



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Configuración de un servidor cloud como router

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Morocho Realpe, Eddy Patrik

Tutor/a: Pons Terol, Julio

CURSO ACADÉMICO: 2023/2024

Resumen

Hoy en día es bastante fácil contratar servidores cloud de bajos recursos a precios muy económicos e incluso gratuitos. Este tipo de servidores suelen limitar la instalación del sistema operativo a una lista reducida, donde aparecen en general distribuciones estándar de Linux, como por ejemplo Ubuntu. Sin embargo, no siempre es posible instalar sistemas operativos específicos como OpenWrt, ddWrt u otros como pfSense. En la asignatura Redes Corporativas se hacen prácticas con máquinas virtuales que tienen ddwrt instalado y una opción de futuro sería sustituir estas máquinas virtuales en ordenadores locales por servidores cloud gratuitos o de bajo coste.

En este Trabajo de Fin de Grado se pretende configurar un sistema operativo estándar como por ejemplo Ubuntu 22.04 para que se comporte con un router ddWrt o similar, permitiendo las funciones básicas de un router que podrán administrarse desde una interfaz web. Además, mediante el uso de VPN permitiría simular una red local con distintas máquinas que tuviesen acceso internet permitiendo por ejemplo combinarlas con máquinas virtuales locales.

Palabras clave: Nube, servidor, Ubuntu, router, máquinas virtuales, VPN

Abstract

Nowadays, It is quite easy to hire low-resource cloud servers at very economical prices, and even for free in some cases. These servers often limit the installation of the operating system to a reduced list, typically consisting of standard Linux distributions like Ubuntu. However, it is not always possible to install specific operating systems such as OpenWrt, ddWrt, or others like pfSense. In the Corporate Networks course, practical exercises are performed with virtual machines that have ddWrt installed, and a future option could be to replace these virtual machines on local computers with free or low-cost cloud servers.

This Bachelor's thesis aims to configure a standard operating system, such as Ubuntu 22.04, to behave like a ddWrt or similar router, allowing basic router functions to be managed through a web interface. Additionally, by using a VPN, it would simulate a local network with different machines having internet access, allowing, for example, their combination with local virtual machines.

Key words: Cloud, server, Ubuntu, router, virtual machines, VPN

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VIII
<hr/>	
1 Introducción	1
1.1 Objetivos	1
1.2 Impacto esperado	2
1.3 Estado del arte	4
1.3.1 Computación en la nube	4
1.3.2 Nube privada virtual (VPC)	5
1.3.3 Red virtual privada (VPN)	6
1.3.4 Crítica estado del arte	7
1.4 Estructura del documento	9
2 Componentes Tecnológicos	11
2.1 Red privada virtual (VPN ZeroTier y Wireguard)	11
2.1.1 VPN ZeroTier	11
2.1.2 VPN Wireguard	13
2.2 Servidor DHCP (Webmin)	14
2.3 Servidores en la nube (Amazon EC2)	15
2.4 Máquinas Virtuales (VMware Workstation)	17
3 Estructura de la red	19
3.1 Componentes de la red	20
3.1.1 Instancias Linux (EC2)	20
3.1.2 Máquinas virtuales linux (VMware)	27
3.2 Conexión de los componentes	29
3.2.1 Configuración del Servidor DHCP en Webmin	29
3.2.2 Creación de VPN ZeroTier	31
3.2.3 Arranque del servidor DHCP y conexión de las máquinas RCO	33
3.2.4 Conexión entre server1 y server2	35
3.3 Direccionamiento del tráfico	38
4 Funcionamiento de la red	41
4.1 Transferencia de datos entre máquinas.	41
4.2 Comprobación de la redirección del tráfico	43
5 Conclusiones y mejoras futuras	45
Bibliografía	49
<hr/>	
Apéndice	
A OBJETIVOS DE DESARROLLO SOSTENIBLE	51
A.1 Objetivos de desarrollo sostenible(ODS)	52

Índice de figuras

1.1	Proveedores Cloud Públicos más importantes.	5
2.1	VPN ZeroTier.	11
2.2	Wireguard.	13
2.3	Webmin.	14
2.4	Amazon EC2.	15
2.5	Icono de VMware.	17
3.1	Esquema de la red.	19
3.2	Lanzar instancia.	21
3.3	Selección de sistema operativo y tipo de instancia.	21
3.4	Selección de clave de inicio de sesión.	22
3.5	Detalles de la instancia server1.	22
3.6	Detalles de la instancia server2.	23
3.7	Reglas de entrada del grupo de seguridad.	23
3.8	Dirección DNS de server 2.	25
3.9	Página de inicio de Webmin server1.	25
3.10	Configuración Webmin.	26
3.11	Ajustes SSL.	26
3.12	Petición de certificado SSL.	27
3.13	Certificado SSL de server1.	27
3.14	Creamos máquinas virtuales.	28
3.15	Cambiamos adaptador para conectarnos mediante NAT.	28
3.16	Iniciar máquina virtual.	29
3.17	Módulo de DHCP Server.	30
3.18	Configuración de subred 1.	30
3.19	Configuración de subred 2.	30
3.20	Redes ZeroTier.	31
3.21	Autorización de conexión a la red ZeroTier para la red 1.	32
3.22	Asignamos IP a nuestra interfaz conectada a ZeroTier.	32
3.23	Cambiamos interfaz de escucha del servidor DHCP.	33
3.24	Conexión de las máquinas RCO con servidores.	33
3.25	Clientes activos en el server1 (arriba) y en el server2 (abajo).	34
3.26	Comprobación de conexión entre máquinas RCO y servidores. Ping desde RCO-1 a server1 (arriba). Ping desde RCO-2 a server2 (abajo)	35
3.27	Interfaces wg0 en server1(arriba) y server2(abajo).	36
3.28	Interfaz wg0 en server1.	36
3.29	Interfaz wg0 en server2.	36
3.30	Monitorización de las interfaces wg0.	37
4.1	Conexión entre RCO-1 y RCO-2.	41
4.2	Ping de RCO-1 a RCO-2.	42
4.3	Transferencia de archivo desde RCO-2 a RCO-1.	43
4.4	Verificación de transferencia.	43

4.5	Dirección IP del dominio.	43
4.6	Dirección IP desde RCO-1.	44
4.7	Dirección IP desde RCO-2.	44

Índice de tablas

3.1	Direcciones IP de los RCO	39
3.2	Direcciones IP de los servidores	39

CAPÍTULO 1

Introducción

El propósito fundamental de este documento es la presentación exhaustiva de dos redes privadas que se asemejan en su estructura y funcionalidad. Cada una de estas redes servirá de demostración en el desempeño de dos servidores hospedados en la nube, configurados para asumir el papel de routers. La función principal de estos servidores consistirá en orquestar el flujo de tráfico proveniente de los clientes conectados, creando, de esta manera, una red autónoma desvinculada de la red de Internet. Este procedimiento dará lugar a la formación de una red separada de internet que canalizará los datos a través de nuestras instancias en la nube, proporcionando así a la infraestructura una capa adicional de seguridad y control. Esta red se trata de una red mixta o híbrida donde convergerán partes en local y otra parte que estará alojada en la nube, con ello podemos mostrar una de las numerosas posibilidades que nos ofrece una tecnología como es la computación en la nube.

En las siguientes secciones, se desglosará detalladamente la tecnología empleada con este fin específico, se centrará también en los pasos que se han seguido para configurar cada componente de la red. Con esto se busca proporcionar una visión clara de los procesos y decisiones implicados en el despliegue de toda la infraestructura final. Para complementar este análisis teórico, se llevará a cabo una demostración práctica del funcionamiento de la red, a modo de tener una mejor comprensión del desempeño y eficacia en la práctica.

Finalmente, se realizará una breve recapitulación que condensará los aspectos clave abordados en este documento, brindando una conclusión breve pero completa que resume de manera efectiva el trabajo realizado y las implicaciones de los resultados obtenidos.

1.1 Objetivos

El propósito fundamental de este proyecto es implementar un sistema de enrutamiento de conexión a Internet a través de servidores en la nube, con el objetivo de beneficiarnos de su capacidad para mejorar la escalabilidad, disponibilidad y muchos otros puntos donde la computación en la nube es especialmente buena. Para este caso hemos optado por una red con dos routers que a parte de servir para redirigir el tráfico, alojarán un servidor DHCP (Protocolo de Configuración Dinámica de Hosts) cada uno con la finalidad de servir direcciones a los clientes (máquinas que se usan en la asignatura de Redes Corporativas o RCO) que se quieran conectar a nuestra red. Los elementos principales serán nuestros servidores en la nube y nuestras máquinas en local RCO.

Al coordinar la conexión a través del servidor en la nube, podemos aprovechar su alta disponibilidad y capacidad de escalabilidad para potenciar tanto el rendimiento como la confiabilidad de la red en su conjunto. Además, la implementación del servidor DHCP posibilita que los dispositivos conectados adquieran automáticamente sus direcciones IP y otra información de red crucial, simplificando en gran medida la configuración y el mantenimiento de la infraestructura de red.

Los dispositivos que se vinculen a los servidores lo harán a través de una red privada virtual (VPN), con el propósito de mantener una conexión segura y cifrada de todo el tráfico desde el cliente hasta el servidor. Este enfoque garantiza una conexión segura desde cualquier host y añade una capa adicional de seguridad a la red. A medida que se profundice en el proceso de configuración, destacaremos los pasos esenciales para establecer una conexión exitosa entre el servidor en la nube, el servidor DHCP y los dispositivos finales. Además, como tendremos dos routers para interconectar toda la red, estas se conectarán mediante otra solución VPN para una conexión segura entre estos como se verá más adelante.

En última instancia, procederemos a evaluar los resultados obtenidos a partir de esta implementación y a analizar las posibles ventajas y desafíos que podrían surgir al utilizar un enrutamiento de conexión a través del servidor en la nube junto con el servidor DHCP. Esta evaluación proporcionará una visión integral de cómo esta solución puede optimizar la gestión de la red y nos permitirá tomar decisiones fundamentadas para futuras mejoras y optimizaciones.

1.2 Impacto esperado

Con este trabajo esperamos conseguir ciertas mejoras con respecto al uso de un servidor convencional con el uso del alojamiento en la nube de nuestros servidores que se comportan como router. Por otro lado sería, el uso de medidas de seguridad que se pueden implementar para interconectar distintos dispositivos como en nuestro caso que se ha usado varias alternativas para ello. Hay ciertos puntos significativos en nuestra red que podríamos destacar, que serían favorables en este entorno. A continuación, los desglosaremos en los siguientes puntos:

1. **Mejora de la escalabilidad y flexibilidad de la red:** Un servidor en la nube tiene la ventaja de ser altamente escalable. Puede adaptarse fácilmente a las necesidades cambiantes de la red, permitiendo agregar o eliminar recursos según sea necesario. Esto brinda una mayor flexibilidad para ajustar la capacidad de enrutamiento de acuerdo con el crecimiento de la red y las demandas de tráfico. Además, el uso de una VPN en nuestro caso de Wireguard es una opción recomendada pues es una herramienta que ofrece un alto rendimiento y adaptabilidad en procesos de escalado. En nuestro caso usaremos dos servidores en la nube, pero sí quisiéramos escalar la red se podría hacer fácilmente añadiendo nuevos servidores y configurándolos debidamente, de manera que si hubiera una necesidad de cubrir la conexión de varios cientos o miles de clientes se podría escalar de manera sencilla.
2. **Mejora en la disponibilidad:** Los servidores alojados en la nube suelen estar respaldados por infraestructuras robustas y redundantes, lo que mejora la disponibilidad de la red. Esto se debe a que las empresas que ofrecen este tipo de servicios cuentan con grandes centros de datos que están dispersos y que proporcionan a los clientes varias regiones y zonas de disponibilidad donde poder desplegar sus servicios. Con esto nos aseguramos que los componentes alojados en la nube estarán operativos con un alto nivel de disponibilidad. Cada proveedor de cloud público tiene una

ciertos contratos de nivel de servicio (SLA) que aseguran a los clientes una gran disponibilidad. En resumen el uso de computación en la nube nos proporcionará una alta accesibilidad a la red en cualquier momento y la despreocupación de los fallos de tipo técnico pues de todo ello es gestionado por el proveedor.

3. **Facilidad de Administración:** La configuración y gestión de un servidor en la nube pueden realizarse de forma centralizada a través de una interfaz de administración basada en web. Esto simplifica las tareas de mantenimiento, monitorización y actualización de la red, lo que se traduce en una administración más eficiente y menos propensa a errores. La administración de los recursos en la nube se pueden gestionar mediante una interfaz en la web y suele ser mucho más sencilla que si se tratase de una infraestructura tradicional. En nuestro caso en el uso de Amazon Web Services, este nos ofrece una interfaz amigable para los usuarios y sencilla donde podemos crear, modificar, eliminar recursos desde cualquier sitio lo único necesario para ello será tener un dispositivo con conexión a internet, simplificando la gestión de manera significativa. Para la administración de las redes de nuestros servidores hemos optado por otra solución, para ello hemos usado Webmin que es una solución muy interesante para la administración y gestión de sistemas en red basados en Linux que nos ofrece para ello una interfaz web. El uso de ambas herramientas para la administración nos proporciona sencillez, reducción de fallos pues es menos propenso a ellos usando estas soluciones, nos centraliza la gestión y nos proporciona una manera para poder personalizar nuestros recursos de la manera que se desee.
4. **Mejoras en la seguridad:** Para la parte de seguridad podemos señalar varias medidas que se han usado para este fin. Por una parte, los proveedores de servicios en la nube suelen ofrecer medidas de seguridad avanzadas, como cortafuegos o cifrado de datos, para proteger los servidores y la información transmitida a través de la red. Esto ayuda a mejorar la seguridad general de la red y protegerla contra amenazas externas, para nuestra red hemos usado por ejemplo una nube privada virtual (VPC) que es un servicio que ofrece Amazon para crear una red aislada de las demás máquinas [1]. Otra medida que hemos implementado es el uso de redes privadas virtuales (VPN) para una conexión segura entre nuestros elementos de manera segura. Para las conexiones mediante VPN nos hemos apoyado en la herramientas ZeroTier y Wireguard como ya se mostrará más adelante, siendo dos opciones más modernas y distintas al que tienen un uso más común como son las soluciones IPsec u OpenVPN.
5. **Reducción de costes:** Al utilizar un servidor en la nube que simule un router, es posible eliminar la necesidad de hardware de enrutamiento dedicado, lo que puede resultar en una reducción de costos de adquisición y mantenimiento en este tipo de casos. Además, el modelo de pago por uso de muchos servicios en la nube permite una mayor optimización de los recursos financieros. Con esto y dado a su finalidad educativa, podemos mostrarlo como una alternativa al uso tradicional de hardware físico cuya inversión inicial es más significativa en comparación con este método. Aunque en este documento no nos centraremos en el estudio económico de estos dos tipos de despliegue, es importante comentarlo pues algo que se debe tener en cuenta en el uso de computación en la nube pues nos despreocupamos de todo tema técnico y de elegir hardware, pues todo es gestionado por el proveedor cloud.

A pesar de todos estos puntos favorables también es importante tener en cuenta que el impacto exacto dependerá de varios factores, como la implementación específica, la calidad del servicio de la nube que se quiera ofrecer y la capacidad técnica del personal

encargado de la administración de la red. A pesar de que el impacto depende de varios factores, en nuestro caso las herramientas y elementos seleccionados para este proyecto cumplen con los objetivos planteados, asegurando un despliegue de la red eficiente, seguro y que puede ser escalable.

1.3 Estado del arte

En la era digital actual, la infraestructura de red se enfrenta a desafíos cada vez más complejos, especialmente en entornos donde la conectividad segura y eficiente es fundamental. En este contexto, este estudio se centra en una solución innovadora que aprovecha la flexibilidad y la escalabilidad de Amazon Web Services (AWS) para facilitar el enrutamiento de tráfico entre máquinas virtuales VMware y redes externas. Esta implementación se basa en el uso de dos instancias en AWS, configuradas para establecer conexiones VPN tanto con las máquinas virtuales de VMware como con redes externas, utilizando tecnologías específicas: ZeroTier para la conexión de las máquinas virtuales. Esta estrategia ofrece una solución robusta y segura para la integración de entornos de infraestructura virtualizada con servicios en la nube, abriendo nuevas posibilidades para la gestión eficiente del tráfico y la colaboración interconectada en redes modernas ya sea para fines empresariales o educativos.

A continuación vamos a desglosar un poco las tecnologías que hemos usado para el montaje de la red.

1.3.1. Computación en la nube

Se ha optado por el uso de la computación en la nube para redirigir el tráfico debido a que será el elemento que tendrá que soportar la carga y el que se debería escalar en caso de que se necesitara. Al ser el punto a escalar esta solución será la idónea. En cuanto a la actualidad este tipo de soluciones de computación en la nube ha ido ganando popularidad durante estos años con respecto a la manera tradicional, que usa máquinas físicas. En primer lugar, este tipo de computación nos ofrece una escalabilidad incomparable, esto significa que las empresas pueden aumentar o disminuir rápidamente sus recursos informáticos según sus necesidades, sin la necesidad de invertir en hardware costoso y con la flexibilidad de adaptarse a picos de demanda o cambios estacionales en el negocio.

