

# Proyecto Final SISTEMAS 101

## INFRAESTRUCTURA DE RED Y SERVICIOS WEB

Para poder llevar a cabo este proyecto he utilizado los siguientes programas:

-**VirtualBox**: <https://www.virtualbox.org>

-**UBUNTU** versiones (imagen ISO) **24.04 y 20.04** :<https://www.ubuntu.com>

### 1.CONFIGURACIÓN DE RED INTERNA ENTRE LAS Vms

\*Una vez instalados todos los programas he procedido a crear las 4 VMs necesarias para la realización del proyecto ajustando los recursos de cada una de ellas (núcleos,disco duro y RAM).

### PASO 1

**1ª VM SERVIDOR WEB:** asigno núcleos,configuro disco duro y tamaño de la RAM,instalo imagen **ISO** e inicio el proceso de instalación.

Un vez terminado este proceso en **CONFIGURACIÓN RED**→ADAPTADOR 1→**CONECTADO A**:→ADAPTADOR PUENTE→**NOMBRE**:Intel (R)wi-fi.  
ADAPTADOR 2→**CONECTADO A**:→RED INTERNA→INTNET

**2ªVM SERVIDOR DE BBDD:** realice el mismo proceso de configuración que con la otra VM,variando un poco los recursos,pero en el momento de la instalación se queda congelada la imagen durante más de 1h,decido cancelar la instalación y realizarla nuevamente volviendo a modificar los recursos por si fuera ese el motivo de la no instalación pero vuelve a pasar lo mismo y finalmente decido clonar las Vms y realizar los cambios de nombre,IP y configuración de RED en cada uno de los clones.

## PASO 2

### \*Configuración de red de las Vms

Accedo a la terminal de mi 1<sup>a</sup> VM SERVIDOR WEB y realizo los siguientes pasos:

**ifconfig**: para ver la dirección IP.

**route -n**: muestra la tabla de enrutamiento de la red.

## PASO 3

Procedo a configurar de manera manual las IPs:

Icono RED→**RUEDA DE AJUSTES**→**MANUAL**→**INTRODUCIR IP** (obtenida del paso anterior)→**MÁSCARA DE SUBRED**→255.255.255.0→**PUERTA DE ENLACE**→IP de mi router. En el apartado DNS→**MANUAL**→>AÑADIR 1.1.1.1, 8.8.8.8(DNS públicas de Cloudflare y Google)

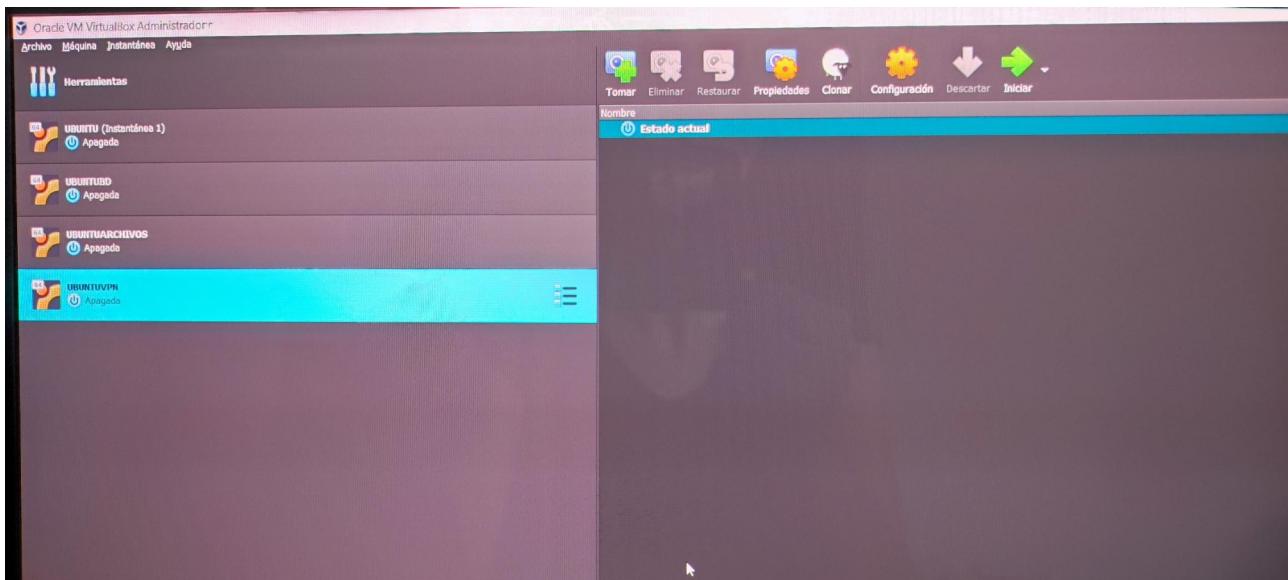
Para verificar que este proceso esta bien realizamos un **ping** a una web desde la terminal para comprobar que hay conexión.

**UBUNTU SERVIDOR WEB**→192.168.1.253

**UBUNTU BBDD**→192.168.1.252

**UBUNTU ARCHIVOS Y MONITOREO**→192.168.1.251

**UBUNTU VPN Y BACKUP**→192.168.1.250



## PASO 4

Realizo la activación del SSH, que es un protocolo de red destinado a conectar las VMs a las que se accede por línea de comandos:

```
mon&mon-server: sudo apt install openssh-server  
mon&mon-server: sudo systemctl enable ssh  
mon&mon-server: sudo ufw allow ssh
```

Ahora ya podemos ir al cmd de Windows y conectarnos con:

**ssh mon@192.168.1.251**

## 2. INSTALAR Y CONFIGURAR SERVICIOS WEB Y DE BASE DE DATOS

### PASO 1

Apache **es un servidor web que se encarga de almacenar, procesar y servir las páginas web a los usuarios de las mismas.**

#### **1. Instalar Apache:**

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

#### **2. Ajustar el firewall:**

```
sudo ufw app list
```

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)**

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

Output

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Apache	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)

Verificamos que Apache (v6) esta permitido.

