Pontificia Universidad Católica del Ecuador
FACULTAD DE INGENIERÍA
COORDINACIÓN DE POSGRADO

PUCE

PUCE

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**FACULTAD DE INGENIERÍA**

Trabajo de Titulación como requisito previo para la obtención del título de

Magíster en Tecnologías de Información mención Gestión y Administración de TI

# SIMULACION SOBRE GNS3 DE UNA ARQUITECTURA DE RED SPINE AND LEAF PARA UN CENTRO DE DATOS MODERNO

**Autor**: Cristian Andrés Morales Robalino

**Director**: Dr. Gustavo Salazar-Chacón

Quito, marzo 2023

## DECLARACIÓN Y AUTORIZACIÓN

Yo, Cristian Andres Morales Robalino, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún postgrado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Pontifica Universidad Católica del Ecuador podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Cristian Morales

Firmado digitalmente por Cristian Morales
Fecha: 2023.03.25 19:38:22 -05'00'

Cristian Andrés Morales Robalino

## APROBACIÓN DEL TUTOR

En mi carácter de Director del Trabajo de Posgrado Titulado: "***Simulación sobre GNS3 de una arquitectura de red Spine and Leaf para un centro de datos moderno***", presentado por el maestrante Cristian Andrés Morales Robalino, titular de la Cédula de Identidad Nº 1721602884 para optar al Grado de Magíster en Tecnologías de la Información, mención en Gestión y Administración de la Tecnología, considero que dicho Trabajo de Investigación reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte de los Lectores – Evaluadores que se designen para tal fin por parte de las autoridades de la Facultad de Ciencias de la Educación.

En la ciudad de Quito, a los 24 días de marzo del 2023

Firmado electrónicamente por:
GUSTAVO DAVID SALAZAR CHACON

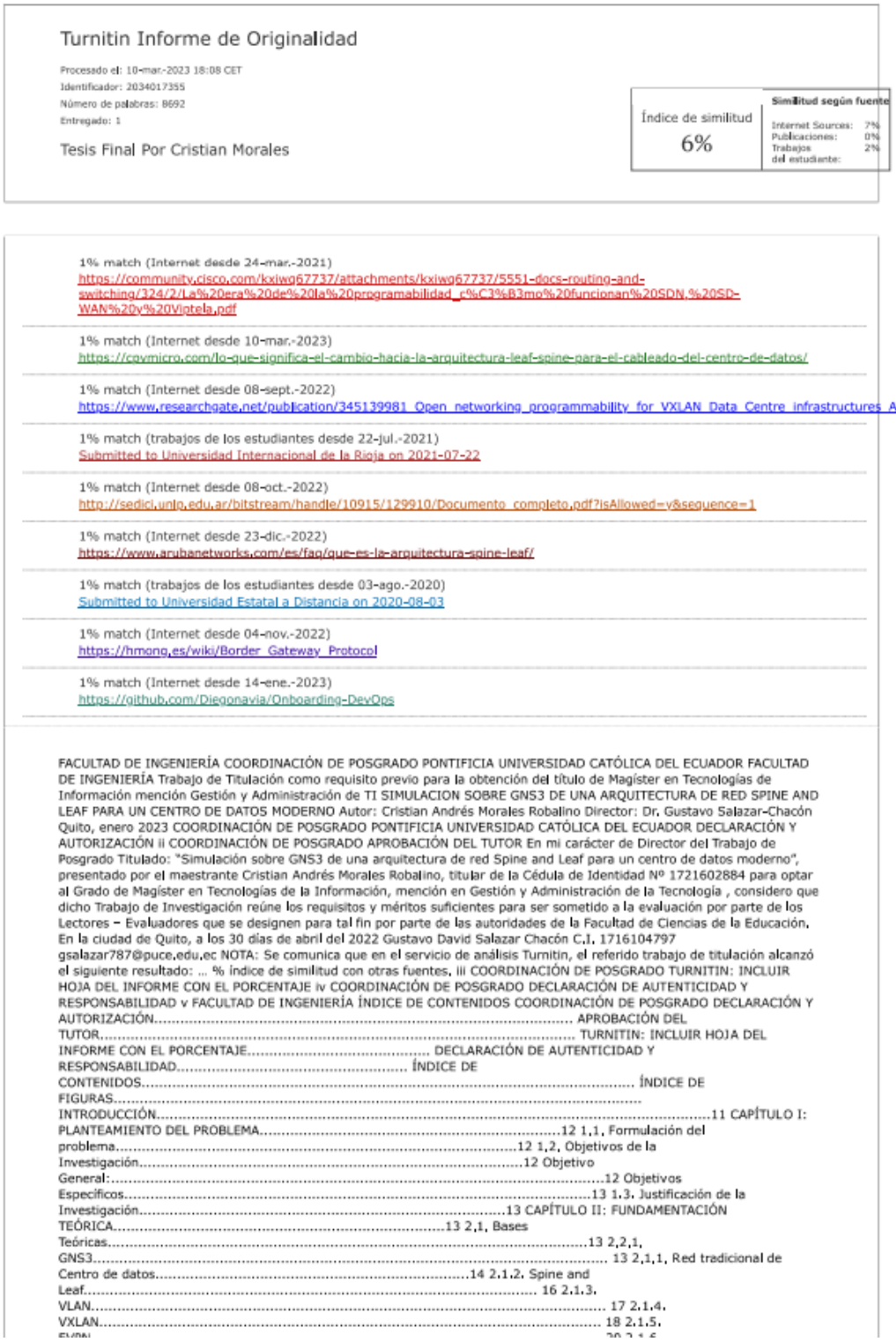
Gustavo David Salazar Chacón

C.I. 1716104797
gsalazar787@puce.edu.ec

NOTA:

Se comunica que en el servicio de análisis Turnitin, el referido trabajo de titulación alcanzó el siguiente resultado: 6 % índice de similitud con otras fuentes.

# INFORME DE TURNITÍN



Turnitin Informe de Originalidad

Procesado el: 10-mar.-2023 18:08 CET
Identificador: 2034017355
Número de palabras: 8692
Entregado: 1

Tesis Final Por Cristian Morales

Índice de similitud 6%

Similitud según fuente
Internet Sources: 7%
Publicaciones: 0%
Trabajos del estudiante: 2%

1% match (Internet desde 24-mar.-2021)
https://community.cisco.com/kxiwg67737/attachments/kxiwg67737/5551-docs-routing-and-switching/324/2/La%20era%20de%20la%20programabilidad_c%C3%B3mo%20funcionan%20SDN,%20SD-WAN%20y%20Vigtela.pdf

1% match (Internet desde 10-mar.-2023)
https://cavmicro.com/lo-que-significa-el-cambio-hacia-la-arquitectura-leaf-spine-para-el-cableado-del-centro-de-datos/

1% match (Internet desde 08-sept.-2022)
https://www.researchgate.net/publication/345139981_Open_networking_programmability_for_VXLAN_Data_Centre_infrastructures_Ansible_an

1% match (trabajos de los estudiantes desde 22-jul.-2021)
Submitted to Universidad Internacional de la Rioja on 2021-07-22

1% match (Internet desde 08-oct.-2022)
http://sedici.unlp.edu.ar/bitstream/handle/10915/129910/Documento_completo.pdf?isAllowed=y&sequence=1

1% match (Internet desde 23-dic.-2022)
https://www.arubanetworks.com/es/faq/que-es-la-arquitectura-spine-leaf/

1% match (trabajos de los estudiantes desde 03-ago.-2020)
Submitted to Universidad Estatal a Distancia on 2020-08-03

1% match (Internet desde 04-nov.-2022)
https://hmong.es/wiki/Border_Gateway_Protocol

1% match (Internet desde 14-ene.-2023)
https://github.com/Diegonavia/Onboarding-DevOps

FACULTAD DE INGENIERÍA COORDINACIÓN DE POSGRADO PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR FACULTAD DE INGENIERÍA Trabajo de Titulación como requisito previo para la obtención del título de Magíster en Tecnologías de Información mención Gestión y Administración de TI SIMULACION SOBRE GNS3 DE UNA ARQUITECTURA DE RED SPINE AND LEAF PARA UN CENTRO DE DATOS MODERNO Autor: Cristian Andrés Morales Robalino Director: Dr. Gustavo Salazar-Chacón Quito, enero 2023 COORDINACIÓN DE POSGRADO PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR DECLARACIÓN Y AUTORIZACIÓN ii COORDINACIÓN DE POSGRADO APROBACIÓN DEL TUTOR En mi carácter de Director del Trabajo de Posgrado Titulado: "Simulación sobre GNS3 de una arquitectura de red Spine and Leaf para un centro de datos moderno", presentado por el maestrante Cristian Andrés Morales Robalino, titular de la Cédula de Identidad Nº 1721602884 para optar al Grado de Magíster en Tecnologías de la Información, mención en Gestión y Administración de la Tecnología , considero que dicho Trabajo de Investigación reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte de los Lectores – Evaluadores que se designen para tal fin por parte de las autoridades de la Facultad de Ciencias de la Educación. En la ciudad de Quito, a los 30 días de abril del 2022 Gustavo David Salazar Chacón C.I. 1716104797 gsalazar787@puce.edu.ec NOTA: Se comunica que en el servicio de análisis Turnitin, el referido trabajo de titulación alcanzó el siguiente resultado: … % índice de similitud con otras fuentes. iii COORDINACIÓN DE POSGRADO TURNITIN: INCLUIR HOJA DEL INFORME CON EL PORCENTAJE iv COORDINACIÓN DE POSGRADO DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD v FACULTAD DE INGENIERÍA ÍNDICE DE CONTENIDOS COORDINACIÓN DE POSGRADO DECLARACIÓN Y AUTORIZACIÓN..... APROBACIÓN DEL TUTOR..... TURNITIN: INCLUIR HOJA DEL INFORME CON EL PORCENTAJE..... DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD..... ÍNDICE DE CONTENIDOS..... ÍNDICE DE FIGURAS..... INTRODUCCIÓN.....11 CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA.....12 1,1, Formulación del problema.....12 1,2, Objetivos de la Investigación.....12 Objetivo General:.....12 Objetivos Específicos.....13 1,3, Justificación de la Investigación.....13 CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA.....13 2,1, Bases Teóricas.....13 2,2,1, GNS3..... 13 2,1,1, Red tradicional de Centro de datos.....14 2.1.2, Spine and Leaf..... 16 2.1.3, VLAN..... 17 2.1.4, VXLAN..... 18 2.1.5,

# ÍNDICE DE CONTENIDOS

DECLARACIÓN Y AUTORIZACIÓN ..... ii

APROBACIÓN DEL TUTOR ..... iii

INFORME DE TURNITÍN ..... iv

ÍNDICE DE CONTENIDOS ..... 5

ÍNDICE DE FIGURAS ..... 8

INTRODUCCIÓN ..... 13

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA ..... 15

1.1. Formulación del problema ..... 15

1.2. Objetivos de la Investigación ..... 15

1.2.1. Objetivo General ..... 15

1.2.2. Objetivos Específicos ..... 16

1.3. Justificación de la Investigación ..... 16

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA ..... 17

2.1. GNS3 ..... 17

2.1.1. Ventajas de utilizar GNS3. ..... 17

2.1.2. Red tradicional de Centro de datos....18

2.2. Spine and Leaf....20

2.2.1. Ventajas de la arquitectura Spine and leaf. ....20

2.3. VLANS....21

2.4. VXLAN....22

2.4.1. Beneficios de VXLAN ....23

2.4.2. VTEP....24

2.4.3. VNI (Virtual Tunnel Endpoint)....25

2.4.4. VNID (Virtual Network ID)....26

2.4.5. Spine....26

2.4.6. Bridge-Domain....26

2.5. EVPN ....26

2.5.1. Beneficios que ofrece EVPN....27

2.6. BGP ....29

2.6.1. Sistema Autónomo (AS) ....29

2.6.2. BGP Externo e Interno ....30

2.7. Overlay-Underlay....30

2.8. Open Networking....32

2.8.1. Beneficios del Open Networking sobre la red tradicional ....35

2.9. Cumulus Linux....36

2.9.1. Arquitectura de Cumulus Linux ..... 37

2.9.2. Beneficios de Cumulus Linux ..... 39

2.9.3. Propósito de Cumulus Linux ..... 40

2.10. DevOps y NetDevOps ..... 40

2.11. Ansible ..... 43

2.11.1. Arquitectura Ansible ..... 44

2.11.2. Ventajas de Ansible ..... 45

CAPÍTULO III: METODOLOGÍA ..... 47

3.1. Tipo de Investigación ..... 47

3.2. Diseño de Investigación ..... 47

3.3. Técnicas e instrumentos de recolección de datos ..... 47

3.4. Técnica de Análisis de Datos ..... 48

CAPÍTULO IV: EMULACIÓN DE LA INFRAESTRUCTURA ..... 49

4.1. Etapas ..... 49

CONCLUSIONES ..... 73

RECOMENDACIONES ..... 76

REFERENCIAS BIBLIGRÁFICAS ..... 77

## ÍNDICE DE FIGURAS

**Figura 1** *Red GNS3* ..... 17

**Figura 2** *Red Tradicional* ..... 19

**Figura 3** *Red Spine and Leaf* ..... 20

**Figura 4** *VLANS* ..... 22

**Figura 5** *Header VXLAN* ..... 23

**Figura 6** *VTEP* ..... 25

**Figura 7** *VXLAN Header* ..... 25

**Figura 8** *Ejemplo de Arquitectura: Spine and Leaf L3 ECMPXVLANS & EVPN* ..... 27

**Figura 9** *EVPN* ..... 28

**Figura 10** *Ejemplo BGP para dos clientes diferentes* ..... 29

**Figura 11** *BGP Externo e Interno* ..... 30

**Figura 12** *Red Underlay* ..... 31

**Figura 13** *Red Overlay* ..... 32

**Figura 14** *Open Networking* ..... 33

**Figura 15** *Open Onie* ..... 34

**Figura 16** *Open Networking* ..... 35

**Figura 17** *Network Software Modules Cumulus Linux* ..... 36

**Figura 18** *Arquitectura Cumulus Linux* ..... 37

**Figura 19** *Cumulus Linux* ..... 38

**Figura 20** *Command Line Cumulus Linux* ..... 39

**Figura 21** *DevOps Tools* ..... 42

**Figura 22** *Open Source NetDevOps Tools* ..... 43

**Figura 23** *Ansible* ..... 44

**Figura 24** *Playbook* ..... 45

**Figura 25** *Wireshark capture* ..... 48

**Figura 26** *Arquitectura de la práctica* ..... 49

**Figura 27** *Data Center Oeste* ..... 50

**Figura 28** *Acceso a Daemons Cumulus* ..... 50

**Figura 29** *Activación de protocolos* ..... 51

**Figura 30** *Configuración Leafs* ..... 52

**Figura 31** *Configuración de Spine* ..... 53

**Figura 32** *Pruebas de ICMP Spine & Leaf DC OESTE* ..... 54

**Figura 33** ..... 55

**Figura 34** ICMP entre PC1-PC2 OESTE ..... 55

**Figura 35** *DC ESTE* ..... 56

**Figura 36** *Configuración Spine DC - ESTE* ..... 57

**Figura 37** *Configuración Leaf DC - ESTE* ..... 58

**Figura 38** *ICMP Nodos DC ESTE* ..... 59

**Figura 39** *IP PC ESTE* ..... 60

**Figura 40** *ICMP PC1-PC2 ESTE*.....60

**Figura 41** *ICMP entre DCs ESTE Y OESTE*.....61

**Figura 42** *ICMP PC1- OESTE Y PC2 - ESTE*.....62

**Figura 43** *VXLAN (VNI) 10*.....63

**Figura 44** *Rutas BGP Leaf*.....64

**Figura 45** *Ruta BGP OESTE*.....65

**Figura 46** *Conexión SW hacia todos los nodos*.....66

**Figura 47** *Clave SSH*.....67

**Figura 48** *Hosts en Asible*.....68

**Figura 49** *Llaves SSH Ansible*.....69

**Figura 50** *Conexión hacia todos los Nodos*.....70

**Figura 51** *Playbook de configuración*.....71

**Figura 52** *Resultado del despliegue de configuración Playbook Ansible*.....72

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

MAESTRIA EN TECNOLIGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN

Y ADMINISTRACIÓN DE TI

# SIMULACION SOBRE GNS3 DE UNA ARQUITECTURA DE RED SPINE AND LEAF PARA UN CENTRO DE DATOS MODERNO

Autor: Cristian Morales

Director -Tutor: Dr. Gustavo Salazar

Fecha: 30/10/2022

## RESUMEN

A medida que transcurre el tiempo, las empresas se han enfrentado a la transformación digital y aún más con el impacto global de la pandemia lo han hecho a un ritmo muy acelerado, el cual ha demandado aprovisionarse de servicios para mantener la continuidad del negocio, tecnologías que otorguen una mayor rapidez y performance en la red para suplir las necesidades que demandan las tecnologías de nueva generación; implementar nuevos protocolos de comunicación de red no solo mejorará el rendimiento y generará un mayor ROI (retorno sobre la inversión), sino que también permitirá mantenerse competitivos en el mercado, por lo cual este caso de estudio se basa en establecer una prueba de concepto para analizar la factibilidad de implementar una red Spine and leaf a través de protocolos como: VXLAN, EVPN y automatizado con Ansible, en un entorno de virtualización controlado en GNS3 utilizando software libre como NOS (Cumulus Linux) con el fin de optimizar la infraestructura de red de un centro de datos moderno.

Una emulación en un entorno virtualizado nos demostrará las ventajas al utilizar una red *Spine and leaf* con los protocolos de nueva generación.

En base a los resultados podría ser una propuesta para el aprovisionamiento de los clientes y proveedores de centros de datos modernos, teniendo en cuenta que nos estamos preparando para la industria 4.0.

