



Contenedores con servicios de red en Proxmox

Ciclo Superior de Administración de Sistemas Informáticos en Redes

IES Medina Azahara

Fecha entrega: 19/06/2023

Autor: Pablo Garrido Garrido

Índice

1. Introducción.....	6
1.1 Contexto y justificación del Proyecto.....	6
1.2 Objetivos del Proyecto.....	6
1.3 Enfoque y método seguido.....	6
1.4 Planificación.....	7
1.4.1 Tareas a realizar.....	7
1.5 Costes del Proyecto.....	10
2. Resto de capítulos.....	15
2.1 Diagrama de como quedará la red de la clase:.....	15
2.2 Instalación de proxmox, todos los contenedores y servicios.....	16
2.2.1 Descarga de proxmox.....	16
2.2.2 Descargar rufus.....	16
2.2.3 Entrar en el UEFI e instalar Proxmox.....	18
2.2.4 Configuración tras la instalación.....	24
2.2.5 Descarga y creación del contenedor.....	28
2.2.6 Configuraciones Iniciales del contenedor.....	40
2.2.7 Instalamos el servidor DHCP y vemos la IP.....	41
2.2.8 Configuramos el servidor DHCP y comprobamos que funciona.....	41
2.2.9 Instalamos y configuramos el servidor Web.....	46
2.2.10 Comprobamos que funciona la configuración.....	50
2.2.11 Instalamos jitsi.....	50
2.2.12 Instalamos y creamos las zonas el servidor DNS.....	54
2.2.13 Creamos la carpeta de zonas, copiamos el archivo de la zona directa a las carpeta de zonas y creamos la zona directa.....	55
2.2.14 Copiamos el archivo de la zona inversa a la carpeta de zonas y creamos la zona inversa.....	56
2.2.15 Configuramos el servidor DHCP y el Proxmox para el bind9.....	57
2.2.16 Vemos los resultados.....	59
2.2.17 Asignamos una IP fija mediante el DHCP.....	60
2.2.18 Crear el contenedor para el servidor FTP y configuraciones iniciales.....	63
2.2.19 Instalar servidor FTP y configurarlo.....	73
2.2.20 Instalamos apache y creamos los usuarios virtuales de vsftpd.....	74
2.2.21 Crear usuario vsftpd y archivo de configuración de los usuarios.....	75
2.2.22 Crear certificado para vsftpd.....	76
2.2.23 Terminar de configurar /etc/vsftpd.conf.....	77
2.2.24 Comprobar que funciona vsftpd en Ubuntu.....	78
2.2.25 Instalamos la entidad certificadora del servidor VPN y creamos la infraestructura de clave pública de la entidad certificadora.....	81
2.2.26 Creamos la clave pública, la clave privada y el certificado de la Entidad Certificadora.....	82
2.2.27 Instalamos OpenVPN y creamos la solicitud de certificado del servidor OpenVPN y la clave privada del servidor OpenVPN.....	84
2.2.28 Firmar la solicitud de certificado del servidor OpenVPN.....	85
2.2.29 Configuración del material criptográfico de OpenVPN, generación de una solicitud de certificado de cliente y el par de claves.....	87
2.2.30 Firmamos la solicitud del cliente con la entidad certificadora.....	88
2.2.31 Configurar Openvpn y tener credenciales no predeterminadas.....	90
2.2.32 Ajuste de la configuración de red y firewall del servidor OpenVPN.....	92
2.2.33 Iniciamos OpenVPN y creamos la configuración del cliente.....	95

2.2.34	Generamos la configuración del cliente y lo copiamos al cliente.....	99
2.2.35	Instalamos OpenVPN y la configuración en Lubuntu con systemd-resolved	100
2.2.36	Conectarse a OpenVPN desde Lubuntu y comprobar que funciona correctamente.....	101
2.2.37	Hacer instantáneas por si algo falla y poder recuperarlo al mismo punto donde la hicimos el contenedor.....	103
2.2.38	Podemos monitorear el equipo.....	106
2.3	Viabilidad.....	108
3.	Conclusiones.....	108
4.	Glosario.....	109
5.	Referencias.....	114



Esta obra está sujeta a una licencia de Reconocimiento –
No Comercial – Sin Obra Derivada [3.0 España de
Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL PROYECTO FINAL

Título del trabajo:	Contenedores con servicios de red en Proxmox
Nombre del autor:	Pablo Garrido
Fecha de entrega:	06/2023
Área del Trabajo Final:	Instalación de contenedores en Proxmox y servicios de red
Ciclo Grado Superior:	Administración de Sistemas Informáticos en Red
Resumen del Trabajo (máximo 250 palabras):	
<p>Este es un trabajo de como mejorar los servicios de red de la clase que tiene una Raspberry Pi que solo tiene 1 GB de RAM, va muy lenta, tiene una tarjeta microsd como unidad de almacenamiento que es muy lenta y vamos a instalar un servidor con 128 GB de RAM y dos discos duros de 2 TB para que todo vaya fluidamente todos los servicios de red: servidor DHCP, servidor DNS, servidor Web, servidor FTP, servidor VPN, usarlo como router, etc.</p>	

1. Introducción

1.1 Contexto y justificación del Proyecto

La necesidad que hay que cubrir es **tener** un **servidor** con varios **contenedores** con distintos **servicios** de **red** e internet en la **clase** de **2º ASIR**.

Es un tema relevante porque la **Raspberry Pi** se suele **quedar colgada a menudo** y **va mal** porque tiene una tarjeta **microsd** como **disco duro** y **1 GB de RAM**.

Queremos instalar un **servidor** con **128 GB de RAM**, **dos procesadores** de **10 núcleos**, **2 discos de 3 TB**, ponemos **RAID1** para **no perder** los **datos** si **falla un solo disco**. El **servidor** es como si fueran **2 ordenadores en 1**.

Se **resuelve** el **problema** de momento con la **Raspberry Pi** que tiene **muy pocos recursos** como acabamos de ver.

La **aportación realizada** es **instalar Proxmox** y luego **instalar** los **contenedores** con los distintos **servicios de red**.

Los **resultados** que **queremos obtener**, es que **funcionen mejor** los **servicios de red** de la **clase** al tener **muchos** más **recursos**.

1.2 Objetivos del Proyecto

- **Mejorar** los **servicios** de **red** de la **clase** porque las **Raspberry Pi** tienen **muy poco almacenamiento** y solo **1 GB de RAM**.
- Vamos a **mejorar** los **siguientes servicios de red**:
 - **Servidor DHCP**: para poder asignar direcciones IP a los dispositivos de la clase sin configurarlas manualmente, sean fijas las direcciones IP de cada ordenador al asignar una IP fija con el DHCP a una dirección MAC determinada.
 - **Servidor DNS**: asignar un nombre de host a cada equipo para hacer ping o conectarse más fácilmente memorizando el nombre de la zona y el dispositivo en vez de la IP.
 - **Servidor Web**: tiene jitsi para hacer videollamadas y páginas webs con contraseña que solo pueden ver las personas que saben la contraseña.
 - **Servidor FTP**: para almacenar las imágenes ISO, archivos compartidos, máquinas virtuales con contraseña para entrar en el FTP.
 - **Servidor VPN**: con OpenVPN para poder por ejemplo, navegar de forma segura en clase, ya que se encripta la conexión.

1.3 Enfoque y método seguido

Una **estrategia** sería **instalar varias máquinas virtuales** en **Virtualbox**, pero **requieren más** memoria **RAM** que los **contenedores** y ocupa **más** espacio en **disco**.

Otra estrategia y la más apropiada es instalar Proxmox en el servidor. Luego instalar distintos contenedores para los diferentes servicios: DHCP, DNS, Web, router, jitsi, FTP, VPN y configurarlos.

1.4 Planificación

Los recursos necesarios son el servidor proxmox, el router, un switch, cables ethernet, un rack y los equipos clientes.

El seguimiento del proyecto lo hace mi tutora Josefa María López viéndolo en la clase o mandando los correos electrónicos a ella.

La puede hacer quién tenga un ordenador que no use con al menos 8 GB de RAM y 2 núcleos del procesador exclusivo para Proxmox.

Tienes que tener conocimientos avanzados de informática para hacer la configuración de Proxmox con todos sus servicios.

1.4.1 Tareas a realizar

1. Descargar Proxmox.
2. Descargar Rufus y meter Proxmox en un pendrive.
3. Entrar en la UEFI para instalar Proxmox desde el pendrive.
4. Instalar Proxmox configurando la dirección IP, servidor DNS, puerta de enlace, nombre del host, fecha y hora, idioma del teclado, disco duro donde se va a instalar y contraseña de root para entrar en Proxmox.
5. Entramos a proxmox con el usuario root, contraseña y lo ponemos en español.
6. Deshabilitar el repositorio Enterprise en Proxmox y activar el repositorio No-Subscription.
7. Recargamos repositorios y actualizamos Proxmox.
8. Creamos el primer contenedor poniendo identificador, nombre del host, contraseña del root, archivo del contenedor, almacenamiento, núcleos de la CPU, memoria RAM, tarjeta de red puente, dirección IP, puerta de enlace y servidor DNS.
9. Arrancamos el contenedor, recargamos repositorios y actualizamos el contenedor.
10. Configuramos la hora correcta en el contenedor.
11. Instalamos el servidor DHCP.
12. Configuramos el servidor DHCP.
13. Comprobamos que funciona el servidor DHCP en el cliente.
14. Instalamos apache2.

- 15. Deshabilitamos el sitio por defecto.**
- 16. Configuramos apache2 habilitando el módulo ssl para tener https y cambiando la carpeta.**
- 17. Copiamos el archivo de configuración y le ponemos otro nombre para editarlo y creamos los usuarios y las contraseñas para la carpeta privada apache2.**
- 18. Cambiamos el puerto de apache2, ponemos una carpeta privada con contraseña y una pública.**
- 19. Comprobamos que funciona la carpeta privada con contraseña y la carpeta pública.**
- 20. Instalamos el servidor DNS bind9 y creamos la zona directa.**
- 21. Creamos la zona inversa.**
- 22. Ponemos la IP del servidor DNS en el DHCP y ponemos en los reenviadores los DNS de nuestro router para poder acceder a páginas webs.**
- 23. Ponemos en el DNS de nuestro contenedor en nuestro propio contenedor Proxmox.**
- 24. Comprobamos que funciona el servidor DNS.**
- 25. Asignamos una IP fija mediante el DHCP al Lubuntu con el nombre que le hemos puesto en el DNS.**
- 26. Descargamos e instalamos los repositorios de jitsi, instalamos jitsi poniendo nuestro dominio de la zona DNS y comprobamos que funciona.**
- 27. Creamos un contenedor para el servidor FTP securizado y VPN.**
- 28. Instalamos el servidor ftp y configuramos el archivo de configuración FTP.**
- 29. Instalamos el servidor apache y creamos los usuarios del servidor FTP y editamos el archivo de autenticación PAM del FTP para que se autentifique con el archivo de los usuarios del servidor FTP.**
- 30. Creamos los usuarios para el ftp, creamos los archivos de configuración de los usuarios del servidor FTP, creamos las carpetas dentro de /var/www para los usuarios y cambiamos el propietario de las dos carpetas creadas a vsftp.**
- 31. Creamos el certificado y activamos el servidor FTP seguro poniendo certificado al archivo de configuración del FTP, también ponemos los puertos pasivos para conexiones pasivas.**
- 32. Comprobamos que funciona el servidor FTP en Lubuntu instalando Filezilla y funcionan correctamente los dos usuarios viendo sus carpetas.**
- 33. Instalamos la entidad certificadora y creamos la infraestructura de clave pública en la entidad certificadora.**

- 34. Editamos el archivo vars en la entidad certificadora con los datos de organización, ciudad, provincia, etc y poner criptografía de curva elíptica ECC ahí.**
- 35. Creamos el par de claves raíz pública y privada para su autoridad de certificación, creamos una contraseña y un nombre común.**
- 36. Instalamos el VPN, hacemos la solicitud de firma del certificado del servidor y creamos la clave privada.**
- 37. Importamos el solicitud de firma del certificado en la entidad certificadora, lo firmamos, obtenemos clave pública del servidor VPN, la firma de la entidad certificadora y lo copiamos a /tmp.**
- 38. Generar clave TLS adicional compartida.**
- 39. Hacemos la solicitud del certificado del cliente y creamos la clave privada.**
- 40. Copiamos la solicitud del certificado del cliente proyectopggvpnciente a /tmp y importamos el certificado en la entidad certificadora, lo firmamos y lo copiamos a ~/client-configs/keys/.**
- 41. Copiamos el archivo de configuración y configuramos OpenVPN con credenciales no predeterminadas.**
- 42. Configuramos enrutamiento para tener el reenvío de tráfico.**
- 43. Abrir los puertos necesarios para el servidor OpenVPN, servidor FTP, servidor SSH y servidor WEB.**
- 44. Agregamos tun0 al contenedor proxmox.**
- 45. Iniciamos OpenVPN.**
- 46. Creamos la configuración del cliente.**
- 47. Creamos un script y generamos la configuración de nuestro cliente con el script.**
- 48. Copiamos la configuración al cliente.**
- 49. Instalamos OpenVPN en el cliente, comprobamos que el cliente usa systemd-resolved y instalamos openvpn-systemd-resolved para el systemd-resolved.**
- 50. Editamos la configuración del cliente para el systemd-resolved y lo iniciamos en el cliente.**
- 51. Comprobamos que funciona el cliente de OpenVPN.**

1.5 Costes del Proyecto

Router Asus RT-AX58U

- **Precio:** 124,86€
- **URL de compra:** https://www.amazon.es/ASUS-RT-AX58U-Servidor-repetidor-AiProtection/dp/B07YN5496D/ref=cm_cr_ar_p_d_product_top?ie=UTF8
- **Wifi 6** con bandas de 2,4 GHz y 5 GHz.
- **Hasta 574 Mbit** en la banda de 2,4 GHz con 40 Mhz de ancho de canal y **2402 Mbit** en la banda de 5 GHz con 160 Mhz de ancho de canal.
- **Para conectar muchos dispositivos.**
- **Más alcance.**
- **Cifrado WPA3.**

Proxmox

El Proxmox es gratis si tenemos el repositorio No-Subscription.

Ordenadores

- **Precio de cada ordenador:** 531,99€ x 20 = 10.639,80 € 20 ordenadores.
- **Procesador:** AMD Ryzen 5 4600G 4.20GHz 100,99€ x 20 = 2019,80€.
- **Placa base:** MSI B450M PRO-VDH Max 73,99€ x 20 = 1479,80€.
- **Memoria RAM:** Kingston FURY Beast DDR4 3200 MHz 16GB 2x8GB CL16 44,99€ x 20 = 899,8€.
- **Torre:** Tempest Vision RGB Torre ATX Negra 40,99€ x 20 = 819,80€.
- **Ventilador:** Tempest Cooler 4Pipes Black Ventilador CPU 120mm Negro 19,99€ x 20 = 399,80 €.
- **SSD:** Crucial BX500 SSD 240GB 3D NAND SATA3 33€ x 20 = 660€.
- **Fuente de alimentación:** Nox Urano VX 650W 80+ Bronze 120MM PWM 49,99€ x 20 = 999,80€.
- **Monitor:** Keep Out XGM22BV2 21.5" LED FullHD 75Hz 89,99€ x 20 = 1.799,80€.
- **Teclado y ratón:** Logitech MK295 Silent Wireless Combo de Ratón y Teclado Inalámbricos 35,78€ x 20 = 715,60€.
- **Altavoces:** Hama Sonic Mobil 185 Altavoces 2.0 Negros 6,59€ x 20 = 131,80€.
- **Tarjeta de red/wifi:** Asus PCE-AX3000 Tarjeta de Red WiFi 6 PCIe AX3000 35,69€ x 20 = 713,80€.

Cable Ethernet:

- Nixsto Cable Ethernet, Cat 8 Plano Cable de red RJ45 Alta Velocidad 40 Gbps 2000MHz, Cable LAN para Routers, Módems, Consolas y TV Box, PS5, PS4, más Rápido que el Cable Cat5e/Cat6/Cat7
- Uno de 2 metros 4,99€, uno de 3 metros 5,49€, dos de 5 metros 5,99€ x 2 = 11,98€, dos de 7 metros 7,99€ x 2 = 15,98€, tres de 10 metros 9,99€ x 3 = 29,97€, 5 de 15 metros 13,99€ * 5 = 69,95€, 5 de 20 metros 16,99€ x 5 = 84,95€, 2 de 30 metros 19,99€ x 2 = 39,98€, total 263,29€.
- Enlace de compra: https://www.amazon.es/Ethernet-Velocidad-2000MHz-Routers-Consolas/dp/B0B24NY46W/ref=sr_1_5?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=3IX7LAU7FK3Z1&keywords=cable%2Bethernet%2Bde%2B1%2Bmetro&qid=1685867578&srefix=cable%2Bethernet%2Bde%2B1%2Bmetro%2C
- El cable ethernet Cat 8 proporciona hasta 40Gbps de transferencia de datos por segundo y 2000MHz de ancho de banda, utilizado para 25/40 GBase-T(más rápido que el cable Cat 7 de 10Gbps 600MHz). Este cable Cat8 se puede utilizar para redes domésticas y de oficina por cable, conectarse a segmentos de LAN/WAN y equipos de red a alta velocidad, y le permite navegar por Internet, jugar a juegos y transmitir vídeos con fluidez.
- Conector RJ45 Antiinterferente. El cable de Internet profesional Cat8 está hecho de 4 pares trenzados blindados de 30AWG 100% de cobre. En comparación con cat5e/cat6/cat7, minimizará la interferencia que tiene un impacto negativo en la calidad de la señal; y los conectores RJ45 chapados en oro, duraderos y fáciles de enchufar, se ajustan estrechamente con los contactos chapados en oro 8P8C para proporcionar conexiones más rápidas y estable.
- Cable Estable, Duradero y Flexible. Cada cable de red adopta un exterior de pvc y se somete a más de miles de pruebas de flexión, lo que es muy duradero y tiene una larga vida útil. Si tiene algún problema con nuestro producto, le proporcionaremos un servicio post-venta amigable a tiempo.
- Diseño Plano único. Este cable de red Cat 8 plano, delgado y flexible, tiene 7,2+ - 0,1mm de ancho y 2,2+ - 0,08mm de grosor, que es súper flexible y duradero para pasar por las esquinas, las costuras de las puertas, o ponerlo fácilmente debajo de la alfombra sin que se retuerza, ondule, se agriete o se enrede como los redondos, lo que hace que sus espacios se vean ordenados y limpios.
- Amplia Compatibilidad. Este cable de alta velocidad Cat 8 RJ45 lan no sólo es compatible con cat7, Cat6e, Cat6, Cat5, Cat5e cables, pero también funciona bien con todos los dispositivos RJ45 como PC, impresora de red, WiFi Router, WiFi Extender, Virgin módem, Switch, TV Box.

Switch para poder conectar varias tarjetas red de los ordenadores y el servidor

- D-Link DGS-1100-24V2, Switch Smart 24 puertos Gigabit, RJ-45, Gestión web, Layer 2, VLAN, sobremesa, sin ventilador, IGMP Snooping, VoIP VLAN, QoS, Seguridad red, montaje rack; sin ventilador, Green.
- Precio 149,24€.
- URL de compra: https://www.amazon.es/D-Link-DGS-1100-24V2-sobremesa-ventilador-Seguridad/dp/B0876G6ZC2/ref=sr_1_7?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=1KZVP4057UG2H&keywords=switch+24+puertos+gestionable&qid=1685868593&prefix=switch+24+puertos+gestionable%2C
- 24 puertos Gigabit, modos sin gestión para ser Plug&Play o gestionable Smart para crear segmentaciones de red, priorización de tráfico, Port-Based VLAN, QoS, IGMP Snooping.
- Montaje en rack estándar de 19 pulgadas o sobremesa, diseño sin ventiladores, tecnología D-Link Green y compatibilidad 802.3az EEE para una gestión eficiente del consumo eléctrico.
- Gestión Smart mediante Web GUI o con el software gratuito D-Link Network Assistant Utility para una gestión intuitiva de la red y los dispositivos conectados.
- Funciones Auto Voice VLAN y Auto Surveillance VLAN para agregar teléfonos IP y cámaras de forma fácil y con priorización de tráfico para asegurar la calidad del servicio.
- Garantía limitada de por vida con servicio de reemplazo Next Business Day (NBD) tras registro de producto.
- La interfaz web es una forma fácil de administrar las características de capa 2 (Layer 2), tales como la creación de VLANs, Spanning Tree Protocol (STP) y agregación de enlaces (estático). Con potentes funciones de seguridad como IGMP snooping, MAC estático y Storm Control, la familia DGS-1100 Smart es una solución versátil para entornos que requieren soluciones fáciles de implementar y configurar sin complejidades añadidas.
- El Asistente de red (DNA) sin coste de D-Link muestra todos los dispositivos conectados y acelera la configuración por primera vez. El sistema de administración de redes D-View 7 viene con una licencia sin coste para 25 nodos y 2 sondas y acceso completo a las herramientas de administración avanzada. Las licencias de actualización opcionales le permiten ampliar la gestión a más redes.
- Opcionalmente puede seleccionar el modo Surveillance en la interfaz web para agregar de forma instantánea las cámaras y grabadores NVR, además de una interfaz más amigable para gestionar su topología de red CCTV. Detecta automáticamente sus dispositivos de seguridad y los segmenta en una VLAN dedicada para administrar el tráfico de videovigilancia de forma segura y eficiente.
- La familia de Switches Smart DGS-1100 incluye múltiples opciones con puertos Power Over Ethernet PoE y PoE+ que suministran la alimentación eléctrica por el

cable de red de datos a dispositivos compatibles como Teléfonos de Voz IP, cámaras, puntos de acceso WiFi, etc. De esta forma, se facilita su despliegue y se ahorran costes de instalación eléctrica adicional.

- La gama DGS-1100 ofrece Loopback Detection y Cable Diagnostics para ayudar a los administradores de red a encontrar y resolver problemas de red rápida y fácilmente. Loopback Detection se utiliza para detectar bucles creados por un puerto específico y cerrar automáticamente el puerto afectado. La función de Cable Diagnostics está diseñada para que los administradores de red puedan examinar rápidamente la calidad de los cables de cobre, reconocer su tipo y detectar errores en la transmisión.
- La tecnología D-Link Green permite a la gama Smart DGS-1100 ahorrar energía sin sacrificios. La mayoría de la familia utiliza un diseño sin ventilador, para un funcionamiento silencioso y eficiente a nivel de consumo eléctrico. Todos son compatibles con IEEE 802.3az y cuentan con detección de estado de enlace y PoE basado en tiempo (excepto los modelos DGS-1100-05PD y DGS-1100-08P).

Rack para Servidor

- StarTech.com Armario de Red 12U 19" para Montaje en Pared, Armario para Equipos Informáticos con 50cm/20" de Profundidad y 4 Columnas, Puerta con Bisagras y Estante (RK1224WALHM)
- **Precio:** 465,12€.
- Este armario 12U para racks de servidores o redes le permite instalar sus equipos compatibles con la norma EIA-310 en una pared, en una caja segura con puerta trasera abisagrada, para facilitar el acceso a su equipo. La caja permite una profundidad ajustable de montaje, entre 6,1cm a 60,9cm, lo cual constituye una solución resistente de almacenamiento para su equipo de montaje en rack.
- Para proporcionar estabilidad segura a su equipo más pesado, este rack tiene un diseño de montaje con 4 postes, lo cual permite una mayor capacidad de peso por U que los racks con 2 postes, hasta una capacidad total de 90kg.
- Ahorre valioso espacio en el suelo gracias al montaje en pared de la caja. El diseño para montaje en pared de esta caja resulta ideal para optimizar el almacenamiento de su equipo en un aula de clases, tienda de venta minorista, sala de servidores u oficina. Este rack también incluye una bandeja 1U que ofrece una superficie estable para colocar equipos no aptos para montaje en rack o para guardar sus herramientas.
- El rack para redes de montaje en pared facilita el acceso a su equipo y cableado, gracias a la bisagra trasera que aleja la caja de la pared. Al girar la caja en 180 grados tras la instalación, es posible invertir la dirección de la bisagra, de manera que la caja pueda balancearse y abrirse desde la izquierda o la derecha, como prefiera.

- El armario 12U para redes incluye una puerta frontal reversible y paneles laterales extraíbles, cada uno con mecanismos de rápido desbloqueo, a fin de acceder fácilmente a su equipo. Además, cada puerta y panel del rack tiene un candado independiente, lo cual garantiza la seguridad.
- Contenido caja: caja para montaje en pared, bandeja 1U, rollo de anclajes de gancho y bucle, tuercas enjauladas 12-24, tornillos 12-24, llaves de la puerta y llaves para el panel lateral.

Servidor:

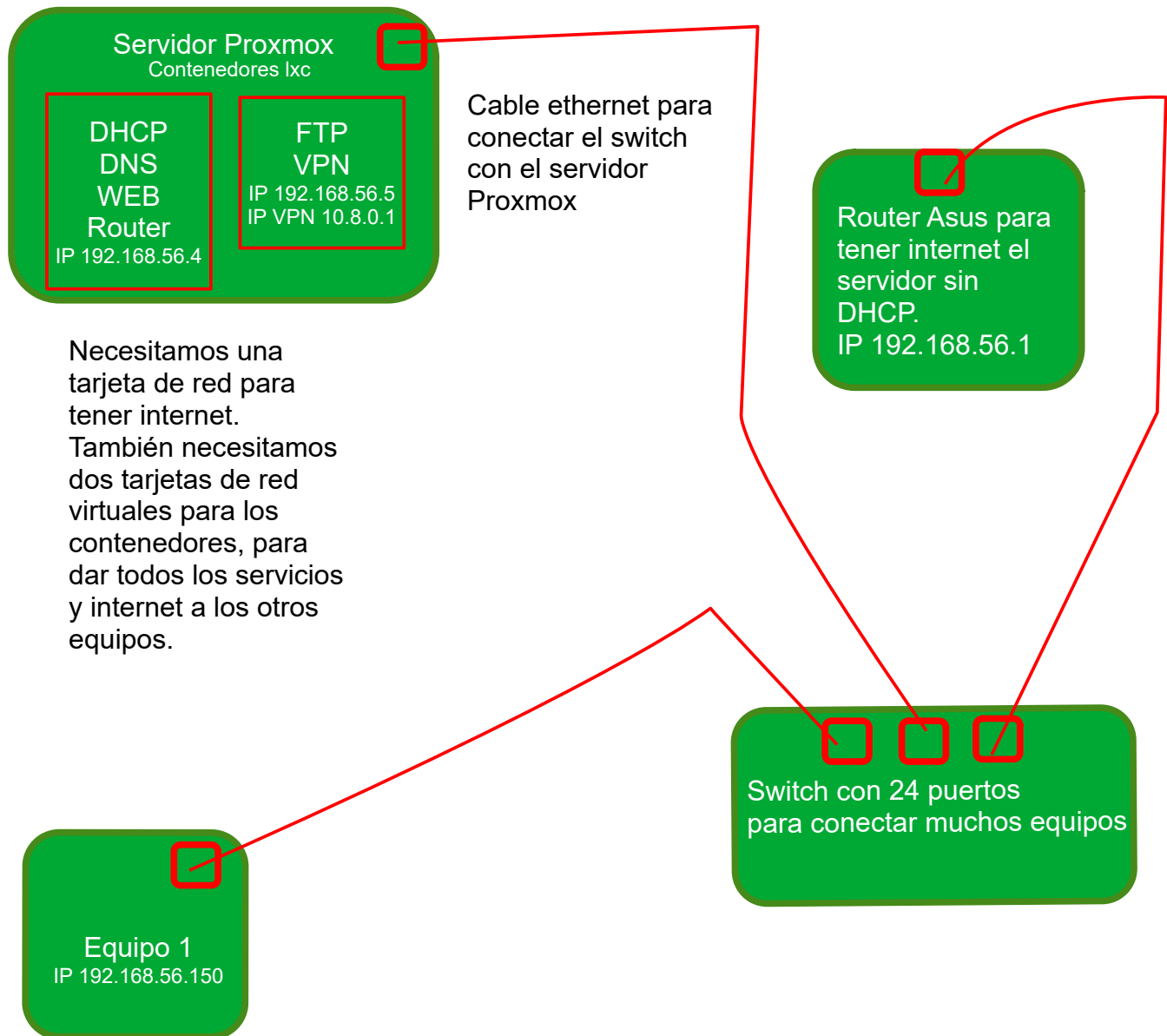
Dell R720 | 8x LFF | 2x Intel Xeon 10-Core E5-2660 V2 | 128GB RAM DDR3 | 2x 3TB SAS 10000 Rpm | H710 Ctrl | 2xPSU

Precio 999€.

