

UNIVERSIDAD RAFAEL LANDÍVAR

FACULTAD DE INGENIERÍA

REDES II

SECCIÓN 1 VESPERTINA

MGTR. DENNIS JAVIER DONIS DE LEÓN

PROYECTO FINAL

Julio Anthony Engels Ruiz Coto 1284719

Eddie Alejandro Girón Carranza 1307419

Rafael Andrés Alvarez Mazariegos 1018419

GUATEMALA DE LA ASUNCIÓN, NOVIEMBRE 15 DE 2024

ÍNDICE

1. RESUMEN EJECUTIVO.....	4
2. DESARROLLO DE LA INVESTIGACIÓN.....	6
2.1 SERVICIOS WEB (JULIO RUIZ).....	6
2.1.1 HISTORIA.....	6
2.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES.....	6
2.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO.....	7
2.2 CERTIFICADOS DIGITALES (JULIO RUIZ).....	8
2.2.1 HISTORIA.....	8
2.2.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES.....	8
2.2.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO.....	9
2.3 BASE DE DATOS (RAFAEL ALVAREZ).....	10
3.1.1 HISTORIA.....	10
3.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES.....	10
3.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO.....	11
2.4 VPN (RAFAEL ALVAREZ).....	12
4.1.1 HISTORIA.....	12
4.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES.....	12
4.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO....	13
2.4 FIREWALL (Eddie Girón).....	14
2.4.1 HISTORIA.....	14
2.4.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES.....	14
2.5 SISTEMA DE MONITOREO (Eddie Girón).....	16
5.1.1 HISTORIA.....	16
5.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES.....	16
5.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO....	17
2.5.2 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO.....	18
2.6 ALCANCES POR SERVICIO (SE HIZO TODOS LOS INTEGRANTES).....	19
3. CONFIGURACIÓN DEL SERVICIO.....	25
3.1 SERVICIOS WEB:.....	25
3.2 CERTIFICADOS:.....	29
3.3 BASE DE DATOS:.....	30
3.4 VPN:.....	32
3.5 FIREWALL.....	34
3.6 ZABBIX.....	39
4. SIMULACIÓN EN GNS3.....	40
5. BIBLIOGRAFÍA.....	41

1. RESUMEN EJECUTIVO

El presente proyecto tiene como objetivo la implementación integral de una red que ofrezca una variedad de servicios esenciales, tanto internos como públicos, emulando un entorno corporativo real. Este trabajo será desarrollado en grupos de tres estudiantes de la misma sección de laboratorio, aplicando los conceptos aprendidos en clase y realizando una investigación exhaustiva para abordar los desafíos planteados.

La red implementada proporcionará servicios como DHCP para la asignación automática de direcciones IPv4 a los clientes internos, y DNS interno y público para la resolución de nombres dentro y fuera de la red. Se establecerá un servidor de correo electrónico que permitirá la comunicación interna entre los miembros del grupo y externa con otros dominios de grupo. Un firewall perimetral será configurado para controlar el acceso entre las redes interna, DMZ y externa, aplicando reglas de seguridad estrictas que permitan sólo el tráfico autorizado.

En cuanto a los servicios web, se publicará un sitio básico que mostrará datos generales del grupo obtenidos a través de un web service o API conectada a una base de datos. Este sitio será accesible mediante un dominio único y estará configurado para alta disponibilidad mediante un balanceador de carga que distribuirá el tráfico entre al menos dos nodos. Las conexiones al sitio web se asegurarán mediante certificados SSL autofirmados generados por un servidor dedicado.

El servidor de base de datos almacenará información de los integrantes del grupo y será accesible únicamente a través de los nodos web back-end, reforzando la seguridad y privacidad de los datos. Además, se implementará un servidor VPN que permitirá a los miembros del grupo conectarse de forma segura a la red interna desde redes externas mediante túneles cifrados, facilitando el acceso remoto a los recursos internos.

Para garantizar el correcto funcionamiento y la seguridad de la red, se configurará un sistema de monitoreo utilizando Zabbix. Este sistema tendrá acceso vía SNMP a todos los dispositivos activos de la red, como switches y routers, para generar estadísticas de rendimiento y monitorear proactivamente los recursos de hardware y la disponibilidad de los servicios. Esto permitirá una gestión eficiente y la detección temprana de posibles incidencias.

La estructura del proyecto incluye la configuración de la red interna y pública, asignando direcciones IPv4 estáticas a los servidores internos y utilizando NAT en el firewall perimetral para asignar direcciones públicas a los servicios externos. Se establecerán reglas de acceso que restringirán el acceso a los servidores públicos únicamente a través de los puertos y servicios permitidos, ubicando estos servidores en una DMZ separada de la red interna para aumentar la seguridad.

Todos los dispositivos sincronizarán su hora con un servicio NTP en red, asegurando la coherencia temporal en toda la infraestructura. El servicio de correo electrónico permitirá la comunicación efectiva dentro del dominio del grupo y entre diferentes dominios de grupo, facilitando la colaboración y el intercambio de información.

Al finalizar el proyecto, se espera que los estudiantes demuestren una comprensión profunda y práctica de la implementación y gestión de una red compleja con múltiples servicios. El éxito del proyecto se evidenciará mediante un informe detallado y pruebas funcionales presentadas en video, reflejando la capacidad para integrar diversas tecnologías y garantizar un entorno de red seguro, eficiente y altamente disponible.

Este proyecto no sólo consolidará los conocimientos teóricos adquiridos, sino que también fortalecerá competencias en seguridad informática, trabajo en equipo y gestión eficiente de recursos de red. Los estudiantes desarrollarán habilidades en monitoreo proactivo y serán capaces de asegurar la alta disponibilidad de servicios críticos, preparándose para enfrentar desafíos en entornos profesionales reales.

2. DESARROLLO DE LA INVESTIGACIÓN

2.1 SERVICIOS WEB (JULIO RUIZ)

2.1.1 HISTORIA

- Origen de los Servicios Web: Surgieron a finales de los años 90 y principios de los 2000 como una solución para permitir la comunicación entre aplicaciones diferentes a través de internet.
- Evolución:
 - Primeras Implementaciones: Utilizaban protocolos como SOAP (Simple Object Access Protocol) y WSDL (Web Services Description Language).
 - Auge de REST: A medida que la necesidad de servicios más ligeros y flexibles creció, REST (Representational State Transfer) se convirtió en el estilo arquitectónico predominante.
 - Servicios Web Modernos: Integración con microservicios, APIs, y el uso de protocolos como GraphQL para optimizar las consultas de datos.

2.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES

- Protocolos:
 - HTTP/HTTPS: Base para la comunicación en la web; HTTPS añade una capa de seguridad mediante SSL/TLS.
 - SOAP: Protocolo basado en XML para intercambiar información estructurada.
 - REST: Estilo arquitectónico que utiliza HTTP y enfatiza la escalabilidad y simplicidad.
 - GraphQL: Lenguaje de consulta para APIs, desarrollado por Facebook.
- Estándares:
 - WSDL: Lenguaje para describir servicios web y cómo acceder a ellos.
 - UDDI: Directorio para publicar y descubrir servicios web.
 - OpenAPI/Swagger: Especificación para describir APIs RESTful.
- Organismos Reguladores:
 - W3C (World Wide Web Consortium): Desarrolla estándares web como SOAP y WSDL.
 - OASIS (Organization for the Advancement of Structured Information Standards): Trabaja en estándares como UDDI.

2.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO

- Componentes de un Servicio Web:
 - Servidor Web: Software que maneja las solicitudes HTTP/HTTPS (e.g., Apache, Nginx).
 - API: Interfaz que define cómo interactuar con el servicio.
 - Base de Datos: Almacena y gestiona los datos que el servicio web utiliza.
 - Balanceador de Carga: Distribuye el tráfico entre múltiples servidores para mejorar la disponibilidad.
 - Proxy Inverso: Gestiona las solicitudes entrantes y las dirige al servidor apropiado.
- Funcionamiento:
 - Un cliente realiza una solicitud HTTP/HTTPS al servidor web.
 - El servidor procesa la solicitud, posiblemente interactuando con una base de datos.
 - El servidor devuelve una respuesta al cliente, que puede ser en formato HTML, JSON, XML, etc.
 - En arquitecturas de alta disponibilidad, el balanceador de carga distribuye las solicitudes entre varios servidores para evitar sobrecargas.
 - El proxy inverso puede manejar múltiples dominios y rutas, redirigiendo las solicitudes al servicio correspondiente.

2.2 CERTIFICADOS DIGITALES (JULIO RUIZ)

2.2.1 HISTORIA

- Inicios de la Seguridad en Internet:
 - A mediados de los 90, con el crecimiento de internet, surgió la necesidad de asegurar las comunicaciones.
- Desarrollo de SSL:
 - SSL (Secure Sockets Layer) fue desarrollado por Netscape en 1995 para asegurar las conexiones web.
- Evolución a TLS:
 - SSL evolucionó a TLS (Transport Layer Security), estandarizado por la IETF para mejorar la seguridad y corregir vulnerabilidades.
- Actualidad:
 - Uso generalizado de certificados digitales para asegurar no solo sitios web, sino también correos electrónicos, software, etc.

2.2.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES

- Protocolos:
 - SSL/TLS: Proporcionan cifrado y autenticación en las comunicaciones.
 - OCSP (Online Certificate Status Protocol): Verifica el estado de revocación de un certificado.
- Estándares:
 - X.509: Estándar para la estructura de los certificados digitales.
 - PKCS (Public Key Cryptography Standards): Conjunto de estándares para criptografía de clave pública.
- Organismos Reguladores:
 - IETF (Internet Engineering Task Force): Desarrolla y promueve protocolos y estándares como TLS.
 - CA/Browser Forum: Alianza de Autoridades Certificadoras y fabricantes de navegadores que define estándares para la emisión de certificados.
 - Autoridades Certificadoras (CAs): Organizaciones que emiten certificados digitales (e.g., Let's Encrypt, DigiCert).

2.2.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO

- Componentes de un Sistema de Certificados:
 - Certificado Digital: Archivo electrónico que vincula una clave pública con la identidad de una entidad.
 - Autoridad Certificadora (CA): Entidad que emite y firma certificados digitales.
 - Infraestructura de Clave Pública (PKI): Conjunto de políticas, procedimientos y tecnologías para gestionar claves y certificados.
- Funcionamiento:
 - Generación de Claves: Se crean un par de claves (pública y privada).
 - Solicitud de Certificado: El solicitante envía una CSR (Certificate Signing Request) a una CA.
 - Emisión del Certificado: La CA verifica la identidad y emite el certificado firmado.
 - Implementación: El certificado se instala en el servidor para habilitar conexiones seguras.
 - Validación: Los clientes verifican la validez del certificado al conectarse, asegurando que la comunicación es segura.
 - Renovación y Revocación: Los certificados tienen una vigencia limitada y pueden ser revocados si se comprometen.