Por otro lado, la flexibilidad y la sencillez que proporciona la computación en la nube coge más importancia en el mundo actual, donde el trabajo remoto se ha vuelto cada vez más común. La capacidad de acceder a aplicaciones y datos desde cualquier lugar con acceso a internet, y en cualquier momento brinda a los usuarios la libertad de poder administrar este tipo de redes de manera eficiente desde casa, oficina o mientras estas de viaje pues solo es necesario una conexión a internet.

En términos de costos, la computación en la nube puede resultar significativamente más económica que mantener una infraestructura tradicional. Esto se debe a que este tipo de soluciones tienen un modo de pago por uso, con lo que solo se paga por el uso que se haga, eliminando la necesidad de invertir en hardware y software costosos que pueden volverse obsoletos en unos años.

La seguridad es otra ventaja importante a tener en cuenta pues muchos proveedores de servicios en la nube ofrecen medidas de seguridad avanzadas y cumplen con estándares de seguridad reconocidos a nivel mundial, lo que puede proporcionar tranquilidad a los usuarios o empresas preocupados por la protección de datos sensibles. Además, este tipo de servicios en la nube también suelen proporcionar opciones de respaldo y

recuperación de datos automáticos y redundantes, lo que garantiza la disponibilidad e integridad de los datos en caso de fallo del sistema o desastre.

Hoy en día, hay una acotada lista de proveedores de servicios en la nube que ofrecen soluciones para empresas y usuarios individuales. Algunos de los principales proveedores de computación en la nube de uso público incluyen:

1. **Amazon Web Services (AWS)**
2. **Microsoft Azure**
3. **Google Cloud Platform (GCP)**
4. **IBM Cloud**
5. **Oracle Cloud**



Figura 1.1: Proveedores Cloud Públicos más importantes.

En nuestro caso, usaremos el proveedor de cloud público de Amazon, más específicamente su servicio EC2 (Nube de Computo Elástico de Amazon), que es uno de los servicios más populares de Amazon Web Services (AWS). Este se trata de una plataforma de computación en la nube que permite alquilar servidores virtuales, denominados instancias. La elección de este proveedor se ha debido a que Amazon es uno de los mayores líderes en este sector y en principio da más opciones y más variedad con respecto a los otros proveedores en la nube ya mencionados.

1.3.2. Nube privada virtual (VPC)

En comparación con una red en local, una red en la nube si se alojan las máquinas en una infraestructura de este tipo sin una medida como esta, estarían todas las instancias en una misma red dando lugar a muchos problemas de red. Como solución a ello los proveedores de cloud ofrecen una solución de nube privada virtual, que como se deduce es una red virtual dentro de la nube que esta aislada de las demás máquinas que estén dentro. En nuestro caso usaremos la herramienta que nos proporciona Amazon para este servicio es Virtual Private Cloud (VPC) que nos ofrece un aislamiento en la nube dentro de la red de Amazon. Resumiendo las ventajas de este servicio para la configuración de nuestras instancias en la nube podemos destacar ciertos puntos:

- **Aislamiento:** Te permite crear redes virtuales privadas aisladas en la nube de AWS, proporcionándonos un entorno seguro y controlado para nuestros recursos. También, podemos crear subredes en esta y con ello dividir la VPC en subredes públicas y privadas, permitiendo segmentar los recursos de manera lógica y de acuerdo a las necesidades que se deseen cubrir.
- **Escalabilidad:** Si fuera necesario, en nuestro caso no es de relevancia, pero esta herramienta permite el escalamiento de los sistemas pues se puede adaptar a un

incremento de instancias según las necesidades, además que dentro de estas “nubes privadas” aisladas se pueden crear otras subredes permitiendo bastante flexibilidad al usuario. Además, AWS nos proporciona herramientas para balancear la carga y así distribuir el tráfico de entrada entre múltiples instancias dentro de la VPC.

- **Seguridad:** Nos proporciona una lista de control de accesos de red y grupos de seguridad para gestionar el tráfico entrante y saliente de las instancias.

En la red que se va a desplegar este punto cobra importancia a la hora de crear nuestros servidores, ya que esta tecnología nos proporcionará un entorno seguro donde crear las instancias que redirigirán el tráfico de red.

1.3.3. Red virtual privada (VPN)

Una VPN (Red Privada Virtual) es una herramienta fundamental que proporciona una conexión segura y privada a través de una red pública, como Internet. Al utilizar una VPN, el tráfico de Internet se cifra y se redirige a través de un servidor remoto, lo que brinda a quién lo use una serie de ventajas clave:

- **Ocultar la dirección IP:** La dirección IP del usuario es una identificación única que puede revelar tu ubicación y actividad en línea. Al ocultar tu dirección IP mediante VPN, tu actividad en Internet se vuelve anónima, lo que dificulta que terceros rastreen tu ubicación real o accedan a tu información personal.
- **Encriptar tu tráfico:** La VPN utiliza protocolos de encriptación para cifrar tu tráfico de Internet. Esto significa que incluso si alguien intercepta tus datos, como tu proveedor de servicios de Internet (ISP) o hackers, no podrán leer la información, lo que garantiza la privacidad y la seguridad de tus comunicaciones en línea.
- **Acceder a contenido restringido:** Muchas plataformas y sitios web restringen su contenido según la ubicación geográfica del usuario. Con una VPN, puedes conectarte a un servidor en otro país y simular estar físicamente en esa ubicación. Esto te permite eludir las restricciones geográficas y acceder al contenido que de otro modo estaría bloqueado en tu ubicación actual.
- **Proteger tu privacidad:** Una VPN actúa como una barrera entre tu actividad en línea y los posibles observadores, como puede ser un proveedor de servicio de internet (ISP), el gobierno, hackers y otras entidades. Al cifrar tu tráfico y ocultar tu dirección IP, una VPN protege tu privacidad y te brinda tranquilidad al navegar por Internet, especialmente en redes Wi-Fi públicas o en países con restricciones de censura en línea.

En nuestro caso nos proporcionará una conexión a nuestra red privada en la nube (servidores en AWS) desde las instancias locales que tengamos, con ello nos aseguramos de que todo el tráfico esté encriptado asegurándonos de una conexión segura.

La solución, de todas las opciones disponibles, que se ha elegido es ZeroTier[2]. Esta es una red privada virtual de código abierto y gratuita que te permite conectar dispositivos de forma segura a través de internet. Es una alternativa a las VPN tradicionales. Trabaja de manera que conecta los dispositivos directamente entre sí, sin tener que pasar por un servidor centralizado. Con ello damos a entender que su arquitectura es de tipo peer-to-peer (P2P)[3]. Además ZeroTier nos proporciona una red virtual de nivel 2 por lo que es perfecta para reemplazar la funcionalidad de una red local.

1.3.4. Crítica estado del arte

La finalidad de este trabajo es redirigir el tráfico de máquinas virtuales a través de servidores en la nube. Para el despliegue de nuestra red, hemos utilizado diversas tecnologías como VPN y proveedores de nube pública. Relacionando este trabajo con otros proyectos que han empleado estas tecnologías, podemos destacar varios ejemplos relevantes de los que hemos obtenido un "feedback".

Relacionado al uso de VPN, podemos destacar un trabajo donde se emplea un servidor VPN para conectarse de manera segura a un NAS [4]. En este proyecto, que se ha tomado como referencia, se ha utilizado una VPN IPSec/IKEv2 [5], un protocolo VPN que proporciona autenticación y cifrado a nivel de red. En ese caso, se utilizó para conectarse de manera segura a un NAS/multimedia, siendo IPSec una opción adecuada debido a la necesidad de estabilidad y seguridad para los dispositivos.

Hoy en día, existen muchas herramientas para establecer una conexión segura. En nuestro caso, se han optado por dos opciones diferentes: Zerotier y Wireguard.

A modo de comparación, IPSec es una opción segura y fiable que se ha utilizado durante varios años. Sin embargo, para este proyecto, hemos optado por herramientas más modernas como Zerotier [6], que lleva en funcionamiento desde 2011, y Wireguard, que ha estado en uso desde 2015. Podemos destacar distintas ventajas de estas opciones:

- De Zerotier destacamos que es adecuada para usuarios que necesitan una solución de red definida por software que sea sencilla de usar y gestionar en redes distribuidas, evitando la complejidad de la implementación de IPSec para personas sin amplio conocimiento en este campo de seguridad. En otros trabajos de fin de carrera podemos ver también el uso de esta solución en la creación de un nodo multi-VPN en la nube para el ámbito empresarial [7]
- De Wireguard destacamos que esta solución promete ser una solución innovadora para clientes que buscan alto rendimiento y una configuración sencilla, beneficiándose de la criptografía moderna para sus conexiones.

A la hora de elegir cómo establecer conexiones seguras, es importante entender las necesidades específicas, ya que tanto una opción como otra son aceptables, pues su finalidad es proporcionar seguridad a quienes las utilicen.

El siguiente punto a tratar será la revisión de otros trabajos que se han beneficiado de proveedores en la nube para su desarrollo. Primero, podemos tomar como referencia un estudio interesante sobre la comparación de diferentes proveedores en la nube [8]. En este documento, se analiza la comparación entre varios proveedores de servicios en la nube y las opciones más destacadas que existen actualmente. Aunque el trabajo sea del 2018, la situación no ha cambiado significativamente, ya que las mismas empresas siguen dominando el mercado.

En nuestro caso, se consideraron varias opciones para la creación del proyecto, como Oracle Cloud, Google Cloud Platform (GCP) y Amazon Web Services (AWS), siendo esta última la opción elegida. La elección de AWS se debe a que ofrece una amplia gama de servicios en la nube y es la empresa más robusta y con más experiencia en este tipo de servicios, ya que fue pionera en ofrecer servicios cloud desde 2003. Además, su servicio de computación EC2, que se ha utilizado para las instancias, es sencillo y nos ofrece una gestión completa de las máquinas que creemos.

Aunque hayamos utilizado Amazon, también es importante mencionar que, si hubiéramos usado los servicios de Google (GCP) o Microsoft (Azure), entre otros, existen muchos casos en los que la elección de otro proveedor puede ser más adecuada. Como

hemos observado en otros trabajos realizados por compañeros de la Universidad Politécnica de Valencia, es posible crear sistemas utilizando otros servicios. Un ejemplo de esto es el despliegue de un clúster de Kubernetes altamente disponible en Alibaba Cloud [9], una empresa que forma parte del Alibaba Group, líder indiscutible en el mercado chino.

En este contexto, podemos ver cómo es posible emplear otro tipo de servicio, en este caso un servicio de Kubernetes. Esto no significa que Amazon no ofrezca una opción para Kubernetes, sino que se optó por utilizar Alibaba Cloud debido a sus características específicas. Además, cabe destacar que, aunque Alibaba no tenga tanta presencia fuera del mercado asiático en cuanto a servicios cloud, se distingue por ofrecer precios bastante económicos y ser adaptable a una amplia variedad de necesidades.

Concluyendo con este apartado, se ha querido mostrar como en este proyecto nos hemos realimentado de trabajos ya realizados con este tipo de temas que tienen gran importancia para la creación de los recursos necesarios de nuestra red y sus conexiones. También destacamos que actualmente hay un gran abanico de posibilidades con lo relacionado a sistemas de seguridad y de servicios en la nube.

1.4 Estructura del documento

La planificación de este trabajo se dividirá en varios puntos. Primero comentaremos detalladamente las tecnologías que se han usado, además comentaremos brevemente el porqué de nuestras elecciones en cada elemento de nuestro proyecto, ya que se verá en el desarrollo que cada herramienta usada ya sea para la creación de máquinas virtuales en local o en la nube o el uso de conexiones seguras tiene varias alternativas con la misma finalidad. Luego, detallaremos los pasos que se han seguido para crear cada elemento y comentaremos las características que tienen.

Seguidamente se mostrará la configuración de los servidores que ofrecerán servicio a las máquinas virtuales y como se han interconectado los elementos entre sí. Después, se realizará la configuración del direccionamiento de los servidores en la nube para que se comporten como routers redirigiendo el tráfico.

Una vez todo conectado se procederá a probar el funcionamiento de nuestra red, para ello haremos varias pruebas. Las pruebas que se han realizado probarán tanto el funcionamiento de las conexiones entre clientes como la redirección del tráfico para comprobar que se cumple el objetivo principal.

Por último, realizaremos una conclusión del trabajo realizado y comentaremos las mejoras que hemos pensado para nuestra red a futuro.

CAPÍTULO 2

Componentes Tecnológicos

A continuación, expondremos en detalle los elementos que se han empleado en la configuración de la infraestructura de red. En este análisis, describiremos los componentes utilizados y su función en la red, abordando de manera precisa cada uno de los aspectos relevantes de la configuración.

2.1 Red privada virtual (VPN ZeroTier y Wireguard)

Como veremos en los siguientes puntos, las conexiones entre los distintos hosts dentro de nuestra red, se han realizado mediante el uso de VPN para obtener una capa de seguridad donde todo el tráfico vaya cifrado. A continuación, se comentarán las dos opciones que hemos implementado en cada caso. Los dos casos son las conexiones entre las maquinas RCO y los servidores alojados en la nube (mediante ZeroTier); y la conexión entre los servidores en la nube (mediante Wireguard).

2.1.1. VPN ZeroTier



Figura 2.1: VPN ZeroTier.

ZeroTier es una herramienta para conectar dispositivos a través de su propia red privada en cualquier parte del mundo. Para ello, crea una red y luego une dos o más dispositivos a esa red. Puede utilizarse ZeroTier para jugar, conectarse a recursos empresariales remotos o incluso como backplane en la nube para empresas, este término se refiere a una infraestructura en la red que conecta y facilita la comunicación entre diferentes componentes o servicios dentro de un entorno en la nube. Esta red utiliza un conjunto de servidores y controladores de red distribuidos globalmente para negociar automáticamente conexiones par a par (P2P) para los dispositivos y aprovisionar las redes que queramos. En nuestro caso nos serviremos de esta solución para conectar los dispositivos clientes,

que serán las máquinas virtuales, para tener una conexión segura entre las instancias en la nube y las máquinas virtuales en local.

Funcionamiento de ZeroTier

ZeroTier es un hipervisor de red distribuido construido sobre una red global de igual a igual criptográficamente segura. Proporciona capacidades avanzadas de virtualización y administración de redes a la par de un conmutador de SDN (redes definidas por software) empresarial, pero a través de redes de área local y amplia y conectando casi cualquier tipo de aplicación o dispositivo.

Esto se logra combinando una red de igual a igual segura y direccionada criptográficamente (denominada VL1) con una capa de emulación de Ethernet algo similar a VX-LAN (denominada VL2). La capa de virtualización Ethernet VL2 incluye funciones SDN empresariales avanzadas, como reglas de control de acceso detalladas para la microsegmentación de la red y la monitorización de seguridad.

Con respecto al protocolo que se sigue en esta red todo el tráfico está cifrado de extremo a extremo mediante claves secretas que nosotros podemos controlar. Todos en la red están controlados por dos tipos de identificadores: direcciones ZeroTier de 40 bits/10 dígitos e ID de red de 64 bits/16 dígitos. Estos se distinguen fácilmente por su longitud. Una dirección ZeroTier identifica un nodo o "dispositivo"(ordenador, teléfono, servidor, etc.) mientras que en una ID de red identifica una red Ethernet virtual a la que se pueden unir dispositivos. Las direcciones de ZeroTier se podrían considerar como números de puerto en un enorme conmutador inteligente Ethernet que admite VLAN (red de área local virtual). Por otro lado, los ID de red son ID de VLAN a los que se pueden asignar estos puertos. Se puede asignar un solo puerto a más de una VLAN. Tocando un poco la parte de seguridad enfocándonos en la parte de los datos encriptados de la conexión podemos comentar que en toda conexión los paquetes están cifrados y no pueden ser leídos por los root ni por nadie más, se usa para ello criptografía moderna de 256 bits en las formas recomendadas. El cifrado de clave pública asimétrica es Curve25519/Ed25519[10], una variante de curva elíptica de 256 bits.

Definido un poco como funciona esta tecnología, en nuestra red se usará para la conexión entre las instancias en AWS y nuestras máquinas virtuales RCO. En cuanto a la creación de las redes y de como se han conectado los dispositivos entre sí, se detallará más adelante en los siguientes puntos.

En resumen, la elección de ZeroTier es porque es una VPN que utiliza una arquitectura distribuida, cifrado sólido y autenticación de alto nivel para proporcionar conectividad segura y eficiente a través de una red global. Su diseño de capas y enfoque en la seguridad la convierten en una elección sólida para empresas y usuarios que buscan privacidad y confiabilidad en sus comunicaciones en línea, ajustándose bien a nuestra red.

2.1.2. VPN Wireguard



Figura 2.2: Wireguard.

Wireguard es una aplicación de código abierto y software libre, así como un protocolo de comunicación. Implementa técnicas de redes privadas virtuales (VPN) para establecer conexiones seguras punto a punto en configuraciones enrutadas o puenteadas. Se centra en la simplicidad y la seguridad, y está diseñado para ser más rápido, más fácil de configurar y más eficiente que los protocolos VPN tradicionales como OpenVPN[11] o IPsec[12]. Además, tiene la ventaja de que se puede implementar en muy pocas líneas de código y su auditoría para detectar vulnerabilidades de seguridad es más sencilla, ya que en principio sería suficiente con una persona para revisarlo de manera exhaustiva en comparación con las otras soluciones VPN alternativas.