### 3.Comprobar el Servidor Web

```
sudo systemctl status apache2
```

Output

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor
preset: enabled)
   Active: active (running) since Thu 2020-04-23 22:36:30 UTC; 20h ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 29435 (apache2)
    Tasks: 55 (limit: 1137)
   Memory: 8.0M
      CGroup: /system.slice/apache2.service
              ├─29435 /usr/sbin/apache2 -k start
              ├─29437 /usr/sbin/apache2 -k start
              └─29438 /usr/sbin/apache2 -k start
```

Acceder a la página de destino predeterminada de Apache para confirmar que el software funcione correctamente mediante su dirección IP

```
hostname -I
```

Cuando tenga la dirección IP de su servidor, se introduce en la barra de direcciones del navegador:

```
http:// IP del servidor
```

## 4.Configurar y crear hosts dinámico virtual:



### Apache2 Ubuntu Default Page

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

#### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|       '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin/apache2` directly will not work with the default configuration.

#### Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, `public_html` directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

#### Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Crear un servidor dinámico DNS en NO-IP

monicadual.sytess.net

The screenshot shows the IPVanish dashboard with the following sections:

- Nombre de host usado:** 1 de 1
- All Hostnames are updating successfully:** 0
- Notificaciones importantes:** 1
- Nombres de host DNS dinámicos:** 1 of 1  
monicadual.sytes.net [Modificar](#)  
[Actualizar a DNS Dinámico mejorado](#)
- Información de la cuenta:**
  - IP de último inicio de sesión: 90.163.47.57
  - Último inicio de sesión: Jun 17, 2024 10:30:54 PDT
  - Estado de autenticación de dos factores:
    - No activo
    - Activar
- Cliente de actualización dinámica para Windows:** Mantenga su dirección IP actual sincronizada con su nombre de host o dominio de No-IP con nuestro Cliente de Actualización Dinámica (DUC).
- Secure your Online Privacy:** IP VANISH
- No-IP Estado del sistema:**

Hacemos un nslookup

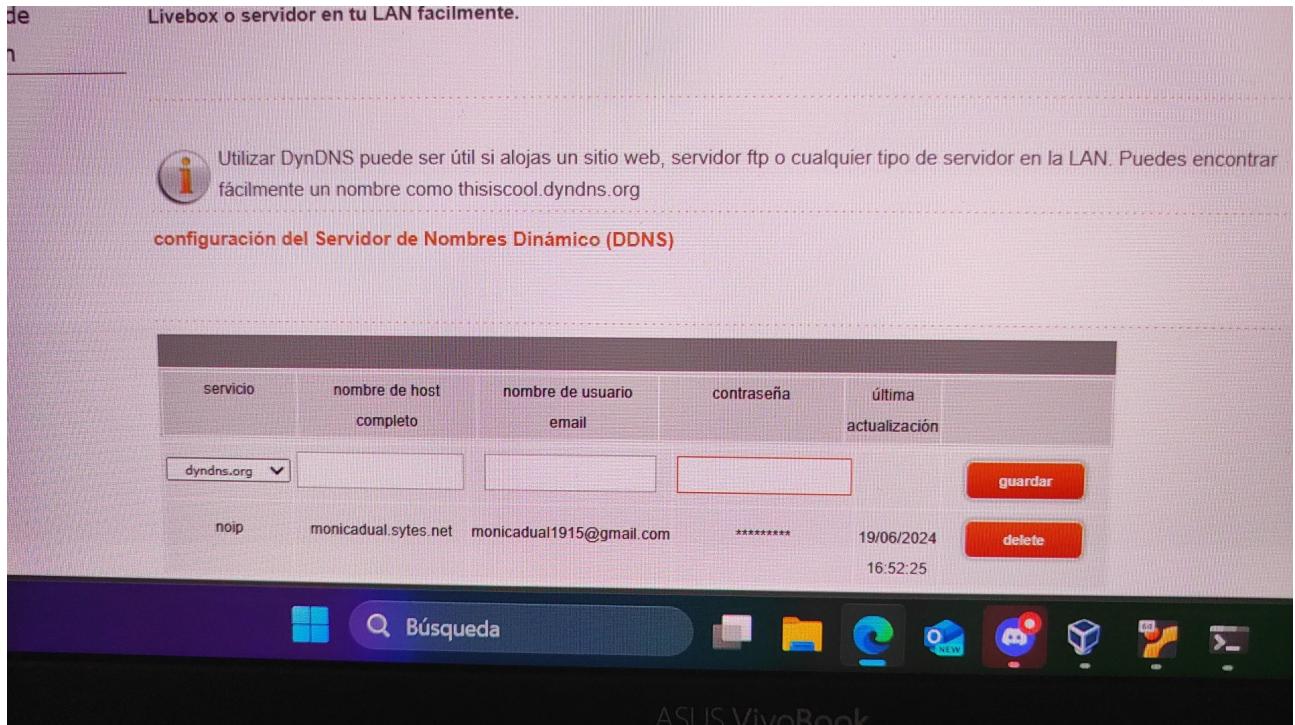
monicadual.sytes.net 90.163.47.57

o un networkctl status (para obtener información de un dominio y ver si tenemos acceso a internet)

Accedo a mi operadora de servicios de telefonía e internet con mi n.<sup>o</sup> de Gateway del router y realizo los ajustes necesarios en configuración de avanzada-> configuración de red->para mapear el puerto 80.

SSH	TCP					22	accept	<a href="#">delete</a>			
NTP	UDP					123	accept	<a href="#">delete</a>			
NNTP	TCP					119	accept	<a href="#">delete</a>			
NNTPS	TCP					563	accept	<a href="#">delete</a>			
DNS	both					53	accept	<a href="#">delete</a>			
IMAP	TCP					143	accept	<a href="#">delete</a>			
IMAPS	TCP					993	accept	<a href="#">delete</a>			
STUN	UDP					3478	accept	<a href="#">delete</a>			
IRC	TCP					6666-6667	accept	<a href="#">delete</a>			
mDNS	UDP					5353	accept	<a href="#">delete</a>			
UPnP	UDP					1900	accept	<a href="#">delete</a>			
HTTP	TCP	192.168.1.253	255.255.255.0	80	90.163.47.57	255.255.255.0	80	accept	<a href="#">delete</a>		

Configuro el nombre de mi sitio web en DynDNS:



Se crea el directorio:

```
sudo mkdir /var/www/monicadual.sytes.net
```

Asigno la propiedad del directorio:

```
sudo chown -R $USER:$USER /var/www/monicadual.sytes.net
```

Permisos:

```
sudo chmod -R 755 /var/www/monicadual.sytes.net
```

Creo una página de ejemplo index.html:

```
sudo nano /var/www/monicadual.sytes.net/index.htm
```

```
<html>
<head>
<title>Welcome to Your_domain!</title>
</head>
<body>
<h1>Success! The your_domain virtual host is working!</h1>
</body>
</html>
```

**Se crea un nuevo archivo de configuración:**

```
sudo nano /etc/apache2/sites-available/monicadual.sytes.net
```

```
<VirtualHost *:80>
```

```
    ServerAdmin monicadual1915@gmail.com
```

```
    ServerName monicadual.sytes.net
```

```
    ServerAlias www.monicadual.sytes.net
```

```
    DocumentRoot /var/www/monicadual.sytes.net
```

```
    ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
</VirtualHost>
```

Se habilita el archivo con:

```
sudo a2ensite monicadual.sytes.net.conf
```

Se deshabilita el sitio predeterminado por defecto:

