

Indice

- Indice
- Objetivo
- Maquinas virtuales
 - Creación de máquina virtual
 - Configuración básica Ubuntu
 - Establecer IP estatica
 - Configurar SSH
 - Configurar FTP
- Apache
 - Instalar Apache2
 - Editar archivo host
 - Añadir certificado SSL
 - Crear web
 - Configuracion web
- Samba
 - Instalar samba
 - Acceso en windows
 - Acceso en Ubuntu
- DuckDNS
 - Creación de dominio
 - Crontab
- mySQL
 - Instalar mySQL
 - Configurar autenticacion y privilegios de usuario
 - Configuración para conexión remota
- PHP
 - Instalar PHP
 - Conectar con mySQL
- Nagios
 - Instalación
 - Configuración en apache
 - Acceso a nagios
 - Configurar autenticación
- OVPN
 - Instalación
 - Configuración manual
 - Generar certificado cliente
 - Archivo de configuracion de OVPN
- Rsync
 - Instalacion rsync
 - Programar copias de seguridad

Objetivo

El proyecto consiste en levantar 4 VMs con los siguientes servicios:

ubuntu1: Servidor apache2 para servir una página web expuesta de forma pública, con un dominio y certificado SSL. Samba para compartir archivos con las demás máquinas.

ubuntu2: Servidor mySQL debe poder ser accedido por apache2, mostrando una query ejecutada correctamente en el navegador.

ubuntu3: Servidor Nagios, monitorizando el estado de las demás máquinas.

ubuntu4: Servidor de backups con bacula, openVPM para acceder a la red privada.