2.3 BASE DE DATOS (RAFAEL ALVAREZ)

En la presente parte se detalla sobre el uso de mongoDB como servicio de base de datos dentro de una infraestructura de red gestionada por pfSense. MongoDB, una base de datos NoSQL de alto rendimiento, ha sido seleccionada por su capacidad para manejar grandes volúmenes de datos no estructurados, su flexibilidad y escalabilidad. Se abarca la evolución histórica de MongoDB, los protocolos y estándares que regulan su funcionamiento, así como una descripción de sus componentes y su operativa.

3.1.1 HISTORIA

- Origen de MongoDB: Se creó en 2007 por la compañía 10gen (hoy en día conocida como MongoDB Inc.) en respuesta a las restricciones de las bases de datos relacionales convencionales ante el incremento en el volumen y diversidad de datos producidos por aplicaciones actuales. Inspirado en el modelo de documentos de CouchDB, MongoDB presenta una estructura versátil fundamentada en documentos BSON (una versión binaria de JSON), lo que facilita el almacenamiento de datos semi-estructurados y la añadida de información de forma eficaz.
- Evolución de MongoDB
 - Primeras Implementaciones
 - 2007: Lanzamiento inicial de MongoDB por 10gen.
 - 2009: Introducción de la funcionalidad de replicación, permitiendo la creación de conjuntos de réplicas para alta disponibilidad.
 - Auge y Expansión
 - 2013: MongoDB Inc. realiza una oferta pública inicial (IPO), consolidando su posición en el mercado tecnológico.
 - 2014: Lanzamiento de MongoDB Atlas, un servicio de base de datos como servicio (DBaaS) en la nube, facilitando la gestión y escalabilidad de bases de datos MongoDB.
 - Servicios Web Modernos
 - Integración con microservicios, permitiendo que MongoDB funcione de manera eficiente en arquitecturas distribuidas.
 - Implementación de GraphQL como una alternativa para optimizar las consultas de datos, mejorando la eficiencia en la recuperación de información.

3.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES

- Protocolos:
 - TCP/IP (Transmission Control Protocol/Internet Protocol)

- HTTP/HTTPS (HyperText Transfer Protocol Secure)
 - SSL/TLS (Secure Sockets Layer/Transport Layer Security)
- Estándares:
 - JSON (JavaScript Object Notation)
 - BSON (Binary JSON)
 - ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)
 - CQL (Cassandra Query Language)
- Organismos Reguladores:
 - Open Source Initiative (OSI)
 - International Organization for Standardization (ISO):

3.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO

- Componentes de un mongoDB
 - MongoDB Server (mongod): El núcleo de la base de datos, responsable de gestionar las operaciones de lectura y escritura, el almacenamiento de datos y la replicación. Se ejecuta como un proceso que puede operar en diferentes configuraciones, incluyendo servidores individuales, réplicas y clusters distribuidos.
 - MongoDB Shell (mongosh): Una interfaz de línea de comandos que permite a los administradores y desarrolladores interactuar directamente con la base de datos mediante comandos y scripts.
 - Drivers: Bibliotecas específicas de cada lenguaje de programación (como Python, Java, Node.js) que facilitan la conexión y la interacción con MongoDB desde aplicaciones externas.
 - MongoDB Compass: Una interfaz gráfica para explorar y gestionar datos, permitiendo a los usuarios visualizar esquemas, realizar consultas y optimizar el rendimiento de la base de datos.
- Funcionamiento del Servicio
 - MongoDB opera bajo un modelo de documentos, donde cada registro en la base de datos se almacena como un documento BSON en una colección. Este enfoque permite una mayor flexibilidad en la estructura de los datos, facilitando la evolución y adaptación a cambios en los requisitos de la aplicación sin necesidad de migraciones complejas.

2.4 VPN (RAFAEL ALVAREZ)

Implementación y configuración de un Servidor VPN dentro de una infraestructura de red gestionada por pfSense. El objetivo principal es permitir que los miembros de un grupo se conecten de manera segura desde redes públicas hacia la red local mediante un túnel punto a punto cifrado. Se aborda la evolución histórica de las tecnologías VPN, los protocolos y estándares que regulan su funcionamiento, así como una descripción exhaustiva de sus componentes y operativa.

4.1.1 HISTORIA

- Origen: Las Redes Privadas Virtuales (VPN) surgieron en la década de 1990 como una solución para permitir la comunicación segura a través de redes públicas, como Internet. Inicialmente, las VPN se utilizan para conectar sucursales de empresas distantes geográficamente, permitiendo el acceso seguro a recursos compartidos y facilitando la colaboración entre empleados.
- Evolución de VPN
 - Primeras Implementaciones
 - Década de 1990: Introducción de protocolos como Point-to-Point Tunneling Protocol (PPTP) y Layer 2 Tunneling Protocol (L2TP) para establecer túneles seguros sobre Internet.
 - 1996: Estándar IPsec (Internet Protocol Security) es desarrollado para asegurar las comunicaciones a nivel de red, proporcionando autenticación y cifrado.
 - Auge y Expansión
 - 2000s: Creciente demanda de soluciones VPN debido al aumento del teletrabajo y la necesidad de acceder a recursos corporativos de manera remota.
 - 2001: Implementación de SSL/TLS VPNs, que utilizan el protocolo HTTPS para establecer conexiones seguras, facilitando el acceso a aplicaciones web de manera segura.
 - Servicios Web Modernos
 - 2010s: Integración de VPN con tecnologías de nube, permitiendo conexiones seguras a infraestructuras en la nube.

4.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES

- Protocolos:
 - IPsec (Internet Protocol Security)
 - OpenVPN

- WireGuard
 - PPTP (Point-to-Point Tunneling Protocol)
 - L2TP (Layer 2 Tunneling Protocol)
 - SSL/TLS (Secure Sockets Layer/Transport Layer Security)
- Estándares:
 - RFC 4301: Define el marco de seguridad para IPSec.
 - RFC 7425: Específica WireGuard como protocolo VPN.
 - IEEE 802.1X: Estándar para control de acceso a redes, utilizado en algunas configuraciones VPN para autenticación.
 - OpenVPN Specifications: Documentación oficial que define el funcionamiento y configuración de OpenVPN.
- Organismos Reguladores:
 - IETF (Internet Engineering Task Force)
 - IEEE (Institute of Electrical and Electronics Engineers)
 - ISO/IEC (International Organization for Standardization/International Electrotechnical Commission)
 - OpenVPN Project

4.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO

- Componentes de una VPN
 - Un Servidor VPN se compone de varios componentes esenciales que colaboran para asegurar una conexión segura y eficaz. Primero que nada, el propio Servidor VPN puede ser un equipo dedicado o una máquina virtual que opera software especializado como OpenVPN o WireGuard. Este servidor tiene la tarea de autenticar a los usuarios, crear túneles de seguridad y administrar el tráfico de red entrante y saliente. Los Clientes VPN son las aplicaciones o dispositivos que se instalan en los dispositivos de los usuarios para establecer conexiones seguras con la red local desde lugares alejados.
- Funcionamiento del Servicio
 - Establecimiento de la Conexión VPN
 - Negociación del Túnel Seguro
 - Acceso a la Red Local
 - Gestión del Tráfico y Seguridad

2.4 FIREWALL (Eddie Girón)

Implementación y configuración de un Firewall dentro de una infraestructura de red gestionada por pfSense. El objetivo principal es proteger la red interna de posibles amenazas externas, controlando el tráfico entrante y saliente mediante reglas de filtrado específicas, segmentación y políticas de acceso. Se aborda la evolución histórica de las tecnologías de firewall, los protocolos y estándares que regulan su funcionamiento, así como una descripción exhaustiva de sus componentes y operativa.

2.4.1 HISTORIA

- Origen: Los firewalls se originaron a finales de la década de 1980, cuando los primeros ataques cibernéticos comenzaron a surgir. La primera generación de firewalls eran dispositivos de hardware que inspeccionaban los paquetes de red entrantes y salientes para determinar si se permitían o se bloqueaba. Con la evolución de la tecnología y la complejidad de los ataques cibernéticos, los firewalls también han evolucionado para ofrecer más funcionalidades y opciones de seguridad. Actualmente existen diferentes tipos de firewalls, como los de próxima generación (NGFW) que utilizan tecnologías avanzadas de detección de amenazas, y los firewalls de aplicaciones web (WAF) que protegen las aplicaciones web y los servidores.
- Evolución de los Firewall
 - Primeras Implementaciones
 - Década de 1980: Introducción de los firewalls de filtrado de paquetes, que evaluaban paquetes según reglas estáticas.
 - 1990s: Aparición de los firewalls de inspección de estado, que rastrean conexiones activas para mejorar la seguridad y el rendimiento.
 - Auge y Expansión
 - 2000s: Integración de firewalls con tecnologías de prevención de intrusiones (IPS) y detección de intrusiones (IDS), permitiendo una defensa más robusta contra amenazas avanzadas.
 - Firewalls de Próxima Generación
 - 2010s: Aparición de los NGFW (Next-Generation Firewalls) que ofrecen inspección profunda de paquetes, control de aplicaciones y protección contra amenazas avanzadas.
 - 2010s: Integración con servicios en la nube y protección contra amenazas avanzadas.