```
sudo a2dissite 000-default.conf
```

Realizo una prueba para verificar si todo ok:

```
sudo apache2ctl configtest
```

Output

```
Syntax OK
```

**Se reinicia Apache para implementar los cambios:**

```
sudo systemctl restart apache2.
```

Con el servicio web estuve bastante atascada intentando descifrar el motivo por el que no llegaba a publicar mi web, volví a montarlo varias veces desde el principio pero seguía sin salir con lo cuál opte por dejarlo así y esperar a que me revisaras el proceso.

### **3.INSTALACIÓN Y CONFIGURACIÓN DE MySQL**

**MySQL es un sistema de administración de bases de datos de código abierto, que comúnmente se instala como parte de la popular pila LAMP(Linux, Apache, MySQL, PHP/Python/Perl). Implementa el modelo relacional y utiliza el Structured Query Language (más conocido como SQL) para administrar los datos.**

#### **1 PASO**

Instalar MySQL:

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

```
sudo apt install mysql-server
```

#### **2 PASO**

Configurar MySQL

```
sudo mysql_secure_installation
```

Configurar el complemento para validad contraseña

*Output*

```
Securing the MySQL server deployment.
```

```
Connecting to MySQL using a blank password.
```

```
VALIDATE PASSWORD COMPONENT can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD component?
```

```
Press y|Y for Yes, any other key for No: Y
```

```
There are three levels of password validation policy:
```

```
LOW      Length >= 8  
MEDIUM  Length >= 8, numeric, mixed case, and special characters  
STRONG   Length >= 8, numeric, mixed case, special characters and dictionary  
file
```

```
Please enter 0 = LOW, 1
```

Contraseña para el usuario root en MySQL

*Output*

```
Please set the password for root here.
```

```
New password:
```

```
Re-enter new password:
```

Si utilizó el complemento para validar la contraseña, recibirá un comentario acerca de la seguridad de su contraseña nueva.

**Output**

```
Estimated strength of the password: 100
Do you wish to continue with the password provided? (Press y|Y for Yes, any other key for No) : Y
```

Desde allí, puede pulsar **Y** y luego **ENTER** para aceptar los valores predeterminados para todas las preguntas siguientes. Con esto, se eliminarán algunos usuarios anónimos y la base de datos de prueba, se deshabilitarán las credenciales de inicio de sesión remoto de root y se cargarán estas nuevas reglas para que MySQL aplique de inmediato los cambios que realizó.

Tenga en cuenta que aunque ha configurado una contraseña para el **root** user de MySQL, este usuario no está configurado para autenticar con una contraseña cuando se conecte al shell de MySQL. Si lo desea, puede ajustar este ajuste siguiendo el Paso 3.

### **PASO 3**

Ajustar la autenticación y los privilegios de usuario (opcional)

En los sistemas Ubuntu con MySQL 5.7 (y versiones posteriores), el usuario **root** de MySQL se configura para la autenticación usando el complemento **auth\_socket** de manera predeterminada en lugar de una contraseña. Esto en muchos casos proporciona mayor seguridad y utilidad, pero también puede generar complicaciones cuando deba permitir que un programa externo (como phpMyAdmin) acceda al usuario.

Para usar una contraseña para conectar con MySQL como **root**, deberá cambiar su método de autenticación de **auth\_socket** a otro complemento, como **caching\_sha2\_password** o **mysql\_native\_password**. Para hacer esto, abra la consola de MySQL desde su terminal:

```
sudo mysql
```

**Comprobar con el siguiente comando el método de autenticación utilizado por una de sus cuentas de usuario de MySQL:**

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

**Output**

```
+-----+
+-----+
+-----+-----+
```

```
| user           | authentication_string
| plugin        | host      |
+-----+
+-----+-----+
| debian-sys-maint | $A$005$1S|M#3K
#XslZ.xXUq.crEqTjMvhgOIX7B/zki5DeLA3JB9nh0KwENTwQ4 | caching_sha2_password |
localhost |
| mysql.infoschema |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED |
caching_sha2_password | localhost |
| mysql.session   |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED |
caching_sha2_password | localhost |
| mysql.sys       |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED |
caching_sha2_password | localhost |
| root           |
| auth_socket    | localhost |
+-----+
+-----+-----+
5 rows in set (0.00)
```

Asegúrese de cambiar `password` por una contraseña segura de su elección, y tenga en cuenta que este comando cambiará la contraseña `root` que estableció en el Paso 2:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'password';
```

**Nota:** La instrucción `ALTER USER` previa establecer el `root` user de MySQL para autenticar con el complemento `caching_sha2_password`. Según la documentación oficial de MySQL `caching_sha2_password` es el complemento de autenticación preferido por MySQL, ya que proporciona un cifrado de contraseña más seguro que el anterior, aunque aún usado ampliamente, `mysql_native_password`.

Sin embargo, muchas aplicaciones PHP, como por ejemplo `phpMyAdmin`, no funcionan de forma fiable con `caching_sha2_password`. Si planea usar esta base de datos con una aplicación PHP, es posible que desee establecer `root` para que autentique con `mysql_native_password`:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
```

A continuación, ejecute `FLUSH PRIVILEGES` para indicar al servidor que vuelva a cargar la tabla de permisos y aplique sus nuevos cambios:

```
FLUSH PRIVILEGES;
```

Compruebe de nuevo los métodos de autenticación empleados por cada uno de sus usuarios para confirmar que **root** deje de realizarla usando el complemento de **auth\_socket**:

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

**Output**

```
+-----+
+-----+-----+
| user           | authentication_string
| plugin         | host      |
+-----+
+-----+-----+
| debian-sys-maint | $A$005$1S|M#3K
#Xs1Z.xxUq.crEqTjMvhgOIX7B/zki5DeLA3JB9nh0KwENTwQ4 | caching_sha2_password |
localhost |
| mysql.infoschema |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED |
caching_sha2_password | localhost |
| mysql.session   |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED |
caching_sha2_password | localhost |
| mysql.sys       |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED |
caching_sha2_password | localhost |
| root           | *3636DACC8616D997782ADD0839F92C1571D6D78F
| caching_sha2_password | localhost |
+-----+
+-----+-----+
5 rows in set (0.00 sec)
```

Puede ver en este resultado de ejemplo que el root user de MySQL ahora autentica usando **caching\_sha2\_password**. Una vez que confirme esto en su propio servidor, podrá cerrar el shell de MySQL:

```
mysql> exit
```

## **PASO 4**

Crear una base de datos

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

```
create data base monica;
```

```
show databases;
```

```
use monica;
```

```
Query OK, 1 row affected (0,01 sec)

A mysql> CREATE USER 'mon'@'localhost' IDENTIFIED BY 'monica';
Query OK, 0 rows affected (0,02 sec)

