

# CALANET



# CALANET

**Marcos garrido**  
**Saúl Pérez**  
**Proyecto Asic2**



**CALANET**

**Cicle Formatiu de Grau Superior d'**

**Administració de Sistemes Informàtics i Ciberseguretat**

## **MEMÒRIA DEL PROJECTE**

**TÍTOL:** Calanet

**AUTOR:** Marcos Garrido y Saul Perez

### **QUALIFICACIÓ**

**Projecte defensat a la convocatòria del dia:**        /        /        .

**Ha obtingut la qualificació de**

**TUTOR/A**

**2on de ASIC**

**TRIBUNAL**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

<b>Explicación de la idea.....</b>	<b>3</b>
<b>Explanation of the idea.....</b>	<b>4</b>
<b>Infraestructura corporativa.....</b>	<b>5</b>
<b>1. Distribución de servidores de la red.....</b>	<b>6</b>
<b>2. Configuración del OpenWRT.....</b>	<b>6</b>
<b>3. Instalación del servidor DNS.....</b>	<b>12</b>
<b>4. Instalación del servidor FTP.....</b>	<b>14</b>
<b>5. Instalación de ldap.....</b>	<b>17</b>
<b>6. Instalación del NFS.....</b>	<b>22</b>
<b>7. Instalación del servidor Base de datos(MySql).....</b>	<b>25</b>
<b>8. Instalación del servidor web(Wordpress).....</b>	<b>28</b>
<b>9. Instalación del servidor de correo.....</b>	<b>32</b>
a. Configuración del MTA (Postfix).....	32
b. Configuración del MDA (Dovecot).....	38
c. Instalación de ThunderBird.....	39
d. Automatización de la instalación del servidor de correo:.....	40
<b>12. Instalación del servidor de monitorización(Zabbix).....</b>	<b>44</b>
12.1 Añadir host en zabbix.....	50
<b>13. Instalación del proxy inverso.....</b>	<b>52</b>
<b>14. Herramientas para aumentar medidas de seguridad.....</b>	<b>55</b>
<b>15. Anexo.....</b>	<b>57</b>
Instalación de Servidor openvpn:.....	57
Instalación de easy-rsa.....	66
Instalación de la máquina bastión.....	69
Instalación de Samba DC.....	72
<b>16. Conclusión.....</b>	<b>81</b>
<b>17. Bibliografía.....</b>	<b>82</b>

## Explicación de la idea.

- Personalización de soluciones: Ofrecer soluciones personalizadas de infraestructura tecnológica para satisfacer las necesidades específicas de cada cliente. Esto podría incluir la creación de plantillas de infraestructura que se adapten a las necesidades de cada empresa, así como la posibilidad de elegir entre diferentes tipos de infraestructuras, como servidores, almacenamiento, redes, seguridad, etc.
- Automatización y optimización: Desarrollar scripts que permitan automatizar y optimizar el proceso de instalación y configuración de la infraestructura tecnológica. Esto ayudaría a reducir costos y mejorar la eficiencia de la instalación y el mantenimiento de la infraestructura.
- Servicios de mantenimiento y soporte: Ofrecer servicios de mantenimiento y soporte técnico para aquellos clientes que no cuentan con un equipo de informáticos propios. Esto podría incluir la gestión de actualizaciones de seguridad, respaldo de datos, monitoreo de sistemas, entre otros servicios.
- Monitoreo y seguimiento: Implementar un sistema de monitoreo y seguimiento que permita a los clientes supervisar y controlar el rendimiento de su infraestructura tecnológica en tiempo real. Esto podría incluir la gestión de alertas y notificaciones, así como la capacidad de generar informes y análisis de rendimiento.
- Escalabilidad: Diseñar la solución de manera escalable, permitiendo a los clientes ampliar o reducir su infraestructura según sus necesidades. Esto podría incluir la posibilidad de agregar o eliminar servidores, almacenamiento, redes y otros componentes de la infraestructura según sea necesario.
- Seguridad: Garantizar la seguridad de la infraestructura tecnológica de los clientes mediante la implementación de medidas de seguridad adecuadas, como autenticación y autorización, cifrado de datos, protección contra malware y otras amenazas cibernéticas.
- Integración con otras soluciones: Permitir la integración de la infraestructura tecnológica con otras soluciones de software y herramientas de terceros, como sistemas de gestión de relaciones con clientes (CRM), plataformas de comercio electrónico, sistemas de gestión de proyectos, entre otros.
- Formación y capacitación: Ofrecer formación y capacitación a los clientes para que puedan aprovechar al máximo su infraestructura tecnológica. Esto podría incluir sesiones de capacitación en línea, tutoriales y documentación detallada.

- Flexibilidad en la facturación: Ofrecer diferentes opciones de facturación para adaptarse a las necesidades de los clientes, como facturación por uso, suscripción mensual o anual, entre otras opciones.
- Fomento de la innovación: Establecer un programa de innovación que fomente la creación de nuevas soluciones y servicios, así como la mejora continua de los servicios existentes. Esto podría incluir la creación de un laboratorio de innovación, la participación en eventos de tecnología y la colaboración con startups y universidades.

## Explanation of the idea.

- Solution customization: Offer personalized solutions technological infrastructure to meet the specific needs of each client. This could include creating infrastructure templates to suit the needs of each company, as well as the ability to choose between different types of infrastructure, such as servers, storage, networking, security, etc.
- Automation and optimization: Develop scripts that allow you to automate and optimize the installation and configuration process of the technological infrastructure. This would help reduce costs and improve the efficiency of infrastructure installation and maintenance.
- Maintenance and support services: Offer maintenance and technical support services for those clients who do not have their own computer team. This could include managing security updates, data backup, system monitoring, among other services.
- Monitoring and tracking: Implement a monitoring and tracking system that allows clients to monitor and control the performance of their technological infrastructure in real time. This could include alert and notification management, as well as the ability to generate reports and performance analysis.
- Scalability: Design the solution in a scalable manner, allowing clients to expand or reduce their infrastructure according to their needs. This could include the ability to add or remove servers, storage, networking, and other infrastructure components as needed.
- Security: Ensure the security of clients' technology infrastructure by implementing appropriate security measures, such as authentication and authorization, data encryption, protection against malware and other cyber threats.

- Integration with other solutions: Allow the integration of technological infrastructure with other software solutions and third-party tools, such as customer relationship management (CRM) systems, e-commerce platforms, project management systems, among others.
- Training and training: Provide training and training to clients so that they can make the most of their technological infrastructure. This could include online training sessions, tutorials, and detailed documentation.
- Flexibility in billing: Offer different billing options to adapt to customer needs, such as per-use billing, monthly or annual subscription, among other options.
- Promotion of innovation: Establish an innovation program that encourages the creation of new solutions and services, as well as the continuous improvement of existing services. This could include creating an innovation lab, participating in technology events, and collaborating with startups and universities.

## Infraestructura corporativa.

La infraestructura de nuestra empresa contará con tres zonas de red diferentes: DMZ, corporativa y privada.

DMZ: Esta zona desmilitarizada servirá para que los clientes puedan acceder a nuestra página web desde internet sin necesidad de acceder a un servidor de nuestra red local, esto nos lo permitirá un servidor con nginx, configurado para ser usado como reverse-proxy.

Corporativa: Esta será la red interna de la empresa que contendrá todos los servicios necesarios. En esta se dividirán los diferentes departamentos de trabajadores con vlans, para evitar que hayan muchos paquetes de broadcast. Estos son los diferentes servicios que tendremos en esta red: DNS, monitorización, ftp, nfs, ldap, web

Privada: Esta zona es la más privada de todas, con acceso restringido. La utilizaremos para poner el sql para el servidor web.

## 1. Distribución de servidores de la red.

Para la infraestructura, utilizaremos los siguientes servicios en red:

En la red DMZ, tendremos un servidor de proxy inverso que trabajara con nginx.

En la red corporativa, tendremos la mayoría de nuestros servicios, el dhcp lo llevará el router. Tendremos un servidor DNS con las direcciones de nuestros diferentes servidores para facilitar la manera de trabajar y hacerlo con nombres dominios en vez de con las ips. Un servidor FTP para repartir el material de trabajo como los scripts, manuales de información... También dispondremos de un ldap para gestionar los usuarios, y un árbol de directorios que será donde podrán trabajar los empleados, utilizando NFS. Para el servidor web utilizaremos apache y montaremos un wordpress para poder publicitar y explicar lo que hacemos en la empresa.

Para el correo utilizaremos postfix, para tener un correo dentro de la empresa. Y zabbix será la aplicación que utilizaremos para monitorizar todos los servicios de la red.

En la red privada, tendremos la máquina con sql para poder hacer el wordpress y tener la base de datos en un sitio seguro.

## 2. Configuración del OpenWRT.

### Create Interface

Name of the new interface

The allowed characters are: A-Z, a-z, 0-9 and

Note: interface name length Maximum length of the name is 15 characters in

Protocol of the new interface


Create a bridge over multiple interfaces ☐

Cover the following interface

Creamos las 3 diferentes interfaces cada una con su puerto correspondiente eth1/eth2/eth3

## Common Configuration

General Setup **Advanced Settings** Physical Settings Firewall Settings

Status  **Device:** eth1  
RX: 0 B (0 Pkts.)  
TX: 0 B (0 Pkts.)

Protocol Static address ▾

IPv4 address 192.168.10.1

IPv4 netmask 255.255.255.0 ▾


IPv4 gateway 192.168.10.1


IPv4 broadcast 192.168.10.255


Ponemos ip del router, y red las redes seran 192.168.10.0/24, 192.168.20.0/24, 192.168.30.0/24


## DHCP Server

General Setup **Advanced Settings** IPv6 Settings

Ignore interface ☐  
 Disable DHCP for this interface.

Start 5  
 Lowest leased address as offset from the network address.

Limit 150  
 Maximum number of leased addresses.

Lease time 12h  
 Expiry time of leased addresses, minimum is 2 minutes (2m).

Hacemos un DHCP server en el router para las diferentes zonas.

Name	Zone ⇒ Forwardings	Input	Output	Forward	Masquerading	MSS clamping		
lan	lan ⇒ wan	accept ▾	accept ▾	accept ▾	<input type="checkbox"/>	<input type="checkbox"/>	Edit	Delete
wan	wan ⇒ REJECT	reject ▾	accept ▾	reject ▾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit	Delete
Corp	Corp ⇒ DMZ Privada wan	accept ▾	accept ▾	reject ▾	<input type="checkbox"/>	<input type="checkbox"/>	Edit	Delete
Privada	Privada ⇒ Corp	accept ▾	accept ▾	reject ▾	<input type="checkbox"/>	<input type="checkbox"/>	Edit	Delete
DMZ	DMZ ⇒ Corp wan	accept ▾	accept ▾	reject ▾	<input type="checkbox"/>	<input type="checkbox"/>	Edit	Delete

Creamos el zone forwarding, para indicar cómo se comunicarán las zonas.. El wan estará en el corp de manera provisional.



[General Settings](#) [Port Forwards](#) [Traffic Rules](#) [Custom Rules](#)

## Firewall - Traffic Rules - DNS

This page allows you to change advanced properties of the traffic rule entry, such as matched source an

Rule is enabled

Name

Restrict to address family

Protocol

Source zone

Source MAC address

Source address

Source port

Destination zone

Destination address

Destination port

Creamos las diferentes reglas de tráfico, primero hacemos la del DNS para que permita el dns de la interfaz corporativa que es donde estará el dns a todas Any zone

## Firewall - Traffic Rules - AllowSql

This page allows you to change advanced properties of the traffic rule en

Rule is enabled

Name

Restrict to address family

Protocol

Source zone

Source MAC address

Source address

Source port

Destination zone

Permitimos el tráfico sql de la red corporativa a la privada. Podríamos marcar únicamente la ip del servidor web a la sql, pero como no tenemos aún instalada ninguna máquina pondremos any host.

## rewall - Traffic Rules - SMTP

This page allows you to change advanced properties of the traffic rule entry,

Rule is enabled

Name

Restrict to address family

Protocol

Source zone

Source MAC address

Source address

Source port

Destination zone

Permitimos el tráfico de smtp que tendrá que ir por el puerto 587 en el caso que sea smtps, sino habría que cambiarlo a 25.

## Firewall - Traffic Rules - zabbix

This page allows you to change advanced properties of the traffic rule entry, such as

Rule is enabled

Name

Restrict to address family

Protocol

Source zone

Source MAC address

Source address

Source port

Destination zone

Destination address

Destination port

Permitimos el tráfico del zabbix de la red corporativa a cualquier zona

## Firewall - Traffic Rules - Proxy

This page allows you to change advanced properties of the traffic rule entry, such as m:

Rule is enabled

Name

Restrict to address family

Protocol

Source zone

Source MAC address

Source address

Source port

Destination zone

Destination address

Destination port

Permitimos el tráfico de DMZ a corporativa por el puerto 80 ya que es el que usa nginx.

## Firewall - Traffic Rules - POP3

This page allows you to change advanced properties of the traffic rule entry, such as matche:

Rule is enabled

Name

Restrict to address family

Protocol

Source zone

Source MAC address

Source address

Source port

Destination zone

Destination address

Permitimos el pop3 que trabaja en el puerto 110 sin tls

## Firewall - Traffic Rules - IMAP

This page allows you to change advanced properties of the traffic rule en

Rule is enabled

Disable

Name

IMAP


Restrict to address family

IPv4 and IPv6

Protocol

TCP+UDP

Source zone

corporativa: Corporativa: 

Source MAC address

any

Source address

any

Source port

143

Destination zone

Any zone (forward)

Y el IMAP que trabaja en el puerto 143 sin tls

### 3. Instalación del servidor DNS.

Para la instalación del servidor DNS utilizaremos un script que lo instala automáticamente con las configuraciones básicas necesarias para su funcionamiento. Pero para ejecutarlo correctamente tendremos que tener unos prerequisites en el ordenador.

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      # dhcp4: false
      addresses:
        - 192.168.10.5/24
      nameservers:
        addresses: [192.168.10.5, 1.1.1.1]
      routes:
        - to: default
          via: 192.168.10.1
```

Primero tendremos que tener la ip estática.

```
dns@dns-01:~$ cat /etc/hostname
dns-01
dns@dns-01:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 dns-01
```

El archivo hostname y hosts con el nombre que queramos que tenga el servidor DNS.

```
dns@dns-01:~$ chmod +x autoDNS.sh
dns@dns-01:~$ ls -la
total 36
drwxr-x--- 4 dns dns 4096 mar 11 15:47 .
drwxr-xr-x 3 root root 4096 feb 19 15:42 ..
-rwxrwxr-x 1 dns dns 2036 mar 11 15:47 autoDNS.sh
-rw----- 1 dns dns 1752 mar 11 15:40 .bash_history
-rw-r--r-- 1 dns dns 220 ene 6 2022 .bash_logout
-rw-r--r-- 1 dns dns 3771 ene 6 2022 .bashrc
drwx----- 2 dns dns 4096 feb 19 15:42 .cache
-rw-r--r-- 1 dns dns 807 ene 6 2022 .profile
drwx----- 2 dns dns 4096 feb 19 15:42 .ssh
-rw-r--r-- 1 dns dns 0 feb 19 15:45 .sudo_as_admin_successful
dns@dns-01:~$
```

Teniendo el script de instalación del servidor, DNS y con los permisos necesarios ya podemos ejecutarlo.

