado e iniciado, y de hecho vendrá ya con excepciones para SSH y DHCPv6-client (en la zona *public*). Como en nuestra máquina de servidor web queremos tener abiertos los puertos para acceder a HTTP y HTTPS, lo activaremos con el comando siguiente:

```
# firewall-cmd --permanent --zone=public --add-service=http
success
# firewall-cmd --permanent --zone=public --add-service=https
success
# firewall-cmd --reload
success
```

Al hacer esto, ya estará disponible nuestro *firewall* para prohibir el acceso a otro tipo de conexiones que no sean HTTP o HTTPS.

4. Active Directory

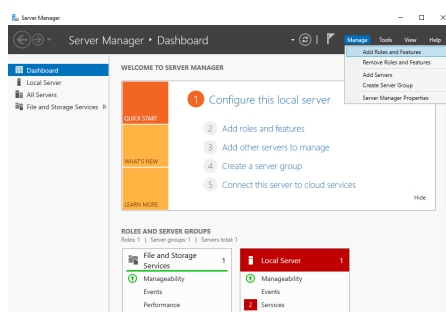
En la máquina Windows Server 2019 montaremos el servicio de Active Directory. Este proceso lo podemos dividir en dos partes distintas: la instalación de Active Directory en el servidor de Windows, y su configuración en los demás servidores Linux.

4.1. Instalación

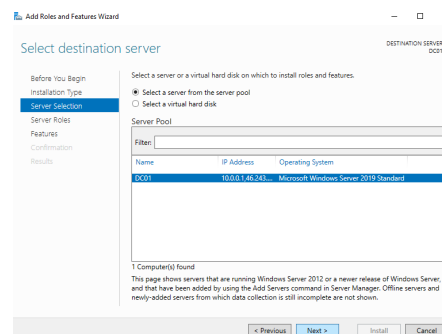
Cuando lanzamos el servidor Windows Server 2019, lo primero que nos abrirá será el panel de control de nuestro servidor. Aquí, si nos vamos al apartado «Manage», queremos dar a «Add Roles and Features» para poder añadir un nuevo servicio a este (figura 3a). Esto abrirá una nueva interfaz donde podremos comenzar el proceso de instalación.

Lo primero que nos preguntará será el tipo de instalación, dándonos dos opciones:

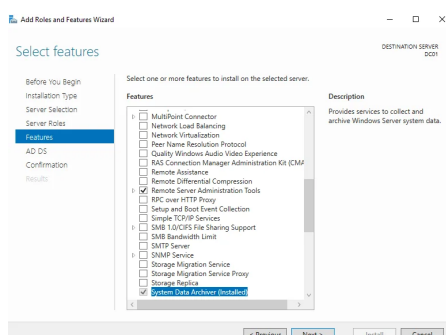
- *Role-Based or feature-based installation*: configuración de un único servidor en donde instalamos servicios.



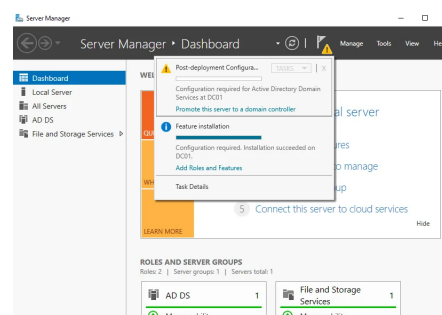
(a) Añadir un nuevo servicio.



(b) Selección del servidor.



(c) Seleccionar características del servidor.



(d) Promocionar nuestro servidor a controlador.

Figura 3: Proceso instalación de Active Directory.

- *Remote Desktop Services installation*: instalación de requisitos para terminales virtuales.

En nuestro caso, nos interesa el primero, «Role-based or feature-based installation», ya que queremos instalar el Active Directory en el mismo servidor, no en una máquina virtualizada encima. Entonces en la página siguiente nos preguntará en qué servidor queremos hacer esta instalación, mostrándonos una lista de servidores Windows en nuestra red (que nosotros tan sólo tenemos uno). Escogemos el nuestro y seguimos (figura 3b).