En comparación con ZeroTier, esta herramienta nos ofrece un conjunto de características y beneficios que lo convierten en una opción más adecuada para las necesidades específicas de nuestro proyecto. La combinación de seguridad robusta, alto rendimiento, facilidad de implementación y compatibilidad nos permite establecer una conexión VPN confiable y segura entre los servidores, reforzando así la protección general de la red.

Funcionamiento de Wireguard

Tratando un poco el tema técnico de Wireguard, este funciona agregando una interfaz de red (o varias), como `eth0` o `wlan0`, llamada `wg0` o similar. Esta interfaz de red se puede configurar normalmente usando `ifconfig` o `ip-address`, con rutas agregadas o eliminadas usando `route` o `ip-route`, y así para cualquier utilidad de red común que se necesite. Esta interfaz (`wg0`) actúa como una interfaz túnel.

Por otro lado, el enrutamiento que se hace es de clave criptográfica que funciona asociando claves públicas con una lista de direcciones IP del túnel permitidas dentro del túnel. Cada interfaz de red tiene una clave privada y una lista de pares. Cada par tiene una clave pública, estas son breves y simples, y los pares las utilizan para autenticarse entre sí. Al enviar paquetes, la lista de IP permitidas se comporta como una especie de tabla de enrutamiento, y al recibir paquetes, la lista de IP permitidas se comporta como una especie de lista de control de acceso. Este concepto dentro de Wireguard se le denomina "Cryptokey Routing Table", algo así como una tabla de enrutamiento de claves encriptadas. Se puede usar cualquier combinación de IPv4 e IPv6, para cualquiera de los campos pues es capaz de encapsular uno dentro del otro si es necesario.

En resumen, Wireguard es una solución VPN moderna y eficiente que combina simplicidad, seguridad y rendimiento. Por ello hemos optado para esta conexión esta herramienta; además de que se ha podido probar otra opción distinta a las que se suelen usar comúnmente.

2.2 Servidor DHCP (Webmin)



Figura 2.3: Webmin.

Webmin se destaca como una herramienta interesante para administradores de sistemas y profesionales de tecnología que buscan gestionar servidores de manera efectiva mediante una interfaz gráfica basada en web. Al simplificar la complejidad característica de la administración del sistema, Webmin potencia la capacidad de los usuarios, incluso a aquellos sin experiencia en la línea de comandos, para realizar tareas esenciales de configuración y gestión. Su enfoque en la accesibilidad, combinado con su capacidad de extensión, lo convierte en una opción popular y confiable para simplificar la gestión de servidores. Webmin nos ofrece varias utilidades a modo de resumen estas serían las que destacamos:

- **Configuración del sistema:** Webmin permite configurar una variedad de aspectos del sistema, como la red, los usuarios, los grupos, los servicios y el software instalado. Desde la interfaz de Webmin, los administradores pueden ajustar la configuración del sistema de manera intuitiva y eficiente, lo que facilita la administración de servidores y estaciones de trabajo.
- **Administración de archivos:** Con Webmin, los usuarios pueden realizar tareas comunes de administración de archivos, como crear, editar, eliminar y mover archivos y directorios. Esta funcionalidad es especialmente útil para gestionar el contenido de los servidores y asegurar que los archivos estén organizados de manera eficiente.
- **Gestión de paquetes:** Webmin simplifica el proceso de gestión de paquetes de software, permitiendo a los administradores instalar, actualizar y eliminar paquetes de software de manera centralizada. Esta funcionalidad es fundamental para mantener los sistemas actualizados y seguros, ya que facilita la instalación de parches y actualizaciones de software.
- **Copia de seguridad y restauración:** Una de las características más importantes de Webmin es su capacidad para realizar copias de seguridad y restaurar sistemas. Los administradores pueden programar copias de seguridad automáticas, configurar políticas de retención de datos y restaurar sistemas desde puntos de restauración previamente creados. Esto garantiza la integridad y disponibilidad de los datos en caso de fallos del sistema o pérdida de datos.
- **Monitorización del sistema:** Webmin proporciona herramientas integrales de monitorización del sistema que permiten a los administradores supervisar el rendimiento del sistema y analizar registros de eventos. Esta funcionalidad es fundamental para detectar problemas de rendimiento, identificar cuellos de botella y

mantener la salud general del sistema. Los administradores pueden configurar alertas personalizadas para recibir notificaciones sobre eventos importantes, lo que les permite tomar medidas pro-activas para evitar problemas.

Debido a que se puede acceder a través de un navegador web, podemos iniciar sesión a Webmin desde cualquier sistema conectado a la red. No hay diferencia entre ejecutarlo localmente o remotamente, además es mucho más fácil de usar en la red que otros programas de configuración gráfica. Esta herramienta tiene un diseño modular donde cada función está contenida en un módulo de manera que se puede instalar o quitar de forma independiente del resto del programa. Cada módulo gestiona algún servicio o servidor, como los usuarios Unix, el servidor web Apache o los paquetes de software.

Por otro lado, como algo a tener en cuenta y que es interesante conocer es que si se ha configurado manualmente el sistema, Webmin reconoce todas las configuraciones existentes. Webmin lee los archivos de configuración estándar en sus sistema y los actualiza directamente en lugar de usar su propia base de datos. Esto significa que puedes mezclar libremente Webmin, configuración manual y otros programas o scripts que funcionan de la misma manera.

Ahora bien, la finalidad por la cual usamos Webmin es la configuración de un servidor DHCP[13]. Esto como todas las funciones se hará mediante un módulo de servidor DHCP. Antes de comentar como funciona este módulo haremos una breve introducción al protocolo de configuración dinámica de host.

DCHP es un protocolo que permite que los hosts soliciten y se les asigne una dirección IP en una red de área local (LAN). Se utiliza para simplificar el proceso de asignación de direcciones IP, ya que solo un servidor puede administrar las direcciones de varios clientes. También es útil para sistemas como computadoras portátiles que se mueven en múltiples redes, ya que no necesitan reconfigurarse para cada LAN. Cuando un servidor asigna una IP a un cliente, se le otorga un contrato de arrendamiento de esa dirección por un cierto período de tiempo, durante el cual a ningún otro cliente se le asignará la misma dirección. Cuando vence el contrato de arrendamiento, el cliente debe volver a ponerse en contacto con el servidor. Por lo general, se le asignará la misma dirección IP que antes y el arrendamiento se extenderá por el mismo período de tiempo. Si un cliente no se comunica con el servidor cuando finaliza su arrendamiento, el servidor asume que el cliente se ha apagado y marca la dirección como disponible para asignarla a otros hosts.

El módulo para configurar el servidor DHCP se puede usar para configurar su sistema para que los clientes de una LAN se les puedan asignar automáticamente direcciones IP, servidores DNS y otra información. Si ya hay un servidor en su red, configurar otro es una mala idea, ya que pueden interferir entre sí.

2.3 Servidores en la nube (Amazon EC2)



Figura 2.4: Amazon EC2.

1. **Flexibilidad:** La gama diversa de tipos de instancias que ofrece EC2 permite a los usuarios seleccionar la configuración óptima para sus aplicaciones. Desde instan-

cias con alta capacidad de CPU hasta instancias optimizadas para almacenamiento, EC2 proporciona la flexibilidad necesaria para satisfacer una amplia variedad de necesidades computacionales.

2. **Escalabilidad:** La capacidad de escalar vertical u horizontalmente en EC2 es fundamental para manejar cargas de trabajo dinámicas. Esta capacidad permite a los usuarios ajustar rápidamente los recursos disponibles para satisfacer la demanda de sus aplicaciones, lo que resulta en una mayor agilidad a la hora de operar y una mayor capacidad de manejar picos de tráfico sin interrupciones.
3. **Modelo de pagos por consumo:** El modelo de precios de pago por uso de EC2 significa que los usuarios solo pagan por los recursos que realmente utilizan. Esto proporciona una flexibilidad financiera significativa al poder evitar costos fijos asociados con la infraestructura física tradicional. Además, nos permite ajustar el costo de la infraestructura dependiendo de nuestra economía ya seamos una persona que quiera probar los servicios o una empresa que tenga un recursos limitados.
4. **Amplia selección de imágenes preconfiguradas:** Las imágenes de máquinas virtuales preconfiguradas, o AMIs (término que se les acuña en AWS), simplifican el proceso de implementación al proporcionar una base lista para usar. Con una variedad de sistemas operativos y software preinstalados disponibles, los usuarios pueden comenzar a trabajar rápidamente sin la necesidad de realizar configuraciones extensas.
5. **Opciones de almacenamiento flexible:** EC2 ofrece una variedad de opciones de almacenamiento que se adaptan a diferentes requisitos de rendimiento y durabilidad. Desde almacenamiento local en la instancia hasta almacenamiento en bloque (EBS) y almacenamiento de objetos en S3, los usuarios tienen la capacidad de elegir la solución que mejor se adapte a sus necesidades específicas.
6. **Seguridad y redes:** Con características como grupos de seguridad y configuración de redes virtuales(VPC), EC2 garantiza un entorno seguro para las instancias en la nube. Los grupos de seguridad actúan como firewalls virtuales, controlando el tráfico de red y protegiendo las instancias de posibles amenazas.
7. **Acceso y control total:** Los usuarios tienen un control completo sobre sus instancias EC2, lo que permite personalizar configuraciones, instalar software y administrar sistemas operativos según sea necesario. Esta capacidad brinda a los usuarios un nivel de flexibilidad y control similar al que tendrían con servidores físicos tradicionales.

Amazon EC2 es una herramienta versátil y adaptable que se utiliza en una variedad de escenarios, desde alojar aplicaciones web hasta ejecutar cargas de trabajo intensivas en procesamiento, análisis de datos, pruebas y desarrollo de software. Su flexibilidad y escalabilidad lo convierten en un componente esencial en la infraestructura de muchas empresas y organizaciones que buscan aprovechar los beneficios de la computación en la nube.

En nuestro caso particular, optamos por utilizar instancias sencillas de EC2, como las t2.micro, que ofrecen 1 virtual CPU y 1 GB de memoria. Estas instancias son perfectas para nuestras necesidades, ya que nuestra red no requiere de grandes recursos para su funcionamiento. Serán estas máquinas las que configuraremos para actuar como routers, dirigiendo el tráfico entre nuestras máquinas virtuales, que serán los hosts clientes cuyo tráfico queremos redirigir. Las denominaremos **server1** y **server2**.

Crear y configurar estas instancias será fundamental para establecer la infraestructura necesaria para nuestro proyecto. A través de los siguientes puntos, detallaremos el proceso de creación y configuración de estas instancias, asegurándonos de que estén listas para manejar el tráfico de nuestra red de manera eficiente y segura.

2.4 Máquinas Virtuales (VMware Workstation)

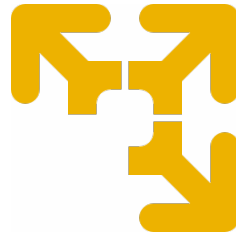


Figura 2.5: Icono de VMware.

VMware Workstation es un software de virtualización desarrollado por VMware, Inc. que permite crear y gestionar máquinas virtuales en un sistema operativo anfitrión. Una máquina virtual es un entorno informático emulado que puede ejecutar un sistema operativo completo junto con aplicaciones y programas, todo ello aislado del sistema operativo principal del host.

VMware Workstation permite a los usuarios crear y ejecutar múltiples máquinas virtuales en una sola máquina física. Esto es especialmente útil para tareas como desarrollo de software, pruebas de aplicaciones, configuración de sistemas, capacitación en diferentes entornos, entre otros casos. Cada máquina virtual creada con VMware Workstation funciona como una entidad independiente, con su propio sistema operativo, recursos de hardware virtuales y configuración.

Algunas características notables de VMware Workstation incluyen:

1. **Captura de instantáneas:** Los usuarios pueden tomar instantáneas del estado actual de una máquina virtual y volver a ese estado en cualquier momento. Esto es útil para realizar pruebas y experimentos sin preocuparse por que la configuración existente dañe la máquina.
2. **Clonación:** Permite crear copias idénticas de máquinas virtuales, lo que es útil para replicar entornos o distribuir configuraciones específicas.
3. **Redes virtuales personalizadas:** VMware Workstation permite configurar redes virtuales dentro de las máquinas virtuales, lo que permite simular una variedad de entornos de red para pruebas y desarrollo.
4. **Compatibilidad con múltiples sistemas operativos:** Puedes crear máquinas virtuales con diferentes sistemas operativos, como Windows, Linux, macOS, y otros, todo en la misma máquina física anfitrión.
5. **Aislamiento y seguridad:** Las máquinas virtuales se ejecutan de forma aislada del sistema operativo principal, lo que ayuda a prevenir conflictos y problemas de seguridad entre diferentes entornos.
6. **Migración y transferencia:** VMware Workstation permite mover máquinas virtuales entre diferentes hosts, lo que facilita la migración y la colaboración en entornos de desarrollo y pruebas.

En resumen, VMware Workstation es una herramienta poderosa para la virtualización que permite a los usuarios crear, administrar y ejecutar múltiples sistemas operativos y aplicaciones en una sola máquina física, lo que resulta útil en una amplia gama de situaciones profesionales y de desarrollo.

Para este proyecto hemos usado esta herramienta para la creación de nuestros hosts que harán de clientes y que serán los que se conecten a los servidores en la nube. Luego con ellas haremos las pruebas para probar el funcionamiento de la red y que el tráfico se enruta correctamente desde las máquinas a internet mediante los servidores. Las llamaremos **RCO1** y **RCO2**; mostraremos como las creamos en los siguientes puntos.

CAPÍTULO 3

Estructura de la red

Ahora se detallará la estructura de nuestra red. Primero que nada listaremos todos los componentes:

- **Máquinas:** dos instancias en AWS que serán los router (server1 y server2) y dos máquinas virtuales (RCO1 y RCO2) en local con VMware Workstation que serán los hosts clientes de nuestra red.
- **Conexiones:** Por un lado, tendremos dos conexiones con ZeroTier desde los clientes a los routers; estas conexiones estarán conectadas a dos redes independientes que se denominarán red-tfg1 y red-tfg2. Por otro parte, estará la conexión entre los servidores AWS que se hará también mediante una solución VPN, en este caso se usará Wireguard que es otra opción para conectar de manera segura dos máquinas.

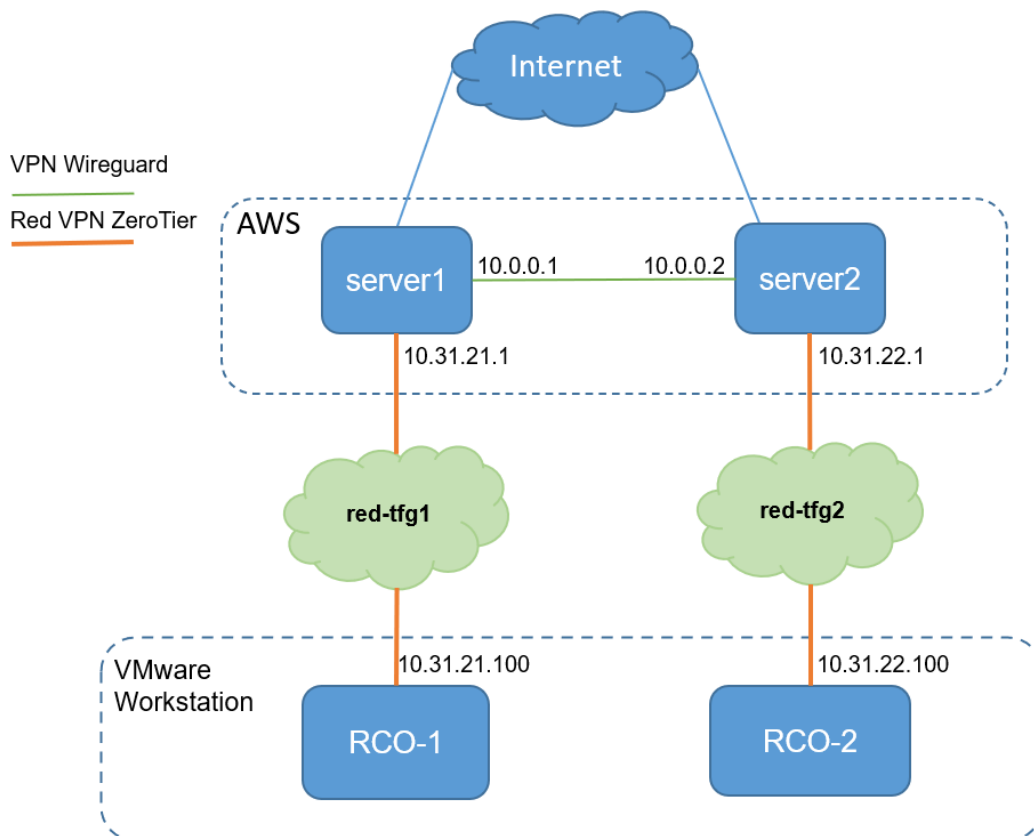


Figura 3.1: Esquema de la red.

Los servidores `server1` y `server2` tendrán alojado un servidor DHCP de manera que los clientes en nuestro caso las máquinas `RCO-1` y `RCO-2` obtendrán una dirección servida por su respectivo servidor, cada servidor tendrá una subred configurada para proporcionar una serie de direcciones (como ya se verá en los siguientes puntos). La conexión entre estos se hará mediante una conexión a una red ZeroTier.