```
Introduce el nombre de dominio: calanet.cat
Ingresa la direccion ip del servidor DNS: 192.168.10.5
Configuracion completada para calanet.cat
dns@dns-01:~$
```

Nos pedirá el nombre para el dominio y la ip del servidor DNS para que se reconozca.

Este es el script:

```
#!/bin/bash
```

```
#Actualizamos los repositorios
```

```
apt-get update
```

```
#Instalamos el bind9
apt-get install -y bind9
```

```
#Copiamos el archivo de configuracion default como copia de seguridad
cp /etc/bind/named.conf.options /etc/bind/named.conf.options.V2
```

```
#Con cat leemos el archivo que con el parametro EOF (End of file) podemos escribir
#toda la configuracion necesaria.
```

```
cat <<EOF > /etc/bind/named.conf.options
```

```
options {
```

```
    directory "/var/cache/bind";
```

```
    forwarders {
```

```
        8.8.8.8;
```

```
        1.1.1.1;
```

```
    };
```

```
    dnssec-validation auto;
```

```
    listen-on-v6 { any; };
```

```
};
```

```
#Finalizamos el EOF para que no siga leyendo.
```

```
EOF
```

```
echo " "
```

```
cp /etc/bind/named.conf.local /etc/bind/named.conf.localV2
```

```
#Aqui le pedimos al usuario que introduzca el nombre de su dominio para la primera zona
primaria
```

```
read -p "Introduce el nombre de dominio: " dominio
```

```
cat <<EOF > /etc/bind/named.conf.local
```

```
//
```

```
// Do any local configuration here
```

```
//
```

```
// Consider adding the 1918 zones here, if they are not used in your
```

```
// organization
```

```
//include "/etc/bind/zones.rfc1918";
```

```
zone "$dominio" {
```

```
    type master;
```

```
    file "/etc/bind/db.$dominio";
```

```
};
```

```
EOF
```

```
echo " "
```

```
#Copiamos la configuracion local de la base de datos del dns y le ponemos
```

```
#el nombre del dominio
```

```
cp /etc/bind/db.local /etc/bind/db.$dominio
#Le pedimos la ip del servidor DNS
read -p "Ingresa la direccion ip del servidor DNS: " ip_dns
#Obtenemos el nombre del hostname automaticamente.
ns1=$(hostname)
```

```
cat <<EOF > /etc/bind/db.$dominio
```

```
\$TTL 604800
@ IN SOA $ns1.$dominio. root.$dominio. (
        2          ; Serial
        604800     ; Refresh
        86400      ; Retry
        2419200    ; Expire
        604800 )   ; Negative Cache TTL
;
@ IN NS $ns1.$dominio.
$ns1 IN A $ip_dns
```

```
EOF
#Reiniciamos el servicio
systemctl restart bind9
echo "Configuracion completada para $dominio"
```

Cuando lo ejecutemos, nos pedirá el nombre que le queremos dar al dominio y la ip del servidor.

Este script lo que hace es instalar el servicio bind9, y hacer una copia de seguridad de todos los archivos de configuración, el de la base de datos lo copia y lo crea con el nombre del dominio, y hace que se conozca a sí mismo como el servidor dns. También tiene el archivo con los forwarders para poner los proxys alternativos por si no resuelve ip's que no tenga en su registro, y el archivo de configuración de la zona primaria con el nombre de dominio y el archivo de base de datos con el nombre también de dominio para reconocerlo fácilmente para modificaciones.

## 4. Instalación del servidor FTP

El servidor FTP lo utilizaremos para compartir los scripts que utilizarán los trabajadores, crearemos un usuario que esté enjaulado en FTP para que solo pueda ver el directorio donde se encuentran los scripts, usar el mismo usuario para una conexión FTP podría generar conflictos pero al ser únicamente para la descarga de estos scripts no hay problema ya que no podran editar ni subir archivos.

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 192.168.10.10/24
      nameservers:
        addresses: [192.168.10.5]
      routes:
        - to: default
          via: 192.168.10.1
```

Ponemos la ip estática del servidor FTP.

```
ftp_server@ftp:~$ sudo apt update
[sudo] password for ftp_server:
Des:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1.460 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [283 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1.559 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [259 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1.054 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [238 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42,1 kB]
Des:12 http://es.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1.241 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu jammy-security/main Translation-en [223 kB]
Des:14 http://es.archive.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1.526 kB]
Des:15 http://es.archive.ubuntu.com/ubuntu jammy-security/restricted Translation-en [253 kB]
Des:16 http://es.archive.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [848 kB]
Des:17 http://es.archive.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]
Des:18 http://es.archive.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37,1 kB]
Descargados 9.413 kB en 8s (1.227 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se pueden actualizar 77 paquetes. Ejecute «apt list --upgradable» para verlos.
ftp_server@ftp:~$
```

Actualizamos los repositorios.

```
ftp_server@ftp:~$ sudo apt install vsftpd
```

Instalamos el servicio de vsftpd.

```
ftp_server@ftp:~$ sudo adduser ftp_user
Adding user 'ftp_user' ...
Adding new group 'ftp_user' (1001) ...
Adding new user 'ftp_user' (1001) with group 'ftp_user' ...
Creating home directory '/home/ftp_user' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftp_user
Enter the new value, or press ENTER for the default
  Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
ftp_server@ftp:~$
```

Añadimos un usuario que será con el que gestionaremos los scripts.

```
# (the user does not have write access)
# chroot)
chroot_local_user=YES
```

Descomentamos esta línea del archivo /etc/vsftpd.conf, esto sirve para enjaular al usuario local que se conecte por ftp

```
# capabilities.
#
user_config_dir=/etc/vsftpd/user_conf_
```

Nos vamos al archivo /etc/vsftpd.conf y añadimos esta línea al principio del archivo que sirve para indicar dónde estará la configuración del directorio del usuario.

```
ftp_server@ftp:~$ sudo mkdir -p /etc/vsftpd/user_conf
ftp_server@ftp:~$
```



Creamos el directorio donde hemos indicado que estará la configuración del directorio.

```
ftp_server@ftp:~$ sudo mkdir /var/scripts  
[sudo] password for ftp_server:
```

Creamos el directorio donde estarán los scripts.

```
GNU nano 6.2 /etc/vsftpd/user_conf/ftp_user  
local_root=/var/scripts/_
```

Le marcamos la ruta donde estará enjaulado y donde estarán los scripts.

```
GNU nano 6.2 /etc/bind/db.calanet.cat *  
$TTL 604800  
@ IN SOA dns-01.calanet.cat. root.calanet.cat. (  
    2 ; Serial  
    604800 ; Refresh  
    86400 ; Retry  
    2419200 ; Expire  
    604800 ) ; Negative Cache TTL  
;  
@ IN NS dns-01.calanet.cat.  
dns-01 IN A 192.168.10.5  
ftp IN A 192.168.10.10_
```

Para conectarnos al ftp lo haremos a través del dominio y para ello tendremos que crear la entrada para que sepa que servidor es el ftp.

```
dns@dns-01:~$ ftp ftp_user@192.168.10.10  
Connected to 192.168.10.10.  
220 (vsFTPD 3.0.5)  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> pwd  
Remote directory: /  
ftp>
```

Ya podemos conectarnos y ver que estamos en el directorio raíz, para evitar que a través del ftp puedan ver más cosas.



```
604800 ) ; Negative
;
@      IN      NS      dns-01.calanet.cat.
dns-01 IN      A       192.168.10.5
ftp    IN      A       192.168.10.10
ldap   IN      A       192.168.10.15
```

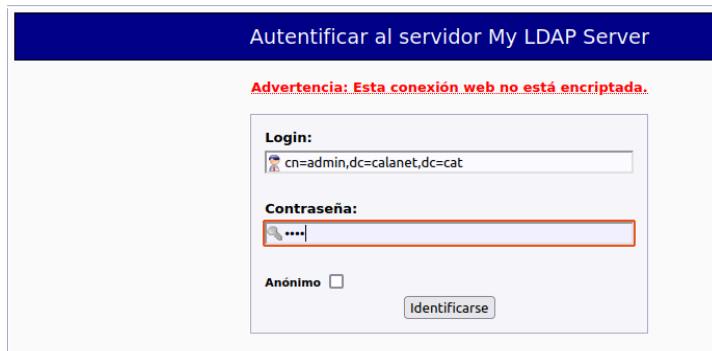
Añadimos el ldap a la configuración del DNS.

```
/* Array of base DNS of your LDAP server. Leave this blank to have phpLDAPadmin
   auto-detect it for you. */
$servers->setValue('server','base',array('dc=calanet,dc=cat'));
```

Cambiamos la siguiente configuración del archivo /etc/phpldapadmin/config.php para que reconozca el mismo dominio que tenemos en el DNS.

```
auth_type, then you can also specify the bind_id/bind_pass here for
the directory for users (ie, if your LDAP server does not allow and
binds. */
$servers->setValue('login','bind_id','cn=admin,dc=calanet,dc=cat');
```

Y en el login también le cambiamos al usuario admin el dominio para que sepa donde encontrarse.



Nos conectamos al LDAP con la cuenta de admin.



Cogiendo el dominio calanet.cat nos vamos a crear un objeto hijo.

### Crear objeto

Servidor: **My LDAP Server** Contenedor: **dc=calanet,dc=cat**

#### Seleccione una plantilla para el proceso de creación

**Plantillas:**

<input type="radio"/> Courier Mail: Cuenta	<input checked="" type="radio"/> Samba: Dominio
<input type="radio"/> Courier Mail: Alias	<input type="radio"/> Samba: Mapeo de Grupo
<input type="radio"/> Genérico: Entrada en la Libreta de Direcciones	<input type="radio"/> Samba: Equipo
<input type="radio"/> Genérico: Entrada DNS	<input checked="" type="radio"/> Sendmail: Alias
<input type="radio"/> Genérico: Alias LDAP	<input checked="" type="radio"/> Sendmail: Cluster
<input type="radio"/> Genérico: Rol Organizacional	<input checked="" type="radio"/> Sendmail: Dominio
<input type="radio"/> Genérico: Unidad Organizacional	<input checked="" type="radio"/> Sendmail: Relevos
<input checked="" type="radio"/> Genérico: Grupo Posix	<input checked="" type="radio"/> Sendmail: Dominio Virtual
<input type="radio"/> Genérico: Objeto de Seguridad Simple	<input checked="" type="radio"/> Sendmail: Usuarios Virtuales
<input type="radio"/> Genérico: Cuenta de Usuario	<input type="radio"/> Thunderbird: Entrada de Libreta Direcciones
<input type="radio"/> Kolab: Entrada de Usuario	<input type="radio"/> Predeterminado

Creamos grupos posix para los diferentes departamentos de la empresa: informatica, rrhh, administración, contabilidad, marketing

#### Nuevo Grupo Posix (Paso 1 de 1)

**Grupo** alias, requerido, rdn

**Número GID** alias, requerido, nota, ro

**Usuarios** alias, nota

Creamos el grupo.

#### Seleccione una plantilla para el proceso de creación

**Plantillas:**

<input type="radio"/> Courier Mail: Cuenta	<input checked="" type="radio"/> Samba: Dominio
<input type="radio"/> Courier Mail: Alias	<input type="radio"/> Samba: Mapeo de Grupo
<input type="radio"/> Genérico: Entrada en la Libreta de Direcciones	<input type="radio"/> Samba: Equipo
<input type="radio"/> Genérico: Entrada DNS	<input checked="" type="radio"/> Sendmail: Alias
<input type="radio"/> Genérico: Alias LDAP	<input checked="" type="radio"/> Sendmail: Cluster
<input type="radio"/> Genérico: Rol Organizacional	<input checked="" type="radio"/> Sendmail: Dominio
<input type="radio"/> Genérico: Unidad Organizacional	<input checked="" type="radio"/> Sendmail: Relevos
<input type="radio"/> Genérico: Grupo Posix	<input checked="" type="radio"/> Sendmail: Dominio Virtual
<input type="radio"/> Genérico: Objeto de Seguridad Simple	<input checked="" type="radio"/> Sendmail: Usuarios Virtuales
<input checked="" type="radio"/> Genérico: Cuenta de Usuario	<input type="radio"/> Thunderbird: Entrada de Libreta de Direcciones
<input type="radio"/> Kolab: Entrada de Usuario	<input type="radio"/> Predeterminado
<input type="radio"/> Samba: Cuenta	

Para las cuentas de usuario, crearemos un objeto hijo dentro de cada objeto como cuenta de usuario. Para los CTO's de la empresa es decirse Marcos y Saul. Crearemos estos usuarios fuera de los grupos.

mgarrido \*

**Contraseña** alias, nota

\*\*\*\* md5  
 \*\*\*\* (confirmar)

[Revisar clave...](#)

**Número UID** alias, requerido, nota, no

1000

**Número GID** alias, requerido, nota

**Directorio personal** alias, requerido

/home/users/mgarrido \*

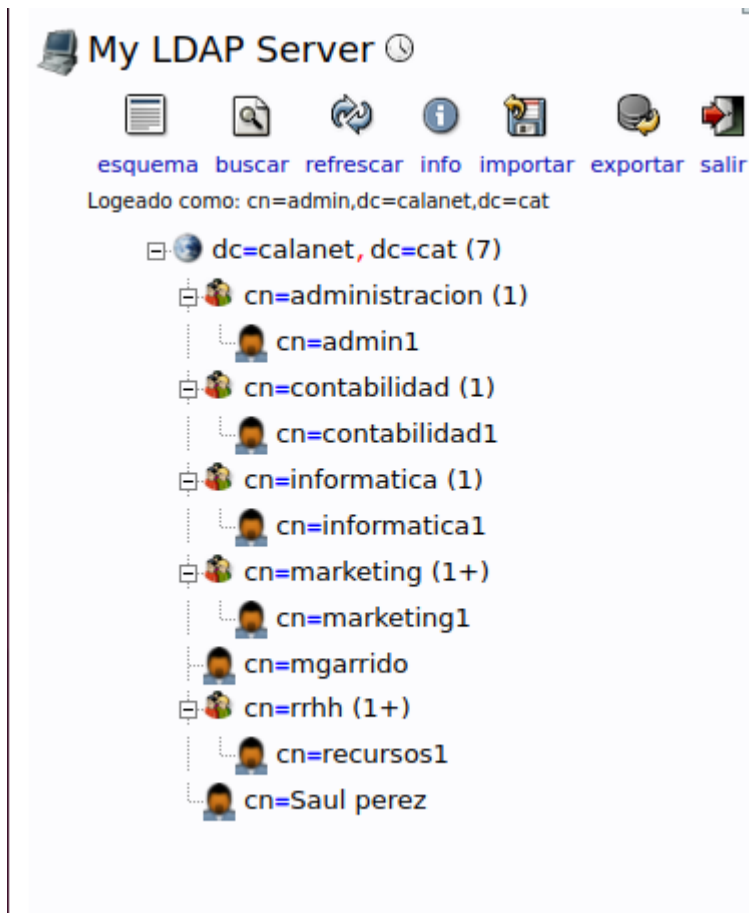
**Consola de Login** alias

Bash

[Crear objeto](#)

1.2.6.3

Creamos los usuarios añadiendo nombre, apellido, alias, contraseña, grupo, directorio personal y consola de login.



El árbol del dominio sería así.



Si queremos añadir algún atributo a un grupo, podemos irnos a un grupo y añadir atributo.

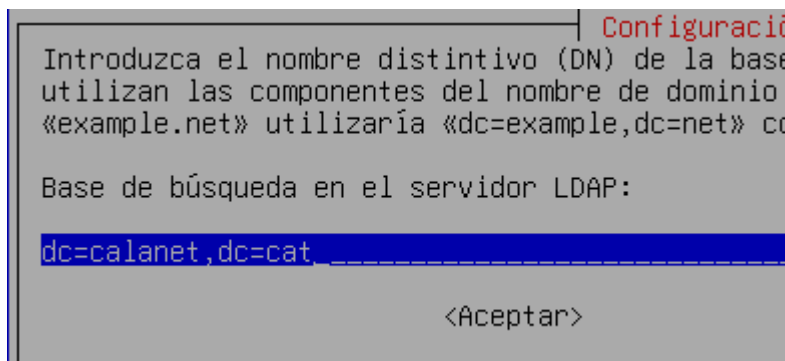


Podemos añadir un memberUid al grupo.

### Conexión ldap desde cliente:

```
root@pc-1:~# apt install libpam-ldap ldap-utils nss-updatedb
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Nota, seleccionando «libpam-ldapd» en lugar de «libpam-ldap»
Se instalarán los siguientes paquetes adicionales:
  libnss-db libnss-ldapd nscd nslcd nslcd-utils
Paquetes sugeridos:
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal kstart
Se instalarán los siguientes paquetes NUEVOS:
  ldap-utils libnss-db libnss-ldapd libpam-ldapd nscd nslcd nslcd-utils nss-updatedb
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 571 kB de archivos.
Se utilizarán 1.905 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

Para loggarse desde un cliente instalamos los paquetes necesarios para la conexión. Nos pedirá el URI del dominio de ldap o la ip.



Ahora nos pedirá el nombre distintivo que sería: dc=calanet,dc=cat.

```
Configuración de libnss-ldapd
Para que este programa funcione, debe modificar el archivo «/etc/nsswitch.conf» para que utilice la fuente de datos de LDAP.

Puede escoger los servicios que se deben habilitar para las búsquedas de LDAP. Las búsquedas de LDAP se añadirán como última fuente de datos. Asegúrese de reiniciar los servicios después de los cambios.

Indique los servicios de nombre a configurar:
[*] passwd
[*] group
[*] shadow
[ ] hosts
[ ] networks
```

Le decimos que usaremos los servicios passwd, group y shadow.

```
GNU nano 7.2 /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, run:
# `info libc "Name Service Switch"' for information about this configuration.
passwd:          compat ldap
group:           compat ldap
shadow:          compat ldap
gshadow:         files systemd
```

Cambiamos la configuración del archivo /etc/nsswitch.conf

```
GNU nano 7.2 confpam
auth sufficient pam_ldap.so
account sufficient pam_ldap.so
session sufficient pam_ldap.so
```

Añadimos estas tres líneas a estos archivos: /etc/pam.d/common-session, /etc/pam.d/common-auth, /etc/pam.d/common-password, /etc/pam.d/common-account.

```
root@pc-1:~# systemctl restart nscd.service
root@pc-1:~#
```

Reiniciamos el servicio.

## 6. Instalación del NFS.

Para ahorrar una máquina, instalaremos el servidor de nfs en la máquina de ldap.

```
root@ldap:~# apt install nfs-kernel-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  keyutils libevent-core-2.1-7 libnfsidmap1 libyam1-0-2 nfs-common python3
Paquetes sugeridos:
  open-iscsi watchdog
Se instalarán los siguientes paquetes NUEVOS:
  keyutils libevent-core-2.1-7 libnfsidmap1 libyam1-0-2 nfs-common nfs-kernel-server rpcbind
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 14 no actualizados
Se necesita descargar 872 kB de archivos.
Se utilizarán 3.342 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Actualizamos los repositorios e instalamos el paquete necesario para el NFS.

~nfs-kernel-server

```
root@ldap:/home# mkdir calanet
root@ldap:/home# ls
calanet  ldap
root@ldap:/home# _
```

Creamos la carpeta donde montaremos el nfs.

```
GNU nano 7.2 /etc/exports
# /etc/exports: the access control list for filesystems which may be
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home/calanet 192.168.10.0/24(rw,sync,no_subtree_check)
```

Configuramos la carpeta que compartiremos en el /etc/exports.

La ip de red en la red que queramos compartir la carpeta compartida.

rw: permisos de lectura y escritura en la carpeta compartida

sync: Los cambios se sincronizan de forma inmediata en el servidor nfs

no\_subtree\_check: Desactiva la comprobación de subdirectorios, para mejorar el rendimiento.

```
root@ldap:/home/calanet# ls
admin1  contabilidad1  informatica1  marketing1  mgarrido  recursos1  sperez
root@ldap:/home/calanet# _
```

Creamos cada carpeta de los usuarios.