2.4.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES

- Protocolos:
 - TCP/IP
 - ICMP

- UDP
 - HTTP/HTTPS
 - FTP
 - DNS
 - SNMP
- Estándares:
 - ISO/IEC 27001: Estándares de seguridad de la información
 - NIST SP 800-41: Directrices para firewalls y políticas de firewall
 - PCI DSS: Requisitos de seguridad para el procesamiento de tarjetas de pago
 - Common Criteria (ISO/IEC 15408): Evaluación de seguridad de productos
- Organismos Reguladores:
 - IETF (Internet Engineering Task Force)
 - ISO (International Organization for Standardization)
 - NIST (National Institute of Standards and Technology)
 - PCI SSC (Payment Card Industry Security Standards Council)

4.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO

- Componentes de un Firewall
 - Filtrado: Componente principal que examina y filtra el tráfico según las reglas establecidas
 - Reglas: Conjunto de políticas que definen qué tráfico se permite o bloquea
 - Módulo de inspección de estado: Mantiene registro de las conexiones activas
 - Sistema de logs: Registra eventos y actividades del firewall
 - Interfaz de administración: Permite configurar y gestionar el firewall
 - Módulos de seguridad adicionales: IPS, antivirus, filtrado web, etc.
- Funcionamiento del Servicio
 - Inspección de Paquetes
 - Análisis de cabeceras de paquetes
 - Verificación contra reglas establecidas
 - Control de estado de conexiones
 - Políticas de Seguridad
 - Definición de zonas de seguridad
 - Reglas de acceso entre zonas
 - Políticas de NAT y enrutamiento
 - Monitorización y Registro
 - Registro de eventos de seguridad
 - Alertas de amenazas
 - Reportes de actividad

2.5 SISTEMA DE MONITOREO (Eddie Girón)

Implementación y configuración de un sistema de monitorización Zabbix dentro de una infraestructura de red. El objetivo principal es supervisar proactivamente el estado y rendimiento de los equipos activos de red, interfaces y servicios mediante el protocolo SNMP, permitiendo generar estadísticas de salud y detectar tempranamente posibles problemas en la infraestructura.

5.1.1 HISTORIA

- Origen: Los servidores de monitoreo, como Zabbix, surgieron en la década de 1990 como soluciones para supervisar el rendimiento de redes y servicios en tiempo real, proporcionando visibilidad sobre la infraestructura de TI de las organizaciones. Inicialmente, se utilizaron para monitorear equipos individuales y pequeñas redes, con el objetivo de detectar fallos antes de que afectarán los servicios críticos.
- Evolución de VPN
 - Primeras Implementaciones
 - Década de 1990: Se empezaron a implementar herramientas básicas de monitoreo, como sistemas de notificación por correo electrónico y monitoreo de servidores individuales.
 - Crecimiento de la demanda de soluciones de monitoreo más robustas, en especial para redes empresariales más grandes. Emergen plataformas como Nagios y Zabbix, las cuales introducen una interfaz centralizada para la gestión de la infraestructura TI.
 - Zabbix se consolida como una de las soluciones de monitoreo más populares, con soporte para monitoreo en tiempo real, integración con tecnologías de nube y capacidades de monitoreo proactivo.

5.1.2 PROTOCOLOS, ESTÁNDARES Y ORGANISMOS REGULADORES

- Protocolos:
 - SNMP (Simple Network Management Protocol): Protocolo fundamental para la gestión y monitoreo de dispositivos de red como switches y routers.
 - ICMP (Internet Control Message Protocol): Usado para comprobar la conectividad y latencia de la red.
 - JMX (Java Management Extensions): Utilizado para monitorear aplicaciones basadas en Java.
 - SSH (Secure Shell): Usado para monitorear servidores y sistemas mediante la ejecución de comandos remotos.
- Estándares:

- RFC 1157: Define el protocolo SNMP, que es uno de los más utilizados en plataformas de monitoreo como Zabbix.
- RFC 1911: Especifica el uso de SNMPv2 para monitoreo en redes.
- ISO/IEC 27001: Estándar de gestión de seguridad de la información que puede guiar el monitoreo de infraestructuras críticas.
- Organismos Reguladores:
 - IETF (Internet Engineering Task Force): Establece los estándares para los protocolos de red, incluidos SNMP y otros utilizados para monitoreo.
 - IEEE (Institute of Electrical and Electronics Engineers): Regula las normas para la comunicación y gestión de redes de TI, influenciando el desarrollo de herramientas de monitoreo.
 - ISO/IEC: Desarrolla estándares que impactan en el monitoreo de la infraestructura tecnológica, como los relacionados con la seguridad y disponibilidad de la información.

5.1.3 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO

- Componentes de una VPN
 - Servidor Zabbix: Es el componente principal que se encarga de recolectar, procesar y almacenar los datos de monitoreo. Este servidor debe estar configurado para acceder a los equipos activos de red de la organización (switches, routers, servidores).
 - Agentes Zabbix: Son los componentes instalados en los dispositivos de la red que recolectan información y la envían al servidor Zabbix. Los agentes pueden recolectar métricas como el uso de CPU, memoria, espacio en disco, entre otros.
 - Frontend Zabbix: Interfaz web que permite a los administradores visualizar los datos de monitoreo y configurar el sistema. Desde aquí se gestionan alertas, se visualizan gráficos y se analizan los eventos.
 - Base de Datos Zabbix: Almacena toda la información de monitoreo, incluyendo datos históricos, configuraciones y registros de eventos.
- Funcionamiento del Servicio
 - El servicio de monitoreo obtiene acceso a los equipos activos de la red utilizando el protocolo SNMP. Esto permite recolectar estadísticas sobre la salud de los dispositivos de red, como el estado de las interfaces, el rendimiento de los routers, y la utilización de recursos.

2.5.2 DESCRIPCIÓN DE COMPONENTES Y FUNCIONAMIENTO DEL SERVICIO

- Servidor Zabbix
 - Motor principal de monitorización
 - Base de datos para almacenamiento de métricas
 - Interfaz web de administración
- Agentes SNMP
 - Configuración en equipos activos (switches, routers)
 - MIBs específicas por fabricante
- Sistema de Alertas
 - Notificaciones por correo, SMS o mensajería
 - Escalamiento de incidentes
- Funcionamiento del Servicio
 - Monitoreo
 - Supervisión continua de interfaces de red
 - Estado operativo (up/down)
 - Utilización de ancho de banda
 - Errores y descartes de paquetes
 - Monitorización de servicios críticos
 - Disponibilidad de servicios
 - Tiempos de respuesta
 - Estado de procesos
 - Estadísticas de Salud de Equipos
 - Uso de CPU y memoria
 - Temperatura y estado de hardware

2.6 ALCANCES POR SERVICIO (SE HIZO TODOS LOS INTEGRANTES)

2.6.1 DHCP

Historia: El protocolo DHCP (Dynamic Host Configuration Protocol) es un protocolo de red utilizado para proporcionar configuración automática de direcciones IP, máscaras de subred, servidores DNS y puertas de enlace predeterminadas a dispositivos en una red. Su principal función es simplificar la configuración de red, especialmente en redes grandes. DHCP permite que los dispositivos se configuran automáticamente con direcciones IP, máscaras de red, puertas de enlace predeterminadas y servidores DNS, en lugar de tener que configurar manualmente cada dispositivo. Al utilizar DHCP, se pueden evitar errores de configuración manual, se puede ahorrar tiempo en la asignación de direcciones IP y se puede facilitar la administración de la red al reducir la carga de trabajo de los administradores. También permite la reutilización eficiente de direcciones IP, lo que p

2.6.2 DNS

Historia: El DNS (Sistema de Nombres de Dominio) utiliza el puerto 53 y su función es la de traducir nombres de dominio en direcciones IP. En el proyecto, se utilizarán dos servidores DNS: uno interno y uno público. El servidor DNS interno se encargará de traducir los nombres de dominio de los recursos internos de la red, como servidores y estaciones de trabajo. Su objetivo es proporcionar una manera fácil de acceder a los recursos en la red, en lugar de tener que recordar direcciones IP complejas. Por otro lado, el servidor DNS público se encargará de traducir los nombres de dominio de recursos que se encuentran en internet, como páginas web y servicios en línea. Su objetivo es proporcionar a los usuarios una manera fácil de acceder a recursos externos, como sitios web populares y servicios en línea.

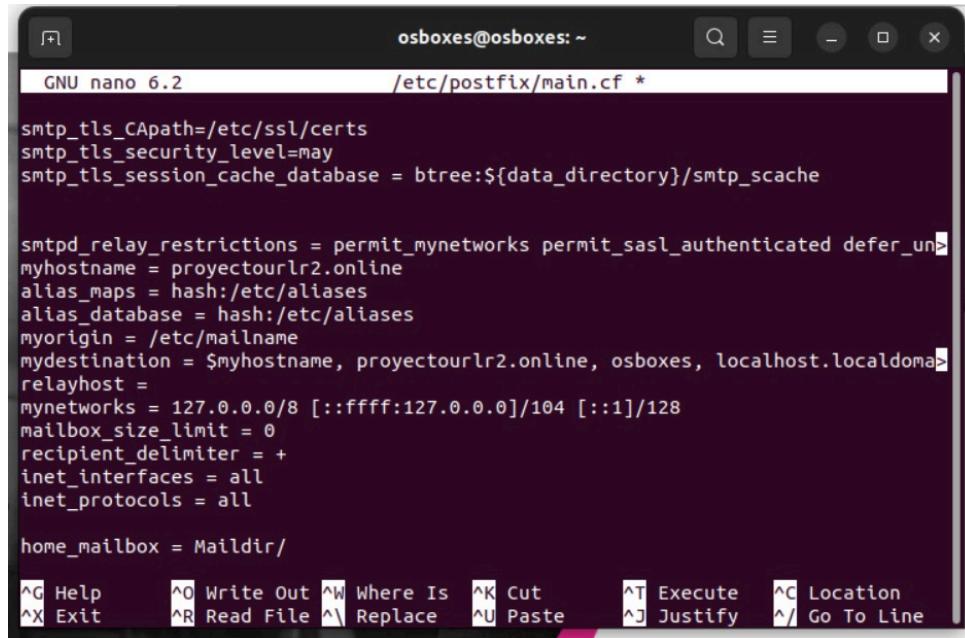
2.6.3 SERVICIO NTP

2.6.4 CORREO ELECTRÓNICO

Historia Para llevar a cabo la configuración de se utilizaron los siguientes servicios: postfix, dovecot y thunderbird: Número de puerto: SMTP (25) y IMAP (143) para MailServer. Los objetivos principales de un servidor de correo electrónico son:

- Permitir la recepción y envío de correos electrónicos tanto dentro como fuera de la red de la empresa.
- Proporcionar un sistema de almacenamiento y gestión de correos electrónicos.
- Garantizar la seguridad de la información transmitida a través del correo electrónico, mediante la implementación de medidas de autenticación y encriptación.
- Facilitar la colaboración y comunicación entre los usuarios de la red, ya que el correo electrónico es uno de los principales medios de comunicación utilizados en el ámbito laboral.

En cuanto a Thunderbird, es un cliente de correo electrónico que se puede utilizar para acceder a las cuentas de correo electrónico en el servidor. No utiliza un número de puerto específico ya que se comunica con el servidor a través de los protocolos de correo electrónico estándar (SMTP, POP3 o IMAP). Su objetivo es proporcionar una interfaz fácil de usar para que los usuarios puedan acceder y gestionar sus correos electrónicos desde sus dispositivos.