A primera vista, tras la descripción de los componentes y conexiones, la imagen de la figura 3.1 muestra como será nuestra red de manera gráfica, donde se muestra donde se sitúan las máquinas y como se conectan entre sí. Podríamos decir que se compone de dos redes similares, una red sería la red 1 que se compone de `server1` y `RCO-1` que están conectados mediante VPN, `red-tfg1`, y la red 2 compuesta por `server2` y `RCO-2` conectadas también mediante VPN, `red-tfg2`.

3.1 Componentes de la red

En esta sección vamos a comentar como hemos creado cada host de nuestra red, para ello usaremos dos herramientas, una será la herramienta EC2 de Amazon donde crearemos en la nube dos instancias, que serán nuestros servidores, y la otra será VMware Workstation Player que será donde crearemos dos máquinas virtuales que harán de hosts en nuestra red y se conectarán a los servidores.

3.1.1. Instancias Linux (EC2)

Para usar la herramienta basta con registrarse en la página de Amazon Web Services y verificar cuenta; una vez registrado correctamente, ya tendremos acceso a todo servicio que nos proporcione Amazon Web Services. Tras eso procederemos a crear nuestras dos instancias Ubuntu.

Las dos instancias tienen características similares pues no hace falta para nuestro caso que se diferencien una de otra. Con ello las propiedades de cada una son:

- Sistema operativo Ubuntu Server 22.04 LTS.
- El tipo de instancia será micro, con 1 CPU virtual y 1GiB de memoria.
- Con un almacenamiento de 8 GiB (valor predeterminado).

Con las características definidas entraremos en la plataforma de AWS seleccionaremos el servicio EC2. Una vez en este servicio, nos aparecerá un panel de control donde aparecen las instancias que creemos que estén ejecución. Seleccionamos "Lanzar instancia" para crear una instancia, como se puede ver en la figura 3.2.

Nota: Este procedimiento se realizará dos veces, una para cada máquina.



Figura 3.2: Lanzar instancia.

En pantalla de creación de instancia elegiremos el sistema operativo deseamos, en nuestro caso vamos a elegir **Ubuntu** y el tipo de instancia elegimos **t2.micro**, el tipo de instancia elegido es un de los muchos tipos de instancia que se puede elegir, estas suelen ser generalmente equilibradas para nuestro caso este tipo tiene la característica de contar con 1 CPU virtual y 1 GB de memoria.

El almacenamiento será de 8 GB (valor predeterminado), igualmente si se hubiera necesitado ampliar ese valor se podría cambiar en cualquier momento. En nuestro caso se deja ese valor pues no es necesario proveernos de tanta memoria para su finalidad.

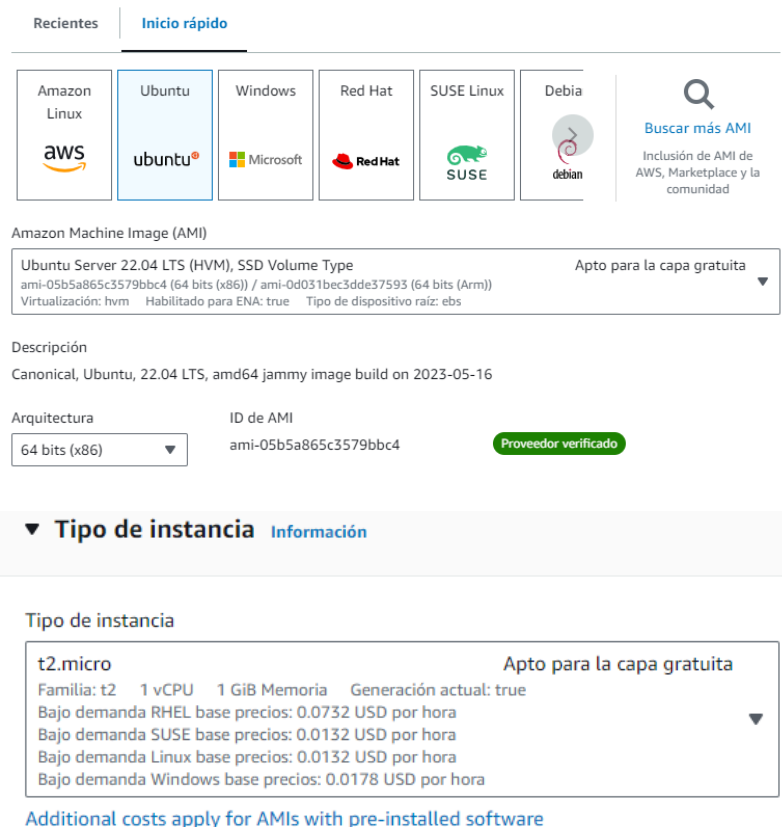


Figura 3.3: Selección de sistema operativo y tipo de instancia.

Después, otra opción de seguridad es la creación el uso de un par de claves para iniciar sesión. Se han creado un par de claves en un archivo (se ha nombrado como *claves*), este será necesario para conectarnos mediante ssh a las máquinas, ssh es un protocolo que nos ofrece una manera segura de conectarnos y administrar de manera segura una

máquina en remoto. Esto es muy útil pues así no necesitaremos entrar a la plataforma de Amazon para entrar a los servidores. Con esto evitamos que cualquiera pueda conectarse a nuestras máquinas, necesitando el archivo con la clave para entrar a ellas mediante ssh.

Como van a ser iguales los dos servidores, no vemos necesario la creación de otro par de claves, con lo que se ha optado por usar el mismo par de claves 3.4 para entrar tanto en una máquina como en la otra, aunque si lo necesitáramos podríamos crear distintas para cada servidor.

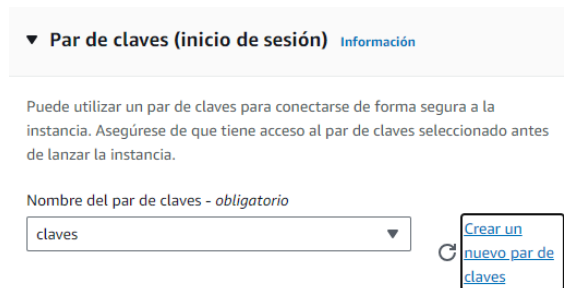


Figura 3.4: Selección de clave de inicio de sesión.

Con respecto a la nube privada virtual (VPC) usaremos en ambos casos la misma, dejaremos la selección por defecto, para que ambas estén en la misma red en la nube. Y para el grupo de seguridad crearemos uno para cada uno. Luego se pulsa en "Lanzar instancia" para crear e inicializar la instancia, después de lanzar las instancias tardará unos segundos o minutos como mucho en cambiar su estado a 'En ejecución', como se puede observar en las imágenes 3.5 y 3.6. Se recuerda que este procedimiento lo repetiremos una segunda vez con las mismas características para el server2, realizado esto ya tendremos creadas las dos instancias necesarias para nuestra red, no se ha adjuntado de como se ha realizado para evitar ser repetitivo pues lo único que cambia en su creación es en el nombre de la instancia.

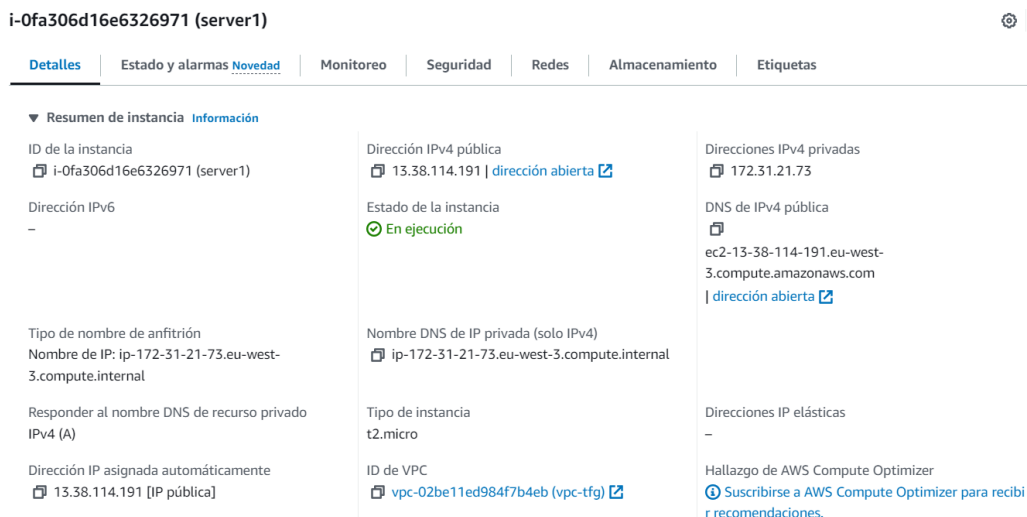


Figura 3.5: Detalles de la instancia server1.

The screenshot displays the 'Detalles' (Details) page for an EC2 instance with ID 'i-08b93e7990069ba56 (server2)'. The page is organized into several tabs: Detalles, Estado y alarmas, Novedad, Monitoreo, Seguridad, Redes, Almacenamiento, and Etiquetas. The 'Resumen de instancia' (Instance Summary) section is expanded, showing the following information:

- ID de la instancia:** i-08b93e7990069ba56 (server2)
- Dirección IPv4 pública:** 15.188.193.123 | [dirección abierta](#)
- Direcciones IPv4 privadas:** 172.31.20.118
- Estado de la instancia:** En ejecución (green circle with checkmark)
- DNS de IPv4 pública:** ec2-15-188-193-123.eu-west-3.compute.amazonaws.com | [dirección abierta](#)
- Nombre de IP:** ip-172-31-20-118.eu-west-3.compute.internal
- Nombre DNS de IP privada (solo IPv4):** ip-172-31-20-118.eu-west-3.compute.internal
- Tipo de instancia:** t2.micro
- Dirección IP asignada automáticamente:** 15.188.193.123 [IP pública]
- ID de VPC:** vpc-02be11ed984f7b4eb (vpc-tfg) | [vpc](#)
- Direcciones IP elásticas:** -
- Hallazgo de AWS Compute Optimizer:** [Suscribirse a AWS Compute Optimizer para recibir recomendaciones.](#)

Figura 3.6: Detalles de la instancia server2.

Si navegamos entre las pestañas que tiene el panel de control de cada servidor podemos visualizar todos los datos que proporciona EC2. En los detalles de cada instancia, por ejemplo, podemos ver varios datos de interés como el estado, el tipo de instancia, su VPC, sus direcciones IP públicas y privadas; las públicas como se puede deducir son las direcciones visibles desde el exterior y las privadas son las internas de cada máquina.

Para terminar con la creación de las instancias vamos a modificar en las dos máquinas los puertos para permitirnos usar la conexión mediante ssh, poder comprobar conexiones entre máquinas mediante tráfico ICMP (ping) y activar el puerto 10000 para el uso de Webmin. Para ello nos iremos a *Grupos de seguridad* al que corresponda a cada una de las instancias y añadimos las reglas necesarias para permitir el tráfico de entrada en cada puerto, como se aprecia en la imagen 3.7. El nombre del grupo de seguridad será el que nos salió en la creación de la instancia, también se puede ver en los detalles de la instancia en el apartado *Seguridad*.

The screenshot shows the 'Editar reglas de entrada' (Edit inbound rules) page for a security group. The breadcrumb navigation is: EC2 > Grupos de seguridad > sg-055eae59b60341c91 - launch-wizard-pc1 > Editar reglas de entrada. The page displays a table of inbound rules with the following columns: Tipo (Type), Protocolo (Protocol), and Intervalo de puertos (Port range).

Tipo	Protocolo	Intervalo de puertos
TCP personalizado	TCP	10000
SSH	TCP	22
HTTP	TCP	80
Todos los ICMP IPv4	ICMP	Todo

Figura 3.7: Reglas de entrada del grupo de seguridad.

Creados los server1 y server2 se procederá a instalar la herramienta **Webmin** para poder administrar nuestro servidor mediante una interfaz gráfica. Con esta herramienta

configuraremos en los siguientes puntos el servidor DHCP para que tanto el server1 y el server2 proporcionen direcciones a los RCO-1 y RCO-2.

Los pasos que se van a detallar son para la instalación de Webmin en server2, pero sería lo mismo para las dos instancias.

1. Nos conectamos mediante ssh a nuestra máquina, para ello lo haremos usando una terminal. Primero nos situaremos en el directorio donde esté situada nuestra clave de inicio de sesión, en nuestro caso estará en el archivo "claves.pem", que generamos cuando se crearon las instancias.

```
1 # ssh -i "claves.pem" ubuntu@ec2-15-188-193-123.eu-west-3.compute.
    amazonaws.com
```

2. Cambiamos la contraseña por una segura que usaremos para poder iniciar sesión en Webmin. Para este paso necesitaremos permisos de usuario root.

```
1 # sudo su
2 # passwd root
```

3. Actualizamos nuestro sistema y descargamos wget para la descarga web.

```
1 # sudo apt update
2 # sudo apt install wget apt-transport-https software-properties-
    common
```

4. Importar clave del repositorio Webmin:

```
1 # wget -q -O- http://www.webmin.com/jcameron-key.asc | sudo apt-key
    add-
```

5. Instalar Webmin en nuestra maquina Ubuntu (AWS EC2 Instance).

```
1 # sudo apt install webmin
```

Una vez hayamos completado los pasos mencionados, las máquinas estarán listas y disponibles para su uso de Webmin. A partir de este punto, tendremos la capacidad de establecer conexiones con ellas a través de cualquier navegador web. Esto te permitirá llevar a cabo tareas de administración y configuración en nuestros servidores de acuerdo con nuestras necesidades y preferencias específicas.

Al acceder a estas máquinas mediante un navegador, tendremos a nuestra disposición una interfaz en línea que facilitará la gestión de los servidores. A través de esta interfaz, podremos ajustar parámetros, establecer configuraciones, y tomar decisiones en función de los requerimientos particulares de nuestra red.

En esencia, el acceso a estas herramientas a través de un navegador se traduce en una administración más intuitiva y accesible. Ya no será necesario lidiar con procesos técnicos complicados; en cambio, podremos acceder a una interfaz amigable que simplifica el proceso de administrar los servidores como podremos observar más adelante.

Para entrar, al menos la primera vez lo haremos con la dirección DNS de ipv4 pública que nos genera AWS, figura 3.8, en el puerto 10000 que es donde escucha Webmin. Para este ejemplo, se ha usado el server2.



Figura 3.8: Dirección DNS de server 2.

Luego, tras iniciar sesión con usuario y contraseña podremos ver un panel de control como el de la figura 3.9. Hay que recordar que el puerto en el que escucha para poder entrar es el 10000, con lo que para entrar se deberá poner la dirección DNS añadiendo al final el puerto. La dirección para entrar entonces será: `ec2-15-188-193-123.eu-west-3.compute.amazonaws.com:10000` o bien usando su IP pública `15.188.193.123:10000`.

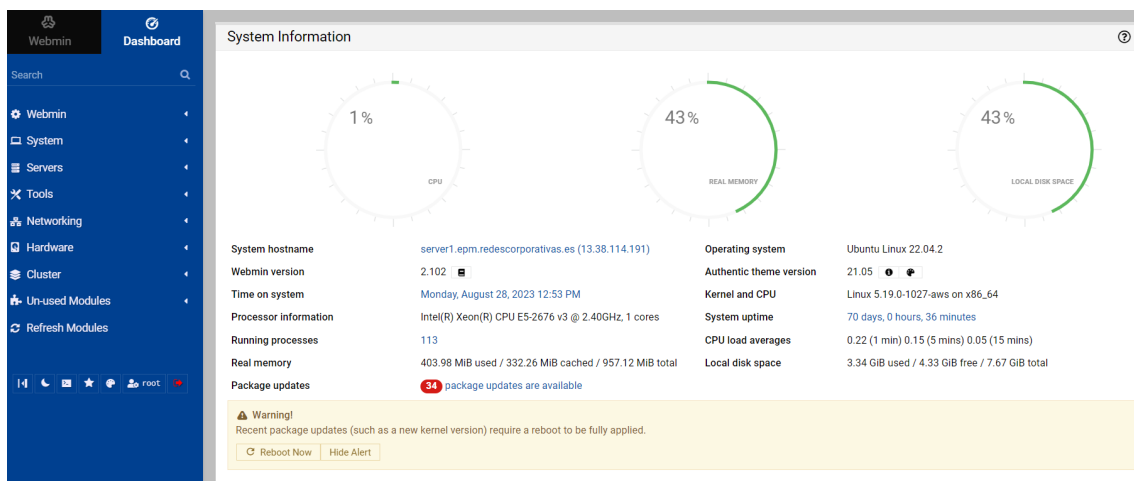


Figura 3.9: Página de inicio de Webmin server1.

Como se han observado en las imágenes dichas anteriormente se ha usado la dirección DNS que nos proporciona EC2 y se crea al lanzar la instancia; ahora bien, para poder conectarnos a la máquina de manera sencilla vamos a cambiar el nombre del cliente (hostname) de los servidores. Para ellos usaremos el comando `hostname` de la siguiente manera:

1. En el servidor 1:

```
1 # hostname server1.epm.redescorporativas.es
```

2. En el servidor 2:

```
1 # hostname server2.epm.redescorporativas.es
```

Luego de esto, haremos un `reboot` para que se cambie en cada máquina el nombre en ambas máquinas. Ahora podemos conectarnos a ellos con ese nombre, que es mucho más

amigable para el usuario. Una vez cambiados los nombres haremos un petición de certificado SSL [14] para estos hostname. Estos certificados permiten proteger la comunicación en línea, mejorar la confianza del usuario y cumplir con los requisitos de seguridad de los navegadores web modernos. Para obtener estos certificados lo gestionaremos mediante Webmin, para ello entraremos en *Webmin* en *Webmin Configuration*, figura 3.10.