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@ldap:/home/calanet# ls
admin1  contabilidad1  informatica1  marketing1  mgarrido  recursos1  sperez
root@ldap:/home/calanet# chown admin1:administracion admin1/
root@ldap:/home/calanet# chown contabilidad1:contabilidad contabilidad1/
root@ldap:/home/calanet# chown informatica1:informatica informatica1/
root@ldap:/home/calanet# chown marketing1:marketing marketing1/
root@ldap:/home/calanet# chown mgarrido mgarrido/
root@ldap:/home/calanet# chown recursos1:rrhh recursos1/
root@ldap:/home/calanet# chown sperez sperez/
root@ldap:/home/calanet# ls -la
total 36
drwxr-xr-x 9 root      root      4096 abr  2 15:52 .
drwxr-xr-x 4 root      root      4096 abr  2 15:51 ..
drwxr-xr-x 2 admin1    administracion 4096 abr  2 15:52 admin1
drwxr-xr-x 2 contabilidad1 contabilidad 4096 abr  2 15:52 contabilidad1
drwxr-xr-x 2 informatica1 informatica 4096 abr  2 15:52 informatica1
drwxr-xr-x 2 marketing1 marketing 4096 abr  2 15:52 marketing1
drwxr-xr-x 2 mgarrido  root      4096 abr  2 15:52 mgarrido
drwxr-xr-x 2 recursos1 rrhh      4096 abr  2 15:52 recursos1
drwxr-xr-x 2 sperez    root      4096 abr  2 15:52 sperez
root@ldap:/home/calanet# _
```

Y cambiamos los usuarios y grupos de cada directorio.



```
root@pc-1:~# apt install nfs-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  keyutils libevent-core-2.1-7 libnfsidmap1 libwrap0 rpcbind
Paquetes sugeridos:
  open-iscsi watchdog
Se instalarán los siguientes paquetes NUEVOS:
  keyutils libevent-core-2.1-7 libnfsidmap1 libwrap0 nfs-common rpcbind
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 603 kB de archivos.
Se utilizarán 2.169 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Para la conexión del nfs, instalamos nfs-common en la máquina que sería de algún trabajador.

```
GNU nano 7.2 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=d96b25d8-7300-4929-bbf2-2415c3b04894 / ext4 errors=remount-ro 0
# swap was on /dev/sda5 during installation
UUID=06dbc882-2075-4731-8d9c-10cf174a59b2 none swap sw 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
192.168.10.15:/home/calanet /mnt nfs defaults 0 0_
```

Añadimos la línea en /etc/fstab para que se monte automáticamente el nfs al arrancar el sistema.

homeDirectory
/mnt/mgarrido

Cambiamos el homeDirectory del usuario de ldap.

```
Debian GNU/Linux 12 pc-1 tty2
pc-1 login: mgarrido
Password:
Linux pc-1 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
mgarrido@pc-1:~$ pwd
/mnt/mgarrido
mgarrido@pc-1:~$
```

Para que lo tenga en cuenta como home cuando nos loggeemos con el usuario.

## 7. Instalación del servidor Base de datos(MySql).

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernet:
    enp0s3:
      dhcp4: false
      addresses:
        - 192.168.30.50/24
      nameservers:
        addresses: [192.168.10.5]
      routes:
        - to: default
          via: 192.168.30.1
```

Configuramos la dirección ip como estática.

Para la instalación del servidor sql utilizaremos el siguiente script:

```
#!/bin/bash
```

```
#Funcion para validar una direccion ip.
```

```
validate_ip() {
```

```
    #Cogemos la primera opcion que ponemos junto a la funcion
```

```
    local ip=$1
```

```
    #Creamos las reglas para validar la ip.
```

```
    local reglas="^[0-9]{1,3}\.){3}[0-9]{1,3}$"
```

```
    #Comparamos la ip con el patron de referencia, esto compara str
```

```
    if [[ $ip =~ $reglas ]]; then
```

```
        #Si coincide devuelve 0 que es True
```

```
        echo "IP valida."
```

```
        return 0
```

```
    else
```

```
        #Sino devuelve 1 que es false.
```

```
        echo "IP no válida. Pon una ip valida."
```

```
        return 1
```

```
    fi
```

```
}
```

```
#Actualizar repositorios e instalar Mysql
```

```
apt-get update
```

```
apt-get install -y mysql-server
```

```
#Pedimos la contraseña para el usuario root de mysql
```

```
read -p "Introduce la contraseña para el usuario root de mysql: " rootpass
```

```
#Pedimos informacion para la base de datos y usuario de apache
```

```
read -p "Introduce el nombre de la base de datos: " database
```

```
read -p "Nombre para el usuario admin del servidor apache: " adminapache
```

```
read -p "Contraseña para el usuario $adminapache: " contrapache
```

```
#Validamos la ip
```

```
while true; do
```

```
    read -p "IP del servidor apache: " ip_apache
```

```
    if validate_ip "$ip_apache"; then
```

```
        break
```

```
    fi
```

```
done
```

```
#Acceder a MYSQL y ejecutar comando sql para crear base de datos
```

```
#usuarios y cambiar la contraseña del root de sql
```

```
mysql -u root <<MYSQL_SCRIPT
```

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

```
FLUSH PRIVILEGES;
```

```
CREATE DATABASE $database;
```

```
CREATE USER '$adminapache'@'$ip_apache' IDENTIFIED BY '$contrapache';
```

```
GRANT ALL PRIVILEGES ON $database.* TO '$adminapache'@'$ip_apache';
```

```
FLUSH PRIVILEGES;
```

```
MYSQL_SCRIPT
```

```
echo "Configuracion de MySQL completada"
```

```
conf="/etc/mysql/mysql.conf.d/mysqld.cnf"
```

```
cat <<EOF > "$conf"
```

```
[mysqld]
```

```
user      = mysql
```

```
bind-address      = 0.0.0.0
```

```
mysqlx-bind-address  = 0.0.0.0
```

```
key_buffer_size     = 16M
```

```
myisam-recover-options = BACKUP
```

```
# max_connections      = 151
```

```
# table_open_cache     = 4000
```

```
# Error log - should be very few entries.
```

```
#
```

```
log_error = /var/log/mysql/error.log
```

```
max_binlog_size = 100M
```

```
EOF
```

```
sudo systemctl restart mysql
```

```
echo "Archivo de configuracion creado en $conf. "
```

Este script lo que hace, es instalar el servidor mysql y te pide una serie de parámetros. Crea la base de datos que usaremos para el wordpress, le cambia la contraseña al usuario root para el mysql, crea el usuario que tendrá permisos sobre la base de datos del wordpress y le da los permisos.

Después edita el archivo de configuración /etc/mysql/mysql.conf.d/mysqld.cnf y hace que pueda escuchar en todas direcciones, para permitir conexiones en más sitios que no sean local.

```
Introduce la contraseña para el usuario root de mysql: 1234
Introduce el nombre de la base de datos: wordpress
Nombre para el usuario admin del servidor apache: eladmin
Contraseña para el usuario eladmin: 1234
IP del servidor apache: 192.168.10.12
IP valida.
Configuracion de MySQL completada
Archivo de configuracion creado en /etc/mysql/mysql.conf.d/mysqld.cnf.
sql@sql:~$
```

Una vez nos pida las diferentes configuraciones, ya estará listo para funcionar.

## 8. Instalación del servidor web(Wordpress).

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 192.168.10.12/24
      nameservers:
        addresses: [192.168.10.5]
      routes:
        - to: default
          via: 192.168.10.1
```

Le configuramos la ip estática al servidor web.

```
192.168.10.12
www      IN      A       192.168.10.12
```

Añadimos el servidor web al dns para que lo reconozca. Nuestra página de apache se buscaría así: **www.calanet.cat**

Para la instalación del wordpress utilizaremos el siguiente script para hacerlo de manera automatizada:

```
#!/bin/bash
```

```
#Instalacion de apache, php y extensiones necesarias.
```

```
apt update
```

```
apt install -y apache2 libapache2-mod-php mysql-client php php-bcmath php-curl
php-imagick php-intl php-json php-mbstring php-mysql php-xml php-zip unzip
```

```
#Habilitamos el modulo php y el remoteip.
```

```
a2enmod php*
```

```
a2enmod remoteip
```

```
#Pedimos la ip de red
```

```
echo "En caso de error cambiar ip de red en /etc/apache2/apache2.conf"
```

```
read -p "Introduce la ip de red donde estara colocado el proxy. ej: 10.0.2.0/24: " ipred
```

```
#Añadimos la configuracion de headers de proxy.
```

```
echo "RemoteIPHeader X-Forwarded-For" | tee -a /etc/apache2/apache2.conf
```

```
echo "RemoteIPInternalProxy $ipred" | tee -a /etc/apache2/apache2.conf
```

```
#Instalacion de wordpress
```

```
cd /tmp
```

```
wget https://es.wordpress.org/latest-es_ES.zip
```

```
mkdir -p /var/www/cms-web/
```

```
unzip latest-es_ES.zip -d /var/www/cms-web/
```

```
mv /var/www/cms-web/wordpress/* /var/www/cms-web/
```

```
chown -R www-data:www-data /var/www/cms-web/
```

```
chmod -R 755 /var/www/cms-web/
```

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

```
cat <<EOF > /etc/apache2/sites-available/cms-web.conf
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/cms-web
    ServerAlias www.cms-web.com
    ErrorLog ${APACHE_LOG_DIR}/cms-web-error.log
    CustomLog ${APACHE_LOG_DIR}/cms-web-access.log combined
</VirtualHost>

EOF
```

```
a2ensite cms-web.conf
systemctl reload apache2
systemctl restart apache2
systemctl status apache2
```

Este script lo que hace es instalar el apache, el mysql-client y todas las dependencias que necesita wordpress. Una vez instaladas, pide la ip de red donde se encontrará nuestro servidor proxy, que sería la 192.168.20.0/24. Después habilita los módulos de php y remoteip para que pueda trabajar correctamente el wordpress.

Añade las cabeceras de proxy para que pueda funcionar el nginx correctamente, descarga el comprimido de wordpress, lo descomprime en la nueva raíz que haremos para nuestra web. Una vez tengamos todo hecho solo queda deshabilitar el sitio por defecto y crear el virtualhost para que tenga como DocumentRoot la nueva ruta para el wordpress.

```
apache@www:~$ mysql -h 192.168.30.50 -u eladmin -p wordpress
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.22.04.1 (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.

mysql> _
```

Una vez ejecutado el script comprobamos que funciona correctamente la conexión con el sql.



## Empezamos la instalación del wordpress.

A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.

**Nombre de la base de datos**   
El nombre de la base de datos que quieres usar con WordPress.

**Nombre de usuario**   
El nombre de usuario de tu base de datos.

**Contraseña**  [Mostrar](#)  
La contraseña de tu base de datos.

**Servidor de la base de datos**   
Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.

**Prefijo de tabla**   
Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

[Enviar](#)

Le introducimos la configuración de base de datos que hayamos introducido anteriormente con el script.

¡Muy bien! Ya has terminado esta parte de la instalación. Ahora WordPress puede comunicarse con tu base de datos. Si estás listo, es el momento de...

[Realizar la instalación](#)

## Ya podemos realizar la instalación.

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

**Título del sitio**

**Nombre de usuario**   
Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

**Contraseña**  [Ocultar](#)  
**Muy débil**  
**Importante:** Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

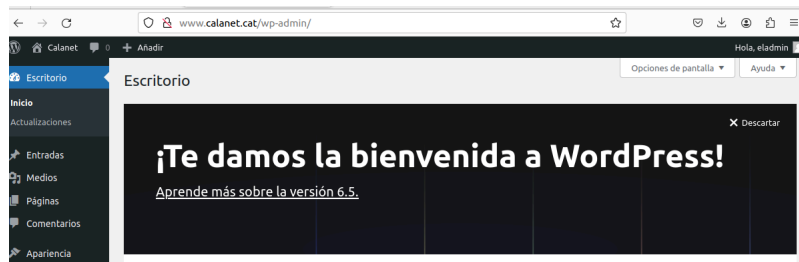
**Confirma la contraseña** ☒ Confirma el uso de una contraseña débil.

**Tu correo electrónico**   
Comprueba bien tu dirección de correo electrónico antes de continuar.

**Visibilidad en los motores de búsqueda** ☐ Pedir a los motores de búsqueda que no indexen este sitio  
Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

Configuramos el título del sitio, nombre de usuario, la contraseña y el correo electrónico.



Una vez le demos a instalar wordpress, ya podemos hacer login y tener el control de nuestro wordpress.



## 9. Instalación del servidor de correo.

### a. Configuración del MTA (Postfix)

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 192.168.10.142/24
      nameservers:
        addresses: [192.168.10.5]
      routes:
        - to: default
          via: 192.168.10.1
      version: 2
```

Primero de todo, le configuramos una ip estática a la máquina donde tendremos el servidor de correo.

#### Configuración del servidor DNS:

```
mailing IN      A      192.168.10.142
mailing IN      MX     10      mailing.calanet.cat.
smtp      IN      CNAME   mailing
pop3      IN      CNAME   mailing
imap      IN      CNAME   mailing
```

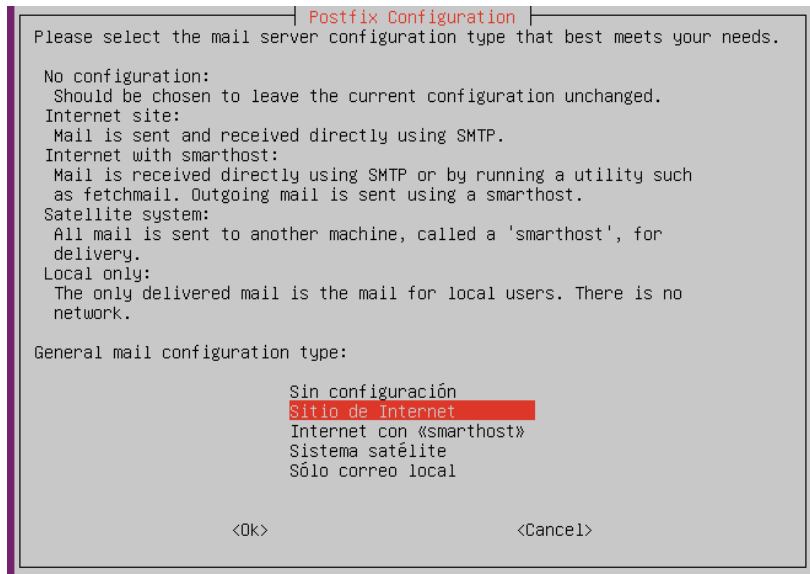
Modificamos el archivo `/etc/bind/db.calanet.cat` y añadimos las entradas de mailing. Los cname los podemos añadir para ponerle un alias a la entrada del servidor mailing para facilitar el uso de ese protocolo.

```
dns@dns-01:~$ sudo named-checkzone calanet.cat /etc/bind/db.calanet.cat
zone calanet.cat/IN: loaded serial 2
OK
dns@dns-01:~$ _
```

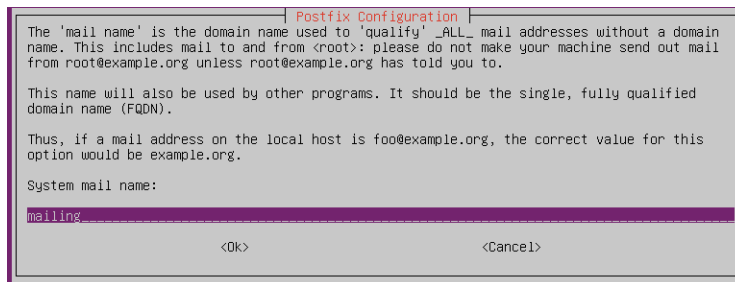
Podemos comprobar que no hay errores en los registros con el siguiente comando. Y podemos reiniciar el servicio para que se apliquen los cambios.

```
mailing@mail:~$ sudo apt install postfix
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  ssl-cert
Paquetes sugeridos:
  procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb postfix-sqlite
  sasl2-bin | dovecot-common resolvconf postfix-cdb mail-reader postfix-mta-sts-resolver
  postfix-doc
Se instalarán los siguientes paquetes NUEVOS:
  postfix ssl-cert
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 67 no actualizados.
Se necesita descargar 1.265 kB de archivos.
Se utilizarán 4.248 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

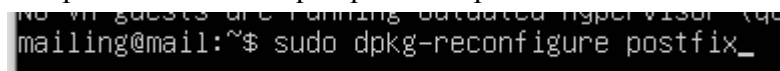
Instalamos el postfix con: **`~apt install postfix`**



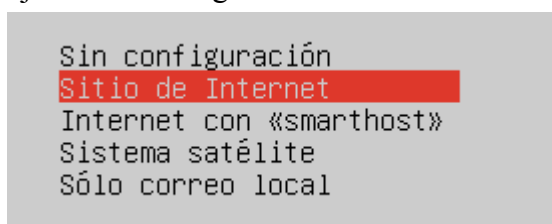
Nos pedirá el tipo de servidor de mail, que queremos que sea. Le decimos que queremos que sea un sitio de internet.



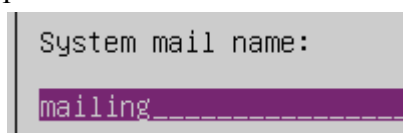
Nos pedirá el nombre que queremos para el sistema de correo.



Para acabar de configurarlo, ejecutamos `sudo dpkg-reconfigure postfix`. Ejecutamos el siguiente comando.



Podemos elegir que sea un sitio de internet si en la instalación no lo hemos puesto.



El nombre del sistema de mail.

**Postfix Configuration**

Mail for the 'postmaster', 'root', and other system accounts needs to be redirected to the user account of the actual system administrator.

If this value is left empty, such mail will be saved in /var/mail/nobody, which is not recommended.

Mail is not delivered to external delivery agents as root.

If you already have a /etc/aliases file and it does not have an entry for root, then you should add this entry. Leave this blank to not add one.

Recipient for root and postmaster mail:

mailing

<Ok> <Cancel>

Después del hostname nos pide un usuario que sea el administrador del postfix, añadimos nuestro usuarios con sudo que en este caso también se llama mailing.

**Postfix Configuration**

Introduzca una lista, separada por comas, de dominios para los que esta máquina considerarse como su destino final. Si esta es una pasarela de correo del dominio probablemente querrá incluir el dominio padre.

Otros destinos para los cuales aceptar correo (en blanco para ninguno):

\$myhostname, calanet.cat, localhost

<Ok> <Cancel>

Hacemos que la máquina pueda enviar los correos a los siguientes destinos, nuestro nombre de equipo, el dominio y nuestro host.