Próximamente nos pedirá qué características queremos para nuestro servicio. Nos interesa escoger «Active Directory Domain Services» (figura 3c). En la pantalla siguiente veremos una lista de cajas de opciones, pero aquí no hace falta hacer nada. Y ya podremos seguir con el proceso que nos instalará este servicio.

Una vez instalado, debemos de *promocionar* nuestro servidor a ser el controlador de dominio. Esto se puede hacer accediendo a la banderita que está arriba a la derecha y eligiendo «Promote this server to a domain contro-

ller» (figura 3d). Esto nos abrirá un diálogo para configurar el controlador. Nos interesa crear un bosque nuevo y ponerle de nombre de dominio raíz **nortega.local**. Para el nombre de NetBIOS usaremos también este nombre, menos la TLD de **.local**: **NORTEGA**. Todo lo demás lo dejamos por defecto.

Ya, al reiniciar nuestro servidor, podremos lanzar el controlador de dominios de Active Directory y gestionar nuestro árbol de **nortega.local** (figura 4).

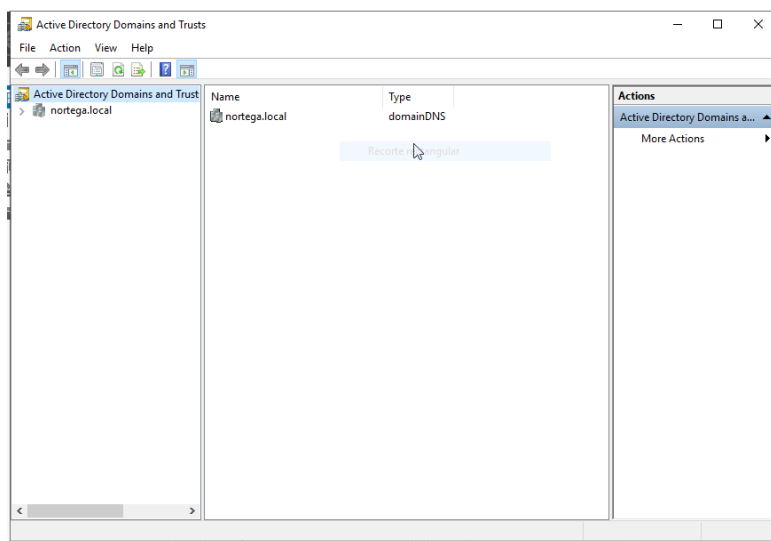


Figura 4: Active Directory instalado.

Queremos también cambiar el usuario de nuestro usuario administrador. Lo cambiamos para que la contraseña sea **I Hate Window\$** (figura 5).

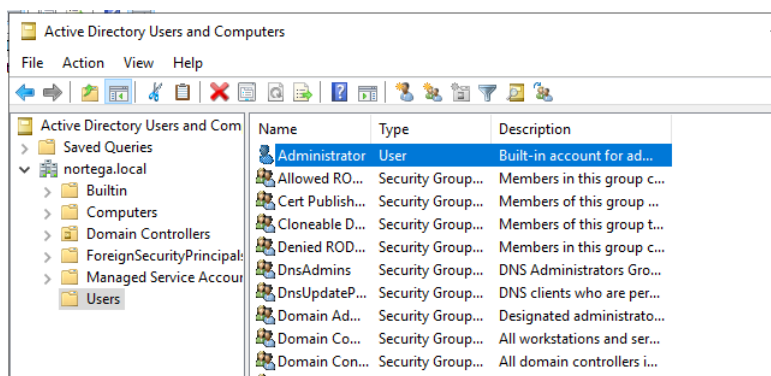


Figura 5: Cambiar contraseña del usuario «Administrator».

