

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE SERVICIOS DE NETWORKING MEDIANTE
DEVOPS**

**IMPLEMENTACIÓN DE UNA SD-WAN OPEN SOURCE MEDIANTE
MÁQUINAS VIRTUALES Y ANSIBLE**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

KEVIN JIMMY ALVARADO GREFA

kevin.alvarado@epn.edu.ec

DIRECTOR: FERNANDO VINICIO BECERA CAMACHO

fernando.becerrac@epn.edu.ec

DMQ, marzo 2024

CERTIFICACIONES

Yo, Kevin Jimmy Alvarado Grefa declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

KEVIN ALVARADO

kevin.alvarado@epn.edu.ec

xkevinsantos00@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Kevin Jimmy Alvarado Grefa, bajo mi supervisión.

FERNANDO VINICIO BECERA CAMACHO

DIRECTOR

fernando.becerrac@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Kevin Jimmy Alvarado Grefa

CI: 150083844-4

DEDICATORIA

Con cariño y profunda gratitud, dedico este proyecto de titulación a mi amada familia, con especial reconocimiento a mi querida madre y abuela, quienes han sido un apoyo incondicional a lo largo de mi travesía profesional. Expreso mi agradecimiento a mi padre por ser mi guía, alentándome siempre a perseguir la meta de obtener un título profesional.

Con la bendición de Dios y el respaldo constante de cada uno de mis seres queridos, este logro se convierte en un testimonio tangible de la posibilidad de convertir un sueño en realidad. Cada uno de ustedes ha sido una fuente de inspiración y fortaleza en este camino, y hoy celebro este logro sabiendo que lo hacemos juntos como familia.

Con amor y agradecimiento sincero, Kevin.

AGRADECIMIENTO

Ante todo, quiero expresar mi profundo agradecimiento a Dios por guiarme hasta este punto de mi vida, permitiéndome culminar mi camino profesional. Este logro marca el comienzo de un viaje lleno de éxitos que la vida tiene reservados para mí.

A mi amada madre, Roció Grefa, le agradezco de corazón por su inquebrantable apoyo en cada paso de mi trayectoria. A pesar de las caídas, su respaldo y bendición han sido un faro de luz en mi camino. Ella es, sin duda, uno de los pilares fundamentales que ha contribuido de manera significativa a mis logros emocionales y profesionales.

A mi querida abuela, Mónica Aguinda, mi segunda madre, le dedico un agradecimiento especial. Su constante fe en mi capacidad de alcanzar el éxito ha sido un motor que me impulsa a seguir adelante. Todos mis esfuerzos han estado inspirados por el deseo de honrar a esta persona tan especial en mi vida.

A mi amor, Elisa Cobos, agradezco infinitamente su presencia en mi vida. Su amor y apoyo incondicional han sido el combustible que me ha impulsado a perseguir la meta de obtener un título profesional. Juntos, miramos hacia un futuro mejor, lleno de oportunidades y sueños compartidos.

Finalmente, quiero expresar mi gratitud a todas aquellas personas que han sido parte de mi vida, contribuyendo con su apoyo y aliento. Cada uno de ustedes ha dejado un valioso grano de arena en mi progreso y aprendizaje, y por eso, les estoy eternamente agradecido. Con sus bendiciones y respaldo, hoy celebro este logro, sabiendo que no caminé solo en esta travesía.

Con gratitud y esperanza, Kevin.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	1
DECLARACIÓN DE AUTORÍA	2
DEDICATORIA	3
AGRADECIMIENTO.....	4
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	2
1.3 Alcance	2
1.4 Marco Teórico	3
SDN.....	3
SD-WAN.....	3
<i>Software open source</i>	3
DevOps.....	3
Ansible.....	4
Virtualización	4
Servicios de <i>networking</i>	4
2 METODOLOGÍA.....	4
3 RESULTADOS	5
3.1 Analizar la herramienta de DevOps que soluciona cada componente del proyecto de titulación. En primera instancia se realizará una investigación sobre el manejo de DevOps, Ansible, SD-WAN y equipos de <i>networking</i>	6
DevOps.....	6
Herramientas DevOps.....	7
Ansible.....	8
SDN.....	8
SD-WAN.....	10
Equipos de <i>networking</i>	11

3.2	Diseñar la solución para cada servicio de <i>networking</i> mediante herramientas de DevOps. Después de haber realizado la investigación sobre los temas se realizará el diseño dinámico de la configuración de SD-WAN y los equipos de <i>networking</i> con Ansible.	12
	Creación de máquinas virtuales.....	14
	Instalación de sistemas operativos.....	14
	Configuraciones de máquinas virtuales.....	16
	Instalación Ansible	18
	Configuración de SSH en VyOS.....	19
	Configuración de claves públicas para comunicación SSH entre Ansible y VyOS	20
3.3	Implementar las soluciones mediante herramientas de DevOps para el despliegue de los servicios de <i>networking</i> . Se implementará el algoritmo el cual incluye SD-WAN en un ambiente virtualizado.	22
	<i>Playbooks</i>	23
3.4	Verificar el funcionamiento de cada servicio de <i>networking</i> implementado mediante DevOps. Finalmente se realizará pruebas del algoritmo diseñado mediante una herramienta de DevOps y se verificará su buen funcionamiento.	29
4	CONCLUSIONES.....	35
5	RECOMENDACIONES	36
6	Referencias Bibliográficas	37
7	ANEXOS.....	42
	ANEXO I: Certificado de Originalidad	1
	ANEXO II: Enlaces	2
	ANEXO III: Códigos Fuente	3

RESUMEN

El crecimiento del tráfico de datos y la demanda de mayor capacidad en las redes de área amplia (WAN) plantea un desafío importante. A medida que los dispositivos WAN buscan mejorar su rendimiento, esto puede resultar en un aumento de los costos. Este problema ha llevado al desarrollo de tecnologías como las redes definidas por software (SDN) y la virtualización de máquinas virtuales como soluciones para abordar las limitaciones de los dispositivos actuales.

Para comprender mejor el problema, es necesario explorar las características esenciales de las redes WAN en crecimiento. Como la eficiencia y escalabilidad las cuales son factores cruciales para un rendimiento óptimo. No obstante la mejora de las prestaciones de los dispositivos WAN mediante hardware especializado es una opción, pero esta puede enfrentar desafíos, por ejemplo: costos adicionales y posibles limitaciones.

En cuanto a soluciones innovadoras esta SD-WAN y la virtualización de las redes. Pues bien, su tecnología no solo simplifica la red, sino que también mejora el ancho de banda, el rendimiento y la redundancia de la red, además de permitir una gestión y control centralizado.

El proyecto de titulación tiene como objetivo principal la implementación de una SD-WAN open source mediante máquinas virtuales y Ansible. En principio se incluirá una investigación exploratoria a las SDN, SD-WAN *open source*, DevOps, Ansible, máquinas virtuales para una solución SD-WAN.

Finalizada la investigación sobre el funcionamiento, características, arquitectura, ventajas y desventajas de cada componente. Se realizará una comparación y selección de la SD-WAN óptima para integrar con DevOps Ansible y máquinas virtuales. El diseño de la virtualización se presentará, destacando la automatización mediante Ansible y mostrando los resultados obtenidos en el despliegue de SD-WAN en VyOS.

PALABRAS CLAVE: WAN, SDN, SD-WAN, GESTION, CONTROL, DEVOPS, ANSIBLE, MAQUINAS VIRTUALES, VYOS.

ABSTRACT

The increasing volume of data traffic and the need for greater capacity in Wide Area Networks (WANs) present significant challenges. As WAN devices strive for enhanced performance, they often encounter rising costs. This dilemma has spurred the development of technologies such as Software-Defined Networking (SDN) and the virtualization of network functions to overcome the constraints of current devices.

To fully comprehend this issue, we must examine the key attributes of expanding WANs, such as efficiency and scalability, which are vital for achieving optimal performance. Although enhancing WAN device performance with specialized hardware is one approach, it can be cost-prohibitive and may introduce other limitations.

Innovative solutions like SD-WAN and network virtualization offer promising alternatives. These technologies not only streamline network complexities but also boost bandwidth, performance, and redundancy. They facilitate centralized management and control as well.

The principal aim of this thesis project is to implement an open-source SD-WAN using virtual machines and Ansible. Initially, it will involve exploratory research on SDN, open-source SD-WAN solutions, DevOps, Ansible, and virtualization technologies pertinent to SD-WAN implementation.

Subsequent to evaluating the performance, characteristics, architecture, benefits, and drawbacks of each component, a comparative analysis will be conducted to select the optimal SD-WAN for integration with DevOps tools like Ansible and virtual machines. The project will culminate in presenting a virtualization design, emphasizing the automation capabilities of Ansible, and demonstrating the results from deploying an SD-WAN on VyOS.

KEYWORDS: WAN, SDN, SD-WAN, MANAGEMENT, CONTROL, DEVOPS, ANSIBLE, VIRTUAL MACHINES, VYOS.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el marco de este proyecto de titulación, se busca llevar a cabo la implementación de una SD-WAN de código abierto mediante el uso de máquinas virtuales y la herramienta Ansible. Con el fin de alcanzar los objetivos establecidos, se requiere realizar una exhaustiva investigación sobre elementos fundamentales para asegurar una correcta implementación de la SD-WAN. Esto incluye la comprensión de conceptos clave como SDN, que constituye la base fundamental para el desarrollo de las SD-WAN, así como el análisis detallado de las características, ventajas y desventajas asociadas con la elección de una versión de código abierto. Además, se abordarán aspectos cruciales como la virtualización de máquinas virtuales y la aplicación de Ansible en entornos de redes, asegurando así una implementación integral y eficiente.

Tras evaluar la información disponible, se ha elegido una solución SD-WAN que se presenta como viable para su implementación en un entorno virtualizado, gestionado a través de la herramienta Ansible. En el ámbito de las SD-WAN de código abierto, se encuentran diversas alternativas, cada una con características distintivas. Es importante destacar que la elección de esta solución se fundamentó principalmente en la compatibilidad con un entorno virtualizado que sea compatible con Ansible, además de ser de código abierto.

Como conclusión de la evaluación, se optó por VyOS, un proyecto de código abierto desarrollado por una comunidad global de colaboradores. VyOS, basado en el sistema operativo Debian, se destaca por su compatibilidad con SD-WAN, convirtiéndose en una elección destacada gracias a sus numerosas ventajas y su integración con Ansible. Esta selección proporciona una combinación óptima para la automatización y administración eficiente de redes.

En el contexto de un entorno virtualizado mediante máquinas virtuales de red, se despliegan nodos VyOS que serán supervisados desde un nodo central basado en Ubuntu, que albergará la herramienta Ansible como controlador principal de los nodos VyOS. Para llevar a cabo la implementación de la SD-WAN y un controlador Ansible, es esencial realizar una configuración previa para establecer la comunicación entre el nodo central y los nodos receptores. Esta configuración reviste importancia crucial, ya que la operación se realiza a través de una conexión SSH. La aplicación de Ansible posibilita la centralización de las configuraciones para la automatización y gestión eficiente de la red. Con las configuraciones realizadas, se habilita la administración remota y la

ejecución de *playbooks*, que son archivos YAML que contienen las tareas a ejecutarse de manera automatizada en la red.

Conforme avanza la ejecución de las tareas descritas en el *playbook*, se llevan a cabo las acciones en los nodos de destino. Una vez verificadas y confirmadas las configuraciones correspondientes en los nodos gestionados, se considera completada la ejecución del *playbook*. Tras asegurar el correcto funcionamiento de la *playbook*, se procede a la creación de un conjunto de *playbooks* para diversas tareas automatizadas. Finalmente, se realiza la recopilación de los resultados obtenidos durante la ejecución de los *playbooks*.

1.1 Objetivo general

Implementar una SD-WAN *open source* mediante máquinas virtuales y Ansible, utilizando herramienta de DevOps para el análisis, diseño, implementación y verificación del funcionamiento correctos de los servicios de *networking*.

1.2 Objetivos específicos

1. Analizar la herramienta de DevOps que soluciona cada componente del proyecto de titulación. En primera instancia se realizará una investigación sobre el manejo de DevOps, Ansible, SD-WAN y equipos de *networking*.
2. Diseñar la solución para cada servicio de *networking* mediante herramientas de DevOps. Después de haber realizado la investigación sobre los temas se realizará el diseño dinámico de la configuración de SD-WAN y los equipos de *networking* con Ansible.
3. Implementar las soluciones mediante herramientas de DevOps para el despliegue de los servicios de *networking*. Se implementará el algoritmo el cual incluye SD-WAN en un ambiente virtualizado.
4. Verificar el funcionamiento de cada servicio de *networking* implementado mediante DevOps. Finalmente se realizará pruebas del algoritmo diseñado mediante una herramienta de DevOps y se verificará su buen funcionamiento.