? mysql> GRANT ALL PRIVILEGES ON mon.* TO 'mon'@'localhost';
Query OK, 0 rows affected (0,01 sec)

> mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| monica |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0,01 sec)

mysql>
```

Crear tablas:

create table alumnos (nombre varchar (20), correo varchar (20), edad int)

show tables;

describe alumnos;

```
Falló la conexión Falló la activación de la conexión de red

mysql> use monica;
Database changed

mysql> create table alumnos ( nombre varchar(20), correo varchar (20),
Query OK, 0 rows affected (0,02 sec)

A mysql> show tables;
+-----+
| Tables_in_monica |
+-----+
| alumnos |
+-----+
1 row in set (0,01 sec)

? mysql> describe alumnos;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nombre | varchar(20) | YES |   | NULL    |       |
| correo | varchar(20) | YES |   | NULL    |       |
| edad   | int      | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0,01 sec)

mysql>
```

## PASO 5

### INSTALACIÓN DE PHP

**PHP** es el componente de nuestra configuración que procesará el código para mostrar contenido dinámico al usuario final.

```
sudo apt install php libapache2-mod-php php-mysql
```

```
php -v
```

Output

```
PHP 7.4.3 (cli) (built: Mar 26 2020 20:24:23) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
```

Probar el proceso de PHP con el servidor web

Crear un archivo nuevo llamado `info.php` dentro de su carpeta root web personalizada:

```
nano /var/www/your_domain/info.php
```

Con esto se abrirá un archivo vacío. Añada el siguiente texto, que es el código PHP válido, dentro del archivo:

```
<?php
```

```
phpinfo();
```

Para probar esta secuencia de comandos, diríjase a su navegador web y acceda al nombre de dominio o la dirección IP de su servidor, seguido del nombre de la secuencia de comandos, que en este caso es `info.php`:

<http://moni/info.php>

(<http://dominio> o IP/info.php)

Se ve una imagen como esta:

## PHP Version 7.4.3



System	Linux sassy-starfish 5.4.0-26-generic #30-Ubuntu SMP Mon Apr 20 16:58:30 UTC 2020 x86_64
Build Date	Mar 26 2020 20:24:23
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlind.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-filinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v3.4.0, Copyright (c) Zend Technologies  
with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies



En esta página, se proporciona información básica sobre su servidor desde la perspectiva de PHP. Es útil para la depuración y para asegurarse de que sus ajustes se apliquen correctamente.

Si puede ver esta página en su navegador, su instalación de PHP funciona según lo previsto.

Tras comprobar la información pertinente sobre su servidor PHP a través de esa página, es recomendable que elimine el archivo que creó, dado que contiene información confidencial sobre su entorno PHP y su servidor de Ubuntu. Puede usar `rm` para hacerlo:

```
sudo rm /var/www/your_domain/info.php
```

Ahora, podrá crear una secuencia de comandos PHP que se conecte a MySQL y realice consultas relacionadas con su contenido. Cree un nuevo archivo PHP en su directorio web root personalizado usando su editor preferido. En este caso, usaremos nano:

```
nano /var/www/moni/todo_list.php
```

The screenshot shows a terminal window with the MySQL monitor running. The user has entered several commands to connect to the MySQL server as root and run the nano editor on the todo\_list.php file. The terminal also displays the MySQL copyright notice and the command prompt.

```
mysql> nano /var/www/html/moni/bbdd.php
-> ^C
mysql> exit
Bye
mon@mon-server:~$ sudo nano /var/www/html/moni/bbdd.php
mon@mon-server:~$ sudo nano /var/www/html/moni/bbdd.php
mon@mon-server:~$ sudo nano /var/www/html/moni/bbdd.php
mon@mon-server:~$ sudo nano /var/www/html/moni/bbdd.php
mon@mon-server:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 8.0.37-Ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

La siguiente secuencia de comandos PHP establece conexión con la base de datos de MySQL, realiza consultas relacionadas con el contenido de la tabla **todo\_list** y muestra los resultados en una lista. Si hay un problema con la conexión de la base de datos, generará una excepción. Copie este contenido en su secuencia de comandos todo\_list.php:

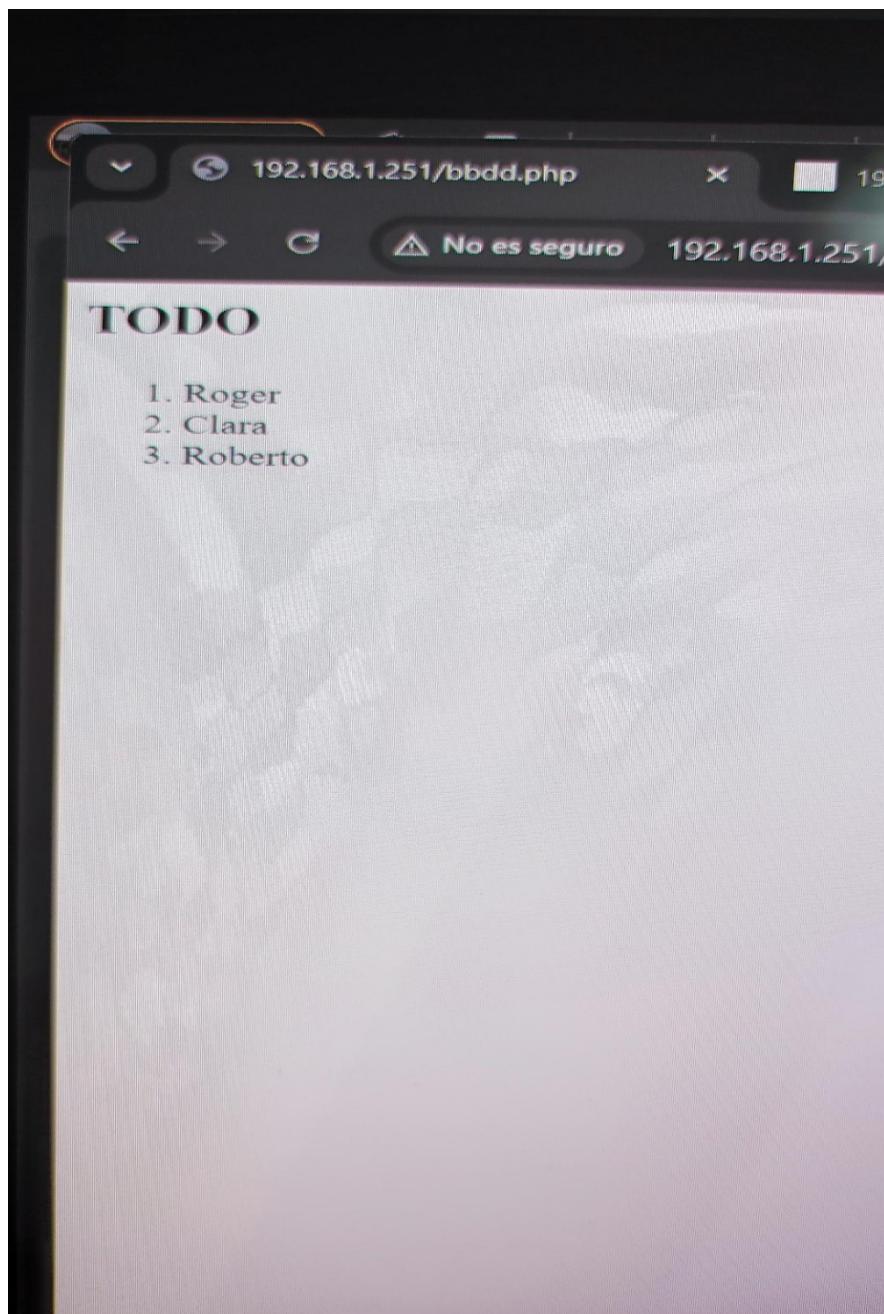
```
/var/www/your_domain/todo_list.php
```

```
<?php
$user = "example_user";
$password = "password";
$database = "example_database";
$table = "todo_list";

try {
    $db = new PDO("mysql:host=localhost;dbname=$database", $user, $password);
    echo "<h2>TODO</h2><ol>";
    foreach($db->query("SELECT content FROM $table") as $row) {
        echo "<li>" . $row['content'] . "</li>";
    }
    echo "</ol>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```

Ahora, puede acceder a esta página en su navegador web al visitar el nombre de dominio o la dirección IP pública de su sitio web seguido de /todo\_list.php:

[http://moni/todo\\_list.php](http://moni/todo_list.php)



### **3.IMPLEMENTAR UN SERVIDOR DE ARCHIVOS Y UNA VPN**

**Samba** es un **protocolo red cliente-servidor** que **permite** a los ordenadores con **sistemas de tipo Unix** (GNU/Linux, MacOS, Solaris, iOS, etc.) **compartan recursos en red** (carpetas, impresoras, discos) con equipos **Windows** y viceversa.

#### **Instalación SAMBA**

##### **Actualización del sistema**

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

#### **Instalación SAMBA**