The screenshot shows a terminal window titled "osboxes@osboxes: ~" running the "nano" text editor. The file being edited is "/etc/postfix/main.cf". The content of the file is as follows:

```
GNU nano 6.2          /etc/postfix/main.cf *

smtpd_tls_CApth=/etc/ssl/certs
smtpd_tls_security_level=may
smtpd_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = projectourl2.online
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, projectourl2.online, osboxes, localhost.localdomain
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter =
inet_interfaces = all
inet_protocols = all

home_mailbox = Maildir/

^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^Y Replace  ^U Paste    ^J Justify  ^/ Go To Line
```

Instalación de Postfix:

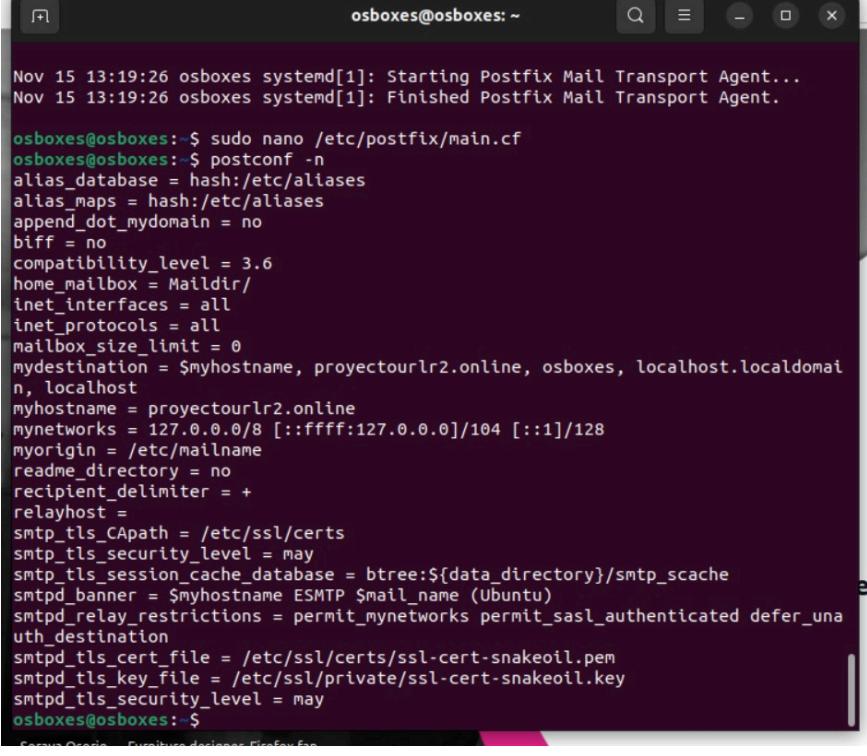
Postfix es un Agente de Transferencia de Correo (MTA) que se encarga de enviar y recibir correos electrónicos.

Instalación:

```
sudo apt-get update
```

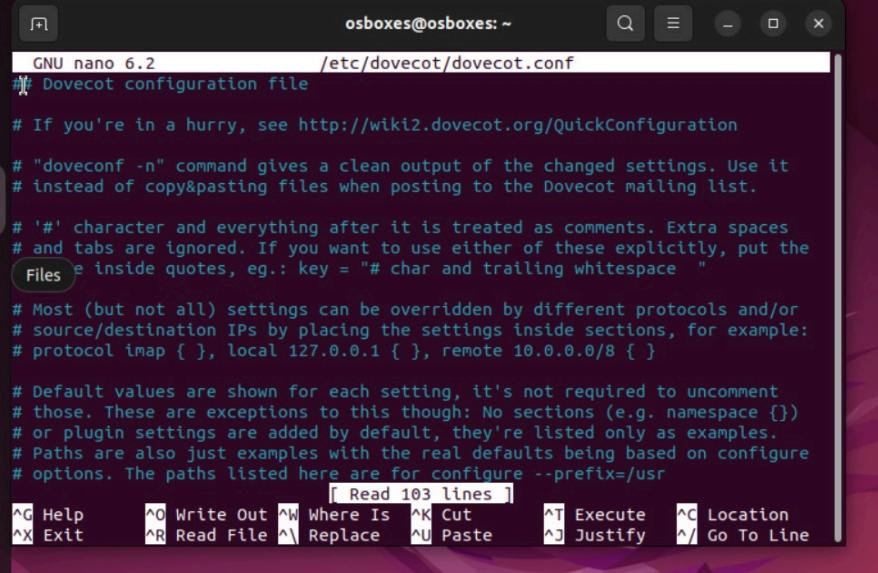
```
sudo apt-get install postfix
```

- Configuración durante la instalación:
 - Selecciona "Sitio de Internet" cuando se te pregunte el tipo de configuración.
 - Ingresá el nombre de dominio que utilizarás para los correos electrónicos (por ejemplo, projectourl2.online).



```
Nov 15 13:19:26 osboxes systemd[1]: Starting Postfix Mail Transport Agent...
Nov 15 13:19:26 osboxes systemd[1]: Finished Postfix Mail Transport Agent.

osboxes@osboxes: $ sudo nano /etc/postfix/main.cf
osboxes@osboxes: $ postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
append_dot_mydomain = no
biff = no
compatibility_level = 3.6
home_mailbox = Maildir/
inet_interfaces = all
inet_protocols = all
mailbox_size_limit = 0
mydestination = $myhostname, proyecto"url2.online, osboxes, localhost.localdomain, localhost
myhostname = proyecto"url2.online
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
myorigin = /etc/mailname
readme_directory = no
recipient_delimiter = +
relayhost =
smtp_tls_CPath = /etc/ssl/certs
smtp_tls_security_level = may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level = may
osboxes@osboxes: $
```



```
GNU nano 6.2          /etc/dovecot/dovecot.conf
# Dovecot configuration file

# If you're in a hurry, see http://wiki2.dovecot.org/QuickConfiguration

# "doveconf -n" command gives a clean output of the changed settings. Use it
# instead of copy&pasting files when posting to the Dovecot mailing list.

# '#' character and everything after it is treated as comments. Extra spaces
# and tabs are ignored. If you want to use either of these explicitly, put the
# Files "e inside quotes, eg.: key ="# char and trailing whitespace " "

# Most (but not all) settings can be overridden by different protocols and/or
# source/destination IPs by placing the settings inside sections, for example:
# protocol imap { }, local 127.0.0.1 { }, remote 10.0.0.0/8 { }

# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace [])
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
[ Read 103 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File ^Y Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Instalación de Dovecot:

Dovecot es un servidor IMAP/POP3 que permite a los usuarios acceder a sus correos electrónicos.

Instalación:

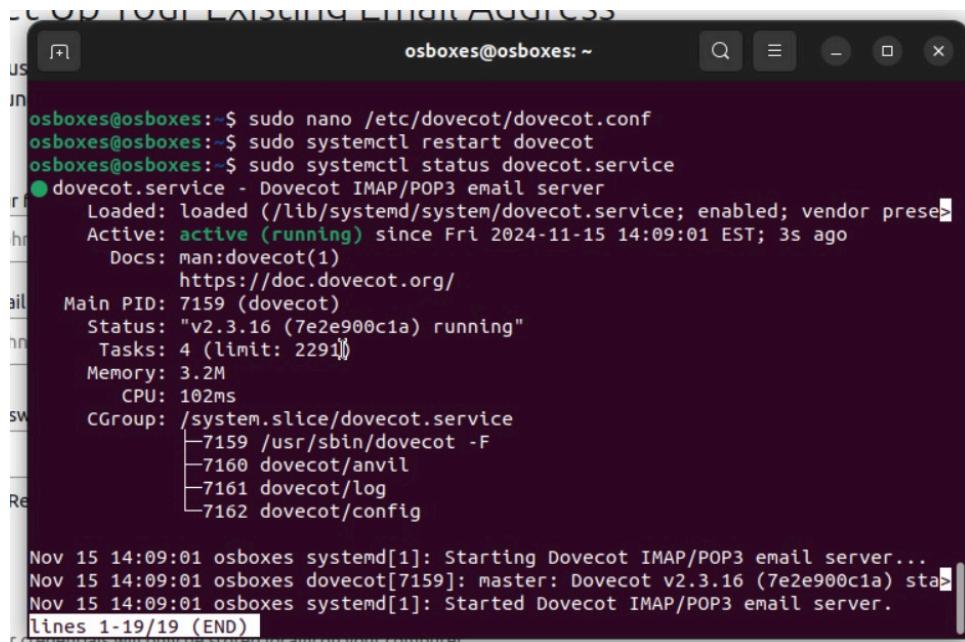
sudo apt-get install dovecot-imapd dovecot-pop3d

- Configuración básica:

Edita el archivo /etc/dovecot/dovecot.conf y asegúrate de que los protocolos IMAP y POP3 estén habilitados:

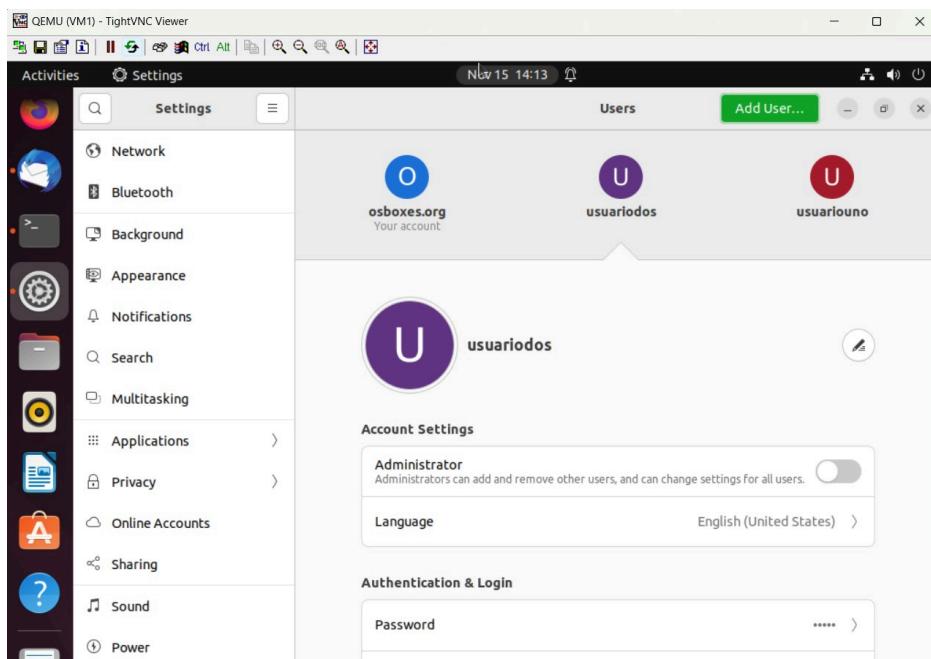
protocols = imap pop3

- Configura la ubicación del correo en /etc/dovecot/conf.d/10-mail.conf:
mail_location = maildir:~/Maildir