Coste total del proyecto: 12.641,91€

2. Resto de capítulos

2.1 Diagrama de como quedará la red de la clase:

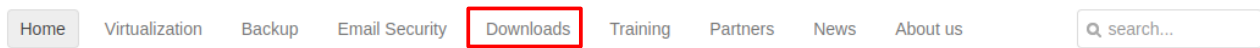


2.2 Instalación de proxmox, todos los contenedores y servicios

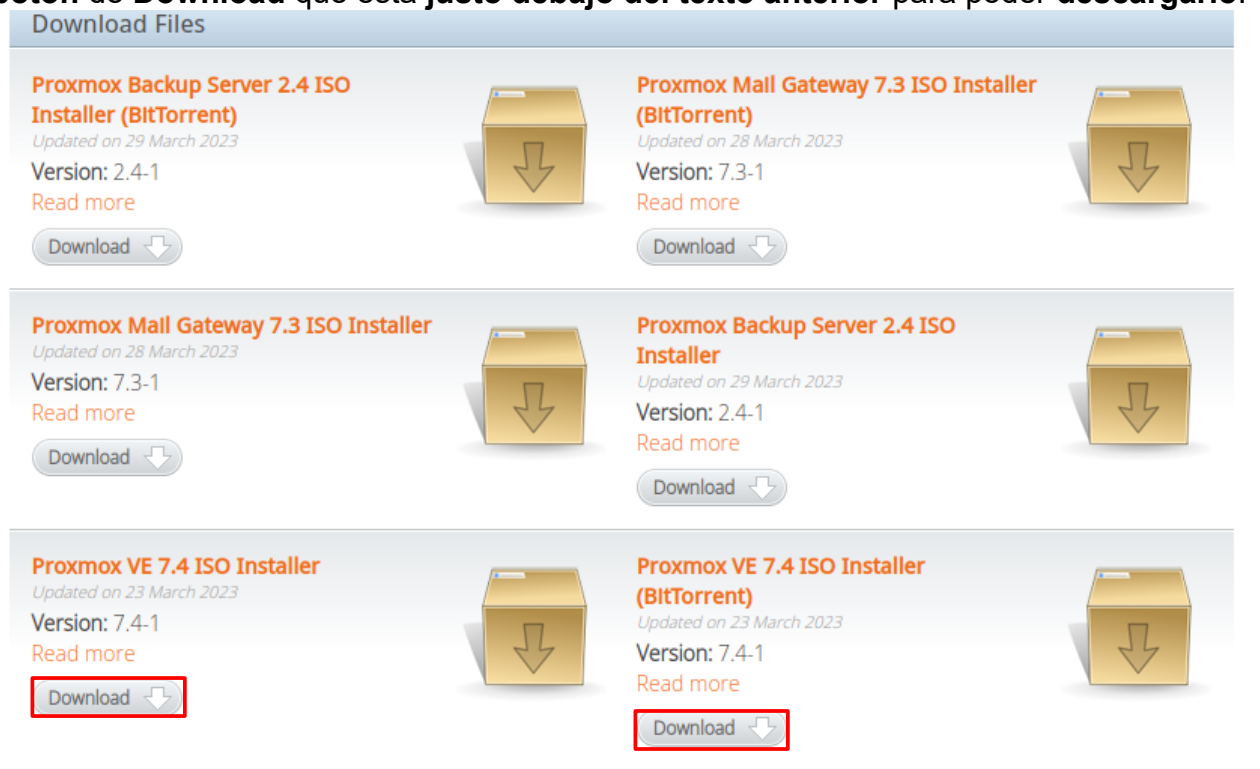
2.2.1 Descarga de proxmox

Descargamos proxmox desde su página web poniendo <https://www.proxmox.com>

Hacemos clic en **Downloads** para entrar en las descargas y poder descargarlo.



En **Download Files** hacemos clic en **Proxmox VE 7.4 ISO Installer** o **Proxmox VE 7.4 ISO Installer (BitTorrent)** si lo queremos **descargar** como **Torrent** y hacemos clic en el botón de **Download** que está **justo debajo del texto anterior** para poder **descargarlo**.

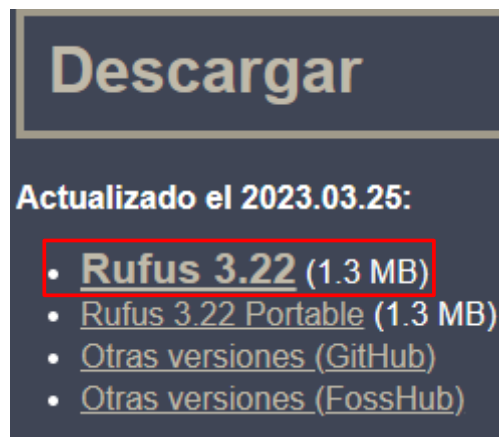


2.2.2 Descargar rufus

Rufus sirve para meter en un **pendrive** la **imagen ISO** de nuestro **Proxmox** para poder **instalarlo** en nuestro equipo.

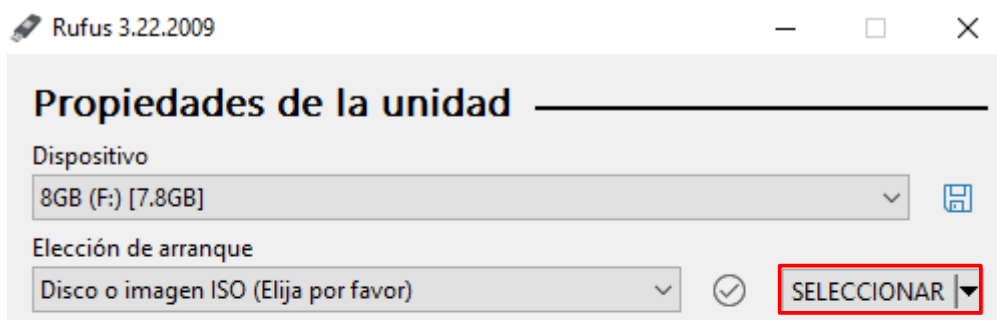
Ponemos la **siguiente URL** para descargar **Rufus** <https://rufus.ie/es/>

Hacemos clic en la sección de **Descargar** en el nombre de la última versión que en este caso en **Rufus 3.22** para poder descargar **Rufus**.

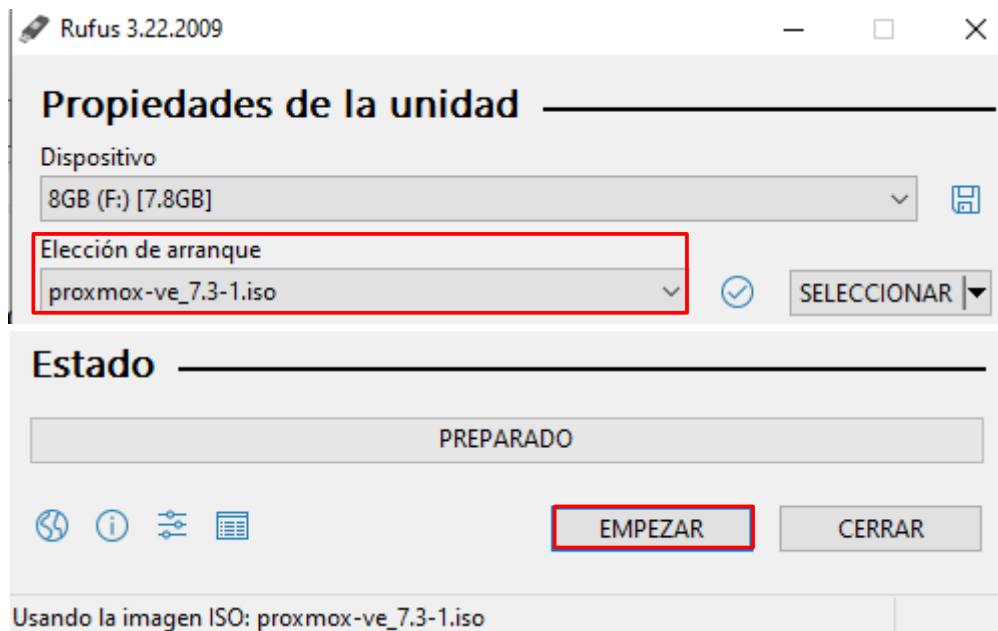


Insertamos el pendrive y lo vemos en **Dispositivo**, si tenemos varios pendrives, seleccionamos el que queramos utilizar para poder meter el proxmox en el pendrive.

En **Elección de arranque** seleccionamos **Disco o imagen ISO (Elija por favor)** y hacemos clic en **SELECCIONAR** para seleccionar la imagen ISO del proxmox.



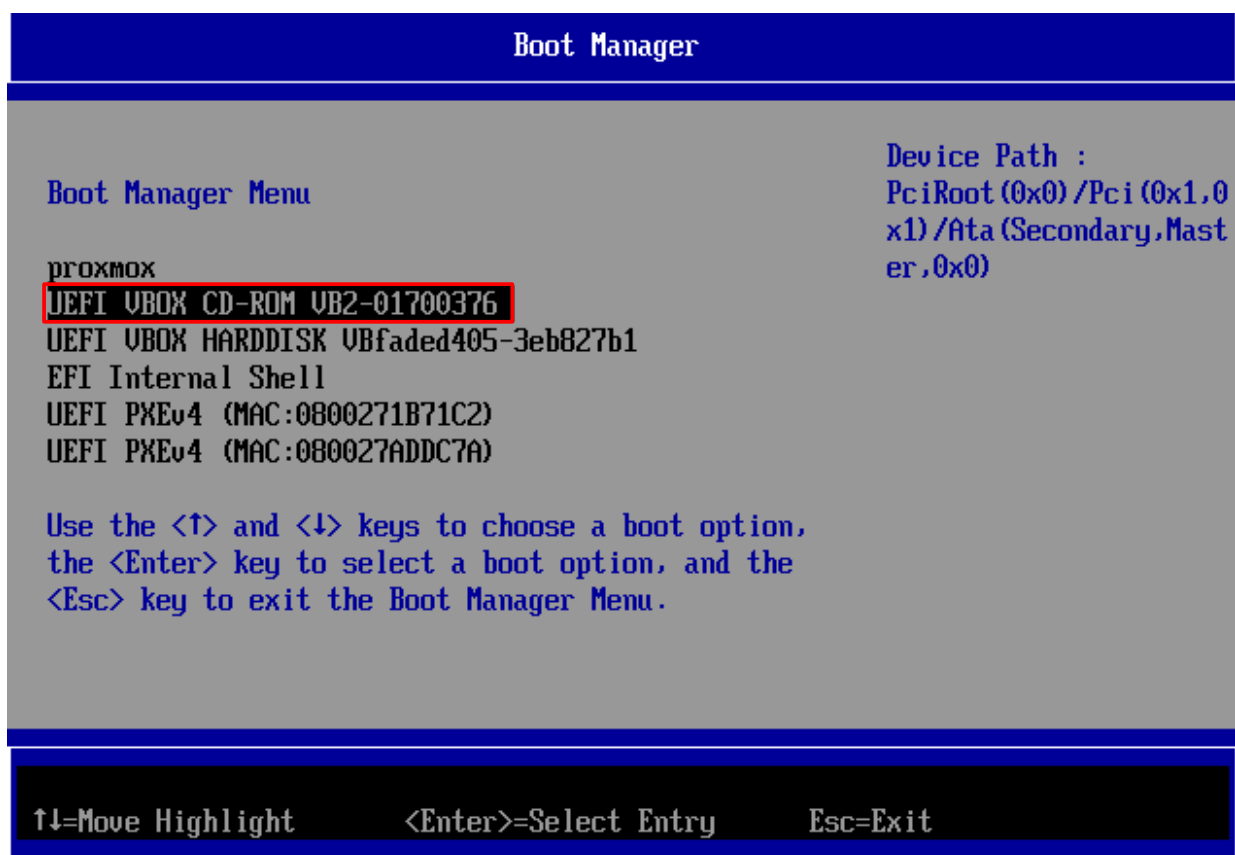
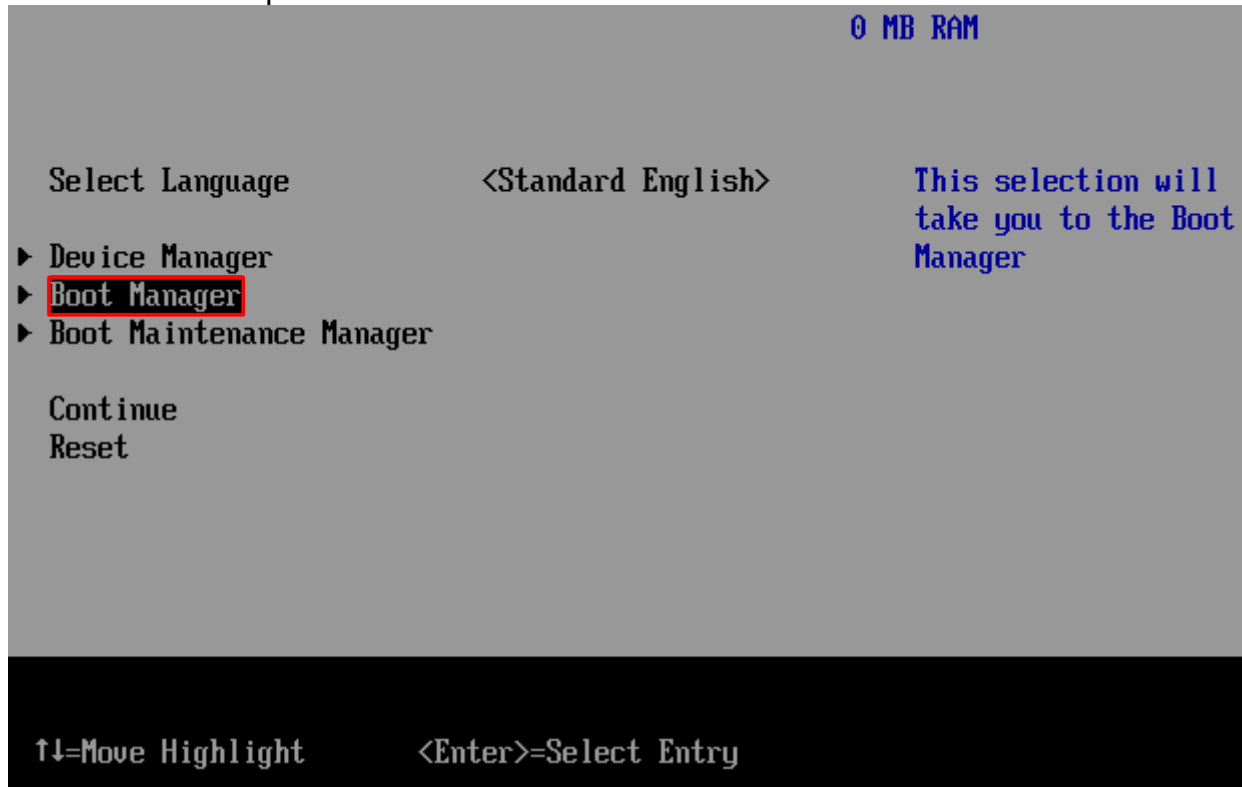
Ya vemos la **imagen ISO** seleccionada en **Elección de arranque**.



Hacemos clic en **EMPEZAR** para **empezar a meter** el **Proxmox** en el **pendrive**. Hacemos clic en **Aceptar** para **eliminar los datos** del **pendrive** y **meter Proxmox** en el **pendrive**.

2.2.3 Entrar en el UEFI e instalar Proxmox

Entramos en el UEFI con una tecla, nos movemos en las opciones con las flechas arriba o abajo, pulsamos en Enter en Boot Manager, seleccionamos el pendrive como UEFI o la unidad óptica como UEFI si hay disco insertado para arrancar el equipo y presionamos Enter para arrancarlo.



Pulsamos en Enter en Install Proxmox VE para empezar a instalar Proxmox.

Proxmox VE 7.3 (iso release 1) - <https://www.proxmox.com/>



Install Proxmox VE

Advanced Options

Leemos los términos y condiciones o END USER LICENCE AGREEMENT (EULA), y hacemos clic en I agree para aceptar los términos y condiciones.

END USER LICENSE AGREEMENT (EULA)

END USER LICENSE AGREEMENT (EULA) FOR PROXMOX VIRTUAL ENVIRONMENT (PROXMOX VE)

By using Proxmox VE software you agree that you accept this EULA, and that you have read and understand the terms and conditions. This also applies for individuals acting on behalf of entities. This EULA does not provide any rights to Support Subscriptions Services as software maintenance, updates and support. Please review the Support Subscriptions Agreements for these terms and conditions. The EULA applies to any version of Proxmox VE and any related update, source code and structure (the Programs), regardless of the the delivery mechanism.

1. License. Proxmox Server Solutions GmbH (Proxmox) grants to you a perpetual, worldwide license to the Programs pursuant to the GNU Affero General Public License V3. The license agreement for each component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (certain obligations in some cases), both in source code and binary code forms, with the exception of certain binary only firmware components and the Proxmox images (e.g. Proxmox logo). The license rights for the binary only firmware components are located within the components. This EULA pertains solely to the Programs and does not limit your rights under, or grant you rights that supersede, the license terms of any particular component.

2. Limited Warranty. The Programs and the components are provided and licensed "as is" without warranty of any kind, expressed or implied, including the implied warranties of merchantability, non-infringement or fitness for a particular purpose. Neither Proxmox nor its affiliates warrants that the functions contained in the Programs will meet your requirements or that the operation of the Programs will be entirely error free, appear or perform precisely as described in the accompanying documentation, or comply with regulatory requirements.

3. Limitation of Liability. To the maximum extent permitted under applicable law, under no

Abort

Previous

I agree

Seleccionamos el disco donde vamos a instalarlo en **Target Harddisk**, se nos borrarán todas las particiones y todos los datos, y hacemos clic en **Next** para empezar la instalación.



Proxmox Virtual Environment (PVE)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and makes the system bootable from the hard disk. All existing partitions and data will be lost.

Press the Next button to continue the installation.

- **Please verify the installation target**
The displayed hard disk will be used for the installation.
Warning: All existing partitions and data will be lost.
- **Automatic hardware detection**
The installer automatically configures your hardware.
- **Graphical user interface**
Final configuration will be done on the graphical user interface, via a web browser.

Seleccionamos el país haciendo clic en **Country** y seleccionamos **Spain** que es España en inglés, en **Time zone** seleccionamos **Europe/Madrid**, ya que estamos en la península ibérica y en **Keyboard Layout** seleccionamos **Spanish** o español en inglés. Hacemos clic en **Next** para continuar.

Location and Time Zone selection

The Proxmox Installer automatically makes location-based optimizations, like choosing the nearest mirror to download files from. Also make sure to select the correct time zone and keyboard layout.

Press the Next button to continue the installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speed up downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.

Ponemos una contraseña para entrar en el root del proxmox en Password y la confirmamos en Confirm, ponemos un correo en Email para enviarnos notificaciones de eventos importantes. Hacemos clic en Next para continuar.

Administration Password and Email Address

Proxmox Virtual Environment is a full featured, highly secure GNU/Linux system, based on Debian.

In this step, please provide the *root* password.

- **Password:** Please use a strong password. It should be at least 8 characters long, and contain a combination of letters, numbers, and symbols.
- **Email:** Enter a valid email address. Your Proxmox VE server will send important alert notifications to this email account (such as backup failures, high availability events, etc.).

Press the Next button to continue the installation.

The screenshot shows the 'Administration Password and Email Address' step of the Proxmox VE installation. It features three input fields: 'Password' (masked with 8 dots), 'Confirm' (masked with 8 dots and a cursor), and 'Email' (containing 'pgg@pgg.pgg'). These fields are enclosed in a red rectangular box. Below the fields are three buttons: 'Abort' on the left, 'Previous' in the center, and 'Next' on the right. The 'Next' button is highlighted with a red rectangular box.

En **Management Interface** seleccionamos nuestra **tarjeta de red principal** con su **dirección MAC**, en **Hostname (FQDN)** ponemos **pve**, que es el **nombre de nuestro servidor proxmox**, nuestro **prefijo**, después **punto**, nuestro **nombre de host principal** que en este caso es **pgg** y luego nuestro **nombre de dominio**, que en este caso es **punto net**, ponemos nuestra **dirección IP** y nuestra **máscara de subred** en **IP Address (CIDR)**, ponemos nuestra **puerta de enlace** en **Gateway** y nuestro **servidor de nombres de dominio** en **DNS Server** y hacemos clic en **Next** para **continuar**.

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installing.

After you have finished, press the Next button. You will be shown a list of the options that you chose during the previous steps.

- **IP address (CIDR):** Set the main IP address and netmask for your server in CIDR notation.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

Management Interface: enp0s3 - 08:00:27:1b:71:c2 (e1000) ▼

Hostname (FQDN): pve.pgg.net

IP Address (CIDR): 172.16.20.3 / 24

Gateway: 172.16.20.1

DNS Server: 1.1.1.1

[Previous](#)[Next](#)

Vemos un **resumen** de lo que **va a hacer el instalador** con **todas las opciones que hemos seleccionado**, dejamos **marcada la casilla Automatically reboot after successful installation** para **reiniciar el equipo y arrancar el proxmox tras la instalación** y hacemos **clic en Install** para **instalar el proxmox**.

Summary

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	Spain
Timezone:	Europe/Madrid
Keymap:	es
Email:	pgg@pgg.pgg
Management Interface:	enp0s3
Hostname:	pve
IP CIDR:	172.16.20.3/24
Gateway:	172.16.20.1
DNS:	1.1.1.1

☒ Automatically reboot after successful installation

Abort

Previous

Install

Vemos el **proceso de instalación** que **tardará muy poco tiempo**.

Virtualization Platform

Open Source Virtualization Platform

- Enterprise ready
- Central Management
- Clustering
- Online Backup solution
- Live Migration
- 32 and 64 bit guests

Visit **www.proxmox.com** for additional information and the Wiki about Proxmox VE.

Container Virtualization

Only 1-3% performance loss using OS virtualization as compared to using a standalone server.

Full Virtualization (KVM)

Run unmodified virtual servers - Linux or Windows.

creating root filesystem
5%

Abort

Install

2.2.4 Configuración tras la instalación

Cuando se nos reinicie, entramos en la dirección ip y el puerto que nos dice para configurar el servidor y crear los contenedores, que en este caso es `https://172.16.20.3:8006`.

```
Welcome to the Proxmox Virtual Environment. Please use your web browser to
configure this server - connect to:
```

```
https://172.16.20.3:8006/
```

```
pve login: _
```

Al poner la dirección IP en un cliente conectado a la misma red nos sale **Advertencia: riesgo potencial de seguridad a continuación**. Hacemos clic en **Avanzado...**

 **Advertencia: riesgo potencial de seguridad a continuación**

Firefox ha detectado una posible amenaza de seguridad y no ha cargado 172.16.20.3. Si visita este sitio, los atacantes podrían intentar robar información como sus contraseñas, correos electrónicos o detalles de su tarjeta de crédito.

¿Qué puede hacer al respecto?

El problema está probablemente en el sitio web, y no hay nada que pueda hacer para resolverlo.

Si está en una red corporativa o utilizando un antivirus, puede ponerse en contacto con el equipo de asistencia para obtener ayuda. También puede notificar el problema al administrador del sitio web.

[Más información...](#)

[Retroceder \(recomendado\)](#) [Avanzado...](#)

Vemos que **Firefox no confía** en el **certificado del sitio**, porque la **entidad certificadora no la reconoce** y hacemos clic en **Aceptar el riesgo y continuar**.

Alguien podría estar tratando de suplantar el sitio y usted no debería continuar.

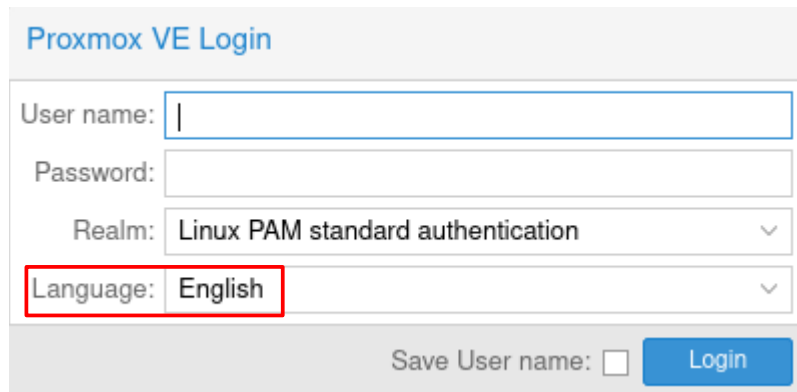
Los sitios web prueban su identidad mediante certificados. Firefox no confía en 172.16.20.3:8006 porque su emisor del certificado es desconocido, el certificado es autofirmado o el servidor no está enviando los certificados intermediarios correctos.

Código de error: `SEC_ERROR_UNKNOWN_ISSUER`

[Ver certificado](#)

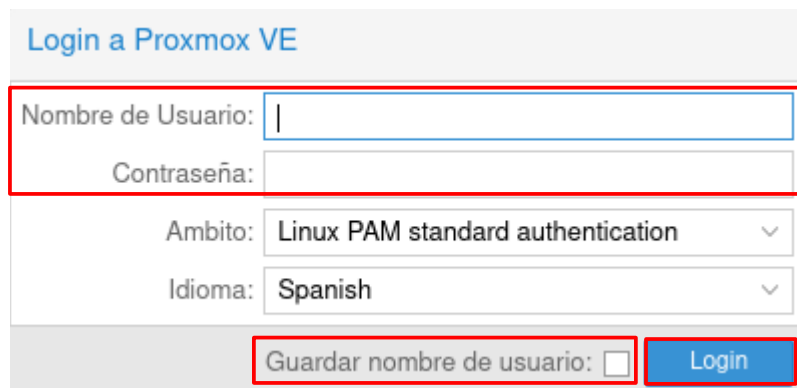
[Retroceder \(recomendado\)](#) [Aceptar el riesgo y continuar](#)

Cambiamos el idioma haciendo clic en Language y ponemos Spanish o Español.



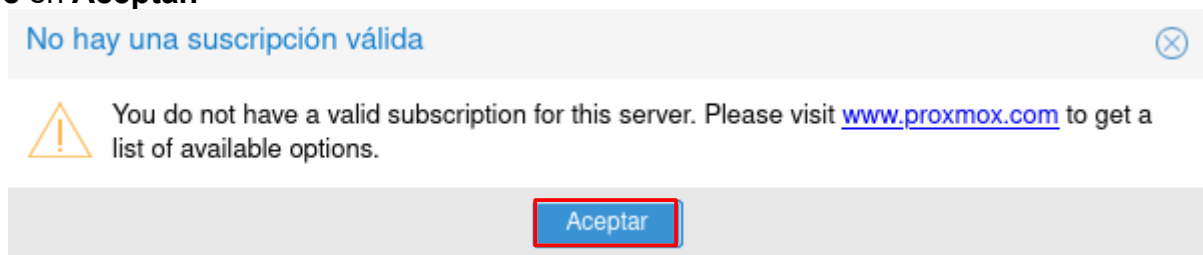
The image shows the 'Proxmox VE Login' form. It has a title 'Proxmox VE Login' in blue. Below the title are four input fields: 'User name:' with a text box, 'Password:' with a text box, 'Realm:' with a dropdown menu showing 'Linux PAM standard authentication', and 'Language:' with a dropdown menu showing 'English'. The 'Language:' dropdown is highlighted with a red box. At the bottom, there is a checkbox labeled 'Save User name:' and a blue 'Login' button.

En **Nombre de Usuario** ponemos **root** que es el **administrador**, que **puede hacer de todo** y es el **usuario** que hemos **creado en la instalación**, **ponemos la contraseña de ese usuario**, **dejamos el Ámbito en Linux PAM standard authentication**, porque **solo vamos a usar ese usuario por ahora**, podemos **guardar el nombre de usuario** haciendo **clic en la casilla correspondiente** y hacemos **clic en Login** para **iniciar sesión**.



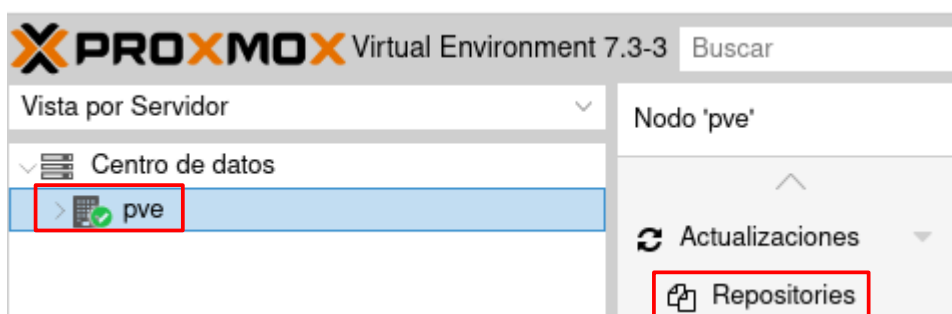
The image shows the 'Login a Proxmox VE' form. It has a title 'Login a Proxmox VE' in blue. Below the title are four input fields: 'Nombre de Usuario:' with a text box, 'Contraseña:' with a text box, 'Ambito:' with a dropdown menu showing 'Linux PAM standard authentication', and 'Idioma:' with a dropdown menu showing 'Spanish'. The 'Nombre de Usuario:' and 'Contraseña:' text boxes are highlighted with a red box. At the bottom, there is a checkbox labeled 'Guardar nombre de usuario:' and a blue 'Login' button.

Nos sale que no hay una suscripción válida porque no la hemos pagado y hacemos clic en Aceptar.

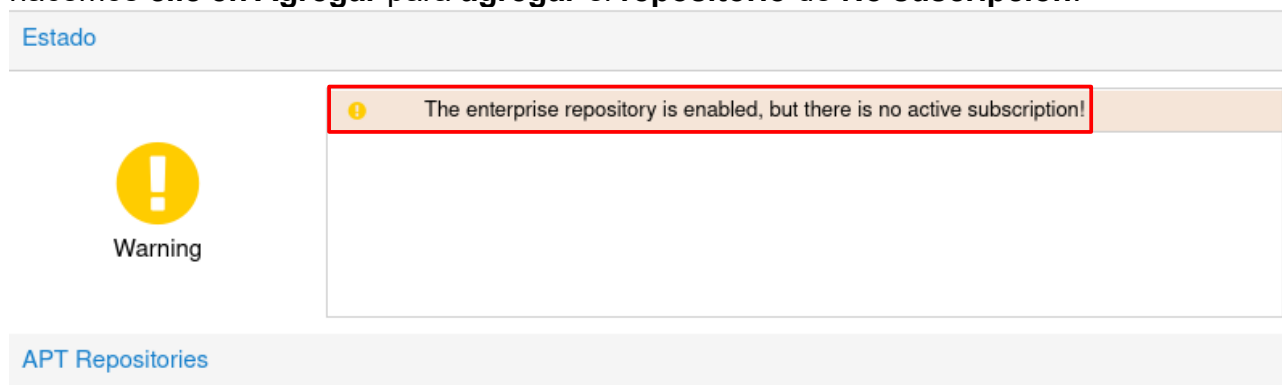


The image shows a message box with a title bar that says 'No hay una suscripción válida' and a close button (X). The message text is: 'You do not have a valid subscription for this server. Please visit www.proxmox.com to get a list of available options.' There is a yellow warning icon (triangle with exclamation mark) to the left of the text. At the bottom, there is a blue 'Aceptar' button.