Si se fuerzan las actualizaciones síncronas, el correo será procesado más lentamente. Si no se fuerzan, existe la posibilidad remota de perder algunos correos si el sistema colapsa en un momento inoportuno y no está usando un sistema de archivos transaccional (como ext3).

¿Forzar actualizaciones síncronas en la cola de correo?

<Yes> <No>

No forzamos las actualizaciones síncronas en la cola de correo.

**Postfix Configuration**

Please specify the network blocks for which this host should relay mail. The default is just the local host, which is needed by some mail user agents. The default includes local host for both IPv4 and IPv6. If just connecting via one IP version, the unused value(s) may be removed.

If this host is a smarthost for a block of machines, you need to specify the netblocks here, or mail will be rejected rather than relayed.

To use the Postfix default (which is based on the connected subnets), leave this blank.

Local networks:

127.0.0.0/8 192.168.10.0/24 192.168.20.0/24 192.168.30.0/24

<Ok> <Cancel>

Introducimos las redes internas por las que enviara lo correos.

Mailbox size limit (bytes):

1048576000

<Ok>

Ponemos que de correo como máximo se puedan enviar 1000MB

```
To not use address extensions, leave the string bla  
Local address extension character:  
+  
  
<Ok>
```

Dejamos el carácter de extensión por defecto.

```
Protocolos de Internet a usar:  
  
todos  
ipv6  
ipv4
```

Le decimos que escuche por el protocolo ipv4

Ahora para el sistema de buzones utilizaremos maildir, existe maildir o mailbox pero hay una diferencia clave por la que lo voy a usar. La diferencia entre Mailbox y Maildir es que Mailbox guarda los mensajes en un único fichero y Maildir guarda los mensajes en una estructura de ficheros y directorios, con lo cual no requiere bloque de ficheros para mantener la integridad de los mensajes. Si bien ambos formatos están ampliamente extendidos y son eficaces, el que se utilizará será Maildir, debido en gran parte a la comodidad e independencia de los mensajes que aporta la estructuración en directorios.

```
home_mailbox = Maildir/_
```

Para ello añadimos esta línea al final del archivo `/etc/postfix/main.cf`. Maildir creará este directorio en cada usuario.

```
cliente@cliente: ~/Desktop  
File Edit View Search Terminal Help  
cliente@cliente:~/Desktop$ telnet 192.168.10.142 25  
Trying 192.168.10.142...  
Connected to 192.168.10.142.  
Escape character is '^]'.  
220 mail ESMTPE Postfix (Ubuntu)
```

Para probar el funcionamiento del postfix, vamos a enviar un correo desde una máquina externa para probar su funcionamiento. Utilizaremos telnet.

```
mail from: <prueba@calanet.cat>
250 2.1.0 Ok
rcpt to: <mailing@calanet.cat>
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Mensaje de prueba
.
250 2.0.0 Ok: queued as 0D9FF814ED
quit
221 2.0.0 Bye
Connection closed by foreign host.
cliente@cliente:~/Desktop$
```

Enviamos un correo al usuario [mailing@calanet.cat](mailto:mailing@calanet.cat) que hemos marcado antes en la configuración del postfix.

```
mailing@mail:~$ ls Maildir/
cur new tmp
mailing@mail:~$ ls Maildir/new/
1712586991.Vfd00I4593aM860353.mail
mailing@mail:~$
```

Y podemos ver que ha llegado el mail de prueba que hemos enviado, en Maildir/new/

TLS encripta por cifrado simétrico todo el tráfico de datos que se realiza a través de conexiones TCP. El cliente comprueba la validez del certificado y envía al servidor un número aleatorio cifrado con la clave pública del servidor.

```
# TLS parameters
smtpd_use_tls=yes_
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
```

Para utilizar el tls tenemos que añadir esta línea en el fichero /etc/postfix/main.cf, y dejamos los certificados por defecto.

```
cliente@cliente:~/Desktop$ telnet 192.168.10.142 25
Trying 192.168.10.142...
Connected to 192.168.10.142.
Escape character is '^]'.
220 mail ESMTP Postfix (Ubuntu)
ehlo testing
250-mail
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
```

Volvemos a conectarnos al servidor de correo por telnet al puerto del smtp, y ejecutamos ehlo testing. Y nos devolverá todo lo que soporta el protocolo.

```
STARTTLS
220 2.0.0 Ready to start TLS
```

Ejecutamos startls y podemos ver que ya está preparado para utilizar el TLS.

```
mailing@mail:~$ sudo adduser marcos
Adding user `marcos' ...
Adding new group `marcos' (1001) ...
```

Creamos el usuario marcos.

```
GNU nano 6.2 /etc/aliases
# See man 5 aliases for format
postmaster: root
root: mailing@calanet.cat
admin: marcos@calanet.cat
```

Para gestionar los correos de los usuarios podemos utilizar los alias, el usuario marcos que por defecto tendría el correo [marcos@calanet.cat](mailto:marcos@calanet.cat), creando este alias podría recibir mensajes como [admin@calanet.cat](mailto:admin@calanet.cat).

```
mailing@mail:~$ sudo newaliases
mailing@mail:~$
```

Ejecutamos sudo newaliases para aplicar los cambios.

## b. Configuración del MDA (Dovecot).

```
mailing@mail:~$ sudo apt install dovecot-imapd dovecot-pop3d
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  dovecot-core libexttextcat-2.0-0 libexttextcat-data liblua5.3-0
Paquetes sugeridos:
  dovecot-gssapi dovecot-ldap dovecot-lmtpd dovecot-lucene dovecot-managesieved dovecot-mysql
  dovecot-pgsql dovecot-sieve dovecot-solr dovecot-sqlite dovecot-submissiond ntp
Se instalarán los siguientes paquetes NUEVOS:
  dovecot-core dovecot-imapd dovecot-pop3d libexttextcat-2.0-0 libexttextcat-data liblua5.3-0
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 67 no actualizados.
Se necesita descargar 3.883 kB de archivos.
Se utilizarán 12,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

Instalamos los paquetes necesarios para el agente de entrega de mensajes, que serían dovecot-imapd y dovecot-pop3d.

```
mailing@mail:~$ sudo cp /etc/dovecot/dovecot.conf /etc/dovecot/dovecotcp1.conf
mailing@mail:~$ sudo cp -R /etc/dovecot/conf.d/ /etc/dovecot/confcp1.d/
mailing@mail:~$ _
```

Hacemos una copia de seguridad de los archivos de configuración.

```
# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespaces)
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on config
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var
# Enable installed protocols
!include_try /usr/share/dovecot/protocols.d/*.protocol
```

En el archivo de configuración de /etc/dovecot/dovecot.conf podemos ver la línea !include\_try esto lo que hace es que en la carpeta que indica se configuran los protocolos.

```
GNU nano 6.2 /usr/share/dovecot/protocols.d/imapd.protocol
protocols = $protocols imap
```

Comprobamos que estén los protocolos configurados.

```
mailing@mailing:/etc/dovecot$ sudo grep -lir mail_location
confcp1.d/10-mail.conf
conf.d/10-mail.conf
dovecot-sql.conf.ext
mailing@mailing:/etc/dovecot$
```

Para cambiar el sistema de buzones a maildir, tenemos que configurar el archivo donde indica que sistema utilizar.

```
# mail_location = maildir:~/Maildir
# mail_location = mbox:~/mail:INBOX=/var/mail
# mail_location = mbox:/var/mail/%d/%n/%n:
#
# <doc/wiki/MailLocation.txt>
#
mail_location = maildir:~/Maildir
```

Cambiamos el mail\_location a maildir:~/Maildir

```
mailing@mailing:/etc/dovecot$ sudo grep -lir ssl_cert
confcp1.d/10-ssl.conf
conf.d/10-ssl.conf
dovecot-sql.conf.ext
mailing@mailing:/etc/dovecot$
```

Para la configuración del ssl podemos buscar la directiva ssl\_cert

```
GNU nano 6.2 conf.d/10-ssl.conf
##
## SSL settings
##
# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>
ssl = yes
# PEM encoded X.509 SSL/TLS certificate and private key. They're opened
# dropping root privileges, so keep the key file unreadable by anyone
# root. Included doc/mkcert.sh can be used to easily generate self-sig
# certificate, just make sure to update the domains in dovecot-openssl
ssl_cert = </etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key = </etc/ssl/private/ssl-cert-snakeoil.key
# If key file is secured, protect it with a password here. Alternati
```

Cambiamos las rutas donde se encuentran los certificados.

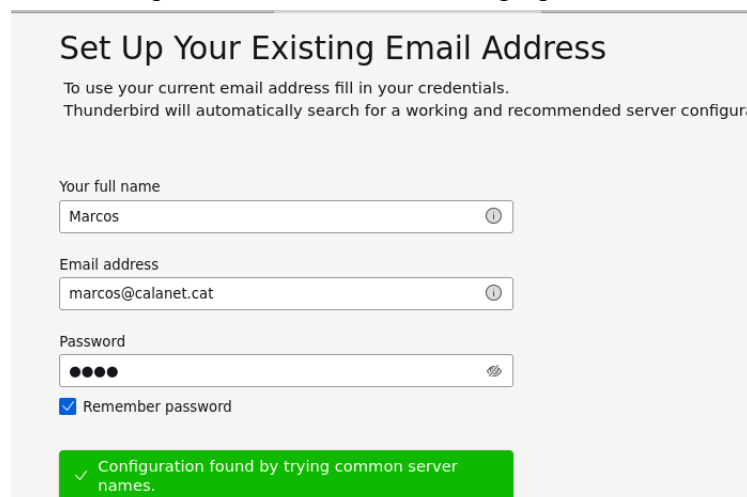
```
GNU nano 6.2 conf.d/10-auth.conf
##
## Authentication processes
##
# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed
# See also ssl-required setting.
disable_plaintext_auth = no
```

Ya que es una conexión segura haremos que los usuarios se conecten por texto plano, y que solo puedan hacer login desde usuarios que este en el /etc/passwd

### c. Instalación de ThunderBird

```
cliente@cliente:~/Desktop$ sudo apt install thunderbird
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

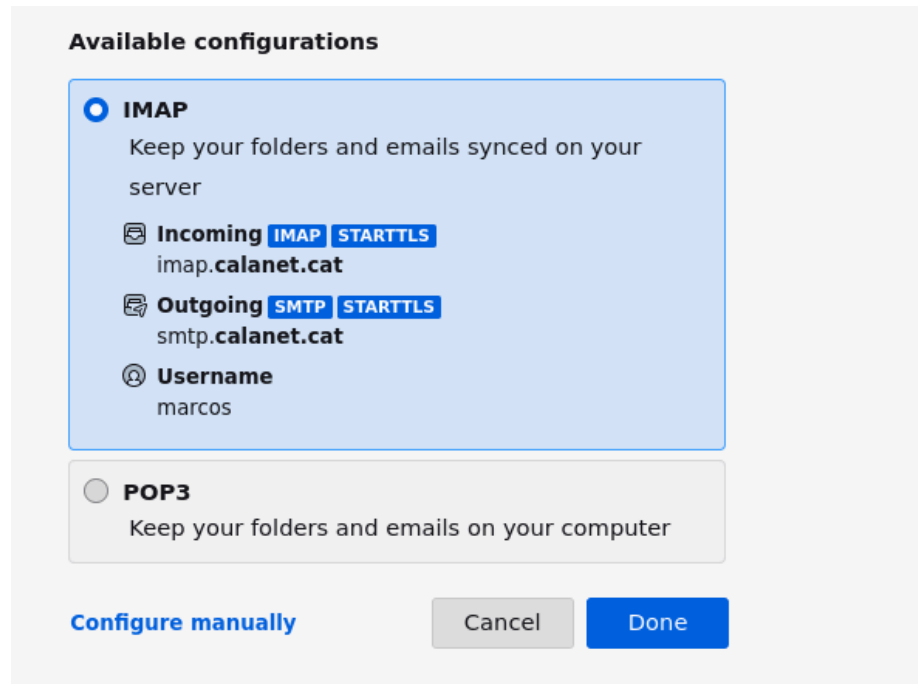
En una máquina cliente, instalamos el paquete de thunderbird.



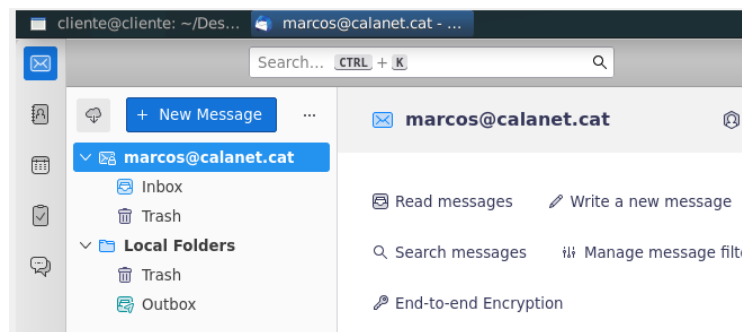
The image shows the Thunderbird 'Set Up Your Existing Email Address' window. It has a light gray background. At the top, the title 'Set Up Your Existing Email Address' is in bold. Below it, a subtitle says 'To use your current email address fill in your credentials. Thunderbird will automatically search for a working and recommended server configuration'. There are three input fields: 'Your full name' with the value 'Marcos', 'Email address' with the value 'marcos@calanet.cat', and 'Password' with four dots. Below the password field is a checked checkbox labeled 'Remember password'. At the bottom, a green button with a white checkmark and the text 'Configuration found by trying common server names.' is visible.

Abrimos el thunderbird, y empezaremos con la configuración de la cuenta que vamos a usar.





Una vez introducidas las credenciales, nos mostrará las posibles configuraciones para el servidor de correo.



Una vez seleccionada la configuración que utilizaremos, ya tendremos nuestra bandeja de entrada y podremos empezar a enviar mensajes.

#### d. Automatización de la instalación del servidor de correo:

```
#!/bin/bash
```

```
#Actualizamos los repositorios antes de instalar el postfix
apt update
```

```
#Instalamos el paquete de postfix desactivando la forma interactiva
DEBIAN_FRONTEND=noninteractive apt-get -y install postfix
```

```
#Hacemos una copia de seguridad del archivo de configuracion del postfix.
cp /etc/postfix/main.cf /etc/postfix/maincp1.cf
```

```
#Editamos el archivo de configuracion
cat <<EOF > /etc/postfix/main.cf
```

```
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no
append_dot_mydomain = no
readme_directory = no

compatibility_level = 3.6

# TLS parameters
smtpd_use_tls=yes
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CAuth=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions    =    permit_mynetworks    permit_sasl_authenticated
defer_unauth_destination
myhostname = mail
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, calanet.cat, localhost
relayhost =
mynetworks = 127.0.0.0/8 192.168.10.0/24 192.168.20.0/24 192.168.30.0/24
mailbox_size_limit = 1048576000
recipient_delimiter = +
inet_interfaces = all
inet_protocols = ipv4
home_mailbox = Maildir/

EOF

#Una vez este listo el postfix, comenzamos con la instalacion del MDA.
apt update
#Instalamos los paquetes de los protocolos que utilizaremos
apt install -y dovecot-imapd dovecot-pop3d
#Hacemos una copia de seguridad de los archivos de configuracion.
cp /etc/dovecot/dovecot.conf /etc/dovecot/dovecotcp1.conf
cp -R /etc/dovecot/conf.d/ /etc/dovecot/confcp1.d/

#Editamos el archivo de configuracion de dovecot.
cat <<EOF > /etc/dovecot/dovecot.conf

# Enable installed protocols
```

```
!include_try /usr/share/dovecot/protocols.d/*.protocol
```

```
dict {  
    #quota = mysql:/etc/dovecot/dovecot-dict-sql.conf.ext  
}
```

```
# Most of the actual configuration gets included below. The filenames are  
# first sorted by their ASCII value and parsed in that order. The 00-prefixes  
# in filenames are intended to make it easier to understand the ordering.  
!include conf.d/*.conf
```

```
# A config file can also tried to be included without giving an error if  
# it's not found:  
!include_try local.conf
```

EOF

```
#Confirmacion de la configuracion de protocolos  
cat <<EOF > /usr/share/dovecot/protocols.d/imapd.protocol  
protocols = \${protocols} imap  
EOF
```

```
cat <<EOF > /usr/share/dovecot/protocols.d/pop3d.protocol  
protocols = \${protocols} pop3  
EOF
```

```
#Configuracion del buzón de dovecot.  
cat <<EOF > /etc/dovecot/conf.d/10-mail.conf
```

```
mail_location = maildir:~/Maildir
```

```
namespace inbox {  
    inbox = yes
```

```
}
```

```
mail_privileged_group = mail
```

```
protocol !indexer-worker {  
}
```

```
EOF  
#Configuracion del ssl en el dovecot  
cat <<EOF > /etc/dovecot/conf.d/10-ssl.conf
```

```
ssl = yes
```

```
ssl_cert = </etc/ssl/certs/ssl-cert-snakeoil.pem  
ssl_key = </etc/ssl/private/ssl-cert-snakeoil.key
```

```
ssl_client_ca_dir = /etc/ssl/certs  
ssl_dh = </usr/share/dovecot/dh.pem
```

```
EOF  
#Configuracion del tipo de conexion  
cat <<EOF > /etc/dovecot/conf.d/10-auth.conf
```

```
disable_plaintext_auth = no
```

```
auth_mechanisms = plain
```

```
!include auth-system.conf.ext
```

```
EOF
```

## 12. Instalación del servidor de monitorización(Zabbix).

Zabbix es una herramienta de monitorización de red y servidores de código abierto que permite supervisar y analizar el rendimiento y el estado de los sistemas y servicios en tiempo real, lo utilizaremos para monitorizar la infraestructura de la empresa.

```
zabbix@zabbix:~$ cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 192.168.10.225/24
      nameservers:
        addresses: [192.168.10.5]
      routes:
        - to: default
          via: 192.168.10.1
```