```
osboxes@osboxes:~$ sudo nano /etc/dovecot/dovecot.conf
osboxes@osboxes:~$ sudo systemctl restart dovecot
osboxes@osboxes:~$ sudo systemctl status dovecot.service
● dovecot.service - Dovecot IMAP/POP3 email server
   Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor prese>
   Active: active (running) since Fri 2024-11-15 14:09:01 EST; 3s ago
     Docs: man:dovecot(1)
           https://doc.dovecot.org/
 Main PID: 7159 (dovecot)
   Status: "v2.3.16 (7e2e900c1a) running"
    Tasks: 4 (limit: 2291)
   Memory: 3.2M
      CPU: 102ms
     CGroup: /system.slice/dovecot.service
             └─7159 /usr/sbin/dovecot -F
                  ├─7160 dovecot/anvil
                  ├─7161 dovecot/log
                  ├─7162 dovecot/config

Nov 15 14:09:01 osboxes systemd[1]: Starting Dovecot IMAP/POP3 email server...
Nov 15 14:09:01 osboxes dovecot[7159]: master: Dovecot v2.3.16 (7e2e900c1a) sta
Nov 15 14:09:01 osboxes systemd[1]: Started Dovecot IMAP/POP3 email server.
lines 1-19/19 (END)
```



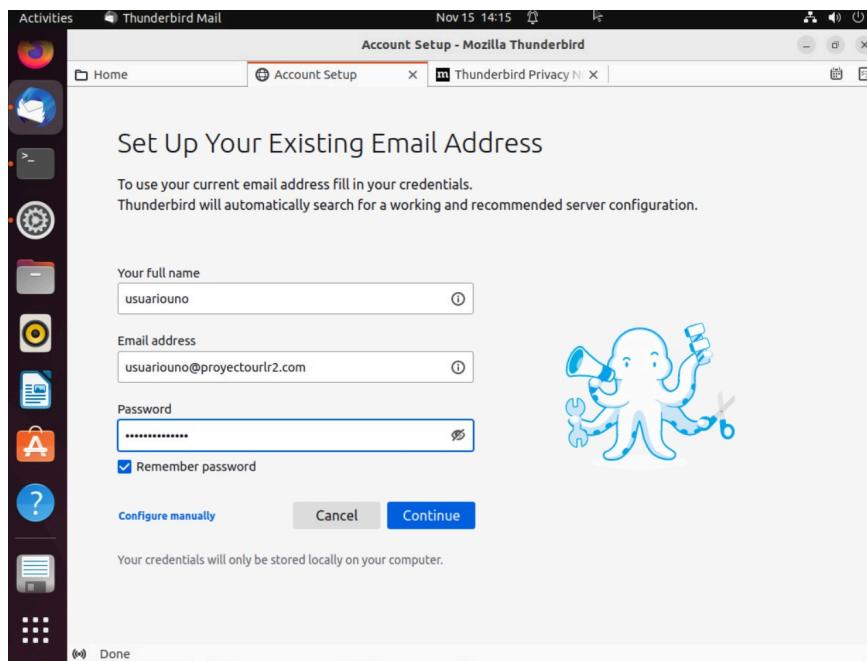
Creación de usuarios de correo:

Cada miembro del grupo debe tener una cuenta de usuario en el sistema.

Creación de usuarios:

sudo adduser nombre_usuario

- Sigue las instrucciones para establecer la contraseña y la información del usuario.

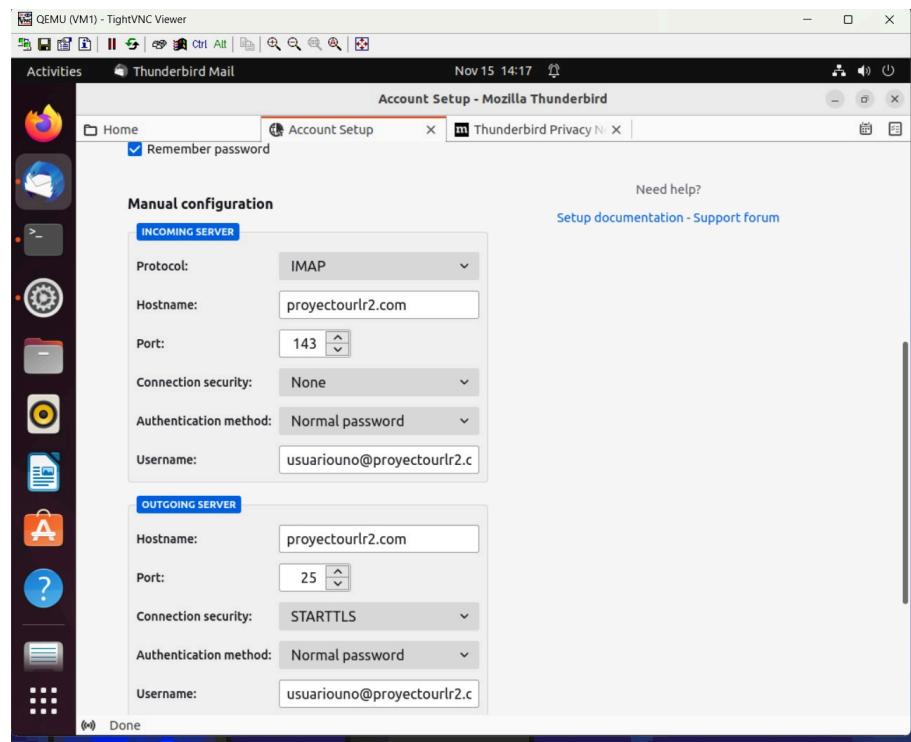


Configuración de clientes de correo:

Los usuarios pueden utilizar clientes de correo como Thunderbird para enviar y recibir correos.

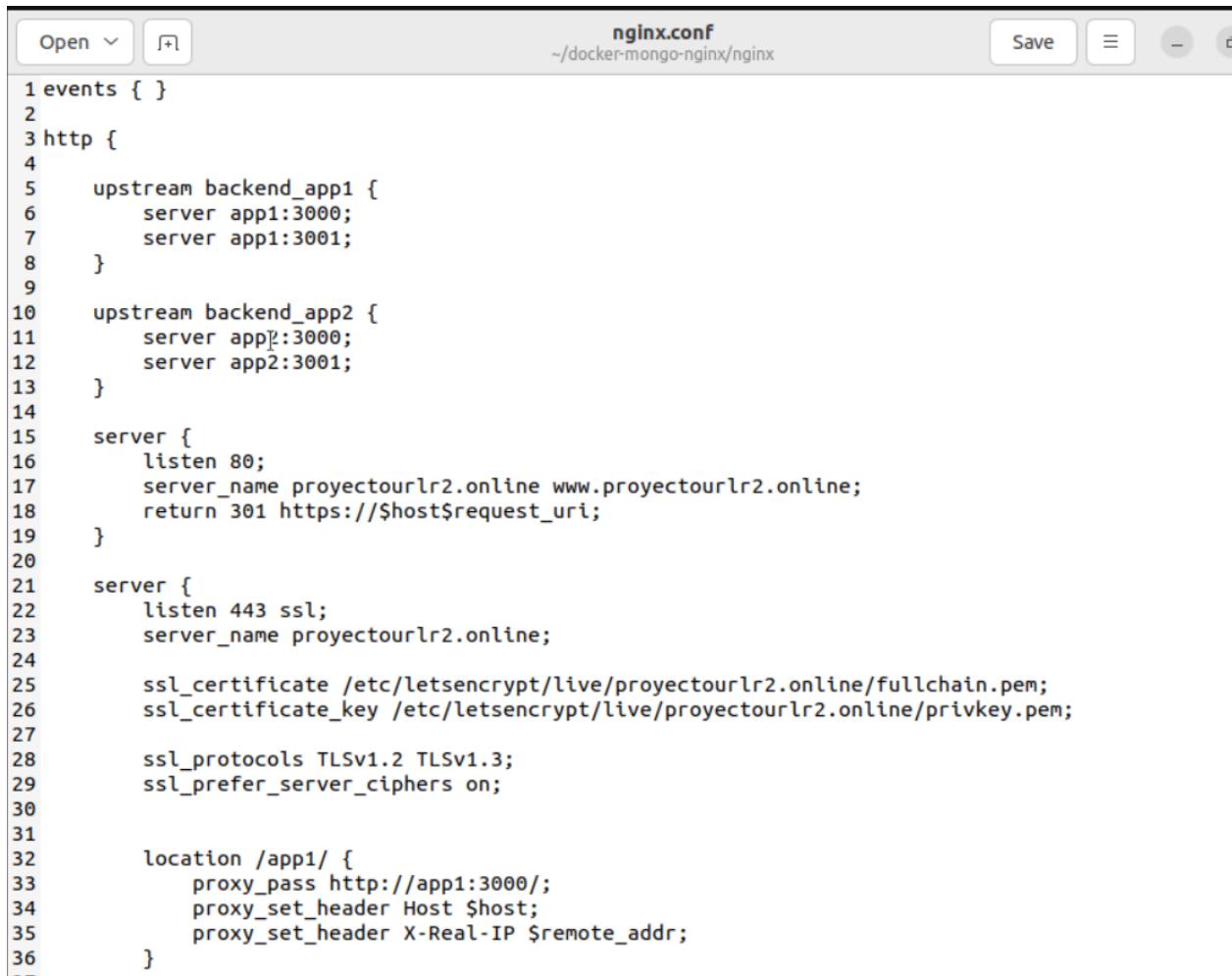
- **Configuración del cliente:**

- **Servidor entrante (IMAP/POP3):**
 - Servidor: proyectourl2.online
 - Puerto IMAP: 143 (sin SSL) o 993 (con SSL)
 - Puerto POP3: 110 (sin SSL) o 995 (con SSL)
 - Método de autenticación: Contraseña normal
- **Servidor saliente (SMTP):**
 - Servidor: proyectourl2.online
 - Puerto: 25 o 587 (usando STARTTLS)
 - Método de autenticación: Contraseña normal



3. CONFIGURACIÓN DEL SERVICIO

3.1 SERVICIOS WEB:



The screenshot shows a code editor window with the title "nginx.conf" and the path "~/docker-mongo-nginx/nginx". The editor interface includes standard buttons for "Open", "Save", and "Edit". The code itself is a configuration file for Nginx, detailing how it acts as a reverse proxy for two backend applications, app1 and app2.