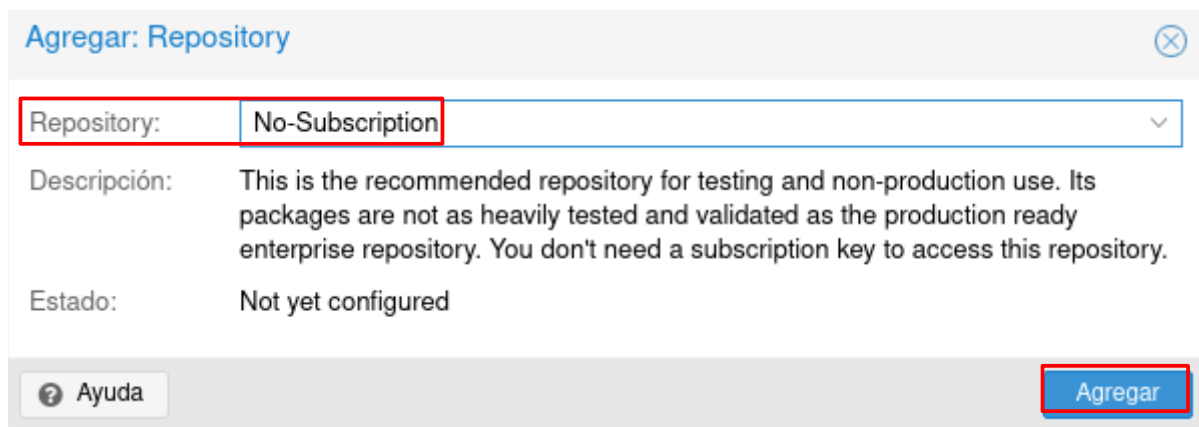
Hacemos **clic** en **nuestro servidor** que es **pve** y hacemos **clic** en **Repositories** para **habilitar** el **repositorio** de **no suscripción**.



Nos **sale** que el **repositorio enterprise** está **activo** pero que **no hay suscripción**, hacemos **clic** en **Agregar** para **agregar** el **repositorio** de **No suscripción**.



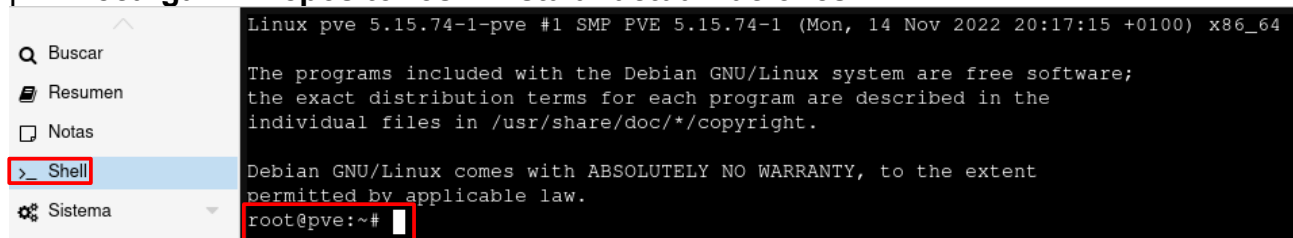
Seleccionamos en **Repository No-Subscription** y hacemos **clic** en **Agregar**



Hacemos **clik** en la **casilla** del **repositorio** **enterprise.proxmox.com** y hacemos **clik** en **Disable**.



Hacemos **clik** en **Shell** y **actualizamos** el **proxmox** con **apt update && apt upgrade** para **recargar** los **repositorios** e **instalar actualizaciones**.



Presionamos Y para **instalar** las **actualizaciones**.

Ponemos apt dist-upgrade para **actualizar** la **distribución** y **presionamos** también **Y** para **continuar**.

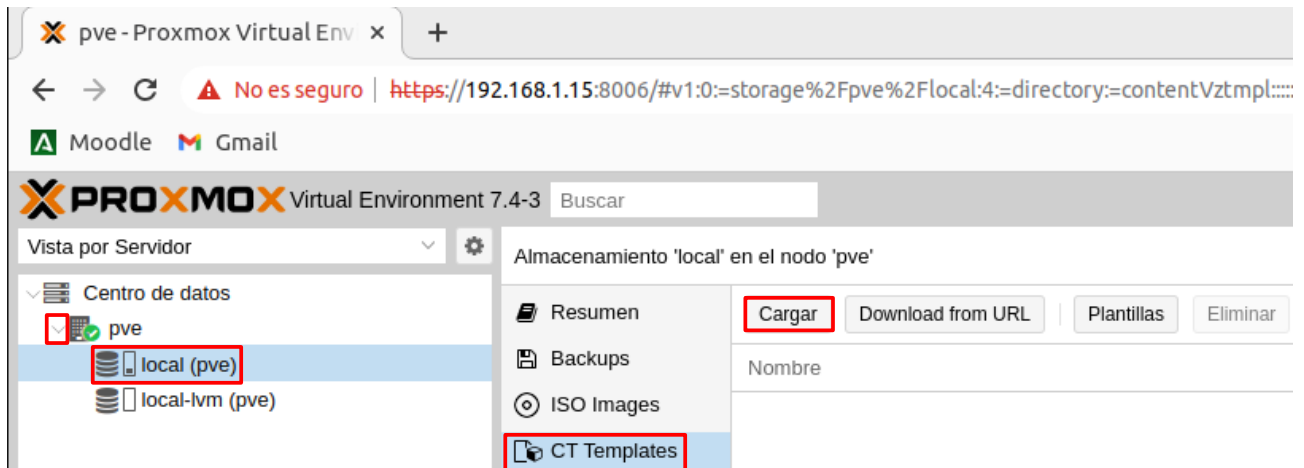
2.2.5 Descarga y creación del contenedor

Entramos en la siguiente URL para descargar los contenedores de proxmox.
<http://download.proxmox.com/images/system/>

Index of /images/system/ x +			
No es seguro download.proxmox.com/images/system/			
Moodle Gmail			
gentoo-current-default 20200310 amd64.tar.xz	25-Apr-2020	13:57	266501136
gentoo-current-openrc 20220622 amd64.aplinfo	23-Jun-2022	14:26	501
gentoo-current-openrc 20220622 amd64.tar.xz	22-Jun-2022	17:14	286606860
opensuse-15.0-default 20180907 amd64.tar.xz	07-Sep-2018	10:26	65127228
opensuse-15.1-default 20190719 amd64.tar.xz	19-Jul-2019	14:15	38630420
opensuse-15.2-default 20200824 amd64.tar.xz	24-Aug-2020	04:31	38154084
opensuse-15.3-default 20210925 amd64.aplinfo	29-Sep-2021	15:37	503
opensuse-15.3-default 20210925 amd64.tar.xz	25-Sep-2021	05:27	40009704
opensuse-15.4-default 20221109 amd64.aplinfo	09-Nov-2022	16:49	503
opensuse-15.4-default 20221109 amd64.tar.xz	09-Nov-2022	05:18	41548280
opensuse-42.2-default 20170406 amd64.tar.xz	06-Apr-2017	01:25	52470844
opensuse-42.3-default 20171214 amd64.tar.xz	15-Dec-2017	05:43	52786256
rockylinux-8-default 20210929 amd64.aplinfo	29-Sep-2021	15:37	502
rockylinux-8-default 20210929 amd64.tar.xz	29-Sep-2021	03:42	112551304
rockylinux-9-default 20221109 amd64.aplinfo	09-Nov-2022	16:05	502
rockylinux-9-default 20221109 amd64.tar.xz	09-Nov-2022	02:28	102704656
ubuntu-12.04-standard 12.04-1 amd64.tar.gz	16-Mar-2017	15:58	122350417
ubuntu-14.04-standard 14.04-1 amd64.tar.gz	16-Mar-2017	15:58	144858568
ubuntu-14.04-standard 14.04.5-1 amd64.tar.gz	24-Jan-2019	07:41	147917095
ubuntu-16.04-standard 16.04-1 amd64.tar.gz	16-Mar-2017	15:58	197123375
ubuntu-16.04-standard 16.04.5-1 amd64.aplinfo	24-Jan-2019	07:47	573
ubuntu-16.04-standard 16.04.5-1 amd64.tar.gz	24-Jan-2019	07:32	210160363
ubuntu-16.10-standard 16.10-1 amd64.tar.gz	16-Mar-2017	15:58	194375872
ubuntu-17.04-standard 17.04-1 amd64.tar.gz	02-May-2017	05:16	201336676
ubuntu-17.10-standard 17.10-1 amd64.tar.gz	12-Dec-2017	11:15	206248020
ubuntu-18.04-standard 18.04-1 amd64.tar.gz	30-Apr-2018	07:59	211328062
ubuntu-18.04-standard 18.04.1-1 amd64.aplinfo	24-Jan-2019	07:47	573
ubuntu-18.04-standard 18.04.1-1 amd64.tar.gz	24-Jan-2019	07:32	213430501
ubuntu-18.10-standard 18.10-1 amd64.tar.gz	22-Oct-2018	09:22	217327250
ubuntu-18.10-standard 18.10-2 amd64.tar.gz	24-Jan-2019	07:41	218147062
ubuntu-19.04-standard 19.04-1 amd64.tar.gz	26-Apr-2019	09:24	213467952
ubuntu-19.10-standard 19.10-1 amd64.tar.gz	17-Oct-2019	16:19	219093821
ubuntu-20.04-standard 20.04-1 amd64.aplinfo	25-Apr-2020	14:43	578
ubuntu-20.04-standard 20.04-1 amd64.tar.gz	25-Apr-2020	13:58	214203058
ubuntu-20.10-standard 20.10-1 amd64.tar.gz	16-Nov-2020	16:18	221056452
ubuntu-21.04-standard 21.04-1 amd64.tar.gz	25-Apr-2021	16:13	225859786
ubuntu-21.10-standard 21.10-1 amd64.tar.zst	12-Nov-2021	14:27	125886811
ubuntu-22.04-standard 22.04-1 amd64.aplinfo	24-Apr-2022	11:18	612
ubuntu-22.04-standard 22.04-1 amd64.tar.zst	24-Apr-2022	10:24	129824858
ubuntu-22.10-standard 22.10-1 amd64.aplinfo	21-Oct-2022	15:03	611
ubuntu-22.10-standard 22.10-1 amd64.tar.zst	21-Oct-2022	15:00	129577718

Seleccionamos el que dice **ubuntu 22.04** y acaba en punto **zst**, porque **vemos** que ocupa más tamaño.

Cuando se descargue la plantilla del contenedor.

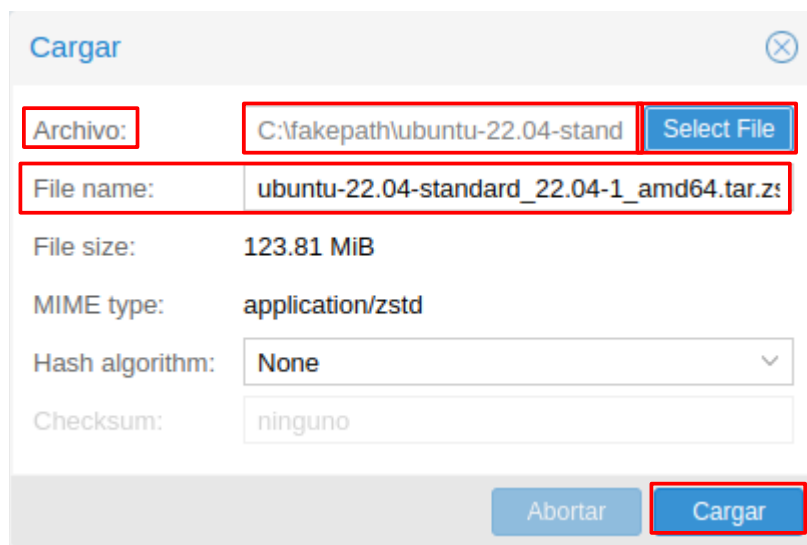


Hacemos clic en la **flecha que hay al lado de pve** para **desplegar el menú de almacenamiento y contenedores**.

Hacemos **clic** después en **local (pve)** para **gestionar el almacenamiento de nuestro proxmox**.

Hacemos **clic** después en **CT Templates** para **gestionar nuestras plantillas de contenedores**.

Para **cargar** nuestras **plantillas de contenedores** hacemos **clic** en **Cargar**.



Hacemos **clic** en **Select File** para **seleccionar** nuestra **plantilla de contenedor** que hemos **descargado** en **Archivo**.

En **File name**, sale el **nombre** de nuestra **plantilla de contenedor**.

Hacemos **clic** en **Cargar** para **cargar** nuestra **plantilla de contenedor** a **Proxmox**.

Cuando se acabe de cargar la plantilla de contenedor.

En **Task viewer: Copiar datos** vemos que pone **TASK OK** porque se ha cargado correctamente el archivo y cerramos la ventana.



Vemos la plantilla de contenedor ya cargado en el proxmox.

Resumen	Cargar	Download from URL	Plantillas	Eliminar	Buscar:	Name, Format
Backups	Nombre				Fecha	Formato
ISO Images	ubuntu-22.04-standard_22.04-1_amd64.tar.zst				2023-05-04 19:34:24	tzst
CT Templates						129.82 MB

Hacemos clic arriba a la derecha en **Crear CT** para crear nuestro contenedor.



En **Nodo**, dejamos **pve** porque **solo tenemos un nodo**, en **CT ID**, ponemos el **ID** de **nuestro contenedor**, que en este caso **es 100**, el **nombre de host** ponemos el **nombre** que se **identificará** nuestro **contenedor**, **desmarcamos** la **casilla Contenedores sin privilegios**, no **ponemos ningún Conjunto de Recursos**, porque **solo lo va a usar el administrador**, **ponemos** nuestra **contraseña** de **administrador** en el **contenedor** y la **confirmamos** en **Confirmar contraseña** y hacemos **clic** en **Siguiente** para **ir al siguiente paso**.

Crear: Contenedor LXC

General Plantilla Discos CPU Memoria Red DNS Confirmar

Nodo: pve

CT ID: 100

Nombre del Host: proyectopgg

Contenedores sin privilegios: ☐

Nesting: ☒

Conjunto de Recursos:

Contraseña:

Confirmar contraseña:

Clave pública SSH:

Carga archivo de clave SSH

Ayuda Avanzado ☐ Atrás Siguiente

En **Plantilla**, en **Almacenamiento** dejamos **local** ya que **solo tenemos** un **disco de almacenamiento** y en **Plantilla seleccionamos** nuestra **plantilla de contenedor** que hemos **cargado antes** al **proxmox**. Hacemos **clic** en **Siguiente** para continuar.

Crear: Contenedor LXC

General **Plantilla** Discos CPU Memoria Red DNS Confirmar

Almacenamiento: local

Plantilla: ubuntu-22.04-standard_22.04-1_a

Ayuda

Avanzado ☐

Atrás

Siguiente

En **Discos** en **Almacenamiento** dejamos el que está para **almacenar** los **datos** del **contenedor** ya que **solo tenemos uno**.

En **Tamaño de disco (GiB)** ponemos **16 GiB**, ya que es el **espacio** que **necesitaremos** en **nuestro contenedor**.

Hacemos **clic** en **Siguiente** para **continuar**.

The screenshot shows the 'Crear: Contenedor LXC' window with the 'Discos' tab selected. The 'Almacenamiento' dropdown is set to 'local-lvm'. The 'Tamaño de disco (GiB)' dropdown is set to '16'. The 'rootfs' entry is listed in the left sidebar. At the bottom, the 'Siguiente' button is highlighted with a red box.

Crear: Contenedor LXC

General Plantilla **Discos** CPU Memoria Red DNS Confirmar

rootfs Almacenamiento: local-lvm

Tamaño de disco (GiB): 16

+ Agregar

Ayuda Avanzado ☐ Atrás **Siguiente**

En **CPU** dejamos **1 núcleo** ya que **tenemos más que suficiente** para **nuestros servicios** y hacemos **click** en **Siguiente**.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Núcleos:

1

Ayuda

Avanzado ☐

Atrás

Siguiente

En **Memoria (MiB)** ponemos **1536 MiB**, ya que es **más que suficiente** para nuestras **tareas** y en **Swap (MiB)** ponemos lo mismo también.

Hacemos clic en **Siguiente** para continuar.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Memoria (MiB):

1536

Swap (MiB):

1536

Ayuda

Avanzado ☐

Atrás

Siguiente

En **Red**, dejamos el **Nombre** como está porque **solo vamos a tener una tarjeta de red**, en **Puente** seleccionamos nuestro **adaptador de red** que da **conexión a internet**, en este caso es **vmbr0**, **desmarcamos** la **casilla de Cortafuego**, en **IPv4** dejamos **Estático**, ya que lo **necesitaremos** así para **nuestros servicios de red**, en **IPv4/CIDR** ponemos **nuestra dirección IP/nuestra máscara de subred** como **número decimal**, en **Puerta de enlace (IPv4)** ponemos la **dirección de nuestro router** para **tener internet**.

Si **nuestra operadora tiene IPv6** tocamos también **IPv6**, de lo **contrario** hacemos **clic** en **Siguiente**.

Crear: Contenedor LXC

General Plantilla Discos CPU Memoria **Red** DNS Confirmar

Nombre:

Dirección MAC:

Puente:

Etiqueta VLAN:

Cortafuego: ☐

IPv4: ☒ Estático ☐ DHCP

IPv4/CIDR:

Puerta de enlace (IPv4):

IPv6: ☒ Estático ☐ DHCP ☐ SLAAC

IPv6/CIDR:

Puerta de enlace (IPv6):

Avanzado ☐

En **DNS**, por ahora en **Dominio DNS** y **Servidores DNS** usamos las **configuraciones** del **host** hasta que **instalemos** los **servicios DNS** y hacemos **click** en **Siguiente**.

Crear: Contenedor LXC ✕

General Plantilla Discos CPU Memoria Red **DNS** Confirmar

Dominio DNS:

usar configuraciones del host

Servidores DNS:

usar configuraciones del host

Avanzado ☐

Atrás

Siguiente

En **Confirmar** vemos los **valores que hemos puesto**, podemos **marcar** la **casilla Start after created** para **iniciar** el **contenedor** una vez **creado** y hacemos **clic** en **Finalizar**.

Crear: Contenedor LXC ✕

General Plantilla Discos CPU Memoria Red DNS **Confirmar**

Key ↑	Value
cores	1
hostname	proyectopgg
memory	1536
net0	name=eth0,bridge=vmbr0,ip=192.168.56.4/24,gw=192.168.56.1
nodename	pve
ostemplate	local:vztmpl/ubuntu-22.04-standard_22.04-1_amd64.tar.zst
pool	
rootfs	local-lvm:16
swap	1536
vmid	100

☒ Start after created

Avanzado ☐ **Atrás** **Finalizar**

Cuándo ponga **TASK OK** cerramos la ventana.

Task viewer: CT 100 - Crear

Salida

Estado

Parar

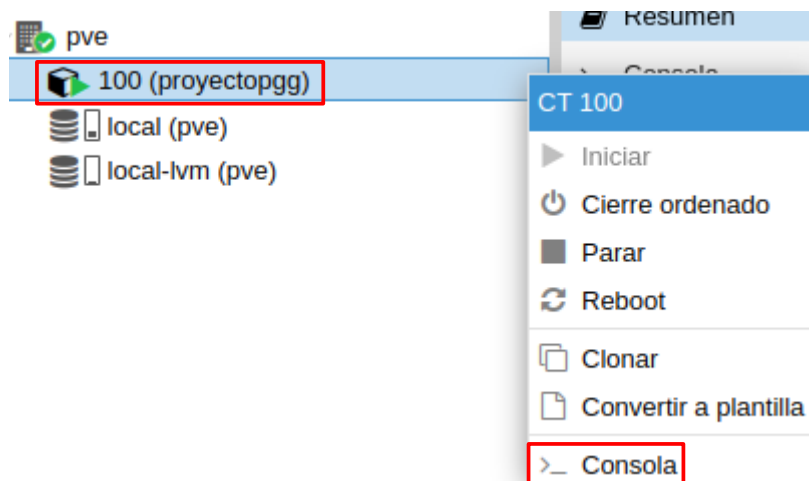
Descargar

```

Logical volume "vm-100-disk-0" created.
Creating filesystem with 4194304 4k blocks and 1048576 inodes
Filesystem UUID: 140f5a10-d64f-45bf-98c0-06eb0703e559
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
extracting archive 'var/lib/vz/template/cache/ubuntu-22.04-standard_22.04-1_amd64.tar.zst'
Total bytes read: 508579840 (486MiB, 142MiB/s)
Detected container architecture: amd64
Creating SSH host key 'ssh_host_ecdsa_key' - this may take some time ...
done: SHA256:Nfks2UWptVTzoOQfV+zu3qch+yyzkO2mgNBe9l0xkSA root@proyectopgg
Creating SSH host key 'ssh_host_ed25519_key' - this may take some time ...
done: SHA256:WtCxfwWr82607w1ZomfsGxBxqVLqm+X+ODvJMBTAig root@proyectopgg
Creating SSH host key 'ssh_host_rsa_key' - this may take some time ...
done: SHA256:gEqBk2SJFoOEJmEnq03XHi0hbKrn+jl2o4peGDY/8iU root@proyectopgg
Creating SSH host key 'ssh_host_dsa_key' - this may take some time ...
done: SHA256:yyCja6zRgkBwX0FdyOnvAP4b7Z016IEVTixP+tbTR0A root@proyectopgg
TASK OK

```

Para **utilizar el contenedor**, hacemos **clic derecho** en el **contenedor** que **acabamos de crear** y hacemos **clic** en **Consola**.



Ponemos **nuestro** nombre de **usuario** que es **root** en **nombre de máquina login** y ponemos nuestra **contraseña** para **entrar** en **Password**.

```
Ubuntu 22.04 LTS proyectopgg tty1
proyctopgg login: root
Password: 
```

Vemos que **ya podemos empezar a instalar los servicios**.

```
root@proyctopgg:~# 
```

2.2.6 Configuraciones Iniciales del contenedor

Ponemos **sudo apt update && sudo apt upgrade && sudo apt dist-upgrade** para **cargar los paquetes** que hay que **actualizar de los repositorios**, **actualizar los paquetes** y **actualizar la distribución**.

Vemos que **112 paquetes van a ser actualizados**.

```
112 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Calculating upgrade... Done
The following NEW packages will be installed:
systemd-hwe-hwdb
The following packages will be upgraded:
apparmor apt apt-utils base-files bind9-dnsutils bind9-host bind9-libs ca-certificates dbus distro-info-data dmidecode dpgk e2fsprogs
gcc-12-base gpgv gzip isc-dhcp-client isc-dhcp-common kbd less libapparmor1 libapt-pkg6.0 libbpf0 libc-bin libc6 libcom-err2
libcryptsetup12 libdbus-1-3 libdrm-common libdrm2 libexpat1 libext2fs2 libfribidi0 libgcc-s1 libglib2.0-0 libglib2.0-data libgnutls30
libgssapi-krb5-2 libjson-c5 libk5crypto3 libkrb5-3 libkrb5support0 libnetplan0 libnftables1 libnss-systemd libntfs-3g89 libpam-modules
libpam-modules-bin libpam-runtime libpam-systemd libpam0g libpcre2-8-0 libpcre3 libpython3-stdlib libpython3.10-minimal
libpython3.10-stdlib libsass2-2 libsass2-modules-db libsasl2-3 libss2 libssl3 libstdc++6 libsystemd0 libtirpc-common libtirpc3
libudev1 libusb-1.0-0 libxml2 locales login logrotate logsave netplan.io networkd-dispatcher nftables ntfs-3g openssh-client
openssh-server openssh-sftp-server openssl passwd perl-base postfix python-apt-common python3 python3-apt python3-distupgrade
python3-gdbm python3-gi python3-minimal python3-pkg-resources python3-update-manager python3.10 python3.10-minimal rsync rsyslog ssh
sudo systemd systemd-sysv systemd-timesyncd tar tcpdump tzdata ubuntu-advantage-tools ubuntu-release-upgrader-core udev
update-manager-core vim-common vim-tiny xxd zlib1g
112 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
54 standard security updates
Need to get 50.0 MB of archives.
After this operation, 1068 kB disk space will be freed.
Do you want to continue? [Y/n] Y
```

Vemos que **un paquete nuevo va a ser instalado**.

Presionamos Y para continuar.

Ponemos **reboot** para **reiniciar nuestro contenedor**.

Vemos nuestra **fecha y hora** con **date**.

```
root@proyctopgg:~# date
Fri May 5 06:55:40 UTC 2023
```

Vemos que nuestra **hora** está mal.

Vemos los **husos horarios** de **Ubuntu** con **timedatectl list-timezones**

Vemos que está **Europe/Madrid**.

Presionamos **q** para salir.

Para **cambiar** la zona horaria a Europa/Madrid, ponemos el siguiente comando:

sudo timedatectl set-timezone Europe/Madrid

Vemos que el **poner date** otra vez ya tenemos la hora correcta.

```
root@proyectopgg:~# date
Fri May 5 09:04:57 CEST 2023
```

Reiniciamos otra vez el contenedor.

2.2.7 Instalamos el servidor DHCP y vemos la IP

Ponemos **sudo apt install isc-dhcp-server** para instalar nuestro servidor dhcp.

Vemos que **van a ser instalados 3 paquetes nuevos**.

Presionamos Y para continuar.

```
root@proyectopgg:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 1a:7d:de:07:45:39 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.56.4/24 brd 192.168.56.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::187d:deff:fe07:4539/64 scope link
        valid_lft forever preferred_lft forever
```

Con **ip a** vemos nuestra tarjeta de red que es **eth0@if9**, nuestra IP que es **192.168.56.4**, nuestro broadcast o última dirección es **192.168.56.255**, significa que nuestra máscara de subred es **255.255.255.0**.

2.2.8 Configuramos el servidor DHCP y comprobamos que funciona

Abrimos el siguiente archivo para configurar el servidor DHCP con **sudo nano /etc/dhcp/dhcpd.conf**

```
subnet 192.168.56.0 netmask 255.255.255.0 {
    range 192.168.56.150 192.168.56.200;
    option routers 192.168.56.4;
    option domain-name-servers 208.67.222.222, 208.67.220.220;
}
```

En **subnet**: Ponemos la **dirección** de nuestra red, que **acaba en 0**, en nuestro caso es **192.168.56**

En **netmask**: Ponemos la máscara de subred en decimal, con números de **0** a **255** separado por **3 puntos**. La máscara de subred es más alta cuanto menos equipos queramos conectar a la red.

En **range**: Ponemos el número de direcciones que puede dar el DHCP y desde que número puede darlas hasta que número puede darlas.

En este caso ponemos **192.168.56.150** a la **192.168.56.200**, que son **50 dispositivos** a conectar.

En **option routers**: Ponemos la dirección **IP** de nuestro **contenedor**, que es la que hemos visto antes para que **tenga internet** el **cliente** del **dhcp**.

En **option domain-name-servers**: Ponemos los **DNS** que tengan los **clientes separados por coma**.

Guardamos el fichero con Ctrl + X como vemos abajo:

```
subnet 192.168.56.0 netmask 255.255.255.0 {
    range 192.168.56.150 192.168.56.200;
    option routers 192.168.56.4;
    option domain-name-servers 208.67.222.222, 208.67.220.220;
}
```

^G Help ^O Write Out ^W Where Is ^K Cut
 ^X Exit ^R Read File ^\ Replace ^U Paste

En **Save modified buffer?** Que es para **guardar** el fichero, ponemos **Y Yes** para guardar el fichero.

En **File Name to Write** dejamos el que **está** y **seleccionamos enter** para **guardar**.

Reiniciamos el servicio con

sudo systemctl restart isc-dhcp-server

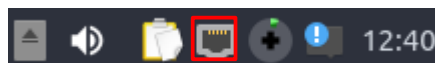
Vemos que funciona correctamente con sudo systemctl status isc-dhcp-server

```
root@proyectopgg:~# sudo systemctl status isc-dhcp-server
* isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-05-05 12:28:27 CEST; 1min 25s ago
```

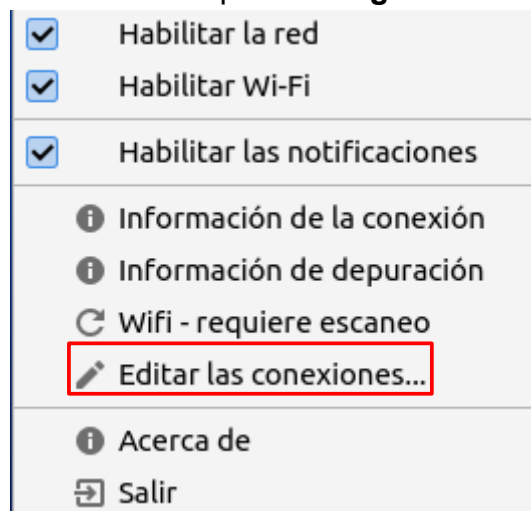
Vemos que pone en Active: active (running) por lo que **funciona correctamente**.