4.2. Configuración en Linux

En nuestros sistemas Debian, hemos de instalar las dependencias necesarias para configurar nuestros servidores con el Active Directory. Para esto, primero instalamos los paquetes:

```
# apt install realmd libnss-sss libpam-sss sssd sssd-tools \
    adcli samba-common-bin oddjob oddjob-mkhomedir packagekit
```

Una vez instalados actualizamos el *host name* de nuestro dispositivo para que sea igual al formato `<hostname>.nortega.local` (como subdominio):

```
# hostnamectl set-hostname <hostname>.nortega.local
```

Una vez hecho esto, podremos buscar nuestro servicio Active Directory utilizando el comando de `realm`, y finalmente conectándonos al servicio con el mismo comando de la forma siguiente:

```
# realm discover nortega.local
nortega.local
type: kerberos
realm-name: NORTEGA.LOCAL
domain-name: nortega.local
configured: no
server-software: active-directory
client-software: sssd
required-package: sssd-tools
required-package: sssd
required-package: libnss-sss
required-package: libpam-sss
required-package: adcli
required-package: samba-common-bin
# realm join -U administrator nortega.local
```

Una vez conectado, hemos de editar el archivo de configuración de PAM localizado en `/usr/share/pam-configs/mkhomedir` para que pueda crear los directorios *home* de nuestros usuarios. Lo editamos para que se parezca a lo siguiente:

```
Name: Create home directory on login
Default: yes
Priority: 900
Session-Type: Additional
Session:
    optional          pam_mkhomedir.so
```

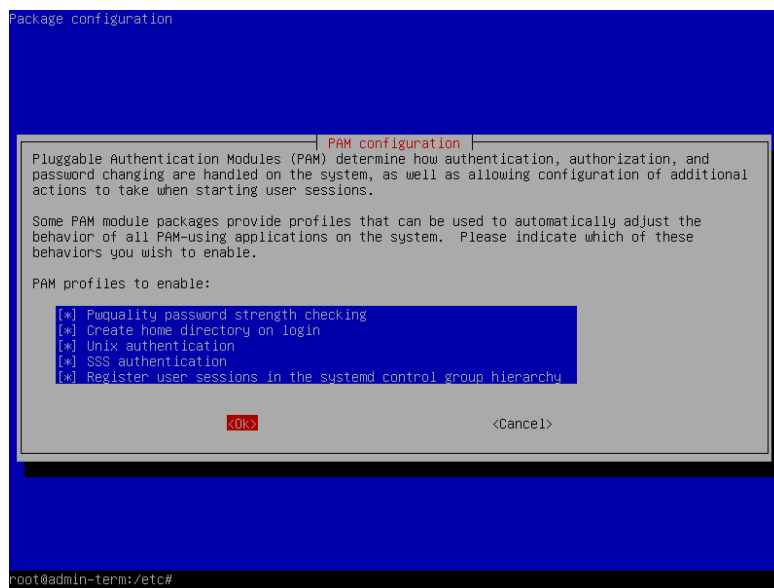



Figura 6: Habilitando la opción de crear una carpeta home al entrar.

Hecho esto, corremos el comando `pam-auth-update` y cambiamos la opción de «Create home directory on login» para habilitarlo (figura 6).

Una vez que lo tengamos configurado, ya podemos reiniciar el servicio `sssd` y podremos usar el Active Directory en nuestro dispositivo:

```
# systemctl restart sssd
```

5. Terminal de Administración

En este dispositivo lo más importante que debemos instalar, para el propósito de administrar los servidores de la red, es SSH. Esto debería de estar disponible ya si hemos instalado nuestro Debian con la opción «SSH server» mencionado antes (que instala el cliente y el servidor).

