

Guía de Despliegue

Grupo 4



Luis Izquierdo Gutiérrez

Adrián De Diego Fernández

Saúl García Girón

Naiara Martínez Odriozola

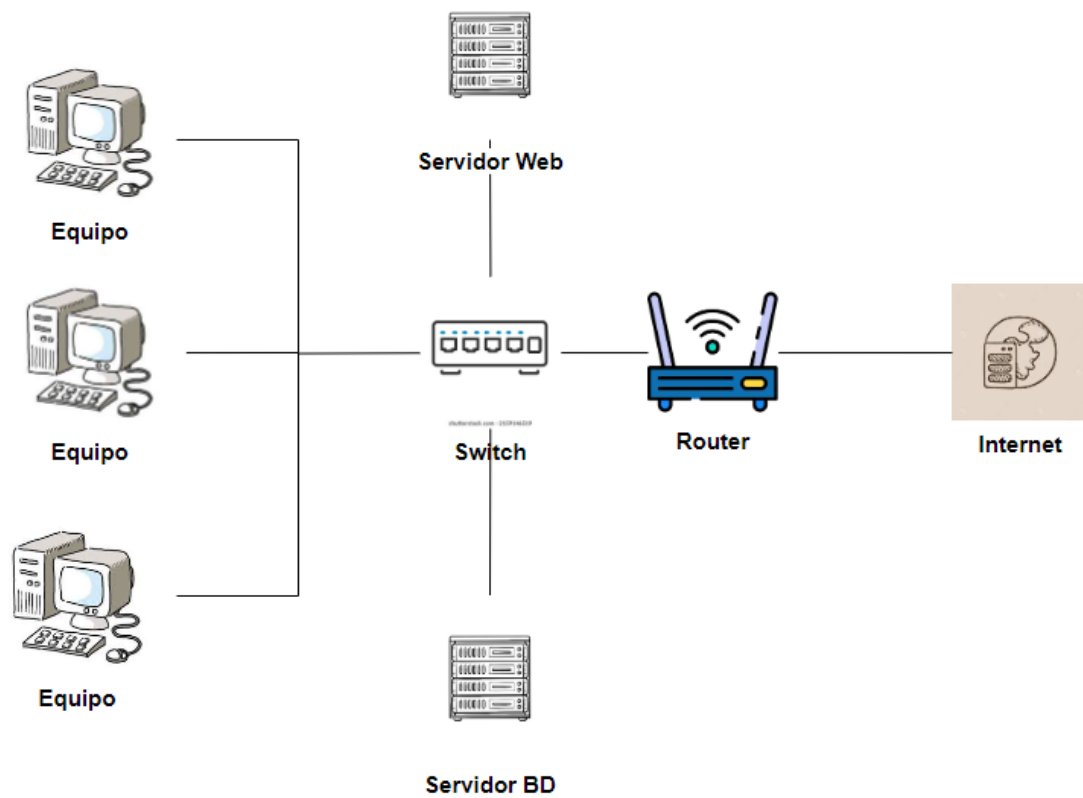
Miguel Gutiérrez Caso

Diego Rodríguez Rubio

ÍNDICE

Diagrama de Red.....	3
Servidor de base de datos.....	4
Servidor web.....	5-7
Bibliografía.....	8

Diagrama de Red



Servidor de base de datos

Elección de la infraestructura del servidor de base de datos

Hemos utilizado una base de datos Mariadb hosteado en un docker en una máquina virtual Alpine Linux por su facilidad de uso y funcionamiento estable

Proceso de instalación y configuración

Primero instalamos Alpine Linux en una máquina virtual, después en dicho Alpine instalamos un docker de Mariadb.

Creamos un nuevo usuario “developer” de Mariadb con permisos para conectarse desde cualquier ip.

Usando dicho usuario en mysql workbench reproducimos el sql que está en [github](#).

Por último, borrar la base de datos por defecto.

Acceso desde la aplicación

La aplicación accede a la base de datos usando un [patron singleton](#) el cual utiliza la ip y el usuario para hacer una conexión a la base de datos.

Servidor Web

Hemos utilizado la máquina virtual Ubuntu server debido a su fácil manera de instalar apache y ssh por comandos. También probamos a hacerlo con Alpine, pero nos parecía menos intuitivo para esta tarea en concreto.