1.3 Alcance

Este proyecto de titulación se centra en la implementación de una SD-WAN de código abierto en un entorno virtualizado, dirigido desde un nodo central mediante Ansible. La

aplicación de SD-WAN se realiza a través del sistema operativo de red VyOS en los nodos gestionados, mientras que el nodo central utiliza Ansible en un entorno basado en Ubuntu. VyOS se posiciona como una opción altamente eficiente para la implementación de redes SD-WAN, gracias a una serie de ventajas que la convierten en una herramienta poderosa y versátil.

1.4 Marco Teórico

SDN

SDN es una tecnología que surgió en la década de 1990 con el propósito de desarrollar un sistema operativo altamente adaptable en un plano de datos completamente programable. Su principal meta es facilitar una estructura de control centralizada mediante el uso de aplicaciones que generalmente hacen uso de una interfaz de programación de aplicaciones (API) abierta. Esta innovadora aproximación brinda flexibilidad y eficiencia al permitir una gestión más dinámica y personalizada de las redes, adaptándose a las necesidades específicas de cada entorno [1] [2].

SD-WAN

Las Redes de Área Amplia Definidas por *Software*, constituyen arquitecturas innovadoras que aplican la tecnología SDN a las WAN, con el objetivo de aprovechar las destacadas ventajas que ofrece SDN. Este enfoque posibilita la gestión y el control centralizados de los diversos componentes que conforman las WAN, permitiendo una administración más eficiente y ágil de la infraestructura de red [3].

Software open source

El *software* de código abierto, también conocido como software libre, es aquel cuyo código fuente es accesible al público, lo que permite su modificación, estudio y uso sin restricciones. En gran medida, este tipo de *software* se desarrolla, mantiene y distribuye de manera gratuita gracias a la colaboración de una comunidad activa de desarrolladores y usuarios. La transparencia y accesibilidad del código fuente fomentan la innovación, la personalización y la colaboración abierta, creando un ecosistema dinámico donde el conocimiento y la mejora constante son valores fundamentales [4].

DevOps

DevOps representa la adopción de metodologías innovadoras de desarrollo que buscan estrechar la colaboración entre equipos de desarrollo y operaciones. Su objetivo principal es mejorar y potenciar la eficiencia en la interacción entre estas dos áreas, fomentando un enfoque colaborativo que resulta en un desarrollo ágil y eficaz de

software. Al integrar prácticas, herramientas y procesos, DevOps facilita una entrega más rápida y continua de *software*, promoviendo la innovación y la adaptabilidad en todo el ciclo de vida del desarrollo [5]

Ansible

Es una herramienta avanzada DevOps de código abierto que integra una plataforma de automatización, facilitando la configuración y gestión eficiente de extensos conjuntos de nodos desde un nodo principal. Esta solución centraliza las configuraciones en un nodo de control, permitiendo la aplicación uniforme de cambios en todos los nodos de manera eficaz [6].

Virtualización

Se trata de una tecnología que posibilita la creación de una representación virtual de un sistema informático, emulando las funciones del *hardware* físico. A través de una capa de abstracción de las capacidades del *hardware*, se fragmenta en máquinas virtuales que actúan como entidades computacionales independientes [7] [8].

Servicios de *networking*

Los servicios de *networking* desempeñan un papel fundamental en el éxito de cualquier organización, ya que permiten una conexión eficiente, una comunicación fluida, una colaboración efectiva e intercambio seguro de datos [9].

2 METODOLOGÍA

En primer lugar, se llevó a cabo una exhaustiva investigación de las SD-WAN de código abierto disponibles, analizando sus características y su capacidad para ejecutarse en entornos de virtualización. Posteriormente, se procedió al estudio detallado de DevOps y del funcionamiento de la herramienta de automatización Ansible. Una vez comprendido el funcionamiento de estas herramientas, se buscó una solución que fuera compatible con SD-WAN de código abierto y aplicable mediante Ansible. Como resultado de esta investigación, se identificó VyOS, un sistema operativo accesible y compatible con entornos de virtualización, como la solución ideal para integrar SD-WAN de código abierto mediante Ansible.

Con el sistema operativo seleccionado de manera definida, se procedió a la búsqueda de un entorno adecuado para implementar la SD-WAN. Después de explorar diversas opciones para virtualización, se optó por una combinación de VirtualBox y GNS3, dos aplicaciones ampliamente reconocidas y accesibles para el público en general.

Se logró con éxito la creación de una topología de red con tres nodos destinados a la implementación de zonas SD-WAN. Para llevar a cabo la configuración mediante Ansible, se desarrolló previamente una configuración SSH entre los nodos gestionados con VyOS y el nodo central de control con Ubuntu, donde reside Ansible. Con el objetivo de garantizar una comunicación segura entre el nodo central y los nodos gestionados, se llevó a cabo una validación mediante el intercambio de claves y públicas privadas.

Una vez establecida la comunicación entre el nodo central y los nodos gestionados, se procedió a la configuración de las *playbooks* destinadas a implementar las configuraciones SD-WAN. Entre los ajustes a realizar en los nodos SD-WAN se incluyen configuraciones fundamentales como la personalización de banners, la activación de interfaces, la creación de VLAN, la configuración del protocolo de enrutamiento OSPF, la habilitación de servicios como SSH, DHCP y NTP.

Para concluir la implementación de las *playbooks* desarrolladas, se llevó a cabo una verificación de las modificaciones realizadas en los nodos gestionados, asegurándonos también de que las configuraciones aplicadas a dichos nodos funcionaran correctamente.

3 RESULTADOS

En esta sección, se presenta el proceso de implementación de una SD-WAN de código abierto en un entorno virtualizado, utilizando Ansible como herramienta central. Se inicia proporcionando información esencial sobre los conceptos fundamentales de DevOps, Ansible, SD-WAN y equipos de *networking*, estableciendo así una base de datos. Luego, se aborda el diseño e implementación de máquinas virtuales con sistemas operativos de red, seguido por el desarrollo de *playbooks* que ejecutan las configuraciones específicas de SD-WAN.

Finalmente, se explora la correcta ejecución de las *playbooks* y se realiza un análisis de los resultados obtenidos en los dispositivos de red virtualizados, ofreciendo una visión del funcionamiento y los beneficios de esta implementación.

3.1 Analizar la herramienta de DevOps que soluciona cada componente del proyecto de titulación. En primera instancia se realizará una investigación sobre el manejo de DevOps, Ansible, SD-WAN y equipos de *networking*.

DevOps

DevOps se define como una metodología de desarrollo de *software* y una cultura organizacional orientada a fomentar la integración colaborativa y productiva entre los equipos de desarrollo y operaciones. Fundamentado en los principios de integración continua, entrega continua y automatización de infraestructuras TI, DevOps tiene como finalidad optimizar el desarrollo de *software* y acelerar la entrega de soluciones confiables y seguras [10].

Esta metodología tiene su origen en el año 2007, cuando Patrick Debois, un consultor *freelance*, experimentó conflictos e inconvenientes entre los equipos de operaciones y desarrollo en una organización donde trabajaba. En el año 2008, el término fue introducido por primera vez durante una conferencia Agile titulada "Infraestructura Ágil", a cargo de Yhens Wasna y Patrick Debois [11].

El movimiento DevOps sigue experimentando un constante crecimiento y evolución, consolidándose como una cultura integral en TI. Su enfoque centrado en fortalecer la colaboración y productividad entre los equipos de desarrollo y operaciones facilita la automatización de infraestructuras, optimiza los flujos de trabajo y proporciona una medición continua del rendimiento de las aplicaciones.

DevOps proporciona diversos beneficios clave a destacar como:

Velocidad de entrega:

DevOps permite una comunicación más fluida y transparente dentro las organizaciones dando como resultado la entrega de aplicaciones y servicios a un ritmo más acelerado [12] [13].

Colaboración entre equipos:

La implementación de procesos de desarrollo ágil y otras metodologías de comunicación permiten una construcción innovadora y colaborativa entre las áreas de desarrollo y operaciones [12] [13].

Satisfacción del cliente:

La adopción de métricas y medición facilita la detección y solución eficaz de problemas, respaldando un software de calidad y una confiabilidad superior [12] [13].

Costos:

La automatización de procesos disminuye los errores en el ciclo de desarrollo y ayuda disminuir significativamente los costos operativos [12] [13].

Herramientas DevOps

En la aplicación de la filosofía DevOps, las herramientas desempeñan un papel fundamental en los procesos de desarrollo de *software*. La elección de estas herramientas se realiza cuidadosamente, teniendo en cuenta las necesidades específicas del equipo y la envergadura del proyecto. De esta manera, se consigue simplificar y automatizar de manera efectiva el ciclo completo de desarrollo de *software*.

A continuación, se puede ver algunos ejemplos más destacados de herramientas DevOps:

Jenkins

Es una herramienta de código abierto de integración y desarrollo continuos, permite automatizar tareas y acelera el desarrollo de *software* de manera continua. Logra mantener al tanto de los cambios y notificar las nuevas versiones, garantizando una rápida retroalimentación [14].

Kubernetes

Comúnmente denominado "k8s" o simplemente "kube", Kubernetes es una plataforma de código abierto diseñada para la automatización del despliegue, escalado y gestión de aplicaciones en entornos basados en contenedores. Integra los procesos de construcción y desarrollo de aplicaciones con la capacidad de escalar horizontalmente, ajustando dinámicamente la cantidad de contenedores en función de la demanda. Además, gestiona eficientemente los recursos adaptándolos según las necesidades específicas de las aplicaciones [15].

Docker

Es una plataforma de código abierto que capacita a los desarrolladores para empaquetar, distribuir y ejecutar aplicaciones de manera ágil y eficiente es la tecnología de contenedores. Estas unidades ligeras, portátiles y autosuficientes simplifican significativamente el proceso de despliegue, escalabilidad y administración de aplicaciones [16].

GitHub

GitHub se presenta como una herramienta esencial para cualquier desarrollador, ya que constituye una plataforma colaborativa basada en la nube para el desarrollo de *software*. Facilita el alojamiento de proyectos de manera pública o privada, ofreciendo un espacio donde almacenar, compartir y colaborar con la comunidad en la creación de código. Su utilización contribuye significativamente a optimizar la gestión de proyectos y el control de versiones [17].

Ansible

Ansible destaca como una de las herramientas de automatización más ampliamente utilizadas en el ámbito de TI. Esta herramienta no solo posibilita la automatización de una diversidad de procesos informáticos, sino que también desempeña funciones clave en la gestión de configuraciones y la organización de sistemas. Lo más notable es su capacidad para unificar los recursos de múltiples sistemas, simplificando así su administración desde una única plataforma en lugar de abordar tareas de manera individual [18]. Las ventajas más destacadas de Ansible son:

Sencillez: Intuitivo, claro y fácil de aprender, inclusive para alguien sin experiencia [19].

Flexibilidad: Permite el desarrollo de una amplia gama de funciones.

Escalabilidad: Se puede automatizar desde un solo punto a miles de elementos de infraestructura [18].

Sin agentes: No es necesario la instalación de *software* en los dispositivos a gestionar [18] [19].

Código abierto: Herramienta gratuita y una gran comunidad de desarrolladores contribuyentes [18] [20].

SDN

Las SDN representan una innovación revolucionaria en el ámbito de las comunicaciones en red. A pesar de que tecnologías como IP, DNS y DHCP han sido fundamentales en la infraestructura de red durante los últimos 30 años, las SDN surgen como una respuesta potente a la necesidad actual de adaptabilidad en las redes. Ofrecen una alternativa flexible y escalable a las arquitecturas tradicionales de red. A continuación, profundizaremos en las características distintivas de las SDN que las convierten en un cambio revolucionario en este campo [21].

Infraestructura programable

Se destaca por su capacidad de ser programable, lo que simplifica su adaptabilidad directa a las características de la red, permitiendo una flexibilidad al ajuste de la infraestructura de red de manera dinámica y eficiente. A diferencia de las infraestructuras tradicionales, que tienden a ser rígidas [22].

Gestión centralizada

La gestión de la red se centraliza a través de un único controlador de *software*, lo que potencia la administración inteligente del entorno. Esta centralización proporciona una visión integral de los ajustes y actividades de la red, permitiendo un control más eficiente y una toma de decisiones informada en todo el sistema [22] [23].

Optimización dinámica

La gestión dinámica de los parámetros de la infraestructura de red se realiza eficientemente a través de sistemas informáticos y la centralización de la red. Mediante el empleo de algoritmos avanzados y capacidades analíticas, es posible lograr una optimización rápida y adaptativa en cuanto al tamaño, topología y capacidad de la red. Esto asegura una operación eficaz y ajustada a las necesidades cambiantes del entorno de red [22] [23].

Automatizadas

La capacidad de programar la red facilita la ejecución automática en respuesta a eventos y cambios, lo que posibilita la toma de decisiones inteligentes, reduce el tiempo de inactividad y mejora la eficiencia operativa. Esta habilidad se presenta como una clara diferencia frente a las redes tradicionales, donde el crecimiento de la red aumenta la complejidad de gestionar la infraestructura. La automatización no solo contribuye a una eficiencia operativa mejorada, sino que también permite una implementación ágil, una adaptabilidad superior y un rendimiento optimizado de la red [22] [23].