**Palabras clave:** Cumulus-Linux; EVPN; VXLAN; Spine and Leaf; VTEP; Open-Networking; Overlay-Underlay; Ansible

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

MAESTRIA EN TECNOLIGÍAS DE LA INFORMACIÓN MENCIÓN GESTIÓN Y ADMINISTRACIÓN DE TI

# SIMULATION ON GNS3 OF A SPINE AND LEAF NETWORK ARCHITECTURE FOR A MODERN DATA CENTER

Autor: Cristian Morales

Director -Tutor: Dr. Gustavo Salazar

Date: 30/10/2022

## ABSTRACT

As time goes by, companies have faced digital transformation and even more so with the global impact of the pandemic, they have done so at a very fast pace, which has demanded the provision of services to maintain business continuity, technologies that provide greater speed and performance in the network to meet the needs demanded by new generation technologies; Implementing new network communication protocols will not only improve performance and generate a higher ROI (return on investment), but will also allow them to remain competitive in the market, for which this case study is based on establishing a proof of concept for analyze the feasibility of implementing a spine and leaf network through protocols such as: VXLAN, EVPN and automated with Ansible, in a virtualization environment controlled in GNS3 using free software such as NOS (Cumulus Linux) in order to optimize the network infrastructure of a modern data center.

An emulation in a virtualized environment will show us the advantages of using a spine and leaf network with the new generation protocols.

Based on the results, it could be a proposal for the provisioning of clients and providers of modern data centers, considering that we are preparing for industry 4.0.

**Keywords:** Cumulus-Linux; EVPN; VXLAN; Spine and Leaf; VTEP; Open-Networking; Overlay-Underlay, Ansible

# INTRODUCCIÓN

Las redes en los centros de datos van evolucionando constante y rápidamente, esto debido a la gran cantidad de información que procesan, gracias también a la transformación digital y el acercamiento a la industria 4.0, demandando muchos más recursos para suplir la carga de trabajo de las nuevas tecnologías, el cual requiere un mayor performance, rapidez en la red, escalabilidad, resiliencia y menor latencia la cual no ha logrado ser suplida por los centros de datos tradicionales: los centros de datos tradicionales con su arquitectura de 3 capas (núcleo, distribución-agregación y acceso), que maneja el tráfico de norte a sur (tráfico que se mueve entre el centro de datos y el resto de toda la red) que también utilizan el protocolo de árbol de expansión (STP) considerado "La edad de piedra y bronce" , según (Salazar-Chacón, 2020), el tráfico del modelo tradicional conduce a problemas de rendimiento, escalabilidad y alta latencia en los centros de datos con grandes flujos de datos, debido a la dependencia de tecnologías como la virtualización, el Big Data, las redes definidas por software y los recursos de almacenamiento compartidos que admiten aplicaciones emergentes con uso intensivo de datos, para reducir la latencia y mejorar las comunicaciones de servidor a servidor , los centros de datos han optado por realizar la migración a una arquitectura *Spine-Leaf*, topología de dos capas "La era del Renacimiento", una topología de malla completa, la arquitectura spine and leaf está pensado para los centros de datos de nueva generación debido a su escalabilidad, fiabilidad y un mejor rendimiento, ayuda a garantizar que el tráfico tenga siempre el mismo número de saltos hasta su próximo destino, por lo que mejora de una manera importante la latencia, de igual manera al no requerir STP habilita rutas redundantes entre switches por protocolos de enrutamiento múltiples rutas de mismo costo (ECPM) esto ayuda a balancear el tráfico y evitar bucles de red, al tener una mayor demanda de entornos físicos de infraestructura de red, un servidor tiene varias máquinas virtuales cada una con su dirección MAC, esto hará una tabla de direcciones MAC mucho más grande y más si toma de cientos o miles de máquinas

virtuales, en caso de que se segmenten por VLANS solo tendrá un máximo de 4094 VLANS, para los centros de datos multi-tenant ya no es suficiente las VLANS para aislar a los inquilinos por lo cual se empezó a implementar el protocolo de VXLAN el cual permite tener 16,777,216 VXLANs el cual suple las necesidades de los proveedores de servicios (ISPs), VXLAN es un tipo de vpn que trabaja (overlay) de capa 2 operando sobre una infraestructura (underlay) de capa 3, VXLAN al no depender de STP (Spanning-tree-Protocol) puede balancear la carga a través de todos los links activos evitando el desuso de los recursos en la infraestructura, adicional a esto automatizaremos la red con un motor open source como lo es Ansible que nos permite preparar la infraestructura, gestionar la configuración y organizar los sistemas, entre otros procedimientos de IT.

# CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

## 1.1.Formulación del problema

Actualmente con la transformación digital, las empresas han requerido un gran cambio para suplir las necesidades del negocio: los Data Center Tradicionales con su modelo jerárquico de 3 capas comenzaron a presentar problemas de latencia, rendimiento, no respondían de manera adecuada a las aplicaciones que demandaban un mayor performance y menor latencia, tienen poca escalabilidad y demandaban un mayor gasto en Capex y Opex, al tener más equipos se complica mantener una gestión adecuada de los recursos y es más propenso a presentar errores humanos al realizar mantenimientos, configuraciones, etc. Esto es un riesgo para la continuidad del negocio, hoy en día la infraestructura de los data center se han modernizado a un modelo jerárquico de 2 capas spine and leaf, la cual trae una mayor rapidez, simplicidad, soporta nuevos protocolos de comunicación (VXLAN) que es muy adecuado para los data center multi-tenant , con spine and leaf mejoramos el desempeño de comunicaciones de los servidores que tengan hiperconvergencia (latencia de propagación y latencia de procesamiento), adicional conFigurar cientos y miles de equipos uno por uno involucra una pérdida de recursos, por lo cual utilizaremos Ansible como motor open source para optimizar la red y desde un solo docker enviar la configuración a todos los switchs , el cual genera un mayor valor y optimización de recursos.

## 1.2.Objetivos de la Investigación

### *1.2.1. Objetivo General*

Realizar una prueba de concepto en GNS3 que permita visualizar y validar la factibilidad de uso de una red *spine and leaf* para la comunicación de data centers en centros de datos modernos.

### *1.2.2. Objetivos Específicos*

- Emular una infraestructura spine and leaf utilizando la herramienta de virtualización GNS3.
- Definir protocolos de nueva generación como VXLAN, Evpn para la comunicación undelay y overlay.
- Establecer una arquitectura en Cumulus Linux como parte del Open Networking.
- Automatizar la red diseñada con Ansible.

## 1.3.Justificación de la Investigación

Hoy en día los centros de datos han evolucionado, con esto también sus cargas de trabajo, las aplicaciones demandan un mayor rendimiento y recursos en red, por lo cual se requieren data centers que respondan con rapidez a las demandas de los usuarios y para que esto se cumpla se debe implementar una infraestructura acorde a la necesidad.

Los centros de datos modernos con la arquitectura de spine and leaf nos permite optimizar a los centros de datos multi-tenant.

# CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

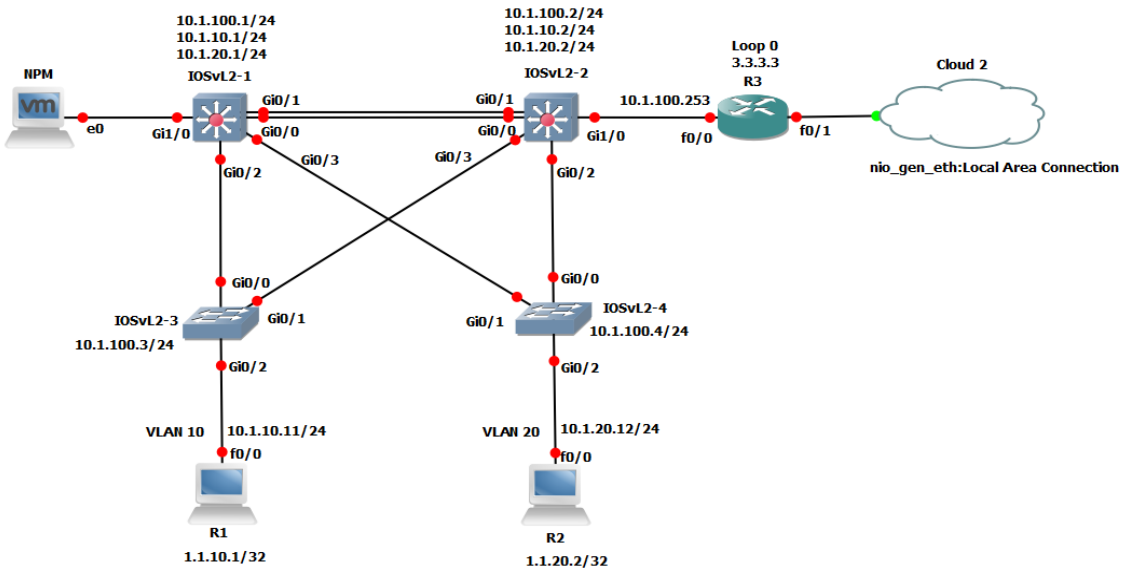
## 2.1.GNS3

GNS3 es un emulador grafico de red que nos permite diseñar topologías complejas. Actualmente es utilizado por un sin número de ingenieros en redes en todo el mundo para diseñar, probar, configurar y emular soluciones de redes virtuales reales, gns3 consta de dos componentes de software; el software todo en uno que se instala en el PC local, por otro lado, la máquina virtual GNS3 (VM) la cual utilizaremos para nuestra prueba de concepto se va a ejecutar localmente en el PC utilizando como software de virtualización al Vmware Workstation.

GNS3 permite emular el hardware de un dispositivo ejecutando las imágenes reales y también permite simular las características y funcionalidades de los dispositivos.

**Figura 1**

*Red GNS3*




*Nota*. Adaptado de *Getting started with GNS3*, 2021, GNS3 (https://docs.gns3.com/docs/)

### *2.1.1. Ventajas de utilizar GNS3.*

- Software de código abierto

- No requiere licencias
- No tiene limitación en la cantidad de dispositivos compatibles, la limitación se da de los recursos del equipo
- Admite entornos de múltiples proveedores.

### *2.1.2. Red tradicional de Centro de datos*

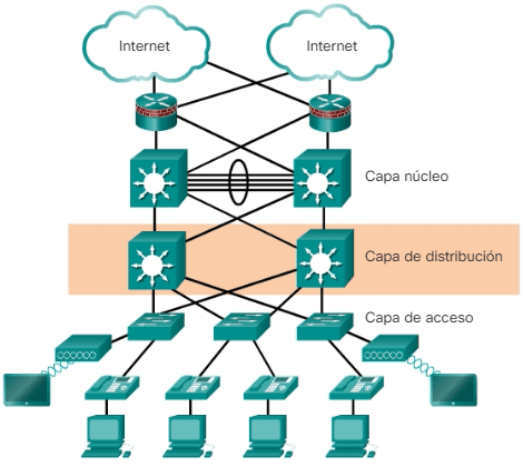
Los centros de datos tradicionales tienen un conjunto de software y hardware heterogéneo creado para alojar servicios y almacenar información de la compañía. Con esto los administradores de TI se enfrentan a grandes desafíos como la administración, gestión y operación en los centros de datos, requiriendo gestionar los recursos de manera óptima y adecuada para minimizar los problemas físicos o lógicos que pudieran terminar afectando al usuario.

#### 2.1.2.1.Características de una red tradicional de Centro de datos.

La infraestructura de red de un centro de datos está basada en el modelo jerárquico de 3 capas como se indica en la Figura 2; este modelo de red jerárquica ofrece una interconexión de redes escalable, rentable y estable para la compañía, se constituye de:

**Figura 2**

*Red Tradicional*




*Nota*. Adaptado de *Diseño jerárquico red*, 2006, CCNA (https://ccnadesdecero.es/diseno-jerarquico-de-redes/)

- Capa de núcleo (Core)

Es la capa central, la parte superior de la jerarquía, es la responsable de transportar grandes cantidades de tráfico, generalmente es un equipo de capa L3.

- Capa de distribución (Distribution)

Se le denomina capa de grupo de trabajo y es el punto de comunicación entra la capa de acceso y el núcleo, garantiza que los paquetes se enruten de manera correcta entre la subred y las VLANS, proporciona una conectividad basada en política.

- Capa de Acceso (Access)

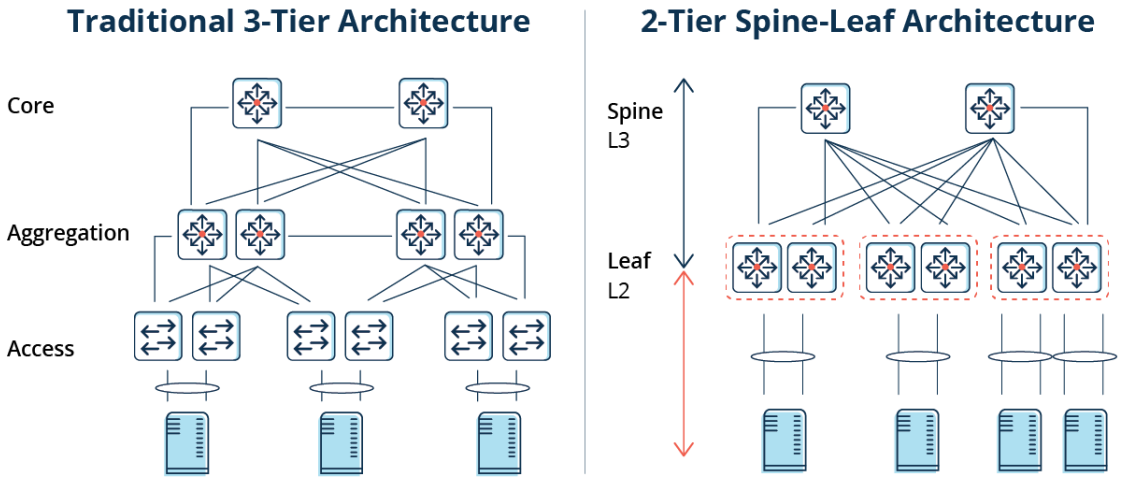
Controla el acceso de los usuarios y grupos de trabajo de la red interna, están conectados a los dispositivos finales (pcs, servidores, aps, impresoras, etc), garantiza la entrega de paquetes a los dispositivos finales.

## 2.2.Spine and Leaf

La arquitectura spine and leaf es una topología de centro de datos que tiene dos capas: "spine" o columna vertebral y "leaf" las ramificaciones, la capa *spine* está formada por switches que realizan el enrutamiento y funcionan como la columna de la red, la capa *leaf* consta de switches de acceso que agregan el tráfico desde los servidores y se conectan directamente al spine, como muestra en la Figura 3.

**Figura 3**

*Red Spine and Leaf*




*Nota*. Adaptado de *What is Spine-Leaf Architecture*, 2022, HPE Aruba networking (https://www.arubanetworks.com/faq/what-is-spine-leaf-architecture/)

La arquitectura *spine and leaf* está diseñado para dar una mayor conectividad entre los switches a través del centro de redes llegar a obtener una mejor escalabilidad, eficiencia, fiabilidad y un mejor rendimiento.

### *2.2.1. Ventajas de la arquitectura Spine and leaf.*

- Una de las principales ventajas es la latencia mejorada, los cuellos de botella reducidos, el ancho de banda ampliado y la escalabilidad

- Spine and leaf utiliza todos los enlaces de interconexión, en los centros de datos modernos puede haber cientos o miles de servidores conectados a la red, por lo cual el conmutador leaf se puede implementar como un puente entre el servidor y la red central, cada leaf se conecta a todas las Spines
- Eliminación del protocolo Spanning Tree (STP)
- Mayor cableado de switches de puerto fijo que de modelos modulares para realizar la red troncal
- Un escalamiento horizontal frente al aumento de infraestructura

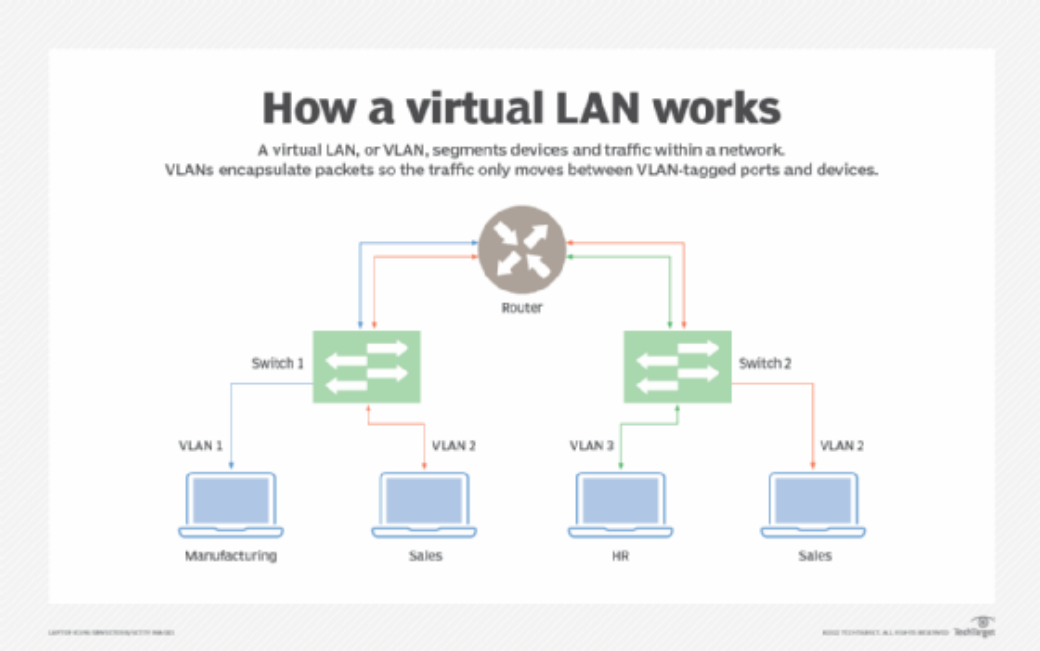
**2.3.VLANS**

Una VLANS es una LAN independiente lógica, las cuales se permiten segmentar los dominios de broadcast, crear grupos de host de manera lógica como si estuvieran en el mismo concentrador, incluso estando en otro segmento diferente de la red esto dependiendo de la infraestructura física de la compañía (Tigre Cortes & Mejía, 2012).

El protocolo solo anota el ID de VLANS real (VID) en los doce últimos bits. La longitud de la sección de trama significa que hay disponible 4.096 VLANS diferentes. Utiliza el protocolo IEEE 802.1q que define un sistema de etiquetado de VLANS para las tramas de Ethernet y los procedimientos que lo acompañan para ser utilizados por switches y routers en el manejo de dichas tramas.