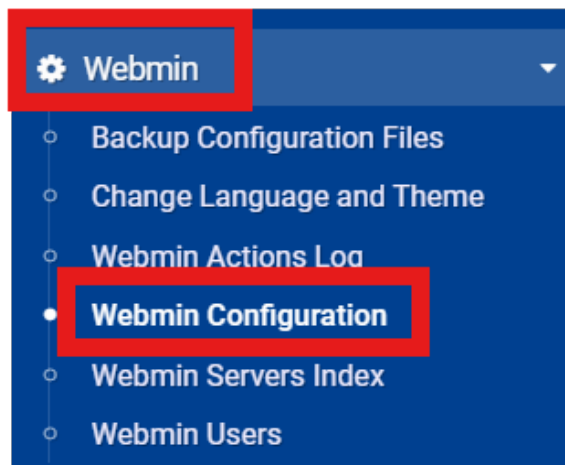


Figura 3.10: Configuración Webmin.

Dentro de la configuración de Webmin entraremos en **SSL Encryption** y en **Let's Encrypt**, una vez ahí podremos usar esta herramienta para pedir un certificado para nuestra la página. Aunque primero vamos a cambiar los ajustes SSL como se muestra en la figura 3.11.

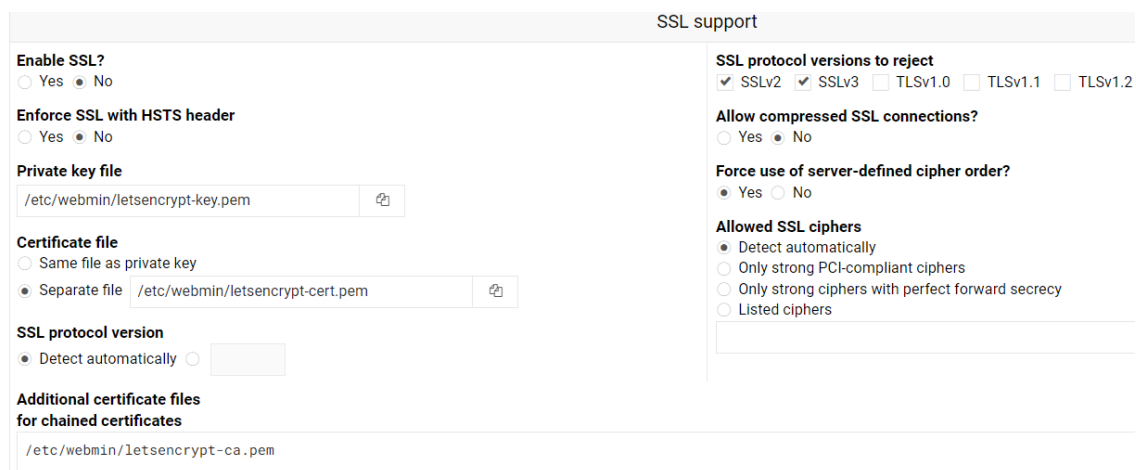


Figura 3.11: Ajustes SSL.

Luego, se ha realizado una petición pulsando en la pestaña **Let's Encrypt** para obtener un certificado, hemos rellenado con los siguientes datos y al finalizar se pulsará *Request Certificate*; en la figura 3.12 podemos ver un ejemplo para la petición del certificado para el **server2**.

Options for new SSL certificate

Hostnames for certificate

server2.epm.redescorporativas.es

Website root directory for validation file

Apache virtual host matching hostname

A different Apache virtual host <Default>

Other directory /var/www/html

SSL key size

Default bits

Let's Encrypt server

Real Staging (test only)

Months between automatic renewal

Only renew manually 2

Copy new key and certificate to Webmin?

Yes No

Request Certificate Just Update Renewal

Figura 3.12: Petición de certificado SSL.

Una vez obtenido el certificado podremos usar la dirección *server2.epm.redescorporativas.es* en el puerto 10000 para acceder a Webmin. Este método se realizará de manera similar en el *server1 server1.epm.redescorporativas.es* y se le asignará la dirección para obtener otro certificado. Ahora cada vez que entremos nos saldrá el certificado que hemos obtenido para ambos servidores como se puede ver en la figura 3.13. Con esto hecho, ya tenemos las máquinas listas para continuar con la configuración.

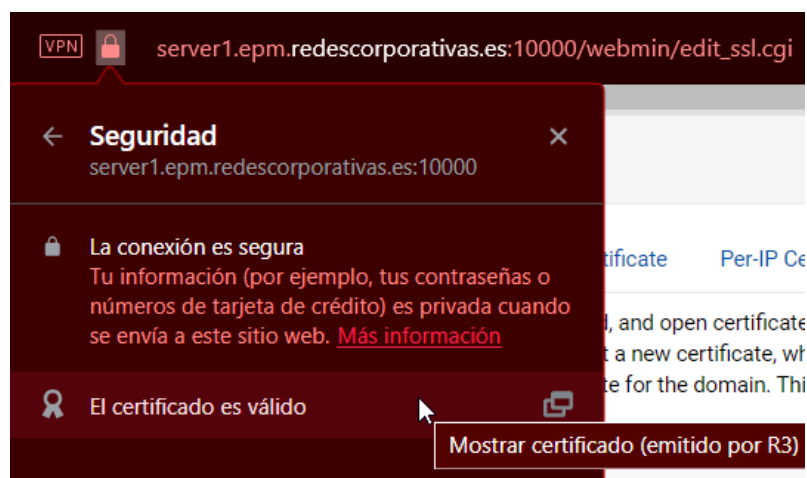


Figura 3.13: Certificado SSL de server1.

3.1.2. Máquinas virtuales linux (VMware)

Lo primero de todo para crear una máquina virtual necesitaremos una herramienta para poder virtualizar máquinas, en este caso se ha optado por la solución de VMware Workstation Player. Ahora mostraremos el proceso de creación de las máquinas virtuales que serán los hosts de nuestra red. Las características de estas máquinas son:

- Sistema operativo Almalinux 8.8.
- Con 2 CPU virtuales y 3 GB de memoria.
- Y con 20 GB de almacenamiento.

Tras descargar las imágenes y descomprimir simplemente abriremos la aplicación de VMware, pulsaremos "Open a Virtual Machine" y seleccionamos el archivo de la máquina virtual. Después, se renombrarán las máquinas para poder diferenciarlas: una será RCO-1 y RCO-2.



Figura 3.14: Creamos máquinas virtuales.

Para poder acceder a internet con estas máquinas primero deberemos configurar el adaptador de red para que se conecte a nuestro portátil mediante NAT.

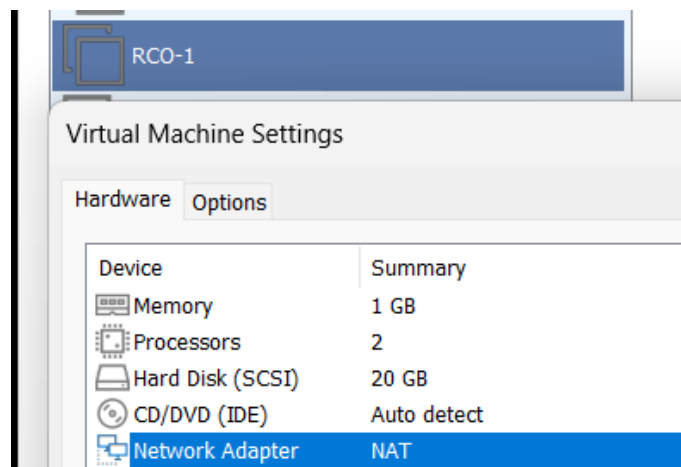


Figura 3.15: Cambiamos adaptador para conectarnos mediante NAT.

Luego de esto ya tendríamos nuestras máquinas ya operativas y preparadas para continuar con la configuración de nuestra red. Para poder entrar a ellas simplemente se pulsaría en la máquina que queramos y luego pulsar en *Play virtual machine*, figura 3.16, y con ello se iniciará nuestra máquina en VMware.

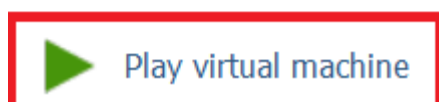
Virtual Machine Name:**RCO-1****State:** Powered Off**OS:** CentOS 8 64-bit**Version:** Workstation 15.x virtual machine**RAM:** 1 GB

Figura 3.16: Iniciar máquina virtual.

Una vez iniciada la maquina, habría que actualizar e instalar los paquetes necesarios para la configuración de la red. Para ello usaremos las instrucciones siguientes.

```
1 # apt-get update
2 # apt-get install ip-tools
3 # apt-get install iproute2
```

Estos pasos que se han realizado para la creación de la máquina RCO-1. Para la RCO-2 son los mismos pasos. Realizado esto nuevamente ya tendríamos preparadas las máquinas para continuar.

3.2 Conexión de los componentes

Para la conexión de nuestros componentes usaremos tres herramientas, una será un servidor DHCP que configuraremos en nuestros servidores cloud que proporcionarán una dirección IP a cada máquina RCO, una red ZeroTier que usarán cada máquina virtual para conectarse a los servidores, y una VPN Wireguard para conectar las instancias AWS.

3.2.1. Configuración del Servidor DHCP en Webmin

Para su configuración nos deberemos conectar a Webmin desde nuestro navegador iniciando sesión con la contraseña que hemos cambiado durante su instalación. Lo primero que haremos ahora será instalar el módulo que usaremos para su configuración; iremos a la parte de módulos "Unused Modules" donde buscaremos "DHCP Server" y lo instalaremos.

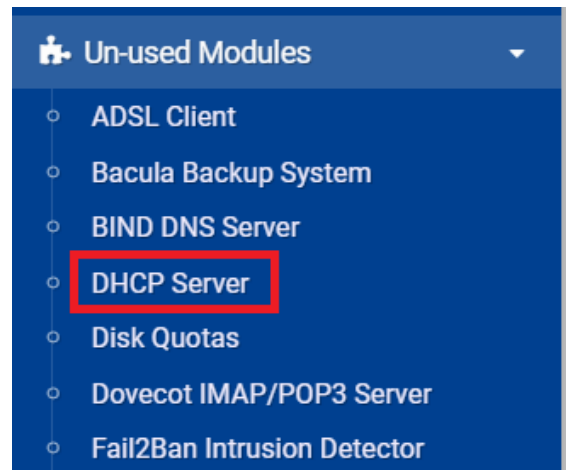


Figura 3.17: Módulo de DHCP Server.

Tras la instalación del módulo, ya podremos configurar nuestro servidor DHCP. Continuaremos creando una subred para contener nuestros hosts, donde se configurarán las direcciones y el rango que se les asignará cuando se conecten. Nuestras subredes tienen las siguientes características:

1. Para el *server1* el nombre del servidor DHCP será **subred-server1**. Elegiremos un rango de direcciones no muy amplio pues la finalidad de estas redes es de prueba y tampoco vamos a conectar al servidor muchos dispositivos, aun así este rango nos permitiría servir direcciones para un máximo de 11 dispositivos. Las características del servidor teniendo en cuenta lo anterior tendrán la siguiente configuración:
 - Dirección de red: 10.31.21.0
 - Rango de direcciones: 10.31.21.100-10.31.21.110
 - Máscara: 255.255.255.0

Subnet Details			
Subnet description	subred-server1		
Network address	10.31.21.0	Netmask	255.255.255.0
Address ranges	10.31.21.100 - 10.31.21.110	<input type="checkbox"/> Dynamic BOOTP ?	
	<input type="checkbox"/> Dynamic BOOTP ?		

Figura 3.18: Configuración de subred 1.

2. Para el *server2* el nombre del servidor DHCP será **subred-server2**. Elegiremos un rango de direcciones corto al igual que en el server1. Las características de este servidor tendrán la siguiente configuración:
 - Dirección de red: 10.31.22.0
 - Rango de direcciones: 10.31.22.100-10.31.22.110
 - Máscara: 255.255.255.0

Subnet Details			
Subnet description	subred-server2		
Network address	20.31.21.0	Netmask	255.255.255.0
Address ranges	20.31.21.100 - 20.31.21.110	<input type="checkbox"/> Dynamic BOOTP ?	
	<input type="checkbox"/> Dynamic BOOTP ?		

Figura 3.19: Configuración de subred 2.

Antes de arrancar nuestros servidores debemos de crear y conectar máquinas a la red ZeroTier; que se hará a continuación. Esto es necesario para el enrutamiento final entre el servidor y la VPN como se verá más adelante.

3.2.2. Creación de VPN ZeroTier

Para crear una red ZeroTier, lo primero será crear una cuenta en la página de inicio y entrar en el inicio. Una vez iniciado sesión, simplemente sería pulsar en "Create A Network", figura 3.20. Luego de crear las dos redes, las nombraremos para diferenciar una de otra: **red-tfg1** y **red-tfg2** para la red 1 y 2, respectivamente.

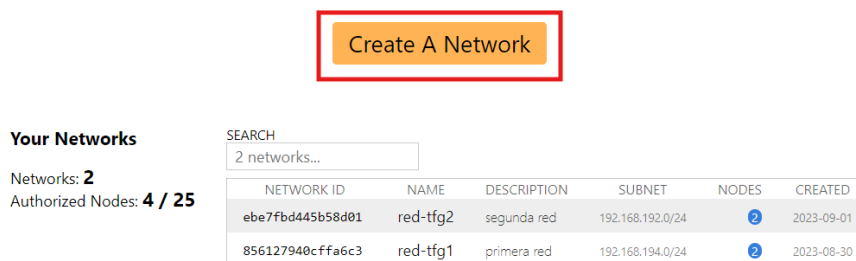


Figura 3.20: Redes ZeroTier.

En cuanto a la configuración de la red para que no entren en conflicto las máquinas conectadas con el servidor DHCP, desactivaremos el "autoasignamiento ipv4" de ZeroTier.

Una vez creadas la **red-tfg1** y **red-tfg2** ya estarán preparadas para conectar nuestras máquinas y autorizar su conexión; a continuación, comentaremos los pasos seguidos para ello. Tanto en los servidores como en las máquinas RCO debemos instalar el cliente de ZeroTier, para ello se han ejecutado los siguientes comandos:

```

1 # curl -s https://install.zerotier.com | sudo bash
2 # sudo systemctl start zerotier-one
3 # sudo systemctl enable zerotier-one
4 # sudo zerotier-cli join <codigo-de-red>
5 # sudo zerotier-cli info

```

El primer comando lo usamos para instalar el cliente; el segundo para iniciar el servicio; el tercer comando para habilitar el servicio al iniciar la máquina; el cuarto para unirnos a la red ZeroTier, donde tendremos que poner el código de nuestra red ZeroTier que encontraremos en el inicio de la página debajo del nombre; y el último comando, sirve para verificar el estado de nuestra conexión.

Una vez conectados a la red ZeroTier, en la página de monitorización deberemos de autorizar cada máquina para confirmar la conexión en cada red. Por ejemplo en la siguiente imagen se muestra como salen los dispositivos RCO1 y server1 en la red1, en la imagen salen los dos ya autorizados en la red pero si no lo estuvieran valdría con darle click. Por otro lado se les ha asignado un nombre para diferenciar entre RCO y server ya que al conectarnos a la red cada dispositivo se identifica con una dirección única, esta dirección se asigna una vez instalamos el cliente y la podemos consultar ejecutando el comando `zerotier-cli info`. Además en este paso, asignaremos al server1 y server2 una dirección estática y válida dentro de la subred de cada servidor DHCP; estas direcciones serán 10.31.21.1 y 10.31.22.1, respectivamente.

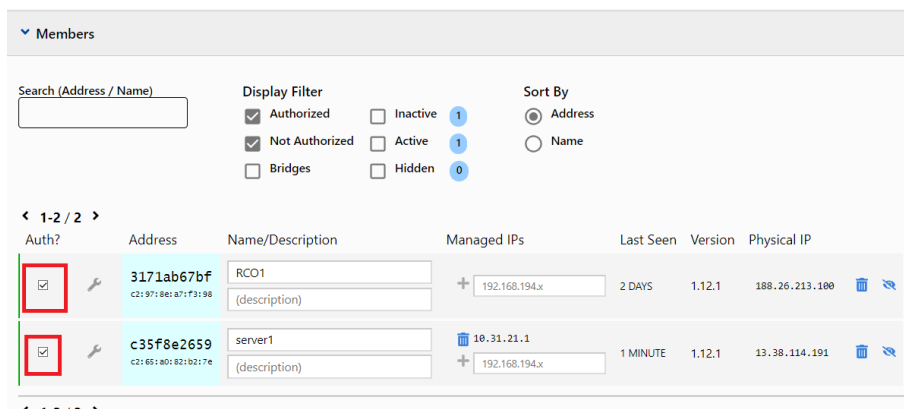


Figura 3.21: Autorización de conexión a la red ZeroTier para la red 1.

Una vez conectados los dispositivos a la red procederemos a configurar las interfaces creadas con el cliente ZeroTier, en server1 y server2. Lo primero será averiguar el nombre de nuestra interfaz con el comando *ifconfig*; una vez ya sabemos el nombre de la interfaz del cliente le asignaremos la IP que se ha puesto en la VPN para que sea nuestra puerta de enlace entre servidor DHCP y la VPN. Esta asignación se ha realizado con el comando *ifconfig <nombre-interfaz><IP>*.