Para empezar la instalación le configuraremos una ip estática.  
Y para la instalación del zabbix utilizaremos el siguiente script:

```
#!/bin/bash

echo "Iniciando instalacion de zabbix"

#Instalamos los repositorios de zabbix
wget
https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-release/zabbix-release\_6.4-1+ubuntu22.04\_all.deb

dpkg -i zabbix-release_6.4-1+ubuntu22.04_all.deb
#Actualizamos los repositorios
apt update
#Instalamos los paquetes necesarios para el zabbix.
apt install -y mysql-server zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf
zabbix-sql-scripts zabbix-agent

#Le pedimos la contraseña del usuario zabbix
read -p "Introduce la contraseña del usuario zabbix de la base de datos: " zabbixpass
echo "A continuacion introduce la contraseña"
#Creamos la base de datos de zabbix y el usuario zabbix para que la instalacion por web
#sea automatica (todo next)
mysql -u root <<MYSQL_SCRIPT
create database zabbix character set utf8mb4 collate utf8mb4_bin;
create user zabbix@localhost identified by '$zabbixpass';
```

```
grant all privileges on zabbix.* to zabbix@localhost;
set global log_bin_trust_function_creators = 1;
flush privileges;
MYSQL_SCRIPT
#Descomprimos uno de los ficheros que viene con los paquetes de zabbix
zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4
-u zabbix -p zabbix
#Volvemos a desactivar la credibilidad de los creadores.
mysql -u root <<MYSQL_SCRIPT
set global log_bin_trust_function_creators = 0;
quit;
MYSQL_SCRIPT

#Creamos el archivo de configuracion de zabbix_server
cat <<EOF > /etc/zabbix/zabbix_server.conf

#Ya que el archivo de configuración es muy largo, dejo un link donde tengo subido el script.
https://github.com/GokeOne/zabbix.sh

EOF

#Instalamos la lengua inglesa en las locales del equipo para solucionar un error que puede
#haber en el dashboard al no poder traducir.
apt install -y locales
echo "en_US.UTF-8 UTF-8" >> /etc/locale.gen
locale-gen
#Reiniciamos servicios y los habilitamos
systemctl restart zabbix-server zabbix-agent apache2
systemctl enable zabbix-server zabbix-agent apache2
```

```
Introduce la contraseña del usuario zabbix de la base de datos: 1234
A continuacion introduce la contraseña
Enter password:
```

Al ejecutar el script nos pedirá la contraseña del usuario zabbix que hemos creado con el script y nos la pedirá para descomprimir el archivo .sql.gz

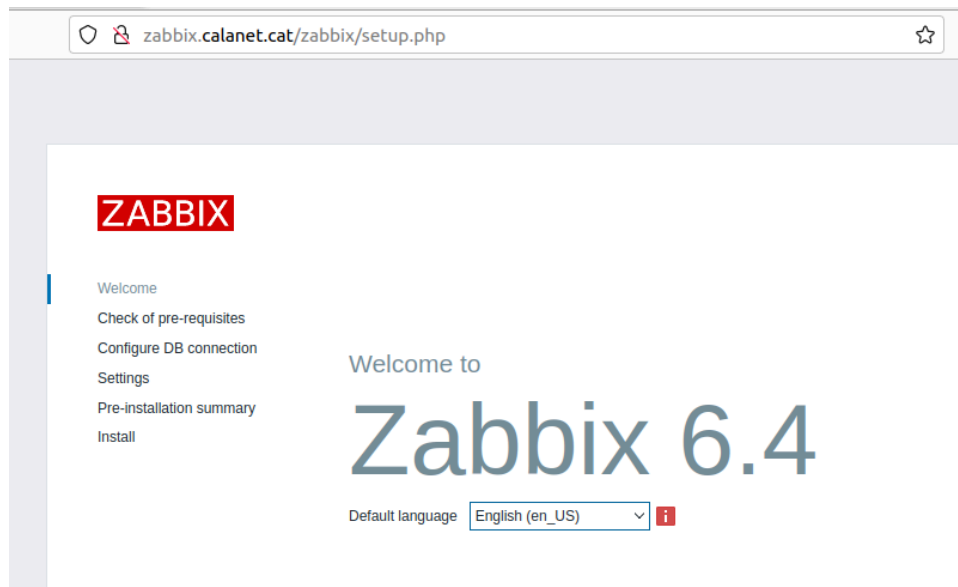
```
• zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; vendor preset: enabled)
   Active: active (running) since Tue 2024-04-09 14:01:13 UTC; 1min 1s ago
     Main PID: 15889 (zabbix_server)
        Tasks: 48 (limit: 2221)
      Memory: 62.6M
         CPU: 382ms
       CGroup: /system.slice/zabbix-server.service
               └─15889 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix.conf
                 └─15891 "/usr/sbin/zabbix_server: ha manager" ""
```

Podemos comprobar con systemctl que el archivo está corriendo.

### Registro del servidor Zabbix en el DNS.

```
smtp    IN      CNAME  mailing
pop3    IN      CNAME  mailing
imap    IN      CNAME  mailing
zabbix  IN      A      192.168.10.225
```

Añadimos el registro en la máquina DNS en /etc/bind/db.calanet.cat



Ahora ya podemos ir al setup.php del zabbix.

## ZABBIX

### Check of pre-requisites

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

	Current value	Required	
PHP version	8.1.2-1ubuntu2.14	7.4.0	OK
PHP option "memory_limit"	128M	128M	OK
PHP option "post_max_size"	16M	16M	OK
PHP option "upload_max_filesize"	2M	2M	OK
PHP option "max_execution_time"	300	300	OK
PHP option "max_input_time"	300	300	OK
PHP databases support	MySQL		OK
PHP bcmath	on		OK
PHP mbstring	on		OK
PHP option "mbstring.func_overload"	off	off	OK

Back

Next step

Chequeamos que todos los prerequisites estén correctamente.

### Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type

Database host

Database port  0 - use default port

Database name

Store credentials in

User

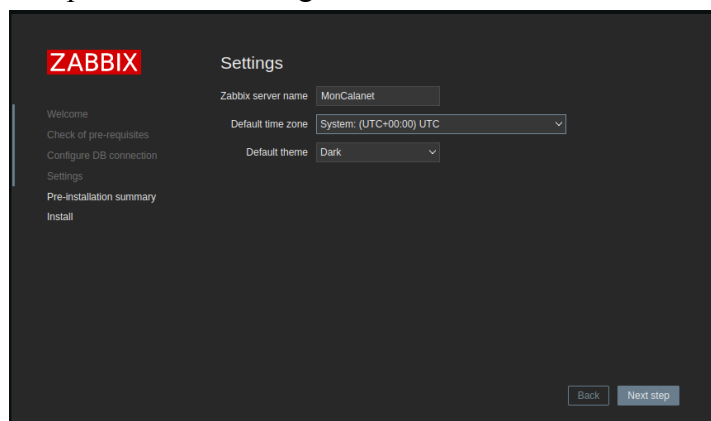
Password

Database TLS encryption *Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).*

Back

Next step

Comprobamos la configuración con la base de datos.



**ZABBIX** Settings

Zabbix server name

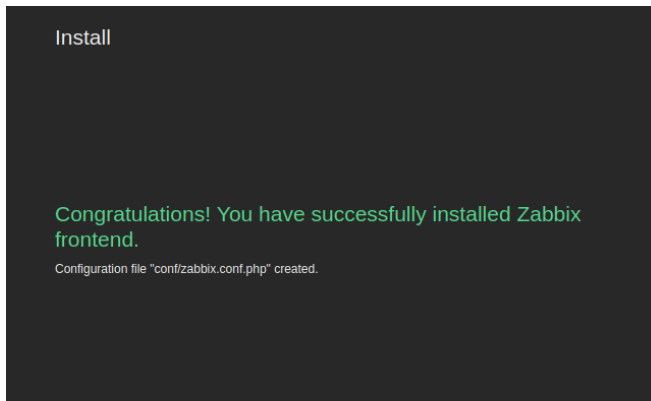
Default time zone

Default theme

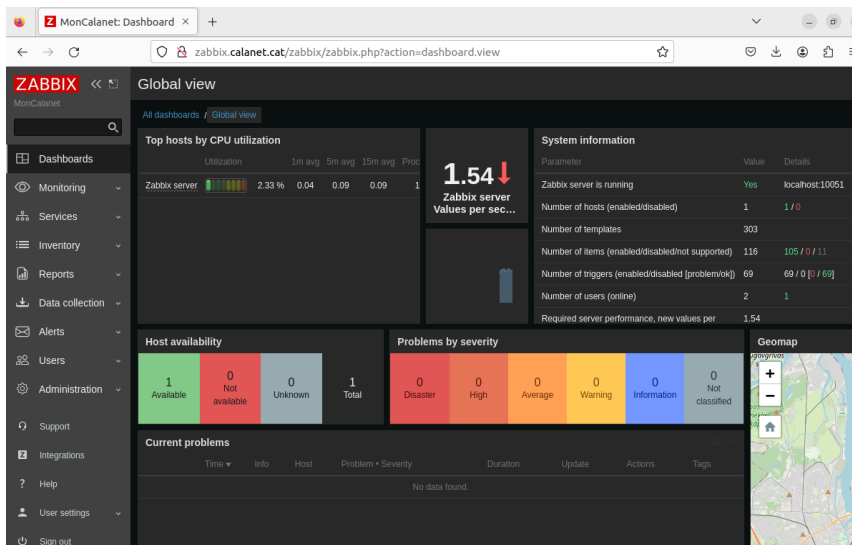
Back Next step

Le introducimos la configuración general.





Le damos a next y llegamos al mensaje donde nos indica que ya tenemos instalado el Zabbix.



Ya podemos entrar al Dashboard.

### Instalación del agente de zabbix:

Para la instalación del agente de zabbix, usaremos el siguiente script:

```
#!/bin/bash
```

```
#Actualizamos repositorios
```

```
apt update
```

```
#Descargamos los repositorios de zabbix
```

```
wget
```

```
https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-release/zabbix-release\_6.4-1+ubuntu22.04\_all.deb
```

```
dpkg -i zabbix-release_6.4-1+ubuntu22.04_all.deb
```

```
apt update
```

```
#Instalamos el paquete del zaabbix agent
```

```
apt install -y zabbix-agent
```

```
#Pedimos la ip del servidor zabbix
```

```
read -p "Introduce la ip del servidor zabbix: " ipzabbixServer
```

```
#Cogemos el nombre del equipo para la configuracion
```

```
nsI=&(hostname)
```

```
systemctl restart zabbix-agent
```

```
systemctl enable zabbix-agent
```

```
cat <<EOF > /etc/zabbix/zabbix_agentd.conf
```

```
PidFile=/run/zabbix/zabbix_agentd.pid
```

```
LogFile=/var/log/zabbix/zabbix_agentd.log
```

```
LogFileSize=0
```

```
# Default:
```

```
ListenPort=10050
```

```
Server=$ipzabbixServer
```

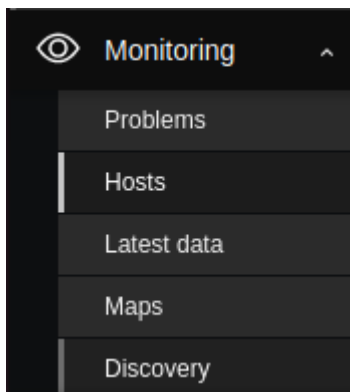
```
HostName=$nsI
```

```
EOF
```

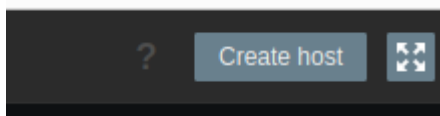
```
#Reiniciamos y habilitamos el servicio
```

```
systemctl restart zabbix-agent
```

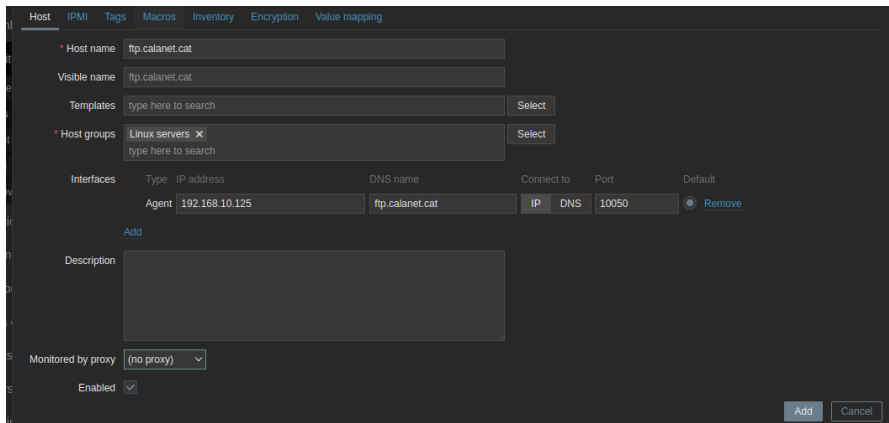
## 12.1 Añadir host en zabbix.



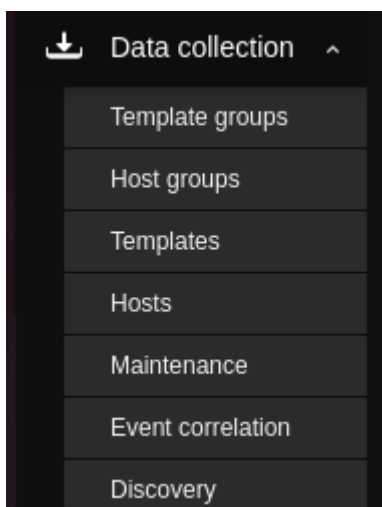
Nos vamos al apartado de monitoring y nos vamos a hosts.



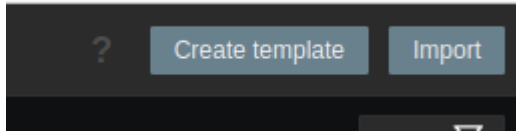
En la derecha arriba de la página podemos ver que hay un create host



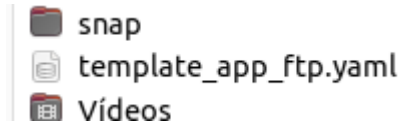
Creamos el host con el nombre de hostname que le hayamos dado al agente.



Nos vamos a Data collection → templates.

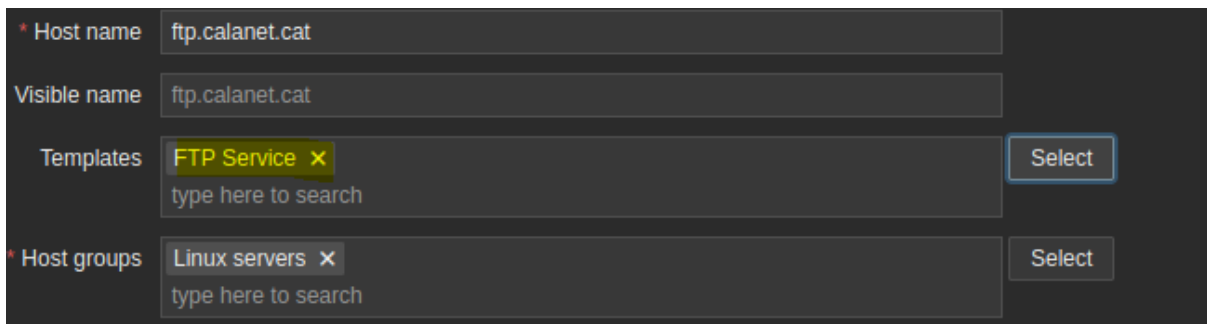


Nos vamos a import template.

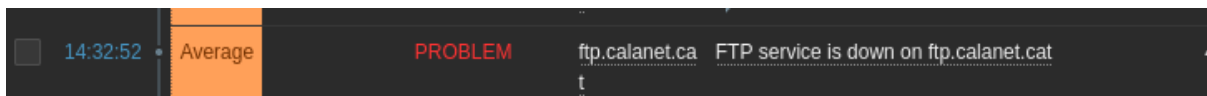


Importamos este archivo.

[https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/app/ftp\\_service?at=release/6.2](https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/app/ftp_service?at=release/6.2)



Le añadimos al host el template que hemos importado.

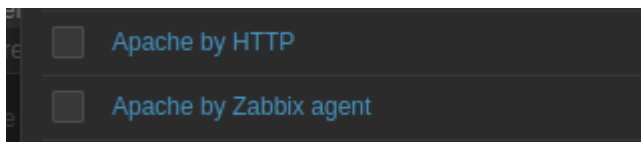


Si paramos el servicio podemos ver que el servicio de FTP está parado.



Y una vez lo activemos nos dirá que está resuelto.

Para hacerlo en otros servicios como por ejemplo apache o nginx, podemos usar templates que ya vienen definidos por defecto en zabbix.



Para apache podemos usar el apache by zabbix agent.



Para nginx podemos usar el nginx by zabbix agent.

### 13. Instalación del proxy inverso.

Para el proxy inverso utilizaremos nginx, un proxy inverso se encarga de redirigir las solicitudes de un servicio a otro servidor. Gracias a esto podemos evitar ataques Ddos que tiren nuestro servidor principal.