**Figura 4**

*VLANS*




*Nota.* Adaptado de *VLANS (Virtual LAN)*, 2022, TechTarget (https://www.techtarget.com/searchnetworking/definition/virtual-LAN)

## 2.4.VXLAN

VXLAN (*Virtual Extensible LAN*) definida en el RFC 7348 del año 2014, es una nueva tecnología , la cual nos permite solventar problemas de escalabilidad en la infraestructura de data center modernos , VXLAN es un tipo de VPN de capa 2 overlay (MAC) operando sobre una infraestructura underlay (transporte) capa 3 (UDP/IP) , para lograr ese túnel VXLAN emplea una encapsulación similar a la del tráfico con VLANS en cuanto a que va encapsulando la trama con el direccionamiento MAC dentro de una paquete IP, permite la extensión de dominios VLANS a través de muchos enrutadores manteniendo el direccionamiento original con casi 16 millones de hosts disponibles a través de la red , las VLANS no puede extenderse a través de un dominio de capa 3, VXLAN si puede hacerlo al tratar dos dominios de capa dos separados y haciéndoles

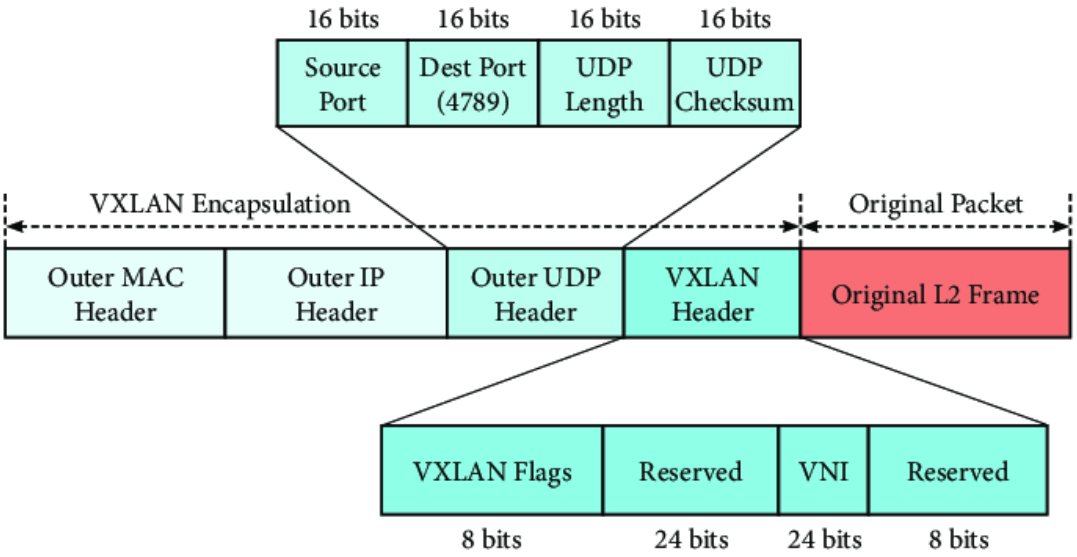
parecer como uno, no depende del protocolo de spanning-tree (STP) para converger ya que emplea protocolos de capa 3 (Salazar Chacón, 2022 b)

El Datagrama UDP se encapsula en paquetes IP, la dirección de IP destino puede ser:

- Unicast: representa la dirección del VTEP destino
- Multicast: para manejo de broadcast como ARP

**Figura 5**

*Header VXLAN*




*Nota.* Adaptado de *VXLAN PACKET ENCAPSULATION*, 2021, ResearchGate (https://www.researchgate.net/figure/VXLAN-Packet-Encapsulation_fig1_354232739)

### *2.4.1. Beneficios de VXLAN*

- La mayoría de DC enfocados en entregar servicios en la nube, ISPs con cientos o miles de clientes se enfrentan a la dificultad por la baja escalabilidad que otorgan las VLANS con sus 4094 tags, la cual reduce la cantidad de segmentación con VXLAN se aumenta considerablemente a más de 16 millones el cual solventa el problema de escalabilidad y alta disponibilidad

- Ayuda al balanceo de carga a través de todos sus enlaces activos evitando la perdida de recursos de infraestructura y otorgando un mayo retorno de inversión
- Al no depender de Spanning-Tree Protocol para converger ya que emplea protocolos de capa 3 optimiza la red al no tener bloqueos al contrario converge por sus enlaces activo/activo
- Las VLANS no pueden extenderse a través de un dominio de capa 3, VXLAN si al tratar dos dominios L2 separados y haciéndolos parecer como uno solo
- Aprovecha las rutas múltiples de igual costo (ECMP) para lograr un uso óptimo de la ruta en la red de transporte

### *2.4.2. VTEP*

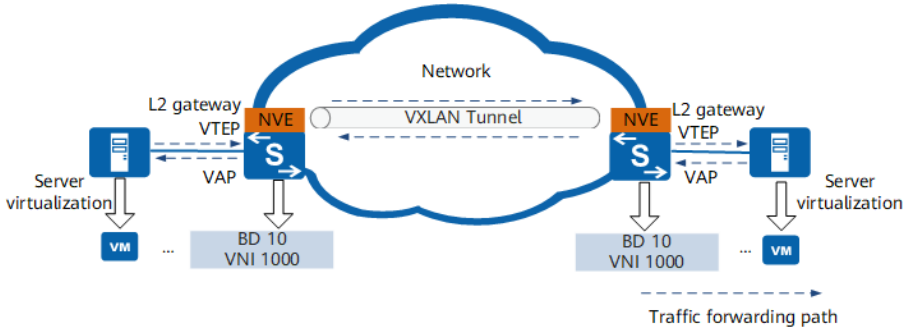
VTEP (Virtual Tunnel Endpoint) es un elemento definido ya sea físico o virtual encargado de instanciar el tuneleamiento de VXLAN y de llevar a cabo el proceso de encapsulación y desencapsulación, en los centros de datos modernos se le conoce como LEAF.

Cada VTEP tiene 2 interfaces:

- Local LAN Interfaces: Proveen la conectividad a los hosts en el segmento de LAN
- IP Interface: Interfaz orientada para VXLAN, la dirección IP de esta interfaz ayuda a identificar el VTEP en la red, también se utiliza para encapsular y desencapsular el tráfico VXLAN.

**Figura 6**

*VTEP*



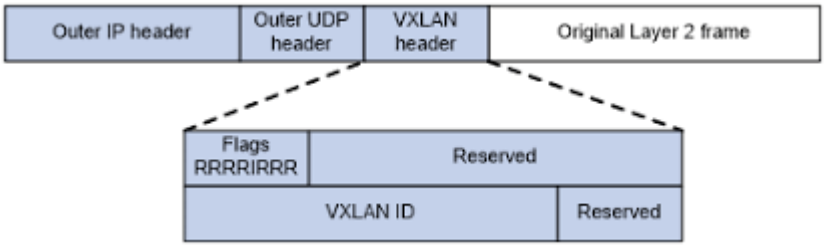

*Nota.* Adaptado de VXLAN Network Architecture, 2020, HUAWEI (https://support.Huawei.com/enterprise/en/doc/EDOC1000178119/5f42f6b3/VXLAN-network-architecture)

### *2.4.3. VNI (Virtual Tunnel Endpoint)*

Es una instancia lógica de una red VXLAN que provee servicios a nivel de capa 2 y capa 3, definiendo esa manera el dominio de broadcast de Capa 2. (Salazar Charcón, 2022 b)

**Figura 7**

*VXLAN Header*




*Nota.* Adaptado de *VXLAN Packet Format*, 2018, HPE FlexFabric 5710 Switch Series VXLAN Configuration Guide (https://techhub.hpe.com/eginfolib/networking/docs/switches/5710/52005004_VXLAN_cg/content/517705090.htm)

### *2.4.4. VNID (Virtual Network ID)*

Es el identificador lógico de 24-bits que permite obtener 16,777,216 redes lógicas en contraposición a las 4094 VLANS posibles con el tag tradicional.

### *2.4.5. Spine*

Dispositivos encargados de conectar los VTEPs entre sí.

### *2.4.6. Bridge-Domain*

Conjunto de puertos físicos o lógicos que comparten el mismo dominio de capa 2.

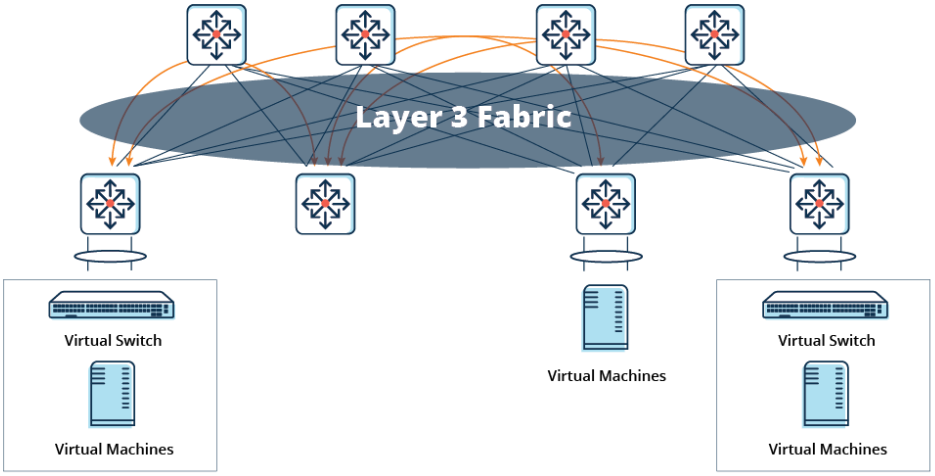
## 2.5.EVPN

EVPN es una solución de VPN portadora del servicio de siguiente generación , permite conectar sitios de clientes en diferentes locaciones mediante un puente virtual de capa 2 , EVPN consta de dispositivos de borde cliente (CE) (host enrutador o conmutador) conectados a enrutadores de borde de proveedor (PE) , unifica los planos de control para varios servicios VPN y utiliza extensiones BGP para transmitir información de accesibilidad de L2 o L3 , dividiendo el plano de reenvió del plano de control.

L2VPN tradicional carece de capacidad de equilibro de cargas y consume bastantes recursos de red, EVPN supera los estos inconvenientes y puede realizar el equilibrio de cargas, trafico, EVPN se utiliza para interconectar redes de L2 que abarcan grandes centros de datos como multi-tenant.

**Figura 8**

*Ejemplo de Arquitectura: Spine and Leaf L3 ECMPXVLANS & EVPN*



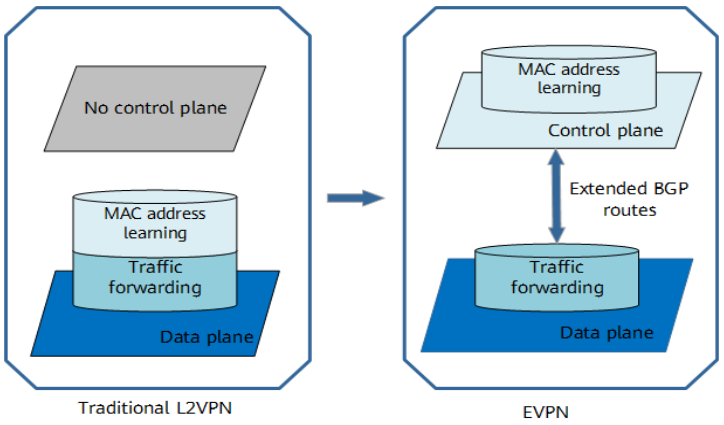

*Nota*. Adaptado de *What is Spine-Leaf Architecture*, 2022, HPE Aruba networking (https://www.arubanetworks.com/faq/what-is-spine-leaf-architecture/)

### *2.5.1. Beneficios que ofrece EVPN*

- EVPN utiliza extensiones BGP para mover el aprendizaje y él envió de direcciones MAC entre redes L2 desde el plano de datos al plano de control, esto permite que los dispositivos administren las direcciones MAC de la misma forma que administra sus rutas, lo que hace posible el balanceo de cargas entre rutas EVPN con la misma dirección Mac de destino, pero en distintos saltos

**Figura 9**

*EVPN*




*Nota*. Adaptado de *What is EVPN*, 2021, HUAWEI IP Encyclopedia (https://info.support.Huawei.com/info-finder/encyclopedia/en/EVPN.html)

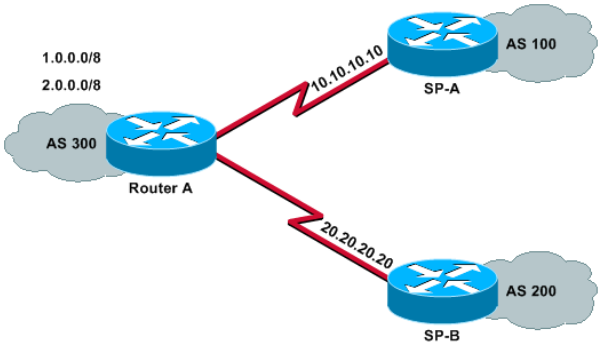
- EVPN elimina la necesidad de establecer una malla completa entre PE, esto se debe a que los PE se comunican entre sí por BGP que proporciona la función de reflexión de ruta.
- EVPN permite que los PE aprender las direcciones MAC locales utilizando ARP y aprendan dirección MAC y IP remotas usando las rutas de publicación MAC/IP, los PE almacenan las direcciones localmente, después de recibir una solicitud ARP , un PE busca su dirección almacenada en la cache local y la información de la dirección IP en función de la dirección IP de destino en la solicitud ARP, después de encontrar la información correspondiente el PE desencapsula una respuesta ARP , esto hace que se reduzca el consumo de recursos en la red porque el PE ya no necesita transmitir solicitudes ARP a los otros PE.

## 2.6.BGP

Border Gateway protocol es un protocolo de puerta de enlace exterior estandarizado diseñado para intercambiar información de enrutamiento y accesibilidad entre sistemas autónomos (AS), BGP se clasifica como un protocolo de enrutamiento de vector ruta y toma decisiones de enrutamiento basado en rutas, política de red o conjunto de reglas conFigurados por un administrador de red.

**Figura 10**

*Ejemplo BGP para dos clientes diferentes*




*Nota*. Adaptado de *Diseño jerárquico red*, 2006, CCNA (https://ccnadesdecero.es/diseno-jerarquico-de-redes/)

### *2.6.1. Sistema Autónomo (AS)*

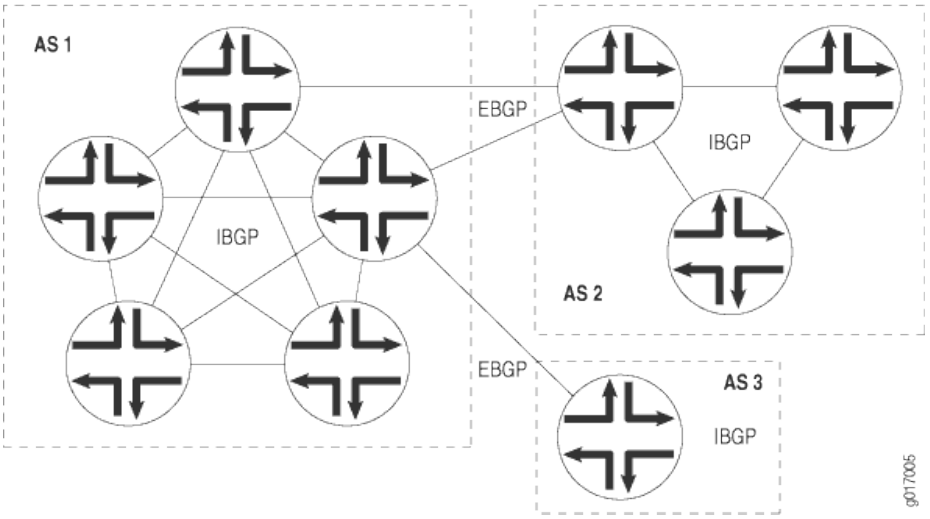
Para la información de enrutamiento que intercambian los sistemas BGP incluyen todas las rutas de su destino, la ruta a cada destino se denomina AS y la información de ruta adicional se incluye en los atributos de ruta, BGP utiliza la ruta AS y los atributos para determinar por completo la topología de red diseñada , una vez BGP entiende y aprende toda la topología de red puede detectar y eliminar loops , bucles de enrutamiento y seleccionar entre grupos de rutas para aplicar las preferencias administrativas y decisiones de políticas de enrutamiento.

### *2.6.2. BGP Externo e Interno*

BGP admite dos tipos de intercambio de información por enrutamiento, el intercambio entre diferentes sistemas autónomos e intercambio entre un solo sistema autónomo, cuando se utiliza entre mismos sistemas autónomos se denomina BGP Externo (EBGP) y cuando se utiliza dentro de un AS, se denomina BGP Interno (IBGP).

**Figura 11**

*BGP Externo e Interno*



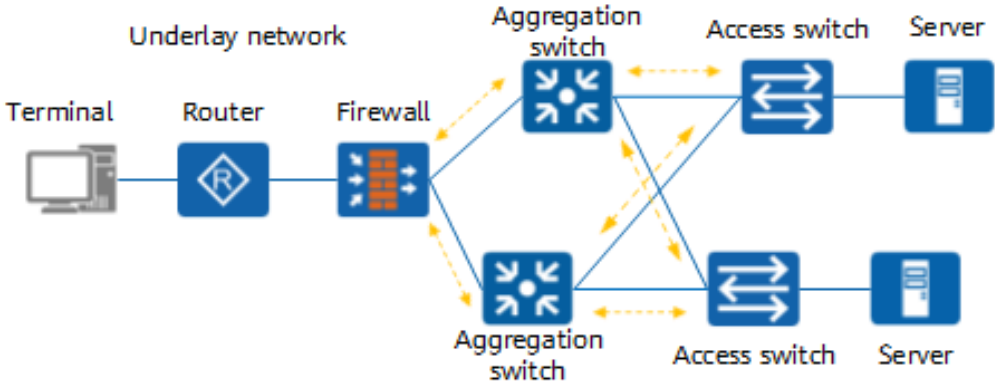

*Nota*. Adaptado de *Guía del usuario el BGP*, 2002, TechLibrary (https://www.juniper.net/documentation/mx/es/software/junos/bgp/topics/topic-map/bgp-overview.html)

### 2.7.Overlay-Underlay

- Las redes underlay como su nombre lo indica es una infraestructura física que se encuentra en la red
- En las redes underlay se pueden interconectar dispositivos como routers, firewalls, switches, balanceadores, etc
- Los equipos underlay pueden ser de capa 2 o capa 3

**Figura 12**

*Red Underlay*



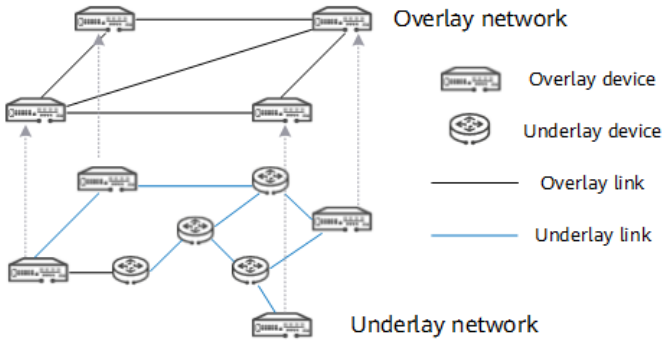

*Nota*. Adaptado de *What is an overlay Network*, 2021, Huawei IP Encyclopedia (https://info.support.Huawei.com/info-finder/encyclopedia/en/Overlay+network.html)

Overlay para deshacerse de las limitaciones de las redes undelay se pueden crear redes overlay utilizando la virtualización.