```
sudo apt install samba -y
```

#### **CONFIGURACIÓN DE SAMBA**

##### **sudo mkdir /samba**

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.backup
```

#### **EDITAR ARCHIVOS**

```
sudo nano /etc/samba/smb.conf
```

#### **AGREGAR UN USUARIO A SAMBA**

##### **sudo useradd [monica]**

#### **ASIGNAR UNA CONTRASEÑA**

```
sudo smbpasswd -a monica
```

#### **CREAR NUEVO DIRECTORIO PÚBLICO**

##### **sudo mkdir /samba/public**

#### **MANEJO DE PERMISOS**

##### **sudo chown -R**

##### **sudo chmod -R**

##### **sudo chgrp sambashare /samba/public**

```
jun 20 22:47:27 mon-server systemd[1]: Starting nmbd.service - Samba NMB Da...
jun 20 22:47:27 mon-server (nmbd)[14474]: nmbd.service: Referenced but unse...
jun 20 22:47:27 mon-server systemd[1]: Started nmbd.service - Samba NMB Da...
mon@mon-server:~$ sudo mkdir /samba
mon@mon-server:~$ sudo nano /etc/samba/smb.conf
mon@mon-server:~$ sudo useradd [nombre-usuario]
mon@mon-server:~$ sudo useradd [monica]
mon@mon-server:~$ sudo mkdir /samba/public
mon@mon-server:~$ sudo chown -R nobody:nogroup /samba/public
mon@mon-server:~$ sudo chmod -R 0777 /samba/public
mon@mon-server:~$ sudo chgrp sambashare /samba/public
mon@mon-server:~$ sudo systemctl restart smbd.service
mon@mon-server:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.251 netmask 255.255.255.0 broadcast 192.168.1.255
          ether 08:00:27:9e:78:bb txqueuelen 1000 (Ethernet)
            RX packets 24890 bytes 19400102 (19.4 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 6003 bytes 690498 (690.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet6 fe80::d054:ec03:4fb0:22c0 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:b9:24:4d txqueuelen 1000 (Ethernet)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 740 bytes 128679 (128.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
          inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Bucle local)
            RX packets 315 bytes 42299 (42.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 315 bytes 42299 (42.2 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mon@mon-server:~$ |
```

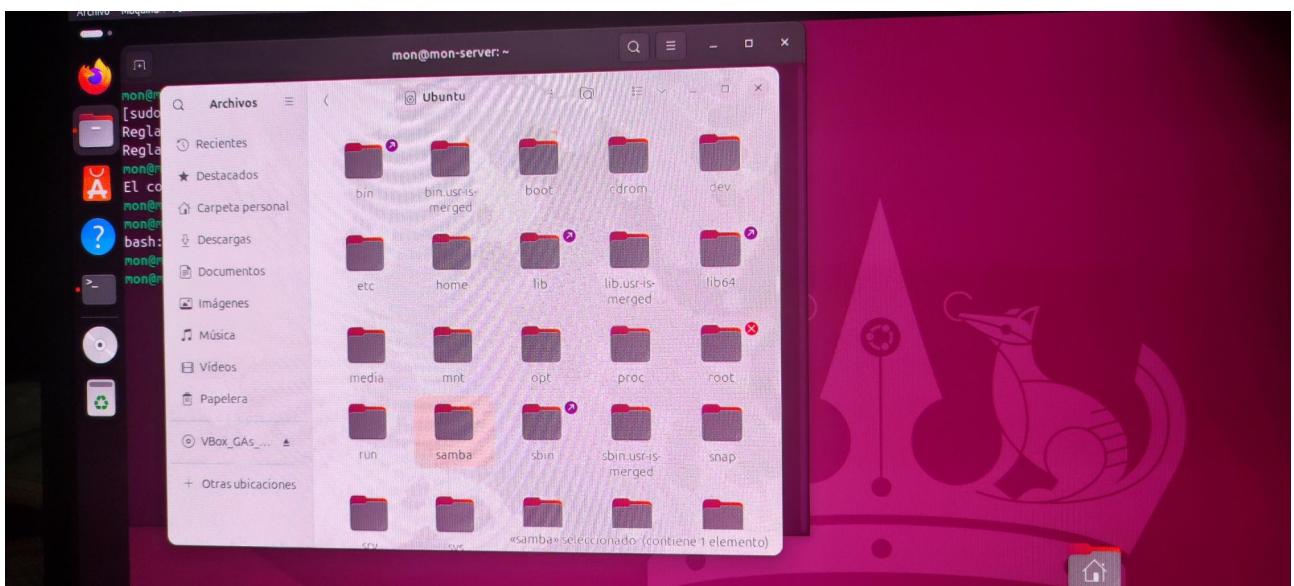
## REINICIAR Y CONFIGURAR EL SERVICIO DE SAMBA

`sudo systemctl restart smbd`

```
sudo systemctl status smbd
```

## CONEXIÓN A SAMBA:

Desde LINUX→EXPLORADOR DE ARCHIVOS→RED



## INSTALAR Y CONFIGURAR UN SERVIDOR OPENVPN

**OpenVPN** es un protocolo de cifrado de código abierto que se utiliza para la comunicación segura a través de la red.

### PASO1

Descargar ejecutable en shell script de github

```
cd /tmp/  
wget https://git.io/vpn -O openvpn-install.sh
```