Las SDN desvinculan el *software* del *hardware*. El plano de control se traslada al software, transformando la gestión de las redes a una flexible y adaptable [22].

Existen tres tipos principales de aplicaciones SDN:

SDN basada en dispositivos

Esta configuración permite programar directamente a los dispositivos de red, mediante aplicaciones desarrolladas por el programador. Por esta razón es necesario tener conocimientos de programación y puede ser más complejo de gestionar. Ofrece una mayor flexibilidad y adaptabilidad [24].

SDN basada en controlador

En esta implementación se centraliza la gestión y control de la red en un único controlador. Este controlador central envía instrucciones de como enrutar los paquetes de datos. Facilita la administración y ofrece una mayor capacidad de adaptación [24].

SDN basada en políticas

La implementación de políticas es definida por el administrador de la red, esta gestiona y controla la red a través de un controlador SDN, que mantiene un motor de políticas de aplicaciones [24].

SD-WAN

La SD-WAN se presenta como una extensión de las SDN, centrandose su enfoque en la gestión especializada WAN. Mientras que las redes WAN tradicionales exigen una intervención manual para la gestión y configuración de los dispositivos de red, la SD-WAN destaca por simplificar este proceso mediante configuraciones centralizadas y automatizadas [21].

En una red convencional, la lógica de control está estrechamente vinculada al *hardware*, lo que dificulta la gestión y escalabilidad de las redes, y crea vulnerabilidades frente a ataques. A diferencia de esto, las SD-WAN aborda estos desafíos al separar el plano de control del plano de datos, brindando así un control más eficiente. Esta separación no solo simplifica la administración de la red y reduce los costos operativos, sino que también mejora la utilización de los recursos de la red [25].

Desafíos de SD-WAN

La adopción de SD-WAN conlleva beneficios notables, aunque también plantea desafíos. La compatibilidad entre *hardware* y *software* se vuelve crucial, ya que la incorporación de nuevos dispositivos puede generar incrementos en los costos. La interoperabilidad entre diferentes marcas de productos SD-WAN puede complicar la escalabilidad. Por lo tanto, una planificación cuidadosa y una evaluación detallada de los aspectos técnicos y económicos son esenciales al decidir implementar SD-WAN para satisfacer las necesidades específicas de la red [26] [27].

SD-WAN open source

Las SD-WAN *open source* ofrece numerosas ventajas, al ser de código abierto está disponible a visualización, modificación y distribución. Permitiendo a desarrolladores personalizar y adaptar la fuente a las necesidades específicas [28].

En el entorno del código abierto existe varias soluciones SD-WAN que las organizaciones pueden adoptar. Aquí se presentan ejemplos de soluciones de SD-WAN de código abierto:

flexiWAN: Es una solución de SD-WAN de código abierto integral que ofrece una arquitectura modular y flexible [21].

ZeroTier: Es una aplicación de código que permite una comunicación criptográfica de clave pública para implementar redes privadas y seguras entre dispositivos SD-WAN [29].

OpenDaylight: Proyecto de código abierto y plataforma SDN, proporciona varias herramientas y capacidades, entre ellas SD-WAN.

VyOS: Es un sistema operativo de red de código abierto compatible con funciones SD-WAN.

Beneficios de la SD-WAN *open source*

Las soluciones de SD-WAN de código abierto presentan una variedad de beneficios significativos. Brinda una mayor flexibilidad al permitir la personalización y adaptación de la solución según las necesidades específicas de una organización. Asimismo, fomentan la innovación y la colaboración, ya que desarrolladores de todo el mundo pueden contribuir y mejorar la tecnología. Además, al ser de código abierto, estas soluciones tienden a ser más rentables que las propietarias. Por último, las SD-WAN de código abierto son capaces de integrarse con otras tecnologías de código abierto, posibilitando a las organizaciones construir y gestionar sus redes de manera más eficiente y efectiva [22].

Equipos de *networking*

Los equipos de *networking* son dispositivos diseñados para facilitar la comunicación e interacción entre los dispositivos dentro de una red. Estos incluyen enrutadores, conmutadores, *firewalls*, puertas de enlace, entre otros. En el contexto de dispositivos WAN, posibilitan la conectividad y comunicación en redes considerablemente extensas. En el ámbito de las SD-WAN, se puede abordar una variedad de equipos de red, tales como: *routers*, *firewalls*, *switches*, dispositivos de acceso, entre otros [30].

3.2 Diseñar la solución para cada servicio de *networking* mediante herramientas de DevOps. Después de haber realizado la investigación sobre los temas se realizará el diseño dinámico de la configuración de SD-WAN y los equipos de *networking* con Ansible.

Para llevar a cabo el diseño dinámico de la configuración de SD-WAN, es necesario contar con un entorno que permita la implementación de dispositivos de red y, posteriormente, la configuración correspondiente de SD-WAN. Con el objetivo general de implementar una SD-WAN de código abierto a través de máquinas virtuales y Ansible, se creó un entorno de máquinas virtuales de red mediante la combinación de dos aplicaciones: VirtualBox y GNS3. VirtualBox, una herramienta de código abierto, posibilita la creación y ejecución de máquinas virtuales, mientras que GNS3, una aplicación gratuita y de código abierto, está diseñada para simular redes en el ordenador [31].

Antes de implementar la virtualización de máquinas virtuales, se llevó a cabo un análisis de la compatibilidad con las diferentes plataformas como en la **Tabla 3.1**, configuraciones disponibles como se muestra en la **Tabla 3.2** y requisitos para el funcionamiento del sistema como se muestra en la **Tabla 3.3**, de modo de seleccionar de una solución SD-WAN que se ajuste a las especificaciones del objetivo general. En este contexto, se optó por VyOS como sistema operativo de red destinado a desempeñar el papel crucial en la funcionalidad de dispositivos de red WAN. Asimismo, se tomó la decisión de emplear la distribución Linux Ubuntu como sistema operativo para el controlador central Ansible. Esta elección se basa por estabilidad y seguridad que caracterizan a Ubuntu, haciéndola una opción sobresaliente para servidores y ordenadores que albergan componentes críticos de la infraestructura.

Tabla 3.1 Plataformas compatibles con VyOS [32].

Plataformas de HW soportadas:	Supermicro Lanner, Dell, Edge-core.
Plataformas de virtualización:	Hyper-V, XCP-ng, KVM, Oracle (VirtualBox), Nutanix, VMware, Proxmox.
Los proveedores Cloud más utilizados soportan VyOS:	AWS, Azure, GCP, Openstack, Oracle.
Extra-funcionalidades:	VyOS containers, VyOS API.

Tabla 3.2 Configuraciones disponibles en VyOS [32].

Características de configuración	
Routing	IPv4, IPv6 Static and PBR Dynamic: BGP, OSPFv2/v3, RIP/RIPng Multicast: IGMPv2/v3, PIM
Interfaces	SR-IOV Paravirtualized: VirtIO, VMXNET Emulated
Interfaces internas	Ethernet bridge, 802.1Q VLAN, QinQ, Bonding (LACP and static), WLAN Tunnels: L2TP, L2TPv3, VXLAN, PPTP, GRE, IPIP, SIT, IPIP, IPIP6, IP6IP6, PPPoE, IPoE
Gestión y monitoreo	SSH CLI - API SNMP with extensions, Telegraf, NetFlow, sFlow Ansible, SaltStack, Terraform, Cloud-init, Python library Task scheduling, event handling, scripting Configuration archival wit
Seguridad	Filtrado de tráfico: stateless and stateful firewall Encriptación de tráfico: IPSec, OpenVPN, WireGuard, Openconnect
Servicios de red	DHCP server, client, relay DNS recursive server NAT IGMP-Proxy NTP LLDP mDNS repeater PPPoE server Proxy server Management and monitoring TFTP server

Tabla 3.3 Requisitos de funcionamiento de VyOS [33].

Especificaciones	Mínimas	Recomendadas
Procesador	1 VCPU	1 VCPU
Memoria RAM	512 MB	1 GB
Almacenamiento	4 GB	10 GB

La instalación de VirtualBox y GNS3 resulta intuitiva, siguiendo simplemente las configuraciones recomendadas para llevar a cabo la instalación de ambas aplicaciones. Una vez instaladas, iniciamos VirtualBox para dar inicio al proceso de creación de las máquinas virtuales. En el diseño dinámico de SD-WAN, se decidió establecer tres zonas, cada zona gestionada por una máquina virtual VyOS y supervisada por un controlador central Ubuntu, en el cual se ejecuta Ansible para la coordinación eficiente de la infraestructura.

Creación de máquinas virtuales

La generación de máquinas virtuales en VirtualBox es sencilla con procedimientos guiados. Esto implica ingresar información de requerimientos sobre el sistema operativo que se virtualizará, entre ellos está la capacidad de procesamiento, la asignación de memoria RAM, directorio de creación de la máquina virtual, capacidad de almacenamiento, estos requerimientos agregados serán tomados del ordenador en la cual se está ejecutando la creación de máquinas virtuales. Por esta razón los requerimientos se limitan a la capacidad del ordenador principal. En la Figura 3.1 se tiene un resumen detallado de las configuraciones asignadas a la creación de la máquina virtual VyOS.

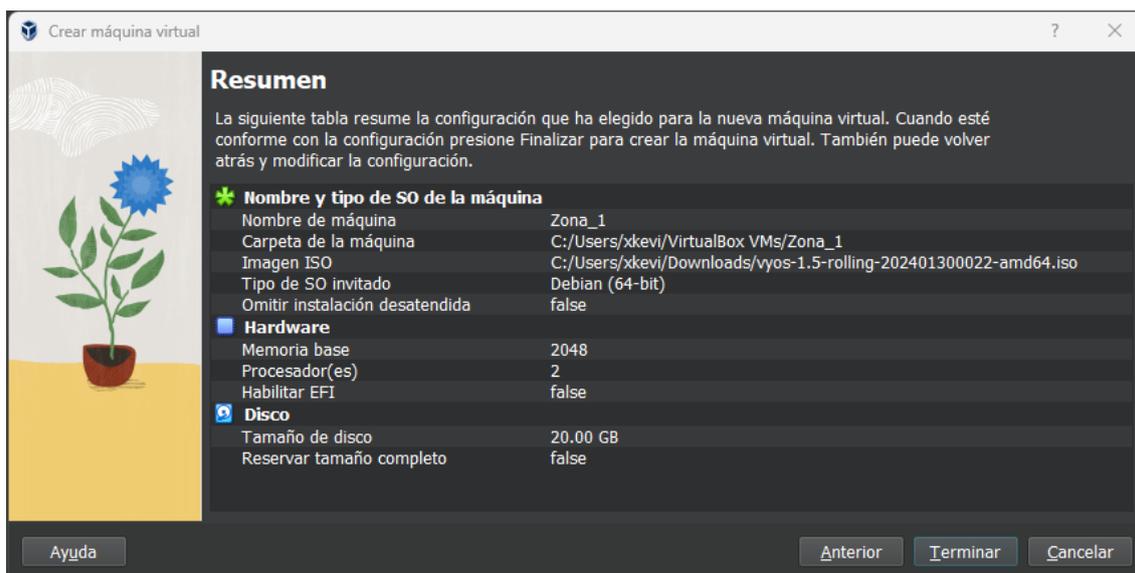


Figura 3.1 Resumen de asignación de recursos a la creación de una máquina virtual

Instalación de sistemas operativos

Creada las máquinas virtuales se procede con la instalación del sistema operativo. Para esta implementación de SD-WAN, se utilizará el sistema operativo Ubuntu con interfaz gráfica, debido a que tiene una mejor interacción con el usuario a través de elementos visuales como; iconos, menús y ventanas. Gracias a las ventajas de un sistema

operativo con interfaz gráfica, la instalación de Ubuntu termina siendo una instalación intuitiva, como se muestra en la Figura 3.2 se tiene una ventana con iconos mostrando las opciones de instalación.



Figura 3.2 Interfaz de instalación Ubuntu Desktop

En cuanto a la instalación del sistema operativo VyOS la instalación es diferente. Para dar inicio la instalación es necesario iniciar sesión e ingresar a la interfaz de línea de comandos. Iniciado el sistema operativo VyOS se tendrá una ventana que solicite iniciar sesión, por defecto el usuario y contraseña es **vyos**. Dentro de la interfaz de línea de comandos de VyOS se debe ejecutar el comando **install image** para iniciar la instalación de VyOS, en la Figura 3.3 se puede observar la ventana de la interfaz de línea de comandos que maneja VyOS. Para la instalación solicitara respuestas por teclado sobre la configuración del sistema a instalarse.