Los dispositivos de redes overlay están interconectados a través de enlaces lógicos. Las redes overlay puede ser VXLAN (virtual extensible LAN), NVGRE (virtualización de red mediante encapsulación de enrutamiento genérico), STT (árbol de expansión único), GRE, EVPN (ethernet virtual la red privada), que utilizaremos en nuestra simulación.

**Figura 13**

*Red Overlay*




*Nota*. Adaptado de *What is an overlay Network*, 2021, Huawei IP Encyclopedia (https://info.support.Huawei.com/info-finder/encyclopedia/en/Overlay+network.html)

## 2.8.Open Networking

A medida que la tecnología avanza la arquitectura de red cambia constantemente para satisfacer las necesidades del negocio por esto han optado algunas compañías en realizar una transición de las redes tradicionales al open networking con el fin de reducir costos, aumentar la flexibilidad y obtener un mayor retorno de inversión.

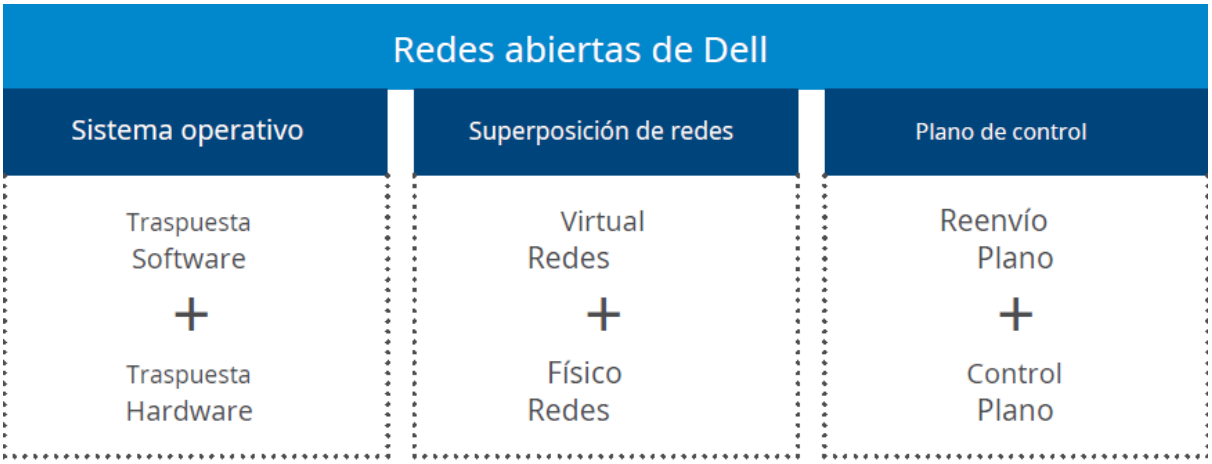
El open Networking es el concepto de desarrollar redes desagregadas separando la plataforma hardware de la red de la plataforma de software, lo que permite a la compañía no atarse de una marca y poder elegir de manera flexible los equipos, software y sistema operativo, esta estrategia del open networking permite otorgar soluciones abiertas capaces de suplir las necesidades del negocio, clientes integrando soluciones de red con el resto de plataforma de IT que conforman sus servicios, ya sea on premise o cloud, con esto el objetivo del open networking es proporcionar una red flexible, ágil y programable.

Cada fabricante de equipos de red tiene patentado sus componentes, tanto a nivel de software como hardware, por ello a pesar de los esfuerzos por estandarizar protocolos, cada

marca implementa a su manera BGP, STP, OSPF, etc. (Salazar Charcón, 2022 a)

**Figura 14**

*Open Networking*




*Nota.* Adaptado de *Dell's market changing Open Networking initiative*, 2017, DELLEMC (https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/the-economic-advantages-of-open-networking.pdf)

En los años 2013 a 2015, se desarrollaron Bare Metal switches con open source hardware con sistema tipo bios denominado ONIE (Open Network Install Enviroment), permitiendo de esta manera que un sistema operativo en un switch o router tome control del procesamiento de los datos que circula por dicho equipo.

**Figura 15**

*Open Onie*

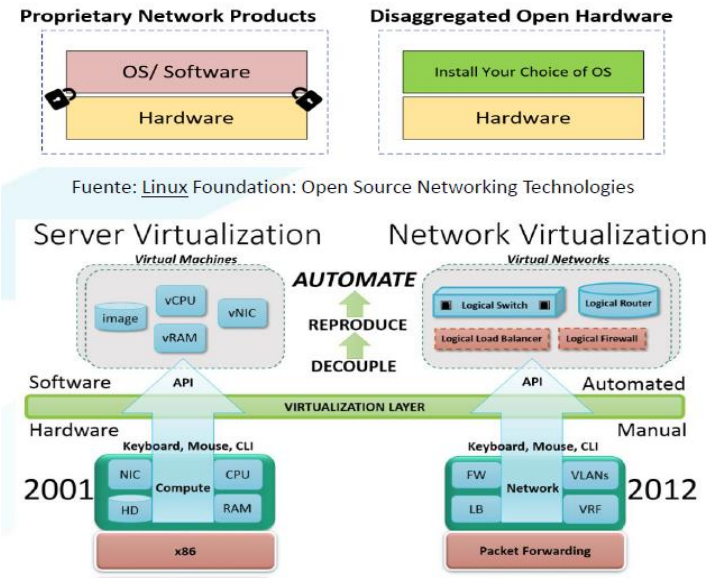



*Nota*. Adaptado de *Las redes WAN y su impacto empresarial*, 2022, Última tendencia en redes de comunicaciones (Pontificia Universidad Católica del Ecuador)

Al separar el hardware del software en los equipos de red, es posible instalar NOS (network Operating System) que sea soportado por nuestros open switchs, entre los NOS mas importantes están Poca8, Cumulus Linux que utilizaremos en nuestra simulación, Open network Linux u open switch.

**Figura 16**

*Open Networking*




*Nota*. Adaptado de *Las redes WAN y su impacto empresarial*, 2022, Última tendencia en redes de comunicaciones (Pontificia Universidad Católica del Ecuador)

### *2.8.1. Beneficios del Open Networking sobre la red tradicional*

- El open networking es más flexible que las redes tradicionales y permiten múltiples opciones para implementar la red de la empresa, con el open networking permite implementar la topología y protocolos apropiados según la arquitectura y necesidad
- Los equipos software y hardware se pueden adquirir de diferentes proveedores no se ata a una marca, esto hace posible la configuración de red más rentable
- En la red tradicional se requiere seguir protocolos específicos, mientras que el open networking permite a los desarrolladores programar la red directamente
- El open networking se administra a través de interfaces y protocolos abiertos que permite pruebas validación e implementación de redes más rápidas, aprovecha al máximo los recursos existentes, por lo cual aumenta la productividad y beneficios en

los tiempos de los recursos

- El costo de implementar open networking es más económico que una red tradicional, porque los desarrolladores, implementadores tienen libertad de elegir el equipo, el hardware abierto es más económicos y se ajustan a las necesidades del negocio, a la vez ahorran espacio, energía y costos en comparación de las redes tradicionales

## 2.9.Cumulus Linux

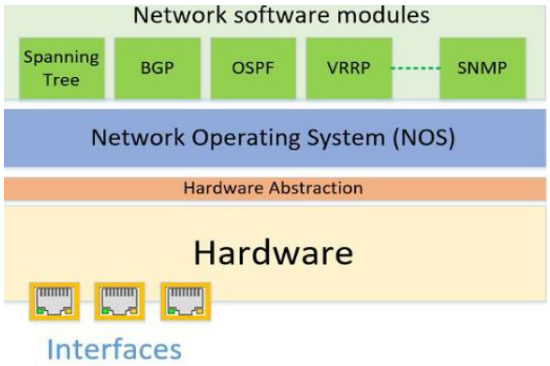
Es uno de los NOS (Network Operating Systems) más utilizados e importantes de la actualidad, cumulus Linux es una distribución GNU/Linux basada en Debían el cual permite automatizar, personalizar y escalar usando principios de escala web como los centros de datos más grandes del mundo.

El Open Hardware se basa en dos componentes como son:

- Procesamiento de paquetes mediante un chip de chipset/silicon
- CPU el cual corre él NOS

**Figura 17**

*Network Software Modules Cumulus Linux*



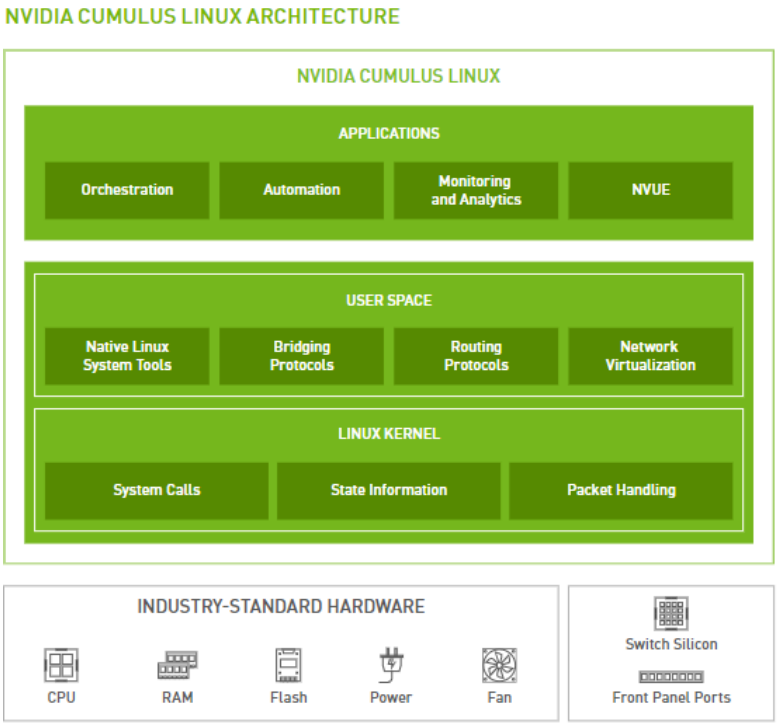

*Nota*. Adaptado de *Las redes WAN y su impacto empresarial*, 2022, Última tendencia en redes de comunicaciones (Pontificia Universidad Católica del Ecuador)

Cumulus es una de las compañías que ofrecen NOS comerciales y soporte a para bare metal switches, Cumulus Linux es empaquetado como un NOS completó basado en Debian Jessie.

### *2.9.1. Arquitectura de Cumulus Linux*

**Figura 18**

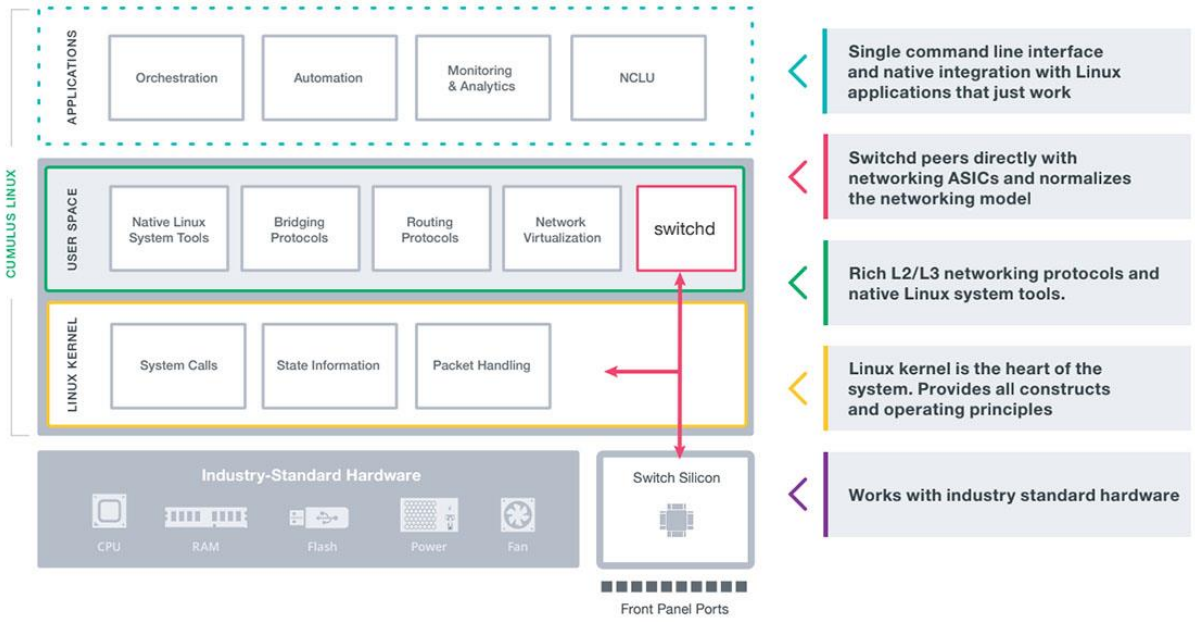
*Arquitectura Cumulus Linux*




*Nota*. Adaptado de *¿Cómo funciona Cumulus Linux?*, 2022, NVIDIA Cumulus Linux (https://www.nvidia.com/es-la/networking/ethernet-switching/cumulus-linux/)

**Figura 19**

*Cumulus Linux*

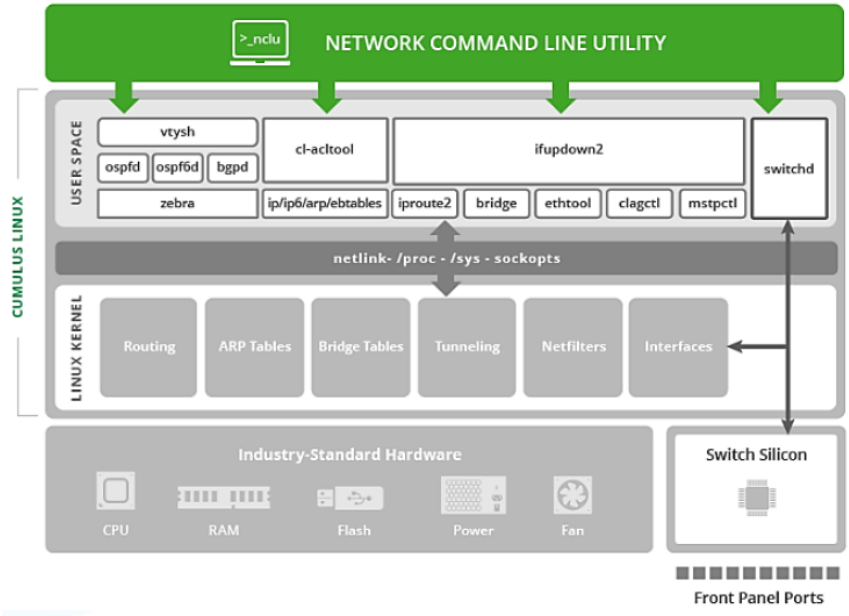



*Nota*. Adaptado de *Las redes WAN y su impacto empresarial*, 2022, Última tendencia en redes de comunicaciones (Pontificia Universidad Católica del Ecuador)

- Cumulus Linux incluye herramienta para manejar capa 2 y capa 3 más fácilmente, posee un proceso llamado *switched* el cual usa para la sincronización de switching / routing al chipset, es ideal para el open networking para configurar playbooks para ansible
- La configuración de Cumulus Linux es la base a una línea de comandos denominada NCLU (*Network Command Line Utility*)
- NCLU provee acceso consistente a comandos de los networking directamente por bash, lo que permite que los procesos sean configurado rápido y sencillo
- La utilidad *net* (NCLU Wrapper Utility) posibilita la configuración de capa 2 y capa 3, direccionamiento IP, VLANS, VXLANs entre otros

**Figura 20**

*Command Line Cumulus Linux*




*Nota*. Adaptado de *Las redes WAN y su impacto empresarial*, 2022, Última tendencia en redes de comunicaciones (Diapositivas PowerPoint)

### *2.9.2. Beneficios de Cumulus Linux*

- Reduce las complejidades y aprovecha la interoperabilidad completa en todo el data center
- Aprovecha al máximo los costos de infraestructura un menor Opex
- Automatización integran competa para los Workflows de integración y entrega continuas
- Crea redes abiertas y escalabres habilitadas por protocolos basados en estándares combinados con la libertad de Linux
- Reduce el costo de Capex y mejora el ROI

- Interfaces sin numerar aprovecha un enfoque simplificado de protocolo de internet IP para el protocolo BGP y ruta abierta más corta OSPF con sola una ip para todos los nodos leaf y para los nodos spine

- Seguridad Nativa desde OpenSSH hasta la autenticación, Cumulus Linux mantiene tus switches seguros

### *2.9.3. Propósito de Cumulus Linux*

- Redes Privadas Virtuales Ethernet (EVPN)

- Lan Virtual Extensible (VXLAN)

- Integración de VMware y OpenStack

- IP y BGP sin numerar

## 2.10. DevOps y NetDevOps

- DevOps es una estrategia de desarrollo de software que cierra la brecha entre los equipos de desarrollo y operativo para construir, probar y desplegar software de manera rápida y confiable.

- Desarrollo (DeV) desarrollo de código, pruebas y control de calidad.

- Operaciones (Ops) Arquitectura de TI, desarrollo e implementación de entornos.