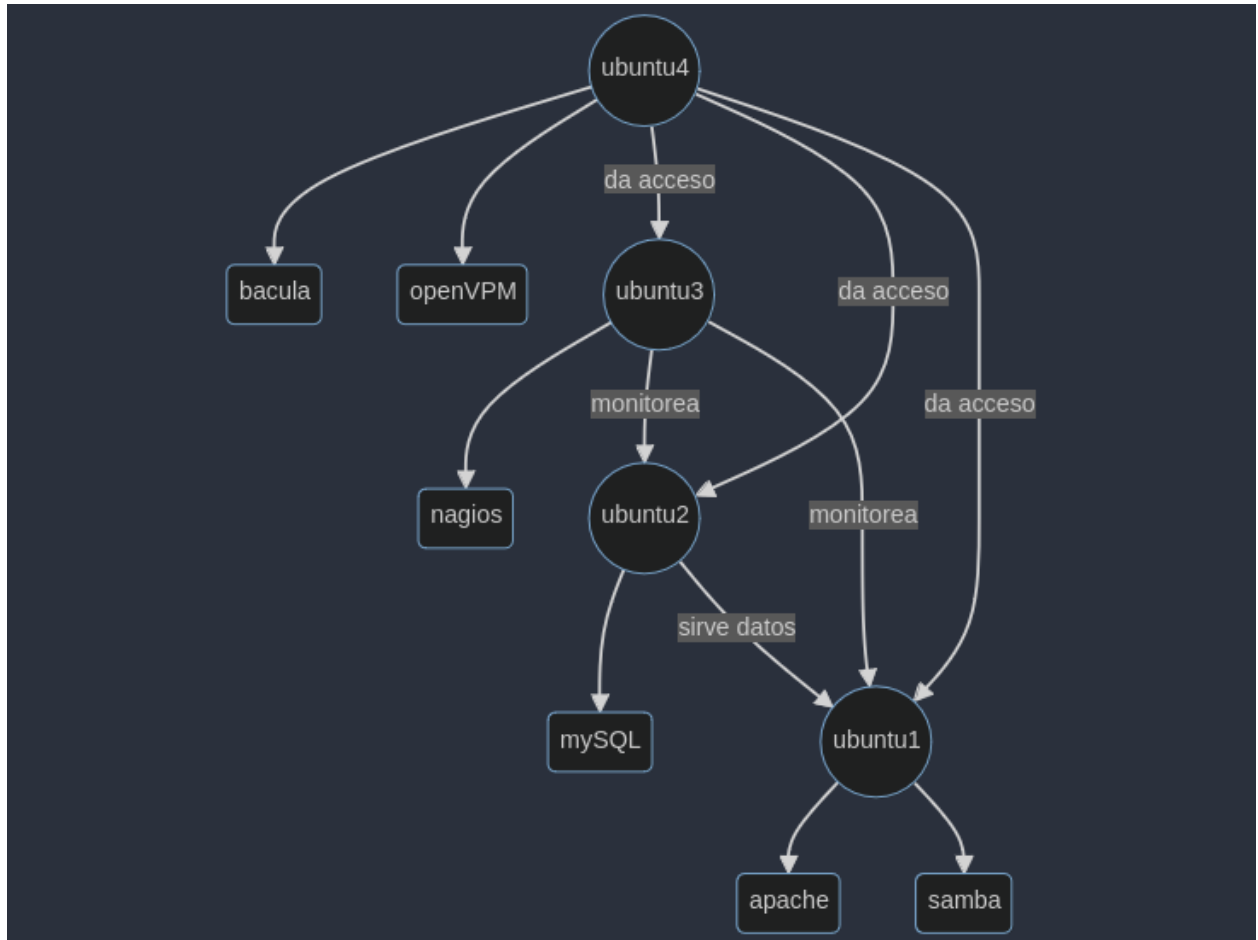


Figure 1: vms

Maquinas virtuales

Creación de máquina virtual

Descarga la imagen de ubuntu en la pagina oficial, en este ejemplo utilizaremos la version 23.10.1 Descarga VirtualBox de la página oficial, instalalo y crea una nueva maquina virtual con la imagen que descargaste. Debemos asegurarnos de varias cosas en la configuración:

- En Settings > Network debemos usar el “Adaptador puente” como adaptador de red
- Debemos dar al menos 4 nucleos, de lo contrario el proceso puede ser muy largo
- Daremos al menos 20GB de disco duro

Una vez Ubuntu está correctamente instalado, procedemos a su configuración

Configuración básica Ubuntu

Antes de empezar, actualiza tu sistema con

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

Instala net-tools

```
sudo apt install net-tools
```

Establecer IP estatica

Para evitar que nuestra maquina cambie de IP, seguimos los siguientes pasos

- Vamos a Settings > Network
- En la seccion “Wired” hacemos click en el icono de configuracion.
- En la pestaña IPv4, seleccionamos el método “Manual”.
- En la seccion “Address” puedes configurar tu dirección IP, asegúrate de no tener otro dispositivo con la misma IP en tu red
- Haz `ifconfig` o `ipconfig` segun tu sistema para obtener tu mascara de red y puerta de enlace predeterminada.
- En la sección “Netmask” introduce tu máscara de red.
- En la sección “Gateway”, introduce tu puerta de enlace, generalmente 192.168.0.1.
- Selecciona el modo manual en la sección “DNS”, introduce 1.1.1.1, 8.8.8.8
- Comprueba que tienes conexión a internet haciendo ping.

Configurar SSH

Vamos a configurar un servidor SSH que nos permita conectarnos a la maquina de forma remota:

Actualiza los repositorios de tu sistema

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

Descarga OpenSSH

```
sudo apt install openssh-server
```

Comprueba el estado del servicio

```
sudo systemctl status ssh
```

Inicia el servicio ssh

```
sudo systemctl start ssh
```

Para hacer que el servicio se inicie siempre al arrancar la maquina:

```
sudo systemctl enable ssh
```

Para entrar desde otro dispositivo, hazlo con

```
ssh usuario@ip
```

Configurar FTP

Vamos a configurar un servidor FTP que nos deje enviar ficheros entre maquinas

Descarga el servidor vsftpd

```
sudo apt install vsftpd
```

Configura el servidor

```
sudo nano /etc/vsftpd.conf
```

Asegurate de cambiar los siguientes parametros:

```
local_enable=YES
```

```
write_enable=YES
```

```
connect_from_port_20=YES
```

Para comprobar el estado del servicio

```
sudo systemctl status vsftpd
```

Para iniciar el servicio y hacer que se lance al iniciar la maquina:

```
sudo systemctl start vsftpd
```

```
sudo systemctl enable vsftpd
```

Para utilizarlo necesitarás usar un cliente FTP como FileZilla, indicar tu nombre de usuario, IP y contraseña.