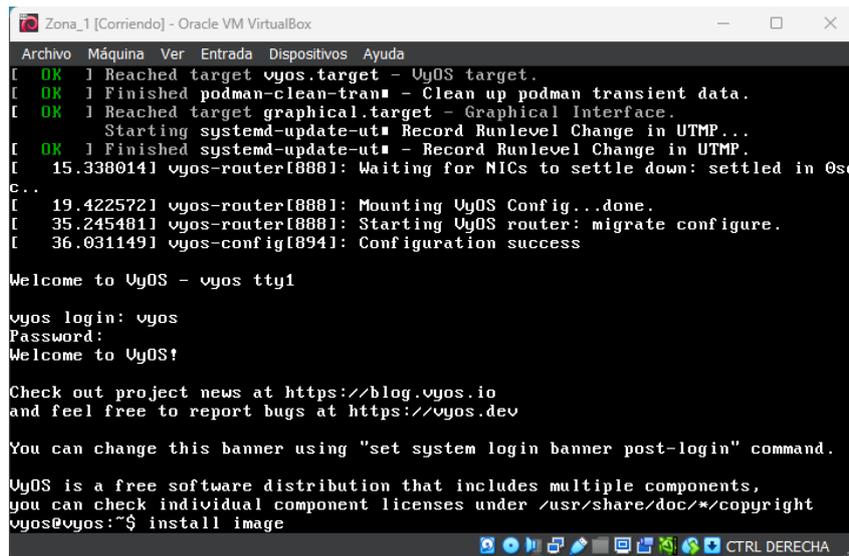


Figura 3.3 Comando de inicio de instalación del sistema operativo VyOS

Configuraciones de máquinas virtuales

Una vez completada la instalación de los sistemas operativos, resulta indispensable llevar a cabo algunas configuraciones específicas dentro de las máquinas virtuales para habilitar la implementación de SD-WAN. Para realizar estos ajustes, es imperativo apagar las virtualizaciones.

El primer cambio para realizar es eliminar la imagen ISO de arranque del sistema, tarea que se realiza en la pestaña de Almacenamiento, específicamente en el apartado de Controlador IDE. Puedes encontrar una guía detallada en la Figura 3.4.

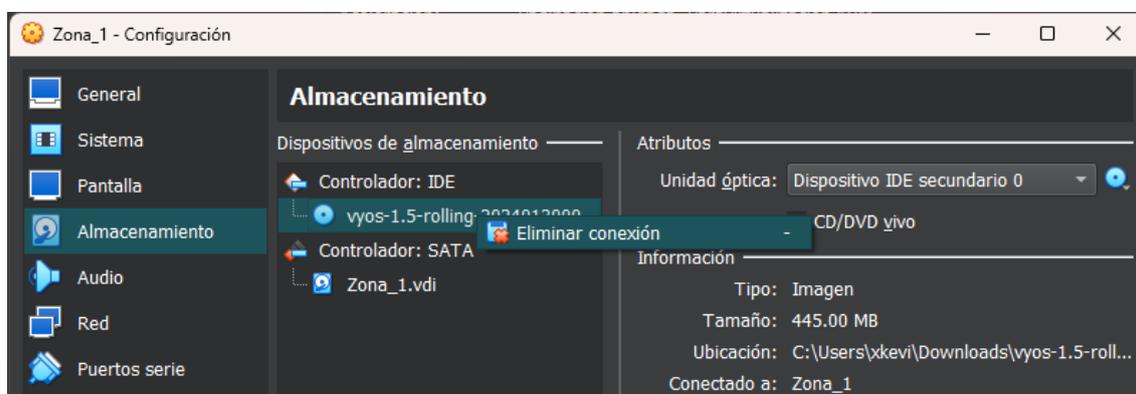


Figura 3.4 Eliminar el SO de arranque VyOS del almacenamiento

La segunda configuración que realizarse es el comportamiento de las interfaces de red de las máquinas virtuales. La configuración en modo puente de las máquinas virtuales servirá para mantener una comunicación a través del dispositivo de red local. La

configuración en no conectado permitirá la conexión de interfaces de red en cuanto se implemente las máquinas virtuales a GNS3.

En la máquina virtual de Ubuntu se tendrá una sola interfaz en modo puente que permita la conexión directa a la red local. En las máquinas virtuales de VyOS se tendrá una interfaz en modo puente y las demás en no conectado. La correspondiente configuración de interfaces de red se puede visualizar en la Figura 3.5 y Figura 3.6.

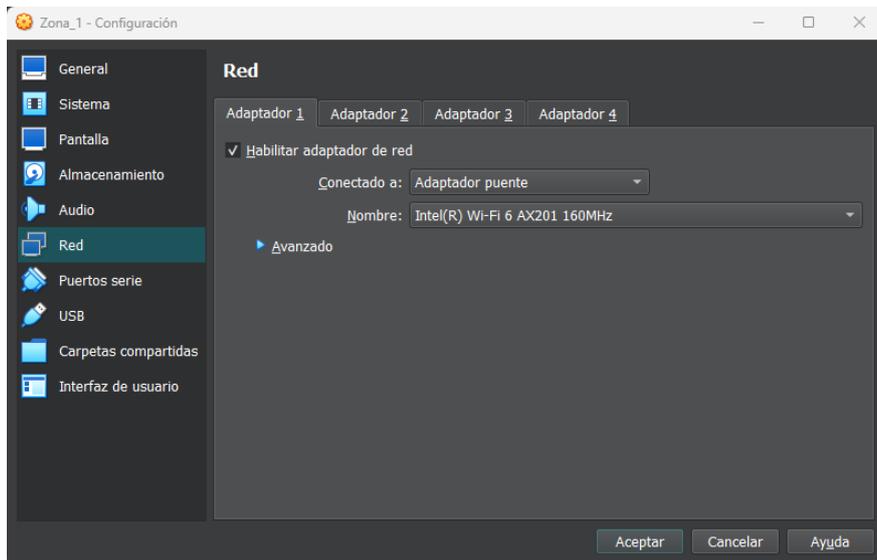


Figura 3.5 Configuración del adaptador de red en Adaptador puente

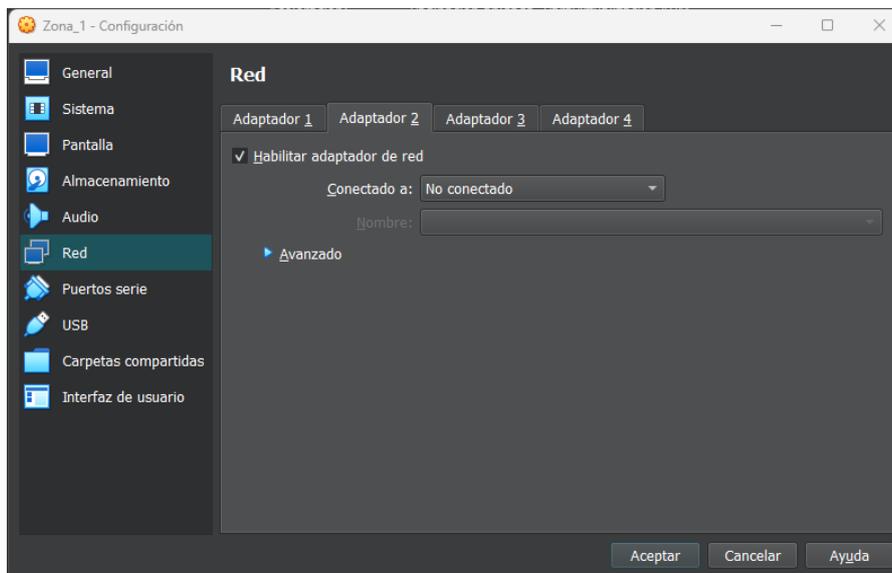


Figura 3.6 Configuración del adaptador de red en No conectado

Instalación Ansible

Para realizar la configuración un controlador central Ansible es necesario implementar el protocolo SSH, este sirve como medio de comunicación de las operaciones entre controlador y dispositivos controlados. Para implementar este protocolo en Ubuntu es necesaria su instalación, esta se ejecuta con el comando **`sudo apt-get install openssh-server -y`**. Realizada la instalación del protocolo SSH se puede establecer conexiones remotas seguras.

Antes de empezar la instalación de Ansible es necesario ejecutar el comando **`sudo apt-add-repository ppa:ansible/ansible`** para agregar el repositorio oficial de Ansible al sistema, en la Figura 3.7 se puede ver la ejecución del comando mencionado, proporcionando una ubicación de paquetes de actualización directa para Ansible. Para iniciar la instalación de Ansible se debe ejecutar el comando **`sudo apt install ansible -y`** esta permite cargar los paquetes de instalación e instalar Ansible como se muestra en la Figura 3.8.

También, es necesario instalar dos paquetes adicionales que mejorara la ejecución de las *playbooks* en VyOS, mediante las siguientes líneas de comandos. El comando **`sudo apt-get install python3-pip`** se utiliza para instalar pip en *Python* 3, este es un gestor de paquetes que permite acceder a módulos adicionales, en la Figura 3.9 se puede ver el comando a ejecutarse en la terminal e instalarse en el sistema.

Por otro lado, el comando **`pip install ansible-pylibssh`** instala la biblioteca **`ansible-pylibssh`**, la cual mejora la ejecución de comandos en servidores remotos gracias a su funcionalidad de cliente **`libssh`**, en la Figura 3.10 se tiene el comando en la terminal.

```
server@server-VirtualBox:~$ sudo apt-add-repository ppa:ansible/ansible
Repositorio: «deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy
main»
```

Figura 3.7 Comando de instalación del repositorio Ansible

```
server@server-VirtualBox:~$ sudo apt install ansible
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
```

Figura 3.8 Comando de instalación de Ansible

```
server@server-VirtualBox:~$ sudo apt install python3-pip
[sudo] contraseña para server:
```

Figura 3.9 Comando de instalación de Python 3

```
server@server-VirtualBox:~$ pip install ansible-pylibssh
```

Figura 3.10 Instalación de asible-pylibssh

Completada la instalación de Ansible se puede verificar su ejecución el comando **ansible --version** como se puede observar en la Figura 3.11. Este comando muestra información detallada sobre la instalación de Ansible en el sistema.

Entre ella está la localizada el directorio del archivo de configuración de Ansible **etc/ansible/ansible.cfg**, dentro de este mismo directorio se puede encontrar un archivo llamado *host*, este archivo es inventario por defecto de Ansible, esta define los *hosts* o nodos que Ansible administrara. Dentro de este archivo por defecto se puede encontrar información comentada de ejemplos de las estructuras para configuraciones de inventario, en la Figura 3.12 se puede observar la configuración de inventario de los dispositivos VyOS.

```
server@server-VirtualBox:~$ ansible --version
ansible [core 2.15.9]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/server/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/server/.ansible/ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
```

Figura 3.11 Información detallada de la instalación Ansible

```
GNU nano 6.2 hosts
[vyos_hosts]
zona1 ansible_host=192.168.0.101

[vyos_hosts:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=zona1
ansible_ssh_pass=vyos
ansible_network_os=vyos
ansible_connection=network_cli
```

Figura 3.12 Estructura de un inventario host

Configuración de SSH en VyOS

Para mantener una comunicación remota entre el controlador Ansible y los nodos VyOS, es necesario la configuración SSH en los receptores VyOS. En las máquinas virtuales VyOS se encuentra 4 interfaces como se muestra en la Figura 3.13.

Para mantener un orden de utilidad de interfaces se optó por aplicar la interfaz eth0 como puerto de comunicación SSH en todos los dispositivos VyOS. Para la configuración SSH en VyOS es necesaria la asignación de una IP y activar el puerto de

comunicación SSH, en la Figura 3.14 y Figura 3.15 se tiene el orden de ejecución de comando en VyOS para realizar los respectivos cambios. En todos los cambios efectos es necesaria la aplicación del comando **commit** y **save** para guardar los cambios realizados.

```

zona1@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface      IP Address      MAC              URF              MTU  S/L  Description
-----
eth0           -               08:00:27:9c:30:79 default         1500 u/u
eth1           -               08:00:27:6f:b1:87 default         1500 u/D
eth2           -               08:00:27:15:d7:6d default         1500 u/D
eth3           -               08:00:27:4d:4f:3b default         1500 u/D
lo             127.0.0.1/8    00:00:00:00:00:00 default         65536 u/u
              ::1/128

```

Figura 3.13 Interfaces en los nodos VyOS

```

zona1@vyos:~$ configure
[edit]
zona1@vyos# set interfaces ethernet eth0 address 192.168.0.101/24
[edit]
zona1@vyos# set interfaces ethernet eth0 description SSH1
[edit]
zona1@vyos# commit
[edit]
zona1@vyos# save
[edit]
zona1@vyos# exit
exit
zona1@vyos:~$

```

Figura 3.14 Asignación de una IP la interfaz de red

```

zona1@vyos# set service ssh port 22
[edit]
zona1@vyos# set system name-server 8.8.8.8
[edit]
zona1@vyos# set protocols static route 0.0.0.0/0 next-hop 192.168.0.1 distance 1
[edit]
zona1@vyos# commit
[edit]
zona1@vyos# save
[edit]
zona1@vyos#

```

Figura 3.15 Asignación de un puerto para la comunicación SSH

Configuración de claves públicas para comunicación SSH entre Ansible y VyOS

Las configuraciones antes realizadas ya permiten una comunicación entre el controlador Ansible y los nodos VyOS, se puede observar la comunicación entre controlador y nodo en la mediante la ejecución del comando de ping-pong **ansible all -m ping -u root** en la Figura 3.16 podemos ver la ejecución del comando y el resultado de una conexión exitosa al *host*.