- DevOps cubre varias áreas como el desarrollo, ejecución, automatización, seguridad, agilidad y pruebas. (Salazar Charcón, 2019)

  El modelo DevOps vincula tres partes:

- CI Continuous Integration: Practica en las cual los desarrolladores mantienen sus códigos en repositorios centrales y colaborativo, en el mismo lugar se realizan pruebas automáticas a la programación generada para lograr integración

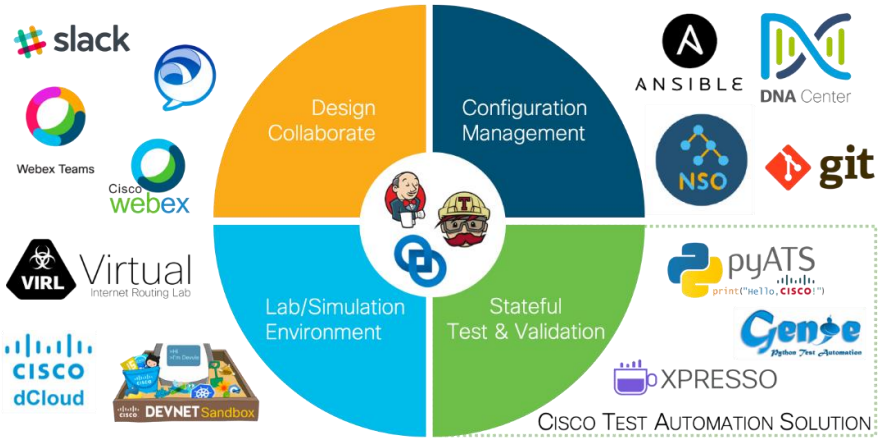
- CD Continuous Delivery: Practica donde los cambios de código son enviados automáticamente a un ambiente de pruebas después de la creación
- IaC Infraestructura as Code: Práctica donde la infraestructura de IT esta provisionada y manejada utilizando scripts

Networking Development Operations: el cual no es más que la combinación de programabilidad en la infraestructura utilizando Apis o herramientas de automatización con el fin de cumplir el ciclo cultural de Devops y así generar una sinergia entra el desarrollo e implementación.

NetDevOps aprovecha las prácticas de DevOps para garantizar que los cambios en la red sean pequeños y frecuentes pero que también se realicen de manera mucho más automatizada, eficiente y confiable , NetDevOps se puede hacer uso de Infraestructura como código para automatizar el aprovisionamiento, la configuración y la gestión de la infraestructura como puede ser servidores, switches, VMs, Firewalls, etc, esto ayuda a mejorar el tiempo de configuración de cambios, actualizar equipos y reduce el riesgo de errores humanos que demandan los cambios manuales.

**Figura 21**

*DevOps Tools*



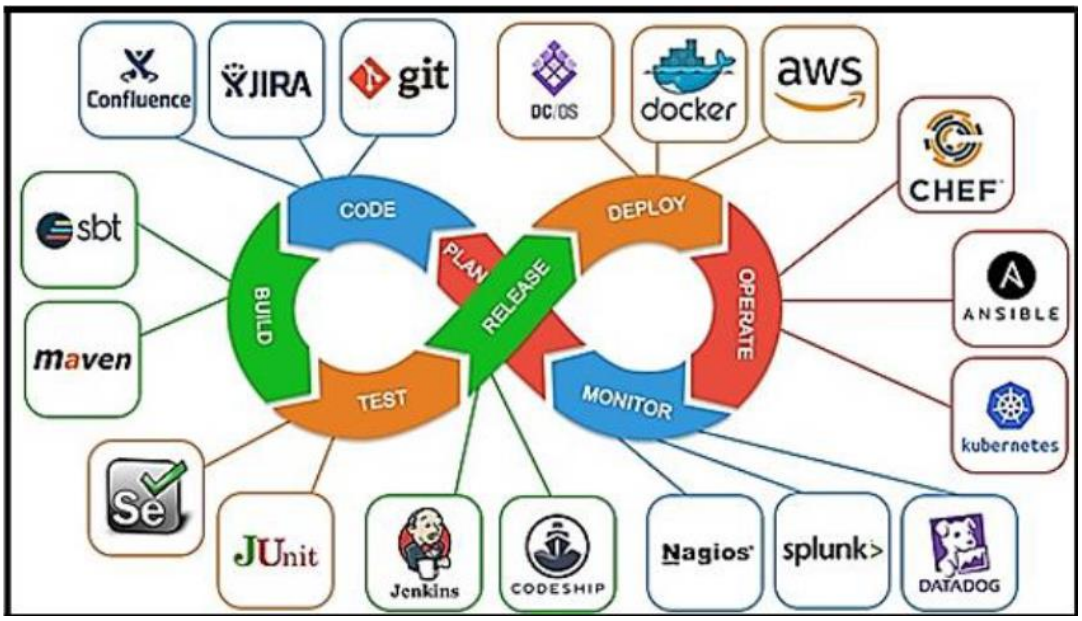

*Nota*. Adaptado de *The Anatomy of NetDevOps*, 2019, CISCO (https://blogs.cisco.com/developer/anatomy-of-netdevops)

Ansible que utilizaremos en nuestra simulación tiene una estructura modular, lo que permite tomar control de la mayoría y casi todos los dispositivos o aplicación, se ha añadidos plugins para controlar de mejor manera los equipos físicos de red o virtuales, pero estos dependen del hardware, de acorde a eso se podrá automatizar.

Algunas de las Open Source de NetDevOps se muestra en la Figura 22.

**Figura 22**

*Open Source NetDevOps Tools*




*Nota.* Adaptado de *Networking, IT y la Transformación Digital*, 2019, Comunidad de Cisco (Diapositivas PowerPoint)

## 2.11. Ansible

Ansible es un motor open source que automatiza procesos para preparar la infraestructura, permite el aprovisionamiento y configuración de máquinas virtuales, servidores, dockers y equipos de infraestructura de red, gestiona la configuración, implementación de las aplicaciones y organiza los sistemas entre otros procedimientos, esta categorizado como una herramienta de orquestación muy útil para DevOps y NetDevOps.

Ansible es una herramienta de automatización y programabilidad basada en Python.

Ansible se distribuye en Fedora, Red Hat, Centos mediante paquetes EPEL. Tiene módulos que funcionan en Json y la salida estándar se puede escribir en cualquier lenguaje. Utiliza de forma nativa secuencias de comandos YAML para describir configuraciones de

sistemas reutilizables.

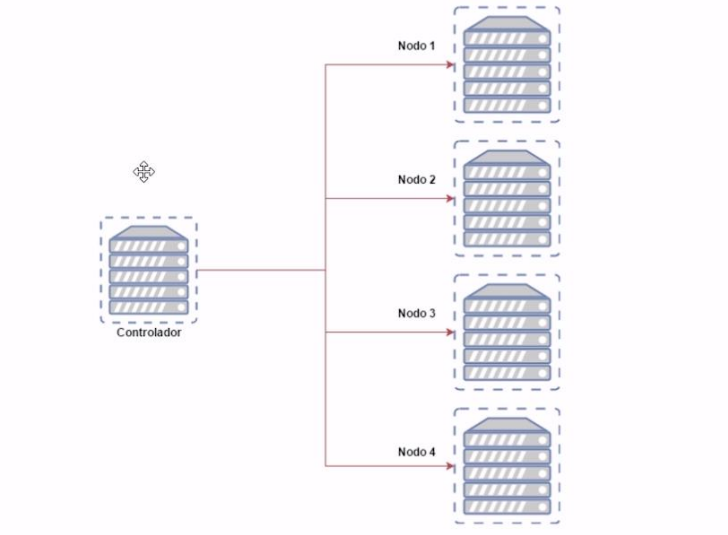
### *2.11.1. Arquitectura Ansible*

Existen dos tipos de servidores:

- Controlador: La máquina desde que comienza la orquestación
- Nodo: Es manejado por el controlador a través de SSH

**Figura 23**

*Ansible*




*Nota.* Adaptado de *Ansible: conceptos básicos*, 2022, Red Hat (https://www.redhat.com/es/topics/automation/learning-ansible-tutorial)

La estructura de ansible consta de tres partes:

- Inventory File: Archivo que enlista los equipos y los agrupa a su función, tipo o tarea que se van a desempeñar. En el archivo se especifica las direcciones IP de administración de cada equipo, también la forma de envió de las configuraciones por lo general mediante el protocolo seguro de SSH.

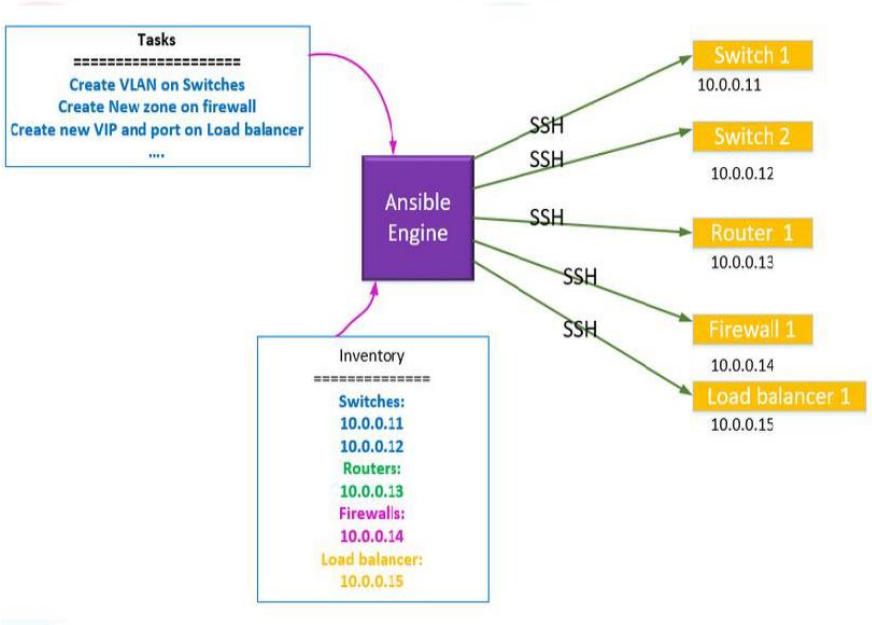
- Task: Ejecución que Ansible puede realizar en uno o más dispositivos a la vez, esto se encuentra dentro del PlayBook.

- Playbook File: Es un archivo YAML (Yet Another Markup Language Ain't Markup Language) Permite combinar tareas y tener algún control de acción al flujo de ejecución, contiene al menos un play y sirve para definir el estado deseado de un sistema.

Los plays son un conjunto ordenado de tareas que se ejecutan en las selecciones del host desde el archivo de inventario de ansible

**Figura 24**

*Playbook*




*Nota.* Adaptado de *Ansible: conceptos básicos*, 2022, Red Hat (https://www.redhat.com/es/topics/automation/learning-ansible-tutorial)

### *2.11.2. Ventajas de Ansible*

- Software claro y fácil de aprender: Su plataforma es muy intuitiva clara y fácil para el usuario, quien ejecute la herramienta podrá utilizar ficheros YAML a través de los

Playbooks lo que facilita la legibilidad al cliente y hará que las tareas tomen menor recursos de tiempo y personas.

- Rápida Implementación: Automatiza los procesos ya que no necesitaras configurar equipo por equipo ya que sus funciones permitirán desplegar la configuración de varios equipos en cuestión de minutos.

- Eficiencia de sus tareas: logra los objetivos propuestos, utilizando solamente los recursos necesarios evitando que el equipo de trabajo malgaste el tiempo en configuraciones y reduce el error humano

- Seguridad en sus procesos: al utilizar SSH permite que el cliente se conecte a la consola remota de otro equipo, minimizando los riesgos de seguridad también permite cifrar las conexiones y utilizar otros métodos criptográficos.

# CAPÍTULO III: METODOLOGÍA

## 3.1.Tipo de Investigación

En la prueba de concepto de la implementación de una red spine and leaf para un centro de datos moderno utilizaremos la investigación aplicada ya que el fin es poder aplicarlo en la realidad en un entorno de producción y sea una herramienta para los centros de datos multi-tenant, también una investigación descriptiva ya que busca conocer el funcionamiento y la estructura de los nuevos protocolos de comunicación como lo es VXLAN , Evpn y por último experimental ya que vamos a simular en GNS3 para crear un ambiente aislado para realizar las pruebas y también poder implementar una herramienta de automatización como lo es Ansible.

## 3.2.Diseño de Investigación

Se definió una prueba de concepto para la Emulación de una red spine and leaf para un centro de datos moderno y así verificar la factibilidad de estos escenarios en un entorno controlado.

La POC o prueba de concepto es un estudio de la viabilidad que se puede realizar antes de implementar un proyecto o idea, tiene como objetivo demostrar si es posible implementar alguna idea, método o producto en nuestro caso la viabilidad de implementar una red spine and leaf con VXLAN y EVPN

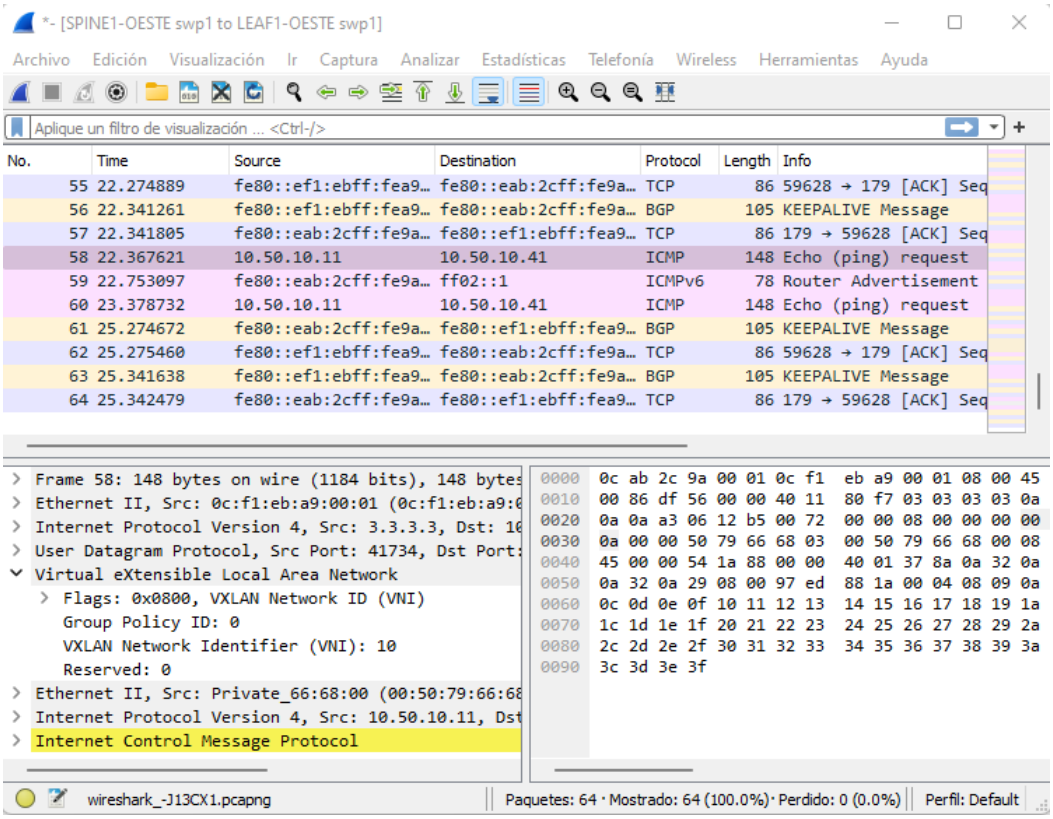
## 3.3.Técnicas e instrumentos de recolección de datos

Se utiliza la herramienta Wireshark para la toma de datos, comparación y análisis.

Wireshark es un analizador de protocolos utilizado para realizar análisis de la red y solucionar problemas de comunicación, nos permite ver qué es lo que sucede en la red, wireshark dispone de varias características como la inspección profunda de varios protocoles, una captura en vivo del análisis de la red.

**Figura 25**

*Wireshark capture*




*Fuente.* Elaboración propia

## 3.4.Técnica de Análisis de Datos

Análisis de resultado (KPI) conectividad y encapsulamiento

KPI (Key Performance Indicator o Medidor de desempeño) es una serie de métricas que se utiliza para extraer la información sobra la eficacia y productividad de las acciones que se llevan a cabo o que se quiere medir con el fin de facilitar la toma de decisiones con esto poder determinar cuáles son las más efectivas a la hora de cumplir objetivos o llenara las metas esperadas y cuales no para poder ajustar y mejore el KPI cuando se realice una nueva medición.

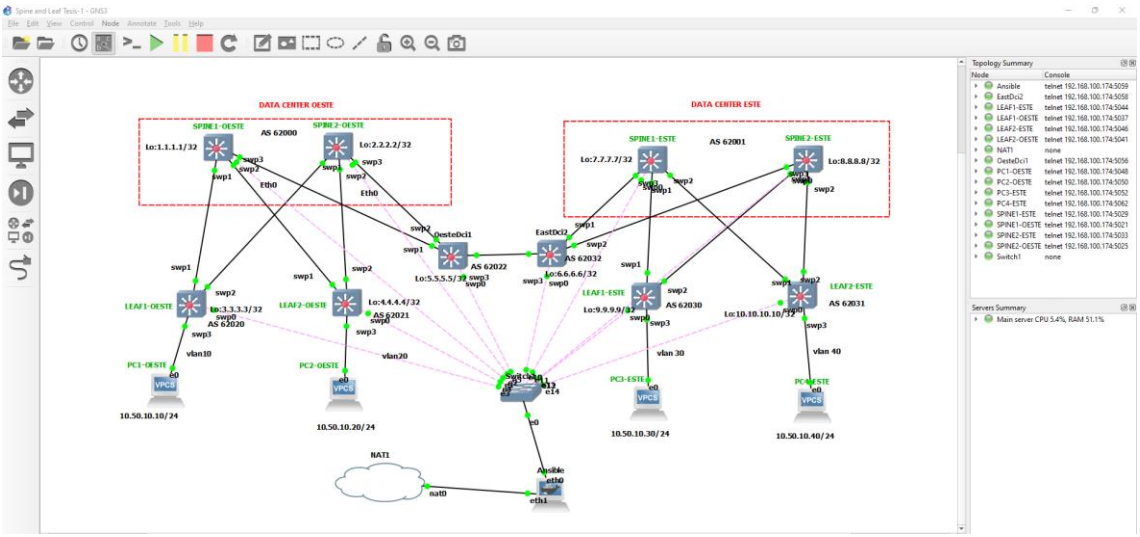
# CAPÍTULO IV: EMULACIÓN DE LA INFRAESTRUCTURA

## 4.1.Etapas

- Definimos la arquitectura que vamos a realizar, diseñamos y etiquetamos todos los nodos para poder desarrollar la emulación de una forma ordenada y correcta.