Configuramos el cliente que en este caso es Lubuntu como DHCP:

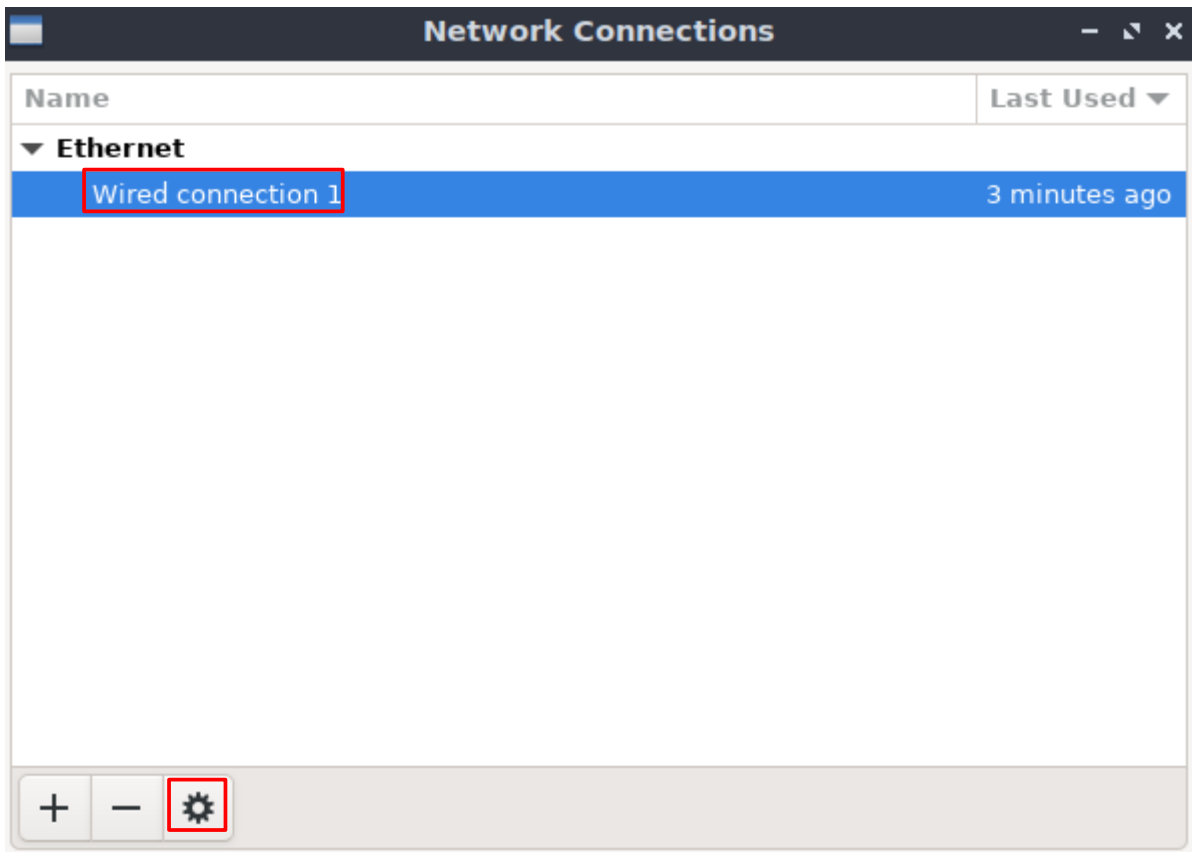
Hacemos **clic derecho** en el **icono de red ethernet**:



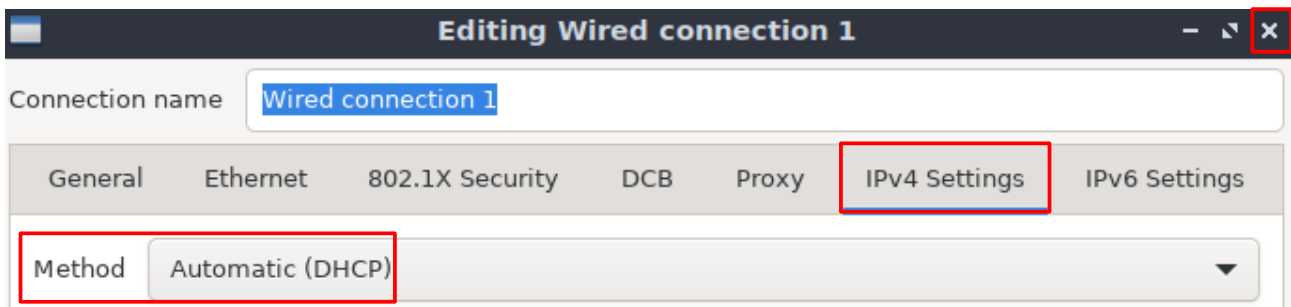
Hacemos **clic** en **Editar** las **conexiones** para **configurarlo como DHCP**



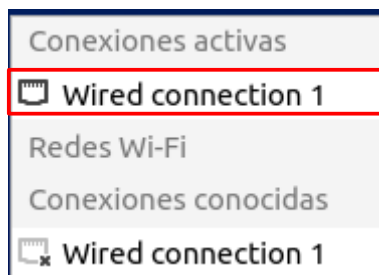
Seleccionamos en **Ethernet** nuestra **única conexión de red** la **Wired connection 1** y hacemos **clic** en el **icono de configuración** para **configurarlo como DHCP**.



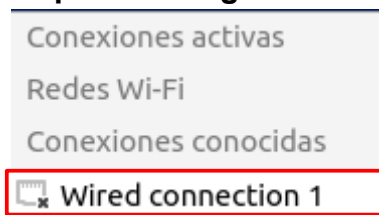
Hacemos **clic** en **IPv4 Settings** que **son** los **ajustes de IPv4** y en **Method** tiene que poner **Automatic(DHCP)**, que **significa** que está **configurado como DHCP**. Cerramos la ventana **haciendo clic en la X**.



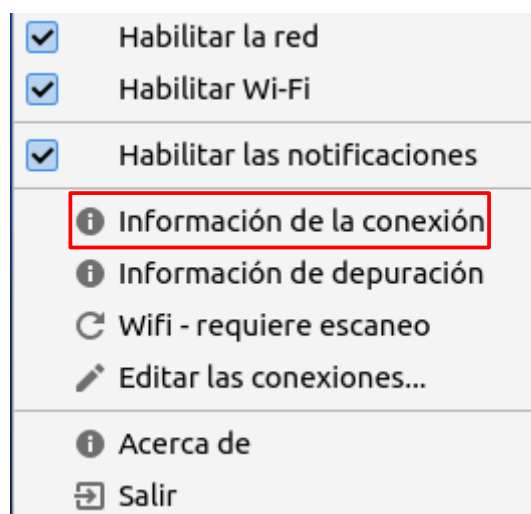
Hacemos clic en el **icono de Ethernet** y en **Conexiones activas** hacemos clic en **Wired connection 1**



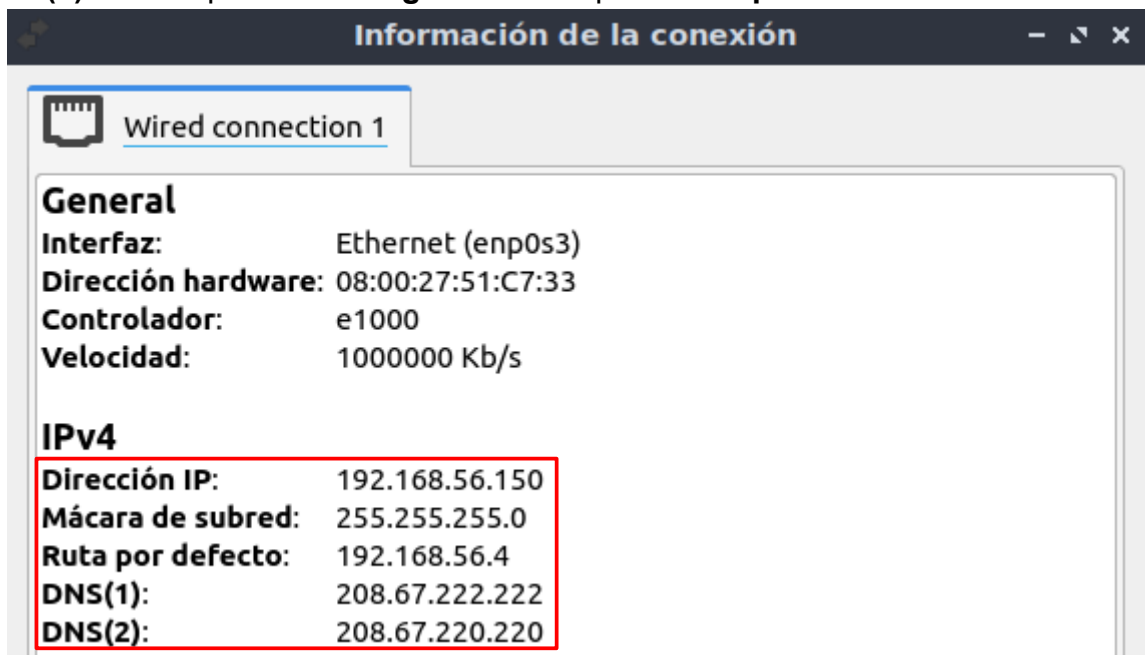
Hacemos **otra vez clic** en el **icono de Ethernet** y en **Conexiones conocidas** hacemos clic en **Wired connection 1** para que nos asigne la dirección IP el DHCP.



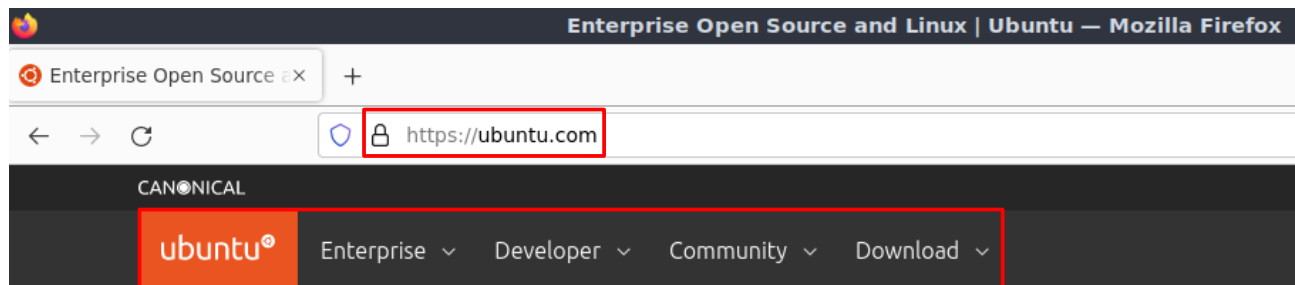
Hacemos **otra vez clic derecho** en el **icono de Ethernet** y **seleccionamos Información de la conexión**.



En **Dirección IP**, vemos que es la **primera dirección IP** del **rango de direcciones** del **DHCP**, en **Máscara de subred** vemos que es la **misma** que **tenemos** en el **DHCP**, en **Ruta por defecto** vemos que es la **dirección IP** que **hemos puesto** en **option routers** del **DHCP**, en **DNS(1)** vemos que **sale el primer DNS** que **hemos puesto** en el **DHCP** y en **DNS(2)** vemos que **sale el segundo DNS** que **hemos puesto** en el **DHCP**.



Comprobamos que **tenemos internet** en el **cliente del DHCP** entrando por ejemplo a <https://ubuntu.com>



2.2.9 Instalamos y configuramos el servidor Web

Ponemos `sudo apt install apache2` para instalar el servidor web.

```
root@proyectopgg:~# sudo apt install apache2
```

Se instalarán 22 paquetes nuevos y presionamos Y para instalar el servidor web.

Configuramos el servidor Web

Desactivamos el sitio web HTTP con `sudo a2dissite 000-default.conf`

Vemos que el sitio HTTP ha sido deshabilitado.

```
Site 000-default disabled.
```

Activamos el módulo SSL para tener HTTPS con `sudo a2enmod ssl`

Vemos el módulo SSL ya activado.

```
Enabling module ssl.
```

Activamos el sitio web https con `sudo a2ensite default-ssl.conf`

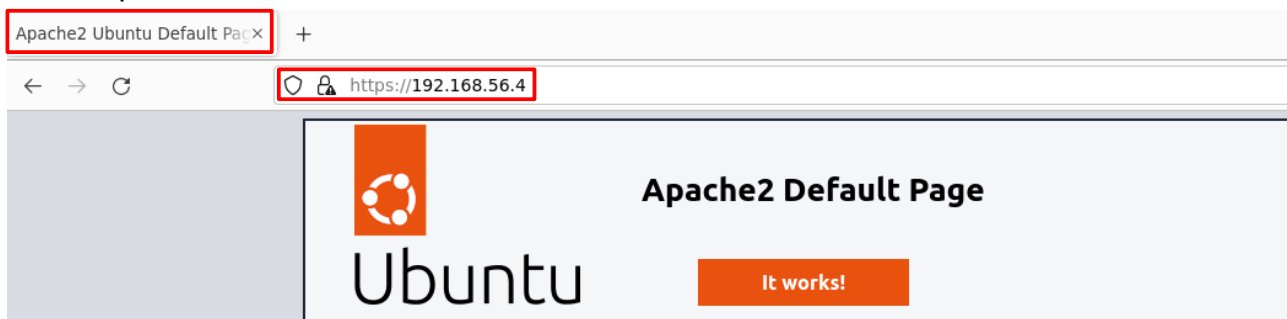
Vemos el sitio web ssl ya activado.

```
Enabling site default-ssl.
```

Reiniciamos el servidor apache con `sudo systemctl restart apache2`

Comprobamos que funcione en el cliente abriendo el navegador y poniendo https://ip_servidor

Vemos que funciona correctamente en el cliente.



Editamos el archivo de configuración con `sudo nano /etc/apache2/sites-available/default-ssl.conf`

Podemos cambiar la carpeta de la página web en DocumentRoot.

```
GNU nano 6.2 /etc/apache2/sites-available/default-ssl.conf
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html
```

Podemos poner esta carpeta por ejemplo en DocumentRoot

```
DocumentRoot /var/www/pgg
```

Guardamos el archivo con el mismo nombre.

Creamos la carpeta de nuestra página web con `sudo mkdir /var/www/pgg`

Creamos el archivo `index.html` para que muestre nuestra página web al entrar en la dirección IP con `sudo nano /var/www/pgg/index.html`

```
root@proyectopgg:~# sudo nano /var/www/pgg/index.html
```

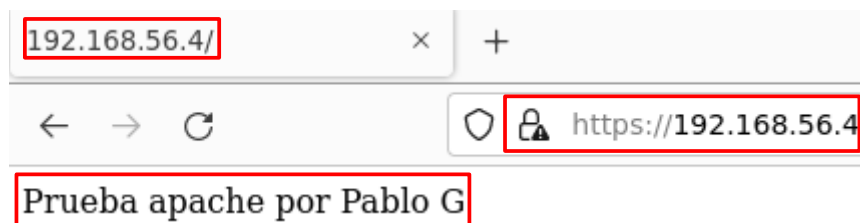
Ponemos `<p>` que es abrir un párrafo en HTML, nuestro texto y `</p>` para cerrar un párrafo en HTML.

```
GNU nano 6.2 /var/www/pgg/index.html
<p>Prueba apache por Pablo G</p>
```

Guardamos el archivo con el mismo nombre con el que lo hemos creado.

Reiniciamos `apache2` de nuevo.

Vemos que se ve correctamente la página.



Copiamos el `default-ssl` que es el original a la misma carpeta y le ponemos de nombre `proyectopgg` con `sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/proyectopgg.conf` esto es para no perder el original y poder recuperarlo en caso necesario.

Deshabilitamos el archivo de configuración antiguo para que no interfiera con el nuevo con `sudo a2dissite default-ssl.conf`

```
Site default-ssl disabled.
```

Creamos un usuario para que solo puedan entrar los usuarios autorizados con `sudo htpasswd -c /etc/apache2/.htpasswd proyectopgg` el `-c` es para crear el fichero de autenticación, `/etc/apache2/.htpasswd` será el fichero de autenticación, `proyectopgg` es el usuario que se autenticará.

Ponemos una contraseña en `New password` para crearla.

Repetimos la contraseña en `Re-type new password`:

Creamos otro usuario con `sudo htpasswd /etc/apache2/.htpasswd proyectopgg2`, quitamos `-c` porque si no se nos sobrescribe el fichero y se nos borra el primer usuario.

Hacemos lo mismo que con el otro usuario para introducir la contraseña.

Vemos el archivo de contraseñas guardadas de apache con `sudo cat /etc/apache2/.htpasswd`

```
root@proyectopgg:~# sudo cat /etc/apache2/.htpasswd
proyectopgg:$apr1$vSqCK3d.$RfWljacJfdyVo.1iYdF4R1
proyectopgg2:$apr1$333Dctic$988qKmzFrrPG5ndUzRzWA1
```

Después del nombre de los usuarios y dos puntos, vemos la contraseña encriptada.

Borramos el archivo de antes con: `sudo rm -r /var/www/pgg/index.html`

Creamos una carpeta pública con: `sudo mkdir /var/www/pgg/publico`

Creamos una carpeta privada con: `sudo mkdir /var/www/pgg/privado`

Creamos la página web de inicio de la carpeta privada con

`sudo nano /var/www/pgg/privado/inicio.html`

Ponemos un texto de ejemplo que será:

`<h1>Prueba carpeta privada</h1>`

Creamos la página web de inicio de la carpeta pública con:

`sudo nano /var/www/pgg/publico/indice.html`

`<h1>Prueba carpeta pública</h1>`

Habilitamos el puerto 44300 con

`sudo nano /etc/apache2/ports.conf`

En `<IfModule ssl_module>`, que es el módulo de HTTPS:

Debajo de `Listen 443` que escucha el servidor HTTPS por el puerto 443, ponemos `Listen 44300`, para que escuche el servidor HTTPS por el puerto 44300.

```
<IfModule ssl_module>
    Listen 443
    Listen 44300
</IfModule>
```

Editamos nuestro archivo de configuración de HTTPS con

`sudo nano /etc/apache2/sites-available/proyectopgg.conf`


```

<IfModule mod_ssl.c>
    <VirtualHost default :44300>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/pgg
        <Directory /var/www/pgg/privado>
            DirectoryIndex inicio.html
            AllowOverride All
        </Directory>
        <Directory /var/www/pgg/publico>
            DirectoryIndex indice.html
            AllowOverride All
        </Directory>

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        <Directory "/var/www/pgg/privado">
            AuthType Basic
            AuthName "Contenido restringido proyecto"
            AuthUserFile /etc/apache2/.htpasswd
            Require valid-user
        </Directory>

```

En `<VirtualHost _default_:44300>` donde pone 44300 es el puerto que debemos poner en vez del 443 que es el que estaba previamente.

En `<Directory /var/www/pgg/privado>` es para que nos muestre la página web de nuestra carpeta privada al entrar en la carpeta privada.

`DirectoryIndex` es el nombre del archivo que mostrará al entrar en la carpeta privada o pública.

`AllowOverride All` sirve para que nos deje entrar en la carpeta.

`</Directory>` es para cerrar la carpeta.

`<Directory "/var/www/pgg/privado">` es nuestro directorio con autenticación.

`AuthType Basic` es el tipo básico de autenticación.

`AuthName "Contenido restringido proyecto"` es el nombre del mensaje que se mostrará en la ventana de autenticación.

`AuthUserFile /etc/apache2/.htpasswd` es el archivo donde están los usuarios y las contraseñas cifradas.

`Require valid-user` requiere un usuario válido con su contraseña correcta.

`</Directory>` es para cerrar la carpeta.

Habilitamos el sitio con `sudo a2ensite proyectopgg.conf`

Enabling site proyectopgg.

Reiniciamos el servidor apache otra vez con `sudo systemctl restart apache2`

2.2.10 Comprobamos que funciona la configuración

Nos metemos con la URL de la carpeta privada con el puerto que hemos puesto.

Q <https://192.168.56.4:44300/privado>

comenzar a usar Fi...

🌐 **192.168.56.4:44300**

Este sitio le pide que inicie sesión.

Nombre de usuario

proyectopgg

Contraseña

●●●●●●●●●●

Cancelar Iniciar sesión

Metemos nuestro nombre de usuario y la contraseña que hemos puesto al crear el usuario y hacemos clic en Iniciar sesión.

← → ↻ 🔒 <https://192.168.56.4:44300/privado/>

Prueba carpeta privada

Vemos que se ve perfectamente el contenido de la carpeta privada.

← → ↻ 🔒 <https://192.168.56.4:44300/publico/>

Prueba carpeta publica

Vemos que se ve perfectamente el contenido de la carpeta pública.

2.2.11 Instalamos jitsi

Antes de instalar jitsi hay que configurar el servidor DNS en el siguiente paso.

Deshabilitamos el sitio actual con `sudo a2dissite proyectopgg`

Reiniciamos apache2 con `sudo systemctl restart apache2`

Instalamos curl y gpg con `sudo apt install curl gpg`

```

The following additional packages will be installed:
dirmngr gnupg gnupg-l10n gnupg-utils gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libksba8 libnpth0 pinentry-curses
Suggested packages:
dbus-user-session pinentry-gnome3 tor parcimonie xloadimage sddm pinentry-doc
The following NEW packages will be installed:
dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libksba8 libnpth0
pinentry-curses
0 upgraded, 14 newly installed, 0 to remove and 0 not upgraded.
Need to get 2309 kB of archives.
After this operation, 5837 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

Vemos que se van a instalar 14 paquetes.

Presionamos Y para instalar gpg y curl.

Descargamos la clave gpg del repositorio de jitsi con curl
<https://download.jitsi.org/jitsi-key.gpg.key> -o jitsi-key.gpg.key

La añadimos a las claves del apt con sudo gpg --output /usr/share/keyrings/jitsi-key.gpg --dearmor jitsi-key.gpg.key

Creamos el archivo del repositorio de jitsi con sudo nano /etc/apt/sources.list.d/jitsi-stable.list

Añadimos el repositorio de jitsi con deb [signed-by=/usr/share/keyrings/jitsi-key.gpg] <https://download.jitsi.org/stable/> pegándolo en el archivo.

```

GNU nano 6.2 /etc/apt/sources.list.d/jitsi-stable.list
deb [signed-by=/usr/share/keyrings/jitsi-key.gpg] https://download.jitsi.org/stable/

```

Presionamos Ctrl + X para guardar el archivo.

Hacemos lo mismo con prosody.

Descargamos la clave gpg del repositorio de prosody con curl
<https://prosody.im/files/prosody-debian-packages.key> -o prosody-debian-packages.key

La añadimos a las claves del apt con sudo gpg --output /usr/share/keyrings/prosody-keyring.gpg --dearmor prosody-debian-packages.key

Creamos el archivo del repositorio de prosody con sudo nano /etc/apt/sources.list.d/prosody.list

Añadimos el repositorio de prosody con deb [signed-by=/usr/share/keyrings/prosody-keyring.gpg] <http://packages.prosody.im/debian> jammy pegándolo en el archivo.

```

GNU nano 6.2 /etc/apt/sources.list.d/prosody.list *
deb [signed-by=/usr/share/keyrings/prosody-keyring.gpg] http://packages.prosody.im/debian jammy

```

Eliminamos las claves gpg de home con sudo rm jitsi-key.gpg.key prosody-debian-packages.key

Actualizamos los repositorios y instalamos jitsi con sudo apt update && sudo apt install jitsi-meet

```
0 upgraded, 132 newly installed, 0 to remove and 0 not upgraded.
Need to get 194 MB of archives.
After this operation, 558 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Vemos que se van a **instalar 132 paquetes**.

Vemos que se necesitan **descargar 194 MB de archivos**.

Vemos que se usará **558 MB de disco**.

Presionamos **Y** para **continuar**.

Ponemos el dominio que en este caso es proyectopgg.net para configurar y instalar jitsi

Configuring jitsi-videobridge2

The value of the domain that is set in the Jitsi Videobridge installation.

The domain of the current installation (e.g. meet.jitsi.com):

proyectopgg.net

<Ok>

Presionamos el tabulador y presionamos enter en OK.

Configuring jitsi-meet-web-config

Jitsi Meet requires an SSL certificate. This installer can generate one automatically for your using "Let's Encrypt". This is the recommended and simplest option for most installations.

In the event you need to use a certificate of your own, you can configure its location which defaults to /etc/ssl/--domain.name--.key for the key and /etc/ssl/--domain.name--.crt for the certificate.

If you are a developer and are only looking for a quick way to test basic Jitsi Meet functionality then this installer can also generate a self-signed certificate.

SSL certificate

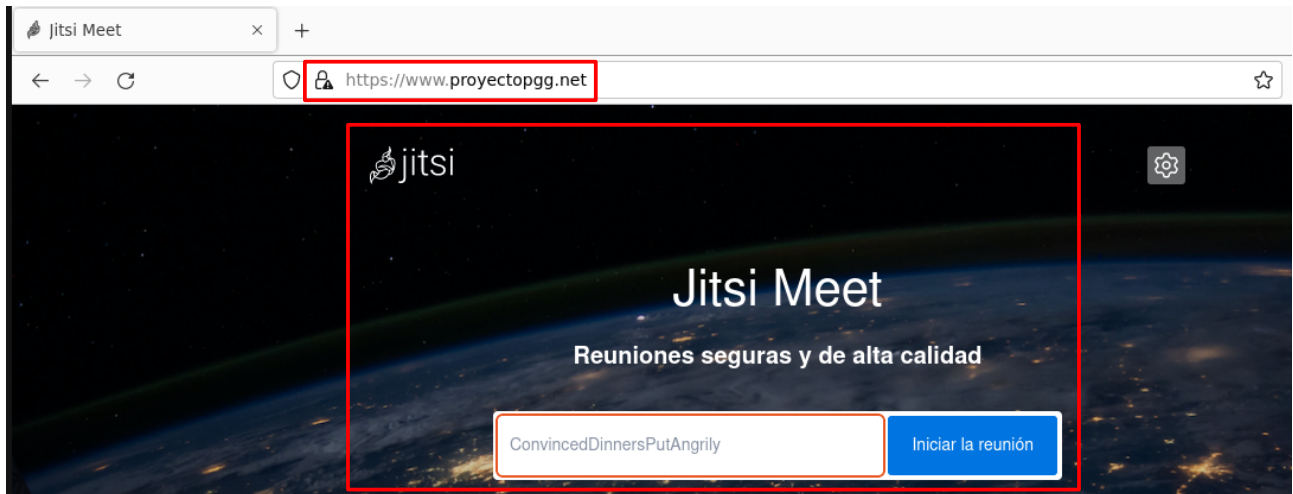
Let's Encrypt certificates
I want to use my own certificate
Generate a new self-signed certificate

<Ok>

Seleccionamos Generate a New self-signed certificate para instalar jitsi y nos genere un certificado.

Reiniciamos apache2 para terminar con sudo systemctl restart apache2

Vemos que al entrar a nuestro dominio <https://www.proyectopgg.net> nos entra a nuestro jitsi.



2.2.12 Instalamos y creamos las zonas el servidor DNS

Ponemos `sudo apt install bind9` para instalar el servidor DNS.

Vemos que se instalarán 3 paquetes nuevos y presionamos **Y** para continuar.

Para crear las zonas editamos el archivo `/etc/bind/named.conf.local` con `sudo nano /etc/bind/named.conf.local`

```
root@proyectopgg:~# sudo nano /etc/bind/named.conf.local
```

Las zonas empiezan por **zone**.

En las zonas directas ponemos el nombre de dominio con su sufijo. Abrimos un corchete y ponemos en **type master**, que es tipo maestro en español, ya que es el servidor primario DNS, ponemos punto y coma, ponemos en **file** la ruta donde se va a guardar la zona directa del servidor DNS, que está entre comillas y suele estar en `/etc/bind/zones` y el nombre de archivo suele ser **db punto nombre de dominio con su sufijo** termina en punto y coma y cerramos corchetes y ponemos punto y coma.

En la zona inversa ponemos de nombre la ip de la red invertida, luego punto in-addr y luego punto arpa. En el nombre del archivo ponemos **db punto ip de la red**.

```
GNU nano 6.2 /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 "proyectopgg.net" {
    type master;
    file "/etc/bind/zones/db.proyectopgg.net";
};

zone "56.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/db.192.168.56";
};
```

2.2.13 Creamos la carpeta de zonas, copiamos el archivo de la zona directa a las carpeta de zonas y creamos la zona directa.

La carpeta de zonas se crea con `sudo mkdir /etc/bind/zones`

```
root@proyectopgg:~# sudo mkdir /etc/bind/zones
```

Copiamos la zona directa de ejemplo a la carpeta zones con el nombre de archivo que le hemos puesto en el archivo de configuración de zonas con el siguiente comando: `sudo cp /etc/bind/db.local /etc/bind/zones/db.proyectopgg.net`

Editamos el archivo de nuestra zona directa con el siguiente comando: `sudo nano /etc/bind/zones/db.proyectopgg.net`

```
GNU nano 6.2 /etc/bind/zones/db.proyectopgg.net
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      proyectopgg.net. admin.proyectopgg.net. (
                        202305101      ; Serial
                        604800          ; Refresh
                        86400           ; Retry
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL
;
@         IN      NS       ns1.
@         IN      A        192.168.56.4
ns1       IN      A        192.168.56.4
server    IN      CNAME    ns1
www       IN      CNAME    ns1
lubuntu   IN      A        192.168.56.150
```

Arriba ponemos en la dirección del DNS en este caso `proyectopgg.net.` `admin.proyectopgg.net`

En **Serial** ponemos en este caso `202305101`, que sería 10/05/2023 y primera versión del archivo.