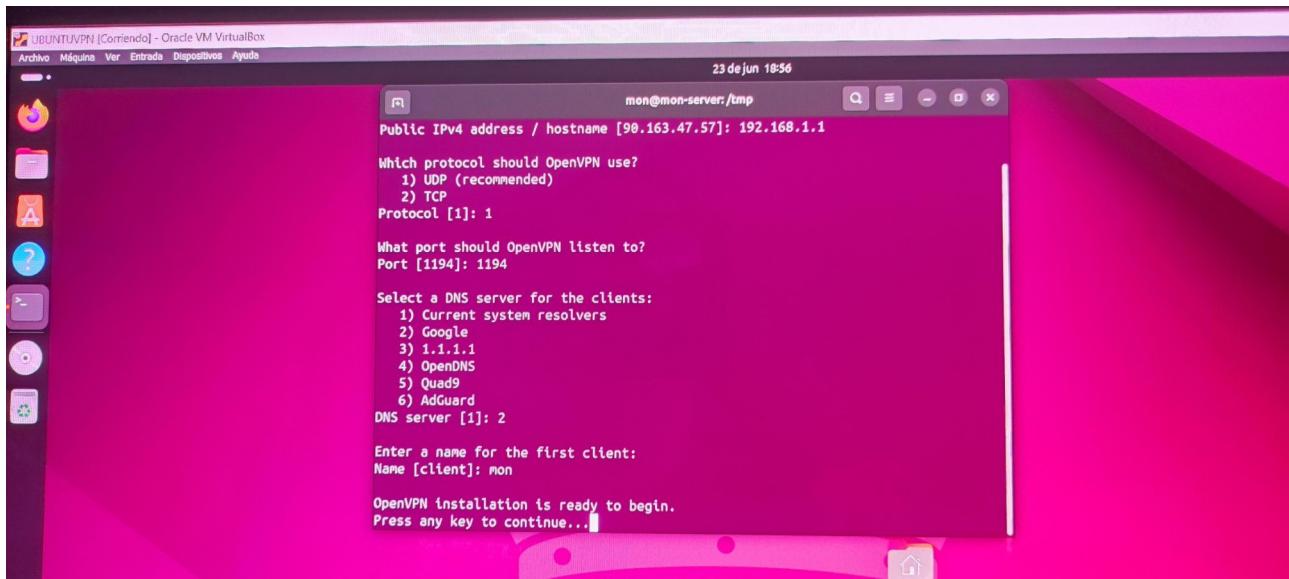
Damos permisos de ejecución al script

```
chmod +x openvpn-install.sh
```

## 1. Ejecutamos el script

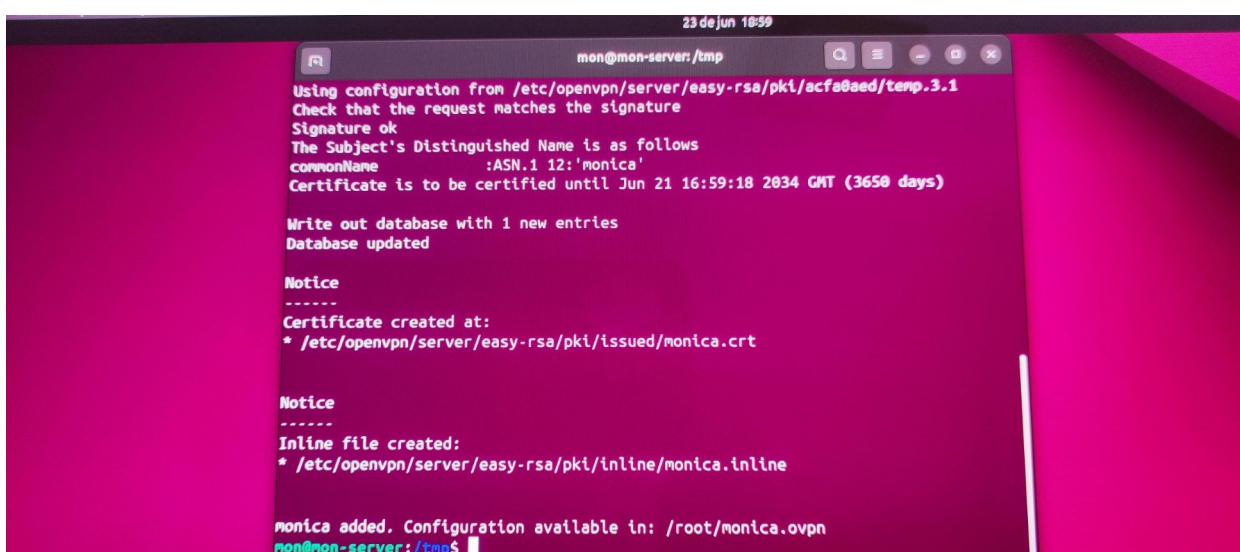
```
sudo ./openvpn-install.sh
```

1. Nos preguntará la dirección pública o hostname (dejamos la IP de nuestro router)
1. Nos preguntará por el protocolo, Protocolo UDP recomendado
1. Nos preguntará por el puerto, dejamos el puerto por defecto.
1. Nos preguntará por la DNS a utilizar dentro de la red, dejamos google por defecto.
1. Nos preguntará por el nombre del cliente, damos un nombre al cliente.
1. Se inicia la instalación. Una vez terminado ya está instalado.
1. Nos informa que la configuración se encuentra en /root/default.ovpn



Si ejecutamos de nuevo **openvpn-install.sh** nos da la opción de crear un nuevo cliente.

```
sudo ./openvpn-install.sh
```



Una vez realizado, podremos coger ese fichero y compartirlo para conectarnos directamente a nuestra máquina.

## **IMPLEMENTAR UN SISTEMA DE MONITOREO PARA LA INFRAESTRUCTURA TI**

**Nagios** es un software de monitorización, diagnóstico y gestión de redes de código abierto. Se trata de una herramienta ampliamente utilizada para vigilar equipos (hardware) y servicios (software), enviando una alerta cuando el comportamiento de los mismos no sea el deseado.