6. Base de Datos

La base de datos, como se explicó anteriormente, será un servidor con el programa MariaDB. Lo primero es instalar MariaDB como tal:

```
# apt update
# apt install mariadb-server
```


Aunque el paquete se llama `mariadb-server`, incluye también un cliente muy espartano: `mysql`. Pero antes de interactuar con la base de datos queremos finalizar la instalación. Esto se hace con el comando siguiente: `mysql_secure_installation`.

Después de configurar la contraseña de `root`, veremos que nos pide si la autenticación debería de ser por «`unix_socket`». Esto no es nuestro caso ya que queremos conectarnos remotamente desde nuestro servidor web, así que respondemos que no: «`n`». Luego nos preguntará si queremos eliminar los usuarios anónimos. Esto puede ser buena idea para evitar que aquellos que no tengan la autorización para acceder a nuestro servidor de base de datos no lo puedan hacer, así que respondemos que sí: «`y`». La próxima pregunta tiene que ver con el acceso remoto al usuario `root` de la base de datos, y si se debería de permitir. Esto puede ser un riesgo de seguridad, ya que `root` tiene privilegios para acceder y manipular todos los datos que tenemos. También, si nosotros queremos acceder a este usuario, lo podemos hacer accediendo primero a la máquina por SSH desde el terminal administrador y luego entrando localmente como el usuario `root` de la base de datos. Por eso lo racional es no permitir (o *despermitir*): «`y`». La próxima opción es para eliminar la base de datos de prueba y todo acceso a ello. Como no tendremos necesidad de utilizarla, podemos eliminarla: «`y`». Finalmente nos pregunta si queremos actualizar todos los privilegios en este momento. Ya se hará en un futuro (cuando reiniciamos la máquina, por ejemplo), pero si queremos empezar a trabajar ya sobre la base de datos, lo que nos conviene es actualizarlos: «`y`». Una vez que hayamos terminado, ya podremos conectarnos al servidor de base de datos con el comando `mysql` de la forma siguiente:

```
# mysql -u root -p
```

```
Enter password:
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
```

```
Your MariaDB connection id is 31
```

```
Server version: 10.5.18-MariaDB-0+deb11u1 Debian 11
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]>
```

Dentro de nuestro base de datos queremos montar la base de datos de nuestro Wiki, que se denominará `my_wiki`. Crearemos también el usuario

que accederá a este base de datos. Finalmente concederemos a ese usuario los privilegios sobre ese base de datos.

```
CREATE DATABASE my_wiki;  
CREATE USER 'mywikiuser'@'10.0.2.%' IDENTIFIED BY 'mywikipass';  
GRANT ALL PRIVILEGES ON my_wiki.* TO 'mywikiuser'@'10.0.2.%';  
FLUSH PRIVILEGES;
```

Para salir sólo hemos de insertar el comando `quit`; o usar la combinación de teclas `Ctrl+D`, como con cualquier otro *shell*.

Aunque ya está configurado para usarse de manera local, aún hemos de configurarlo para su uso de forma remota, ya que el servidor web que hará uso de él estará en otro ordenador que el servidor de base de datos. Para esto hemos de editar el archivo que se encuentra en `/etc/mysql/mariadb.conf.d/50-server.cnf` y comentar – poner un `#` al principio de la línea – la línea que contiene la cadena de texto `bind-address`. Al hacer esto, reiniciamos el servicio:

```
# systemctl restart mariadb.service
```

7. Servidor Web

7.1. Nginx

En nuestro servidor web, con nombre `web0` queremos instalar el programa Nginx. Para esto hemos de correr los comandos siguientes:

```
# apt update  
# apt install nginx
```

Esto no sólo instalará Nginx, sino que también lo inicializará, de tal modo que si intentas acceder a nuestro servidor web en un navegador por el puerto 80, te saldrá la página por defecto de Nginx (figura 7).