Abajo en la primera línea ponemos `@`, luego `IN`, luego `NS` y ponemos `ns1` punto, que es el nombre del servidor DNS, en este caso `ns1`.

Después en la segunda línea ponemos `@`, luego `IN`, luego `A`, `A` es para poner una dirección IP, en este caso la de nuestro servidor DNS que es `192.168.56.4`

Después en la tercera línea ponemos `ns1` que nuestro servidor DNS, luego `IN` y luego `A`, ponemos la dirección IP de nuestro servidor DNS que es `192.168.56.4`, este campo sirve para asociar el nombre del servidor DNS a una dirección IP.

En la cuarta línea ponemos `server`, que es un alias para nuestro servidor DNS, luego `IN` y luego `CNAME`, que es para poner un alias a nuestro servidor DNS que es `ns1`.

En la quinta línea ponemos `www`, que es un alias para nuestro servidor DNS, luego `IN` y luego `CNAME`, que es para poner un alias a nuestro servidor DNS que es `ns1`.

En la **sexta línea** ponemos **lubuntu** que es **nuestro cliente**, luego **IN** y luego **A**, ponemos la **dirección IP** de nuestro cliente que es **192.168.56.150**, este campo **sirve para asociar el nombre del cliente a una dirección IP**.

2.2.14 Copiamos el archivo de la zona inversa a la carpeta de zonas y creamos la zona inversa

Copiamos la zona inversa de ejemplo a la carpeta **zones** con el nombre de archivo que le hemos puesto en el archivo de configuración de zonas con el siguiente comando: **sudo cp /etc/bind/db.127 /etc/bind/zones/db.192.168.56**

Editamos el archivo de nuestra zona inversa con el siguiente comando: **sudo nano /etc/bind/zones/db.192.168.56**

```
GNU nano 6.2 /etc/bind/zones/db.192.168.56
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA     proyectopgg.net. admin.proyectopgg.net. (
202305111 ; Serial
604800    ; Refresh
86400     ; Retry
2419200   ; Expire
604800 )   ; Negative Cache TTL
;
@         IN      NS      ns1.
4         IN      PTR     ns1.proyectopgg.net.
```

Arriba ponemos en la dirección del DNS en este caso **proyectopgg.net. admin.proyectopgg.net.**

En **Serial** ponemos en este caso **202305111**, que sería **11/05/2023** y primera versión del archivo.

Abajo en la primera línea ponemos **@**, luego **IN**, luego **NS** y ponemos **ns1** punto, que es el nombre del servidor DNS, en este caso **ns1**.

Debajo ponemos **4**, que es el último número de los cuatro de nuestra dirección IP que va a identificar al servidor DNS, ponemos **IN**, ponemos **PTR** que es para asociar nuestra dirección IP a nuestro subdominio que es **ns1**, nuestro dominio que es **proyectopgg** y nuestro sufijo que es **punto net**, esta dirección es la que saldrá al hacer ping a la IP.

2.2.15 Configuramos el servidor DHCP y el Proxmox para el bind9

Editamos el servidor DHCP para poner de DNS nuestro servidor DNS

```
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;
```

Ponemos una # a option domain-name y a option domain-name-servers ya que son de ejemplo.

```
# option definitions common to all supported networks...
# option domain-name "example.org";
# option domain-name-servers ns1.example.org, ns2.example.org;
```

Borramos los dos DNS en option domain-name-servers y ponemos nuestro DNS que es el subdominio ns1, dominio proyectopgg y sufijo punto net

```
subnet 192.168.56.0 netmask 255.255.255.0 {
    range 192.168.56.150 192.168.56.200;
    option routers 192.168.56.4;
    option domain-name-servers ns1.proyectopgg.net;
}
```

Editamos el archivo named.conf.options para poder acceder a páginas web desde nuestro servidor DNS

Lo editamos con el siguiente comando `sudo nano /etc/bind/named.conf.options`

En forwarders quitamos las barras a la derecha y el espacio en rojo para poder meter los dos DNS.

```
GNU nano 6.2 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers
    // to talk to, you may need to fix the firewall to allow
    // ports to talk.  See http://www.kb.cert.org/vuls/id/8001
    // on your firewall to allow DNS (ports 53) to pass.

    // If your ISP provided one or more IP addresses for
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses
    // the all-0's placeholder.

    // forwarders {
    // 0.0.0.0;
    // };
```

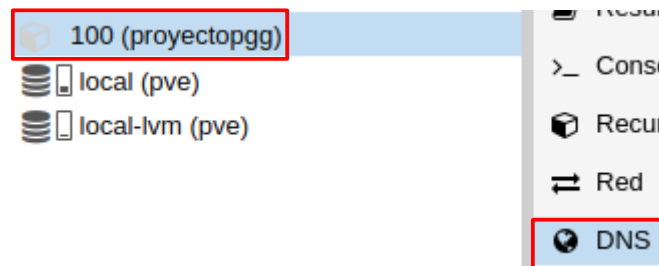
Ponemos entre los corchetes de forwarders, los dos DNS terminados en punto y coma, y uno debajo del otro.

```
forwarders {
    208.67.222.222;
    208.67.220.220;
};
```

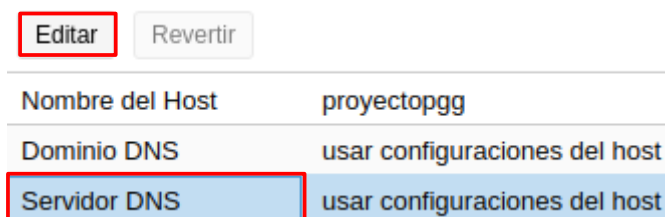
Editamos nuestro servidor DNS en nuestro contenedor de Proxmox.

Apagamos nuestro contenedor con poweroff.

Hacemos clic en nuestro contenedor apagado y hacemos clic en DNS.



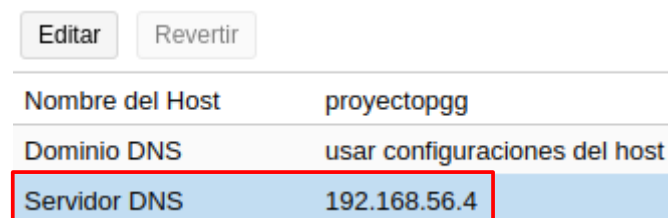
Hacemos clic en Servidor DNS y hacemos clic en Editar.



En Editar: Recursos ponemos en Servidores DNS la IP de nuestro servidor DNS que es 192.168.56.4 . Y hacemos clic en Aceptar para guardar los cambios.



Vemos en Servidor DNS la nueva dirección IP.



2.2.16 Vemos los resultados

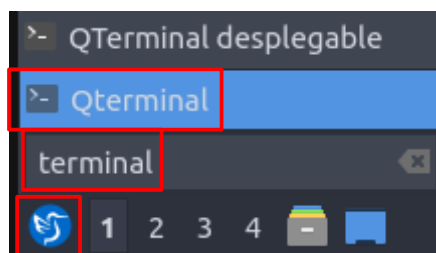
Encendemos el cliente Lubuntu y vemos que en DNS(1) pone la dirección IP de nuestro servidor DNS.

```

General
Interfaz:      Ethernet (enp0s3)
Dirección hardware: 08:00:27:51:C7:33
Controlador:    e1000
Velocidad:      1000000 Kb/s

IPv4
Dirección IP:    192.168.56.150
Máscara de subred: 255.255.255.0
Ruta por defecto: 192.168.56.4
DNS(1):          192.168.56.4
  
```

Hacemos clic en el icono del menú de inicio, buscamos terminal, y hacemos clic en Qterminal.



Hacemos ping desde Lubuntu a nuestro DNS ns1.proyectopgg.net con ping ns1.proyectopgg.net .

```

usuario@cliente:red:~$ ping ns1.proyectopgg.net
PING ns1.proyectopgg.net (192.168.56.4) 56(84) bytes of data.
64 bytes from ns1.proyectopgg.net (192.168.56.4): icmp_seq=1 ttl=64 time=0.338 ms
64 bytes from ns1.proyectopgg.net (192.168.56.4): icmp_seq=2 ttl=64 time=0.592 ms
64 bytes from ns1.proyectopgg.net (192.168.56.4): icmp_seq=3 ttl=64 time=0.612 ms
^C
--- ns1.proyectopgg.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2012ms
rtt min/avg/max/mdev = 0.338/0.514/0.612/0.124 ms
  
```

Vemos que ns1.proyectopgg.net es nuestra dirección IP que hemos puesto al configurar el servidor DNS.

Vemos que da en time menos de 1ms porque funciona correctamente nuestro servidor DNS porque está alojado en nuestra red.

Vemos 3 paquetes transmitidos, 3 paquetes recibidos y 0% de paquetes perdidos.

Hacemos **ping** desde **Proxmox** a nuestro **lubuntu** **lubuntu.proyectopgg.net** con **ping** **lubuntu.proyectopgg.net** .

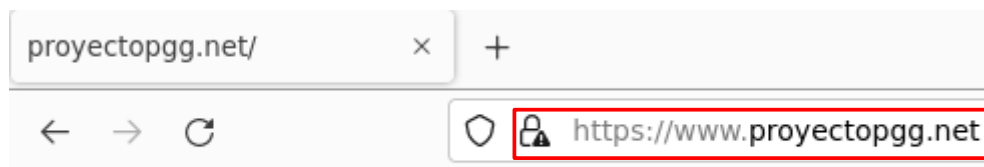
```
root@proyectopgg:~# ping lubuntu.proyectopgg.net
PING lubuntu.proyectopgg.net (192.168.56.150) 56(84) bytes of data:
64 bytes from 192.168.56.150 (192.168.56.150): icmp_seq=1 ttl=64 time=0.305 ms
64 bytes from 192.168.56.150 (192.168.56.150): icmp_seq=2 ttl=64 time=0.496 ms
64 bytes from 192.168.56.150 (192.168.56.150): icmp_seq=3 ttl=64 time=0.626 ms
^C
--- lubuntu.proyectopgg.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.305/0.475/0.626/0.131 ms
```

Vemos que **lubuntu.proyectopgg.net** es la dirección IP de **Lubuntu** que hemos puesto al configurar el servidor DNS.

Vemos que da en time menos de 1ms porque funciona correctamente nuestro servidor DNS, porque está alojado en nuestra red el **Lubuntu** y está encendido.

Vemos 3 paquetes transmitidos, 3 paquetes recibidos y 0% de paquetes perdidos.

Vemos nuestra página web al entrar desde el navegador en **Lubuntu** poniendo **www.proyectopgg.net** .



Prueba apache por Pablo G

2.2.17 Asignamos una IP fija mediante el DHCP

Copiamos la dirección del hardware o dirección MAC que hemos visto antes, que en este caso es **08:00:27:51:c7:33**.

Editamos otra vez el archivo de configuración del DHCP.

```
subnet 192.168.56.0 netmask 255.255.255.0 {
    range 192.168.56.151 192.168.56.200;
    option routers 192.168.56.4;
    option domain-name-servers ns1.proyectopgg.net;
    host lubuntu.proyectopgg.net {
        hardware ethernet 08:00:27:51:c7:33;
        fixed-address 192.168.56.150;
    }
}
```

Cambiamos el range para que empiece por la **192.168.56.151**, porque la dirección **192.168.56.150** se va a asignar a nuestro **lubuntu**.

Después de **option domain-name-servers** ponemos **host lubuntu.proyectopgg.net** que es nuestro **lubuntu** en nuestra zona DNS **proyectopgg.net** .

Abrimos corchetes y ponemos **hardware ethernet** nuestra dirección del hardware o dirección MAC que hemos visto antes, que en este caso es **08:00:27:51:c7:33** .

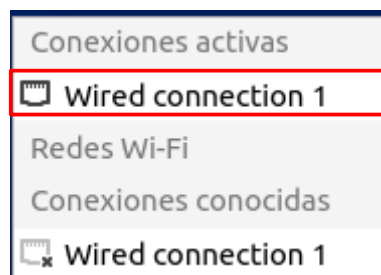
Y ponemos en **fixed-address** la dirección IP que tenemos puesta en el DNS para el Lubuntu, que en este caso es **192.168.56.150**.
Cerramos corchetes y guardamos los cambios.

Reiniciamos bind9 con `sudo systemctl restart bind9`.

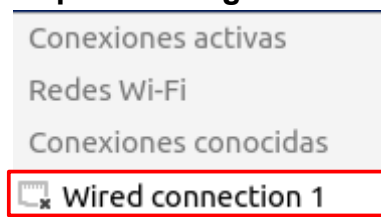
Reiniciamos isc-dhcp-server con `sudo systemctl restart isc-dhcp-server`.

Nos desconectamos y nos volveremos a conectarnos de la red en Lubuntu como hemos visto antes

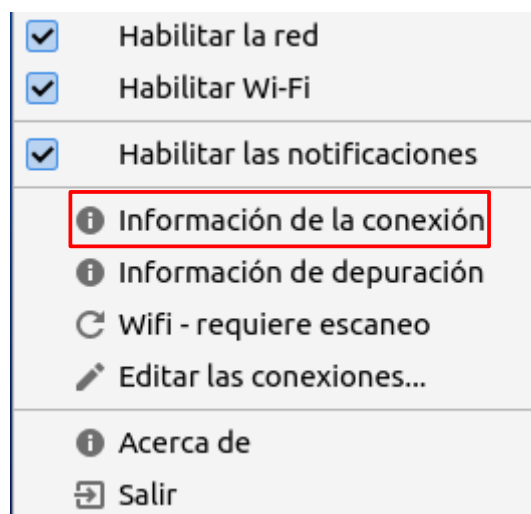
Hacemos clic en el icono de Ethernet y en **Conexiones activas** hacemos clic en **Wired connection 1**.



Hacemos otra vez clic en el icono de Ethernet y en **Conexiones conocidas** hacemos clic en **Wired connection 1** para que nos asigne la dirección IP el DHCP.



Hacemos otra vez clic derecho en el icono de Ethernet y seleccionamos **Información de la conexión**.



En Dirección IP, vemos que es la **dirección IP fija** que **hemos asignado desde el DHCP**.



Vemos que al hacer ping a nuestro **Lubuntu** como está en la zona DNS **proyectopgg.net** se llama **lubuntu.proyectopgg.net**

```
root@proyectopgg:~# ping lubuntu.proyectopgg.net
PING lubuntu.proyectopgg.net (192.168.56.150) 56(84) bytes of data.
64 bytes from 192.168.56.150 (192.168.56.150): icmp_seq=1 ttl=64 time=0.341 ms
```

Vemos que nos sale la **dirección IP** que le hemos asignado y responde en menos de 1 ms, por lo que funciona correctamente.

2.2.18 Crear el contenedor para el servidor FTP y configuraciones iniciales

Hacemos clic arriba a la derecha en Crear CT para crear nuestro contenedor para el servidor FTP y VPN.



En **Nodo**, dejamos **pve** porque **solo tenemos un nodo**, en **CT ID**, ponemos el **ID** de **nuestro contenedor**, que en este caso **es 101** porque **es el segundo que hemos creado**, el **nombre de host** ponemos el **nombre** que se **identificará** nuestro contenedor, **desmarcamos** la **casilla Contenedores sin privilegios**, no **ponemos ningún Conjunto de Recursos**, porque **solo lo va a usar el administrador**, **ponemos** nuestra **contraseña** de **administrador** en el **contenedor** y la **confirmamos** en **Confirmar contraseña** y hacemos **clic** en **Siguiente** para ir al **siguiente paso**.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Nodo:

pve

CT ID:

101

Nombre del Host:

proyectopgg2

Contenedores sin privilegios:

☐

Nesting:

☒

Conjunto de Recursos:

Contraseña:

.....

Confirmar contraseña:

.....

Clave pública SSH:

Carga archivo de clave SSH

Ayuda

Avanzado ☐

Atrás

Siguiente

En **Plantilla**, en **Almacenamiento** dejamos **local** ya que **solo tenemos** un **disco de almacenamiento** y en **Plantilla seleccionamos** nuestra **plantilla de contenedor** que hemos **cargado antes** al **proxmox**. Hacemos **clic** en **Siguiente** para continuar.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Almacenamiento:

local

Plantilla:

ubuntu-22.04-standard_22.04-1_a

Ayuda

Avanzado ☐

Atrás

Siguiente

En **Discos** en **Almacenamiento** dejamos el que está para **almacenar** los **datos** del **contenedor** ya que **solo tenemos uno**.

En **Tamaño de disco (GiB)** ponemos **16 GiB**, ya que es el **espacio** que **necesitaremos** en **nuestro contenedor**.

Hacemos **clic** en **Siguiente** para **continuar**.

The screenshot shows the 'Crear: Contenedor LXC' window with the 'Discos' tab selected. The 'rootfs' disk is listed with 'Almacenamiento' set to 'local-lvm'. The 'Tamaño de disco (GiB)' is set to '16'. The 'Siguiente' button is highlighted with a red box.

Crear: Contenedor LXC

General Plantilla **Discos** CPU Memoria Red DNS Confirmar

rootfs Almacenamiento: local-lvm

Tamaño de disco (GiB): 16

+ Agregar

? Ayuda Avanzado ☐ Atrás **Siguiente**

En **CPU** dejamos **1 núcleo** ya que **tenemos más que suficiente** para **nuestros servicios** y hacemos **click** en **Siguiente**.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Núcleos:

1

Ayuda

Avanzado ☐

Atrás

Siguiente

En **Memoria (MiB)** ponemos **1536 MiB**, ya que es **más que suficiente** para nuestras tareas y en **Swap (MiB)** ponemos lo mismo también.

Hacemos clic en **Siguiente** para continuar.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Memoria (MiB):

1536

Swap (MiB):

1536

Ayuda

Avanzado ☐

Atrás

Siguiente

En **Red**, dejamos el **Nombre** como está porque **solo vamos a tener una tarjeta de red**, en **Puente** seleccionamos nuestro **adaptador de red** que da **conexión a internet**, en este caso es **vmbr0**, **desmarcamos** la **casilla de Cortafuego**, en **IPv4** dejamos **Estático**, ya que lo **necesitaremos** así para **nuestros servicios FTP, VPN**, en **IPv4/CIDR** ponemos **nuestra dirección IP/nuestra máscara de subred** como **número decimal**, ponemos **192.168.56.5/24**, porque es nuestro **segundo contenedor**, en **Puerta de enlace (IPv4)** ponemos la **dirección de nuestro router** para **tener internet**.

Si **nuestra operadora tiene IPv6** tocamos también **IPv6**, de lo **contrario** hacemos clic en **Siguiente**.

Crear: Contenedor LXC

General Plantilla Discos CPU Memoria **Red** DNS Confirmar

Nombre: eth0

Dirección MAC: auto

Puente: vmbr0

Etiqueta VLAN: no VLAN

Cortafuego: ☐

IPv4: ☒ Estático ☐ DHCP

IPv4/CIDR: 192.168.56.5/24

Puerta de enlace (IPv4): 192.168.56.1

IPv6: ☐ Estático ☐ DHCP ☐ SLAAC

IPv6/CIDR: None

Puerta de enlace (IPv6):

Ayuda Avanzado ☐ Atrás **Siguiente**

En **DNS**, en **Dominio DNS** usamos las **configuraciones** del **host** y en **Servidores DNS** ponemos la **dirección IP** del **primer contenedor** que es **192.168.56.4**, que es el que **tiene el servidor DNS** y hacemos clic en **Siguiente**.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Dominio DNS:

usar configuraciones del host

Servidores DNS:

192.168.56.4

Avanzado ☐

Atrás

Siguiente

En Confirmar vemos los valores que hemos puesto, podemos **marcar la casilla Start after created** para **iniciar el contenedor una vez creado** y hacemos **clic en Finalizar**.

Crear: Contenedor LXC

General

Plantilla

Discos

CPU

Memoria

Red

DNS

Confirmar

Key ↑	Value
cores	1
hostname	proyectopgg2
memory	1536
nameserver	192.168.56.4
net0	name=eth0,bridge=vmbr0,ip=192.168.56.5/24,gw=192.168.56.1
nodename	pve
ostemplate	local:vztmpl/ubuntu-22.04-standard_22.04-1_amd64.tar.zst
pool	
rootfs	local-lvm:16
swap	1536
vmid	101

☒ Start after created

Avanzado ☐

Atrás

Finalizar

Cuándo ponga TASK OK cerramos la ventana.

Task viewer: CT 101 - Crear

Salida Estado

Parar Descargar

```

WARNING: You have not turned on protection against thin pools running out of space.
WARNING: Set activation/thin_pool_autoextend_threshold below 100 to trigger automatic extension of thin pools before they get full.
Logical volume "vm-101-disk-0" created.
WARNING: Sum of all thin volume sizes (32.00 GiB) exceeds the size of thin pool pve/data and the amount of free space in volume group (<7.88 GiB).
Creating filesystem with 4194304 4k blocks and 1048576 inodes
Filesystem UUID: f7abd0c4-cf0d-48f6-8a8e-b34807e39626
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
extracting archive '/var/lib/vz/template/cache/ubuntu-22.04-standard_22.04-1_amd64.tar.zst'
Total bytes read: 508579840 (486MiB, 75MiB/s)
Detected container architecture: amd64
Creating SSH host key 'ssh_host_dsa_key' - this may take some time ...
done: SHA256:GqtzRR4aukID+M21ofNbMOeg9Mvvr3ka2KzT+YmHU88 root@proyectopgg2
Creating SSH host key 'ssh_host_ed25519_key' - this may take some time ...
done: SHA256:JUQjduRAZnUfYnDnCXkYJE88kE4plQ5TepOafKPoc root@proyectopgg2
Creating SSH host key 'ssh_host_ecdsa_key' - this may take some time ...
done: SHA256:GukCzLGJ1ay1XoCwJ7BgRgQ8rOUVSG60icAbansw4Lw root@proyectopgg2
Creating SSH host key 'ssh_host_rsa_key' - this may take some time ...
done: SHA256:PqrNTSgevtMrVYBHLZzj6wmTtTmt3EgPpXLhwo4g/Fo root@proyectopgg2
TASK OK

```

Para **utilizar el contenedor**, hacemos **clic derecho** en el **contenedor** que **acabamos de crear** y hacemos clic en **Consola**.



Ponemos nuestro nombre de usuario que es root en nombre de máquina login y ponemos nuestra contraseña para entrar en Password.

```

Ubuntu 22.04 LTS proyectopgg2 tty1
proyectopgg2 login: root
Password:

```

Vemos que ya podemos empezar a instalar los servicios.

```
root@proyectopgg2:~#
```

Configuraciones Iniciales del contenedor

Ponemos **sudo apt update && sudo apt upgrade && sudo apt dist-upgrade** para cargar los paquetes que hay que actualizar de los repositorios, actualizar los paquetes y actualizar la distribución.

Vemos que **117** paquetes van a ser actualizados.

```
117 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
54 standard security updates
Need to get 50.6 MB of archives.
After this operation, 1044 kB disk space will be freed.
Do you want to continue? [Y/n] Y
```

Vemos que un paquete nuevo va a ser instalado.

Presionamos **Y** para continuar.

Ponemos **reboot** para reiniciar nuestro contenedor.

Vemos nuestra fecha y hora con **date**.

```
root@proyectopgg2:~# date
Sun May 28 17:24:26 UTC 2023
```

Vemos que nuestra hora está mal.

Vemos los husos horarios de Ubuntu con **timedatectl list-timezones**

Vemos que está **Europe/Madrid**.

Presionamos **q** para salir.

Para cambiar la zona horaria a **Europe/Madrid**, ponemos el siguiente comando:

sudo timedatectl set-timezone Europe/Madrid

Vemos que el poner **date** otra vez ya tenemos la hora correcta.

```
root@proyectopgg2:~# date
Sun May 28 19:26:50 CEST 2023
```

Reiniciamos otra vez el contenedor.

2.2.19 Instalar servidor FTP y configurarlo

Instalamos el servidor ftp con `sudo apt install vsftpd libpam-pwdfile`

Hacemos una copia de seguridad del archivo de configuración con `sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bck`

Borramos el archivo de configuración antiguo con `sudo rm -r /etc/vsftpd.conf`

Editamos el archivo de configuración nuevo con `sudo nano /etc/vsftpd.conf`

```
listen=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
local_root=/var/www
chroot_local_user=NO
allow_writeable_chroot=YES
hide_ids=YES

# virtual user settings
user_config_dir=/etc/vsftpd_user_conf
guest_enable=YES
virtual_use_local_privs=YES
pam_service_name=vsftpd
nopriv_user=vsftpd
guest_username=vsftpd
```

`listen=Yes` sirve para que se puedan conectar los equipos, si no estuviese la directiva no se pueden conectar los equipos.

`anonymous_enable=NO` sirve para que no se puedan conectar sin usuario ni contraseña.

`local_enable=YES` sirve para que se puedan conectar los usuarios al servidor FTP.

`write_enable=YES` sirve para que los usuarios puedan escribir en el servidor FTP.

`local_umask=022` para que no se pueda escribir en la carpeta raíz por defecto de la página web y solo pueda escribir en la subcarpeta creada para ello.

`local_root=/var/www` carpeta raíz del servidor ftp.

`chroot_local_user=NO` sirve para que no pueda ver las carpetas de otros usuarios.

`allow_writeable_chroot=YES` sirve para que podamos escribir en las carpetas.

`hide_ids=YES` ocultar información acerca del usuario y grupo propietario de los archivos.

`# virtual user settings` ajustes de los usuarios virtuales de vsftpd.

`user_config_dir=/etc/vsftpd_user_conf` directorio de configuración de los usuarios.

`guest_enable=YES` activar el usuario invitado.

`virtual_use_local_privs=YES` los usuarios virtuales tienen los mismos privilegios que los usuarios locales.

`pam_service_name=vsftpd` esta cadena es el nombre del servicio PAM que usará `vsftpd`.

`nopriv_user=vsftpd` este es el nombre del usuario que usa `vsftpd` cuando quiere estar totalmente privado de privilegios. Tenga en cuenta que este debe ser un usuario dedicado, en lugar de `nobody`. El usuario `nobody` tiende a usarse para muchas cosas importantes en la mayoría de las máquinas.

`guest_username=vsftpd` nombre del usuario invitado de `vsftpd`.

2.2.20 Instalamos apache y creamos los usuarios virtuales de vsftpd

Creamos el directorio donde se guardan los usuarios con `sudo mkdir /etc/vsftpd`

Instalamos `apache2` con `sudo apt install apache2`

```
0 upgraded, 22 newly installed, 0 to remove and 0 not upgraded.
Need to get 11.2 MB of archives.
After this operation, 59.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Vemos que 22 paquetes se van a instalar y presionamos Y para continuar.

Creamos el usuario `pggftp` con `sudo htpasswd -c -p -b /etc/vsftpd/ftpd.passwd pggftp $(openssl passwd -1 -noverify password)`

La contraseña del usuario será `password`.

```
Adding password for user pggftp
```

Vemos que se ha añadido una contraseña al primer usuario.

Vemos la contraseña encriptada del primer usuario con `sudo cat /etc/vsftpd/ftpd.passwd`

```
pggftp:$1$/stvgnkd$/yrXHMS6zNPN4cOD3pxc51
```

Creamos el usuario `pggftp2` con `sudo htpasswd -p -b /etc/vsftpd/ftpd.passwd pggftp2 $(openssl passwd -1 -noverify password)`

Quitamos el `-c` para que no se sobrescriba el archivo y se borre el primer usuario.

La contraseña del usuario será `password`.

```
Adding password for user pggftp2
```

Vemos que se ha añadido una contraseña al segundo usuario.

Vemos la contraseña encriptada del primer y segundo usuario con `sudo cat /etc/vsftpd/ftpd.passwd`

```
pggftp:$1$/stvgnkd$/yrXHMS6zNPN4cOD3pxc51
pggftp2:$1$KLPTc3ne$vxDYMHha8KRGOCGBm.175.
```

Archivo de autenticación PAM

Copiamos el archivo antiguo para no perderlo al crear el nuevo y recuperarlo en caso de necesidad con `sudo cp /etc/pam.d/vsftpd /etc/pam.d/vsftpd.bck`

Borramos el archivo antiguo con `sudo rm -r /etc/pam.d/vsftpd`

Creamos el archivo nuevo de autenticación con `sudo nano /etc/pam.d/vsftpd`

`auth required pam_pwdfile.so pwdfile /etc/vsftpd/ftpd.passwd`

`account required pam_permit.so`

`auth required` es la librería que comprueba que el usuario y la contraseña son correctos.

`pwdfile` es nuestro archivo donde están las contraseñas encriptadas.

`account required` es para permitir el acceso al servidor ftp si el usuario y las contraseñas son correctos.

2.2.21 Crear usuario vsftpd y archivo de configuración de los usuarios

Creamos el usuario vsftpd para el servidor ftp con:

`sudo useradd --home /home/vsftpd --gid nogroup -m --shell /bin/false vsftpd`

`/home/vsftpd` es la carpeta donde se guardarán los datos del usuario.

`--gid nogroup` es que no tendrá grupo el usuario.

`-m` es para crear la carpeta donde se guardarán los datos del usuario y se copiarán los archivos necesarios para que el usuario funcione.

`--shell /bin/false` es para que no se pueda iniciar sesión con la terminal porque no tiene entorno de trabajo.

`vsftpd` es el nombre del usuario.

Creamos el archivo de configuración del usuario para cada uno solo pueda escribir en su carpeta

Creamos la carpeta de configuración de los usuarios con `sudo mkdir /etc/vsftpd_user_conf`

Creamos el archivo de configuración del primer usuario con `sudo nano /etc/vsftpd_user_conf/pggftp`

Ponemos `local_root=/var/www/html/pggftp` en el archivo.