```
1 events { }
2
3 http {
4
5     upstream backend_app1 {
6         server app1:3000;
7         server app1:3001;
8     }
9
10    upstream backend_app2 {
11        server app2:3000;
12        server app2:3001;
13    }
14
15    server {
16        listen 80;
17        server_name proyecto"urlr2.online www.proyecto"urlr2.online;
18        return 301 https://$host$request_uri;
19    }
20
21    server {
22        listen 443 ssl;
23        server_name proyecto"urlr2.online;
24
25        ssl_certificate /etc/letsencrypt/live/proyecto"urlr2.online/fullchain.pem;
26        ssl_certificate_key /etc/letsencrypt/live/proyecto"urlr2.online/privkey.pem;
27
28        ssl_protocols TLSv1.2 TLSv1.3;
29        ssl_prefer_server_ciphers on;
30
31        location /app1/ {
32            proxy_pass http://app1:3000/;
33            proxy_set_header Host $host;
34            proxy_set_header X-Real-IP $remote_addr;
35        }
36    }
37}
```

Servicios web de alta disponibilidad y balanceo de carga:

- Se configuró un servidor web utilizando Nginx como proxy inverso para dos aplicaciones (app1 y app2) en alta disponibilidad.
- El archivo nginx.conf muestra cómo se definieron los bloques upstream para redirigir el tráfico hacia múltiples instancias de backend, logrando balanceo de carga.
- Los servidores escuchan en puertos 3000 y 3001 para cada aplicación, lo que permite distribuir las peticiones de manera uniforme.

The screenshot shows a code editor window with the following details:

- Title Bar:** The title bar displays "nginx.conf" and the path "~/.docker/mongo-nginx/nginx".
- Toolbar:** The toolbar includes standard icons for "Open", "Save", and "Close".
- Text Area:** The main area contains the nginx configuration code, which defines two servers: one for app1 and one for app2, both using SSL certificates from Let's Encrypt.
- Status Bar:** The status bar at the bottom shows "Plain Text" and "Tab Width: 8". It also indicates the current position as "Ln 87, Col 4" and has a "INS" indicator.

```
51     server {
52         listen 443 ssl;
53         server_name app1.proyectourlr2.online;
54
55         ssl_certificate /etc/letsencrypt/live/proyectourlr2.online/fullchain.pem;
56         ssl_certificate_key /etc/letsencrypt/live/proyectourlr2.online/privkey.pem;
57
58         ssl_protocols TLSv1.2 TLSv1.3;
59         ssl_prefer_server_ciphers on;
60
61         location / {
62             proxy_pass http://backend_app1;
63             proxy_set_header Host $host;
64             proxy_set_header X-Real-IP $remote_addr;
65         }
66     }
67
68 # Configuración HTTPS para app2.proyectourlr2.online
69 server {
70     listen 443 ssl;
71     server_name app2.proyectourlr2.online;
72
73     ssl_certificate /etc/letsencrypt/live/proyectourlr2.online/fullchain.pem;
74     ssl_certificate_key /etc/letsencrypt/live/proyectourlr2.online/privkey.pem;
75
76     ssl_protocols TLSv1.2 TLSv1.3;
77     ssl_prefer_server_ciphers on;
78
79     location / {
80         proxy_pass http://backend_app2;
81         proxy_set_header Host $host;
82         proxy_set_header X-Real-IP $remote_addr;
83     }
84 }
```

A screenshot of a web browser window titled "Integrantes del Grupo 1". The URL in the address bar is "https://172.16.1.100/app1/". The page displays a table with five columns: Nombre, Apellido, Carnet, Carrera, and Facultad. There are four rows of data, each representing a member of the group.

Nombre	Apellido	Carnet	Carrera	Facultad
Rafael	Alvarez	1018419	Ingeniería	Facultad de Ingeniería
Eddie	Giron	1307419	Ingeniería	Facultad de Ingeniería
Julio	Ruiz	1284719	Ingeniería	Facultad de Ingeniería

The screenshot shows a web browser window with the URL <https://172.16.1.100/app2/>. The title bar of the browser says "Integrantes del Grupo 2". The main content is a table titled "Integrantes del Grupo 2" with the following data:

Nombre	Apellido	Carnet	Carrera	Facultad
Rafael	Alvarez	1018419	Astrologia	Facultad de Humanidades
Eddie	Giron	1307419	Psicologia	Facultad de Humanidades
Julio	Ruiz	1284719	Teologia	Facultad de Humanidades

Redirección HTTP a HTTPS y Certificados SSL:

- Se configuró Nginx para redirigir automáticamente el tráfico HTTP a HTTPS mediante un bloque server en el puerto 80.
- Los certificados SSL (fullchain.pem y privkey.pem) se instalaron usando Certbot en la ruta /etc/letsencrypt/live/proyectourl2.online/ para asegurar la comunicación.
- Las conexiones seguras utilizan protocolos TLSv1.2 y TLSv1.3, con preferencias de cifrado del servidor activadas.

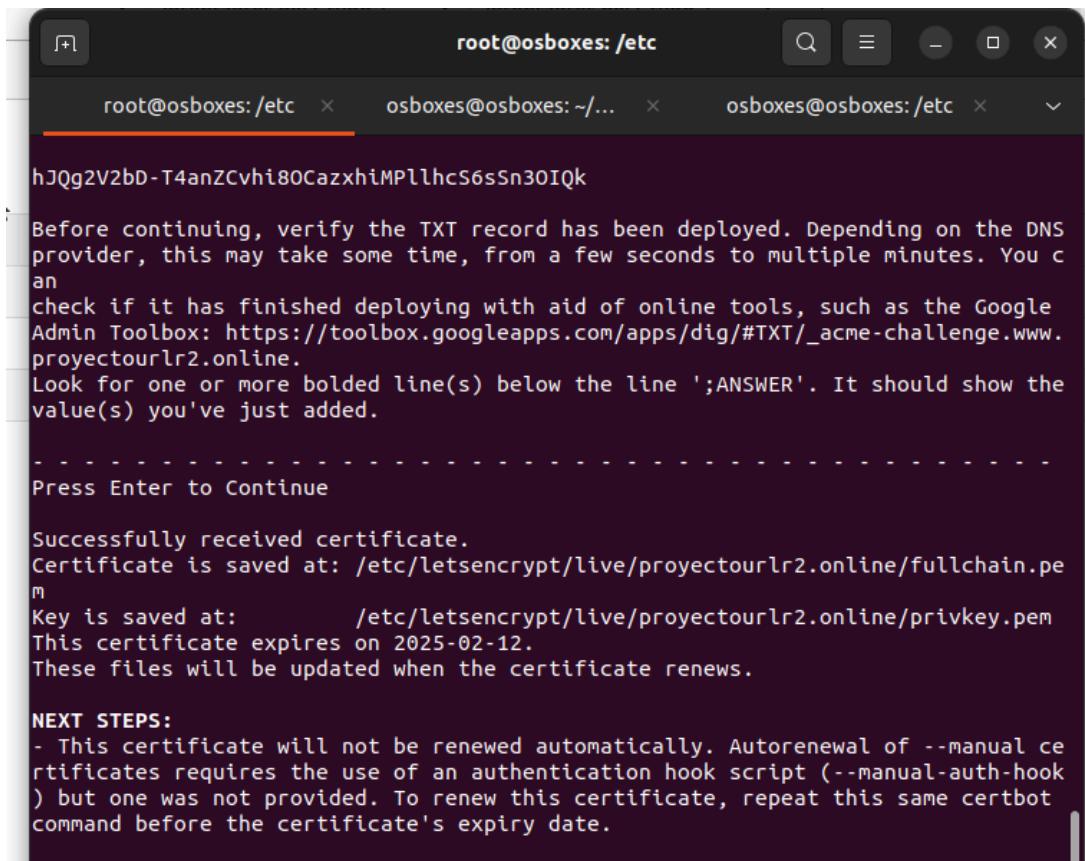
Configuración de Web Services con FQDNs distintos:

- Se definieron diferentes server_name para los servicios (app1.proyectourl2.online y app2.proyectourl2.online), permitiendo que cada URL corresponda a un servicio específico.
- A través del proxy reverso, cada FQDN redirige a su respectivo backend (backend_app1 y backend_app2), logrando aislamiento en la respuesta a las solicitudes.

Publicación de Páginas Web Básicas con Datos del Grupo:

- Las últimas dos imágenes muestran las páginas web de cada grupo, con una tabla de integrantes que incluye nombre, apellido, carnet, carrera y facultad.
- Las páginas son accesibles mediante una dirección IP o FQDN, y conectan con una base de datos para obtener información del grupo, cumpliendo con el requisito de mostrar datos generales a través de un servicio web.

3.2 CERTIFICADOS:



```
hJQg2V2bD-T4anZCvhi80CazxhiMPllhcs6sSn30IQk

Before continuing, verify the TXT record has been deployed. Depending on the DNS provider, this may take some time, from a few seconds to multiple minutes. You can check if it has finished deploying with aid of online tools, such as the Google Admin Toolbox: https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.www.proyectourl2.online.
Look for one or more bolded line(s) below the line ';ANSWER'. It should show the value(s) you've just added.