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 192.168.20.5/24
      nameservers:
        addresses: [192.168.10.5]
      routes:
        - to: default
          via: 192.168.20.1
```

Le configuramos la ip estática de la zona DMZ.

Para la instalación del nginx utilizaremos el siguiente script:

```
#!/bin/bash

#Actualizacion de repositorios
apt-get update

#Instalacion del paquete nginx
apt-get install -y nginx

#Pedimos la ip del apache
read -p "Introduce la direccion ip tu servidor apache: " ipapache

cat <<EOF > /etc/nginx/nginx.conf

user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}
```

*http {*

```
    upstream apache_servers {  
        ip_hash;  
        server $ipapache;  
    }
```

```
server {  
    listen 80;  
    server_name web.calanet.cat;  
  
    location / {  
        proxy_pass http://apache_servers;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        #Headers de seguridad  
        add_header X-Content-Type-Options nosniff;  
        add_header X-Frame-Options "SAMEORIGIN";  
        add_header X-XSS-Protection "1; mode=block";  
    }  
}
```

```
sendfile on;  
tcp_nopush on;  
types_hash_max_size 2048;
```

```
include /etc/nginx/mime.types;  
default_type application/octet-stream;
```

```
##
```

```
# SSL Settings
```

```
##
```

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
```

```
ssl_prefer_server_ciphers on;

##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;


include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}

EOF
```

```
systemctl restart nginx
```

```
systemctl status nginx
```

```
echo "Proxy inverso instalado"
```

```
NO VM guests are running outdated hypervisor (qemu) binaries on this platform
Introduce la direccion ip tu servidor apache: 192.168.10.12
```

Nos pedirá la ip del servidor web.

**Registro del proxy en el dns:**

```
web      IN      A       192.168.20.5
```

Añadimos la máquina proxy como web en el dns.



Y ya se puede entrar al wordpress utilizando el proxy.

## 14. Herramientas para aumentar medidas de seguridad.

Fail2Ban es una herramienta de seguridad informática que se utiliza para proteger servidores y servicios de red de ataques malintencionados y de fuerza bruta. Fail2Ban funciona monitoreando los registros de acceso y detectando intentos de acceso fallidos. Si un usuario tiene varios intentos fallidos de acceso en un período de tiempo determinado, Fail2Ban puede bloquear el acceso a ese usuario durante un período de tiempo determinado como medida de seguridad.

La herramienta utiliza un conjunto de reglas para determinar cuándo bloquear un usuario. Estas reglas pueden ser personalizadas para adaptarse a las necesidades específicas de cada servidor o servicio.

```
server@server:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  python3-pyinotify whois
Paquetes sugeridos:
  mailx monit sqlite3 python-pyinotify-doc
Se instalarán los siguientes paquetes NUEVOS:
  fail2ban python3-pyinotify whois
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 104 no actualizados.
Se necesita descargar 473 kB de archivos.
Se utilizarán 2.486 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Para instalar fail2ban instalamos el paquete de fail2ban

```
server@server:~$ sudo systemctl start fail2ban
server@server:~$ sudo systemctl enable fail2ban
```

Habilitamos el servicio y lo iniciamos.

```
server@server:~$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
server@server:~$
```



Copiamos la configuración de jail.conf a jail.local

```
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned
bantime = 10m

# A host is banned if it has generated "maxretry" during
# seconds.
findtime = 10m

# "maxretry" is the number of failures before a host gets
# banned.
maxretry = 5

# "maxmatches" is the number of matches stored in tickets
maxmatches = %(maxretry)s

# "maxretry" specifies the hostend used to get failed logs
```

Aquí podemos ver varias opciones de configuración de las muchas que hay. En maxretry podemos configurar los accesos antes de bloquear una conexión.

```
server@server:~$ sudo tail -f /var/log/fail2ban.log
2024-04-09 07:52:29,303 fail2ban.jail [2416]: INFO Creating new jail 'sshd'
2024-04-09 07:52:29,307 fail2ban.jail [2416]: INFO Jail 'sshd' uses pyinotify {}
2024-04-09 07:52:29,308 fail2ban.jail [2416]: INFO Initiated 'pyinotify' backend
2024-04-09 07:52:29,309 fail2ban.filter [2416]: INFO maxLines: 1
2024-04-09 07:52:29,315 fail2ban.filter [2416]: INFO maxRetry: 5
2024-04-09 07:52:29,315 fail2ban.filter [2416]: INFO findtime: 600
2024-04-09 07:52:29,315 fail2ban.actions [2416]: INFO bantime: 600
2024-04-09 07:52:29,315 fail2ban.filter [2416]: INFO encoding: UTF-8
2024-04-09 07:52:29,316 fail2ban.filter [2416]: INFO Added logfile: '/var/log/auth.log' (pos = 0, hash = 258
2281448f75e16711e45e85773b37)
2024-04-09 07:52:29,318 fail2ban.jail [2416]: INFO Jail 'sshd' started
2024-04-09 07:54:47,137 fail2ban.filter [2416]: INFO [sshd] Found 172.16.10.81 - 2024-04-09 07:54:47
2024-04-09 07:55:03,106 fail2ban.filter [2416]: INFO [sshd] Found 172.16.10.81 - 2024-04-09 07:55:02
2024-04-09 07:55:07,005 fail2ban.filter [2416]: INFO [sshd] Found 172.16.10.81 - 2024-04-09 07:55:06
2024-04-09 07:55:08,833 fail2ban.filter [2416]: INFO [sshd] Found 172.16.10.81 - 2024-04-09 07:55:08
2024-04-09 07:55:20,079 fail2ban.filter [2416]: INFO [sshd] Found 172.16.10.81 - 2024-04-09 07:55:19
2024-04-09 07:55:20,501 fail2ban.actions [2416]: NOTICE [sshd] Ban 172.16.10.81
```

Podemos usar ssh para comprobar que funciona, y vemos que al quinto intento fallido se banea la ip durante 10 minutos que es el tiempo que hemos puesto en la configuración. Así podemos evitar ataques de fuerza bruta.

## 15. Anexo

### Instalación de Servidor openvpn:

Requisitos:

Ubuntu 22.04.3

Conexión a internet

- Configuración de red:

```
last login: Tue Dec 12 14:20:12 UTC 2023 on tty1
openvpn@openvpn:~$ sudo nano /etc/netplan/00-installer-config.yaml
[sudo] password for openvpn:
```

Antes de empezar con la configuración del servidor tenemos que asegurarnos de que tiene ip estática para que los clientes puedan llegar al servidor.

```
# This is the network config written by 'subiquity'
network:
  ethernet:
    enp0s3:
      dhcp4: false
      addresses:
        - 10.0.2.6/24
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
      routes:
        - to: default
          via: 10.0.2.1
      version: 2
```

Desactivamos el dhcp, y le ponemos la ip estatica: 10.0.2.6/24, le añadimos dos servidores dns, y la ip de gateway.

```
openvpn@openvpn:~$ sudo netplan apply
openvpn@openvpn:~$ _
```

Ejecutamos ~sudo netplan apply para que se apliquen los cambios de la configuración.

- Instalación de Openvpn con rsa:

```
openvpn@openvpn:~$ sudo apt-get update
[sudo] password for openvpn:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,268 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [260 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1,257 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [205 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,021 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [227 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41,7 kB]
Des:12 http://es.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24,3 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,056 kB]
Des:14 http://es.archive.ubuntu.com/ubuntu jammy-security/main Translation-en [200 kB]
Des:15 http://es.archive.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1,233 kB]
Des:16 http://es.archive.ubuntu.com/ubuntu jammy-security/restricted Translation-en [202 kB]
Des:17 http://es.archive.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [824 kB]
Des:18 http://es.archive.ubuntu.com/ubuntu jammy-security/universe Translation-en [156 kB]
Descargados 8.313 kB en 3s (2.836 kB/s)
Reading package lists... Done
openvpn@openvpn:~$ _
```



```

openvpn@openvpn:~$ sudo apt-get install openvpn easy-rsa
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  libccid libpcsclite1 libpkcs11-helper1 opencsc opencsc-pkcs11 pcscd
Paquetes sugeridos:
  pcmciautils resolvconf openvpn-systemd-resolved
Se instalarán los siguientes paquetes NUEVOS:
  easy-rsa libccid libpcsclite1 libpkcs11-helper1 opencsc opencsc-pkcs11 openvpn pcscd
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 54 no actualizados.
Se necesita descargar 2.152 kB de archivos.
Se utilizarán 6.864 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

```

```
openvpn@openvpn:~$ mkdir easy-rsa
openvpn@openvpn:~$ ln -s /usr/share/easy-rsa/* easy-rsa/
openvpn@openvpn:~$ sudo chown openvpn easy-rsa/
openvpn@openvpn:~$ chmod 700 easy-rsa/
openvpn@openvpn:~$
```

- Creación de la clave PKI para openvpn:

```
GNU nano 6.2 easy-rsa/vars
set_var EASYRSA_ALGO "ec"
set_var EASYRSA_DIGEST "sha512"
```

```
openvpn@openvpn:~$ cd easy-rsa/
openvpn@openvpn:~/easy-rsa$ ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/openvpn/easy-rsa/pki

openvpn@openvpn:~/easy-rsa$
```

- Crear una solicitud de certificado y una clave privada del servidor OpenVPN:

[illegible]

```
/home/openvpn/easy-rsa/pki/private/server.key
```

```

Last login: Mon Jan 10 11:16:53 UTC 2022 on tty1
openvpn@openvpn:~$ sudo cp /home/openvpn/easy-rsa/pki/private/server.key /etc/openvpn/server/
[sudo] password for openvpn:
openvpn@openvpn:~$ _

```

Copiamos la clave a la carpeta del servidor.

-Firmar la solicitud de certificado del servidor OpenVPN

```

openvpn@openvpn:~$ scp /home/openvpn/easy-rsa/pki/reqs/server.req rsaserver@10.0.2.15:/tmp
rsaserver@10.0.2.15's password:
server.req
100% 887 810.2KB/s 00:00
openvpn@openvpn:~$

```

En el paso anterior hemos creado un certificado que tiene que ser reconocido por el CA server.

-----Ca Server-----

```

rsaserver@rsaserver:~$ cd easy-rsa/
rsaserver@rsaserver:~/easy-rsa$ ./easynsa import-req /tmp/server.req server
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

The request has been successfully imported with a short name of: server
You may now use this name to perform signing operations on this request.

rsaserver@rsaserver:~/easy-rsa$

```

En el servidor RSA importamos el certificado para firmar.

```

rsaserver@rsaserver:~/easy-rsa$ ./easynsa sign-req server server
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
  commonName               = server

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /home/rsaserver/easy-rsa/pki/easy-rsa-1178.Ii16w9/tmp.XaA0Pq
Enter pass phrase for /home/rsaserver/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'server'
Certificate is to be certified until Apr 15 14:22:25 2026 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/rsaserver/easy-rsa/pki/issued/server.crt

rsaserver@rsaserver:~/easy-rsa$ _

```

Firmamos la solicitud de certificado del servidor openvpn, la firma puede ser como server o cliente, hay que asegurarse de poner server, esto genera el server.crt que es el certificado firmado por el CA

```
rsaserver@rsaserver:~/easy-rsa$ scp pki/issued/server.crt openvpn@10.0.2.6:/tmp
openvpn@10.0.2.6's password:
server.crt                                100% 4609      6.4MB/s   00:00
rsaserver@rsaserver:~/easy-rsa$ scp pki/ca.crt openvpn@10.0.2.6:/tmp
openvpn@10.0.2.6's password:
ca.crt                                    100% 1204      1.8MB/s   00:00
rsaserver@rsaserver:~/easy-rsa$
```

Una vez firmado pasamos el archivo server.crt y ca.crt del servidor RSA al servidor openvpn.

```
openvpn@openvpn:~$ sudo cp /tmp/server.crt /etc/openvpn/server/
[sudo] password for openvpn:
openvpn@openvpn:~$ sudo cp /tmp/ca.crt /etc/openvpn/server/
openvpn@openvpn:~$ _
```

Una vez tengamos los archivos en el servidor openvpn, los copiamos a la carpeta del servidor /etc/openvpn/server/

#### -Configuración de material criptográfico en openvpn

Para una capa adicional de seguridad, agregaremos una clave secreta compartida adicional que el servidor y todos los clientes usarán con la directiva tls-crypt

```
openvpn@openvpn:~/easy-rsa$ openvpn --genkey secret ta.key
openvpn@openvpn:~/easy-rsa$
```

Nos vamos a la carpeta de easy-rsa/ y con openvpn generamos la clave  
**~openvpn --genkey secret ta.key**

```
openvpn@openvpn:~/easy-rsa$ sudo cp ta.key /etc/openvpn/server/
openvpn@openvpn:~/easy-rsa$
```

Copiamos la clave generada a etc/openvpn/server/

Con estos archivos ya podemos crear los certificados de los clientes.

#### -Generando un certificado de cliente y par de claves

```
openvpn@openvpn:~$ mkdir -p client-configs/keys
openvpn@openvpn:~$ chmod -R 700 client-configs/
openvpn@openvpn:~$
```

Creamos el directorio para la estructura de claves.



En el directorio de easy-rsa generamos el par de claves con el nombre de usuario1 y sin contraseña.

```
openvpn@openvpn:~/easy-rsa$ cp pki/private/usuario1.key ../client-configs/keys/
openvpn@openvpn:~/easy-rsa$ _
```

```

openvpn@openvpn:~/easy-rsa$ scp pki/reqs/usuario1.req rsaserver@10.0.2.15:/tmp
rsaserver@10.0.2.15's password:
usuario1.req                                100% 891      1.6MB/s   00:00
openvpn@openvpn:~/easy-rsa$

```

-----RSA SERVER-----

```
rsaserver@rsaserver:~/easy-rsa$ ./easysrsa import-req /tmp/usuario1.req usuario1
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

The request has been successfully imported with a short name of: usuario1
You may now use this name to perform signing operations on this request.
```

```
rsaserver@rsaserver:~/easy-rsa$ ./easynrsa sign-req client usuario1
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
```

```
rsaserver@rsaserver:~/easy-rsa$ scp pki/issued/user1.crt openvpn@10.0.2.6:/tmp
openvpn@10.0.2.6's password:
user1.crt                                100% 4495      6.4MB/s   00:00
rsaserver@rsaserver:~/easy-rsa$ _
```

```
openvpn@openvpn:~/easy-rsa$ cp /tmp/usuario1.crt ../client-configs/keys/
```

```
openvpn@openvpn:~/client-configs$ cp ../easy-rsa/ta.key keys/
openvpn@openvpn:~/client-configs$
```

61

```
openvpn@openvpn:~/easy-rsa$ sudo cp /etc/openvpn/server/ca.crt ../client-configs/keys/
openvpn@openvpn:~/easy-rsa$ sudo chown openvpn:openvpn ../client-configs/keys/*
openvpn@openvpn:~/easy-rsa$
```

Y también el archivo ca.crt, y cambiamos el usuario y el grupo de los archivos del directorio.

### -Configurando Openvpn

```
openvpn@openvpn:~/easy-rsa$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf
/etc/openvpn/server/
openvpn@openvpn:~/easy-rsa$ _
```

Primero de todo copiamos la configuración simple de openvpn a /etc/openvpn/server

```
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret
tls-crypt ta.key
# Select a cryptographic cipher
```

Editamos el archivo /etc/openvpn/server/server.conf y cambiamos el tipo de cifrado con tls.

```
# See also the tls-cipher option
;cipher AES-256-CBC
cipher AES-256-GCM_
# Enable compression on the
```

Debajo del tls-crypt encontramos el cipher, cambiamos el tipo de cifrado de AES-256-CBC a AES-256-GCM que nos ofrece un mejor nivel de cifrado.

```
auth SHA256_
# Enable
```

Añadimos auth SHA256 debajo del cipher

```
# openssl dhparam
;dh dh2048.pem
dh none_
# Note that this
```

Cambiamos el dh que es el algoritmo diffie-hellman que no lo necesitamos ya que tenemos una curva cerrada de certificados.

### -Configuración de la interconexión del servidor openvpn.

```
# Uncomment the next line to enable
net.ipv4.ip_forward=1
```

Editamos el archivo /etc/sysctl.conf y descomentamos la línea de net.ipv4.ip\_forward=1

```
openvpn@openvpn:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
openvpn@openvpn:~$
```

Para leer el archivo y cargar las nuevas configuraciones ejecutamos sysctl -p

### -Configuración del Firewall.

```
#Reglas del openvpn
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.8.0.0/8 -o enp0s3 -j MASQUERADE
COMMIT
#Final de reglas_
```

En el archivo /etc/ufw/before.rules añadimos al principio del archivo estas reglas.

```
# Set the default forward policy to
# if you change this you will most
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Cambiamos a ACCEPT el forward policy en el archivo /etc/default/ufw

```
openvpn@openvpn:~$ sudo ufw allow 1194/udp
Rules updated
Rules updated (v6)
openvpn@openvpn:~$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
openvpn@openvpn:~$ _
```

Creamos 2 reglas de entrada para permitir el puerto y el servicio

```
openvpn@openvpn:~$ sudo ufw disable
Firewall stopped and disabled on system startup
openvpn@openvpn:~$ sudo ufw enable
Firewall is active and enabled on system startup
openvpn@openvpn:~$
```

Desactivamos y activamos el firewall para que se carguen las nuevas reglas, y ya esta preparado para iniciar el servidor.

-Iniciando el servidor openvpn

```
openvpn@openvpn:~$ sudo systemctl -f enable openvpn-server@server.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service → /lib/systemd/system/openvpn-server@.service.
openvpn@openvpn:~$ _
```

Creamos el demonio con **~sudo systemctl -f enable openvpn-server@server.service**

```
openvpn@openvpn:~$ sudo systemctl start openvpn-server@server.service
openvpn@openvpn:~$ sudo systemctl status openvpn-server@server.service
```

Iniciamos el servicio



```
openvpn@openvpn:~$ sudo systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/lib/systemd/system/openvpn-server@.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-01-11 15:12:51 UTC; 11s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 5065 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 2221)
    Memory: 1.8M
       CPU: 9ms
   CGroup: /system.slice/system-openvpn\x2dservice.slice/openvpn-server@server.service
           └─5065 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-verb
```

Y comprobamos que el servicio este funcionando correctamente.