**Figura 26**

*Arquitectura de la práctica*



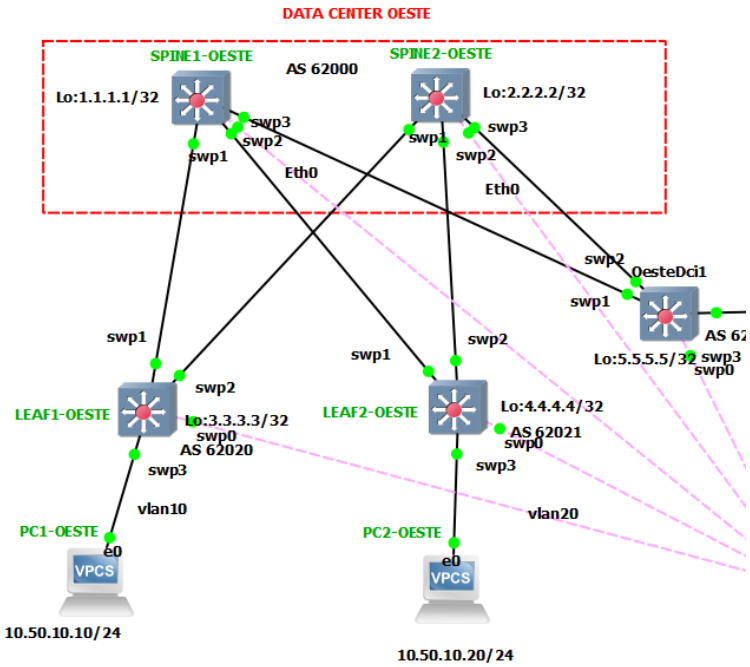

*Fuente.* Elaboración propia

La topología se comprende de dos data centers uno en el Este y el otro el Oste, cada data center dispone de dos spine que se interconectan entre si con sus dos leafs , de igual manera existe un Dci por cada data center que realiza la interconexión entre el DcEste con el DcOeste , cada leaf tiene una pc conectada para realizar las pruebas de comunicación y finalmente disponemos de un Docker de ansible para realizar la automatización, la nube de internet solo se ocupara para la actualización del Docker.

- Realizamos la configuración del Data Center OESTE con esto validamos todos los nodos dispongan de conexión

**Figura 27**

*Data Center Oeste*

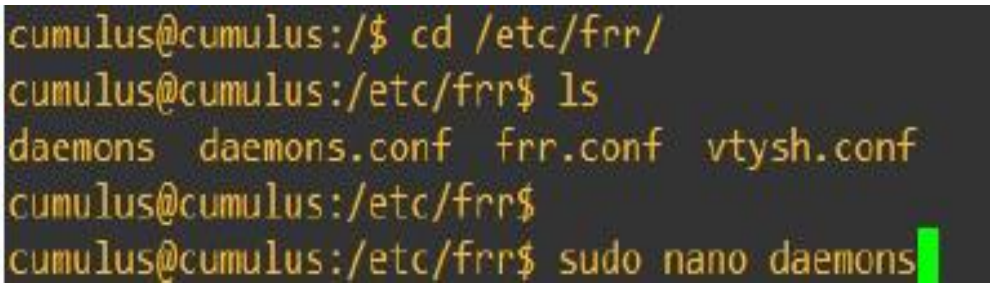



*Fuente.* Elaboración propia

- Activamos en todos los nodos los protocolos que vamos a utilizar en nuestro caso BGP

**Figura 28**

*Acceso a Daemons Cumulus*




*Fuente.* Elaboración propia

**Figura 29**

*Activación de protocolos*




*Fuente.* Elaboración propia

- Iniciamos con la configuración de los nodos de DC Oeste (Spine & Leafs)

**Figura 30**

*Configuración Leafs*

```
leaf1oeste# sh run
Building configuration...

Current configuration:
!
frr version 4.0+cl3u11
frr defaults datacenter
hostname leaf1oeste
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
interface swp1
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
interface swp2
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
router bgp 62020
 bgp router-id 3.3.3.3
 bgp bestpath as-path multipath-relax
 bgp bestpath compare-routerid
 neighbor fabric peer-group
 neighbor fabric remote-as external
 neighbor fabric description Internal Fabric Network
 neighbor fabric capability extended-nexthop
 neighbor swp1 interface peer-group fabric
 neighbor swp2 interface peer-group fabric
 !
 address-family ipv4 unicast
  network 3.3.3.3/32
 exit-address-family
 !
 address-family ipv6 unicast
  neighbor fabric activate
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor fabric activate
  advertise-all-vni
 exit-address-family
```

*Fuente.* Elaboración propia

**Figura 31**

*Configuración de Spine*

```
leaf1oeste# sh run
Building configuration...

Current configuration:
!
frr version 4.0+cl3u11
frr defaults datacenter
hostname leaf1oeste
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
interface swp1
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
interface swp2
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
router bgp 62020
 bgp router-id 3.3.3.3
 bgp bestpath as-path multipath-relax
 bgp bestpath compare-routerid
 neighbor fabric peer-group
 neighbor fabric remote-as external
 neighbor fabric description Internal Fabric Network
 neighbor fabric capability extended-nexthop
 neighbor swp1 interface peer-group fabric
 neighbor swp2 interface peer-group fabric
 !
 address-family ipv4 unicast
  network 3.3.3.3/32
 exit-address-family
 !
 address-family ipv6 unicast
  neighbor fabric activate
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor fabric activate
  advertise-all-vni
 exit-address-family
```

*Fuente.* Elaboración propia

- Pruebas de ICM entre Spine y Leaf para validar que se disponga de conexión entre todos los nodos de DC OESTE

**Figura 32**

*Pruebas de ICMP Spine & Leaf DC OESTE*

```
leaf1oeste# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=2.97 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=1.46 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=64 time=2.31 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=64 time=1.94 ms
^C
--- 1.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.466/2.175/2.975/0.553 ms
leaf1oeste# ping 2.2.2.2
PING 2.2.2.2 (2.2.2.2) 56(84) bytes of data.
64 bytes from 2.2.2.2: icmp_seq=1 ttl=64 time=2.67 ms
64 bytes from 2.2.2.2: icmp_seq=2 ttl=64 time=2.27 ms
64 bytes from 2.2.2.2: icmp_seq=3 ttl=64 time=10.2 ms
64 bytes from 2.2.2.2: icmp_seq=4 ttl=64 time=2.46 ms
^C
--- 2.2.2.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.275/4.420/10.268/3.379 ms
leaf1oeste# ping 4.4.4.4
PING 4.4.4.4 (4.4.4.4) 56(84) bytes of data.
64 bytes from 4.4.4.4: icmp_seq=1 ttl=63 time=2.67 ms
64 bytes from 4.4.4.4: icmp_seq=2 ttl=63 time=2.74 ms
64 bytes from 4.4.4.4: icmp_seq=3 ttl=63 time=2.65 ms
^C
--- 4.4.4.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.651/2.690/2.747/0.041 ms
leaf1oeste# ping 5.5.5.5
PING 5.5.5.5 (5.5.5.5) 56(84) bytes of data.
64 bytes from 5.5.5.5: icmp_seq=1 ttl=63 time=4.16 ms
64 bytes from 5.5.5.5: icmp_seq=2 ttl=63 time=3.86 ms
64 bytes from 5.5.5.5: icmp_seq=3 ttl=63 time=6.61 ms
^C
--- 5.5.5.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.866/4.882/6.613/1.230 ms
leaf1oeste#
```

*Fuente.* Elaboración propia

- Pruebas de ICM entre PC-1 de DC-OESTE  hacia PC-2 DC-EOSTE

**Figura 33**

*IP PC*

```
PC1-OESTE> show ip

NAME          : PC1-OESTE[1]
IP/MASK       : 10.50.10.11/24
GATEWAY       : 0.0.0.0
DNS           :
MAC           : 00:50:79:66:68:00
LPORT         : 20122
RHOST:PORT    : 127.0.0.1:20123
MTU           : 1500

PC1-OESTE>
```

*Fuente.* Elaboración propia

**Figura 34**

ICMP entre PC1-PC2 OESTE

```
PC1-OESTE> ping 10.50.10.21

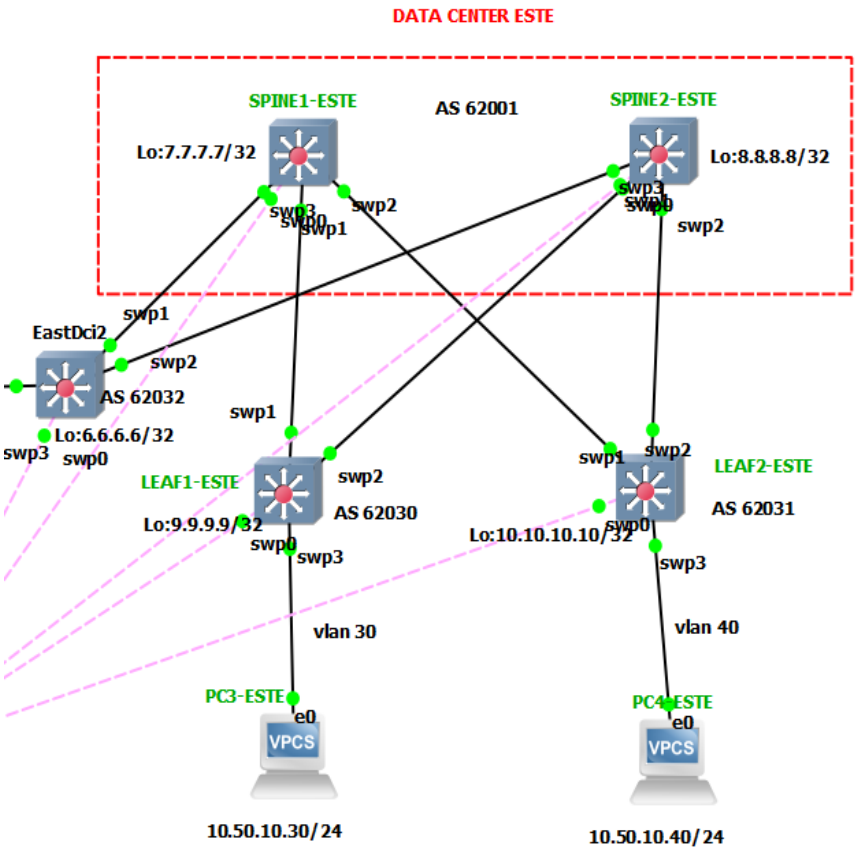
84 bytes from 10.50.10.21 icmp_seq=1 ttl=64 time=13.019 ms
84 bytes from 10.50.10.21 icmp_seq=2 ttl=64 time=12.389 ms
84 bytes from 10.50.10.21 icmp_seq=3 ttl=64 time=8.537 ms
84 bytes from 10.50.10.21 icmp_seq=4 ttl=64 time=8.710 ms
84 bytes from 10.50.10.21 icmp_seq=5 ttl=64 time=4.080 ms
^Z
```

*Fuente.* Elaboración propia

- Iniciar la configuración del Data Center ESTE

**Figura 35**

*DC ESTE*




*Fuente*. Elaboración propia

- Realizamos el mismo proceso de activar los demonios de BGP, configuramos las loopback en los nodos y los protocolos de comunicación VXLAN y BGP para la interconexión entre sedes

**Figura 36**

*Configuración Spine DC - ESTE*

```
spine1este# sh run
Building configuration...

Current configuration:
!
frr version 4.0+cl3u11
frr defaults datacenter
hostname spine1este
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
interface swp1
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
interface swp2
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
interface swp3
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
router bgp 62001
 bgp router-id 7.7.7.7
 bgp bestpath as-path multipath-relax
 bgp bestpath compare-routerid
 neighbor fabric peer-group
 neighbor fabric remote-as external
 neighbor fabric description Internal Fabric Network
 neighbor fabric capability extended-nexthop
 neighbor swp1 interface peer-group fabric
 neighbor swp2 interface peer-group fabric
 neighbor swp3 interface peer-group fabric
 !
 address-family ipv4 unicast
  network 7.7.7.7/32
 exit-address-family
 !
 address-family ipv6 unicast
  neighbor fabric activate
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor fabric activate
  advertise-all-vni
 exit-address-family
!
line vty
!
end
```

*Fuente.* Elaboración propia

**Figura 37**

*Configuración Leaf DC - ESTE*

```
leaf1este# sh run
Building configuration...

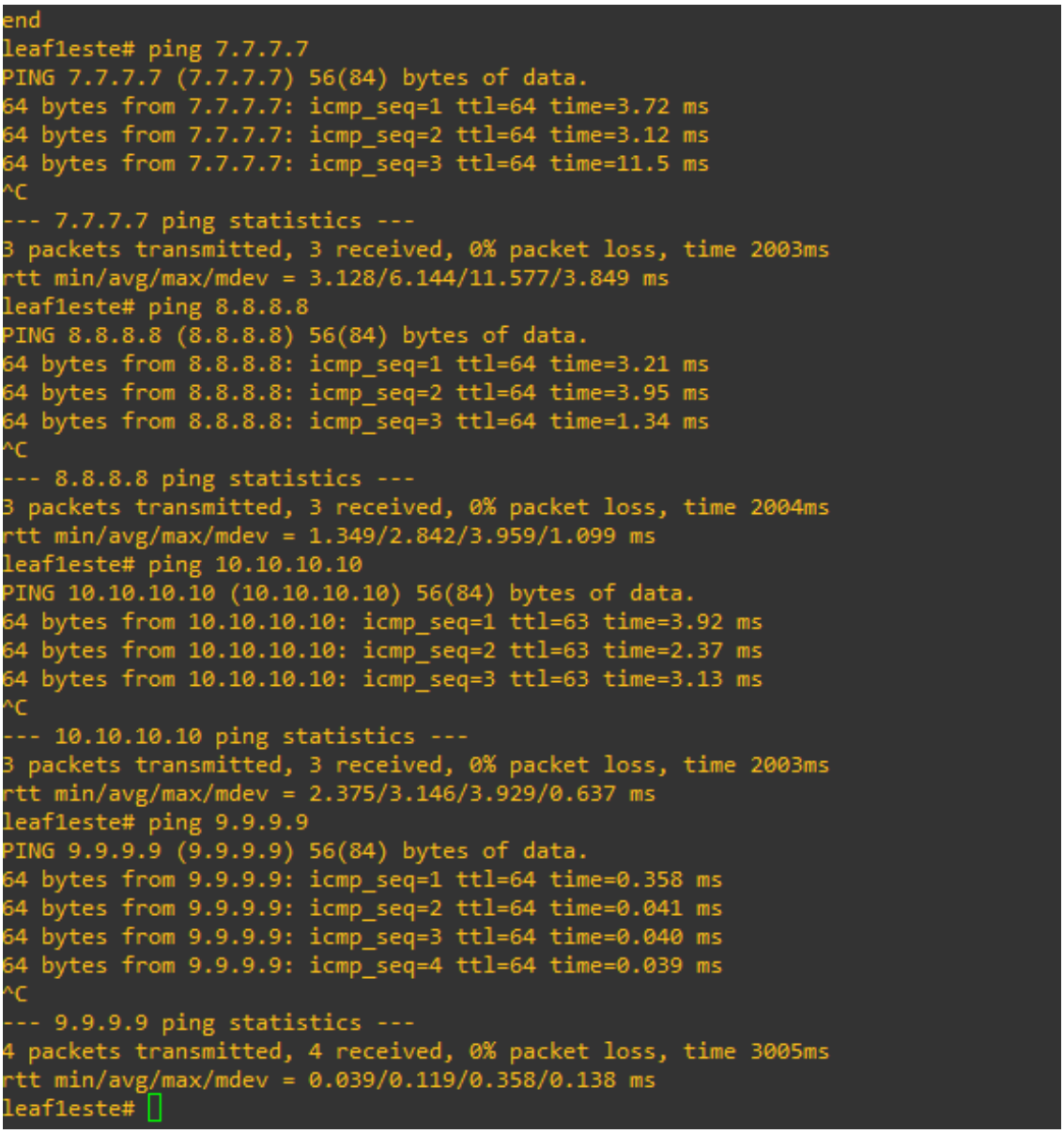
Current configuration:
!
frr version 4.0+cl3u11
frr defaults datacenter
hostname leaf1este
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
interface swp1
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
interface swp2
 ipv6 nd ra-interval 5
 no ipv6 nd suppress-ra
!
router bgp 62030
 bgp router-id 9.9.9.9
 bgp bestpath as-path multipath-relax
 bgp bestpath compare-routerid
 neighbor fabric peer-group
 neighbor fabric remote-as external
 neighbor fabric description Internal Fabric Network
 neighbor fabric capability extended-nexthop
 neighbor swp1 interface peer-group fabric
 neighbor swp2 interface peer-group fabric
 !
 address-family ipv4 unicast
  network 9.9.9.9/32
 exit-address-family
 !
 address-family ipv6 unicast
  neighbor fabric activate
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor fabric activate
  advertise-all-vni
 exit-address-family
!
line vty
!
end
leaf1este#
```

*Fuente.* Elaboración propia

- Realizar pruebas de ICMP entre Spine & Leafs para validar la conexión entre todos los nodos de DC ESTE