Con la intención de mejorar la seguridad SSH se aplica el uso de claves públicas. Para realizar esta configuración es necesaria la creación de llaves públicas y privadas, esta se obtiene mediante el comando **ssh-keygen** como se muestra en la Figura 3.17 se puede ver la creación de las llaves, en ella también se puede observar el directorio que las contiene las llaves públicas y privadas. En este caso la creación de las llaves es necesaria en todas las máquinas virtuales (Ubuntu y VyOS).

Generada las llaves en Ubuntu se debe enviar la llave publica a todos los *hosts*, para realizar este proceso se utiliza el siguiente comando **ssh-copy-id user@ip_host**, este proceso se puede observar en la Figura 3.18.

En VyOS según los manuales de configuración de SSH mediante llaves públicas y privadas se debe realizar la ejecución de estos comandos **set system login user zona1 authentication public-keys zona1@vyos key 'clave_pub'** y **set system login user zona1 authentication public-keys zona1@vyos type 'ssh-rsa'**. La información obtenida para completar estas líneas de comando fue necesario ingresar al directorio de la creación de las claves. Para manipular mejor la información textual y completar estos comandos se utilizó la aplicación Putty para conectarse mediante SSH y aprovechar las funciones de copiar y pegar texto en las líneas de comando, tal y como se muestra en la Figura 3.19. Realizados estos cambios se tiene una conexión más segura entre el controlador y *hosts*.

```
server@server-VirtualBox:/etc/ansible$ ansible all -m ping -u root
zona1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Figura 3.16 Prueba de conexión Ansible - VyOS

```

server@server-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/server/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/server/.ssh/id_rsa
Your public key has been saved in /home/server/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:PVQ4oJ7MjYsAmASG6LQ0Y115q4dYtTHDzSsYDjSPPIc server@server-VirtualBox
The key's randomart image is:
+---[RSA 3072]-----+
|
|=o.o  o +.  ..
|*o..=+ 0 oo.
|* oE+o* * o.
|o= .oB B +
|+.. o O S o
|. o + o .
| . o
|
+----[SHA256]-----+
server@server-VirtualBox:~$ █

```

Figura 3.17 Generación de claves SSH

```

server@server-VirtualBox:~$ ssh-copy-id zona1@192.168.0.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/server/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
zona1@192.168.0.101's password:

```

Figura 3.18 Envío de clave publica a un host

```

zona1@vyos:~$ configure
[edit]
zona1@vyos# set system login user zona1 authentication public-keys zona1@vyos key 'AAAAB3NzaC1yc2EAAAADAQABAAQgQ0D2RUdR05u0I3IVexDlVjXTJE+bvIGACJIytoewWPSnxoTX
dh5qtJezPjHwP6u3fzVTjtd3DCGv1LYCu5Z1yq+5sBQ1BwJoP0j1faB8RKJ9UVDnFF1Tj6jjwra9gAgJ5IpyVp25sy21VyJAhw05bVn544dwQhtChS0gTsmHrh3tH1HIdoTd/01F75cH2bsoy24V4ZhQLlGd1dkj
9LHp3Mg1wCy015tmagb9e4I3mYobx3hk6w54kHTwL4u2cs9G5bwtzvSzuYyDwSHAHNB+yx4yNS1oYD402QnEn3GjRHJ/5TgwTndcf4482n1D0i6h+vvayswb5HqAmzCEFIha1IIRIavr97ypD0WR1bo2nx6oiM2L
FrcV0eh3daKvu0E9RQ9hw+IUB14189/0Fcdmp25yGMQuaycLnTKoRjvnn9119bH1sUpkTtyXkps2D465YwTVmFqFxcJRzCWY0VInQ252gnAhtUcQ+xRREc3gwgTSH1RpbZ8XjCvWbFc1JE='
[edit]
zona1@vyos# set system login user zona1 authentication public-keys zona1@vyos type 'ssh-rsa'
[edit]
zona1@vyos# commit
[edit]
zona1@vyos# save
[edit]
zona1@vyos# exit
exit
zona1@vyos:~$ █

```

Figura 3.19 Ejecución de comandos para enviar llaves públicas en VyOS

3.3 Implementar las soluciones mediante herramientas de DevOps para el despliegue de los servicios de *networking*. Se implementará el algoritmo el cual incluye SD-WAN en un ambiente virtualizado.

Para implementar un ambiente virtualizado se configuro previamente las máquinas virtuales. Se tiene un controlador central Ubuntu con Ansible y 3 dispositivos VyOS como nodos administrables, tales como se puede ver en la Figura 3.20 en la parte izquierda de la interfaz de VirtualBox.

Para mejorar la presentación de un ambiente virtualizado de red, se complementó con la representación gráfica de la red en GNS3 como se puede ver en la Figura 3.21. Estos dispositivos tienen la configuración mostrada anteriormente, por ende, están listos para ejecutar el despliegue de configuraciones automáticas y servicios de *networking* por medio de *playbooks*.

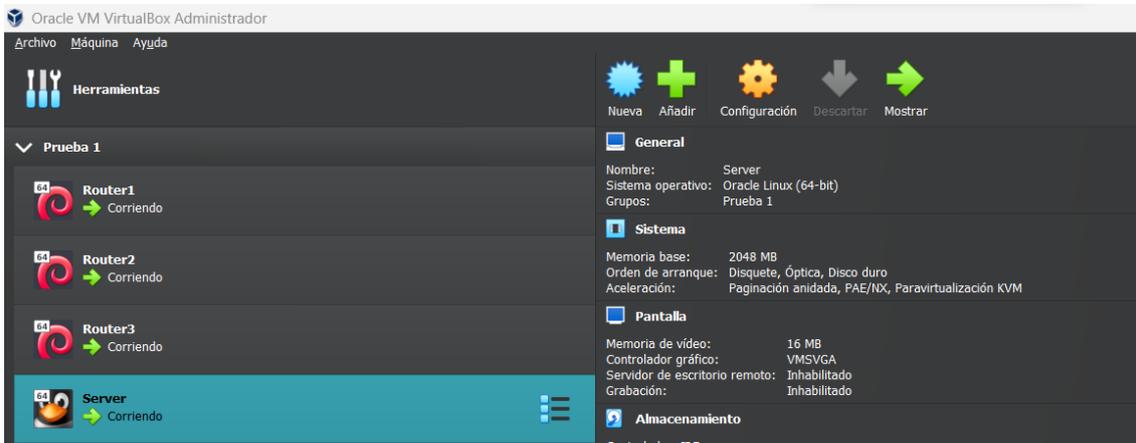


Figura 3.20 Máquinas virtuales en VirtualBox

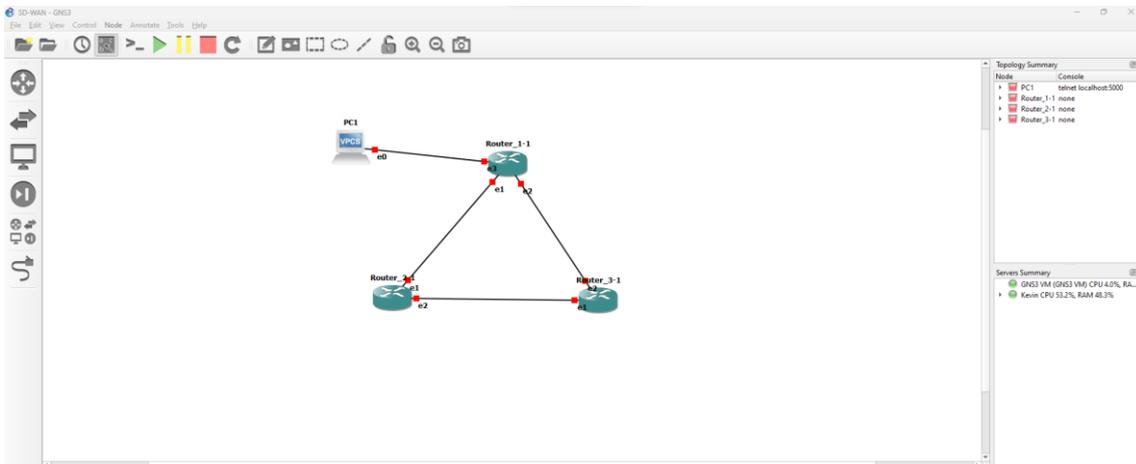


Figura 3.21 Representación gráfica de la topología de red en la plataforma GNS3

Playbooks

Para iniciar la configuración de un dispositivo de red, resulta útil generar anuncios o notificaciones que proporcionen información relevante sobre el dispositivo. Estos mensajes pueden incluir advertencias sobre el uso indebido del dispositivo. En el *playbook* presentada en la Figura 3.22, se puede apreciar la configuración de un texto informativo que se despliega antes y después del inicio de sesión en el dispositivo VyOS. Este enfoque permite comunicar de manera efectiva información crucial al usuario y establecer pautas claras sobre el comportamiento esperado del dispositivo.

```

GNU nano 6.2 conf_banner.yml
# IMPLEMENTACION DE BANNERS
---
- name: Configuracion banner router1
  hosts: router1
  gather_facts: no
  tasks:
    - name: Configuracion de banner antes del inicio sesion
      vyos.vyos.vyos_banner:
        banner: pre-login
        text: |
          \n
          *****
          * Bienvenido a VyOS *
          * Version: 1.5-rolling *
          * Hostname: router1 *
          * IP address: 192.168.0.101 *
          *****
          \n
        state: present

- name: Configuracion de banner router2
  hosts: router2
  gather_facts: no
  tasks:
    - name: Configuracion de banner antes del inicio sesion
      vyos.vyos.vyos_banner:
        banner: pre-login
        text: |
          \n
          *****
          * Bienvenido a VyOS *
          * Version: 1.5-rolling *
          * Hostname: router2 *
          * IP address: 192.168.0.102 *
          *****
          \n

```

Figura 3.22 Configuración de la playbook banners

En el *playbook* siguiente, abordamos la configuración de interfaces de red, asignando direcciones IP, en la Tabla 3.4 se puede llegar a ver las IP asignada a cada puerto y proporcionando breves descripciones de los puertos. En la Figura 3.23, se presenta detalladamente la configuración correspondiente a cada puerto. Este proceso contribuye significativamente a simplificar la configuración de puertos, mejorando la eficiencia y la claridad en la gestión de la red.

```

GNU nano 6.2 conf_interfaces.yml
# IMPLEMENTACION DE INTERFACES DE RED
---
- name: Configuración de interfaces de red router1
  hosts: router1
  become: yes
  gather_facts: no
  tasks:
    - name: Configuracion IP y descripcion
      vyos.vyos.vyos_config:
        lines:
          - set interface ethernet eth1 address 10.0.1.1/24
          - set interface ethernet eth1 description R1-R2
          - set interface ethernet eth2 address 10.0.3.1/24
          - set interface ethernet eth2 description R1-R3
        save: true
- name: Configuración de interfaces de red router2
  hosts: router2
  become: yes
  gather_facts: no
  tasks:
    - name: Configuracion IP y descripcion
      vyos.vyos.vyos_config:
        lines:
          - set interface ethernet eth1 address 10.0.1.2/24
          - set interface ethernet eth1 description R2-R1
          - set interface ethernet eth2 address 10.0.2.1/24
          - set interface ethernet eth2 description R2-R3
        save: true
- name: Configuración de interfaces de red router3
  hosts: router3
  become: yes
  gather_facts: no
  tasks:
    - name: Configuracion IP y descripcion
      vyos.vyos.vyos_config:
        lines:

```

Figura 3.23 Configuración de la playbook de interfaces de red

Tabla 3.4 Tabla de enrutamiento establecido por la playbook

Router1	Eth0: 192.168.0.101/24 Eth1: 10.0.1.1/24 Eth2: 10.0.3.1/24 Eth3: 192.168.1.1/24
Router2	Eth0: 192.168.0.102/24 Eth1: 10.0.1.2/24 Eth2: 10.0.2.1/24 Eth3:
Router3	Eth0: 192.168.0.103/24 Eth1: 10.0.2.2/24 Eth2: 10.0.3.2/24 Eth3:
PC controlador	Eth0: 192.168.0.10/24

En el *playbook* representada en la Figura 3.24, se evidencia la configuración destinada a la creación de VLAN y la sencillez con la que se pueden asignar dichas VLAN a cada

puerto. La implementación de VLAN en la red no solo potencia la seguridad, sino que también optimiza su eficiencia operativa, constituyendo así una estrategia clave para una gestión más robusta y flexible de la infraestructura de red.