-Creando la infraestructura de la configuración de los clientes.

```
openvpn@openvpn:~$ mkdir -p client-configs/files
openvpn@openvpn:~$
```

Creamos el directorio para los archivos.

```
openvpn@openvpn:~$ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf client-configs/base.conf
openvpn@openvpn:~$
```

Copiamos la configuración básica de los clientes y la dejamos en client-configs/

```
# You can have multiple remote
# to load balance between the
remote 10.0.2.6_1194
;remote my-server-2 1194
```

Cambiamos el remote my-server-1 1194 a nuestra ip del servidor.

```
# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody
```

Para los usuario de windows descomentamos estas 2 lineas.

```
# file can be used for a
;ca ca.crt
;cert client.crt
;key client.key
```

Comentamos estas lineas ya que las llevara el archivo con el que nos conectemos.

```
# If a tls-auth key is use
# then every client must a
;tls-auth ta.key 1
```

Hacemos lo mismo con el archivo ta.key

```
# See also the data-ciph
cipher AES-256-GCM
auth SHA256
```

Y cambiamos el tipo de cifrado a AES-256-GCM y añadimos el auth SHA256

```
key-direction 1_
# Enable compression
```

Añadimos el key-direction para que funcione correctamente el openvpn.

```
; script-security 2
; up /etc/openvpn/update-resolv-conf
; down /etc/openvpn/update-resolv-conf

; script-security 2
; up /etc/openvpn/update-systemd-resolved
; down /etc/openvpn/update-systemd-resolved
; down-pre
; dhcp-option DOMAIN-ROUTE .
```

Finalmente añadimos estas líneas comentadas al final de todo para poder evitar fácilmente diferentes problemas.

```
GNU nano 6.2 client-configs/make_config.sh
#!/bin/bash

#Primer argumento: Identificado del cliente

KEY_DIR=/client-configs/keys
OUTPUT_DIR=/client-configs/files
BASE_CONFIG=/client-configs/base.conf

cat ${BASE_CONFIG} \
<(echo -e '<ca>' ) \
${KEY_DIR}/ca.crt \
<(echo -e '</ca>\n<cert>' ) \
${KEY_DIR}/${1}.crt \
<(echo -e '</cert>\n<key>' ) \
${KEY_DIR}/${1}.key \
<(echo -e '</key>\n<tls-crypt>' ) \
${KEY_DIR}/ta.key \
<(echo -e '</tls-crypt>' ) \
> ${OUTPUT_DIR}/${1}.ovpn
```

Creamos el script para hacer el archivo .ovpn con la configuración del cliente y las claves. Este script lo que hace es guardar en tres variables lo directorios donde estan las claves y la configuración. Y con un cat vamos añadiendo en el base.conf el certificado, y {1} es el primer parametro que le pasaremos al script, que sera el nombre que le hayamos dado al certificado/usuario

```
openvpn@openvpn:~$ chmod 700 client-configs/make_config.sh
openvpn@openvpn:~$ _
```

Cambiamos los permisos de este script.

```
openvpn@openvpn:~/client-configs$ ./make_config.sh usuario1
openvpn@openvpn:~/client-configs$ ls files/
client1.ovpn  usuario1.ovpn
openvpn@openvpn:~/client-configs$ _
```

Ejecutamos el script con el nombre del usuario y sacamos el archivo .ovpn

## Instalacion de Servidor RSA:

## Instalación de easy-rsa

```
rsaserver@rsaserver:~$ sudo apt-get update
[sudo] password for rsaserver:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Des:5 http://es.archive.ubuntu.com/ubuntu jammy/main Translation-es [332 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy/restricted Translation-es [964 B]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy/universe Translation-es [1.356 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy/multiverse Translation-es [68,2 kB]
Descargados 1.758 kB en 4s (467 kB/s)
Reading package lists... Done
rsaserver@rsaserver:~$ sudo apt-get install easy-rsa
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  libccid libpcsc-lite openssl-pkcs11 pcscd
Paquetes sugeridos:
  pcminitools
Se instalarán los siguientes paquetes NUEVOS:
  easy-rsa libccid libpcsc-lite openssl-pkcs11 pcscd
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 43 no actualizados.
Se necesita descargar 1.484 kB de archivos.
Se utilizarán 5.038 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Ejecutamos un `sudo apt-get update` para actualizar los repositorios, y `sudo apt-get install easy-rsa` para instalar el paquete.

- Directorio de infraestructura de la clave publica.

```
rsaserver@rsaserver:~$ mkdir easy-rsa
rsaserver@rsaserver:~$ ln -s /usr/share/easy-rsa/* easy-rsa/
rsaserver@rsaserver:~$ _
```

Para preparar el directorio que contendrá la infraestructura de la clave publica, creamos la carpeta `easy-rsa` y hacemos un link de la carpeta `/usr/share/easy-rsa/*` Donde están todas las utilidades del paquete que hemos instalado.

```
rsaserver@rsaserver:~$ chmod 700 /home/rsaserver/easy-rsa/
rsaserver@rsaserver:~$ _
```

Para restringir el acceso a la carpeta cambiamos los permisos de nuestro directorio PKI

```
rsaserver@rsaserver:~/easy-rsa$ ./easynsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/rsaserver/easy-rsa/pki

rsaserver@rsaserver:~/easy-rsa$ _
```

Iniciamos el directorio pki

- Creando certificado de autoridad.

```
GNU nano 6.2 vars
set_var EASYRSA_REQ_COUNTRY "US"
set_var EASYRSA_REQ_PROVINCE "NewYork"
set_var EASYRSA_REQ_CITY "New York City"
set_var EASYRSA_REQ_ORG "DigitalOcean"
set_var EASYRSA_REQ_EMAIL "admin@example.com"
set_var EASYRSA_REQ_OU "Community"
set_var EASYRSA_ALGO "ec"
set_var EASYRSA_DIGEST "sha512"
```

Para empezar necesitamos un archivo que se llama vars con unas configuraciones básicas, en la carpeta que hemos creado de easy-rsa, con nano creamos este fichero.

```
rsaserver@rsaserver:~/easy-rsa$ ./easysrsa build-ca
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/rsaserver/easy-rsa/pki/ca.crt

rsaserver@rsaserver:~/easy-rsa$ _
```

Una vez tengamos los vars, creamos el CA, si queremos que no contenga contraseña para cada vez que interactuemos con el. Hay que ejecutar `~/easysrsa build-ca nopass`.

Ahora tenemos dos archivos importantes:

- easy-rsa/pki/ca.crt
- easy-rsa/pki/private/ca.key

El ca.crt es el archivo de certificado público, que usuarios, servidores y clientes, utilizarán para verificar que forman parte de una misma red de confianza. Cada usuario que utilice el CA tendrá que tener una copia de este archivo. Todas las partes confiarán en el certificado público para garantizar que no se haga un ataque de Man-In-the-middle.

El ca.key es la clave privada que se utiliza para firmar certificados de servidores y clientes. Si un atacante obtiene acceso al ca.key se tendrá que cambiar toda la certificación Este archivo solo tiene que estar en una máquina idealmente apagada cuando no esté realizando ninguna firma.

- Creación de solicitudes de firma de certificados y revocación de certificados:

Para hacer una firma de certificados, o certificate signing request (CSR), consiste en tres partes, clave pública, identificación sobre el sistema que hace la solicitud, y una firma en sí la cual se hace con la clave privada de cada usuario.

```
openvpn@openvpn:~$ sudo apt-get update
[sudo] password for openvpn:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Des:5 http://es.archive.ubuntu.com/ubuntu jammy/main Translation-es [332 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy/restricted Translation-es [964 B]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy/universe Translation-es [1,356 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy/multiverse Translation-es [68,2 kB]
Descargados 1.758 kB en 2s (886 kB/s)
Reading package lists... Done
openvpn@openvpn:~$ sudo apt-get install openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl ya está en su versión más reciente (3.0.2-0ubuntu1.12).
fijado openssl como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 43 no actualizados.
openvpn@openvpn:~$
```

En el servidor openvpn, ejecutamos un `~apt-get update` y un `~ap-get install openssl` para actualizar repositorios e instalar los paquetes de openssl si no los tenemos.

```
openvpn@openvpn:~$ mkdir practice-csr
openvpn@openvpn:~$ cd practice-csr/
openvpn@openvpn:~/practice-csr$ openssl genrsa -out openvpn-server.key
openvpn@openvpn:~/practice-csr$ _
```

Creamos la carpeta donde estarán las claves de los usuarios, y generamos la primera clave para el servidor openvpn.

```
openvpn@openvpn:~/practice-csr$ openssl req -new -key openvpn-server.key -out openvpn-server.req
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:New York
Locality Name (eg, city) []:New York City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DigitalOcean
Organizational Unit Name (eg, section) []:Community
Common Name (e.g. server FQDN or YOUR name) []:openvpn-server
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1234
An optional company name []:
openvpn@openvpn:~/practice-csr$ _
```

Creamos el certificado de firma ahora que tenemos la clave privada.

```
openvpn@openvpn:~/practice-csr$ scp openvpn-server.req rsaserver@10.0.2.15:/tmp/openvpn-server.req
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ED25519 key fingerprint is SHA256:u7ZKUGlt20UFsgQsokQKf7HmU+EfoFwBxVCGTh31zQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.15' (ED25519) to the list of known hosts.
rsaserver@10.0.2.15's password:
openvpn-server.req                               100% 1062    1.2MB/s   00:00
openvpn@openvpn:~/practice-csr$ _
```

Pasamos el certificado de CSR que hemos creado al servidor CA.

```
rsaserver@rsaserver:~/easy-rsa$ ./easyrsa import-req /tmp/openvpn-server.req openvpn-server
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

The request has been successfully imported with a short name of: openvpn-server
You may now use this name to perform signing operations on this request.

rsaserver@rsaserver:~/easy-rsa$
```

Con easyrsa, importamos el .req.

```
rsaserver@rsaserver:~/easy-rsa$ ./easyrsa sign-req server openvpn-server
```

Firmamos el archivo, ponemos server ya que es un certificado para el servidor openvpn, si fuese un cliente pondremos client.

```
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName       :PRINTABLE:'US'
stateOrProvinceName :ASN.1 12:'New York'
localityName      :ASN.1 12:'New York City'
organizationName   :ASN.1 12:'DigitalOcean'
organizationalUnitName:ASN.1 12:'Community'
commonName        :ASN.1 12:'openvpn-server'
Certificate is to be certified until Mar 16 15:12:53 2026 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/rsaserver/easy-rsa/pki/issued/openvpn-server.crt
```

Y nos crea un certificado en easy-rsa/pki/issued/openvpn-server.crt

Ahora ya hemos firmado el openvpn-server.req CSR, usando el la clave privada del servidor rsa /pki/private/ca.key, y nos da el openvpn-server.crt, ahora solo queda pasar los archivos openvpn-server.crt y el /pki/ca.crt al servidor openvpn.

```
rsaserver@rsaserver:~/easy-rsa$ scp pki/issued/openvpn-server.crt openvpn@10.0.2.6:/tmp
The authenticity of host '10.0.2.6 (10.0.2.6)' can't be established.
ED25519 key fingerprint is SHA256:tCMk+Q9eibN/MpIEk+vQE2c/3Ym1Pb+v5RTjlrjFRCY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.6' (ED25519) to the list of known hosts.
openvpn@10.0.2.6's password:
openvpn-server.crt                                100% 4849      4.8MB/s   00:00
rsaserver@rsaserver:~/easy-rsa$ scp pki/ca.crt openvpn@10.0.2.6:/tmp
openvpn@10.0.2.6's password:
ca.crt                                             100% 1204      1.6MB/s   00:00
rsaserver@rsaserver:~/easy-rsa$ _
```

Pasamos los archivos al servidor openvpn.

## Instalación de la máquina bastión

Para la máquina bastión utilizaremos un ArchLinux, ya que es una máquina de bajo coste.

Para la instalación:

Características extendidas: ☒ Habilitar I/O APIC  
☒ Habilitar reloj hardware en tiempo UTC  
☒ Habilitar EFI (sólo SO especiales)  
☐ Habilitar Secure Boot

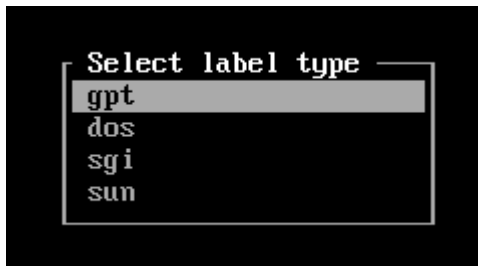
Creamos la MV y le habilitamos el EFI.

~lsblk

```
root@archiso ~ # lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0  7:0    0   671M  1 loop /run/archiso/airootfs
sda     8:0    0    10G  0 disk
sr0    11:0    1 782.3M  0 rom  /run/archiso/bootmnt
root@archiso ~ # _
```

Vemos todos los discos que hay en el sistema.

~cfdisk



Tipo de etiqueta gpt

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	2099199	2097152	1G	Linux swap
/dev/sda2	2099200	3123199	1024000	500M	EFI System
>> /dev/sda3	3123200	20969471	17846272	8.5G	Linux filesystem

Creamos las diferentes particiones, 1G lo dejamos para partición de swap, 500M para la partición EFI. Y el resto lo dejamos para el sistema de archivos.

```

root@archiso ~ # mkswap /dev/sda1
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=c6d8e26f-8131-4bfe-b4ac-f9491e90f0f2
root@archiso ~ # mkfs.fat -F 32 /dev/sda2
mkfs.fat 4.2 (2021-01-31)
root@archiso ~ # mkfs.ext4 /dev/sda3
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 2230784 4k blocks and 558624 inodes
Filesystem UUID: abec80ea-5dbf-41cd-bc4d-efcfec2be6f0
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ #

```

Formateamos las particiones.

~mkswap /dev/sda1 -Activamos el espacio para swap.

~mkfs.fat -F 32 /dev/sda2 -Formateamos la partición de EFI

~mkfs.ext3 /dev/sda3 -Formateamos la partición del sistema de archivos con ext4

```

root@archiso ~ # swapon /dev/sda1
root@archiso ~ # mount /dev/sda3 /mnt
root@archiso ~ # mount --mkdir /dev/sda2 /mnt/boot
root@archiso ~ #

```

Con swapon ponemos en marcha la partición de swap. Montamos la partición del sistema y la partición de efi dentro del sistema y en la carpeta /boot.

```

root@archiso ~ # pacstrap -K /mnt base linux linux-firmware

```

Instalamos los paquetes en el sistema de archivos para tener el firmware de linux.

```

root@archiso ~ # genfstab -U /mnt >> /mnt/etc/fstab
root@archiso ~ #

```

Añadimos las particiones que tenemos montadas en el /etc/fstab del sistema para que reconozca las particiones.

```
root@archiso ~ # arch-chroot /mnt
[root@archiso /]# passwd
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]#
```

Entramos en el sistema con arch-chroot y le cambiamos la contraseña al usuario root.

```
[root@archiso /]# pacman -S nano grub efibootmgr
resolving dependencies...
looking for conflicting packages...

Packages (4) efivar-39-1 efibootmgr-18-3 grub-2:2.12-2 nano-7.2-1

Total Download Size:    7.60 MiB
Total Installed Size:  36.80 MiB

:: Proceed with installation? [Y/n]
```

Le decimos que instale grub, nano y efibootmgr.

```
[root@archiso /]# grub-install --target=x86_64-efi --efi-directory=/boot
Installing for x86_64-efi platform.
Installation finished. No error reported.
[root@archiso /]#
```

Instalamos el grub, y marcamos el directorio donde tenemos montada la partición efi

```
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
[root@archiso /]#
```

Creamos el archivo de grub.cfg con grub-mkconfig para que funcione el grub.

```
[root@archiso /]# ln -sf /usr/share/zoneinfo/Europe/Madrid /etc/localtime
[root@archiso /]# hwclock --systohc
[root@archiso /]# locale-gen
Generating locales...
Generation complete.
[root@archiso /]#
```

Configuraciones básicas para la hora y el idioma del sistema.

```
#es_EC ISO-8859-1
es_ES.UTF-8 UTF-8
#es ES ISO-8859-1
```

Descomentamos el idioma castellano.

```
Arch Linux 6.8.9-arch1-1 (tty1)

archlinux login: root
Password:
[root@archlinux ~]#
```

Iniciamos el sistema.



```
[root@archlinux ~]# ip a add 192.168.10.252/24 broadcast + dev enp0s3
[root@archlinux ~]#
```

Configuramos una ip estática.

```
[root@archlinux ~]# ip link set enp0s3 up
[root@archlinux ~]# ip route add default via 192.168.10.1 dev enp0s3
[root@archlinux ~]#
```

Activamos la interfaz y añadimos la ruta para salir a internet.

```
[root@bastion ~]# cat /etc/resolv.conf
# Resolver configuration file.
# See resolv.conf(5) for details.
nameserver 192.168.10.5

[root@bastion ~]#
```

Le marcamos el nameserver en /etc/resolv.conf

```
[root@bastion ~]# pacman -Sy
:: Synchronizing package databases...
core                               121.9 KiB   185 KiB/s 00:01 (#####) 100
extra                             7.0 KiB   3.40 KiB/s 00:02 (#####) 100
[root@bastion ~]# cat /etc/resolv.conf
# Resolver configuration file.
# See resolv.conf(5) for details.
nameserver 192.168.10.5

[root@bastion ~]#
```

Y ya tenemos salida a internet y podemos instalar los paquetes necesarios para el entorno de escritorio.

```
.../10? including system bus configuration...
[root@archlinux ~]# pacman -S xfce4 xfce4-goodies lightdm lightdm-gtk-greeter
:: There are 15 members in group xfce4:
:: Repository extra
   1) exo  2) garcon  3) thunar  4) thunar-volman  5) tumbler  6) xfce4-appfinder  7) xfce4-pa
  11) xfce4-terminal 12) xfconf 13) xfdesktop 14) xfwm4 15) xfwm4-themes
Enter a selection (default=all):
```

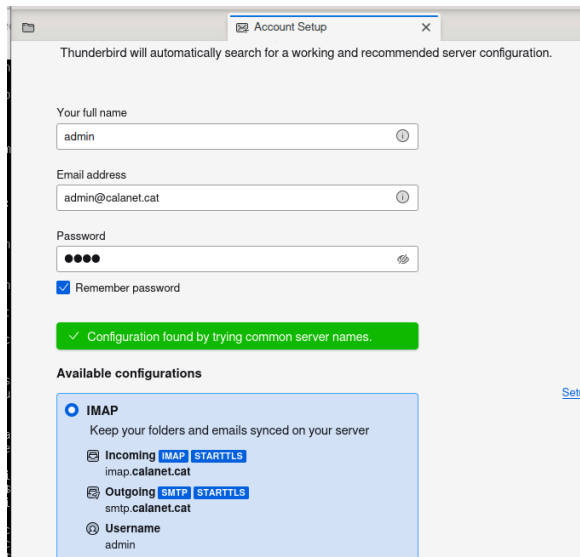
Descargamos los paquetes necesarios para el entorno de escritorio.