### **INSTALACIÓN DE NAGIOS**

1. Instalamos un compilador de C .

**sudo apt install -y build-essential**

2. Instalamos las dependencias necesarias para compilar e instalar Nagios

**sudo apt install wget unzip vim curl gcc openssl build-essential libgd-dev libssl-dev libapache2-mod-php php-gd php apache2**

3. Descargamos Nagios Core (podemos comprobar la última versión en

[https://www.nagios.org/downloads/nagios-core/thanks/?  
skip=1&product\\_download=nagioscore-source](https://www.nagios.org/downloads/nagios-core/thanks/?skip=1&product_download=nagioscore-source)

```
cd /tmp  
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.3.tar.gz
```

4. Extraemos el archivo descargado

**tar -zxvf nagios-4.5.3.tar.gz**

5. Configuramos Nagios

```
cd nagios-4.5.3  
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

6. Una vez termine el paso anterior compilamos Nagios

**sudo make all**

7. Creamos Usuario y Grupo para Nagios, incluimos este usuario en el grupo de Apache www-data

**sudo make install-groups-users**

**sudo groupadd -r nagios**

**sudo adduser -g nagios**

**sudo usermod -aG nagios www-data**

8. Instalamos los binarios, config y los scripts de inicio

**sudo make install**

**sudo make install-daemoninit**

**sudo make install-commandmode**

**sudo make install-config**

**sudo make install-webconf**

9. Configuramos el Servidor Web para Nagios.

**sudo a2enmod rewrite**

**sudo a2enmod cgi**

10. Configuramos el usuario para la interfaz web

**sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin**

Hay que introducir una contraseña para el usuario **nagiosadmin**. Si no existe la carpeta nagios/etc, hay que crearla.

11. Habilitamos el servicio Nagios y comprobamos el estado del servicio, debe estar levantado.

**sudo systemctl enable --now nagios**

**sudo systemctl status nagios**

12. Habilitamos Apache en el firewall y lo habilitamos o recargamos en caso de que no lo tengamos.

**sudo ufw allow apache**

**sudo ufw enable**

**sudo ufw reload**

13. Reiniciar Apache

**sudo systemctl restart apache2**

14. En este momento ya podremos ingresar a Nagios desde <http://localhost/nagios/>

16. Ahora hay que instalar los plugins de Nagios y NRPE tanto en servidor como en clientes. Podemos comprobar la ultima versión en <https://github.com/nagios-plugins/nagios-plugins/releases>

17. Primero instalamos las dependencias necesarias.

```
sudo apt install -y fping libcrypt-x509-perl libdatETIME-format-dateparse-perl libdbi-dev libkrb5-dev libldap2-dev libmysqlclient-dev libnet-snmp-perl libssl-dev libtext-glob-perl libwww-perl postgresql-server-dev-15 qstat rpcbind smbclient snmp
```

18. Después descargamos, descomprimimos y compilamos la última versión de plugin disponible.

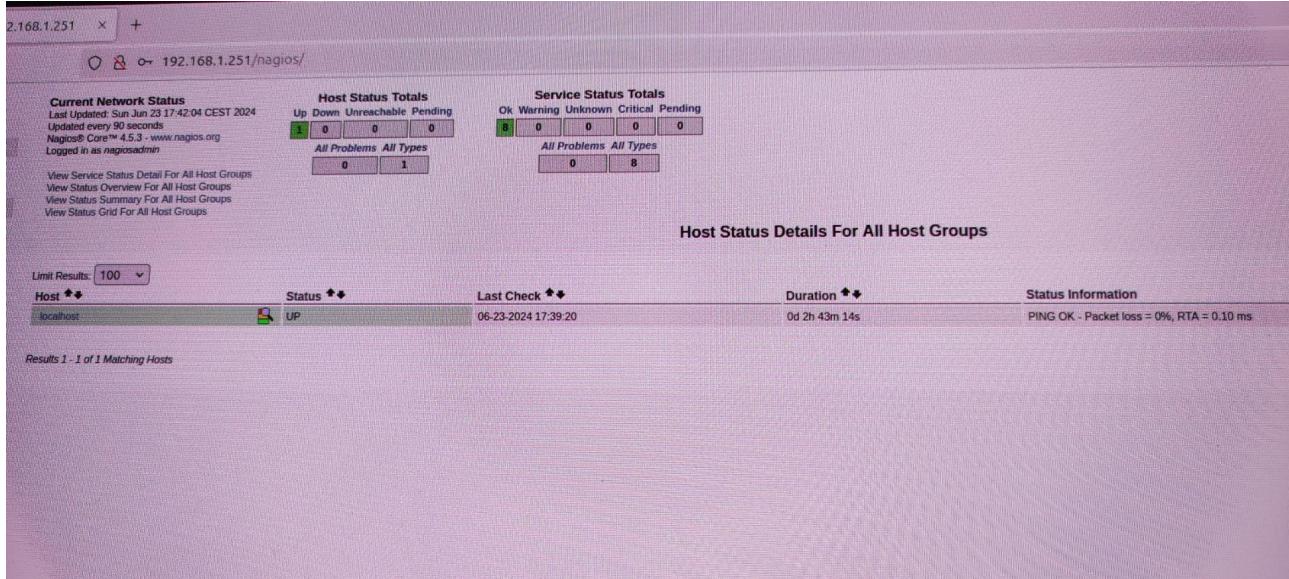
```
cd /tmp
wget http://nagios-plugins.org/download/nagios-plugins-2.4.10.tar.gz
tar xzf nagios-plugins-2.4.10.tar.gz
cd nagios-plugins-2.4.10
sudo ./configure
sudo make
sudo make install
```

19. Una vez finalice ya podremos acceder a los binarios que están en /usr/local/nagios/libexec.

The screenshot shows the Nagios web interface at the URL <http://192.168.1.251/nagios/>. The top navigation bar includes links for 'View History For All hosts', 'View Notifications For All Hosts', and 'View Host Status Detail For All Hosts'. Below the navigation, there are two summary boxes: 'Current Network Status' and 'Service Status Totals'. The 'Current Network Status' box shows 'Last Updated: Sun Jun 23 17:41:02 CEST 2024' and 'Updated every 90 seconds'. The 'Service Status Totals' box shows counts for Ok, Warning, Unknown, Critical, and Pending states across All Problems and All Types. The main content area is titled 'Service Status Details For All Hosts' and displays a table of services for the host 'localhost'. The table columns are: Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The table shows the following data:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	06-23-2024 17:39:20	0d 2h 41m 36s	1/4	OK - load average: 0.13, 0.06, 0.06
localhost	Current Users	OK	06-23-2024 17:39:20	0d 2h 40m 58s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	OK	06-23-2024 17:39:20	0d 2h 40m 21s	1/4	HTTP OK: HTTP/1.1 200 OK - 446 bytes in 0.001 second response time
localhost	PING	OK	06-23-2024 17:39:20	0d 2h 39m 43s	1/4	PING OK - Packet loss = 0%, RTA = 0.12 ms
localhost	Root Partition	OK	06-23-2024 17:39:20	0d 2h 39m 6s	1/4	DISK OK - free space: / 13516 MB (56.96% inode=88%)
localhost	SSH	OK	06-23-2024 17:39:20	0d 2h 38m 28s	1/4	SSH OK - OpenSSH_9_6p1 Ubuntu-3ubuntu13 (protocol 2.0)
localhost	Swap Usage	OK	06-23-2024 17:39:20	0d 2h 37m 51s	1/4	SWAP OK - 92% free (2705 MB out of 2971 MB)
localhost	Total Processes	OK	06-23-2024 17:39:20	0d 2h 37m 13s	1/4	PROCS OK: 56 processes with STATE = R/SZDT

At the bottom left, it says 'Results 1 - 8 of 8 Matching Services'.



20. Llegados a este punto debemos instalar NRPE tanto en servidor como clientes. Si no vamos a monitorizar servicios concretos, con instalarlo en servidor es suficiente.

21. Comprobamos la última versión de NRPE en <https://github.com/NagiosEnterprises/nrpe/releases>. Una vez la sabemos podemos descargar e instalarlo.