Como nuestra intención es crear una página web de WikiMedia, será necesario instalar PHP además de algunos módulos:

```
# apt install php php-intl php-xml php-mbstring php-fpm php-mysql
```

Una vez instalado, ya podemos instalar nuestra aplicación web WikiMedia. Para hacer esto, descargamos primero la aplicación de la página principal¹ y pasarlo a nuestro servidor web (que se puede hacer mediante el comando `scp`). Ahí lo pasamos a un directorio que servirá de directorio raíz

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Figura 7: Página de Nginx por defecto.

para nuestro sitio web, que para nosotros será `/var/www/wiki`, y a este directorio le cambiamos el propietario de forma recursiva a `www-data`. Esto lo hacemos para que Nginx – que corre como este usuario – pueda editar los archivos y crear algunos nuevos.

```
# chown -R www-data:www-data /var/www/wiki
```

Ahora, para configurar Nginx para este directorio, hemos de navegar al directorio `/etc/nginx/sites-available`. Ahí crearemos una copia del archivo `default` que se denominará `wiki.conf`. Lo editamos para que sea de la forma siguiente (habilitando PHP):

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/wiki;  
  
    index index.php index.html index.htm;  
  
    server_name nortega.local;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    location ~ /\.php$ {  
        include snippets/fastcgi-php.conf;  
    }  
}
```

¹<https://www.mediawiki.org/wiki/Download>


```

        fastcgi_pass unix:/run/php/php-fpm.sock;
    }
}

```

Luego, nos pasamos al directorio `/etc/nginx/sites-enabled` y hacemos dos cosas: primero, eliminar el enlace a la configuración por defecto; segundo crear un enlace simbólico a nuestra configuración que acabamos de crear. Una vez hecho esto, podemos verificar si nuestra configuración está bien corriendo una prueba de Nginx.

```

# rm default
# ln -s ../sites-available/wiki.conf ./
# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is
successful

```

7.2. Certificado SSL

A nosotros también nos interesa que los usuarios tengan acceso a un servicio seguro, que puedan conectarse por medio de HTTPS. Para eso es necesario crear un certificado SSL que podamos usar. Normalmente esto se haría desde una entidad autenticadora (como *Let's Encrypt*), pero nosotros vamos a hacer esto de forma *autofirmada*. Para esto hemos de crear el certificado además de la llave asociada. Esto lo podemos hacer con el comando `openssl` de la forma siguiente:

```

# openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/private/nortega.local.key \
    -out /etc/ssl/certs/nortega.local.crt

```

Esto nos generará tanto el certificado localizado en el directorio `/etc/ssl/certs`, como la llave **privada** (que no se debe de compartir) en el directorio `/etc/ssl/private`.

También queremos generar un grupo Diffie-Hellman para la generación de claves de sesiones privadas. Esto permite que las sesiones antiguas puedan seguir siendo privadas aunque haya habido una fuga de la clave de sesión. Este proceso puede tardar un rato dependiendo de las capacidades del servidor.

```

# openssl dhparam -out /etc/nginx/dhparam.pem 4096

```


Una vez que lo hayamos generado, tenemos que añadir esto a nuestro servidor web. Para esto, primero crearemos un *snippet*, que son pequeños trozos de código de configuración en Nginx reutilizables. En `/etc/nginx/snippets` crearemos un archivo con el nombre `self-signed.conf` que contendrá lo siguiente:

```
ssl_certificate /etc/ssl/certs/nortega.local.crt;
ssl_certificate_key /etc/ssl/private/nortega.local.key;

ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:
    DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:
    DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";

ssl_dhparam /etc/nginx/dhparam.pem;
ssl_ecdh_curve secp384r1;
```

Una vez creado este archivo, ya podemos modificar nuestro archivo de configuración de Nginx, no sólo para utilizar este certificado y proveer un servicio HTTPS, sino también para redireccionar todo el tráfico HTTP a nuestro servicio HTTPS (para forzar una conexión segura). Para esto, nuestro archivo `wiki.conf` que hemos editado anteriormente lo debemos de cambiar para que se parezca a lo siguiente:

```
server {
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    include snippets/self-signed.conf;
```



```

    root /var/www/wiki;
    index index.php index.html index.htm;
    server_name nortega.local;

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

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

server {
    listen 80;
    listen [::]:80;

    server_name nortega.local;

    return 301 https://$server_name$request_uri;
}

```

Ya con esto podremos reiniciar el servicio y ya se podrá acceder por medio de HTTPS, y si se intenta acceder por medio de HTTP se redireccionará de forma permanente (el código 301) al servicio HTTPS.

7.3. MediaWiki

Ya podemos instalar lo viene a ser nuestro servicio de MediaWiki en nuestro servidor web. Siguiendo las instrucciones de MediaWiki, lo primero que hemos de hacer es descargar el ZIP de su página principal, y extraerlo sobre nuestro directorio raíz web (i.e. /var/www/wiki). Una vez instalado, hemos de navegar a la dirección de nuestro servidor para su configuración.

Una vez adentro, lo primero que nos preguntará es para especificar la lengua de la interfaz. Elegiremos el inglés, ya que es lengua universal y ayudará también a la hora de buscar ayuda en internet de nuestros problemas. Luego nos preguntará acerca de la configuración de base de datos. Aquí introduciremos la información para conectarse de forma remote a nuestro servidor de base de datos, que se ubica en la dirección IP de 10.0.2.10 (figura 8a).

Database type:
 * MariaDB, MySQL, or compatible

MariaDB/MySQL settings

Database host:

Identify this wiki

Database name (no hyphens):

Database table prefix (no hyphens):

User account for installation

Database username:

Database password:

(a) Configuración de la base de datos.

MediaWiki 1.39.1 installation

Install

- Setting up database... done
- Creating tables, step one... done
- Creating database user... done
- Creating tables, step two... done
- Populating default interwiki table... done
- Initializing statistics... done
- Generating secret keys... done
- Prevent running unneeded updates... done
- Restoring MediaWiki services... done
- Creating administrator user account... done
- Creating main page with default content... done
- Database was successfully set up

(b) Finalización de la instalación de MediaWiki.

Figura 8: Instalación de MediaWiki.

Más adelante en el proceso de configuración de MediaWiki nos pedirá la configuración de el usuario administrador. Aquí meteremos un usuario que se llamará «Admin», por facilidad de uso, y con la contraseña «MyAdminPass».

Una vez que hayamos configurado todo esto, y hemos dado a «Continuar» hasta el final, nos dirá que la instalación ha resultado exitoso (figura 8b). Luego nos hará descargar un archivo `LocalSettings.php` con toda la configuración que acabamos de hacer. Ésto lo hemos de instalar en nuestro directorio raíz también (i.e. `/var/www/wiki`).

Una vez hecho esto, le podemos dar al enlace abajo que pone «enter your wiki» y nos saldrá la Wiki que acabamos de crear.

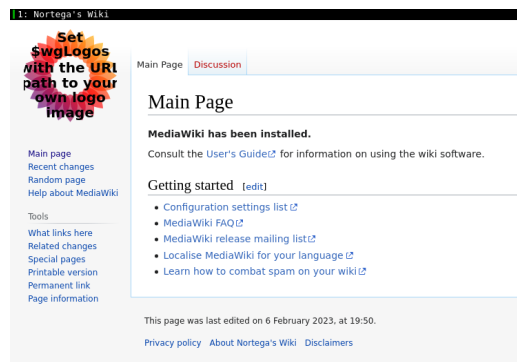


Figura 9: Página inicial de nuestro Wiki.

8. Derechos de Autor y Licencia

Copyright © 2023 Nicolás A. Ortega Froya <nicolas@ortegas.org>

Este documento se distribuye bajo los términos y condiciones de la licencia Creative Commons Attribution No Derivatives 4.0 International.