```
[root@archlinux ~]# systemctl enable lightdm
```

Habilitamos el entorno de inicio de sesión.

```
[root@archlinux ~]# pacman -S thunderbird
resolving dependencies...
:: There are 10 providers available for ttf-font:
```

Instalamos thunderbird para gestionar el correo electrónico.



Y nos conectamos con nuestro correo electrónico de calanet.cat

## Instalación de Samba DC

### CONFIGURACIÓN SERVIDOR

Cambiar hostname

```
sudo hostnamectl set-hostname dc
```

Modificar fichero hosts

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

```
10.0.0.21 dc.calanet.com dc
```

```
10.0.0.21 calanet.com calanet
```

Verificar el FQDN

```
hostname -f
```

```
root@ubuntu:/home/saul# hostname -f
dc.calanet.com
```

Verificar si el FQDN es capaz de resolver la dirección Ip del Samba

```
ping -c2 dc.calanet.com
```

```
root@ubuntu:/home/saul# ping -c2 dc.calanet.com
PING dc.calanet.com (10.0.0.21) 56(84) bytes of data:
64 bytes from dc.calanet.com (10.0.0.21): icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from dc.calanet.com (10.0.0.21): icmp_seq=2 ttl=64 time=0.091 ms

--- dc.calanet.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1055ms
rtt min/avg/max/mdev = 0.027/0.059/0.091/0.032 ms
```

Desactivar servicio systemd-resolved  
`sudo systemctl disable --now systemd-resolved`

Eliminar enlace simbólico al archivo `/etc/resolv.conf`  
`sudo unlink /etc/resolv.conf`

Creamos de nuevo el archivo `/etc/resolv.conf`  
`sudo nano /etc/resolv.conf`

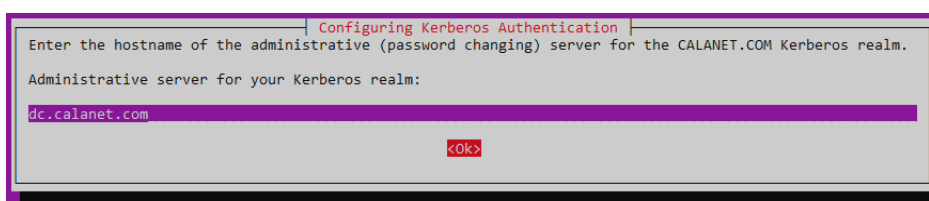
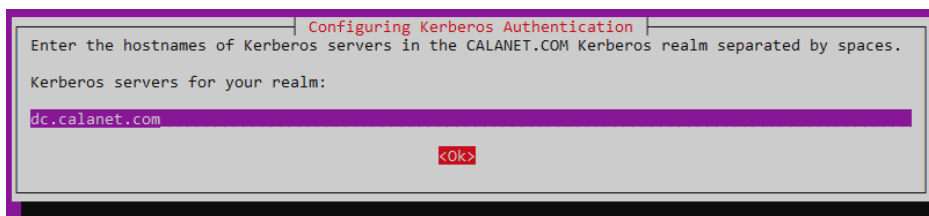
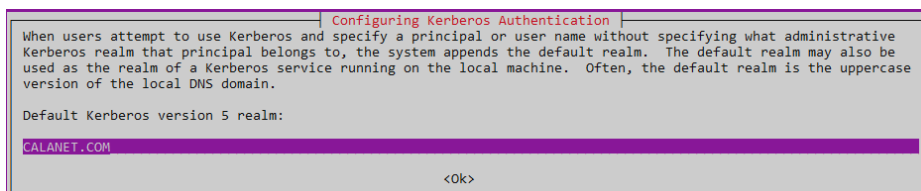
Añadimos las siguientes líneas:  
`nameserver 192.168.10.15`  
`nameserver 1.1.1.1`  
`search calanet.com`

Hacemos inmutable al archivo `/etc/resolv.conf` para que no pueda cambiar  
`sudo chattr +i /etc/resolv.conf`  
**INSTALACIÓN SAMBA**

Actualizar el índice de paquetes  
`sudo apt update`

Instalar samba con sus paquetes y dependencias  
`sudo apt install -y acl attr samba samba-dsdb-modules samba-vfs-modules smbclient winbind libpam-winbind libnss-winbind libpam-krb5 krb5-config krb5-user dnsutils chrony net-tools`

**CALANET.COM**  
`dc.calanet.com`



Detener y deshabilitar los servicios que el servidor de Active Directory de Samba no requiere (smbd, nmbd y winbind)

```
sudo systemctl disable --now smbd nmbd winbind
```

El servidor solo necesita samba-ad-dc para funcionar como Active Directory y controlador de dominio.

```
sudo systemctl unmask samba-ad-dc
```

```
sudo systemctl enable samba-ad-dc
```

## CONFIGURACIÓN SAMBA ACTIVE DIRECTORY

Crear una copia de seguridad del archivo /etc/samba/smb.conf

```
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.orig
```

Ejecutar el comando samba-tool para comenzar a aprovisionar Samba Active Directory.

```
sudo samba-tool domain provision
```

Realm: CALANET.COM

Domain: CALANET

Server Role: dc

DNS backend: SAMBA\_INTERNAL

DNS forwarder IP address: 1.1.1.1

```
root@dc:/home/saul# samba-tool domain provision
Realm [CALANET.COM]:
Domain [CALANET]:
Server Role (dc, member, standalone) [dc]:
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:
DNS forwarder IP address (write 'none' to disable forwarding) [10.0.0.21]: 1.1.1.1
Administrator password:
Retype password:
```

Crear copia de seguridad de la configuración predeterminada de Kerberos.

```
sudo mv /etc/krb5.conf /etc/krb5.conf.orig
```

Reemplazar con el archivo /var/lib/samba/private/krb5.conf.

```
sudo cp /var/lib/samba/private/krb5.conf /etc/krb5.conf
```

Iniciar servicio Samba Active Directory samba-ad-dc

```
sudo systemctl start samba-ad-dc
```

Comprobar servicio

```
sudo systemctl status samba-ad-dc
```

lines 1-29

```

* chrony.service - chrony, an NTP client/server
   Loaded: loaded (/lib/systemd/system/chrony.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-05-14 14:38:41 UTC; 4s ago
     Docs: man:chronyd(8)
           man:chronyc(1)
           man:chrony.conf(5)
  Process: 4527 ExecStart=/usr/lib/systemd/scripts/chronyd-starter.sh $DAEMON_OPTS (code=exited, status=0/SUCCESS)
 Main PID: 4537 (chronyd)
    Tasks: 2 (limit: 4558)
   Memory: 1.3M
      CPU: 190ms
   CGroup: /system.slice/chrony.service
           └─4537 /usr/sbin/chronyd -F 1
             4538 /usr/sbin/chronyd -F 1

May 14 14:38:41 dc systemd[1]: Starting chrony, an NTP client/server...
May 14 14:38:41 dc chronyd[4537]: chronyd version 4.2 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER +SIGN
May 14 14:38:41 dc chronyd[4537]: Frequency 19.727 +/- 8.717 ppm read from /var/lib/chrony/chrony.drift
May 14 14:38:41 dc chronyd[4537]: Using right/UTC timezone to obtain leap second data
May 14 14:38:41 dc chronyd[4537]: MS-SNTP authentication enabled
May 14 14:38:41 dc chronyd[4537]: Loaded seccomp filter (level 1)
May 14 14:38:41 dc systemd[1]: Started chrony, an NTP client/server.

```

## VERIFICAR SAMBA ACTIVE DIRECTORY

Verificar nombres de dominio

host -t A calanet.com

```
calanet.com has address 10.0.0.21
```

host -t A dc.calanet.com

```
dc.calanet.com has address 10.0.0.21
```

Verificar que los registros de servicio kerberos y ldap apunten al FQDN de su servidor Samba Active Directory.

host -t SRV \_kerberos.\_udp.calanet.com

host -t SRV \_ldap.\_tcp.calanet.com

```

root@dc:/home/saul# host -t SRV _kerberos._udp.calanet.com
_kerberos._udp.calanet.com has SRV record 0 100 88 dc.calanet.com.
root@dc:/home/saul# host -t SRV _ldap._tcp.calanet.com
_ldap._tcp.calanet.com has SRV record 0 100 389 dc.calanet.com.

```

Verificar los recursos predeterminados disponibles en Samba Active Directory.

smbclient -L calanet.com -N

```

Anonymous login successful

  Sharename      Type            Comment
  -----
  sysvol         Disk
  netlogon       Disk
  IPC$           IPC             IPC Service (Samba 4.15.13-Ubuntu)
SMB1 disabled -- no workgroup available

```

Comprobar autenticación en el servidor de Kerberos mediante el administrador de usuarios

kinit administrator@CALANET.COM

klist

```
root@dc:/home/saul# kinit administrator@CALANET.COM
Password for administrator@CALANET.COM:
Warning: Your password will expire in 41 days on Tue Jun 25 14:28:29 2024
root@dc:/home/saul# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@CALANET.COM

Valid starting    Expires          Service principal
05/14/24 14:40:27 05/15/24 00:40:27  krbtgt/CALANET.COM@CALANET.COM
renew until 05/15/24 14:40:21
```

Iniciar sesión en el servidor a través de smb

sudo smbclient //localhost/netlogon -U 'administrator'

```
root@dc:/home/saul# smbclient //localhost/netlogon -U 'administrator'
Password for [CALANET\administrator]:
Try "help" to get a list of possible commands.
smb: \>
```

Cambiar contraseña usuario administrator

sudo samba-tool user setpassword administrator

Verificar la integridad del archivo de configuración de Samba.

testparm

```
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed

Server role: ROLE_ACTIVE_DIRECTORY_DC

Press enter to see a dump of your service definitions

# Global parameters
[global]
    dns forwarder = 1.1.1.1
    passdb backend = samba_dsdb
    realm = CALANET.COM
    server role = active directory domain controller
    workgroup = CALANET
    rpc_server:tcpip = no
    rpc_daemon:spoolssd = embedded
    rpc_server:spoolss = embedded
    rpc_server:winreg = embedded
    rpc_server:ntsvcs = embedded
    rpc_server:eventlog = embedded
    rpc_server:svcsvc = embedded
    rpc_server:svcctl = embedded
    rpc_server:default = external
    winbind:use external pipes = true
    idmap config * : backend = tdb
    map archive = No
    vfs objects = dfs_samba4 acl_xattr
```

Verificar funcionamiento WINDOWS AD DC 2008

sudo samba-tool domain level show

```
Domain and forest function level for domain 'DC=calanet,DC=com'

Forest function level: (Windows) 2008 R2
Domain function level: (Windows) 2008 R2
Lowest function level of a DC: (Windows) 2008 R2
```

Crear usuario SAMBA AD

sudo samba-tool user create saul

```
root@dc:/home/saul# samba-tool user create saul
New Password:
Retype Password:
User 'saul' added successfully
root@dc:/home/saul#
```

Listar usuarios SAMBA AD

sudo samba-tool user list

```
root@dc:/home/saul# samba-tool user list
krbtgt
Administrator
Guest
saul
```

Eliminar un usuario

samba-tool user delete <nombre\_del\_usuario>

Listar equipos SAMBA AD

sudo samba-tool computer list

Eliminar equipo SAMBA AD

sudo samba-tool computer delete <nombre\_del\_equipo>

Crear grupo

samba-tool group add <nombre\_del\_grupo>

Listar grupos

samba-tool group list

Listar miembros de un grupo

samba-tool group listmembers 'Domain Admins'

Agregar un miembro a un grupo

samba-tool group addmembers <nombre\_del\_grupo> <nombre\_del\_usuario>

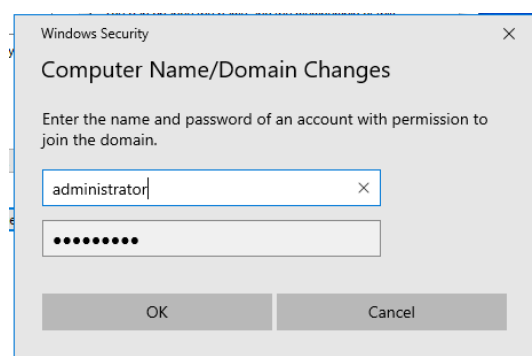
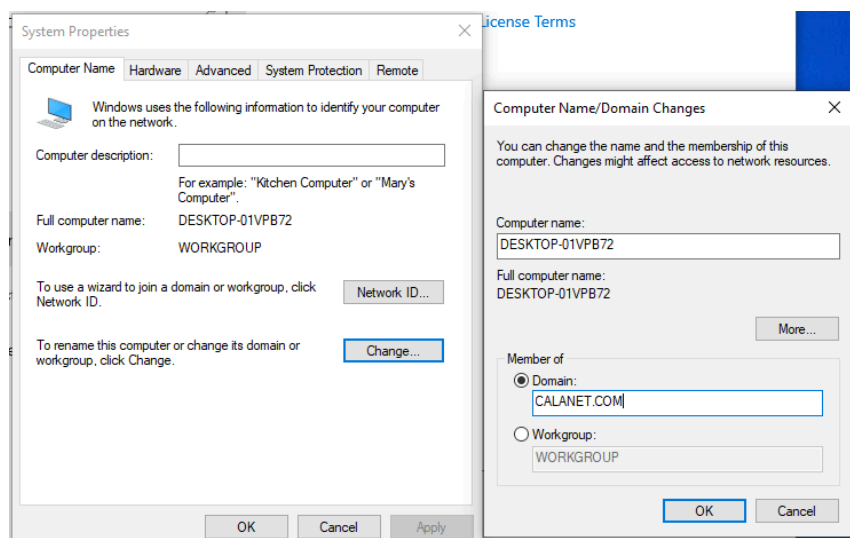
Eliminar un miembro de un grupo

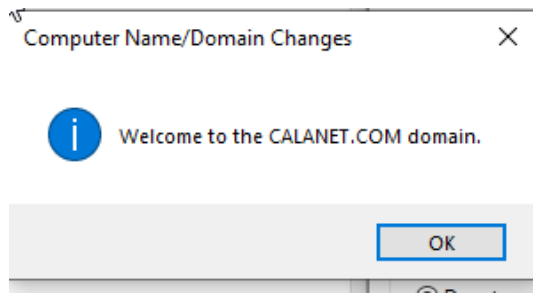
samba-tool group removemembers <nombre\_del\_grupo> <nombre\_del\_usuario>

UNIRSE AL DOMINIO DE SAMBA ACTIVE DIRECTORY

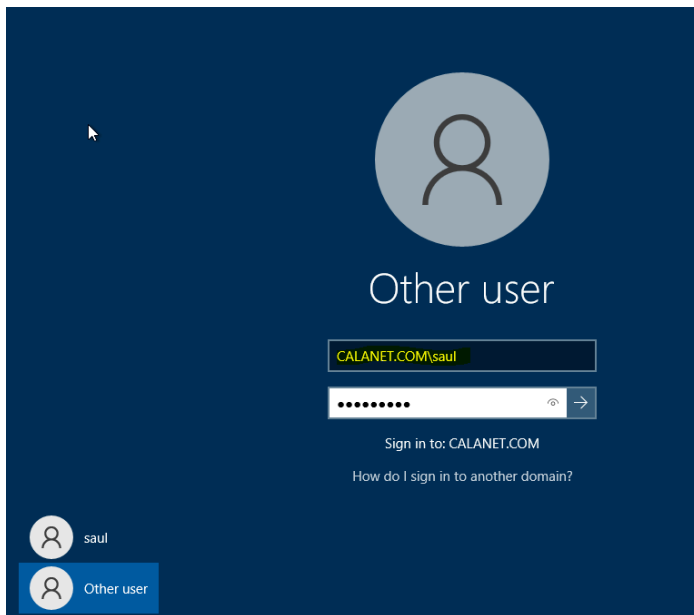
Windows 10:







Reiniciamos



Y podemos entrar en el dominio de samba.

## 16. Conclusión

Desarrollando este proyecto hemos encontrado diferentes obstáculos lo que nos ha llevado a encontrar soluciones, que es lo que buscamos ofrecer a nuestros clientes. Gracias a la automatización y a la gran cantidad de servicios que ofrecemos, podemos satisfacer las necesidades de muchas empresas. Este proyecto muestra la manera de montar la infraestructura de calanet.cat, y la posibilidad de poder adaptar una infraestructura añadiendo o quitando diferentes servicios, para poder reforzar la idea se podrían implementar el uso de más tecnologías como: docker, AWS, diferentes ERP, mikrotik...

## 17. Bibliografía

Conocimientos adquiridos en:

INS Mediterrania

IES Torre Roja

IES El Calamot

DigitalOcean:

<https://www.digitalocean.com/community/tutorials/how-to-set-up-and-configure-an-openvpn-server-on-ubuntu-20-04>

Zabbix:

<https://www.zabbix.com/la/download>

Flota Digital:

<https://flotadigital.com/tutoriales2/instalar-y-configurar-samba-en-ubuntu-20-04/>

ArchLinux:

[https://wiki.archlinux.org/title/installation\\_guide](https://wiki.archlinux.org/title/installation_guide)

Uusario debian:

<https://usuariodebian.blogspot.com/2019/04/samba-4-como-controlador-de-dominios-ad.html>

Clockwork Computer:

<https://www.patreon.com/ClockworkComputer/posts>