Tambien puedes utilizar `ftp usuario@host` en la terminal.

Apache

Instalar Apache2

Apache2 se encarga de servir nuestros documentos HTML, para instalarlo y configurarlo haremos lo siguiente:

Instalamos las dependencias con

```
sudo apt install apache2
```

Para comprobar el estado del servidor

```
sudo systemctl status apache2
```

Abre los puertos necesarios:

```
sudo ufw allow apache
```

Comprueba que tu página web está operativa entrando en tu ip desde el navegador o en localhost

Para editar la página, puedes editar el archivo `index.html` situado en `/var/www/html`

Editar archivo host

Si quieres poder acceder a la web desde un dominio en lugar de su IP, deberás editar el archivo de host de tu sistema. En Ubuntu, el archivo está en `/etc/hosts`, puedes editarlo mediante `sudo nano /etc/hosts`. En Windows, el archivo está en `C:\Windows\System32\drivers\etc\hosts`, puedes editarlo en el bloc de notas, aunque tendrás que editarlo fuera de la carpeta original. Deberás editar el archivo hosts siguiendo el siguiente ejemplo:

```
192.168.0.30    nombre-de-tu-web.com
```

Añadir certificado SSL

Para poder añadir un certificado SSL a nuestra web podemos utilizar certbot.

Para instalar certbot y el plugin de apache:

```
sudo apt install certbot python3-certbot-apache
```

Permite que el firewall habilite el HTTPS

```
sudo ufw allow https
```

Obtener el certificado es fácil, ejecuta

```
sudo certbot --apache
```

El asistente de instalación te ayudará a configurar el certificado, deberás introducir tu correo electrónico y aceptar los términos de uso.

Automatiza la renovación del certificado

```
sudo crontab -e
```

Añade la siguiente línea:

```
0 */12 * * * /usr/bin/certbot renew --quiet
```

Crear web

Crea una carpeta para la página y da permisos de usuario:

```
sudo mkdir /var/www/nombre-de-tu-web
sudo chown -R $USER:$USER /var/www/nombre-de-tu-web/
sudo chmod -R 755 /var/www/nombre-de-tu-web/
```

Ya puedes crear archivos y servirlos:

```
nano /var/www/nombre-de-tu-web/index.html
```

Configuración web

Vamos a editar el siguiente archivo de configuración:

```
sudo nano /etc/apache2/sites-available/nombre-de-tu-web.conf
```

Una configuración básica sería esta:

```
<VirtualHost *:80>
    ServerAdmin webmaster@nombre-de-tu-web.com
    ServerName nombre-de-tu-web.com
    ServerAlias www.nombre-de-tu-web.com
    DocumentRoot /var/www/nombre-de-tu-web
    ErrorLog ${APACHE_LOG_DIR}/nombre-de-tu-web_error.log
    CustomLog ${APACHE_LOG_DIR}/nombre-de-tu-web_access.log combined
</VirtualHost>
```

Aplica los cambios y reinicia el servidor:

```
sudo a2ensite nombre-de-tu-web.conf
sudo systemctl restart apache2
```

Si aun no tienes expuesto el puerto para poder usar la ip publica, puedes editar tu archivo `hosts` para usarla en la red privada:

```
sudo nano /etc/hosts
```

Añade tu ip con el nombre de dominio que quieras usar:

```
<ip>    nombre-de-tu-web.com
```

Samba

Instalar samba

```
sudo apt install samba
```

Edita la configuración:

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

Para añadir una carpeta como recurso a compartir:

```
[shared]
    path = /var/www/html
    available = yes
    valid users = usuario
    read only = no
    browsable = yes
    public = yes
    writable = yes
```

Asigna permisos:

```
sudo chown -R usuario:usuario /home/usuario/shared
```

Crea tu usuario, te preguntará que contraseña quieres usar:

```
sudo smbpasswd -a usuario
```