- En el server1 se ha usado el comando:

```
1 # ifconfig ztcfw5qmw 10.31.21.1
```

- En el server2 se ha usado el comando:

```
1 # ifconfig zth6re4wke 10.31.22.1
```

Una vez hecho esto podemos comprobar la asignación con el comando *ifconfig* como se ha hecho para saber el nombre de las interfaces de ZeroTier de cada servidor.

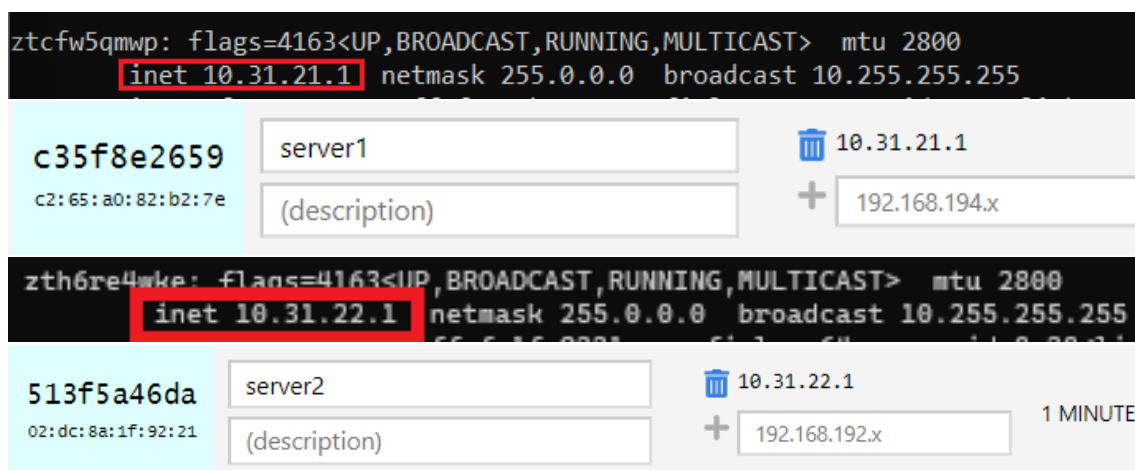


Figura 3.22: Asignamos IP a nuestra interfaz conectada a ZeroTier.

En la imagen 3.22 se pueden observar las direcciones asociadas a cada interfaz de red-tfg1 y red-tfg2, la primera imagen es de la interfaz del server1 y la segunda es de la interfaz del server2. Como aclaración, estas son iguales a las direcciones estáticas que configuramos antes en ZeroTier para cada servidor.

Una vez hecho esto modificaremos la interfaz de escucha del servidor DHCP, esto lo haremos desde la interfaz de Webmin. Desde la pestaña *Servers - DHCP Server - Network Interface*, la interfaz que elegiremos será la misma a la que hemos asignado la dirección anterior, basta con seleccionar y guardar, como se puede observar en la figura 3.23 para el caso del server1. Con esto la interfaz conectada a ZeroTier asigna también las direcciones a los dispositivos que se conecten a esta.

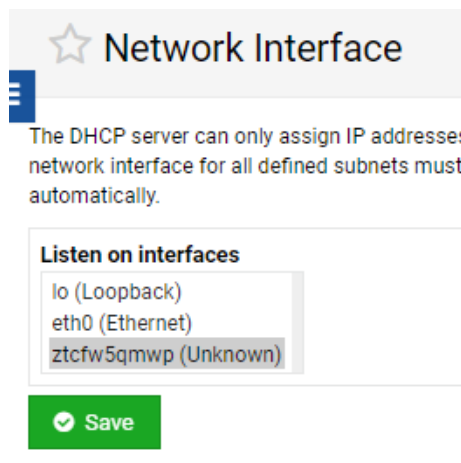


Figura 3.23: Cambiamos interfaz de escucha del servidor DHCP.

Una vez hecho todo esto, los servidores ya estarán conectados a sus respectivas redes ZeroTier y listas para servir direcciones una vez se conecten a las redes **red-tfg1** y **red-tfg2** las máquinas clientes.

3.2.3. Arranque del servidor DHCP y conexión de las máquinas RCO

Con todos los elementos creados y configurados anteriormente ya está todo listo para arrancar el servidor DHCP. Simplemente bastaría con pulsar "Start Server" en Webmin o bien desde la terminal con el comando:

```
1 # systemctl start isc-dhcp-server
```

Una vez teniendo ya operativo el servidor DHCP, conectados y autorizados los servidores y las máquinas RCO en la red ZeroTier correspondiente procederemos a conectarnos. Esto lo haremos desde terminal ejecutando el comando `dhclient <nombre-interfaz-ZeroTier>`, que nos asignará una dirección dentro del rango establecido, estas direcciones serán: 10.31.21.100 y 10.31.22.100 para el RCO-1 y RCO-2, respectivamente en las interfaces ZeroTier **ztcfw5qmw** y **zth6re4wke**.

```
[root@rco-1-redescorporativas-es ~]# dhclient ztcfw5qmw
ztcfw5qmw: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 2800
inet 10.31.21.100 netmask 255.255.255.0 broadcast 10.31.31.255

[root@rco-2-redescorporativas-es ~]# dhclient zth6re4wke
zth6re4wke: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 2800
inet 10.31.22.100 netmask 255.255.255.0 broadcast 10.31.22.255
```

Figura 3.24: Conexión de las máquinas RCO con servidores.

Ya conectados a los servidores podemos ver desde Webmin los clientes conectados y la dirección que se les ha asignado en el apartado "List Active Leases", figura 3.25.

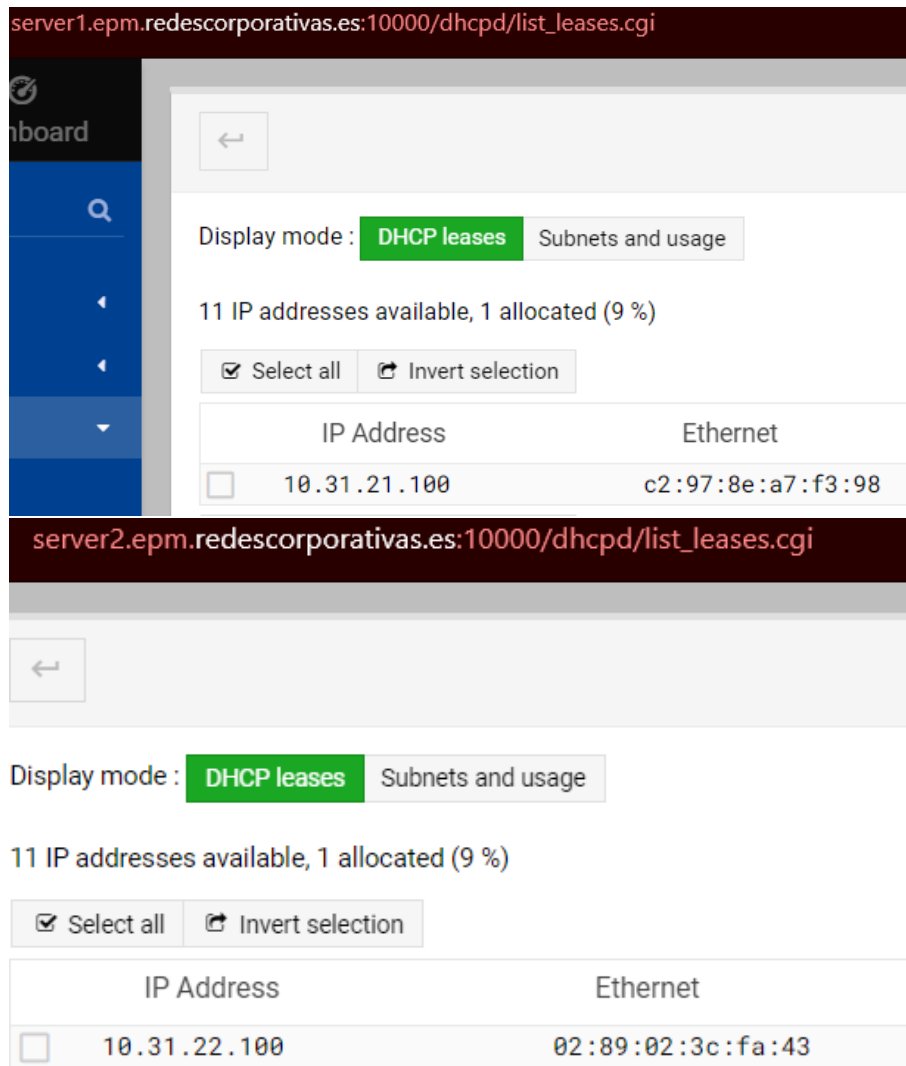


Figura 3.25: Clientes activos en el server1 (arriba) y en el server2 (abajo).

A partir de ahora se puede comprobar la conexión haciendo un “ping” desde un RCO al servidor correspondiente. En las imágenes de la figura 3.26, se comprueban la conectividad en las dos redes, la primera imagen es la de la red 1 (RCO-1 y server1) y la segunda de la red 2 (RCO-2 y server2). En estas las IP de los servidores 1 y 2 son 172.31.21.73 y 172.31.20.118, respectivamente, para poder conectar RCO y servidor añadimos las siguientes reglas mediante comandos.

- En el servidor 1:

```
1 # ip route add 10.31.21.1 via 172.31.21.73
```

- En el servidor 2:

```
1 # ip route add 10.31.22.1 via 172.31.20.118
```

```
[root@rco-1-redescorporativas-es ~]# ping 172.31.21.73
PING 172.31.21.73 (172.31.21.73) 56(84) bytes of data.
64 bytes from 172.31.21.73: icmp_seq=1 ttl=64 time=28.2 ms
64 bytes from 172.31.21.73: icmp_seq=2 ttl=64 time=26.4 ms
64 bytes from 172.31.21.73: icmp_seq=3 ttl=64 time=26.5 ms
64 bytes from 172.31.21.73: icmp_seq=4 ttl=64 time=27.7 ms
^C
--- 172.31.21.73 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 26.443/27.200/28.212/0.794 ms

[root@rco-2-redescorporativas-es ~]# ping 172.31.20.118
PING 172.31.20.118 (172.31.20.118) 56(84) bytes of data.
64 bytes from 172.31.20.118: icmp_seq=1 ttl=64 time=68.9 ms
64 bytes from 172.31.20.118: icmp_seq=2 ttl=64 time=25.0 ms
64 bytes from 172.31.20.118: icmp_seq=3 ttl=64 time=48.0 ms
64 bytes from 172.31.20.118: icmp_seq=4 ttl=64 time=23.7 ms
^C
--- 172.31.20.118 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 23.736/41.417/68.888/18.576 ms
```

Figura 3.26: Comprobación de conexión entre máquinas RCO y servidores. Ping desde RCO-1 a server1 (arriba). Ping desde RCO-2 a server2 (abajo)

Con esto ya tenemos preparadas las conexiones necesarias para continuar. Ahora tenemos la red con todos los hosts que vamos a usar y las conexiones que necesitamos para continuar, aunque todavía falta por configurar lo último que sería la redirección del tráfico para que este vaya por nuestros servidores en la nube. Esto se detallará en el siguiente punto donde se configura todo y se procede con la comprobación de la red y que cumple el objetivo de este proyecto.

3.2.4. Conexión entre server1 y server2

Para la conexión entre servidores se ha realizado estableciendo una conexión Wireguard entre sí. Lo primero que haremos será instalar en los dos servers la aplicación de Wireguard:

```
1 # apt install wireguard
```

Tras la instalación podemos comprobar que su instalación se ha hecho correctamente con el comando:

```
1 # modprobe wireguard
```

Sí está todo bien, como en nuestro caso no dará ningún tipo de error en su salida. Con la herramienta instalada vamos a seguir, como esta tecnología usa claves públicas y privadas codificadas en base62 para vincular los dispositivos se han generado en ambas máquinas sus respectivas claves. Para la creación de las claves se puede usar una herramienta que nos ofrece el sistema de Wireguard, los comandos para la generación de claves y que se deben introducir en los dos servidores son:

```
1 # umask 077
2 # wg genkey > private
3 # wg pubkey < private
```

El primer comando se usa para que cualquier archivo creado después de ejecutarlo tenga permisos de escritura, lectura y ejecución solo para el propietario del archivo, y ningún permiso para el grupo ni otros usuarios. El segundo comando sirve para crear claves privadas y se guarde en el archivo **private**. El tercer comando nos permite extraer la clave pública asociada a la clave privada **private** en nuestro caso.

Con las claves creadas continuamos con la configuración de la interfaz de red de Wireguard. Para ello se realizará lo siguiente:

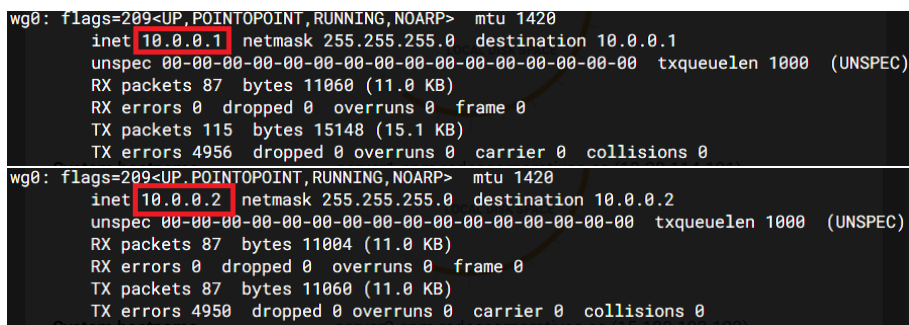
- En **server1**:

```
1 # ip link add wg0 type wireguard
2 # ip addr add 10.0.0.1/24 dev wg0
3 # wg set wg0 private-key ./private
4 # ip link set wg0 up
```

- En **server2**:

```
1 # ip link add wg0 type wireguard
2 # ip addr add 10.0.0.2/24 dev wg0
3 # wg set wg0 private-key ./private
4 # ip link set wg0 up
```

El primer comando realizado sirve para crear una interfaz de red con el nombre **wg0** utilizando el tipo Wireguard que nos proporcionará conexión segura punto a punto entre los dispositivos. El segundo comando es distinto, con este se asigna la dirección IP '10.0.0.1' con una máscara de subred '/24' para el **server1** y la dirección IP '10.0.0.2' con una máscara de subred '/24' para la interfaz del **server2**. Con el tercer comando establecemos la clave privada para la interfaz wg0 en ambos casos, esta clave usa el archivo 'private'. El cuarto comando se introduce para activar las interfaces, lo que con ello estará ya lista para transmitir y recibir datos.

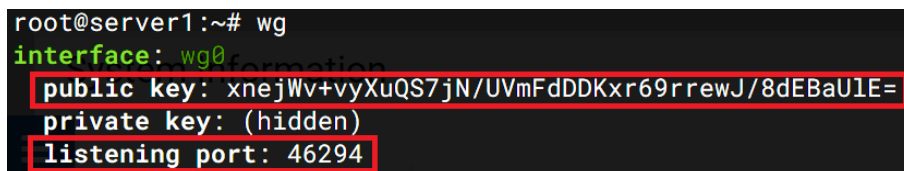


```
wg0: flags=209<UP,POINTOPOINT,RUNNING,NOARP> mtu 1420
inet 10.0.0.1 netmask 255.255.255.0 destination 10.0.0.1
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
RX packets 87 bytes 11060 (11.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 115 bytes 15148 (15.1 KB)
TX errors 4956 dropped 0 overruns 0 carrier 0 collisions 0

wg0: flags=209<UP,POINTOPOINT,RUNNING,NOARP> mtu 1420
inet 10.0.0.2 netmask 255.255.255.0 destination 10.0.0.2
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
RX packets 87 bytes 11004 (11.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 87 bytes 11060 (11.0 KB)
TX errors 4950 dropped 0 overruns 0 carrier 0 collisions 0
```

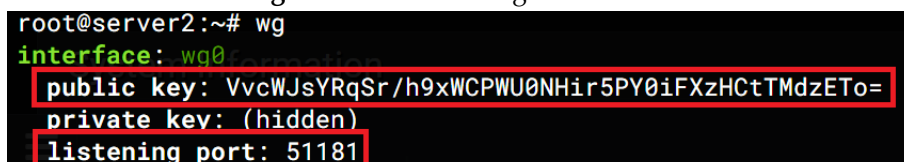
Figura 3.27: Interfaces wg0 en server1(arriba) y server2(abajo).

Ahora los dos servidores tienen Wireguard instalado y sus interfaces están listas para su conexión. Como se aprecian en las figuras 3.28 y 3.29, comprobamos la creación de las interfaces y apuntamos los datos necesarios para conectar ambos hosts, estos son la clave pública (public key) y el puerto de escucha (listening port).



```
root@server1:~# wg
interface: wg0
public key: xnejWv+vyXuQS7jN/UVmFdDDKxr69rrewJ/8dEBaU1E=
private key: (hidden)
listening port: 46294
```

Figura 3.28: Interfaz wg0 en server1.



```
root@server2:~# wg
interface: wg0
public key: VvcWJsYRqSr/h9xWCPWU0NHir5PY0iFXzHCtTMdzETo=
private key: (hidden)
listening port: 51181
```

Figura 3.29: Interfaz wg0 en server2.