**Figura 38**

*ICMP Nodos DC ESTE*

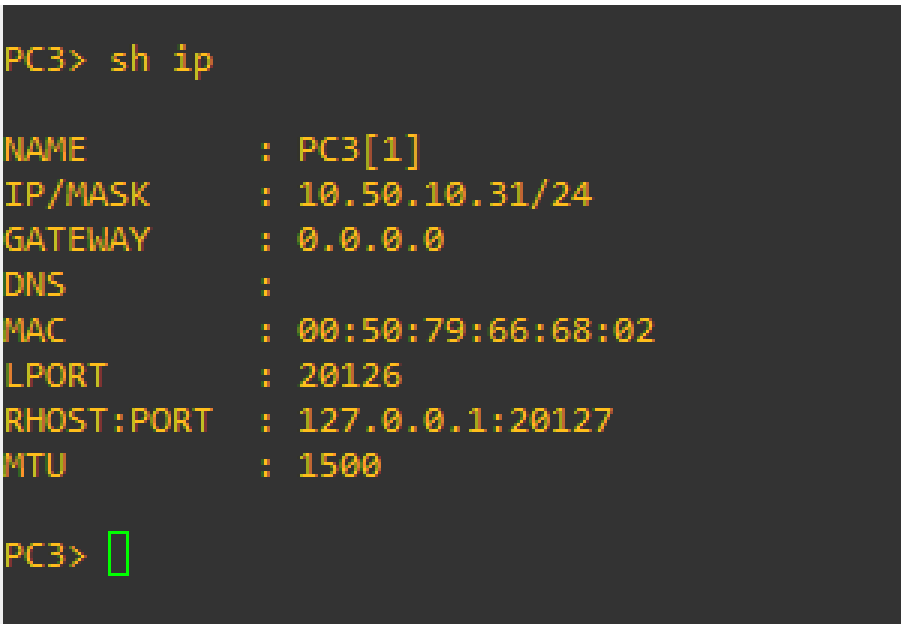



*Fuente.* Elaboración propia

- Colocamos IP a la PC-1 y PC-2 Este y realizamos pruebas de ICMP

**Figura 39**

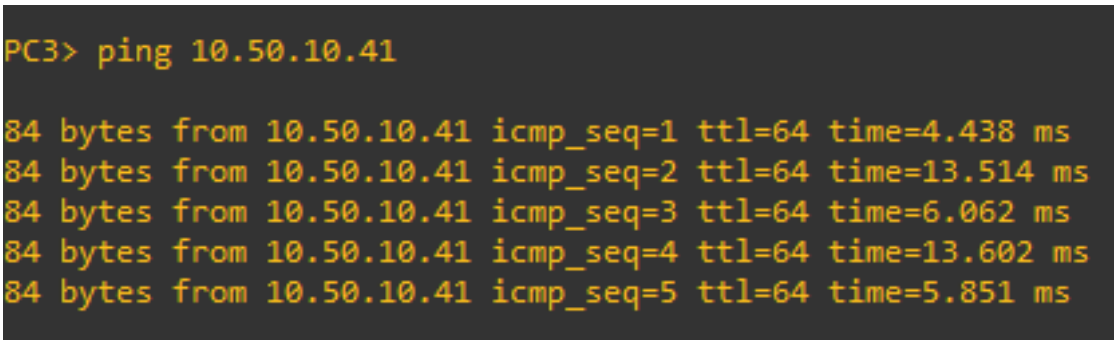
*IP PC ESTE*




*Fuente.* Elaboración propia

**Figura 40**

*ICMP PC1-PC2 ESTE*




*Fuente.* Elaboración propia

- Realizamos pruebas de ICMP entre Leaf1-Oeste y Leaf2-Este

**Figura 41**

*ICMP entre DCs ESTE Y OESTE*

```
leaf1oeste# ping 9.9.9.9
PING 9.9.9.9 (9.9.9.9) 56(84) bytes of data.
64 bytes from 9.9.9.9: icmp_seq=1 ttl=60 time=13.0 ms
64 bytes from 9.9.9.9: icmp_seq=2 ttl=60 time=10.7 ms
64 bytes from 9.9.9.9: icmp_seq=3 ttl=60 time=9.53 ms
64 bytes from 9.9.9.9: icmp_seq=4 ttl=60 time=12.7 ms
64 bytes from 9.9.9.9: icmp_seq=5 ttl=60 time=17.6 ms
64 bytes from 9.9.9.9: icmp_seq=6 ttl=60 time=7.28 ms

--- 9.9.9.9 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5006ms^C
rtt min/avg/max/mdev = 7.283/11.818/17.613/3.238 ms
leaf1oeste# ping 5.5.5.5
PING 5.5.5.5 (5.5.5.5) 56(84) bytes of data.
64 bytes from 5.5.5.5: icmp_seq=1 ttl=63 time=3.52 ms

--- 5.5.5.5 ping statistics ---^C
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.527/3.527/3.527/0.000 ms
leaf1oeste# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=61 time=7.47 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=61 time=4.49 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=61 time=9.66 ms

--- 8.8.8.8 ping statistics ---^C
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 4.495/7.210/9.663/2.119 ms
leaf1oeste# ping 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data.
64 bytes from 10.10.10.10: icmp_seq=1 ttl=60 time=6.29 ms
64 bytes from 10.10.10.10: icmp_seq=2 ttl=60 time=5.99 ms
64 bytes from 10.10.10.10: icmp_seq=3 ttl=60 time=5.68 ms
64 bytes from 10.10.10.10: icmp_seq=4 ttl=60 time=5.17 ms
64 bytes from 10.10.10.10: icmp_seq=5 ttl=60 time=5.87 ms
```

*Fuente.* Elaboración propia

```
cumulus@leaf2este:~$ ping 3.3.3.3
PING 3.3.3.3 (3.3.3.3) 56(84) bytes of data.
64 bytes from 3.3.3.3: icmp_seq=1 ttl=60 time=10.6 ms
64 bytes from 3.3.3.3: icmp_seq=2 ttl=60 time=7.16 ms
64 bytes from 3.3.3.3: icmp_seq=3 ttl=60 time=6.67 ms
64 bytes from 3.3.3.3: icmp_seq=4 ttl=60 time=5.68 ms
^C
--- 3.3.3.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 5.681/7.531/10.600/1.851 ms
cumulus@leaf2este:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=61 time=7.14 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=61 time=5.40 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=61 time=4.35 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=61 time=5.49 ms
^X64 bytes from 1.1.1.1: icmp_seq=5 ttl=61 time=9.18 ms
^C
--- 1.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 4.353/6.315/9.189/1.693 ms
cumulus@leaf2este:~$
```

*Fuente.* Elaboración propia

- Realizamos pruebas de ICMP entre PC1-Oeste y PC2-Este

**Figura 42**

*ICMP PC1- OESTE Y PC2 - ESTE*

```
PC1-OESTE>
PC1-OESTE>
PC1-OESTE> ping 10.50.10.41

84 bytes from 10.50.10.41 icmp_seq=1 ttl=64 time=9.318 ms
84 bytes from 10.50.10.41 icmp_seq=2 ttl=64 time=8.472 ms
84 bytes from 10.50.10.41 icmp_seq=3 ttl=64 time=15.178 ms
84 bytes from 10.50.10.41 icmp_seq=4 ttl=64 time=15.219 ms
84 bytes from 10.50.10.41 icmp_seq=5 ttl=64 time=19.590 ms
```

```
PC3> ping 10.50.10.11

84 bytes from 10.50.10.11 icmp_seq=1 ttl=64 time=7.634 ms
84 bytes from 10.50.10.11 icmp_seq=2 ttl=64 time=8.329 ms
84 bytes from 10.50.10.11 icmp_seq=3 ttl=64 time=7.979 ms
84 bytes from 10.50.10.11 icmp_seq=4 ttl=64 time=18.888 ms
84 bytes from 10.50.10.11 icmp_seq=5 ttl=64 time=8.248 ms

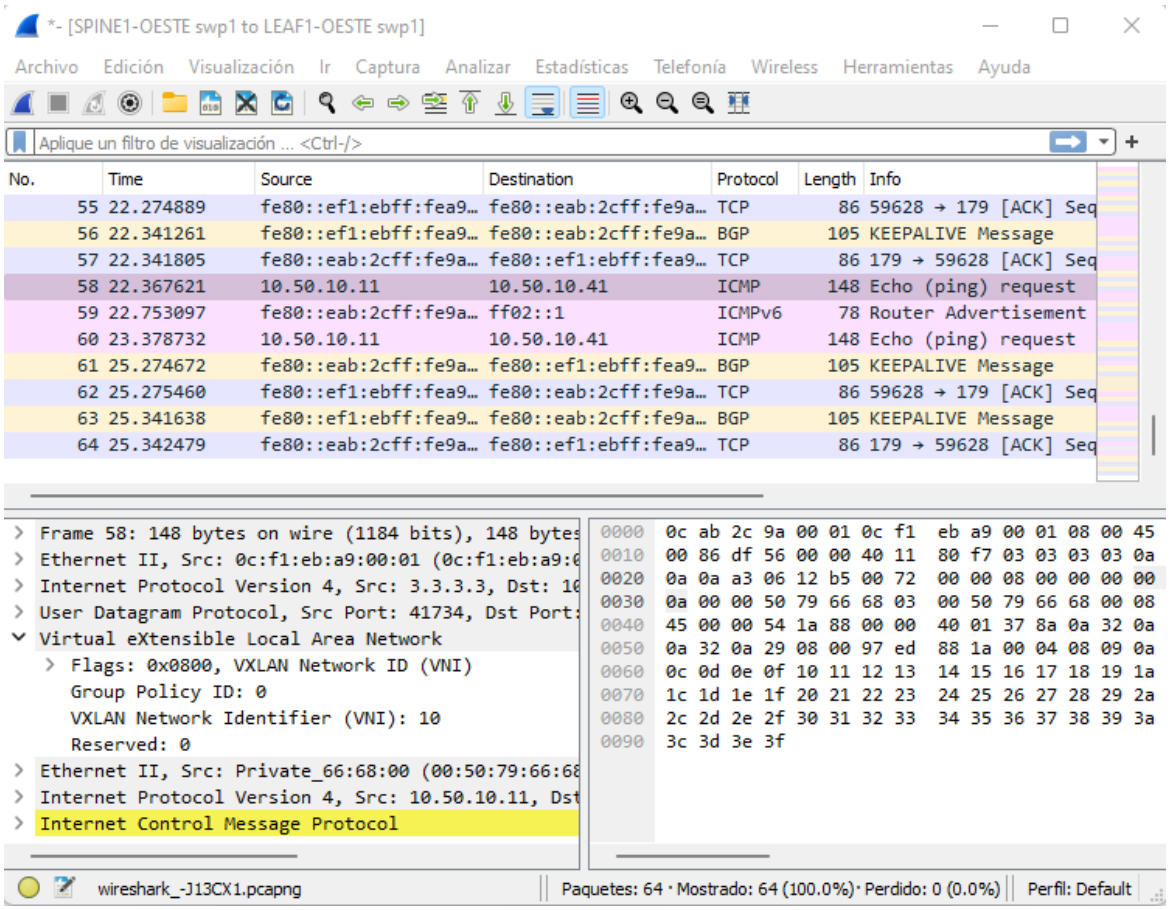
PC3>
```

*Fuente.* Elaboración propia

- Realizamos pruebas de ICMP con wireshark para validar el protocolo VXLAN y el ID

**Figura 43**

*VXLAN (VNI) 10*



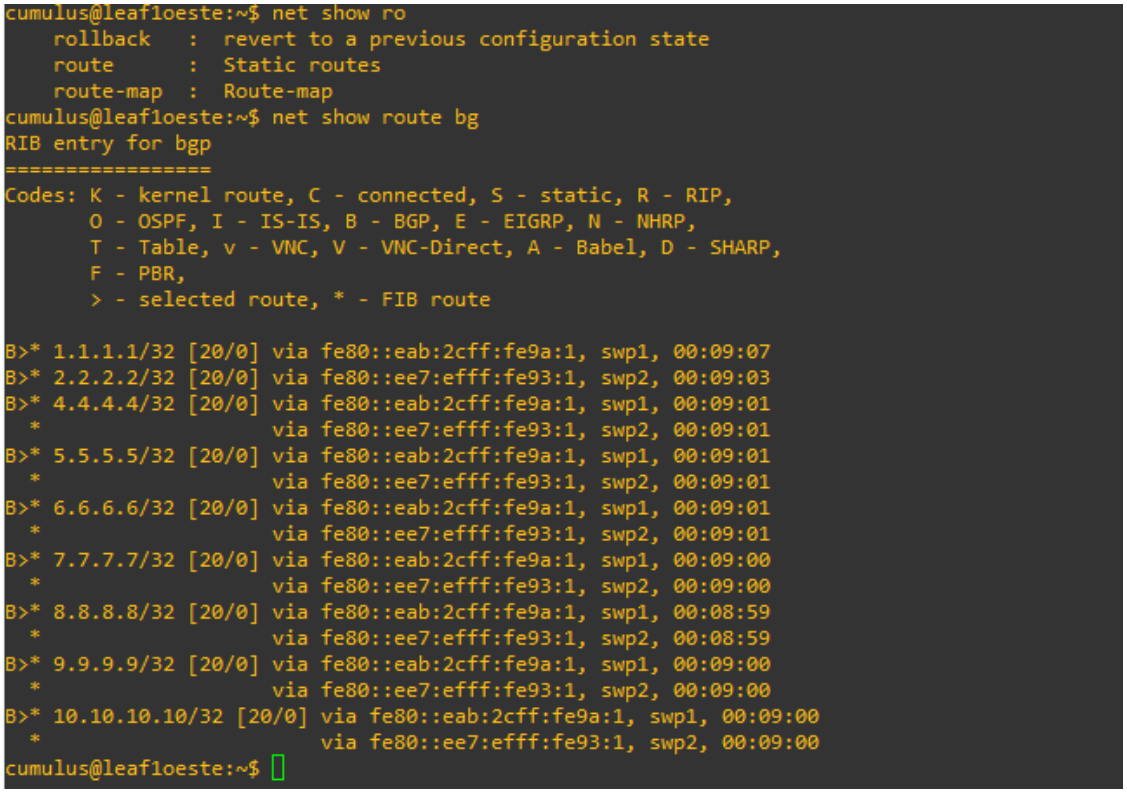

*Fuente.* Elaboración propia

Podemos observar en la prueba de ICMP entre el pc 10.50.10.11 a la 10.50.10.41, el cual se envía desde la loopback 3.3.3.3 a la 10.10.10.10 de un DC a otro, se puede observar el encabezado de VXLAN el cual lleva consigo el identificador de red VNI: 10.

- Rutas BGP hacia todos los nodos Leaf

**Figura 44**

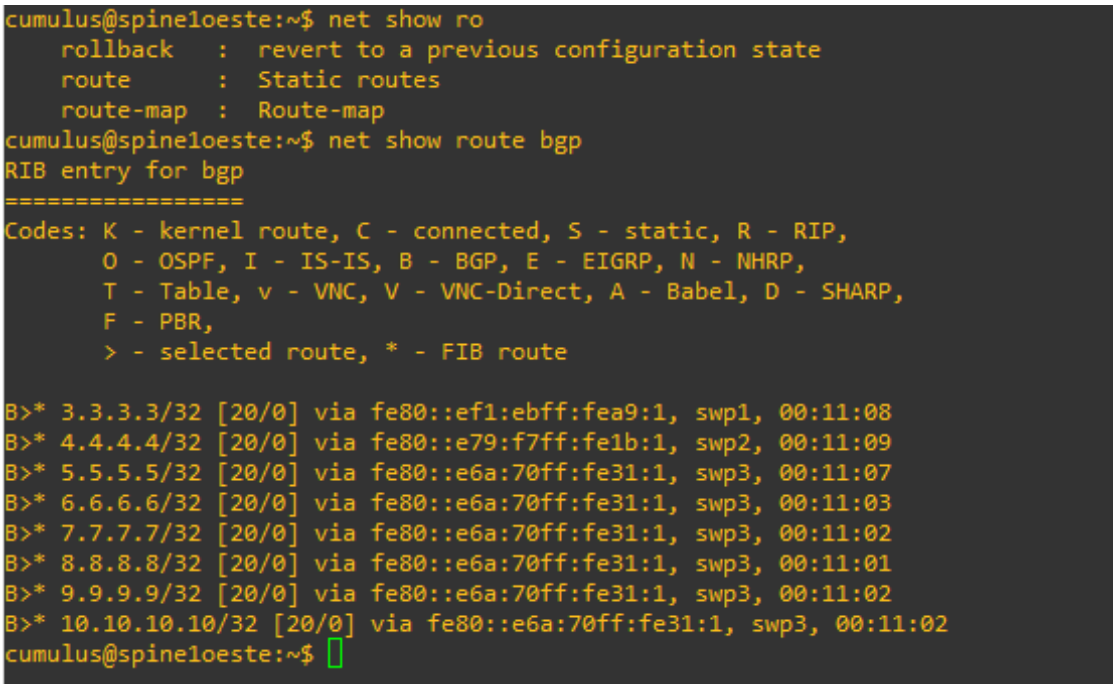
*Rutas BGP Leaf*




*Fuente.* Elaboración propia

**Figura 45**

*Ruta BGP OESTE*



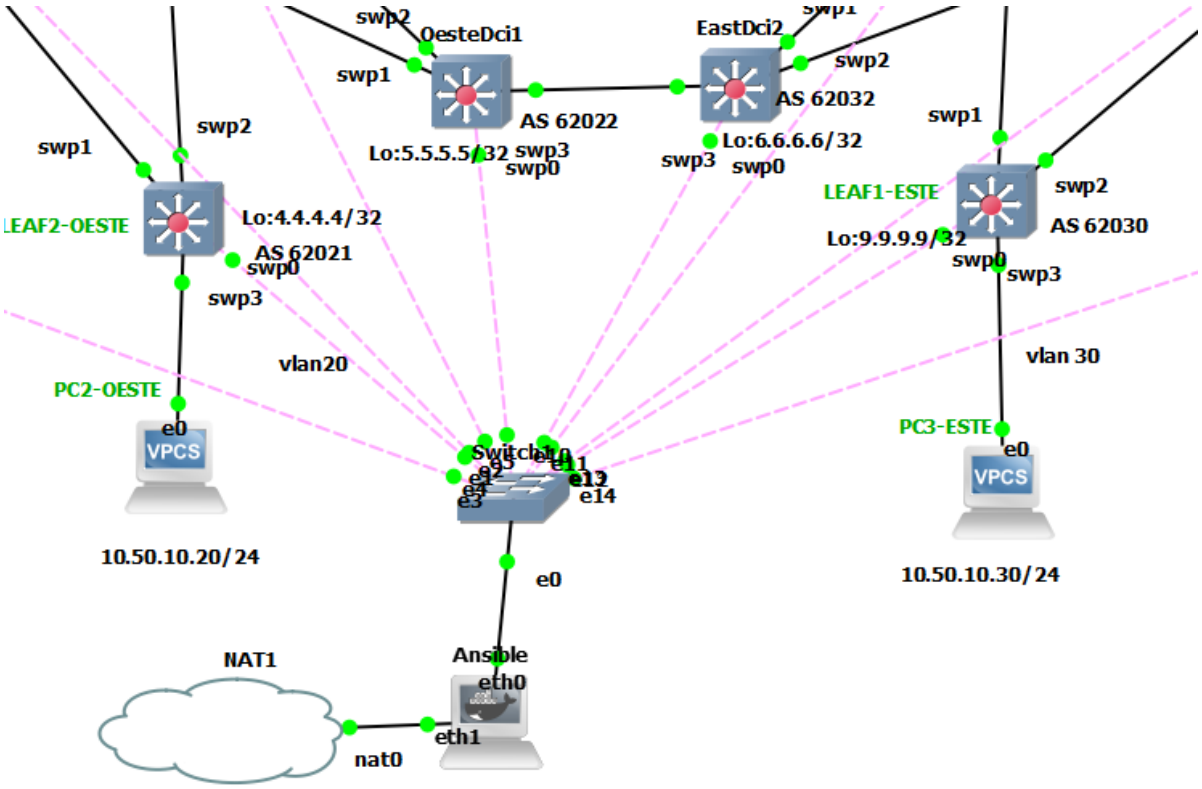

*Fuente.* Elaboración propia

- Optimización de red con Ansible

  Configuramos las IPs hacia todos los nodos en los cuales se conecta el Ansible y el SW.