```
GNU nano 6.2 /etc/vsftpd_user_conf/pggftp
local_root=/var/www/html/pggftp
```

`local_root` es la carpeta que vamos a crear en el siguiente paso.

Creamos la carpeta del primer usuario con `sudo mkdir /var/www/html/pggftp`

Creamos el archivo de configuración del segundo usuario con `sudo nano /etc/vsftpd_user_conf/pggftp2`

Ponemos `local_root=/var/www/html/pggftp2` en el archivo.

```
GNU nano 6.2 /etc/vsftpd_user_conf/pggftp2
local_root=/var/www/html/pggftp2
```

Creamos la carpeta del segundo usuario con `sudo mkdir /var/www/html/pggftp2`

Cambiamos el propietario de las dos carpetas creadas a `vsftpd`

Primero cambiamos la del primer usuario con `sudo chown vsftpd:nogroup /var/www/html/pggftp`

Después cambiamos la del segundo usuario con `sudo chown vsftpd:nogroup /var/www/html/pggftp2`

Reiniciamos el servicio para aplicar los cambios con `sudo service vsftpd restart`

2.2.22 Crear certificado para vsftpd

Creamos el certificado con `sudo openssl req -x509 -days 365 -newkey rsa:2048 -nodes -keyout /etc/vsftpd.pem -out /etc/vsftpd.pem`

`-days` son los días que durará el certificado, por defecto son 365 días o un año.

`-newkey` es para crear una nueva clave.

`-keyout` es para exportar la clave.

`-out` es para exportar el archivo de autenticación.

```
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Andalucía
Locality Name (eg, city) []:Córdoba
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PROYECTOPGG
Organizational Unit Name (eg, section) []:PROYECTOPGG
Common Name (e.g. server FQDN or YOUR name) []:PROYECTOPGG
Email Address []:PROYECTOPGG@PROYECTOPGG.LOCAL
```

Ponemos en `Country Name` con dos letras el nombre del país, que en este caso es España con ES.

Ponemos en `State or Province Name (full name) [Some-State]` el nombre de nuestra comunidad autónoma que en este caso es Andalucía.

`Locality Name (eg, city)` ponemos el nombre de nuestra ciudad, en nuestro caso Córdoba.

En `Organization Name (eg, company)` ponemos el nombre de nuestra empresa, en este caso PROYECTOPGG.

En `Organizational Unit Name (eg, section)` ponemos el nombre de nuestra sección de la empresa, en nuestro caso PROYECTOPGG.

En `Common Name (e.g. server FQDN OR YOUR name)` ponemos nuestro nombre o el nombre de nuestro servidor en nuestro caso PROYECTOPGG.

En `Email Address` ponemos nuestra dirección de correo electrónico, en nuestro caso PROYECTOPGG@PROYECTOPGG.LOCAL

2.2.23 Terminar de configurar /etc/vsftpd.conf

Añadimos lo siguiente en el archivo /etc/vsftpd.conf

```
# enable TLS/SSL
ssl_enable=YES
```

Activamos el servidor ftp seguro con ssl_enable=YES.

```
# force client to use TLS when logging in
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

Esto sirve para forzar a usar TLS cuando se loguean los clientes.

allow_anon_ssl=NO Solo se aplica si **ssl_enable** está activo. Sirve para que los usuarios anónimos no puedan establecer conexiones SSL seguras.

force_local_data_ssl=YES Solo se aplica si **ssl_enable** está activado. Si está activado, todos los inicios de sesión no anónimos están obligados a utilizar una conexión SSL segura para enviar y recibir datos en las conexiones de datos.

force_local_logins_ssl=YES Solo se aplica si **ssl_enable** está activado. Si está activado, todos los inicios de sesión no anónimos están obligados a utilizar una conexión SSL segura para enviar la contraseña.

ssl_tlsv1=YES Es para utilizar el protocolo TLS que es más seguro.

ssl_sslv2=NO Es para no utilizar el protocolo SSL versión 2 que es menos seguro.

ssl_sslv3=NO Es para no utilizar el protocolo SSL versión 3 que es menos seguro.

require_ssl_reuse=NO Si se establece en sí, se requiere que todas las conexiones de datos SSL exhiban la reutilización de la sesión SSL (lo que prueba que conocen el mismo secreto maestro que el canal de control). Aunque este es un valor predeterminado seguro, puede dañar muchos clientes FTP, por lo que es posible que desee deshabilitarlo. Para una discusión de las consecuencias, vea <http://scarybeastsecurity.blogspot.com/2009/02/vsftpd-210-released.html> (Agregado en v2.1.0).

ssl_ciphers=HIGH Esta opción se puede utilizar para seleccionar qué cifrados SSL vsftpd permitirá conexiones SSL cifradas. Consulte la página del manual de cifrados para obtener más detalles. Tenga en cuenta que restringir los cifrados puede ser una precaución de seguridad útil, ya que evita que partes remotas malintencionadas fuercen un cifrado con el que han encontrado problemas.

```
# specify SSL certificate/private key (Debian/Ubuntu)
# For CentOS/Fedora/RHEL, replace it with /etc/ssh/vsftpd.pem
rsa_cert_file=/etc/ssh/vsftpd.pem
rsa_private_key_file=/etc/ssh/vsftpd.pem
```

rsa_cert_file Es el archivo donde está el certificado.

rsa_private_key Es el archivo donde está la clave privada.

```
# define port range for passive mode connections
pasv_max_port=65535
pasv_min_port=64000
```

Esto define el puerto máximo en pasv_max_port de conexiones pasivas y el puerto mínimo en pasv_min_port de conexiones pasivas.

Los puertos pasivos solo se escuchan cuando se llaman.

Guardamos el archivo de configuración y reiniciamos el servidor vsftpd con sudo systemctl restart vsftpd

Desactivamos el firewall con sudo ufw disable

Reiniciamos el sistema.

2.2.24 Comprobar que funciona vsftpd en Lubuntu

Ponemos sudo apt update && sudo apt install filezilla para recargar los repositorios y instalar filezilla en Lubuntu.

```
[sudo] password for usuario:
```

Ponemos la contraseña del usuario para esto.

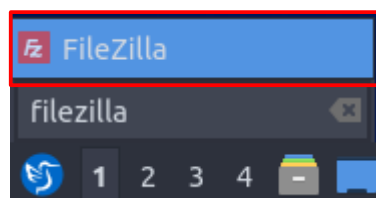
```
Se instalarán los siguientes paquetes adicionales:
filezilla-common libfilezilla-common libfilezilla24 libpugixml1v5 libwxbase3.0-0v5 libwxgtk3.0-gtk3-0v5
Se instalarán los siguientes paquetes NUEVOS:
filezilla filezilla-common libfilezilla-common libfilezilla24 libpugixml1v5 libwxbase3.0-0v5 libwxgtk3.0-gtk3-0v5
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 10,1 MB de archivos.
Se utilizarán 36,9 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Vemos los paquetes adicionales que se van a instalar y vemos también los paquetes nuevos que se van a instalar.

Se instalarán 7 paquetes nuevos.

Presionamos S para continuar.

Comprobamos que funciona FileZilla.



Abrimos el menú de inicio, buscamos FileZilla y hacemos clic en FileZilla.

Ponemos en Servidor 192.168.56.5

Ponemos en Nombre de usuario uno de los dos usuarios que hemos creado para loguearnos son pggftp y pggftp2.

En Contraseña ponemos password ya que es la contraseña que hemos creado para los dos usuarios.

Hacemos clic en Conexión rápida para conectarnos.

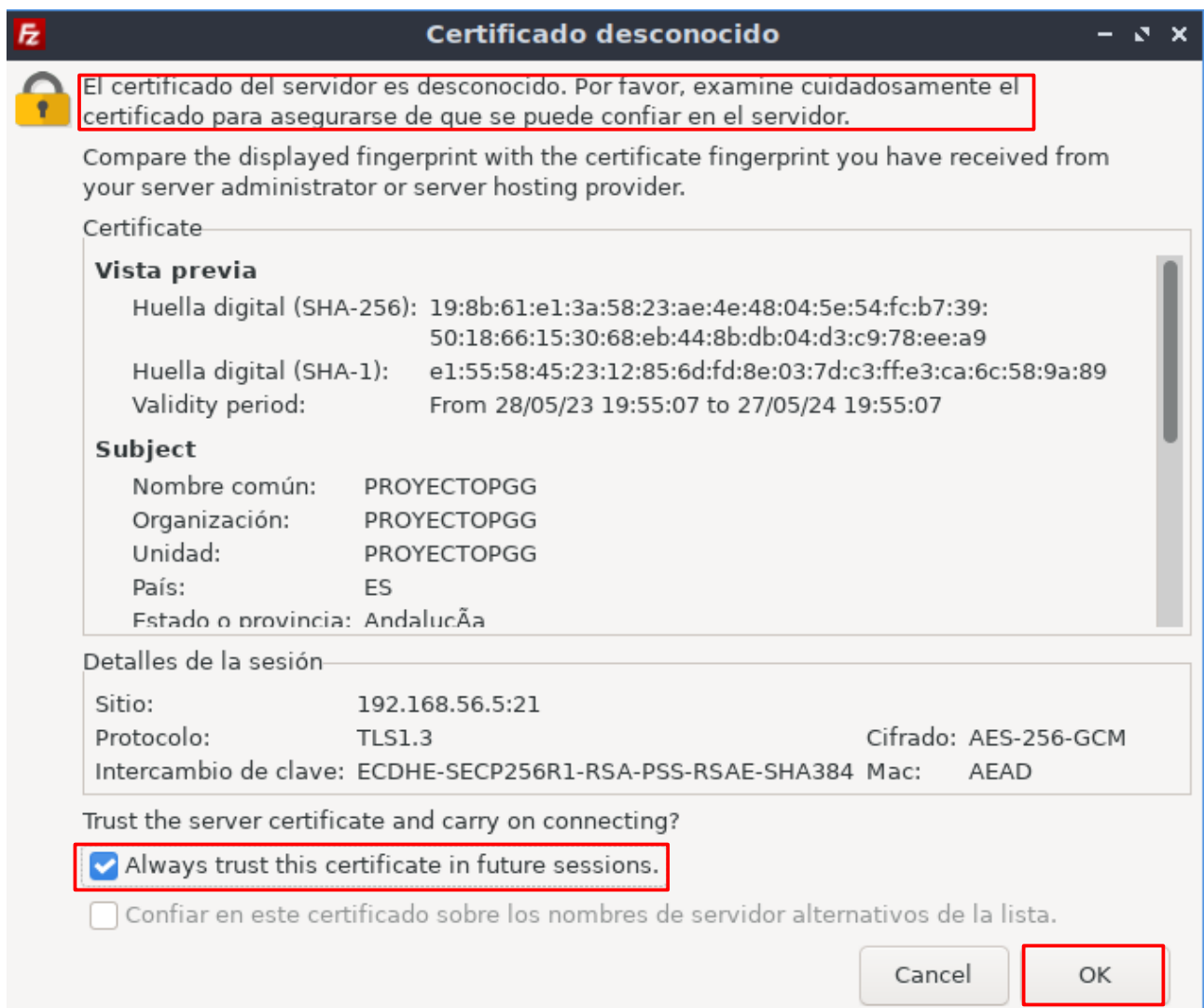
Servidor: 192.168.56.5 Nombre de usuario: pggftp Contraseña: ●●●●●● Puerto: **Conexión rápida** ▼

Nos sale que el certificado es desconocido. Por favor, examine cuidadosamente el certificado para asegurarse de que puede confiar en el servidor.

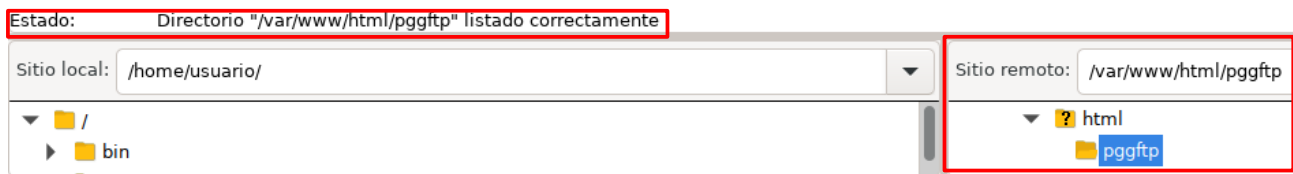
Nos pone si queremos ¿Confiar en el certificado y seguir con la conexión?

Hacemos clic en la casilla Always trust this certificate in future sessions.

Hacemos clic en OK para conectarnos.

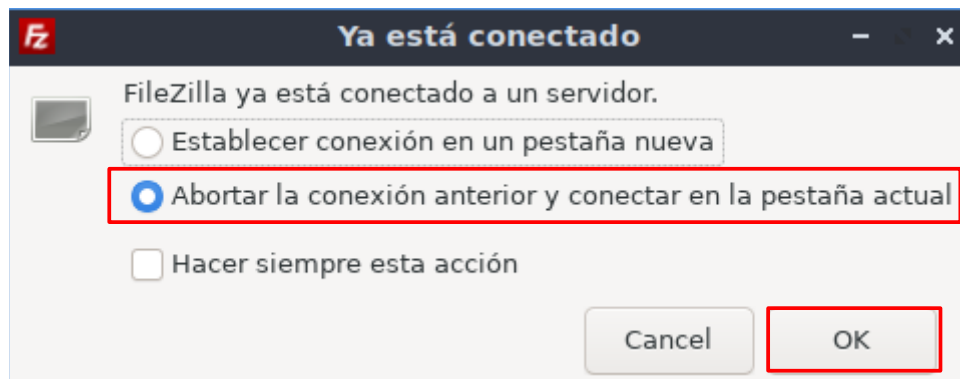
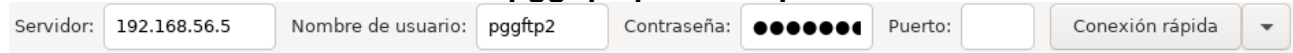


Vemos que nos sale correctamente la carpeta con nuestro usuario.



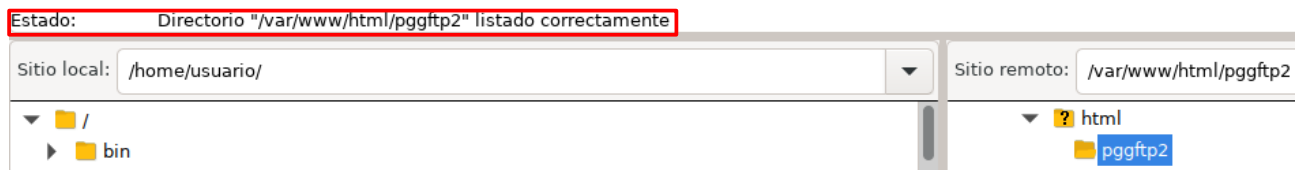
En sitio remoto es la carpeta de nuestro ftp, vemos que sale la ruta correctamente y funciona.

Ponemos en Nombre de usuario pggftp2 para ver que funciona.



Nos sale que Filezilla ya está conectado a un servidor.

Hacemos clic en **Abortar la conexión anterior y conectar en la pestaña actual** para conectarse en la actual pestaña y hacemos clic en **OK**.



Vemos que nos sale correctamente la carpeta con nuestro otro usuario.

En sitio remoto es la otra carpeta de nuestro ftp, vemos que sale la ruta correctamente y funciona.

2.2.25 Instalamos la entidad certificadora del servidor VPN y creamos la infraestructura de clave pública de la entidad certificadora

Instalamos la entidad certificadora con `sudo apt update && sudo apt install easy-rsa`

```
The following NEW packages will be installed:
  easy-rsa libccid libpcsclite1 opensc opensc-pkcs11 pcscl
0 upgraded, 6 newly installed, 0 to remove and 16 not upgraded.
Need to get 1484 kB of archives.
After this operation, 5030 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Vemos los nuevos paquetes que van a ser instalados.

Vemos que 6 nuevos paquetes van a ser instalados.

Presionamos Y para continuar.

Preparamos el directorio de la infraestructura de la clave pública.

Después de instalar la entidad certificadora, es el momento de crear una infraestructura de clave pública básica en la entidad certificadora. Cree un directorio easy-rsa.

Creamos el directorio en nuestra carpeta de usuario con `mkdir ~/easy-rsa`

Usaremos esta carpeta para crear enlaces simbólicos que apunten a los archivos del paquete easy-rsa que hemos instalado. Se encuentran en la carpeta `/usr/share/easy-rsa` de la entidad certificadora.

Creamos el enlace simbólico con `ln`

`ln -s /usr/share/easy-rsa/* ~/easy-rsa/`

Para restringir el acceso a su nueva carpeta, asegúrese de que solo el propietario puede acceder mediante el comando `chmod`:

`chmod 700 /root/easy-rsa`

Cambiamos al directorio easy-rsa con `cd ~/easy-rsa` y iniciamos la clave pública con `./easyrsa init-pki`

Vemos que ahora puede crear una Entidad certificadora o solicitudes

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

Su directorio PKI creado es: `/root/easy-rsa/pki`

Después de completar esta sección, tiene una carpeta que contiene todos los archivos necesarios para crear una Entidad certificadora. Después, creará la clave privada y el certificado público para su Entidad Certificadora.

2.2.26 Creamos la clave pública, la clave privada y el certificado de la Entidad Certificadora

Antes de que pueda crear la clave privada y el certificado de su entidad certificadora, debe crear y completar un archivo llamado vars con algunos valores predeterminados.

Editamos el archivo vars con nano vars

`set_var EASYRSA_REQ_COUNTRY` ponemos nuestro país entre comillas, en nuestro caso España “ES”

`set_var EASYRSA_REQ_PROVINCE` ponemos nuestra comunidad autónoma sin tildes entre comillas, en nuestro caso “Andalucia”

`set_var EASYRSA_REQ_CITY` ponemos nuestra ciudad sin tildes entre comillas, en nuestro caso “Cordoba”

`set_var EASYRSA_REQ_ORG` ponemos nuestra organización, en nuestro caso “ProyectoPGG”

`set_var EASYRSA_REQ_EMAIL` ponemos nuestro correo electrónico, en nuestro caso “gagapa_aliesma20@iesmedinaazahara.es”

`set_var EASYRSA_REQ_OU` ponemos nuestra sección de la organización, en nuestro caso “VPNProyectoPGG”

Se asegurarán de que sus claves privadas y solicitudes de certificados estén configuradas para utilizar la criptografía de curva elíptica (ECC) moderna para generar claves y firmas seguras para sus clientes y el servidor OpenVPN.

Configurar sus servidores OpenVPN y CA para usar ECC significa que cuando un cliente y un servidor intentan establecer una clave simétrica compartida, pueden usar algoritmos de curva elíptica para realizar su intercambio. Usar ECC para un intercambio de claves es significativamente más rápido que usar Diffie-Hellman simple con el algoritmo RSA clásico, ya que los números son mucho más pequeños y los cálculos son más rápidos.

`set_var EASYRSA_ALGO` “ec”

`set_var EASYRSA_DIGEST` “sha512”

Para crear el par de claves raíz pública y privada para su autoridad de certificación, ejecute el comando `./easy-rsa` nuevamente, esta vez con la opción `build-ca`:

`./easyrsa build-ca`

```
Enter New CA Key Passphrase:  
Re-Enter New CA Key Passphrase:
```

Ponemos la clave de la entidad certificadora y repetimos la clave de la entidad certificadora.

Deberá ingresar la contraseña cada vez que necesite interactuar con su Entidad Certificadora, por ejemplo, para firmar o revocar un certificado.

También se le pedirá que confirme el nombre común de su Entidad Certificadora. El nombre común es el nombre que se usa para referirse a esta máquina en el contexto de la Autoridad de certificación.

```
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:PROYECTOPGGCA
```

En nuestro caso el nombre común va a ser PROYECTOPGGCA

Después pone se completó la creación de CA y ahora puede importar y firmar solicitudes de certificado.

Su nuevo archivo de certificado de Entidad certificadora se encuentra en:

/root/easy-rsa/pki/ca.crt

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

El siguiente paso de este tutorial es instalar OpenVPN. Easy-RSA lo utilizará en el servidor OpenVPN para generar una solicitud de certificado que luego verificará y firmará en el servidor CA.

2.2.27 Instalamos OpenVPN y creamos la solicitud de certificado del servidor OpenVPN y la clave privada del servidor OpenVPN

Actualizamos los repositorios e instalamos openvpn con `sudo apt update && sudo apt install openvpn`

```
The following NEW packages will be installed:
  liblzo2-2 libpkcs11-helper1 openvpn
0 upgraded, 3 newly installed, 0 to remove and 16 not upgraded.
Need to get 716 kB of archives.
After this operation, 1988 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Vemos los paquetes que se van a instalar.

Vemos que 3 nuevos paquetes se van a instalar.

Presionamos Y para continuar la instalación.

Crear una solicitud de certificado de servidor OpenVPN y una clave privada

El siguiente paso es generar una clave privada y una solicitud de firma de certificado en su servidor OpenVPN. Después de eso, su Entidad Certificadora lo firmará, creando el certificado requerido. Una vez que tenga un certificado firmado lo instalará para que lo use el servidor.

Ahora ejecutará `easyrsa` con la opción `gen-req` seguida de un nombre común. El Nombre Común del servidor será `proyectopggvpn`. Asegúrese de incluir también la opción de `nopass`. Si no lo hace, protegerá con contraseña el archivo de solicitud, lo que podría generar problemas de permisos más adelante.

La ejecutamos con `./easyrsa gen-req proyectopggvpn nopass`

Presionamos enter para confirmar el nombre común `proyectopggvpn`.

```
Keypair and certificate request completed. Your files are:
req: /root/easy-rsa/pki/reqs/proyectopggvpn.req
key: /root/easy-rsa/pki/private/proyectopggvpn.key
```

Vemos que la solicitud de certificado y la clave ya están creadas, la ubicaciones son las que vemos en la captura.

Esto creará una clave privada para el servidor y un archivo de solicitud de certificado llamado `proyectopggvpn.req`. Copie la clave del servidor en el directorio `/etc/openvpn/server`:

Lo copiamos con: `sudo cp /root/easy-rsa/pki/private/proyectopggvpn.key /etc/openvpn/server/`

Después de completar estos pasos, habrá creado con éxito una clave privada para su servidor OpenVPN. También ha generado una solicitud de firma de certificado

para el servidor OpenVPN. La Solicitud de firma de certificado ahora está listo para ser firmado. En la siguiente sección de este tutorial, aprenderá cómo firmar una Solicitud de firma de certificado con la clave privada de su servidor.

2.2.28 Firmar la solicitud de certificado del servidor OpenVPN

En el paso anterior, creó una solicitud de firma de certificado y una clave privada para el servidor OpenVPN. Ahora la Entidad Certificadora necesita conocer el certificado del servidor VPN y validarlo. Una vez que la Entidad Certificadora valide el certificado del servidor OpenVPN, los clientes que confien en su Entidad Certificadora también podrán confiar en el servidor OpenVPN.

Copiamos la solicitud de certificado del servidor OpenVPN a tmp con

```
cp /root/easy-rsa/pki/reqs/proyectopggvpn.req /tmp
```

Borramos la solicitud del certificado de /root/easy-rsa/pki/reqs con sudo rm /root/easy-rsa/pki/reqs/proyectopggvpn.req

Importamos la solicitud de certificado usando el script easyrsa:

```
./easyrsa import-req /tmp/proyectopggvpn.req proyectopggvpn
```

import-req es para importar la solicitud de certificado a la Entidad Certificadora.

/tmp/proyectopggvpn.req es la ruta de nuestra solicitud de certificado.

proyectopggvpn es el nombre del certificado.

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

Vemos que ha sido importada correctamente con el nombre proyectopggvpn.

Ahora puede usar este nombre para realizar operaciones de firma en esta solicitud.

Después, firme la solicitud ejecutando el script easyrsa con la opción sign-req, seguido del tipo de solicitud y el nombre común. El tipo de solicitud puede ser cliente o servidor. Dado que está trabajando con la solicitud de certificado del servidor OpenVPN, asegúrese de utilizar el tipo de solicitud del servidor:

Lo ejecutamos con ./easyrsa sign-req server proyectopggvpn

En el resultado, se le pedirá que verifique que la solicitud proviene de una fuente confiable. Escriba yes y luego presione ENTER para confirmar:

Pone está a punto de firmar el siguiente certificado.

Por favor revise los detalles que se muestran a continuación para mayor precisión. Tenga en cuenta que esta solicitud no ha sido verificado criptográficamente. Por favor, asegúrese de que proviene de una fuente confiable o que ha verificado la suma de verificación de la solicitud con el remitente.

La solicitud de firma, para ser firmado como un certificado de servidor por 3650 días.

Sujeto =

Nombre común = proyectopggvpn

Ponga yes para continuar o cualquier otro texto para abortar.

```
subject=
  commonName          = proyectopggvpn

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
```

```
Enter pass phrase for /root/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:'proyectopggvpn'
Certificate is to be certified until Sep 11 09:09:25 2025 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /root/easy-rsa/pki/issued/proyectopggvpn.crt
```

Metemos la clave de nuestra entidad certificadora.

Después verifica que la solicitud coincida con la firma.

Vemos que la firma es correcta.

El nombre distinguido del sujeto es el siguiente nombre común proyectopggvpn.

El certificado está certificado hasta el 11 de septiembre de 2025 a las 09:09:25 GMT durante 825 días.

Se ha escrito una nueva entrada en la base de datos.

Base de datos actualizada.

Certificado creado en /root/easy-rsa/pki/issued/proyectopggvpn.crt

Con estos pasos completos, ha firmado la solicitud de certificado del servidor OpenVPN utilizando la clave privada del servidor de Entidad Certificadora. El archivo proyectopggvpn.crt resultante contiene la clave de cifrado pública del servidor OpenVPN, así como una firma del servidor de Entidad Certificadora. El objetivo de la firma es decirle a cualquiera que confíe en el servidor de Entidad Certificadora que también puede confiar en el servidor OpenVPN cuando se conecta a él.

Para terminar de configurar los certificados, copie los archivos `proyectopggvpn.crt` y `ca.crt` a `/tmp`.

Copiamos `proyectopggvpn.crt` con `sudo cp pki/issued/proyectopggvpn.crt /tmp`

Copiamos `ca.crt` con `sudo cp pki/ca.crt /tmp`

Lo copiamos a `/etc/openvpn/server` con `sudo cp /tmp/{proyectopggvpn.crt,ca.crt} /etc/openvpn/server`

Ahora su servidor OpenVPN está casi listo para aceptar conexiones. En el siguiente paso, realizará algunos pasos adicionales para aumentar la seguridad del servidor.

2.2.29 Configuración del material criptográfico de OpenVPN, generación de una solicitud de certificado de cliente y el par de claves

Para una capa adicional de seguridad, agregue una clave secreta compartida adicional que el servidor y todos los clientes usarán con la directiva `tls-crypt` de OpenVPN. Esta opción se usa para ofuscar el certificado TLS que se usa cuando un servidor y un cliente se conectan entre sí inicialmente. El servidor OpenVPN también lo utiliza para realizar comprobaciones rápidas de los paquetes entrantes: si un paquete se firma con la clave precompartida, el servidor lo procesa; si no está firmado, el servidor sabe que proviene de una fuente que no es de confianza y puede descartarlo sin tener que realizar un trabajo de descifrado adicional.

Esta opción ayudará a garantizar que su servidor OpenVPN pueda hacer frente al tráfico no autenticado, los escaneos de puertos y los ataques de denegación de servicio, que pueden ocupar los recursos del servidor. También dificulta la identificación del tráfico de la red OpenVPN.

Para generar la clave precompartida `tls-crypt`, ejecute lo siguiente en el servidor OpenVPN en el directorio `~/easy-rsa: openvpn --genkey --secret ta.key`

El resultado será un archivo llamado `ta.key`. Cópielo en el directorio `/etc/openvpn/server/` con: `sudo cp ta.key /etc/openvpn/server`

Con estos archivos en su lugar en el servidor OpenVPN, está listo para crear certificados de cliente y archivos clave para sus usuarios, que utilizará para conectarse a la VPN.

Generación de un certificado de un cliente y el par de claves

Aunque puede generar una clave privada y una solicitud de certificado en su máquina cliente y luego enviarla a la Entidad Certificadora para que la firme, esta guía describe un proceso para generar la solicitud de certificado en el servidor OpenVPN. El beneficio de este enfoque es que puede crear una secuencia de comandos que generará automáticamente archivos de configuración del cliente que contienen todas las claves y certificados necesarios. Esto le permite evitar tener que transferir claves, certificados y archivos de configuración a los clientes y agiliza el proceso de unirse a la VPN.

Generará una sola clave de cliente y un par certificado para esta guía. Si tienes más de un cliente, puedes repetir este proceso para cada uno. Sin embargo, tenga en cuenta que deberá pasar un valor de nombre único al script para cada cliente. A lo largo de este tutorial, el primer par de certificado/clave se denomina **proyectopggvnpcliente**.

Comience creando una estructura de directorios dentro de su directorio de inicio para almacenar el certificado del cliente y los archivos clave: lo creamos con **mkdir -p ~/client-configs/keys**

Dado que almacenará los pares de certificados/clave de sus clientes y los archivos de configuración en este directorio, debe bloquear sus permisos ahora como medida de seguridad: lo hacemos con **chmod -R 700 ~/client-configs**

A continuación, ejecute el script **easyrsa** con las opciones **gen-req** y **nopass**, junto con el nombre común del cliente:

Lo ejecutamos con **./easyrsa gen-req proyectopggvnpcliente nopass**

Pulsamos Enter para confirmar el nombre común.

Vemos que se ha generado la solicitud de certificado y el par de claves del cliente, los directorios son:

```
Keypair and certificate request completed. Your files are:
req: /root/easy-rsa/pki/reqs/proyectopggvnpcliente.req
key: /root/easy-rsa/pki/private/proyectopggvnpcliente.key
```

Solicitud de certificado: **/root/easy-rsa/pki/reqs/proyectopggvnpcliente.req**

Clave del cliente: **/root/easy-rsa/pki/private/proyectopggvnpcliente.key**

2.2.30 Firmamos la solicitud del cliente con la entidad certificadora

Luego, copie el archivo **proyectopggvnpcliente.key** al directorio **~/client-configs/keys/** con **cp pki/private/proyectopggvnpcliente.key ~/client-configs/keys/**

Copiamos la solicitud de certificado a **/tmp** con: **cp pki/reqs/proyectopggvnpcliente.req /tmp**

Borramos la solicitud de certificado antigua con: **rm -r /root/easy-rsa/pki/reqs/proyectopggvnpcliente.req**

Importamos la solicitud de certificado con **./easyrsa import-req /tmp/proyectopggvnpcliente.req proyectopggvnpcliente**

Vemos que ha sido importada correctamente con el nombre **proyectopggvnpcliente**.

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

Vemos que ya podemos realizar operaciones de firma en esta solicitud.

Luego, firme la solicitud de la misma manera que lo hizo para el servidor en el paso anterior. Esta vez, sin embargo, asegúrese de especificar el tipo de solicitud del cliente:

`./easyrsa sign-req client proyectopggvpncliente`

```
subject=
  commonName                = proyectopggvpncliente

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
```

Vemos que el nombre común es `proyectopggvpncliente`

Ponemos `yes` para continuar.

Ponemos la clave de la entidad certificadora para firmar la solicitud de certificado y que es de una fuente confiable.