```
IMPLEMENTACION DE VLANs
---
- name: Crear VLANs en todos los dispositivos
  hosts: vyos1
  become: yes
  gather_facts: no
  tasks:
    - name: Crear VLAN 100
      vyos.vyos.vyos_vlan:
        vlan_id: 100
        name: VLAN100
        interfaces:
          - eth1
          - eth2
        state: present

    - name: Crear VLAN 200
      vyos.vyos.vyos_vlan:
        vlan_id: 300
        name: VLAN300
        interfaces: eth3
# GUARDAR CONFIGURACION
- name: Guardar configuraciones
  vyos.vyos.vyos_config:
    save: true
```

Figura 3.24 Configuración de la playbook de creación de VLANS

En el *playbook* representada en la Figura 3.25, se muestra la configuración del protocolo de enrutamiento OSPF. VyOS presenta compatibilidad con diversos protocolos de enrutamiento, siendo OSPF uno de ellos. Este protocolo resulta especialmente adecuado para redes WAN, gracias a su capacidad y las opciones de configuración que tiene, lo que contribuye a optimizar la gestión de la red y asegurar una comunicación eficiente entre los dispositivos en entornos de amplia área geográfica.

```

# IMPLEMENTACION DEL PROTOCOLO OSPFv2
---
- name: Implementacion OSPF router1
  hosts: router1
  become: yes
  gather_facts: no
  tasks:
    - name: Configurar OSPF
      vyos.vyos.vyos_ospfv2:
        config:
          areas:
            - area_id: '0'
              area_type:
                stub:
                  default_cost: 15
              network:
                - address: 192.168.0.0/24
                - address: 10.0.1.0/24
                - address: 10.0.3.0/24
        state: merged

- name: Implementacion OSPF router2
  hosts: router2
  become: yes
  gather_facts: no
  tasks:
    - name: Configurar OSPF
      vyos.vyos.vyos_ospfv2:
        config:
          areas:
            - area_id: '0'
              area_type:
                stub:
                  default_cost: 15
              network:
                - address: 192.168.0.0/24

```

Figura 3.25 Configuración de la playbook de protocolo OSPF

En la Figura 3.26, se presenta la configuración del servicio SSH. Inicialmente, fue esencial establecer la configuración SSH para facilitar la comunicación entre el controlador Ansible y los dispositivos *host*. En este *playbook*, se incluye la configuración para modificar el puerto a través del cual se transmite el servicio, optando, en este caso, por mantenerlo en el puerto 22. Este ajuste resulta crucial para garantizar una comunicación segura y eficiente entre los componentes de la infraestructura de manera remota.

```

GNU nano 6.2                                     conf_ssh.yml
# IMPLEMENTACION DE UN PUERTP SSH PARA UNA INTERFACE
---
- name: Configuración SSH router3
  hosts: router3
  become: yes
  gather_facts: no
  tasks:
    - name: Configuracion de puerto SSH
      vyos.vyos.vyos_config:
        lines:
          - set service ssh port 22
...

```

Figura 3.26 Configuración de la playbook del servicio SSH

En el *playbook* representado en la Figura 3.27, se aborda la configuración del servicio DHCP. A través de comandos específicos en el *playbook*, se implementa la activación del servidor DHCP, asignando un rango específico de direcciones, configurando las direcciones de *gateway*, servidor DNS por defecto y especificando la interfaz correspondiente como cliente. Esta configuración posibilita que cualquier dispositivo conectado a esta interfaz obtenga automáticamente una dirección IP asignada por el servidor DHCP, simplificando así la gestión dinámica de direcciones en la red.

```
GNU nano 6.2 conf_dhcp.yml
# IMPLEMENTACION DE DHCP PARA UNA INTERFAZ
---
- name: Configuración DCHP router1
  hosts: router1
  become: yes
  gather_facts: no
  tasks:
    - name: Configuracion DHCP para una interface
      vynos.vyos.vyos_config:
        lines:
          - set interfaces ethernet eth3 address 192.168.1.1/24
          - set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 lease 86400
          - set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 option default-router 192.168.1.1
          - set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 option name-server 8.8.8.8
          - set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 range 1 start 192.168.1.100
          - set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 range 1 stop 192.168.1.100
          - set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 subnet-id 1
...

```

Figura 3.27 Configuración de la *playbook* del servicio DHCP

En la Figura 3.28, se presenta la configuración del servicio NTP. En este proceso, se lleva a cabo la asignación de nuevas direcciones para los servidores NTP, así como la especificación de la interfaz cliente del servicio NTP. Este ajuste posibilita que los dispositivos conectados a este puerto establezcan conexión con los servidores asignados en VyOS, garantizando así una sincronización precisa de los relojes de los dispositivos con el servidor NTP, lo que resulta mantener en sincronía toda la red.

```
GNU nano 6.2 conf_ntp.yml
# IMPLEMENTACION DE DHCP PARA UNA INTERFAZ
---
- name: Configuración NTP router2
  hosts: router1
  become: yes
  gather_facts: no
  tasks:
    - name: Configuracion NTP
      vynos.vyos.vyos_config:
        lines:
          - set interfaces ethernet eth3 address 192.168.1.1/24
        config:
          - set service ntp allow-client address 192.168.1.10
          - set service ntp allow-client address 192.168.0.1
          - set service ntp server 0.pool.ntp.org
          - set service ntp server 1.pool.ntp.org
          - set service ntp server 2.pool.ntp.org
...

```

Figura 3.28 Configuración de la *playbook* del servicio NTP

3.4 Verificar el funcionamiento de cada servicio de *networking* implementado mediante DevOps. Finalmente se realizará pruebas del algoritmo diseñado mediante una herramienta de DevOps y se verificará su buen funcionamiento.

Una vez alcanzado el cuarto objetivo, procederemos a verificar la ejecución de cada una de las playbooks. Para llevar a cabo esta tarea, es necesario ejecutar el comando "*ansible-playbook archivo.yml*". En caso de que se hayan realizado los cambios de manera adecuada, se observará el mensaje "*changed*"; asimismo, al haber completado los cambios de forma correcta, se visualizará el mensaje "*ok*".

En la Figura 3.29, se tiene la correcta ejecución de la *playbook* de banners. Es posible observar el nombre de la configuración, el host en el que se llevaron a cabo los cambios, así como los mensajes "*changed*" y "*ok*".

```
server@server-VirtualBox:~$ ansible-playbook conf_banner.yml

PLAY [Configuración banner router1] *****
TASK [Configuración de banner antes del inicio sesion] *****
changed: [router1]
PLAY [Configuración de banner router2] *****
TASK [Configuración de banner antes del inicio sesion] *****
changed: [router2]
PLAY [Configuración de banners] *****
TASK [Configuración de banner antes del inicio sesion] *****
changed: [router3]
PLAY [Configuración de banners general] *****
TASK [Configuración de banner despues de inicio sesion] *****
changed: [router3]
changed: [router2]
changed: [router1]
TASK [Guardar configuraciones] *****
changed: [router2]
changed: [router1]
changed: [router3]

PLAY RECAP *****
router1      : ok=3    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
router2      : ok=3    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
router3      : ok=3    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figura 3.29 Ejecución de la *playbook* de banners

En la Figura 3.30, se presenta la ejecución de la *playbook* de levantamiento de interfaces. Los cambios efectuados se aplican a cada uno de los dispositivos, por lo tanto, se observa que el mensaje "*changed*" se registra de manera individual para cada dispositivo.

```

server@server-VirtualBox:~$ ansible-playbook conf_interfaces.yml

PLAY [Configuración de interfaces de red router1] *****
TASK [Configuración IP y descripción] *****
[WARNING]: To ensure idempotency and correct diff the input configuration lines should be similar to how they appear if pres
changed: [router1]

PLAY [Configuración de interfaces de red router2] *****
TASK [Configuración IP y descripción] *****
changed: [router2]

PLAY [Configuración de interfaces de red router3] *****
TASK [Configuración IP y descripción] *****
changed: [router3]

PLAY RECAP *****
router1      : ok=1  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
router2      : ok=1  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
router3      : ok=1  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

```

Figura 3.30 Ejecución de la *playbook* de interfaces

En la Figura 3.31, se lleva a cabo la ejecución de la *playbook* de creación de VLAN. Este proceso implica la creación de VLAN en todos los dispositivos, asegurando así que tengan acceso a la nueva VLAN y permitan la configuración necesaria. Se muestra el mensaje "ok", indicando que la creación de la VLAN se ha completado exitosamente.

```

server@server-VirtualBox:~$ ansible-playbook conf_vlans.yml

PLAY [Crear VLANs en todos los dispositivos] *****
TASK [Crear VLAN 100] *****
ok: [router3]
ok: [router2]
ok: [router1]

TASK [Crear VLAN 200] *****
ok: [router3]
ok: [router2]
ok: [router1]

TASK [Guardar configuraciones] *****
changed: [router3]
changed: [router2]
changed: [router1]

PLAY RECAP *****
router1      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
router2      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
router3      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

```

Figura 3.31 Ejecución de la *playbook* de crear VLANs

En la Figura 3.32, se presenta la ejecución de la *playbook* de configuración OSPF. Es evidente que los cambios se aplican de manera individual debido a las distintas configuraciones necesarias para agregar rutas a OSPF en cada dispositivo. El levantamiento del protocolo se lleva a cabo con éxito, y se muestra el mensaje "ok" como confirmación de la ejecución exitosa.

```

server@server-VirtualBox:~$ ansible-playbook conf_ospf.yml
PLAY [Implementacion OSPF router1] *****
TASK [Configurar OSPF] *****
ok: [router1]
PLAY [Implementacion OSPF router2] *****
TASK [Configurar OSPF] *****
ok: [router2]
PLAY [Implementacion OSPF router3] *****
TASK [Configurar OSPF] *****
ok: [router3]
PLAY [Guardar todas la configuraciones] *****
TASK [Guardar configuraciones] *****
changed: [router3]
changed: [router2]
changed: [router1]
PLAY RECAP *****
router1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
router2      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
router3      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Figura 3.32 Ejecución de la playbook de OSPF

En la Figura 3.33, se muestra la ejecución de la *playbook* del servicio SSH. En esta *playbook*, el cambio realizado consistió en la asignación del puerto de comunicación SSH. La configuración se ha completado con éxito, y se observa el mensaje "ok" como indicativo de que la tarea se ha llevado a cabo de manera satisfactoria.

```

server@server-VirtualBox:~$ ansible-playbook conf_ssh.yml
PLAY [Configuración SSH router3] *****
TASK [Configuración de puerto SSH] *****
ok: [router3]
PLAY RECAP *****
router3      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Figura 3.33 Ejecución de la playbook de servicio SSH

En la Figura 3.34, se presenta la configuración del servicio DHCP a través de la *playbook* mencionada en la Figura 3.27. Esta *playbook* se ejecuta exclusivamente en el dispositivo router1, ya que en el diseño de la topología se dispone de una computadora conectada a este puerto para llevar a cabo las pruebas correspondientes a este servicio.

```

server@server-VirtualBox:~$ ansible-playbook conf_dhcp.yml
PLAY [Configuración DHCP router1] *****
TASK [Configuración DHCP para una interface] *****
ok: [router1]
PLAY RECAP *****
router1      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Figura 3.34 Ejecución de la playbook de servicio DHCP

En la Figura 3.35, se observa la implementación de la *playbook* para la configuración del servicio NTP, la cual se aplica al dispositivo router2. Los cambios efectuados se

completan de manera precisa, y se confirma su ejecución exitosa mediante el mensaje "ok".

```
server@server-VirtualBox:~$ ansible-playbook conf_ntp.yml
PLAY [Configuración NTP router2] *****
TASK [Configuración NTP] *****
[WARNING]: To ensure idempotency and correct diff the input configuration lines should be similar to how they appear if present in the running configuration on device
changed: [router1]
PLAY RECAP *****
router1 : ok=1 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Figura 3.35 Ejecución de la playbook de servicio NTP

La correcta ejecución de las configuraciones y servicios de *networking* se presenta de manera visual, mostrando los cambios realizados a los sistemas. Estas *playbooks* ejemplifican la agilidad con la que es posible realizar cambios de manera automatizada y remota. Según los ajustes definidos en el algoritmo, este enfoque representa un potencial para mejorar la red, a amplios aspectos de flexibilidad, escalabilidad y gestión eficiente de dispositivos.