Dar permisos al firewall:

```
sudo ufw allow samba
sudo ufw reload
```

Reinicia samba:

```
sudo systemctl restart smbd
```

Para comprobar que la configuración es correcta:

```
testparm
```

Acceso en windows

Para acceder desde windows: En el explorador de archivos aparecerá en la sección “Red”, si no sale, puedes hacer click en la barra superior y acceder a `\\<IP>\nombre_carpeta_compartida`

Acceso en Ubuntu

Para acceder desde Ubuntu: Abre Nautilus, tu explorador de archivos. Ve a “Otras ubicaciones” En la barra inferior, entra en `smb://<IP>/recurso_compartido`

En ambos te pedirán el nombre de usuario y contraseña para acceder.

DuckDNS

Creación de dominio

Inicia sesión en duckDNS, esto te dará acceso a tu token de identificación. Añade un dominio de duckDNS, debería detectar automáticamente la ip actual si tienes un servidor corriendo, en caso contrario, ingresa tu ip actual.

Crontab

DuckDNS utiliza crontab para actualizar la ip de nuestro servidor, tambien necesitarás curl, en caso de no tenerlos:

Crea una carpeta, crea un archivo `duck.sh` en su interior:

```
echo url="https://www.duckdns.org/update?domains=<DOMINIO>&token=<TOKEN>&ip=" | curl -k -o ~/duckdns/du
```

Sustituye dominio y token por los que tengas en duckDNS.

Otorga permisos de ejecucion:

```
chmod 700 duck.sh
```

Vamos a usar crontab para ejecutar el script cada 5 minutos, usa `crontab -e` para ejecutar crontab, elige un editor de texto y pega la siguiente configuracion:

```
*/5 * * * * ~/duckdns/duck.sh >/dev/null 2>&1
```

Puedes guardar el cron, y comprobar que el script funciona con `./duck.sh`, devuelve OK en caso positivo, y KO en caso negativo.

mySQL

Instalar mySQL

Instala el servidor mySQL

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

Ejecuta el script de seguridad para completar la instalación:

```
sudo mysql_secure_installation
```

Esto iniciara el proceso de configuracion para segurar ciertos aspectos de mysql: - **Validacion de contraseñas:** permite establecer una validacion de contraseñas para usuarios de la base datos, estableciendo varios niveles de seguridad.

- **Eliminar usuarios anonimos**, solo deberian usarse durante el desarrollo, pero nunca en produccion.
- **Desactivar acceso a root remoto**, no permitirá que nos podamos conectar directamente al servidor mysql, deberemos hacerlo desde localhost siempre.
- **Borrar** base de datos de test.
- **Recargar privilegios** para que los cambios se vean reflejados.

Una vez finalizado mySQL está listo para lanzarlo con `sudo mysql`.

Configurar autenticacion y privilegios de usuario

Podemos ver la configuración actual de permisos de los usuarios:

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

Y editarla mediante:

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

Por ultimo, no te olvides de volver a cargar la tabla de permisos de usuario:

```
FLUSH PRIVILEGES;
```

Podemos iniciar sesion sql usando `sudo mysql -p` si queremos acceder como root o `sudo mysql -u username -p` para usar un usuario.

Crear un usuario:

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
```

Dar privilegios, de lo contrario no podremos crear o editar tablas:

```
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost' WITH GRANT OPTION;
```

Una vez has entrado con tu usuario o root, puedes interactuar con el servidor mysql:

status: muestra información sobre la sesión de mysql y el servidor. **source:** ejecuta un archivo sql. **create database <nombre>:** crea una base de datos. **use <database>:** accede a una base de datos.

Puedes ejecutar cualquier sentencia SQL válida desde el CLI.

Configuración para conexión remota

Si queremos poder acceder a la base de datos desde la maquina apache debemos darle acceso a la misma, edita el archivo de configuración:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Añade la IP de la máquina a la que quieras permitir el acceso o utiliza 0.0.0.0 para permitir cualquiera:

```
bind-address = 0.0.0.0
```

Reinicia el servicio:

```
sudo systemctl restart mysql
```

En tu máquina con apache, instala el cliente de mySQL:

```
sudo apt-get install mysql-client
```

Habilita el puerto 3306 para permitir la conexion con mySQL.