```
Enter pass phrase for /root/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:'proyectopggvpncliente'
Certificate is to be certified until Sep 11 10:30:43 2025 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /root/easy-rsa/pki/issued/proyectopggvpncliente.crt
```

Vemos que la firma está bien.

Vemos que el nombre común que es el mismo y que va a durar 825 días.

Vemos que se ha creado una entrada en la base de datos.

Vemos la ruta donde se ha creado el certificado en:
`/root/easy-rsa/pki/issued/proyectopggvpncliente.crt`

Copiamos a `/tmp` el certificado del cliente con: `sudo cp pki/issued/proyectopggvpncliente.crt /tmp`

Copiamos el certificado del cliente a la carpeta `~/client-configs/keys/` con: `cp /tmp/proyectopggvpncliente.crt ~/client-configs/keys/`

A continuación, copie los archivos `ca.crt` y `ta.key` también en el directorio `~/client-configs/keys/`: con `cp ~/easy-rsa/ta.key ~/client-configs/keys/` y `cp /etc/openvpn/server/ca.crt ~/client-configs/keys/`

Con eso, los certificados y claves de su servidor y cliente se han generado y se almacenan en los directorios apropiados en su servidor OpenVPN. Todavía hay algunas acciones que deben realizarse con estos archivos, pero vendrán en un paso posterior. Por ahora, puede continuar con la configuración de OpenVPN.

2.2.31 Configurar Openvpn y tener credenciales no predeterminadas

Como muchas otras herramientas de código abierto ampliamente utilizadas, OpenVPN tiene numerosas opciones de configuración disponibles para personalizar su servidor según sus necesidades específicas. En esta sección, proporcionaremos instrucciones sobre cómo establecer una configuración de servidor OpenVPN basada en uno de los archivos de configuración de muestra que se incluye en la documentación de este software.

Primero, copie el archivo `server.conf` de muestra como punto de partida para su propio archivo de configuración con:

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf /etc/openvpn/server/
```

Editamos el archivo con: `sudo nano /etc/openvpn/server/server.conf`

Deberá cambiar algunas líneas en este archivo. Primero, encuentre la sección HMAC de la configuración buscando la directiva `tls-auth`. Esta línea estará habilitada por defecto. Coméntalo agregando un `;` al comienzo de la línea. Luego agregue una nueva línea después que contenga el valor `tls-crypt ta.key` solamente:

```
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey tls-auth ta.key
#
# The server and each client must have
# a copy of this key.
# 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
```

A continuación, busque la sección sobre cifrados criptográficos buscando las líneas de cifrado. El valor predeterminado se establece en `AES-256-CBC`; sin embargo, el cifrado `AES-256-GCM` ofrece un mejor nivel de cifrado, rendimiento y es bien compatible con los clientes OpenVPN actualizados. Comente el valor predeterminado agregando un `;` firme al principio de esta línea y luego agregue otra línea después que contenga el valor actualizado de `AES-256-GCM`:

Inmediatamente después de esta línea, agregue una directiva de autenticación para seleccionar el algoritmo de resumen de mensajes HMAC. Para esto, `SHA256` es una buena opción:

```
# Select a cryptographic cipher.  
# This config item must be copied to  
# the client config file as well.  
# Note that v2.4 client/server will automatically  
# negotiate AES-256-GCM in TLS mode.  
# See also the ncp-cipher option in the manpage  
;cipher AES-256-CBC  
cipher AES-256-GCM  
auth SHA256
```

A continuación, busque la línea que contiene una directiva `dh`, que define los parámetros Diffie-Hellman. Dado que configuró todos los certificados para usar criptografía de curva elíptica, no es necesario un archivo semilla Diffie-Hellman. Comente la línea existente que parece `dh dh2048.pem` o `dh dh.pem`. El nombre de archivo de la clave Diffie-Hellman puede ser diferente al que se muestra en el archivo de configuración del servidor de ejemplo. Luego agregue una línea después con el contenido `dh none`:

```
# Diffie hellman parameters.  
# Generate your own with:  
# openssl dhparam -out dh2048.pem 2048  
;dh dh2048.pem  
dh none
```

Tener credenciales no predeterminadas

Si seleccionó un nombre diferente durante el comando del servidor `./easyrsa genreq` anteriormente, modifique las líneas `cert` y `key` en el archivo de configuración `server.conf` para que apunten a los archivos `.cert` y `.key` apropiados. Si usó el nombre predeterminado, servidor, esto ya está configurado correctamente:

```
# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key). Each client
# and the server must have their own cert and
# key file. The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys. Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/proyectopggvpn.crt
key /etc/openvpn/server/proyectopggvpn.key # This file should be kept secret
```

El certificado raíz SSL/TLS (`ca`), certificado (`cert`) y clave privada (`clave`). Cada cliente y el servidor debe tener su propio certificado y archivo de clave. El servidor y todos los clientes *use* el mismo archivo `ca`.

Ver el directorio "easy-rsa" para una serie de scripts para generar certificados RSA y claves privadas. Recuerda usar un nombre común único para el servidor y cada uno de los certificados de cliente.

Se puede utilizar cualquier sistema de gestión de claves X509. OpenVPN también puede usar un archivo de clave con formato PKCS 12 (consulte la directiva "pkcs12" en la página del manual).

Ponemos la ruta completa del certificado raíz SSL/TLS (`ca`), certificado (`cert`) y clave privada (`clave`) abajo, que es la misma carpeta que el archivo que estamos editando.

Ahora ha terminado de configurar sus ajustes generales de OpenVPN. En el siguiente paso, personalizará las opciones de red del servidor.

2.2.32 Ajuste de la configuración de red y firewall del servidor OpenVPN

Hay algunos aspectos de la configuración de red del servidor que deben modificarse para que OpenVPN pueda enrutar correctamente el tráfico a través de la VPN. El primero de ellos es el reenvío de IP, un método para determinar dónde se debe enrutar el tráfico de IP. Esto es esencial para la funcionalidad VPN que proporcionará su servidor.

Para configurar el enrutamiento editamos el siguiente archivo `/etc/sysctl.conf` con `sudo nano /etc/sysctl.conf`

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Descomentamos la siguiente línea para enrutar paquetes en IPv4.

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Para leer el archivo y cargar los nuevos valores ponemos: `sudo sysctl -p`

```
root@proyectopgg2:~# sudo sysctl -p
net.ipv4.ip_forward = 1
```

Vemos que se ha cargado correctamente.

Ahora su servidor OpenVPN podrá reenviar el tráfico entrante de un dispositivo ethernet a otro. Esta configuración garantiza que el servidor pueda dirigir el tráfico de los clientes que se conectan en la interfaz VPN virtual a sus otros dispositivos físicos de Ethernet. Esta configuración enrutará todo el tráfico web de su cliente a través de la dirección IP de su servidor, y la dirección IP pública de su cliente se ocultará de manera efectiva.

En el siguiente paso, deberá configurar algunas reglas de firewall para garantizar que el tráfico hacia y desde su servidor OpenVPN fluya correctamente.

Configuración de Firewall

Hasta ahora, instaló OpenVPN en su servidor, lo configuró y generó las claves y los certificados necesarios para que su cliente acceda a la VPN. Sin embargo, aún no ha proporcionado a OpenVPN ninguna instrucción sobre dónde enviar el tráfico web entrante de los clientes. Puede estipular cómo el servidor debe manejar el tráfico de clientes estableciendo algunas reglas de firewall y configuraciones de enrutamiento.

Para permitir OpenVPN a través del firewall, deberá habilitar el enmascaramiento, un concepto de iptables que proporciona traducción dinámica de direcciones de red (NAT) sobre la marcha para enrutar correctamente las conexiones de los clientes.

Antes de abrir el archivo de configuración del firewall para agregar las reglas de enmascaramiento, primero debe encontrar la interfaz de red pública de su máquina, ponemos: `ip route list default`

```
default via 192.168.56.1 dev eth0 proto static
```

Después de `dev`, vemos que la tarjeta de red es `eth0`.

Cuando tenga la interfaz asociada con su ruta predeterminada, abra el archivo `/etc/ufw/before.rules` para agregar la configuración relevante con: `sudo nano /etc/ufw/before.rules`

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#
#
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES

# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines
```

Eso sirve para que pueda enrutar el tráfico de OpenVPN a eth0.

Ahí que ponerlo donde lo hemos puesto.

A continuación, debe indicarle a UFW que también permita los paquetes reenviados de forma predeterminada. Para hacer esto, abra el archivo /etc/default/ufw con: `sudo nano /etc/default/ufw`

```
# Set the default forward policy to ACCEPT, DROP or REJECT. Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="DROP"
```

Por defecto es `DEFAULT_FORWARD_POLICY="DROP"`, lo cambiamos a `DEFAULT_FORWARD_POLICY="ACCEPT"` para poder enrutar por defecto.

```
# Set the default forward policy to ACCEPT, DROP or REJECT. Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="ACCEPT"
```

A continuación, ajuste el cortafuegos para permitir el tráfico a OpenVPN. Si no cambió el puerto y el protocolo en el archivo /etc/openvpn/server.conf, deberá abrir el tráfico UDP al puerto 1194. Si modificó el puerto y/o el protocolo, sustituya los valores que seleccionó aquí.

También añadimos el puerto del SSH y del FTP.

Añadimos VPN al Firewall con `sudo ufw allow 1194/udp`

Añadimos FTP al firewall con `sudo ufw allow 20/tcp` y `sudo ufw allow 21/tcp`, `sudo ufw allow 990/tcp` que es TLS, `sudo ufw allow 64000:65535/tcp`, que son puertos pasivos.

Añadimos SSH con `sudo ufw allow OpenSSH`.

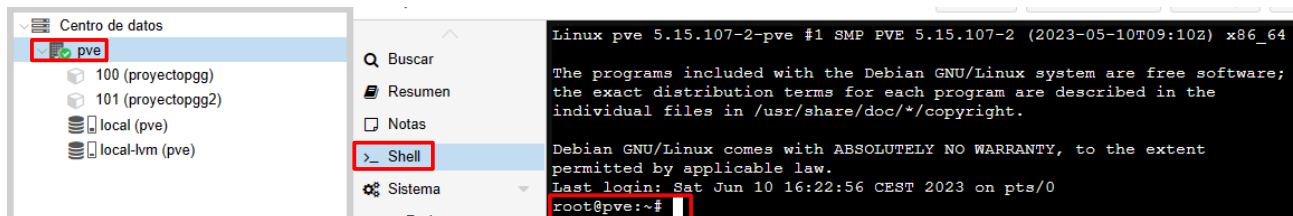
Añadimos apache2 con `sudo ufw allow http` y `sudo ufw allow https`

Desactivamos ufw después de los cambios con `sudo ufw disable`.

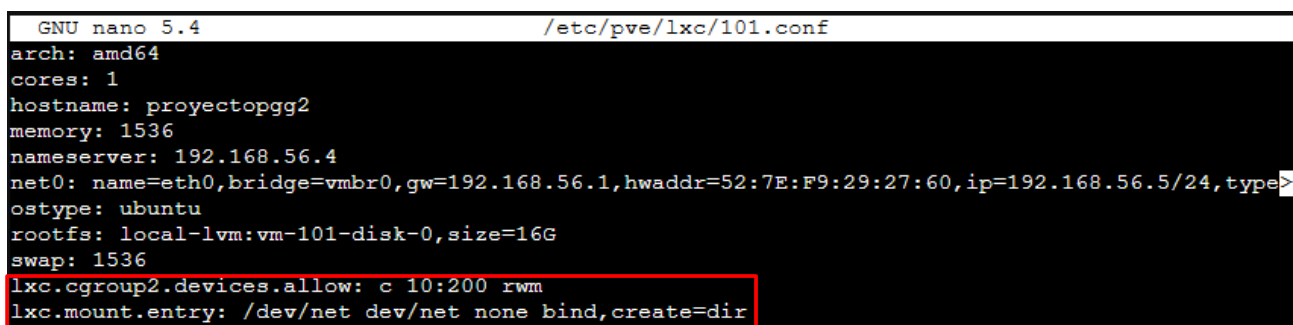
Lo activamos con `sudo ufw enable`.

Apagamos el contenedor con `poweroff`.

Hacemos clic en pve y luego en shell.



Editamos el archivo de configuración de nuestro contenedor con el id 101 con: `nano /etc/pve/lxc/101.conf`



Ponemos estas dos líneas, la primera es para crear el dispositivo, autorizarlo y la segunda es para crear la carpeta para montarlo y montarlo.

Guardamos el archivo y volvemos a encender el contenedor.

Su servidor ahora está configurado para manejar correctamente el tráfico de OpenVPN. Con las reglas del firewall implementadas, puede iniciar el servicio OpenVPN en el servidor.

2.2.33 Iniciamos OpenVPN y creamos la configuración del cliente

OpenVPN se ejecuta como un servicio systemd, por lo que puede usar `systemctl` para administrarlo. Configurará OpenVPN para que se inicie en el arranque para que pueda conectarse a su VPN en cualquier momento mientras su servidor esté funcionando. Para hacer esto, habilite el servicio OpenVPN agregándolo a `systemctl`, lo hacemos con: `sudo systemctl -f enable openvpn-server@server.service`

Esto inicia el servicio OpenVPN con: `sudo systemctl start openvpn-server@server.service`

Vemos que funciona correctamente OpenVPN con: `sudo systemctl status openvpn-server@server.service`

```
* openvpn-server@server.service - OpenVPN service for server
Loaded: loaded (/lib/systemd/system/openvpn-server@server.service; enabled; vendor preset: enabled)
Active: active (running) since Sat 2023-06-10 16:38:50 CEST; 25s ago
Docs: man:openvpn(8)
      https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
      https://community.openvpn.net/openvpn/wiki/HOWTO
Main PID: 193 (openvpn)
Status: "Initialization Sequence Completed"
Tasks: 1 (limit: 4596)
Memory: 5.1M
CPU: 32ms
CGroup: /system.slice/system-openvpn\x2dservers.slice/openvpn-server@server.service
        '-193 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-version 2 --suppress-timestamps --config server.conf

Jun 10 16:38:50 proyectorpgg2 openvpn[193]: net_addr_ptp_v4 add: 10.8.0.1 peer 10.8.0.2 dev tun0
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: net route v4 add: 10.8.0.0/24 via 10.8.0.2 dev [NULL] table 0 metric -1
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: Could not determine IPv4/IPv6 protocol. Using AF_INET
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: Socket Buffers: R=[212992->212992] S=[212992->212992]
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: UDPv4 link local (bound): [AF_INET][undef]:1194
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: UDPv4 link remote: [AF_UNSPEC]
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: MULTI: multi_init called, r=256 v=256
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: IFCONFIG POOL IPv4: base=10.8.0.4 size=62
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: IFCONFIG POOL LIST
Jun 10 16:38:50 proyectorpgg2 openvpn[193]: Initialization Sequence Complete
```

Vemos que pone active (running), por lo que funciona correctamente.

Vemos que se ha iniciado la dirección del servidor VPN 10.8.0.2 correctamente.

Vemos que se ha añadido la ruta de red correctamente.

Vemos que se ha iniciado la secuencia correctamente.

Crear la infraestructura de configuración del cliente

Describe un proceso para crear una infraestructura de configuración de cliente que puede usar para generar archivos de configuración sobre la marcha. Primero creará un archivo de configuración "base" y luego creará un script que le permitirá generar archivos de configuración, certificados y claves de cliente únicos según sea necesario.

Creamos la carpeta donde se almacenará la configuración del cliente que creamos con: `mkdir -p ~/client-configs/files`

A continuación, copie un archivo de configuración de cliente de ejemplo en el directorio client-configs para usarlo como su configuración base con:

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

Lo editamos con: `nano ~/client-configs/base.conf`

Vemos nuestra ip pública en internet en esta web <https://www.cual-es-mi-ip.net/>

Vemos que es:

Tu dirección IP es **80.30.69.42** 

Copiamos la dirección IP haciendo clic derecho y en copiar.

Busque la directiva `remote`. Esto dirige al cliente a la dirección de su servidor OpenVPN, la dirección IP pública de su servidor OpenVPN.

```
remote my-server-1 1194
;remote my-server-2 1194
```

Ponemos donde pone `my-server-1` nuestra dirección IP pública.

```
remote 80.30.69.42 1194
;remote my-server-2 1194
```

Encuentre las directivas que establecen `ca`, `cert` y `key`. Comente estas directivas ya que agregará los certificados y las claves dentro del propio archivo en breve:

```
;ca ca.crt
;cert client.crt
;key client.key
```

Comente la directiva `tls-auth`, ya que agregará `ta.key` directamente en el archivo de configuración del cliente (y el servidor está configurado para usar `tls-crypt`):

```
# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1
```

Refleje la configuración de cifrado y autenticación que estableció en el archivo `/etc/openvpn/server/server.conf`

Comentamos cifrado antiguo:

```
# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the data-ciphers option in the manpage
;cipher AES-256-CBC
```

Y ponemos el cifrado nuevo:

```
;cipher AES-256-CBC
cipher AES-256-GCM
auth SHA256
```

A continuación, agregue la directiva de dirección clave en algún lugar del archivo. Debe establecer esto en "1" para que la VPN funcione correctamente en la máquina cliente:

Lo añadimos después del cifrado:

```
auth SHA256
key-direction 1
```

Finalmente, agregue algunas líneas comentadas para manejar varios métodos que los clientes VPN basados en Linux usarán para la resolución de DNS. Agregará dos conjuntos similares, pero separados, de líneas comentadas. El primer conjunto es para clientes que no usan systemd-resolved para administrar DNS. Estos clientes confían en la utilidad resolvconf para actualizar la información de DNS para clientes Linux.

La agregamos al final del archivo.

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

Ahora agregue otro conjunto de líneas para los clientes que usan systemd-resolved para la resolución de DNS:

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

A continuación, creará una secuencia de comandos que compilará su configuración base con el certificado, la clave y los archivos de cifrado relevantes y luego colocará la configuración generada en el directorio ~/client-configs/files. Abra un nuevo archivo llamado make_config.sh dentro del directorio ~/client-configs con: nano ~/client-configs/make_config.sh

```
#!/bin/bash

# First argument: Client identifier

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
```

Hacemos ejecutable el archivo que acabamos de crear con: chmod 700 ~/client-configs/make_config.sh

Esta secuencia de comandos hará una copia del archivo base.conf que creó, recopilará todos los certificados y archivos clave que haya creado para su cliente, extraerá su contenido, los agregará a la copia del archivo de configuración base y exportará todo este contenido en un nuevo archivo de configuración del cliente. Esto significa que, en lugar de tener que administrar la configuración, el certificado y

los archivos clave del cliente por separado, toda la información requerida se almacena en un solo lugar. El beneficio de usar este método es que si alguna vez necesita agregar un cliente en el futuro, puede ejecutar este script para crear rápidamente un nuevo archivo de configuración y asegurarse de que toda la información importante se almacene en un único archivo de una localización de fácil acceso.

Tenga en cuenta que cada vez que agregue un nuevo cliente, deberá generar nuevas claves y certificados para él antes de poder ejecutar este script y generar su archivo de configuración. Obtendrá algo de práctica usando este script en el siguiente paso.

2.2.34 Generamos la configuración del cliente y lo copiamos al cliente

Puede generar un archivo de configuración para estas credenciales ejecutando el script que hizo al final del paso anterior:

Entramos a client-configs con `cd ~/client-configs`

Ejecutamos el script con: `./make_config.sh proyectopggvpncliente`

Vemos los archivos de la carpeta de configuración de archivos del cliente con:
`ls ~/client-configs/files`

Vemos ya preparado nuestro archivo.

`proyectopggvpncliente.ovpn`

Hay que transferir el archivo a nuestro cliente Ubuntu.

Un método confiable y seguro para transferir es usar SFTP (protocolo de transferencia de archivos SSH) o SCP (Copia segura). Esto transportará los archivos de autenticación VPN de su cliente a través de una conexión cifrada.

Ejecutamos en el terminal lo siguiente:

`scp ~/client-configs/files/proyectopggvpncliente.ovpn usuario@192.168.56.150:~/`

`~/client-configs/files/proyectopggvpncliente.ovpn` es donde está ubicado el archivo del cliente de nuestro VPN en el servidor VPN.

`usuario` es el usuario de nuestro cliente.

`192.168.56.150` es la dirección ip de nuestro cliente.

`~/` es la ruta donde lo vamos a copiar al cliente.

Ponemos la contraseña de nuestro cliente y vemos que se ha copiado.

```
usuario@192.168.56.150's password:
proyectopggvpncliente.ovpn
```

100% 12KB 8.7MB/s 00:00

2.2.35 Instalamos OpenVPN y la configuración en Lubuntu con systemd-resolved

La conexión tendrá el mismo nombre que el archivo.

La mejor forma de conectarse es usar el software OpenVPN.

Recargamos los repositorios y instalamos OpenVPN en Lubuntu con: `sudo apt update && sudo apt install openvpn`

```
Se instalarán los siguientes paquetes NUEVOS:
  libpkcs11-helper1 openvpn
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 50 no actualizados.
Se necesita descargar 663 kB de archivos.
Se utilizarán 1.825 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Vemos los paquetes nuevos que se van a instalar, 2 nuevos paquetes se instalarán y presionamos S para continuar.

Para ver si usa systemd-resolved para la resolución DNS en Lubuntu usamos: `cat /etc/resolv.conf`

Si su sistema está configurado para usar systemd-resolved para la resolución de DNS, la dirección IP después de la opción del servidor de nombres será 127.0.0.53. También debe haber comentarios en el archivo, como el resultado que se muestra, que explican cómo systemd-resolved está administrando el archivo. Si tiene una dirección IP diferente a 127.0.0.53, es probable que su sistema no esté usando systemd-resolved y puede pasar a la siguiente sección sobre cómo configurar clientes Linux que tienen un script update-resolv-conf en su lugar.

```
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8)
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.
nameserver 127.0.0.53
```

Vemos que **systemd-resolved** está administrando el archivo y la dirección IP después de la opción del servidor de nombres será **127.0.0.53**

Primero instale el paquete **openvpn-systemd-resolved**. Proporciona scripts que obligarán a **systemd-resolved** a usar el servidor VPN para la resolución de DNS con:

```
Se instalarán los siguientes paquetes NUEVOS:
  libnss-resolve openvpn-systemd-resolved
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 50 no actualizados.
Se necesita descargar 76,9 kB de archivos.
Se utilizarán 351 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Vemos los paquetes nuevos que se van a instalar, 2 nuevos paquetes se instalarán y presionamos **S** para continuar.

Una vez que el paquete esté instalado, configure el cliente para usarlo y envíe todas las consultas de DNS a través de la interfaz VPN. Abra el archivo VPN del cliente con: **nano proyectopggvpncliente.ovpn**

Ahora descomente las siguientes líneas que agregó anteriormente:

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

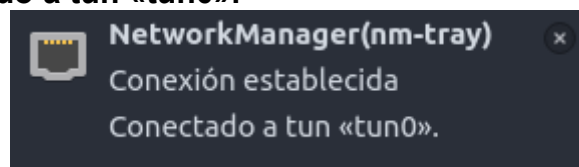
Abrimos los puertos del router para poder usar OpenVPN.

2.2.36 Conectarse a OpenVPN desde Ubuntu y comprobar que funciona correctamente

Ahora, puede conectarse a la VPN simplemente apuntando el comando **openvpn** al archivo de configuración del cliente con: **sudo openvpn --config proyectopggvpncliente.ovpn**

Esto debería conectarlo a su VPN.

Vemos que al iniciar el VPN, pone en NetworkManager (nm-tray), Conexión establecida y Conectado a tun «tun0».



Vemos que pone dispositivo tun0 abierto, vemos que pone que tun0 está encendido,
 Vemos que añade la dirección IP 10.8.0.6 y la puerta de enlace de tun0 10.8.0.5
 Vemos que lo inicia con el update-systemd-resolved el tun0
 Vemos que añade la ruta del dominio DNS el update-systemd-resolved.

```

2023-06-11 13:30:09 TUN/TAP device tun0 opened
2023-06-11 13:30:09 net_iface_mtu_set: mtu 1500 for tun0
2023-06-11 13:30:09 net_iface_up: set tun0 up
2023-06-11 13:30:09 net_addr_ptp_v4_add: 10.8.0.6 peer 10.8.0.5 dev tun0
2023-06-11 13:30:09 /etc/openvpn/update-systemd-resolved tun0 1500 1624 10.8.0.6 10.8.0.5 init
<14>Jun 11 13:30:09 update-systemd-resolved: Link 'tun0' coming up
<14>Jun 11 13:30:09 update-systemd-resolved: Adding DNS Routed Domain .
<14>Jun 11 13:30:09 update-systemd-resolved: SetLinkDomains(11 1 . true)
2023-06-11 13:30:09 net_route_v4_add: 10.8.0.1/32 via 10.8.0.5 dev [NULL] table 0 metric -1
2023-06-11 13:30:09 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2023-06-11 13:30:09 Initialization Sequence Completed
  
```

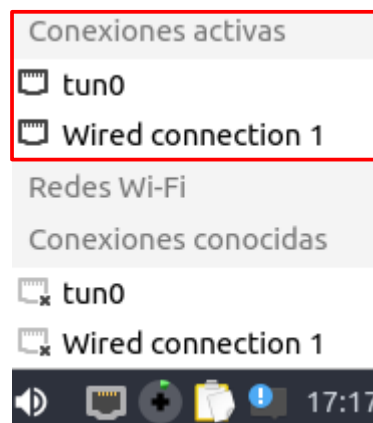
Vemos que añade dominios de enlace el update-systemd-resolved

Vemos que añade la ruta 10.8.0.1/32 via 10.8.0.5.

Pone ADVERTENCIA Esta Configuración cachea contraseñas en memoria, usa la opción auth-nocache para prevenir esto.

Pone Iniciación Secuencial Completada.

Vemos que tenemos dos conexiones de red activas.



Vemos que la máscara de subred es 255.255.255.255



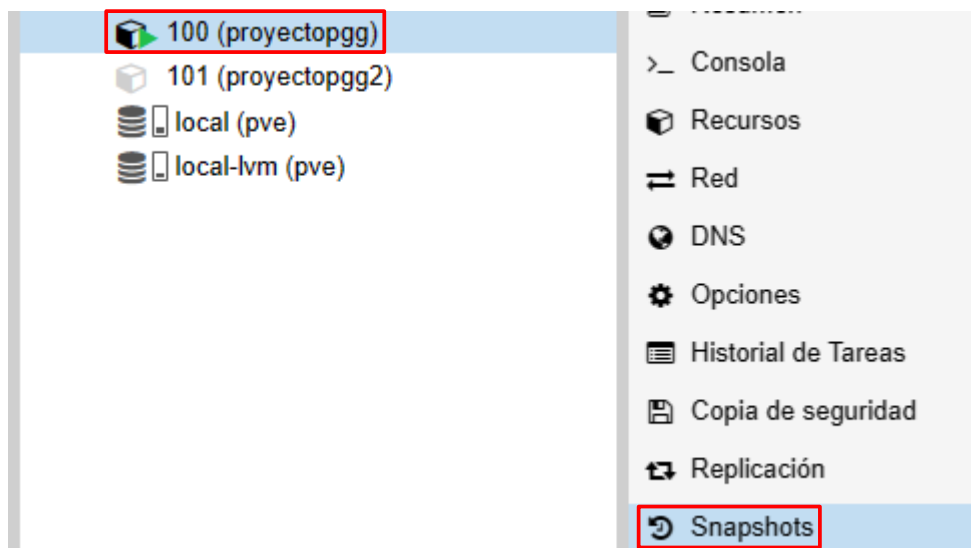
Vemos que tenemos internet.



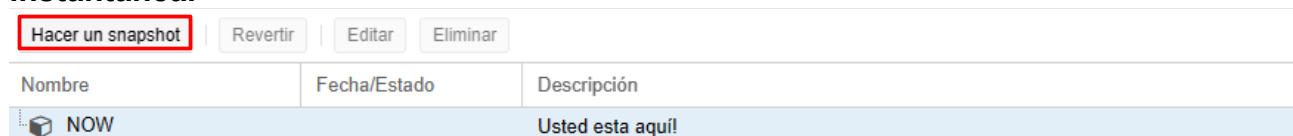
Para desconectarse presionamos **Ctrl + C**.

2.2.37 Hacer instantáneas por si algo falla y poder recuperarlo al mismo punto donde la hicimos el contenedor

Hacemos clic en nuestro contenedor y en **Snapshots** para ver las instantáneas de nuestro contenedor.



Vemos las instantáneas y hacemos clic en **Hacer un snapshot** para hacer una instantánea.



Crear: CT100 Snapshot

Nombre: Prueba

Descripción: Instantánea de prueba.

Hacer un snapshot

Ponemos un nombre, una descripción y hacemos clic en Hacer un snapshot para hacer una instantánea.

Task viewer: CT 100 - Snapshot

Salida Estado

Parar Descargar

WARNING: You have not turned on protection against thin pools running out of space.
 WARNING: Set activation/thin_pool_autoextend_threshold below 100 to trigger automatic extension of thin pools before they get full.
 Logical volume "snap_vm-100-disk-0_Prueba" created.
 WARNING: Sum of all thin volume sizes (48.00 GiB) exceeds the size of thin pool pve/data and the amount of free space in volume group (<7.88 GiB).

TASK OK

Pone Advertencia: No ha activado la protección contra los grupos delgados cuando se quedan sin espacio.

Después pone Advertencia: Configure la activation/thin_pool_autoextend_threshold por debajo de 100 para activar la extensión automática de grupos reducidos antes de que se llenen.

Después pone: Volumen lógico "snap_vm-100-disk-0_Prueba" creado.


Después pone Advertencia: La suma de todos los tamaños de los grupos delgados (48,00 GiB) excede el tamaño del grupo delgado pve/data y la cantidad de espacio libre en el grupo de volumen (<7,88 GiB).

Después pone tarea terminada correctamente y cerramos la ventana.