Proceso de instalación: Se descarga la máquina virtual en nuestro caso Ubuntu server luego se configura la misma. Posteriormente dentro de la máquina virtual se instala el apache con una serie de comandos a su vez configuramos el firewall e instalamos el ssh con todo esto ya tenemos un servidor web al que podemos acceder con la ip del dispositivo en el que se creó .

[Creamos es servidor web](#)

[Paso 1: Instalar Apache](#)

Apache está disponible en los repositorios de software predeterminados de Ubuntu, lo que permite instalarlo con las herramientas convencionales de administración de paquetes.

Comencemos actualizando el índice de paquetes locales para que reflejen los últimos cambios anteriores:

```
sudo apt update
```

A continuación, instale el paquete apache2:

```
sudo apt install apache2
```

Una vez confirmada la instalación, apt instalará Apache y todas las dependencias necesarias.

[Paso 2: Ajustar el firewall](#)

Antes de probar Apache, es necesario modificar los ajustes de firewall para permitir el acceso externo a los puertos web predeterminados. Suponiendo que siguió las

instrucciones de los requisitos previos, debería tener un firewall UFW configurado para que restrinja el acceso a su servidor.

Durante la instalación, Apache se registra con UFW para proporcionar algunos perfiles de aplicación que pueden utilizarse para habilitar o deshabilitar el acceso a Apache a través del firewall.

Enumere los perfiles de aplicación ufw escribiendo lo siguiente:

```
sudo ufw app list
```

Obtendrá una lista de los perfiles de aplicación:

Output

Available applications:

- Apache
- Apache Full
- Apache Secure
- OpenSSH

Como lo indica el resultado, hay tres perfiles disponibles para Apache:

Apache: este perfil abre solo el puerto 80 (tráfico web normal no cifrado)

Apache Full: este perfil abre el puerto 80 (tráfico web normal no cifrado) y el puerto 443 (tráfico TLS/SSL cifrado)

Apache Secure: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado)

Se recomienda habilitar el perfil más restrictivo, que de todos modos permitirá el tráfico que configuró. Debido a que en esta guía aún no configuramos SSL para nuestro servidor, solo deberemos permitir el tráfico en el puerto 80:

```
sudo ufw allow 'Apache'
```

Puede verificar el cambio escribiendo lo siguiente:

```
sudo ufw status
```

Creamos el servidor Sftp

Primero creamos un grupo:

```
$ sudo groupadd sftpgroup
```

Luego creamos un usuario en la carpeta en la que vamos a tener el HTML:

```
$ sudo useradd -G sftpgroup -d /var/www -s /sbin/nologin sftpuser
```

Creamos una contraseña para el usuario:

```
$ sudo passwd sftpuser
```

Configuramos el chroot y creamos la carpeta en la que añadiremos el HTML

```
$ sudo mkdir -p /var/www
```

A continuación, se asigna la propiedad al usuario root con chown:

```
$ sudo chown root /var/www
```

Asigna derechos de lectura y ejecución al grupo:

```
$ sudo chmod g+rx /var/www
```

A continuación, crea un subdirectorio y designa a sftpuser como propietario:

```
$ sudo mkdir -p /var/www/html
```

```
$ sudo chown sftpuser:sftpuser /var/www/html
```

Nos conectamos al Ubuntu server

Usando FileZilla nos conectamos usando la ip y la contraseña

Instalación de ssh

Sin mas comenzamos usando el siguiente comando después de iniciar sesión en el sistema:

```
$ sudo apt install ssh
```

Revisamos que el servicio de SSH este instalado y activado con el siguiente comando:

```
$ systemctl status ssh
```

En caso de que el servicio no este activo puedes activarlo con el siguiente comando:

```
$ sudo systemctl start ssh
```

o desactivarlo con el siguiente:

```
$ sudo systemctl stop ssh
```

Bibliografía

<https://jcastaneda.com/servidores/configurar-ssh-en-ubuntu-server-20-04/>

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>

<https://www.ionos.es/digitalguide/servidores/configuracion/instalar-un-sftp-server-en-ubuntu/>