Conocidos estos datos ya podemos proceder a conectarlos para ello ejecutaremos los comandos siguientes en cada caso:

- En **server1**:

```
1 # wg set wg0 peer VvcWJsYRqSr/h9xWCPWU0NHir5PY0iFXzHCtTMdzETo=
   allowed-ips 10.0.0.2/32, 10.31.22.0/24 endpoint
   172.31.20.118:51181
```

- En **server2**:

```
1 # wg set wg0 peer xnejWv+vyXuQS7jN/UVmFdDDKxr69rrewJ/8dEBaUIE=
   allowed-ips 10.0.0.1/32, 10.31.21.0/24 endpoint 172.31.21.73:46294
```

Con estos comandos lo que hacemos es crear un de enlace desde la interfaz de red **wg0** a la máquina que se desea conectar, para ello se debe introducir la clave pública, el **endpoint** que tendrá la IP del dispositivo que queramos conectar que será la IP de cada máquina. Además, para que pueda aceptar conexiones desde nuestra interfaz **zerotier** vamos a añadir el rango que pertenece a nuestra subred, en cada caso usaremos el rango 10.31.21.0/24 y 10.31.22.0/24 para que cualquier dirección que provenga de la interfaz sea aceptada por esta conexión también.

Ahora ya conectados **server1** y **server2** vamos a comprobar que todo vaya bien. Para ello se ha hecho un ping desde **server1** a **server2** y viceversa usando la IP de la interfaz **wg0**, como se ve en la siguiente imagen. Aunque también podemos ver la transferencia usando el comando **wg**, figura 3.30.

```
root@server1:~# wg
interface: wg0
  public key: xnejWv+vyXuQS7jN/UVmFdDDKxr69rrewJ/8dEBaUIE=
  private key: (hidden)
  listening port: 46294

peer: VvcWJsYRqSr/h9xWCPWU0NHir5PY0iFXzHCtTMdzETo=
  endpoint: 172.31.20.118:51181
  allowed ips: 10.0.0.2/32, 10.31.22.0/24
  latest handshake: 36 seconds ago
  transfer: 70.17 MiB received, 31.08 MiB sent

root@server2:~# wg
interface: wg0
  public key: VvcWJsYRqSr/h9xWCPWU0NHir5PY0iFXzHCtTMdzETo=
  private key: (hidden)
  listening port: 51181

peer: xnejWv+vyXuQS7jN/UVmFdDDKxr69rrewJ/8dEBaUIE=
  endpoint: 172.31.21.73:46294
  allowed ips: 10.0.0.1/32, 10.31.21.0/24
  latest handshake: 1 minute, 31 seconds ago
  transfer: 30.08 MiB received, 70.16 MiB sent
```

Figura 3.30: Monitorización de las interfaces **wg0**.

Como vemos en las anteriores fotos, se aprecia en cada servidor la interfaz de Wireguard con el par conectado a ella. En los dos casos vemos una serie de datos donde nos da datos del dispositivo, el par se identifica con la clave pública de este, el **endpoint** es la IP de la máquina junto el puerto en el que escucha; también nos da las IP permitidas

que sería las IP de tipo Wireguard que creamos en la configuración. También, podemos ver otros datos como hace cuando se hizo la última conexión *latest handshake* y los datos recibidos *received* y enviados *sent*. De esta manera es como trabaja esta herramienta, nos proporciona conexiones seguras donde se vinculan los dispositivos mediante un conjunto de claves codificadas relacionadas a la IP de cada máquina.

Ahora tendremos las dos redes con los servidores en la nube conectados a sus clientes RCO correspondientes de manera segura con ZeroTier y también tenemos una conexión, que es la última que hemos configurado entre los `server1` y `server2` mediante Wireguard.

3.3 Direccionamiento del tráfico

Tras comprobar que las conexiones funcionan, vamos a continuar con el enrutamiento que es el objetivo que tenemos con esta red; el tráfico de red de los clientes RCO debe de ir a través de los servidores.

Para ello en los servidores vamos a crear reglas para redirigir el tráfico, aunque antes deberemos de habilitar el reenvío de paquetes IP que se hará añadiendo la siguiente línea en el archivo `sysctl.conf`, en el directorio `/etc`:

```
1 net.ipv4.net_forward=1
```

Configurar nuestro sistema para que funcione como enrutador o puerta de enlace entre dos redes es esencial en nuestro caso. La línea que hemos añadido tiene la función de configurar el control en nuestro sistema que reenvía los paquetes IP de una interfaz a otra. Al establecer este valor en '1', se habilita el reenvío, lo que permite que nuestro sistema actúe como un enrutador o puerta de enlace. Esto implica dirigir el tráfico de red entre diferentes interfaces de red. En nuestra red, nuestro objetivo es facilitar la comunicación del tráfico de red entre la interfaz de nuestro servidor y la interfaz de nuestra red ZeroTier.

Después de realizar este cambio en el archivo de configuración, es crucial aplicar los cambios ejecutando el comando `sysctl -p` para asegurar que la configuración surta efecto.

Luego, para establecer las reglas que gestionen el tráfico entre las interfaces, se deben ejecutar los siguientes comandos en el terminal:

- En el servidor 1:

```
1 # iptables -t nat -A POSTROUTING -o eth0 -s 10.31.21.0/24 -j
   MASQUERADE
2 # iptables -A FORWARD -i ztcfw5qmwp -o eth0 -j ACCEPT
3 # iptables -A FORWARD -i eth0 -o ztcfw5qmwp -m state --state
   RELATED,ESTABLISHED -j ACCEPT
```

- En el servidor 2:

```
1 # iptables -t nat -A POSTROUTING -o eth0 -s 10.31.22.0/24 -j
   MASQUERADE
2 # iptables -A FORWARD -i zth6re4wke -o eth0 -j ACCEPT
3 # iptables -A FORWARD -i eth0 -o zth6re4wke -m state --state
   RELATED,ESTABLISHED -j ACCEPT
```

Para comprender mejor la función de estas reglas, consideremos el ejemplo de la primera red con el `server1`. El primer comando crea una regla de traducción de direcciones de red (NAT) [15]. Esta regla se aplica después de que un paquete haya sido enrutado y

salga por la interfaz **eth0**. Básicamente, cambia la dirección IP de origen de los paquetes que provienen de la red '10.31.21.0/24', de manera que parezcan provenir de la propia interfaz **eth0**. Este proceso es útil para enmascarar el origen de los paquetes salientes.

El segundo comando establece una regla que permite el reenvío de paquetes desde la interfaz de red **ztcfw5qmwp** hacia la interfaz **eth0**. Esto significa que los paquetes que llegan a través de la interfaz **ztcfw5qmwp** serán permitidos y reenviados por el sistema hacia la interfaz **eth0**. Esta regla es esencial para permitir la comunicación entre las redes conectadas a estas interfaces.

La tercera regla, por otro lado, permite el flujo de paquetes relacionados o establecidos desde la interfaz **eth0** hacia la interfaz **ztcfw5qmwp**. Esto garantiza que las respuestas y los paquetes asociados a las conexiones iniciadas desde la interfaz **ztcfw5qmwp** sean permitidos y reenviados correctamente hacia esa interfaz.

Es importante tener en cuenta que estas reglas son temporales y se perderán si el servidor se apaga o reinicia. Para evitar esta pérdida, se recomienda guardar las reglas en un archivo utilizando el comando `iptables-save > reglas-routing`. Esto permite restaurar las reglas después de un reinicio o apagado del servidor. Además, es posible automatizar la ejecución de estas reglas al inicio del servidor para garantizar una configuración consistente y evitar problemas de conectividad. Aunque no nos enfocaremos en esta automatización en este trabajo, es importante tener en cuenta esta posibilidad como una solución práctica para mantener la integridad de la configuración del sistema.

Antes de continuar, es importante tener en cuenta las direcciones de cada máquina en nuestra red, por ello hemos resumido las direcciones IP de cada máquina en las siguientes tablas. Esto facilitará la identificación del origen y destino de cada paquete en la redirección.

	Interfaz eth0	Interfaz ZeroTier
RCO-1	192.168.240.131	10.31.21.100
RCO-2	192.168.240.134	10.31.22.100

Tabla 3.1: Direcciones IP de los RCO

	Interfaz eth0	Interfaz ZeroTier
server1	172.31.21.73	10.31.21.1
server2	172.31.20.118	10.31.22.1

Tabla 3.2: Direcciones IP de los servidores

Con todo esto tenido en cuenta ya tenemos configurado el direccionamiento en los servidores con lo que todo el tráfico que venga de origen de la interfaz de ZeroTier, desde nuestro cliente RCO saldrá hacia internet por nuestra máquina.

CAPÍTULO 4

Funcionamiento de la red

En el funcionamiento de la red se hará mediante dos pruebas: primero haremos una que compruebe la conexión de nuestras máquinas RCO mediante el envío de datos; y la segunda prueba será el desvío del tráfico de nuestro clientes a la página "https://miip.es".

4.1 Transferencia de datos entre máquinas.

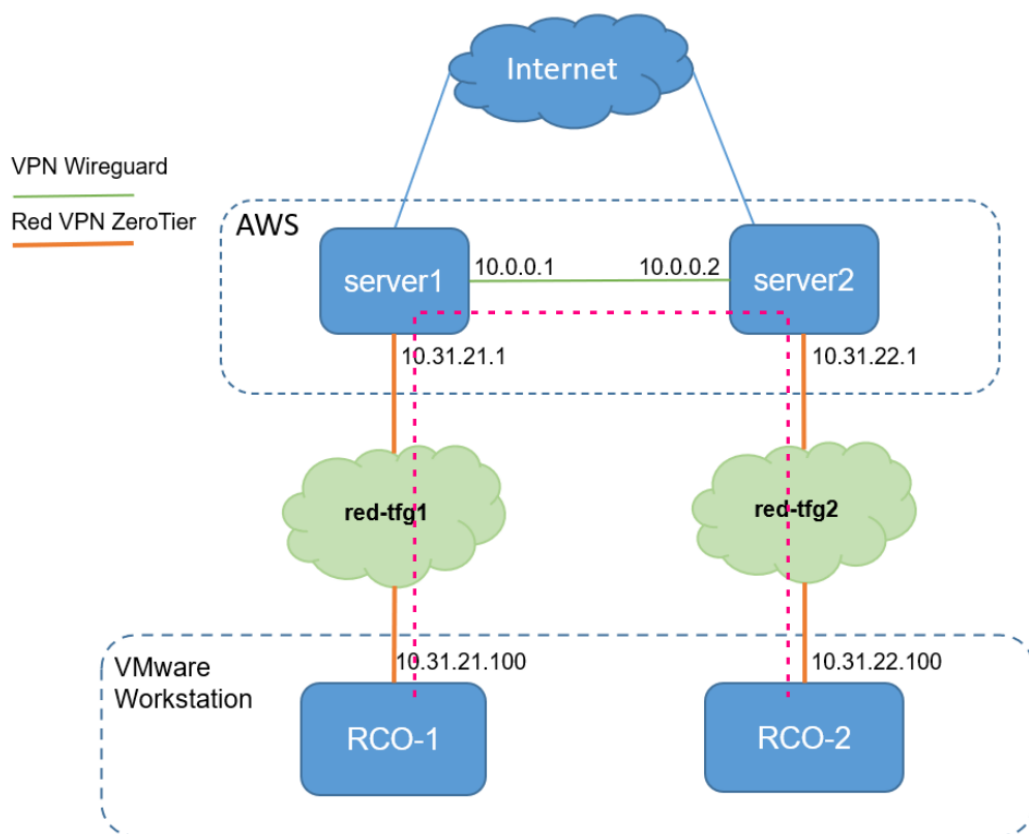


Figura 4.1: Conexión entre RCO-1 y RCO-2.

Para comprobar la interconexión en nuestra red, vamos a realizar una prueba que nos muestre la transferencia de datos entre las máquinas RCO y con ello asegurarnos de que tienen conexión entre ellos, en la figura 4.1 se muestra con una línea discontinua la conexión entre ambas máquinas. Para que se puedan conectar debemos de añadir una

serie de reglas tanto en los servidores como en los RCO. Vamos a mostrar primero las reglas y explicaremos que hace cada una.

Reglas en servidores

- En el server1:

```
1 # ip route add 10.31.22.0/24 dev wg0 scope link
```

- En el server2:

```
1 # ip route add 10.31.21.0/24 dev wg0 scope link
```

Esta regla en los servidores redirigen el tráfico de la red del otro servidor por su interfaz de Wireguard, esto se hace con el fin de crear una conexión para que todo el tráfico que venga de los clientes de la otra red pueda hacer tunneling por su interfaz de Wireguard e interconectarse. Ahora con esto en los RCO debemos de redirigir también todo el tráfico de nuestra red para que vaya por el router de la otra red, de la siguiente manera:

Reglas en RCO

- En el RCO-1:

```
1 # ip route add 10.31.22.0/24 via 10.31.21.1 dev ztcfw5qmwq
```

- En el RCO-2:

```
1 # ip route add 10.31.21.0/24 via 10.31.22.1 dev zth6re4wke
```

Para explicar que hacen estas reglas tomaremos el caso del RCO-1, esta redirige todo el tráfico que tenga un rango de direcciones 10.31.22.0/24 que pertenece a la red 2 por su interfaz Zerotier.

Con estas reglas añadidas, podemos intentar hacer un ping para comprobar la conexión, figura 4.2.

```
[root@rco-1-redescorporativas-es ~]# ping 10.31.22.100
PING 10.31.22.100 (10.31.22.100) 56(84) bytes of data.
64 bytes from 10.31.22.100: icmp_seq=1 ttl=62 time=99.7 ms
64 bytes from 10.31.22.100: icmp_seq=2 ttl=62 time=115 ms
64 bytes from 10.31.22.100: icmp_seq=3 ttl=62 time=134 ms
64 bytes from 10.31.22.100: icmp_seq=4 ttl=62 time=89.9 ms
```

Figura 4.2: Ping de RCO-1 a RCO-2.

Con ello ahora podemos comprobar su funcionamiento transfiriendo un archivo de vídeo llamado *pajaros.mp4* que hemos descargado para este ejemplo. Primero que nada para comparar que todo vaya bien en la transferencia a modo de comprobar que se transfiera de manera correcta generaremos un checksum, que es un valor numérico calculado a partir de un conjunto de datos de manera que al recibir un mensaje si da el mismo valor para ese conjunto de datos (en nuestro caso los de la imagen) verificaremos que se ha realizado correctamente. Generado el checksum procederemos a enviar el archivo *pajaros.mp4* desde RCO-2 a RCO-1 usando SCP (Secure Copy Protocol) [16], que es un

protocolo y herramienta para transferir de manera segura archivos desde una máquina local a una remota. Tras la descarga generaremos el checksum para poder comprobar que el archivo ha sido enviado correctamente.

```
[root@rco-2-redescorporativas-es prueba-transferencia]# scp pajaros.mp4 root@10.31.21.100:/home/prueba-transferencia/
root@10.31.21.100's password:
pajaros.mp4 100% 19MB 422.6KB/s 00:46
```

Figura 4.3: Transferencia de archivo desde RCO-2 a RCO-1.

```
[root@rco-1-redescorporativas-es prueba-transferencia]# md5sum pajaros.mp4
4fc966788985a75a34b1d0d5b0827c56 pajaros.mp4
[root@rco-2-redescorporativas-es prueba-transferencia]# md5sum pajaros.mp4
4fc966788985a75a34b1d0d5b0827c56 pajaros.mp4
```

Figura 4.4: Verificación de transferencia.

Como podemos observar en las figuras 4.3 y 4.4, podemos comprobar que la transferencia entre RCO-2 y RCO-1, cuya dirección es 10.31.21.100, se ha realizado correctamente y verificamos que los datos son los mismos con el checksum que se generan.

4.2 Comprobación de la redirección del tráfico

Para comprobar que el direccionamiento de los datos se ha realizado correctamente vamos a probar a redirigir el tráfico a la página "miip.es". Hemos optado por redirigir esta como ejemplo porque solo tiene una dirección pública. La dirección IP asociada a ese dominio la podemos obtener usando una herramienta como **nslookup** que se usa para consultar servidores DNS y obtener detalles sobre el nombre de dominio específico.

```
[root@rco-2-redescorporativas-es prueba-transferencia]# nslookup miip.es
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
Name:   miip.es
Address: 82.223.37.148
```

Figura 4.5: Dirección IP del dominio.

Obtenemos la dirección **82.223.37.148** para el dominio **miip.es**, con ese dato añadiremos una regla para poder redirigir el tráfico de nuestros RCO hacia esta dirección. Para ello, se ha ejecutado los siguientes comandos:

- En el RCO-1:

```
1 # ip route add 82.223.37.148/32 via 10.31.21.1 dev ztcfw5qmwq
```

- En el RCO-2:

```
1 # ip route add 82.223.37.148/32 via 10.31.22.1 dev zth6re4wke
```

Añadidas esas reglas al entrar en ese dominio que nos proporciona nuestra IP pública podemos saber de donde proceden los datos. Al acceder nos saldrá un mensaje con la

dirección IP, si todo ha ido correctamente nos coincidirá con la IP pública de nuestro servidor en la nube.