En la Figura 3.36, se puede observar los cambios realizados en la máquina virtual router1, un mensaje de notificación con datos del sistema antes del inicio de sesión en el dispositivo y otro mensaje de advertencia luego del inicio de sesión. De este modo se mantiene una mejor comunicación con el usuario.

```
login as: router1
Pre-authentication banner message from server:
\
*****
* Bienvenido a VyOS *
* Version: 1.5-rolling *
* Hostname: router1 *
* IP address: 192.168.0.101 *
*****
\
End of banner message from server
router1@192.168.0.101's password:
\
*****
*
* El acceso no autorizado a este dispositivo esta prohibido *
* Caulquier actividad en este dispositivo esta siendo monitorizado *
*
*****
\
Last login: Mon Feb 19 11:17:00 2024 from 192.168.0.2
router1@vyos:~$
```

Figura 3.36 Cambios realizados de los banners en el router1

En la Figura 3.37 tenemos los cambios realizados a las interfaces de red del dispositivo router1, los cambios realizados son resultado de la ejecución de las *playbooks* mostrada en Figura 3.30 y Figura 3.31, que corresponden al levantamiento de interfaces y creación de VLAN. La automatización de levantamiento de interfaces mejora en el tiempo de realizar la acción y la aplicación de VLAN mejora la seguridad.

```
router1@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface IP Address MAC VRF MTU S/L Description
-----
eth0 192.168.0.101/24 08:00:27:51:13:53 default 1500 u/u SSH1
eth1 10.0.1.1/24 08:00:27:5d:0c:ef default 1500 u/u R1-R2
eth1.100 - 08:00:27:5d:0c:ef default 1500 u/u VLAN100
eth2 10.0.3.1/24 08:00:27:01:bc:21 default 1500 u/u R1-R3
eth2.100 - 08:00:27:01:bc:21 default 1500 u/u VLAN100
eth3 192.168.1.1/24 08:00:27:0a:8e:65 default 1500 u/u
eth3.300 - 08:00:27:0a:8e:65 default 1500 u/u VLAN300
lo 127.0.0.1/8 00:00:00:00:00:00 default 65536 u/u
::1/128
router1@vyos:~$
```

Figura 3.37 Cambios realizados de las interfaces y creación de VLANs en el router1

En la siguiente Figura 3.38 se tiene el levantamiento del protocolo OSPF, con su respectiva área, costo y las interfaces directamente conectadas. Esta aplicación permitirá la comunicación con otros *routers* mediante el enrutamiento de datos.

```
protocols {
  ospf {
    area 0 {
      area-type {
        stub {
          default-cost 15
        }
      }
      network 192.168.0.0/24
      network 10.0.1.0/24
      network 10.0.3.0/24
    }
  }
}
```

Figura 3.38 Cambios realizados del protocolo OSPF en el router1

En la figura Figura 3.39 tenemos el resultado de una comunicación mediante SSH desde la interfaz de *PowerShell* ejecutando el comando **ssh router3@192.168.0.103**. Se puede verificar su correcta comunicación debido a que se muestra los banners antes configurados a los dispositivos.

```
PS C:\Users\xkevi> ssh router3@192.168.0.103
\
*****
* Bienvenido a VyOS *
* Version: 1.5-rolling *
* Hostname: router3 *
* IP address: 192.168.0.103 *
*****
\
router3@192.168.0.103's password:
\
*****
*
* El acceso no autorizado a este dispositivo esta prohibido *
* Caulquier actividad en este dispositivo esta siendo monitorizado *
*
*****
\
Last login: Mon Feb 19 07:11:06 2024 from 192.168.0.2
router3@vyos:~$
```

Figura 3.39 Correcta ejecución del servicio SSH y conexión desde la PowerShell de la maquina principal

En la Figura 3.40, se tiene una correcta asignación de un IP mediante el servidor DHCP ejecutado por la configuración de la *playbook* de la Figura 3.27. El dispositivo que recibe la asignación de la IP es una máquina de simulación de GNS3 tal cual se muestra en la Figura 3.21 la cual está conectada al router1.

```
PC1
PC1> show ip
NAME       : PC1[1]
IP/MASK    : 192.168.1.100/24
GATEWAY    : 192.168.1.1
DNS        : 8.8.8.8
DHCP SERVER : 192.168.1.1
DHCP LEASE : 73757, 86400/43200/75600
MAC        : 00:50:79:66:68:00
LPORT     : 10032
RHOST:PORT : 127.0.0.1:10033
MTU       : 1500
PC1> █
```

Figura 3.40 Correcta asignación de una IP por el servidor DHCP configurado

En la siguiente Figura 3.41, se tiene los cambios realizados como la agregación de nuevas direcciones de servidores NTP al sistema de VyOS además de la IP de los clientes NTP.

```
set service ntp allow-client address '192.168.1.1'  
set service ntp allow-client address '192.168.0.1'  
set service ntp server 0.pool.ntp.org  
set service ntp server 1.pool.ntp.org  
set service ntp server 2.pool.ntp.org  
set service ntp server time1.vyos.net  
set service ntp server time2.vyos.net  
set service ntp server time3.vyos.net
```

Figura 3.41 Correctos cambios realizados sobre los servidores NTP y el cliente

4 CONCLUSIONES

- Integrar DevOps en la automatización de redes, abarca la configuración de *routers*, *switches* y *firewalls*, conlleva ventajas sustanciales como rapidez, eficiencia, calidad, confiabilidad y reducción de costos. Estas ventajas posibilitan a las empresas adaptarse ágilmente a cambios en el entorno empresarial y nuevas tecnologías. No obstante, enfrenta desafíos, como el cambio cultural, la inversión en herramientas y la necesidad de personal capacitado. En la búsqueda de mejoras en las operaciones de TI, la adopción de DevOps se presenta como una opción valiosa.
- Ansible, una herramienta dentro del panorama DevOps, simplifica la implementación y mejora de SD-WAN al proporcionar automatización, agilidad, escalabilidad, confiabilidad y eficiencia económica. Su utilidad abarca la configuración, aprovisionamiento y gestión de dispositivos SD-WAN, así como la resolución de problemas y la recopilación de datos. A pesar de los desafíos asociados, como la curva de aprendizaje y la integración con otras herramientas, la combinación de Ansible DevOps se presenta como una elección significativa para potenciar la eficiencia y agilidad en las redes SD-WAN.
- La conjunción de VirtualBox y GNS3 proporciona una solución integral para la creación de entornos de virtualización de redes. VirtualBox posibilita la creación de máquinas virtuales con diversos sistemas operativos, mientras que GNS3 simplifica la integración de *routers*, *switches* y otros dispositivos de red. Esta combinación destaca por su flexibilidad, escalabilidad y facilidad de uso,

convirtiéndola en la elección ideal para la formación, investigación y evaluación de nuevas tecnologías de red. Aunque presenta desafíos de aprendizaje y la demanda de recursos computacionales, en general, VirtualBox y GNS3 se rigen como herramientas indispensables para establecer entornos de red virtuales seguros y controlados.

- VyOS es un sistema operativo de red que está en crecimiento, tiene una comunidad que respalda las actualizaciones. Al ser un proyecto de código abierto y al tener una versión gratuita, esta puede o no tener algún servicio o configuración. En comparación con las versiones de por vida que cuentan con asistencia directa de algunos profesionales de la plataforma.
- La compatibilidad de los elementos de la infraestructura es un importante punto de análisis, debido a que la interoperabilidad de una red que abarca diverso hardware puede que alguno no tenga compatibilidad con VyOS.
- Implementar VyOS para desplegar SD-WAN es una opción viable, en especial para empresas u organizaciones que deseen mejorar la flexibilidad y escalabilidad, a costa de enfrentar desafíos técnicos.
- El sistema operativo de VyOS es muy ligero y eficiente, por lo cual no consume muchos recursos de *hardware*, de igual manera esta podría ser ejecutado en dispositivos de bajo costo.

5 RECOMENDACIONES

- Antes de desplegar SD-WAN con VyOS es necesario un análisis exhaustivo sobre las necesidades que se tiene. Entre ellas si VyOS cubre las necesidades de forma correcta. De ser así VyOS puede ser una solución adecuada para la organización.
- Antes de implementar VyOS en un entorno de producción, puede hacer pruebas en un ambiente virtualizado y realizar pruebas de funcionamiento. Esto puede ayudar a familiarizarse con el *software* e identificar cualquier falencia.
- VyOS tiene información del despliegue de su sistema operativo en su plataforma web, pero no está mal obtener capacitaciones sobre el manejo de esta. La ayuda profesional se puede obtener en la misma plataforma web de VyOS en la sección de ayuda.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] A. J. Moreno, «Desarrollo de soluciones SD-WAN basadas en SDN,» Universidad Carlos III de Madrid, Leganes, 2018.
- [2] P. B. Agut, «Diseño de redes WAN y nuevas tecnologías,» FUOC, Barcelona, 2019.
- [3] G. C. Vasquez, «Diseño e implementacion de una red SD-WAN para optimizacion de la red corporativa...,» Universidad Tecnologica del Peru, Lima, 2022.
- [4] Tecno-Soluciones, «¿Qué tipo de licencias de software de código abierto (open source) existen...,» 9 4 2023. [En línea]. Available: <https://tecnosoluciones.com/licencias-de-software-de-codigo-abierto-open-source/?pdf=43589>. [Último acceso: 10 1 2024].
- [5] G. J. Marco, «DevOps, la nueva tendencia en el desarrollo de sistemas TI...,» Universidad Politécnica de Catalunya, 2024.
- [6] S. Manjaly, «Qué es Ansible: la herramienta DevOps para automatizar tareas de IT,» invgate, 24 Marzo 2023. [En línea]. Available: <https://blog.invgate.com/es/ansible>. [Último acceso: 15 Enero 2024].
- [7] IBM, «¿Qué es la virtualización?,» IBM, [En línea]. Available: <https://www.ibm.com/mx-es/topics/virtualization>. [Último acceso: 15 Enero 2024].
- [8] vmware, «¿En qué consiste la virtualización?,» [En línea]. Available: <https://www.vmware.com/es/solutions/virtualization.html>. [Último acceso: 15 Enero 2024].
- [9] TELALCA, «SOLUCIONES DE NETWORKING, REDES DE DATOS, SWITCHES, ROUTERS, DATA CENTERS,» [En línea]. Available: <https://www.telalca.com/networking/>. [Último acceso: 20 Enero 2024].
- [10] S/N, «¿Qué es DevOps?,» [En línea]. Available: <https://articulosit.files.wordpress.com/2012/09/devops.pdf>. [Último acceso: 24 Enero 2024].

- [11] K. Team, «Historia y origen de DevOps,» KeepCoding, 10 Mayo 2023. [En línea]. Available: <https://keepcoding.io/blog/historia-y-origen-de-devops/>. [Último acceso: 20 Enero 2024].
- [12] Academia, «DevOps: la cultura ágil para la entrega de Software,» Pragma, [En línea]. Available: <https://www.pragma.com.co/academia/conceptos/devops>. [Último acceso: 20 Enero 2024].
- [13] M. A. Lopez, «DevOps Aplicado a Sistemas IoT: Definición e Implementacion de Procesos,» Universidad Politécnica de Madrid, Madrid, 2021.
- [14] Sentries, «Introducción a Jenkins: ¿qué es, para qué sirve y cómo funciona?,» 16 Septiembre 2021. [En línea]. Available: <https://sentries.io/blog/que-es-jenkins/#:~:text=Jenkins%20es%20un%20servidor%20open%20source%20para%20la,Java%2C%20es%20multiplataforma%20y%20accesible%20mediante%20interfaz%20web..> [Último acceso: 24 Enero 2024].
- [15] Kubernetes, «Orquestación de contenedores para producción,» [En línea]. Available: <https://kubernetes.io/es/>. [Último acceso: 20 Enero 2024].
- [16] AWS, «¿Qué es Docker?,» [En línea]. Available: <https://aws.amazon.com/es/docker/>. [Último acceso: 20 Enero 2024].
- [17] GitHub, «Acerca de GitHub y Git,» [En línea]. Available: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>. [Último acceso: 20 Enero 2024].
- [18] S. Manjaly, «Qué es Ansible: la herramienta DevOps para automatizar tareas de IT,» 24 Marzo 2023. [En línea]. Available: <https://blog.invgate.com/es/ansible>. [Último acceso: 20 Enero 2024].
- [19] NetApp, «¿QUÉ ES ANSIBLE?,» [En línea]. Available: <https://www.netapp.com/es/hybrid-cloud/it-automation/what-is-ansible-configuration-management/#:~:text=Ansible%20es%20una%20plataforma%20de%20gesti%C3%B3n%20de%20la,se%20vuelven%20repetibles%20y%20menos%20vulnerables%20a%20errores..> [Último acceso: 20 Enero 2024].