Press Enter to Continue

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/proyectourl2.online/fullchain.pem
Key is saved at: /etc/letsencrypt/live/proyectourl2.online/privkey.pem
This certificate expires on 2025-02-12.
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To renew this certificate, repeat this same certbot command before the certificate's expiry date.
```

Generación del Certificado SSL:

- Utilizando Certbot, se generó un certificado para el dominio del grupo proyectourl2.online.
- La configuración involucró la creación de un registro TXT en el DNS para la validación, necesaria para obtener el certificado. Esto es parte del proceso de validación de Let's Encrypt.

- El certificado y la clave privada se guardaron en el servidor en las ubicaciones /etc/letsencrypt/live/proyectourl2.online/fullchain.pem y /etc/letsencrypt/live/proyectourl2.online/privkey.pem, respectivamente.

Configuración para el Protocolo HTTPS:

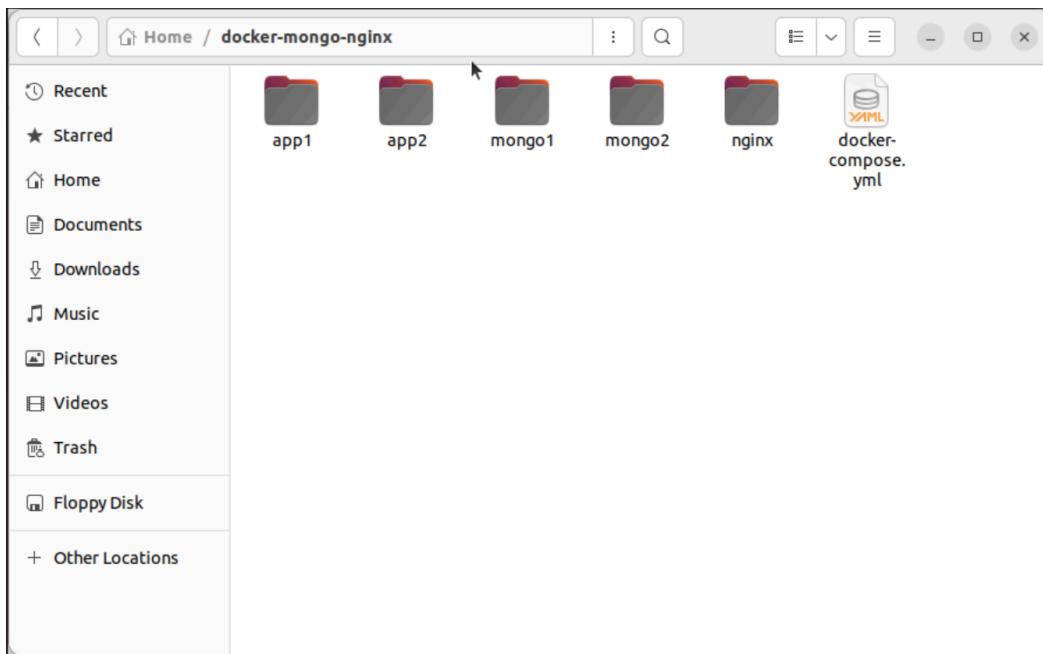
- Una vez generado, el certificado permite conexiones HTTPS seguras hacia el dominio proyectourl2.online.
- En la configuración del servidor web (por ejemplo, en Nginx), se especificaron las rutas de estos archivos para asegurar el tráfico HTTP hacia el servidor mediante HTTPS.

Renovación Manual del Certificado:

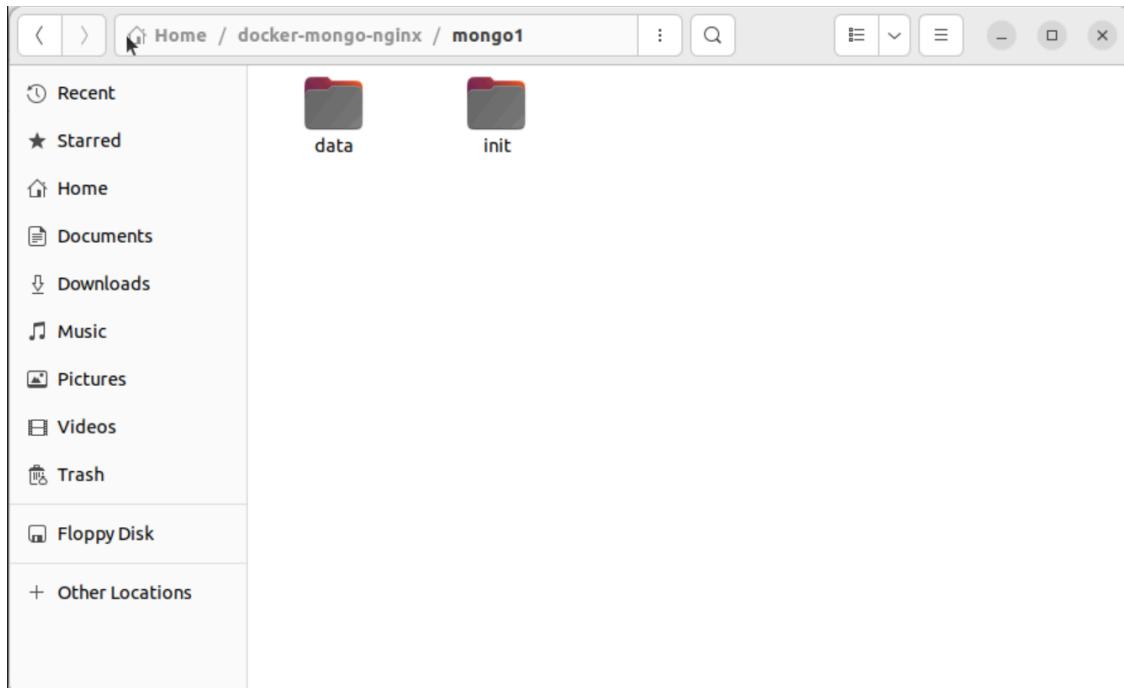
- Dado que el certificado no se renovará automáticamente, es necesario ejecutar el comando de Certbot manualmente antes de su fecha de expiración (12 de febrero de 2025) para actualizarlo.
- Se recomienda programar un recordatorio para renovar el certificado periódicamente o implementar un hook de autenticación manual si es requerido.

3.3 BASE DE DATOS:

Se creó una carpeta para cada una de las instancias



Dentro se configuró su data e init



y se levantan por medio de docker