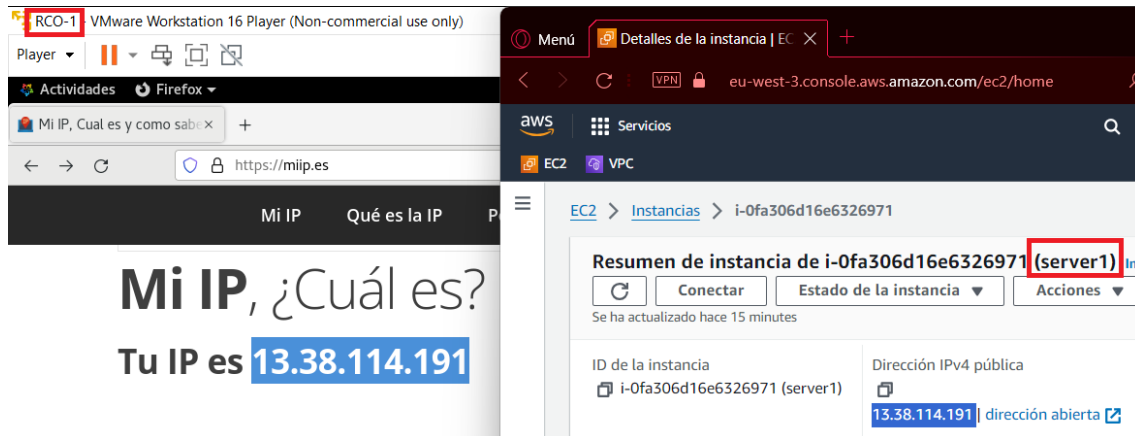


Figura 4.6: Dirección IP desde RCO-1.

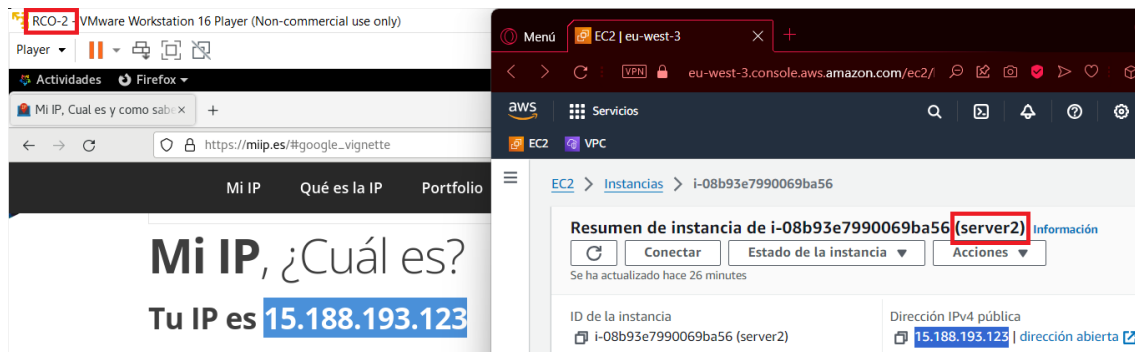


Figura 4.7: Dirección IP desde RCO-2.

En las figuras 4.6 y 4.7 podemos observar como ambas direcciones la pública del servidor coincide con la que nos da en la página *miip.es*, con esto se demuestra que el tráfico ha sido redirigido correctamente y con ello conseguimos el objetivo de este trabajo. Antes que nada, se puede señalar que con esta red se puede redirigir todo tráfico que queramos por nuestro router, ya sea *server1* o *server2*, aunque hay que tener en cuenta los rangos que use el dominio ya que puede dar resultado a errores. En nuestro caso, como se ha usado un dominio con una sola dirección se puede apreciar mucho más fácilmente que todo funciona correctamente. Ahora podemos comentar que es posible usar un servidor en la nube y configurarlo de manera que se comporte como un router.

CAPÍTULO 5

Conclusiones y mejoras futuras

En este proyecto, abordamos una tarea interesante en la administración de redes: la configuración de dos redes idénticas en AWS, cada una compuesta por un servidor Ubuntu que actúa como servidor DHCP, y la implementación de una VPN ZeroTier para establecer conexiones seguras entre los clientes y los servidores. Nuestro objetivo principal fue enrutar el tráfico de red desde los clientes a través de los servidores, y a lo largo de este trabajo, exploramos cada paso necesario para lograrlo con éxito.

En primer lugar, destacamos la elección de Ubuntu como sistema operativo para nuestros servidores en AWS debido a su versatilidad y amplio soporte en la comunidad de código abierto.

La configuración del servidor DHCP fue un componente importante de nuestro proyecto, y optamos por utilizar Webmin para simplificar este proceso. Con Webmin, pudimos establecer un rango de direcciones IP para los clientes, lo que garantiza la asignación automática de direcciones sin necesidad de intervención manual. Esta herramienta de administración web demostró ser valiosa al ofrecer una interfaz intuitiva y fácil de usar para gestionar el servidor DHCP.

La implementación de la VPN ZeroTier fue un paso crucial para asegurar la conectividad segura entre los clientes y los servidores. ZeroTier nos proporcionó una solución eficiente y confiable para crear una red privada virtual, permitiendo que los clientes se conecten a los servidores de manera segura a través de Internet. Además, logramos establecer una conexión entre la VPN ZeroTier y el servidor DHCP, lo que garantiza la asignación coherente de direcciones IP a los clientes incluso cuando están fuera de la red local.

Nuestra misión principal, el enrutamiento del tráfico de red desde los clientes a través de los servidores, se logró con éxito gracias a la implementación cuidadosa de estos componentes. Esto nos proporcionó un mayor control sobre el flujo de datos y nos permitió aplicar políticas de seguridad y de monitorización de manera efectiva. Al enrutar el tráfico a través de los servidores, creamos un punto central de administración y supervisión, facilitando la gestión y la detección de posibles problemas de red.

Además, al optar por el sistema operativo AlmaLinux en nuestros clientes, demostramos nuestra atención a la elección de sistemas seguros y confiables. AlmaLinux es una distribución de Linux respaldada por la comunidad que se centra en la estabilidad y la seguridad, lo que la convierte en una elección inteligente para las máquinas virtuales de nuestros clientes.

Este proyecto ha sido un ejercicio esencial en la configuración y administración de redes en un entorno de nube, utilizando herramientas y tecnologías clave como Ubuntu, Webmin, ZeroTier y AlmaLinux. Hemos logrado nuestro objetivo de enrutar el tráfico

de red desde los clientes para que pase por los servidores, mejorando así la seguridad y la gestión de la red. La combinación de estos elementos proporciona una infraestructura robusta y confiable para las comunicaciones de red, estableciendo un entorno que se adapta a las demandas de las organizaciones modernas en busca de conectividad segura y eficiente. Este proyecto destaca la importancia de una planificación cuidadosa y una implementación precisa en el mundo en constante evolución de la administración de redes y la tecnología de la información.

Por último, es interesante haber añadido una conexión segura también entre los servidores usando una herramienta como es Wireguard, pues se ha podido investigar y probar otra manera de conectar de manera segura que difieren de las típicas opciones que se suele usar como solución VPN.

A modo de reflexión con este proyecto ha sido realmente curioso pues me ha resultado interesante combinar tecnologías actuales dando como resultado una red híbrida o mixta donde hay una parte en local (máquinas RCO) y otra parte alojada en la nube (servidores en la nube). En adición también cabe destacar el uso de nuevas soluciones como son las VPN que se pueden configurar de manera sencilla y proporcionan seguridad a una red como la nuestra. Además, que este tipo de proyectos se pueden usar en redes de cualquier tamaño pues es una solución para enrutar el tráfico de manera elástica, ya que podemos crear más servidores en la nube si nos quedáramos cortos con los dos servidores que ya tenemos, ya sea por banda ancha u otro tipo de problema.

Como se ha apreciado en puntos anteriores se ha conseguido el objetivo principal, ahora comentaremos un poco sobre la red cosas a mejorar y que futuras mejoras se podrían realizar en esta. En cuanto a las tecnologías usadas es posible usar otras opciones tanto para la creación de instancias en la nube como para la interconexión de dispositivos. Una de las cosas a señalar de nuestra red podemos decir que el uso de una redirección a través de una instancia en la nube como la que hemos realizado se ha señalado la complejidad de redirigir el tráfico desde una máquina virtual en local. Esto se debe a que la redirección de un rango amplio de direcciones IP puede hacer que nos quedemos sin conexión a internet a través de nuestra máquina anfitriona. Para ello se debería de direccionar todo tráfico de las RCO asegurándonos de tener conexión con el servidor en la nube de manera que perdamos conexión a internet. A modo de futuras mejoras que se podrían realizar, podríamos hacer que esta infraestructura fuese escalable para ello se podría usar múltiples dispositivos ya sean máquinas virtuales u otro tipo de dispositivos e intentar direccionar el tráfico de estos por nuestros servidores. Para ello podríamos investigar en como probar el escalamiento de esta red mediante el uso de contenedores docker que se conecten a nuestros servidores, además podríamos probar a repartir la carga de los servidores con un balanceador de carga con el fin de repartir las conexiones entre ambos servidores. Con esto podemos probar a implementar un servicio de direccionamiento que podría ser usado por pequeñas, medianas o grandes empresas en sus sistemas, ya que al implementar un sistema escalable haciéndolo idóneo para un entorno profesional por ejemplo donde los servicios suelen trabajar según la demanda. Un ejemplo de la importancia de los sistemas escalables hoy en día se puede observar en cualquier tipo de aplicación ya sea una tienda online de ropa, como Shein, o servicios de streaming como es el caso de Twitch, por ejemplo, donde regularmente la demanda suele ser irregular según ciertos eventos. En el ejemplo de la tienda online de ropa que esta tenga un sistema escalable puede hacer que en día de más ventas como puede ser el "Black Friday" la página pueda atender a millones de peticiones sin sufrir caídas debido a ello; por otro lado, en los servicios de streaming están los casos donde ciertos eventos como puede ser el lanzamiento de un juego nuevo u otros eventos puntuales que hace que millones de personas se conecten puede llegar a hacer que la red se sobrecargue y deje de funcionar; para ello es imprescindible actualmente desarrollar sistemas escalables

y robustos para poder evitar un consumo tanto excesivo de recursos como hacer que un sistema pueda atender a un alto número de clientes ya sea este número de miles o de millones.

Bibliografía

- [1] Amazon Virtual Private Cloud (VPC). Amazon virtual private cloud (vpc) documentation, s.f.
- [2] Documentación ZeroTier, 2023. Disponible en: <https://docs.zerotier.com/zerotier/manual>.
- [3] Jip Kim and Yury Dvorkin. A p2p-dominant distribution system architecture. *IEEE Transactions on Power Systems*, 35(4), 2020.
- [4] Alfonso Cobo Canela. *Servidor multimedia con VPN en Linux*. PhD thesis, Universidad Politécnica de Valencia, Valencia, España, septiembre 2022. Trabajo fin de carrera.
- [5] Pasi Eronen, Yoav Nir, Paul E. Hoffman, and Charlie Kaufman. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 5996, September 2010.
- [6] Mario Sánchez Rubio. *VPN ZeroTier: instalación, configuración y fundamentos teóricos*. PhD thesis, Universitat Politècnica de València, 2022.
- [7] Álvaro Marín García. *Creación de un nodo multi-VPN en la nube para el ámbito empresarial*. PhD thesis, Universitat Politècnica de València, 2023.
- [8] José Miguel Álvarez Vañó. *Modelo Comparativo de Plataformas Cloud y Evaluación de Microsoft Azure, Google App Engine y AmazonEC2*. PhD thesis, Universidad Politécnica de Valencia, Valencia, España, abril 2018. Trabajo fin de carrera.
- [9] Mario Miguel Jaramillo Sizalima. *Despliegue de un cluster Kubernetes altamente disponible en Alibaba Cloud*. PhD thesis, Universidad Politécnica de Valencia, Valencia, España, septiembre 2023. Trabajo fin de carrera.
- [10] Wikipedia contributors. Curve25519 — Wikipedia, the free encyclopedia, 2024. [Online; accessed 12-April-2024].
- [11] Jan Just Keijser. *OpenVPN Cookbook*. Packt Publishing Ltd, 2017.
- [12] Vijay Bollapragada, Mohamed Khalid, and Scott Wainner. *IPSec VPN design*. Cisco Press, 2005.
- [13] Servidor DHCP en webmin, 2023. Disponible en: https://doxfer.webmin.com/Webmin/DHCP_Server.
- [14] Sandra Ortega Martorell and Liusbety Canino Gutiérrez. Protocolo de seguridad ssl. *Ingeniería Industrial*, 27(2-3):57–62, 2006.
- [15] P. Francis K. Egevang. The ip network address translator. Technical report, 1994. Disponible en <https://www.rfc-editor.org/rfc/pdf/rfc/rfc1631.txt.pdf>.

- [16] Wikipedia. Secure copy — wikipedia, la enciclopedia libre, 2024. [Internet; descargado 23-mayo-2024].

APÉNDICE A

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.			X	
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.			X	
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.	X			
ODS 11. Ciudades y comunidades sostenibles.			X	
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.		X		

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

A.1 Objetivos de desarrollo sostenible(ODS)

En nuestro trabajo, nos comprometemos a contribuir activamente al desarrollo de proyectos que cumplan los Objetivos de desarrollo sostenible establecidos por las Naciones Unidas. Reconocemos la importancia de abordar los desafíos globales para garantizar un futuro próspero y equitativo para todos. Con ello con este proyecto, buscamos promover el desarrollo económico, social y ambiental de manera sostenible. A continuación, detallaremos cómo nuestras iniciativas están alineadas con puntos específicos de los ODS, demostrando nuestro compromiso con un mundo mejor y más sostenible para las generaciones presentes y futuras.

1. Trabajo decente y crecimiento económico.

Podemos decir que este tipo de tecnologías que convergen como en nuestro caso tecnologías tradicionales y modernas con infraestructura en la nube pueden ayudar a crear nuevos puestos de trabajo relacionados con este tipo de tecnologías, y ya no solo eso si no que, por ejemplo, si se tiene conocimientos amplios en el sector de sistemas en la nube se puede dar el caso de contribuir a conseguir trabajos decentes relacionados con esta tecnología, además que no es necesario estar presente en cualquier sitio con una conexión estable a internet sería posible trabajar desde cualquier parte del planeta. Con esto se podría contribuir al crecimiento económico de las zonas que estén en desarrollo ya que no hace falta ni infraestructura ni se necesita de una gran inversión para poder en marcha proyectos en cualquier parte del mundo. Por otro lado, el tipo de tecnologías son interesantes actualmente pues el uso de la nube está presente en todo tipo de aplicaciones, sistemas, etcétera; además son cada vez más los proyectos que implementan sus sistemas en la nube debido al coste inicial y a la planificación del pago por uso.

2. Industrial, innovación e infraestructura.

En este punto podemos destacar que nuestro proyecto promueve el desarrollo de la innovación y la infraestructura. Ambos puntos están relacionados pues hasta hoy en día, no es muy común para la gente crear redes de manera virtual con máquinas en la nube. La innovación está en el uso de proveedores de nube pública (Amazon Web Services en nuestro caso) y en la infraestructura está en que toda la parte de ella nos la "ahorramos" debido a que todo lo que creemos en la nube está alojado en las infraestructuras de nuestro proveedor. Gracias a esto último, el usuario solo será solo el responsable de administrar cada recurso y la parte de la infraestructura tendrá como responsable a la empresa que nos proporciona el servicio ya sea Amazon, Google, Microsoft, etc. Con ello todo componente nos lo ahorraremos y gracias a eso nos podremos ahorrar el gasto inicial que conlleva el comprar todo el equipo inicial que se necesitaría si se quisiese hacer de manera tradicional.

3. Reducción de las desigualdades.

Nuestro enfoque en la tecnología basada en la nube no solo impulsa la eficiencia y la innovación, sino que también desempeña un papel crucial en la reducción de las diferencias socio-económicas en todo el mundo. Al utilizar componentes en la nube, nuestro proyecto ayuda a nivelar esta diferencia al hacer que la tecnología sea accesible incluso en entornos con recursos limitados. La infraestructura en la nube permite significativa reducción en una inversión inicial como se ha señalado

anteriormente y se la relaciona con este punto. Esto significa que cualquiera, independientemente de su ubicación o situación financiera, puede desplegar servicios y recursos tecnológicos, democratizando el acceso a oportunidades y posibilitando una participación equitativa en la economía digital global.

Por otro lado, al reducir la barrera de entrada para emprendedores y pequeñas empresas, fomentamos la creación de empleo y el crecimiento económico inclusivo. La flexibilidad y escalabilidad inherentes a la computación en la nube también permiten a las comunidades marginadas acceder a servicios esenciales, como educación en remoto, atención médica remota y servicios gubernamentales digitales. Al capacitar a individuos y comunidades con herramientas tecnológicas y al reducir la desigualdad en varios aspectos socio-económicos podemos decir que encaja en este objetivo de desarrollo sostenible.

4. Producción y consumo responsables.

Todo lo que este relacionado con la nube se relaciona con este objetivo de desarrollo sostenible; esto se debe a que el uso de este tipo de infraestructura hace que toda la infraestructura necesaria para cualquier proyecto se reduzca a elementos necesarios. Ya sean servidores, racks, cables, alquiler de instalaciones, consumo energético de la instalación, etcétera; todo eso no sería necesario pues todo esta alojado físicamente en las instalaciones de nuestro proveedor.

Con esto podemos reducir nuestra huella de carbono pues no es necesario comprar materiales y el consumo energético sería nulo, esto lo hace idóneo para zonas con recursos limitados. Además, si necesitáramos probar cualquier tipo de servicio si lo hacemos en la nube podemos terminar cualquier recurso o si llega a fallar cualquier proyecto, no generaríamos residuos pues son elementos virtuales.