```
sudo ufw allow 3306
```

PHP

Instalar PHP

Instala php y la libreria para apache:

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

Reinicia el servicio y configuralo para iniciarse automaticamente:

```
sudo systemctl restart apache2
```

```
sudo systemctl enable apache2
```

Para comprobar que todo va correctamente:

```
sudo nano /var/www/html/info.php
```

Escribimos lo siguiente:


```
<?php
phpinfo();
?>
```

Si todo ha ido bien, al acceder a `<ip>/info.php` mediante el navegador deberías ver los datos de tu instalación de php.

Conectar con mySQL

Asegúrate de que ambas maquinas pueden hacerse ping.

Instala las librerías y dependencias necesarias:

```
sudo apt-get install mysql-client
```

Asegúrate de abrir el puerto 3306 para el servidor mySQL:

```
ufw allow 3306
```

Prueba la conexión con mySQL:

```
mysql -u usuario -h ip_del_servidor_mysql -p
```

Si obtienes un error, asegúrate de seguir los pasos para autenticar y dar privilegios a tu usuario.

Si ya tienes un usuario con los permisos necesarios y una base de datos con una tabla y datos, puedes crear el siguiente script para probar la conexión mediante php:

```
<?php
$user = "username";
$password = "password";
$database = "test123";
$table = "usuarios";
$host = "IP con el servidor mySQL";
try {
    $db = new PDO("mysql:host=$host;dbname=$database", $user, $password);
    echo "<h2>Usuarios</h2><ol>";
    foreach($db->query("SELECT nombre FROM $table") as $row) {
        echo "<li>" . $row['nombre'] . "</li>";
    }
    echo "</ol>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```

El anterior script lee y muestra los nombres de una tabla creada mediante la siguiente sentencia SQL:

```
CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    contraseña VARCHAR(255) NOT NULL,
    puntos INT DEFAULT 0
);
INSERT INTO usuarios (nombre, contraseña, puntos) VALUES
('admin', '12341234', 100),
('usuario2', 'contraseña2', 150),
('usuario3', 'contraseña3', 200),
```

```
('usuario4', 'contraseña4', 250),  
( 'usuario5', 'contraseña5', 300);
```

Nagios

Instalación

Instalamos los paquetes necesarios: `sudo apt install -y build-essential libgd-dev libssl-dev apache2 php libapache2-mod-php libperl-dev libpq-dev libmysqlclient-dev`

Creamos un usuario y grupo “nagios”, añade un usuario adicional “nagcmd” para comandos externos:

```
sudo useradd nagios  
sudo groupadd nagcmd  
sudo usermod -a -G nagcmd nagios  
sudo usermod -a -G nagcmd www-data
```

Descarga nagios core:

```
cd /tmp  
wget https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.4.9/nagios-4.4.9.tar.gz  
tar -xzf nagios-4.4.9.tar.gz  
cd nagios-4.4.9
```

Compila e instala:

```
./configure --with-command-group=nagcmd  
make all  
sudo make install  
sudo make install-commandmode  
sudo make install-init  
sudo make install-config  
sudo make install-webconf
```

Configura tu correo electronico para las notificaciones:

```
sudo nano /usr/local/nagios/etc/objects/contacts.cfg
```

Añade tu email en la linea del usuario “nagiosadmin”.

Configuración en apache

Ve a tu maquina con apache, reinicia el servicio:

```
sudo a2enmod rewrite  
sudo a2enmod cgi  
sudo systemctl restart apache2
```

Instala los plugins de nagios:

```
cd /tmp  
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz  
tar -xzf nagios-plugins-2.3.3.tar.gz  
cd nagios-plugins-2.3.3  
./configure --with-nagios-user=nagios --with-nagios-group=nagios  
make  
sudo make install
```

Inicia el sistema y configuralo para arrancarlo automaticamente con la maquina:

```
sudo systemctl enable nagios  
sudo systemctl start nagios
```

Acceso a nagios

En tu navegador, entra en `http://<ip_servidor>/nagios`

Configurar autenticación

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

OVPN

Instalación

Podemos usar este script para generar un servidor OpenVPN.

```
wget https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh
chmod +x openvpn-install.sh
sudo ./openvpn-install.sh
```

El script te guiará en la instalación, te preguntará por el puerto, protocolo, DNS y nombre de usuario.