```
cd /tmp
wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.1.0/nrpe-4.1.0.tar.gz
tar xvf nrpe-4.1.0.tar.gz
cd nrpe-nrpe-4.1.0
sudo apt install -y libwrap0-dev
sudo ./configure
sudo make all
sudo make install
sudo make install-config
sudo make install-inetd
```

22. Arrancamos el servicio NRPE y reiniciamos xinetd.

```
sudo systemctl enable nrpe
sudo systemctl start nrpe
sudo systemctl restart xinetd
```

En caso de que no se instale xinetd, deberemos instalarlo manualmente.

```
sudo apt install xinetd
```

23. Se repiten los pasos 15 a 22 para cada una de las máquinas (clientes o servidor)

24. Una vez instalado NRPE en cada cliente, se deberá habilitar el servicio. Para ello debemos editar el siguiente fichero

**sudo nano /usr/local/nagios/etc/nrpe.cfg**

25. Debemos localizar la directiva *allowed\_host* en la que se debe establecer la IP de la máquina a la que queremos permitir la conexión (es decir, el servidor donde esté instalado nagios). Deberá quedar de la siguiente manera

**allowed\_hosts= #(ip del servidor)**

26. Para que Nagios pueda hacer uso del binario nrpe, se deberá definir en este fichero. Localizamos donde se realizan las definiciones de comandos y añadimos

```
define command {
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

En este mismo fichero podemos encontrar los comandos que la máquina cliente aceptará por parte de Nagios.

```
# The following examples use hardcoded command arguments...
# This is by far the most secure method of using NRPE
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_load -r -w .15,.10,.05 -
c .30,.25,.20
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -
p /dev/hda1
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c
10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c
200
```

27. Una vez realizados todos estos ajustes, arrancamos el servicio

**sudo systemctl start nrpe.service**

28. Accedemos al directorio de Nagios y creamos un directorio nuevo para meter los ficheros de configuración de linux.

```
cd /usr/local/nagios/etc
sudo mkdir -p /usr/local/nagios/etc/servers
```

29. Accedemos a la configuración de nagios dentro de /usr/local/nagios/etc

**sudo nano nagios.cfg**

30. Descomentamos esta línea para poder alojar en la carpeta servers que hemos creado el fichero para configurar hosts.

**cfg\_dir=/usr/local/nagios/etc/servers**

31. Creamos un fichero *linux.cfg* que será el que aloje los hosts.

**sudo nano /usr/local/nagios/etc/objects/linux.cfg**

32. Agregamos los hosts al fichero

```
define host {
    use      linux-server      # tipo de host
    host_name  ubuntuapache   # nombre del host para identificarlo
    alias     Ubuntu Apache    # alias que daremos al host
    address   192.168.1.91    # IP del host a supervisar
    check_interval 1          # Tiempo de check
}
```

```
define host {
use      linux-server
host_name  ubuntuarchivos
alias     Ubuntu Archivos
address   192.168.1.251
}
```

## 5.CONFIGURAR UN SERVIDOR DE BACKUP AUTOMATIZADO

**Bacula Backup** es una solución TI de respaldo y recuperación bastante atractiva para las Pymes y grandes empresas ya que la pérdida de datos puede cambiar su posicionamiento.

Consta de una colección de herramientas para cubrir las necesidades de respaldo de equipos en redes IP.

```
apt install bacula-console-qt
```

Puede instalarlo en una máquina remota desde aquella donde se ejecuta Bacula Director.  
En este caso, debe cambiar la configuración para conectarse al Director.

```
Director {  
    Name = Bacula-Director  
    DIRport = 9101  
    address = 10.5.5.5  
    Password = "password"  
}
```

Si lo instaló en la misma computadora donde se ejecuta Bacula Director, no necesita hacer esto. Todo lo que necesita hacer es verificar si el parámetros en el **/etc/bacula/bat.conf** archivo coincide con el parámetros en el **/etc/bacula/bacula-dir.conf** archivo.

## Lanzamiento de la herramienta de administración de Bacula

```
sudo usermod -a -G bacula user
```

Cerrar sesión y volver a iniciar sesión como este usuario usando SSH. Si inicia sesión como solo a través de xRDP, entonces tienes que hacerlo a través de SSH. Por ejemplo, después de iniciar sesión a través de SSH como usuario raíz

```
su - user  
exit
```

Ahora inicie sesión como este usuario normal a través de xRDP e inicie la aplicación.

```
bat
```

O puede encontrar la herramienta de administración de Bacula en la lista de aplicaciones instaladas y ejecutarla.

#### WEBGRAFIA

[www.virtualbox.or](http://www.virtualbox.org)

[www.ubuntu.com](http://www.ubuntu.com)

[www.digitalocean.com](http://www.digitalocean.com)

[www.solvetic.com](http://www.solvetic.com)

[www.serverspace.io](http://www.serverspace.io)

Canal de youtube:

RedesPlus

El inicio no ha sido nada fácil ya que he tenido todo en contra con el PC y las VMs pero estoy bastante satisfecha con el resultado final, jamás pensé poder llegar a crear algo así y aunque no está terminado ni funciona como debería el proceso ha sido todo un reto, desde la recopilación de información por tantas vías siendo unas más fiables y actualizadas que otras hasta la puesta en marcha en los diferentes servidores, es un ENSAYO/ERROR constante.

**GRACIAS ROGER!!**