```
osboxes@osboxes: ~/docker-mongo-nginx
osboxes@osboxes: ~/docker-mongo-nginx
osboxes@osboxes: ~/docker-mongo-nginx$ sudo docker-compose ps
[sudo] password for osboxes:
      Name          Command       State        Ports
----- 
app1    docker-entrypoint.sh npm start   Up           0.0.0.0:3001-
                                                >3000/tcp,:::3001->3000/tcp
app2    docker-entrypoint.sh npm start   Up           0.0.0.0:3002-
                                                >3000/tcp,:::3002->3000/tcp
mongo1  docker-entrypoint.sh mongod     Up           27017/tcp
mongo2  docker-entrypoint.sh mongod     Up           27017/tcp
nginx   /docker-entrypoint.sh ngin ...   Up           0.0.0.0:443->443/tcp,:::443-
                                                >443/tcp, 0.0.0.0:80-
                                                >80/tcp,:::80->80/tcp
osboxes@osboxes: ~/docker-mongo-nginx$
```

y junto con la página web se puede ver el consumo

Integrantes del Grupo 2

Nombre	Apellido	Carnet	Carrera	Facultad
Rafael	Alvarez	1018419	Astrologia	Facultad de Humanidades
Eddie	Giron	1307419	Psicologia	Facultad de Humanidades
Julio	Ruiz	1284719	Teologia	Facultad de Humanidades

3.4 VPN:

Se configuró por medio de pfsense 2 usuarios que son los que van a tener el openvpn

The screenshot shows the pfSense User Manager interface. At the top, there are tabs for 'Users', 'Groups', 'Settings', and 'Authentication Servers'. The 'Users' tab is selected. Below the tabs, there is a table titled 'Users' with columns: 'Username', 'Full name', 'Status', 'Groups', and 'Actions'. Three users are listed: 'admin' (System Administrator, checked), 'openg1' (checked), and 'openg2' (checked). Each user row has edit and delete icons in the 'Actions' column. At the bottom right of the table area, there are 'Add' and 'Delete' buttons.

Username	Full name	Status	Groups	Actions
admin	System Administrator	✓	admins	
openg1		✓		
openg2		✓		

Add **Delete**

Luego se realizaron los certificados

Authorities Certificates Revocation

Search

Search term Both

Enter a search string or *nix regular expression to search certificate names and distinguished names.

Certificate Authorities						
Name	Internal	Issuer	Certificates	Distinguished Name	In Use	Actions
OPENVPN-CA	<input checked="" type="checkbox"/>	self-signed	3	ST=Guatemala, O=MiProyectoRedes, L=Guatemala, CN=internal-ca, C=GT 		    

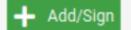
Valid From: Fri, 15 Nov 2024
18:08:52 +0000
Valid Until: Mon, 13 Nov 2034
18:08:52 +0000

 Add

pfSense.home.arpa - System Problem loading page

https://192.168.1.1/system_certmanager.php

Server Certificate	CA: No	Server: Yes	Valid From: Fri, 15 Nov 2024 18:17:15 +0000	Valid Until: Wed, 17 Dec 2025 18:17:15 +0000	Actions
open1 User Certificate	OPENVPN-CA	CA: No Server: No	ST=Guatemala, O=MiProyectoRedes, L=Guatemala, CN=open1, C=GT 	Valid From: Fri, 15 Nov 2024 18:19:27 +0000 Valid Until: Mon, 13 Nov 2034 18:19:27 +0000	    
open2 User Certificate	OPENVPN-CA	CA: No Server: No	ST=Guatemala, O=MiProyectoRedes, L=Guatemala, CN=open2, C=GT 	Valid From: Fri, 15 Nov 2024 18:20:25 +0000 Valid Until: Mon, 13 Nov 2034 18:20:25 +0000	    

 Add/Sign

Y por ultimo el OpenVPN

The screenshot shows a web-based management interface for an OpenVPN server. At the top, there's a navigation bar with tabs: 'VPN / OpenVPN / Servers'. Below the navigation is a horizontal menu with links: 'Servers' (which is underlined in red), 'Clients', 'Client Specific Overrides', 'Wizards', and 'Client Export'. The main content area is titled 'OpenVPN Servers' and contains a table with one row. The table has columns: 'Interface', 'Protocol / Port', 'Tunnel Network', 'Mode / Crypto', 'Description', and 'Actions'. The single row shows: 'VPN' for Interface, 'UDP4 / 1194 (TUN)' for Protocol / Port, '10.10.10.0/24' for Tunnel Network, and detailed crypto settings for Mode, Data Ciphers, Digest, and D-H Params. The 'Description' column says 'VPN para clientes'. The 'Actions' column includes icons for edit, copy, and delete. At the bottom right of the table is a green 'Add' button with a plus sign. A small navigation bar at the very bottom indicates '3.5 FIREWALL'.

OpenVPN Servers					
Interface	Protocol / Port	Tunnel Network	Mode / Crypto	Description	Actions
VPN	UDP4 / 1194 (TUN)	10.10.10.0/24	Mode: Remote Access (SSL/TLS + User Auth) Data Ciphers: AES-256-GCM, AES-128-GCM, CHACHA20-POLY1305, AES-256-CBC Digest: SHA256 D-H Params: 2048 bits	VPN para clientes	

Add

3.5 FIREWALL

3.5 FIREWALL

Se configuró cada una de las interfaces

Mode

Automatic outbound NAT rule generation. (IPsec passthrough included)	Hybrid Outbound NAT rule generation. (Automatic Outbound NAT + rules below)	Manual Outbound NAT rule generation. (AON - Advanced Outbound NAT)	Disable Outbound NAT rule generation. (No Outbound NAT rules)
--	--	---	--

 Save

Mappings

	Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description	Action
<input type="checkbox"/>	WAN	192.168.1.0/24	*	*	*	WAN address	*			
<input type="checkbox"/>	WAN	172.16.1.0/24	*	*	*	WAN address	*			

 Add  Add  Delete  Toggle  Save

WAN

pfSense.home.arpa - Fire X Problem loading page +

https://192.168.1.1/firewall_rules.php

	Status	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
X	0/0 B	*	RFC 1918 networks	*	*	*	*	*		Block private networks	
X	0/0 B	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
<input type="checkbox"/>	X	0/0 B	IPv4 TCP/UDP	*	*	LAN net	*	*		none	
<input type="checkbox"/>	X	0/0 B	IPv4 ICMP any	*	*	LAN net	*	*		none	
<input type="checkbox"/>	✓	0/0 B	IPv4 TCP/UDP	*	*	WAN address	*	*		none	
<input type="checkbox"/>	X	0/0 B	IPv4 TCP/UDP	*	*	INSIDE net	*	*		Denegar todo	
<input type="checkbox"/>	✓	0/0 B	IPv4 TCP	*	*	192.168.4.2	Webservices	*		Permite paginas	

LAN

pfSense.home.arpa - Fire ▾ Problem loading page × +

← → ⌂ ⌂ https://192.168.1.1/firewall_rules.php?if=lan ☆ ⌂ ⌂ ⌂

Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 1/1.09 MIB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
✗ ✓ 0/0 B IPv4 ICMP any	IPv4 ICMP any	DMZ net	*	LAN net	*	*		none		
✗ ✓ 2/24.52 MIB	IPv4 *	LAN net	*	*	*	*		none		
✗ ✓ 0/0 B IPv4 *	IPv4 *	LAN net	*	*	*	*		none	Default allow LAN to any rule	
✗ ✓ 0/0 B IPv6 *	IPv6 *	LAN net	*	*	*	*		none	Default allow LAN IPv6 to any rule	

Add Add Delete Toggle Copy Save Separator

VPN

pfSense.home.arpa - Firewall > Rules (Drag to Change Order)

https://192.168.1.1/firewall_rules.php?if=lan

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	1/1.09 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	✓ 0/0 B	IPv4 ICMP any	DMZ net	*	LAN net	*	*	none			   
<input type="checkbox"/>	✓ 2/24.52 MiB	IPv4 *	LAN net	*	*	*	*	none			   
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	   
<input type="checkbox"/>	✓ 0/0 B	IPv6 *	LAN net	*	*	*	*	none		Default allow LAN IPv6 to any rule	   

 Add  Add  Delete  Toggle  Copy  Save  Separator

OPENVPN

WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager.

Firewall / Rules / OpenVPN

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4	*	*	*	*	*	none	Permitir trafico total	

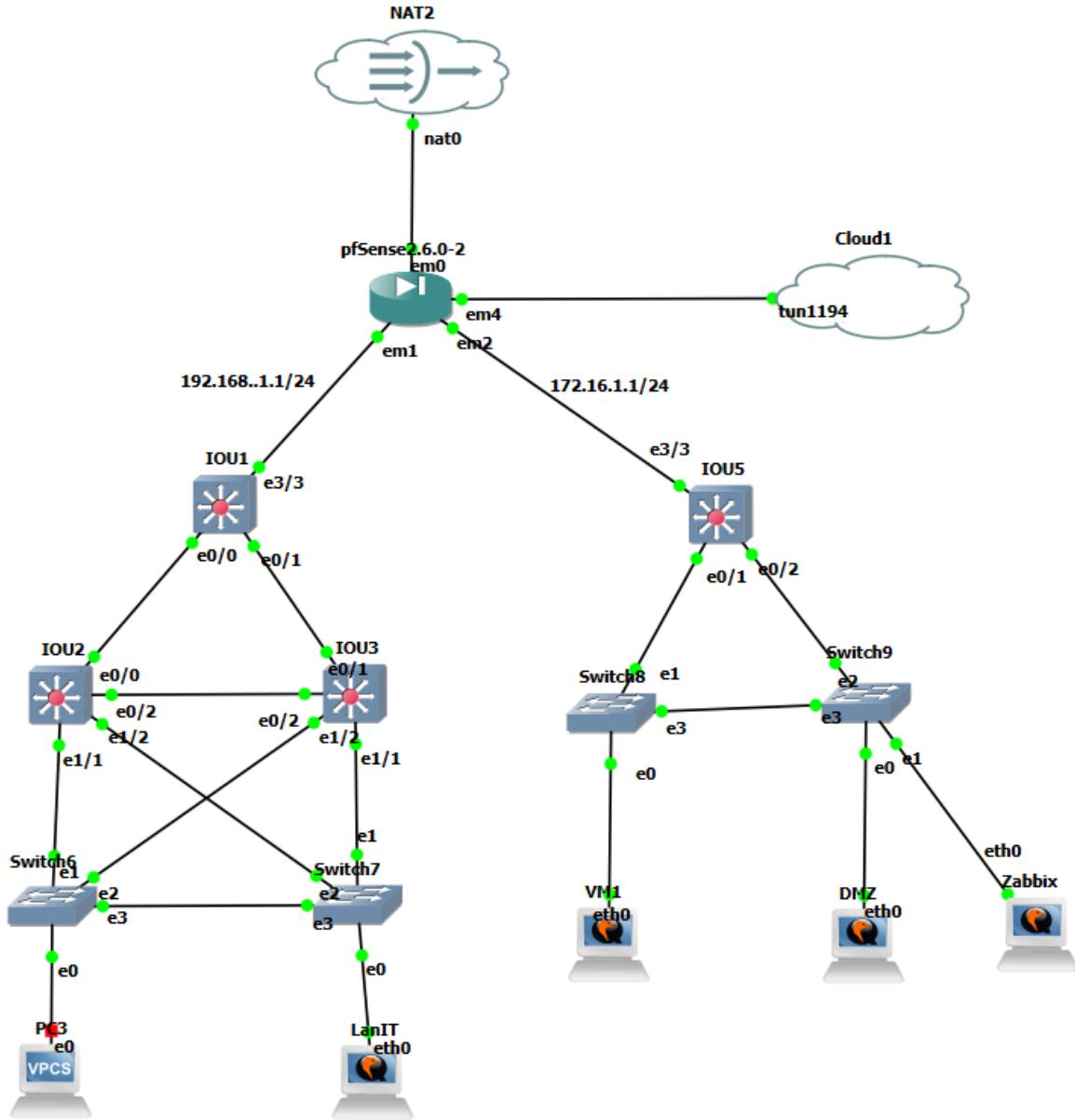
Add **Delete** **Toggle** **Copy** **Save** **Separator**

3.6 ZABBIX

Parámetros para descargar zabbix-client:

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	ZABBIX COMPONENT	DATABASE	WEB SERVER
7.0 LTS	Alma Linux	24.04 (Noble)	Server, Frontend, Agent	---	---
6.4	Amazon Linux	22.04 (Jammy)	Proxy		
6.0 LTS	CentOS	20.04 (Focal)	Agent		
5.0 LTS	Debian	18.04 (Bionic)	Agent 2		
7.2 (pre-release)	Debian (arm64)	16.04 (Xenial)	Java Gateway		
	OpenSUSE Leap		Web Service		
	Oracle Linux				
	Raspberry Pi OS				
	Red Hat Enterprise Linux				
	Rocky Linux				
	SUSE Linux Enterprise Server				
	Ubuntu				
	Ubuntu (arm64)				

4. SIMULACIÓN EN GNS3



5. BIBLIOGRAFÍA

- The Apache Software Foundation. (2023). Documentación oficial de Apache HTTP Server. Recuperado de <https://httpd.apache.org/docs/>
- Nginx, Inc. (2023). Guía de administración de Nginx. Recuperado de <https://docs.nginx.com/nginx/admin-guide/>
- The Apache Software Foundation. (2023). Documentación de Apache Tomcat 9.0. Recuperado de <https://tomcat.apache.org/tomcat-9.0-doc/>
- Ristić, I. (2017). Bulletproof SSL and TLS. Feisty Duck Ltd.
- Stevens, W. R. (1994). TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley Professional.
- Kurose, J. F., & Ross, K. W. (2017). Redes de Computadoras: Un Enfoque Descendente. Pearson Educación.
- RFC Editor. (2014). RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. Recuperado de <https://www.rfc-editor.org/rfc/rfc7230>
- RFC Editor. (2008). RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2. Recuperado de <https://www.rfc-editor.org/rfc/rfc5246>
- GNS3 Technologies Inc. (2023). Documentación oficial de GNS3. Recuperado de <https://docs.gns3.com/>
- OpenSSL Project. (2023). Manual de OpenSSL. Recuperado de <https://www.openssl.org/docs/>
- HAProxy Technologies. (2023). Documentación de HAProxy. Recuperado de <http://www.haproxy.org/#docs>
- Fielding, R., & Reschke, J. (2015). RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. Recuperado de <https://www.rfc-editor.org/rfc/rfc7231>
- OASIS. (2007). Universal Description, Discovery and Integration (UDDI) Version 3.0.2. Recuperado de <https://www.oasis-open.org/standards#uddiv3.0.2>
- World Wide Web Consortium (W3C). (2007). SOAP Version 1.2. Recuperado de <https://www.w3.org/TR/soap12/>
- MySQL AB. (2023). Documentación de MySQL. Recuperado de <https://dev.mysql.com/doc/>
- NIST. (2013). Secure Hash Standard (SHS). Recuperado de <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>
- Stallings, W. (2014). Seguridad en redes: aplicaciones y estándares. Pearson Educación.
- Comer, D. E. (2014). Interconexión de redes con TCP/IP. Prentice Hall.
- IEEE. (2014). IEEE Standard for Ethernet. Recuperado de https://standards.ieee.org/standard/802_3-2018.html
- CISCO Systems. (2023). Guía de configuración de balanceo de carga. Recuperado de <https://www.cisco.com/>