- [20] RedHat, «Conceptos básicos de Ansible,» 21 Junio 2022. [En línea]. Available: <https://www.redhat.com/es/topics/automation/learning-ansible-tutorial>. [Último acceso: 20 Enero 2024].
- [21] D. S. Rubio, «Redes WAN definidas por software,» Universidad Oberta de Catalunya, España, 2020.
- [22] Á. J. Moreno, «Desarrollo de soluciones SD-WAN basadas en SDN,» Universidad Carlos III de Madrid, Leganés, 2018.
- [23] NUTANIX, «¿Qué es la red definida por software (SDN)?,» NUTANIX, [En línea]. Available: <https://www.nutanix.com/mx/info/software-defined-networking>. [Último acceso: 20 Enero 2024].
- [24] VMWARE, «Redes definidas por software,» VMWARE, [En línea]. Available: <https://www.vmware.com/es/topics/glossary/content/software-defined-networking.html>. [Último acceso: 20 Enero 2024].
- [25] M. Velasquez, «Ciberseguridad en la implementacion de SD-WAN,» Universidad Mayor de San Andres, La Paz, 2020.
- [26] aryaka, «SD-WAN Costo y retorno de la inversión,» aryaka, [En línea]. Available: <https://www.aryaka.com/es/sd-wan/sd-wan-cost-sd-wan-roi/>. [Último acceso: 21 Enero 2024].
- [27] Lanner, «3 Opciones de Implementación de SD-WAN para Empresas que Deseen Invertir en SDN,» Lanner, [En línea]. Available: <https://www.lanner-america.com/es/blog-es/3-opciones-de-implementacion-de-sd-wan-para-empresas-que-deseen-invertir-en-sdn/>. [Último acceso: 21 Enero 2024].
- [28] RedHat, «Conceptos básicos de Ansible,» RedHat, 21 Junio 2022. [En línea]. Available: <https://www.redhat.com/es/topics/automation/learning-ansible-tutorial#:~:text=Los%20playbooks%20de%20Ansible%20se,estado%20deseado%20de%20un%20sistema..> [Último acceso: 15 Enero 2024].
- [29] zerotier, «Conecte de forma segura cualquier dispositivo, en cualquier lugar.,» zerotier, [En línea]. Available: <https://www.zerotier.com/>. [Último acceso: 21 Enero 2024].

- [30] O. Espinosa, «Qué es el servicio SD-WAN en las operadoras,» RZ, 31 Enero 2023. [En línea]. Available: <https://www.redeszone.net/tutoriales/redes-cable/que-es-sd-wan/>. [Último acceso: 20 Enero 2024].
- [31] VyOS, «Historia,» 2023. [En línea]. Available: <https://docs.vyos.io/es/latest/introducing/history.html#in-the-beginning>. [Último acceso: 15 Enero 2024].
- [32] S. S.L, «Open source router and firewall platform,» 2022. [En línea]. Available: https://esnog.net/gore28/archivos/VyOS-presentacion-ESNOG_GORE28.pdf. [Último acceso: 10 Enero 2024].
- [33] alvmirandan, «Instalación y configuración de un router virtual VyOS,» WitchIT, 2023. [En línea]. Available: [https://witcherit.com/2023/07/04/instalacion-y-configuracion-del-router-virtual-vyos/#:~:text=VyOS%20soporta%20KVM%2C%20Xen%20\(en,hipervisores%20x86%20tambi%C3%A9n%20pueden%20funcionar..](https://witcherit.com/2023/07/04/instalacion-y-configuracion-del-router-virtual-vyos/#:~:text=VyOS%20soporta%20KVM%2C%20Xen%20(en,hipervisores%20x86%20tambi%C3%A9n%20pueden%20funcionar..) [Último acceso: 10 Enero 2024].
- [34] VirtualBox, «Welcome to VirtualBox.org!,» Oracle, [En línea]. Available: <https://www.virtualbox.org/>. [Último acceso: 15 Enero 2024].
- [35] RedHat, «El concepto de SD-WAN,» RedHat, 8 Noviembre 2021. [En línea]. Available: <https://www.redhat.com/es/topics/edge-computing/what-is-sd-wan>. [Último acceso: 21 Enero 2024].
- [36] RedHat, «El concepto de la automatización,» RedHat, 20 Septiembre 2023. [En línea]. Available: <https://www.redhat.com/es/topics/automation#:~:text=La%20automatizaci%C3%B3n%20consiste%20en%20usar,lleven%20a%20cabo%20tareass%20repetitivas..> [Último acceso: 20 Enero 2024].
- [37] L. Fernandez, «Protocolos de redes: guía completa con los protocolos básicos,» RZredes zone, 24 Diciembre 2023. [En línea]. Available: <https://www.redeszone.net/tutoriales/internet/protocolos-basicos-redes/>. [Último acceso: 20 Enero 2024].
- [38] Connex, «Las redes de hoy necesitan flexibilidad, agilidad y seguridad para soportar la organización digital,» Connex, 9 Marzo 2023. [En línea]. Available:

<https://www.connex.com.mx/redes/las-redes-de-hoy-necesitan-flexibilidad-agilidad-y-seguridad-para-soportar-la-organizacion-digital/>. [Último acceso: 20 Enero 2024].

- [39] H. Campos, «¿Qué es la escalabilidad?,» Microsoft, 21 Julio 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/biztalk/core/what-is-scalability>. [Último acceso: 20 Enero 2024].
- [40] A. M. Arranz, «Plataformas de enrutamiento alternativas,» Universidad Zaragoza, Zaragoza, 2016.
- [41] adaptix, «Tráfico de red,» adaptix, 20 Abril 2018. [En línea]. Available: <https://www.adaptixnetworks.com/trafico-de-red/>. [Último acceso: 20 Enero 2024].

7 ANEXOS

ANEXO I: CERTIFICADO DE ORIGINALIDAD

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 20 de febrero de 2024

De mi consideración:

Yo, FERNANDO VINICIO BECERA CAMACHO, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UNA SD-WAN OPEN SOURCE MEDIANTE MÁQUINAS VIRTUALES Y ANSIBLE elaborado por el estudiante KEVIN JIMMY ALVARADO GREFA de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

LINK

https://epnecuador-my.sharepoint.com/:f/g/personal/fernando_becerrac_epn_edu_ec/EqC_4qgXP6Jlr8MmmP23hQ4BJA00idOa7_SmnW2I1jY3Fw?e=GkLyIH

Atentamente,

FERNANDO VINICIO BECERA CAMACHO

Docente

Escuela de Formación de Tecnólogos

ANEXO II: ENLACES



Anexo II.I Código QR de la implementación y pruebas de funcionamiento

ANEXO III: CÓDIGOS FUENTE

PLAYBOOK 1

```
# IMPLEMENTACION DE BANNERS
```

```
---
```

```
- name: Configuracion banner router1
```

```
hosts: router1
```

```
gather_facts: no
```

```
tasks:
```

```
- name: Configuracion de banner antes del inicio sesion
```

```
vyos.vyos.vyos_banner:
```

```
banner: pre-login
```

```
text: |
```

```
\n
```

```
*****
```

```
* Bienvenido a VyOS *
```

```
* Version: 1.5-rolling *
```

```
* Hostname: router1 *
```

```
* IP address: 192.168.0.101 *
```

```
*****
```

```
\n
```

```
state: present
```

```
- name: Configuracion de banner router2
```

```
hosts: router2
```

```
gather_facts: no
```

```
tasks:
```

- name: Configuracion de banner antes del inicio sesion

vyos.vyos.vyos_banner:

banner: pre-login

text: |

\n

```
*****  
  
* Bienvenido a VyOS      *  
  
* Version: 1.5-rolling   *  
  
* Hostname: router2     *  
  
* IP address: 192.168.0.102 *  
  
*****
```

\n

state: present

- name: Configuracion de banners

hosts: router3

gather_facts: no

tasks:

- name: Configuracion de banner antes del inicio sesion

vyos.vyos.vyos_banner:

banner: pre-login

text: |

\n

```
*****  
  
* Bienvenido a VyOS      *  
  
* Version: 1.5-rolling   *  
  
* Hostname: router3     *
```

```

* IP address: 192.168.0.103 *
*****

\n

state: present

# IMPLEMENTACION DE BANNERS TODOS LOS DISPOSITIVOS

- name: Configuracion de banners general

hosts: vyos1

gather_facts: no

tasks:

- name: Configuracion de banner despues de inicio seccion

vyos.vyos.vyos_banner:

  banner: post-login

  text: |

  \n

  *****

  *                               *

  * El acceso no autorizado a este dispositivo esta prohibido *

  * Caulquier actividad en este dispositivo esta siendo monitorizado *

  *                               *

  *****

  \n

state: present

- name: Guardar configuraciones

vyos.vyos.vyos_config:

  save: true

...

```

PLAYBOOK 2

```
# IMPLEMENTACION DE INTERFACES DE RED
```

```
---
```

```
- name: Configuración de interfaces de red router1
```

```
hosts: router1
```

```
become: yes
```

```
gather_facts: no
```

```
tasks:
```

```
- name: Configuracion IP y descripcion
```

```
vyos.vyos.vyos_config:
```

```
lines:
```

```
- set interface ethernet eth1 address 10.0.1.1/24
```

```
- set interface ethernet eth1 description R1-R2
```

```
- set interface ethernet eth2 address 10.0.3.1/24
```

```
- set interface ethernet eth2 description R1-R3
```

```
save: true
```

```
- name: Configuración de interfaces de red router2
```

```
hosts: router2
```

```
become: yes
```

```
gather_facts: no
```

```
tasks:
```

```
- name: Configuracion IP y descripcion
```

```
vyos.vyos.vyos_config:
```

```
lines:
```

```
- set interface ethernet eth1 address 10.0.1.2/24
```

```
- set interface ethernet eth1 description R2-R1
```

```

- set interface ethernet eth2 address 10.0.2.1/24

- set interface ethernet eth2 description R2-R3

save: true

- name: Configuración de interfaces de red router3

hosts: router3

become: yes

gather_facts: no

tasks:

- name: Configuración IP y descripción

vyos.vyos.vyos_config:

lines:

- set interface ethernet eth1 address 10.0.2.2/24

- set interface ethernet eth1 description R3-R2

- set interface ethernet eth2 address 10.0.3.2/24

- set interface ethernet eth2 description R3-R1

save: true

```

...

PLAYBOOK 3

#IMPLEMENTACION DE VLANs

```

- name: Crear VLANs en todos los dispositivos

hosts: vyos1

become: yes

gather_facts: no

tasks:

- name: Crear VLAN 100

```

vyos.vyos.vyos_vlan:

 vlan_id: 100

 name: VLAN100

 interfaces:

 - eth1

 - eth2

 state: present

- name: Crear VLAN 200

vyos.vyos.vyos_vlan:

 vlan_id: 300

 name: VLAN300

 interfaces: eth3

GUARDAR CONFIGURACION

- name: Guardar configuraciones

vyos.vyos.vyos_config:

 save: true

...

PLAYBOOK 4

IMPLEMENTACION DEL PROTOCOLO OSPFv2

- name: Implementacion OSPF router1

 hosts: router1

 become: yes

 gather_facts: no

 tasks:

- name: Configurar OSPF

vyos.vyos.vyos_ospfv2:

config:

areas:

- area_id: '0'

area_type:

stub:

default_cost: 15

network:

- address: 192.168.0.0/24

- address: 10.0.1.0/24

- address: 10.0.3.0/24

state: merged

- name: Implementacion OSPF router2

hosts: router2

become: yes

gather_facts: no

tasks:

- name: Configurar OSPF

vyos.vyos.vyos_ospfv2:

config:

areas:

- area_id: '0'

area_type:

stub:

default_cost: 15

network:

- address: 192.168.0.0/24

- address: 10.0.1.0/24

- address: 10.0.2.0/24

state: merged

- name: Implementacion OSPF router3

hosts: router3

become: yes

gather_facts: no

tasks:

- name: Configurar OSPF

vyos.vyos.vyos_ospfv2:

config:

areas:

- area_id: '0'

area_type:

stub:

default_cost: 15

network:

- address: 192.168.0.0/24

- address: 10.0.2.0/24

- address: 10.0.3.0/24

state: merged

GUARDAR CONFIGURACIONES

- name: Guardar todas la configuraciones

hosts: vyos1

become: yes

gather_facts: no

tasks:

- name: Guardar configuraciones

vyos.vyos.vyos_config:

save: true

...

PLAYBOOK 5

IMPLEMENTACION DE UN PUERTO SSH PARA UNA INTERFAZ

- name: Configuración SSH router3

hosts: router3

become: yes

gather_facts: no

tasks:

- name: Configuración de puerto SSH

vyos.vyos.vyos_config:

lines:

- set service ssh port 22

...

PLAYBOOK 6

IMPLEMENTACION DE DHCP PARA UNA INTERFAZ

- name: Configuración DHCP router1

hosts: router1

become: yes

gather_facts: no

tasks:

- name: Configuración DHCP para una interfaz

vyos.vyos.vyos_config:

lines:

- set interfaces ethernet eth3 address 192.168.1.1/24

- set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 lease 86400

- set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 option default-router 192.168.1.1

- set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 option name-server 8.8.8.8

- set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 range 1 start 192.168.1.100

- set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 range 1 stop 192.168.1.100

- set service dhcp-server shared-network-name LAN subnet 192.168.1.0/24 subnet-id 1

...

PLAYBOOK 7

IMPLEMENTACION DE DHCP PARA UNA INTERFAZ

- name: Configuración NTP router2

hosts: router1

become: yes

gather_facts: no

tasks:

- name: Configuracion NTP

vyos.vyos.vyos_config:

lines:

- set interfaces ethernet eth3 address 192.168.1.1/24

config:

- set service ntp allow-client address 192.168.1.10

- set service ntp allow-client address 192.168.0.1

- set service ntp server 0.pool.ntp.org

- set service ntp server 1.pool.ntp.org

- set service ntp server 2.pool.ntp.org

...