Configuración manual

Si prefieres crear manualmente el servidor, sigue estos pasos:

En la maquina donde irá alojado el servidor:

```
sudo apt install openvpn easy-rsa
```

Crea un directorio para easy-rsa:

```
make-cadir ~/openvpn-ca
cd ~/openvpn-ca
```

El archivo de configuración se encuentra en `~/openvpn-ca/vars`

Inicializa el PKI:

```
./easyrsa init-pki
```

Construir la CA:

```
./easyrsa build-ca
```

Te preguntará una contraseña y usuario.

Generar certificados y claves:

```
./easyrsa gen-req server nopass
./easyrsa sign-req server server
```

Te preguntará un usuario y contraseña

Genera el parametro Diffie-Hellman:

```
./easyrsa gen-dh
```

Genera el archivo HMAC

```
openvpn --genkey --secret pki/ta.key
```

Generar certificado cliente

Para generar el certificado de un cliente:

```
./easyrsa gen-req keepcoding nopass
./easyrsa sign-req client keepcoding
```

Te preguntará por usuario y contraseña para crear y firmarlo.

Archivo de configuracion de OVPN

Para crear una configuración solo necesitamos crear un archivo .ovpn:

```
nano keepcoding.ovpn
```

Esta plantilla sirve para configurarlo:

```
client
dev tun
proto udp
remote <IP de tu servidor> 1194
resolv-retry infinite
nobind
user nobody
group nogroup
persist-key
persist-tun
remote-cert-tls server
cipher AES-256-CBC
auth SHA256
key-direction 1
verb 3

<ca>
-----BEGIN CERTIFICATE-----
(contenido de ca.crt)
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
(contenido de keepcoding.crt)
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN PRIVATE KEY-----
(contenido de keepcoding.key)
-----END PRIVATE KEY-----
</key>
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
(contenido de ta.key)
-----END OpenVPN Static key V1-----
</tls-auth>
```

Las rutas a los archivos son:

- ~/openvpn-ca/pki/ca.crt
- ~/openvpn-ca/pki/issued/keepcoding.crt
- ~/openvpn-ca/pki/private/keepcoding.key
- ~/openvpn-ca/pki/ta.key

Guarda el archivo, si todo esta correcto, puedes usarlo con un cliente de OpenVPN.

Rsync

Instalacion rsync

Instalamos los paquetes necesarios:

```
sudo apt-get install rsync
```

Crea el siguiente script:

```
#!/bin/bash

# Configuración de variables
ORIGEN="usuario@<IP origen>:/var/www/"
DESTINO="/backup/www/"
LOGFILE="/var/log/rsync-backup.log"

# Ejecutar rsync
rsync -avz $ORIGEN $DESTINO >> $LOGFILE 2>&1

# Registrar la fecha y hora de la copia de seguridad
echo "Backup realizado en: $(date)" >> $LOGFILE
```

Guarda el script en /usr/local/bin/rsync-backup.sh y dale permisos de ejecución:

```
sudo chmod +x /usr/local/bin/rsync-backup.sh
```

Crea un archivo de log en /var/log/rsync-backup.log:

```
sudo touch /var/log/rsync-backup.log
```

Prueba a ejecutar el script:

```
sudo /usr/local/bin/rsync-backup.sh
```

Programar copias de seguridad

Vamos a editar crontab:

```
crontab -e
```

Añade la siguiente línea al final del archivo:

```
0 3 * * * root /usr/local/bin/rsync-backup.sh
```

Guarda los cambios y cierra el editor. Con esta configuración, la copia de seguridad se realizará todos los días a las 3:00 AM.

Los datos en crontab se dividen en 6 campos:

1. Minuto (0-59)
2. Hora (0-23)
3. Día del mes (1-31)
4. Mes (1-12)
5. Día de la semana (0-7, donde 0 y 7 son domingo)
6. Comando a ejecutar