```
The following packages will be upgraded:
  jicofo jitsi-meet jitsi-meet-prosody jitsi-meet-turnserver jitsi-meet-web jitsi-meet-web-config jitsi-videobridge2 libcap2 libcap2-bin
  libqlib2.0-0 libqlib2.0-data libpam-cap libx11-6 libx11-data linux-libc-dev vim-common vim-tiny xxd
18 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
11 standard LTS security updates
Need to get 82.4 MB of archives.
After this operation, 210 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

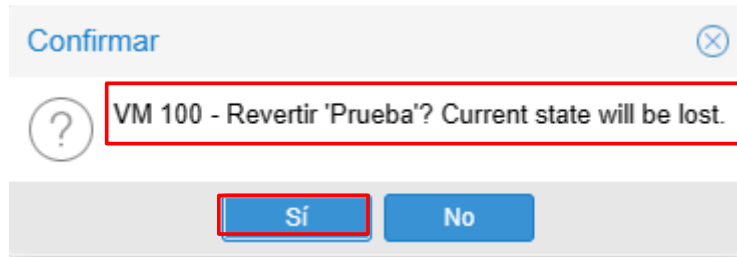
Vemos que todos los paquetes ya están actualizados.

All packages are up to date.

Hacer un snapshot	Revertir	Editar	Eliminar
Nombre	Fecha/Estado	Descripción	
 Prueba	2023-06-18 18:46:14	Instantánea de prueba.	

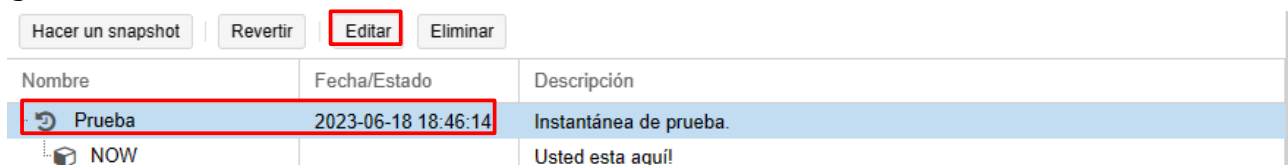
Hacemos clic en el nombre de la instantánea y en Revertir para deshacer los cambios, también vemos la fecha y la hora cuándo se hizo.

Hacemos clic en **Sí** para **revertir** el **contenedor 100** al estado de **Prueba** y el estado actual se perderá.



```
The following packages will be upgraded:
  jicofo jitsi-meet jitsi-meet-prosody jitsi-meet-turnserver jitsi-meet-web jitsi-meet-web-config jitsi-videobridge2 libcap2 libcap2-bin
  libglib2.0-0 libglib2.0-data libpam-cap libx11-6 libx11-data linux-libc-dev vim-common vim-tiny xxd
18 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
11 standard LTS security updates
Need to get 82.4 MB of archives.
After this operation, 210 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Podemos hacer clic en **Editar** para ver la configuración del equipo en la instantánea. **S**





Podemos ver los **núcleos** que **tenía** el **contenedor**, el **nombre**, la **memoria RAM**, la **dirección IP**, etc.

Configuración	
Clave ↑	Valor
arch	amd64
cores	1
hostname	proyectopgg
memory	1536
nameserver	192.168.56.4

Ver el **nombre** de la **tarjeta de red**, el **adaptador puente** de la tarjeta de red, la **puerta de enlace**, la **dirección MAC**, etc.

El **sistema operativo**, donde está ubicado el **disco**, el **nombre del disco**, el **tamaño del disco**, el **tamaño de la memoria RAM virtual** que **está en el disco**.

net0	name=eth0,bridge=vmbr0,gw=192.168.56.1,hwaddr=1A:7D:DE:07:4...
ostype	ubuntu
rootfs	local-lvm:vm-100-disk-0,size=16G
swap	1536


Hacer un snapshot	Revertir	Editar	Eliminar
Nombre	Fecha/Estado	Descripción	
 Prueba	2023-06-18 18:46:14	Instantánea de prueba.	
 NOW		Usted esta aquí!	

Hacemos clic en **Eliminar** para eliminar la instantánea.
 Hacemos clic en **Si** para eliminar el elemento Prueba.
 Vemos que ya no está Prueba.

Nombre	Fecha/Estado	Descripción
 NOW		Usted esta aquí!


2.2.38 Podemos monitorear el equipo

proyectopgg (Tiempo de uso: 00:22:30)




 Estado

running




 HA Estado

ninguno




 Nodo

pve




 Uso de CPU

0.33% de 1 CPU(s)




 Memoria - Uso

27.87% (428.03 MiB de 1.50 GiB)



 Memoria SWAP

0.00% (0 B de 1.50 GiB)



 Tamaño de disco de arranque

8.87% (1.38 GiB de 15.58 GiB)

Notas

Podemos ver el tiempo que ha estado encendido el contenedor, si está encendido o apagado en el estado, si está duplicado en varios proxmox por si alguno se cae y funciona en otro con los mismos datos del disco duro.

El nodo donde está el contenedor.

El uso de todos los núcleos del procesador, el uso de la memoria RAM, el uso de la memoria RAM virtual, el uso del disco.

Podemos poner notas.

Hora (promedio) ▾

Hora (promedio)

Hora (máximo)

Día (promedio)

Día (máximo)

Semana (promedio)

Semana (máximo)

Mes (promedio)

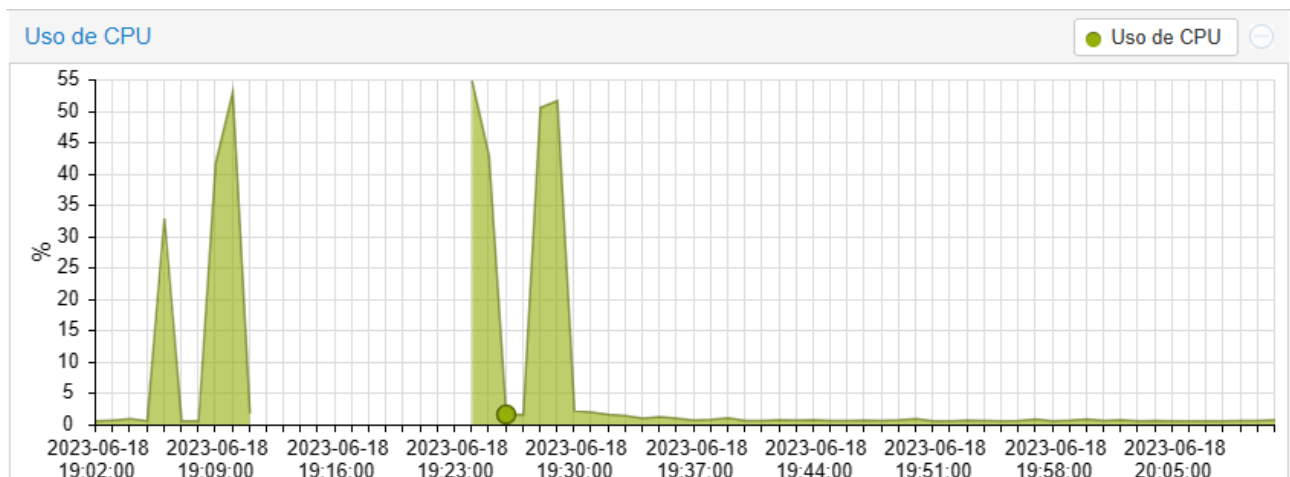
Mes (máximo)

Año (promedio)

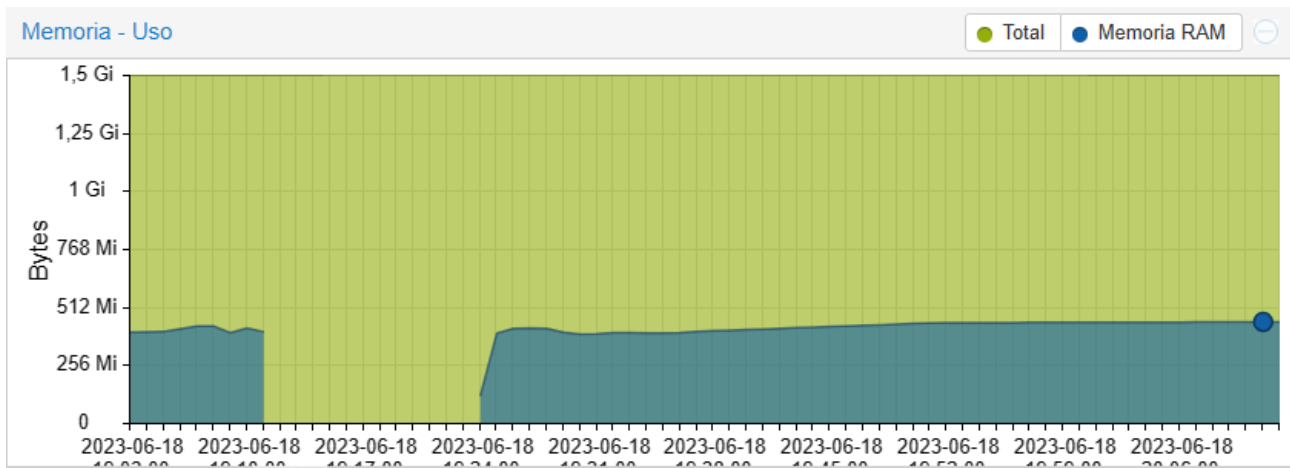
Año (máximo)

106 / 115

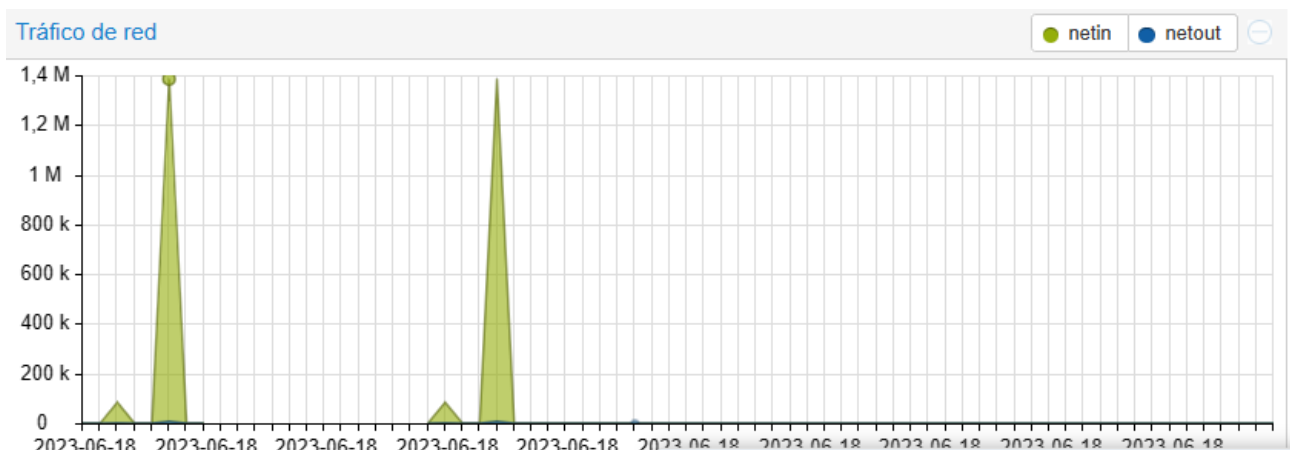
Podemos ver el promedio o el máximo de uso de los recursos, en horas, días, semanas, meses y años.



Podemos ver el % de uso de la CPU.

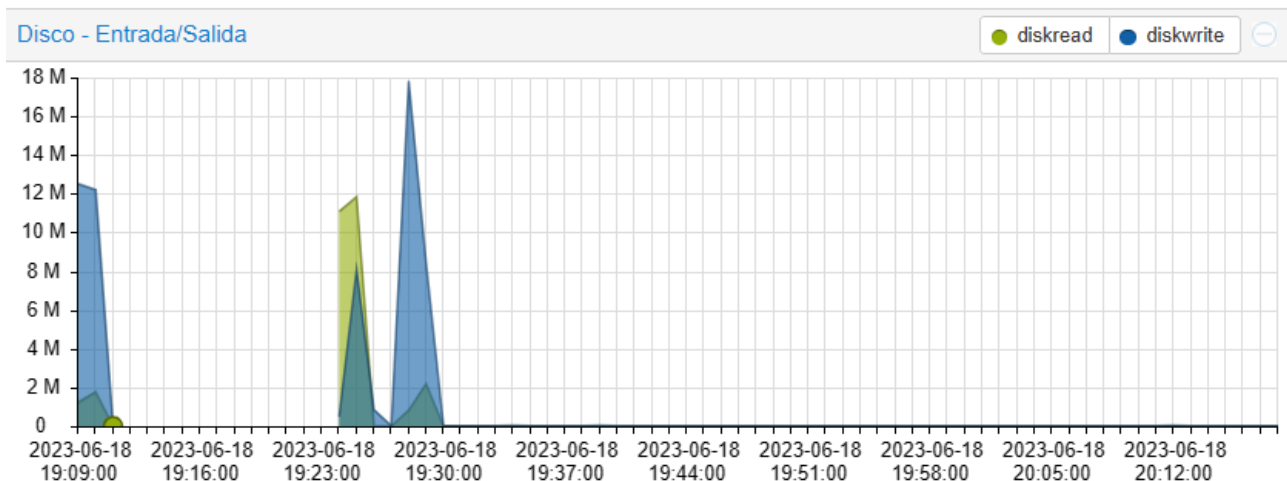


Podemos ver el % de uso de la Memoria RAM.



Podemos ver el tráfico de entrada y salida de la tarjeta de red en Megabytes.

Verde es tráfico de entrada al servidor y azul es el tráfico de salida del servidor.



Podemos ver la lectura y escritura del disco en Megabytes.

Verde es el tráfico de lectura del disco y azul es el tráfico de escritura del disco.

2.3 Viabilidad

El **proyecto es viable**, porque **solo** habría que **comprar el servidor**, ya que lo demás lo tenemos en clase.

3. Conclusiones

Hemos aprendido a como **instalar** distintos **servicios de red en dos contenedores de Proxmox** y ha **funcionado correctamente**.

Hemos **instalado** el **servidor DHCP**, servidor **DNS**, servidor **WEB**, usar el **contenedor como router**, el **servidor FTP** y el **servidor VPN**, he **tardado tiempo en instalarlo**, porque **no me salía a la primera el VPN**. **No he podido instalar el correo electrónico porque al enviar correos electrónicos desde un equipo a otro, no se recibían en el otro equipo a pesar de tener dos zonas directas diferentes en el DNS y escritas correctamente, por lo que no he logrado todos los objetivos.**

He **seguido la planificación y la metodología prevista ha sido la adecuada**, para **garantizar el éxito del trabajo** he tenido que **quitar el correo electrónico porque al enviar correos electrónicos desde un equipo a otro no se recibían en el otro equipo a pesar de tener dos zonas directas diferentes y estar bien escrito.**

Podemos **mejorarlo instalando un correo electrónico si al configurar el DNS funciona correctamente**, un **blog de Wordpress**, un **servicio de chat** para hablar con los que tengan la aplicación y el código para entrar.

4. Glosario

- **DHCP (Dynamic Host Configuration Protocol o protocolo de configuración dinámica de host):** es un protocolo de red de tipo cliente/servidor mediante el cual

un servidor DHCP asigna dinámicamente una dirección IP y otros parámetros de configuración de red a cada dispositivo en una red para que puedan comunicarse con otras redes IP. Este servidor posee un rango de direcciones IP dinámicas y las va asignando a los clientes conforme estas van quedando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después. Así los clientes de una red IP pueden conseguir sus parámetros de configuración automáticamente.

Asignación manual o estática

Asigna una dirección IP a una dirección MAC de la tarjeta de red. Se suele utilizar cuando se quiere controlar la asignación de dirección IP a cada cliente, también evita que se conecten clientes no identificados.

Asignación dinámica

El único método que permite la reutilización de las direcciones IP. El administrador de la red determina un rango de direcciones IP y cada dispositivo conectado a la red está configurado para solicitar su dirección IP al servidor cuando la tarjeta de interfaz de red se inicializa. El procedimiento es que puede usar una dirección IP en un intervalo de tiempo controlable un dispositivo. Esto facilita la instalación de nuevos dispositivos en un wifi gratis, por ejemplo.

- **DNS (Domain Name System o Sistemas de nombre de dominio en español):**

Es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP como Internet o una red privada. Este sistema asocia información variada con nombres de dominio asignados a cada uno de los participantes. Su función más importante es «traducir» nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio Google es 216.58.210.163, la mayoría de la gente llega a este equipo especificando www.google.com y no la dirección IP. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre del sitio web. El usuario recibirá la dirección IP del servidor más cercano con el DNS según su localización geográfica.

- **Servidor Web Apache2:**

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras.

El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (httpd).

Apache tiene amplia aceptación en la red: desde 1996, Apache es el servidor HTTP más usado. Jugó un papel fundamental en el desarrollo de la World Wide Web y alcanzó su máxima cuota de mercado en 2005, siendo el servidor empleado en el 70% de los sitios web en el mundo. Sin embargo, ha sufrido un descenso en su cuota de mercado en los últimos años (estadísticas históricas y de uso diario proporcionadas por Netcraft). En 2009, se convirtió en el primer servidor web que alojó más de 100 millones de sitios web.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Las ventajas son: qué es Modular, de Código abierto, es Multi-plataforma, es Extensible y es Popular (fácil conseguir ayuda/suporte).

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby). Apache está incluido por defecto muchas distribuciones Linux.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

- **Servidor FTP (File Transfer Protocol o Protocolo de transferencia de archivos):**

Es un protocolo de red para la transferencia de archivos basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El servicio FTP usa el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor y/o apropiarse de los archivos transferidos.

Para solucionar este problema son de gran utilidad aplicaciones como SCP y SFTP, incluidas en el paquete SSH, que permiten transferir archivos pero cifrando todo el tráfico.

Servidor FTP:

Un servidor FTP es un programa que se ejecuta en el servidor y permite el intercambio de archivos entre diferentes servidores/ordenadores.

Los servidores FTP no están en ordenadores personales, por lo que un usuario normalmente utilizará el FTP para conectarse remotamente a uno y así intercambiar información con él.

Las aplicaciones más comunes de los servidores FTP suelen ser el alojamiento web, en el que sus clientes utilizan el servicio para subir sus páginas web y sus archivos correspondientes; o como servidor de backup (copia de seguridad) de los archivos importantes que pueda tener una empresa. Para ello, existen protocolos de comunicación FTP para que los datos se transmitan cifrados, como el SFTP (Secure File Transfer Protocol).

Cliente FTP:

Cuando un navegador no está equipado con la función FTP, o si se quiere cargar archivos en un ordenador remoto, se necesitará utilizar un programa cliente FTP. Un cliente FTP es un programa que se instala en el ordenador del usuario, y que emplea el protocolo FTP para conectarse a un servidor FTP y transferir archivos, ya sea para descargarlos o para subirlos.

Para utilizar un cliente FTP, se necesita conocer el nombre del archivo, el ordenador en que reside (servidor, en el caso de descarga de archivos), el ordenador al que se quiere transferir el archivo (en caso de querer subirlo nosotros al servidor), y la carpeta en la que se encuentra.

Algunos clientes de FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Microsoft Windows, DOS, GNU/Linux y Unix. Sin embargo, hay disponibles clientes con opciones añadidas e interfaz gráfica. Aunque muchos navegadores tienen ya integrado FTP, es más confiable a la hora de conectarse con servidores FTP no anónimos utilizar un programa cliente.

Un cliente FTP es Fillezilla.

- **VPN (virtual private network o red privada virtual):**

Es una tecnología de red de ordenadores que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet. Permite que el ordenador en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada, con toda la funcionalidad, seguridad y políticas de gestión de una red privada. Esto se realiza estableciendo una conexión virtual punto a punto mediante el uso de conexiones dedicadas, cifrado o la combinación de ambos métodos.

Ejemplos comunes son la posibilidad de conectar dos o más sucursales de una empresa utilizando como vínculo Internet, permitir a los miembros del equipo de soporte técnico la conexión desde su casa al centro de cómputo o bien que un usuario pueda acceder a su equipo doméstico desde un sitio remoto, como por ejemplo un hotel. Todo ello utilizando la infraestructura de Internet.

La conexión VPN a través de Internet es técnicamente una unión wide area network (WAN) entre los sitios, pero al usuario le parece como si fuera un enlace privado: de allí la designación virtual private network.

Características básicas de seguridad

Para hacerlo posible de manera segura es necesario proporcionar los medios para garantizar la autenticación.

Autenticación y autorización: ¿quién está del otro lado? Usuario/equipo y qué nivel de acceso debe tener.

Integridad: de que los datos enviados no han sido alterados. Para ello se utilizan funciones de Hash. Los algoritmos de hash más comunes son los Message Digest (MD2 y MD5) y el Secure Hash Algorithm (SHA).

Confidencialidad/Privacidad: dado que solamente puede ser interpretada por los destinatarios de la misma. Se hace uso de algoritmos de cifrado como Data Encryption Standard (DES), Triple DES (3DES) y Advanced Encryption Standard (AES).

No repudio: es decir, un mensaje tiene que ir firmado, y quien lo firma no puede negar que envió el mensaje.

Control de acceso: se trata de asegurar que los participantes autenticados tienen acceso únicamente a los datos a los que están autorizados.

Auditoría y registro de actividades: se trata de asegurar el correcto funcionamiento y la capacidad de recuperación.

Calidad del servicio: se trata de asegurar un buen rendimiento, que no haya una degradación poco aceptable en la velocidad de transmisión.

- **Puerta de enlace o gateway:**

La pasarela (en inglés gateway) o puerta de enlace es el dispositivo que actúa de interfaz de conexión entre aparatos o dispositivos, y también posibilita compartir recursos entre dos o más ordenadores.

Su propósito es traducir la información del protocolo utilizado en una red inicial, al protocolo usado en la red de destino.

La pasarela es normalmente un equipo informático configurado para dotar a las máquinas de una red de área local (Local Area Network, LAN) conectadas a él de un acceso hacia una red exterior, generalmente realizando para ello operaciones de traducción de direcciones de red (Network Address Translation, NAT). Esta capacidad de traducción de direcciones permite aplicar una técnica llamada enmascaramiento de IP, usada muy a

menudo para dar acceso a Internet a los equipos de una LAN compartiendo una única conexión a Internet, y por tanto, una única dirección IP externa.

La dirección IP de una pasarela a menudo es 192.168.1.1 o 192.168.0.1 y utiliza algunos rangos predefinidos, como por ejemplo 127.x.x.x, 10.x.x.x, 172.x.x.x, 192.x.x.x. La puerta de enlace de un enrutador se puede averiguar ejecutando el comando `ipconfig` desde el símbolo del sistema de Windows, o con el comando `ip route` desde una terminal en macOS y GNU/Linux.

La puerta de enlace predeterminada (default gateway) es la ruta predeterminada o ruta por defecto que se le asigna a un equipo y tiene como función enviar cualquier paquete del que no conozca por cuál interfaz enviarlo y no esté definido en las rutas del equipo, enviando el paquete por la ruta predeterminada.

Máscara de red:

La máscara de red es una combinación de bits que sirve en el ámbito de las redes de ordenadores, cuya función es indicar a los dispositivos qué parte de la dirección IP es el número de la red, incluyendo la subred, y qué parte es la correspondiente al host.

Ejemplos:

8bit x 4 octetos = 32 bit. (11111111.11111111.11111111.11111111 = 255.255.255.255)

8bit x 3 octetos = 24 bit. (11111111.11111111.11111111.00000000 = 255.255.255.0)

8bit x 2 octetos = 16 bit. (11111111.11111111.00000000.00000000 = 255.255.0.0)

8bit x 1 octetos = 8 bit. (11111111.00000000.00000000.00000000 = 255.0.0.0)

- Jitsi

Jitsi es un software de videoconferencia, VoIP, y mensajería instantánea con aplicaciones nativas para iOS y Android, y con soporte para Windows, Linux y Mac OS X a través de la web. Es compatible con varios protocolos populares de mensajería instantánea y de telefonía, y se distribuye bajo los términos de la licencia Apache, por lo que es software libre y de código abierto. Jitsi se diferencia de otras plataformas de videoconferencia como Zoom en que puede ser instalado en un servidor privado, ya que tanto el software del cliente como del servidor son distribuidos libremente. Además, debido a su modelo de desarrollo abierto, permite añadir y/o modificar su funcionalidad a voluntad del usuario.

- **Características:**

Jitsi soporta varios sistemas operativos, incluyendo Windows, así como sistemas de tipo UNIX, como Linux, Mac OS X y BSD. También incluye:

- Autoreconexión tras corte de red.
- Grabación de llamadas.
- Cifrado con protocolos **SRTP** y **ZRTP**.
- Llamadas de varios participantes.
- Proyección de escritorio.

- Almacenamiento de contraseñas cifradas con una contraseña maestra.
- Transferencia de archivos para los servicios **XMPP**, **AIM/ICQ** y **Yahoo!**
- Cifrado de mensajería instantánea con **Off-the-Record Messaging**.
- Indicador de mensaje en espera.
- Llamadas de voz y vídeo mediante protocolos **SIP** y **XMPP**, con **H.264**, **H.263**, **VP8** para codificación de vídeo.

Jitsi está escrito sobre todo en Java, que ayuda a reutilizar la mayor parte del código en los distintos sistemas operativos que trabaja.

El proyecto utiliza la aplicación **Apache Felix OSGi** para **modularidad**.

Entre otros, **Jitsi** utiliza la **stack** (pila) de **protocolos JAIN-SIP** para el **soporte** de **SIP** y la **biblioteca Smack** para **XMPP**.

- **Protocolos admitidos:**

Los **siguientes protocolos** están **soportados** por Jitsi:

- **Bonjour** (aplicación de Apple Zeroconf).
- **OSCAR** (AIM/ICQ/. Mac).
- **SIP**.
- **Mensajería XMPP**, (Hangouts, Live Journal, Gizmo5, FB Chat, ...).
- **Yahoo!** (sólo funciones básicas de chat y transferencia de archivos).

5. Referencias

Ajustar zona horaria Ubuntu: <http://somebooks.es/establecer-la-fecha-hora-y-zona-horaria-en-la-terminal-de-ubuntu-20-04-lts/>

Servidor DHCP: https://es.wikipedia.org/wiki/Protocolo_de_configuraci%C3%B3n_din%C3%A1mica_de_host

Servidor DNS: https://es.wikipedia.org/wiki/Sistema_de_nombres_de dominio

Servidor Apache2: https://es.wikipedia.org/wiki/Servidor_HTTP_Apache

Servidor FTP: https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_archivos

Cómo configurar un servicio FTP seguro con vsftpd en Linux:

<https://www.xmodulo.com/secure-ftp-service-vsftpd-linux.html>

¿Inicios de sesión anónimos frente a invitados en vsftpd?:

<https://unix.stackexchange.com/questions/351885/anonymous-vs-guest-logins-in-vsftpd>

Página de manual de vsftpd:

https://security.appspot.com/vsftpd/vsftpd_conf.html

Usuarios virtuales VSFTP:

<https://askubuntu.com/questions/575523/how-to-setup-virtual-users-for-vsftpd-with-access-to-a-specific-sub-directory>

Servidor VPN: https://es.wikipedia.org/wiki/Red_privada_virtual

Servidor VPN en contenedor Proxmox:

https://pve.proxmox.com/wiki/OpenVPN_in_LXC

Puerta de enlace: https://es.wikipedia.org/wiki/Puerta_de_enlace

Jitsi: <https://es.wikipedia.org/wiki/Jitsi>

OpenVPN: <https://www.digitalocean.com/community/tutorials/how-to-set-up-and-configure-an-openvpn-server-on-ubuntu-22-04>

Entidad certificadora: <https://www.digitalocean.com/community/tutorials/how-to-set-up-and-configure-a-certificate-authority-on-ubuntu-22-04>