**Figura 46**

*Conexión SW hacia todos los nodos*

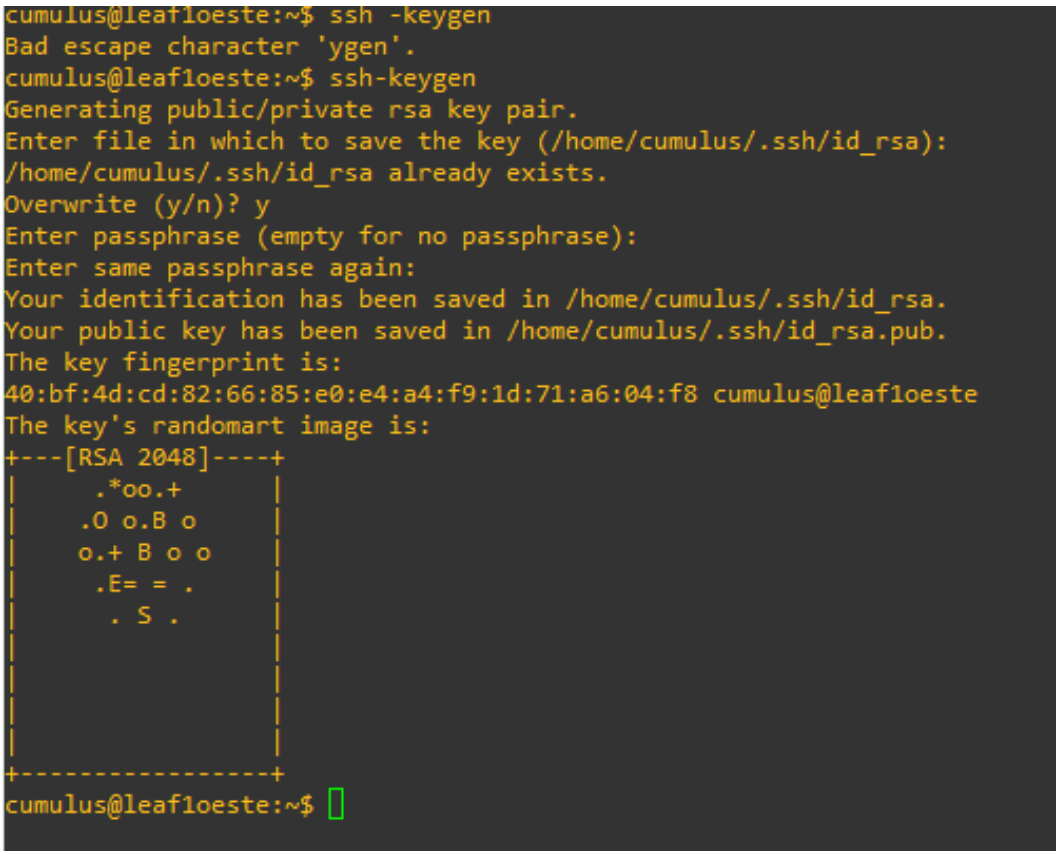



*Fuente.* Elaboración propia

- Generamos las claves SSH en todos los nodos de la red

**Figura 47**

*Clave SSH*

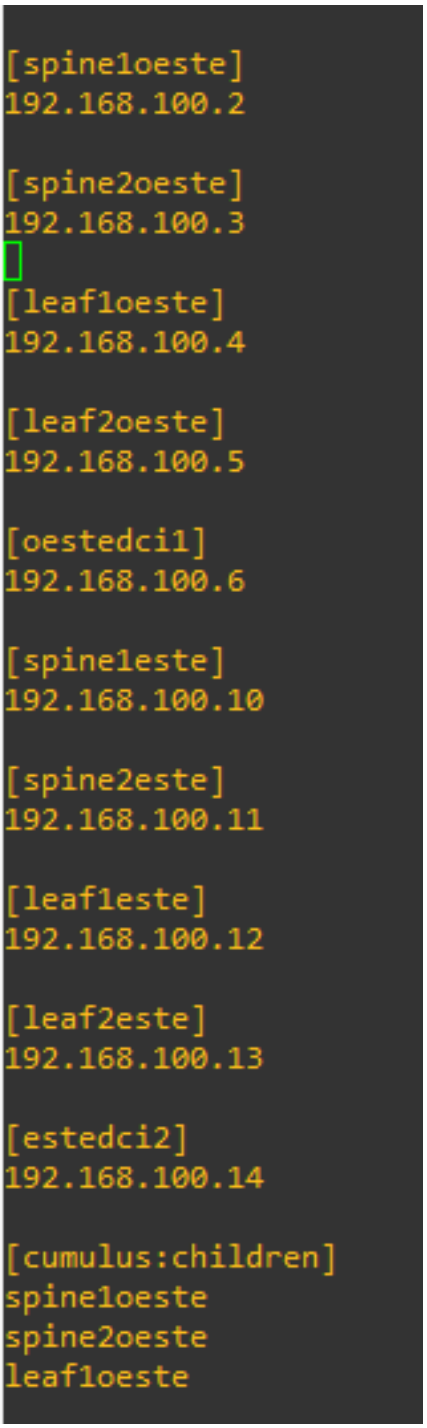



*Fuente*. Elaboración propia

- Creación de los hosts dentro de ansible, aquí se incluye todos los nodos que integraremos en el Ansible

**Figura 48**

*Hosts en Asible*

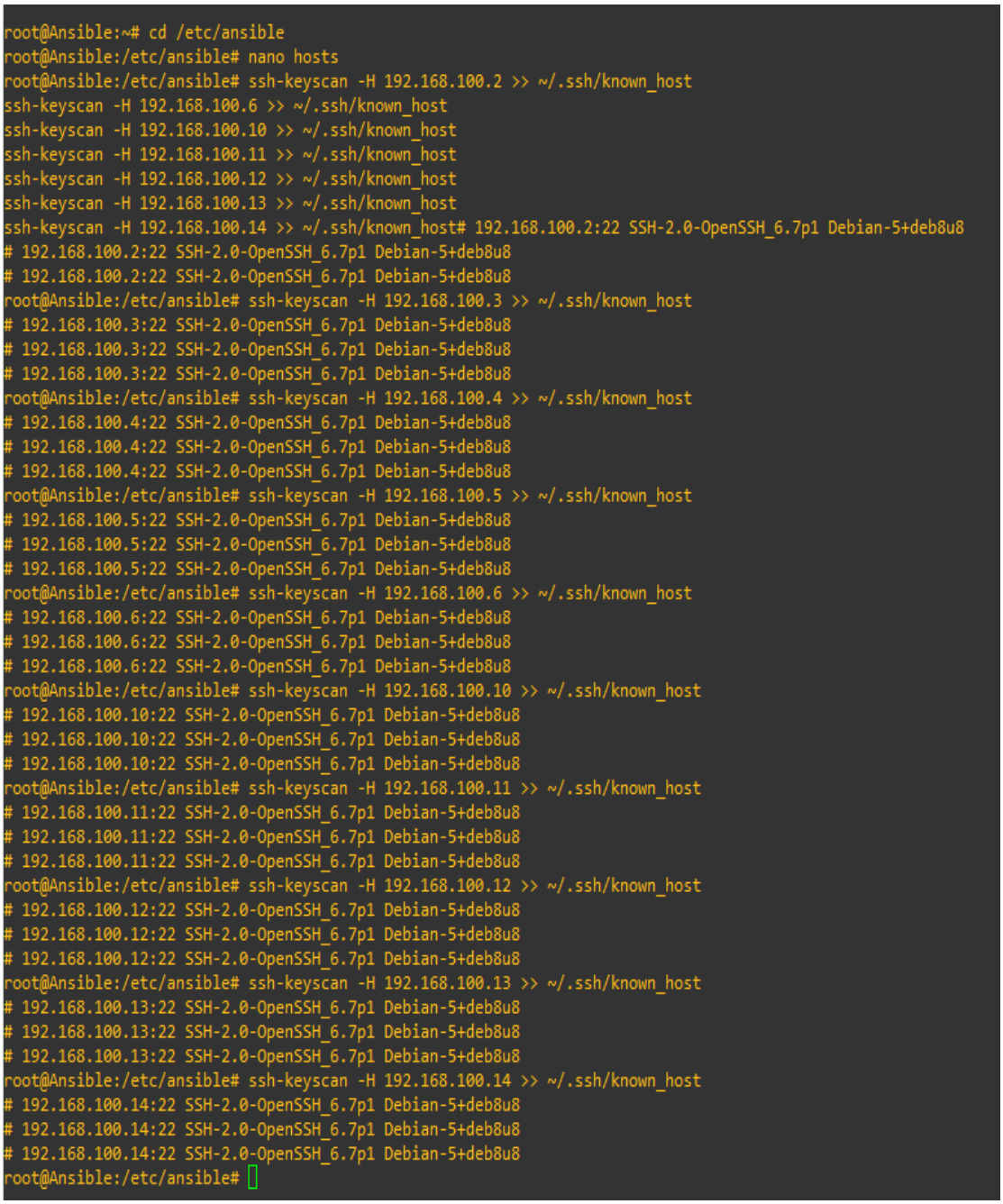



*Fuente.* Elaboración propia

- Generamos las llaves SSH en Ansible para enviar las configuraciones a cada equipo usa SSH, por lo que, para facilita el proceso de enlace, se generaron llaves en cada nodo Cumulus

**Figura 49**

*Llaves SSH Ansible*

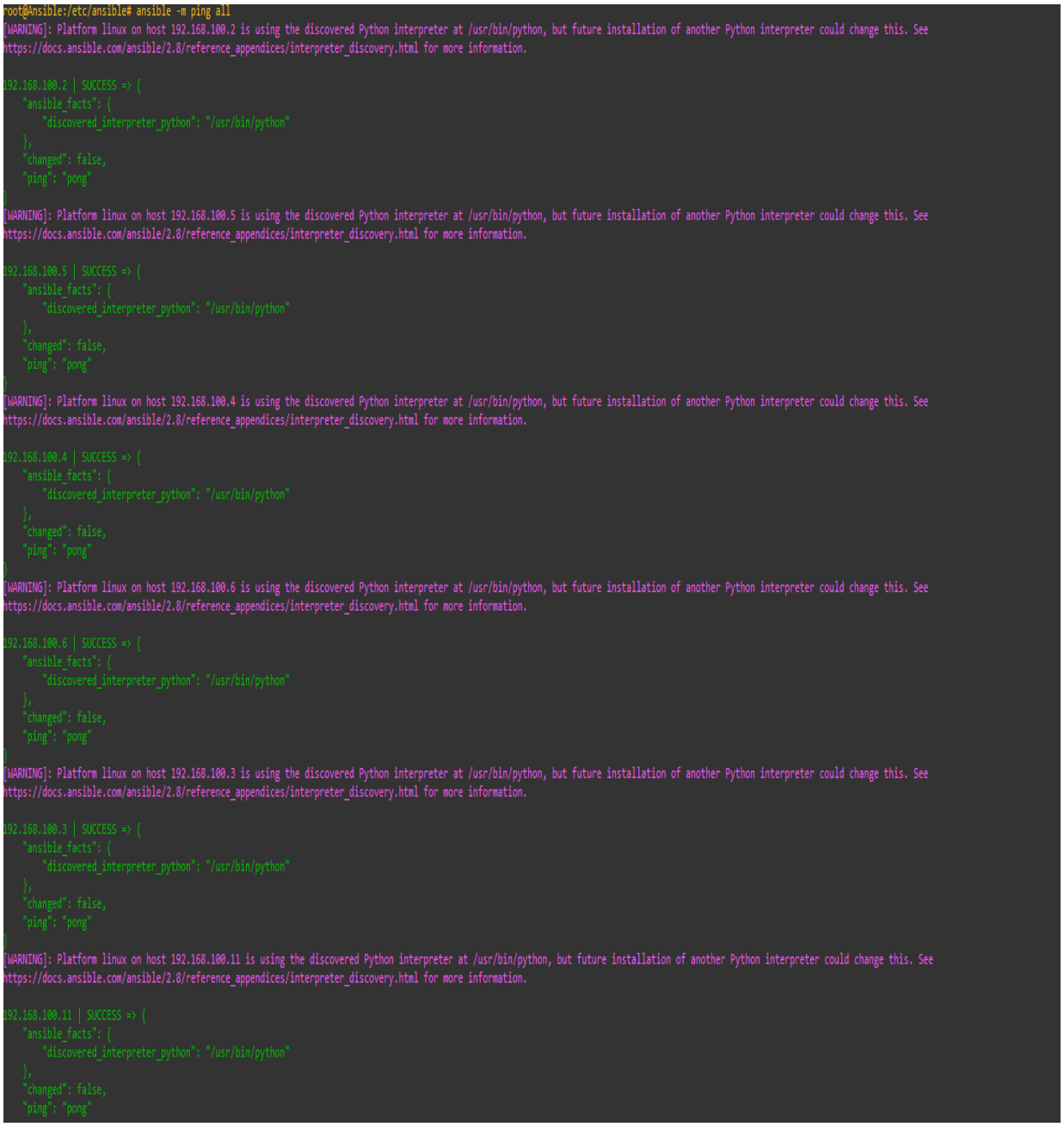



*Fuente*. Elaboración propia

- Realizamos un ping all desde ansible hacia todos los hosts para verificar la conexión exitosa

**Figura 50**

*Conexión hacia todos los Nodos*

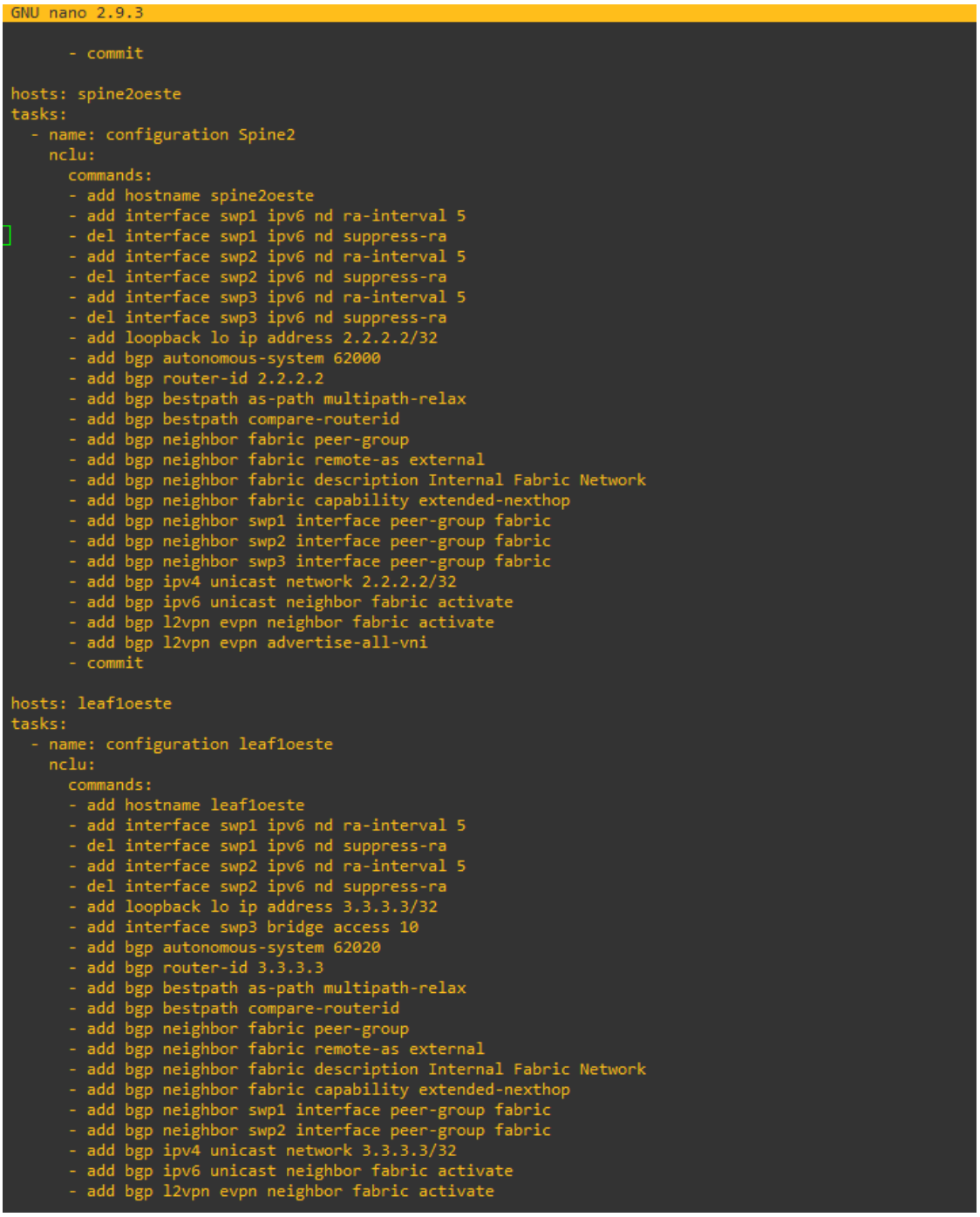



*Fuente*. Elaboración propia

- Creación de nuestro Playabook VXLAN.yml el cual será nuestro archivo que contendrá todas las configuraciones necesarias que desplegaremos hacia los nodos de la red

**Figura 51**

*Playbook de configuración*

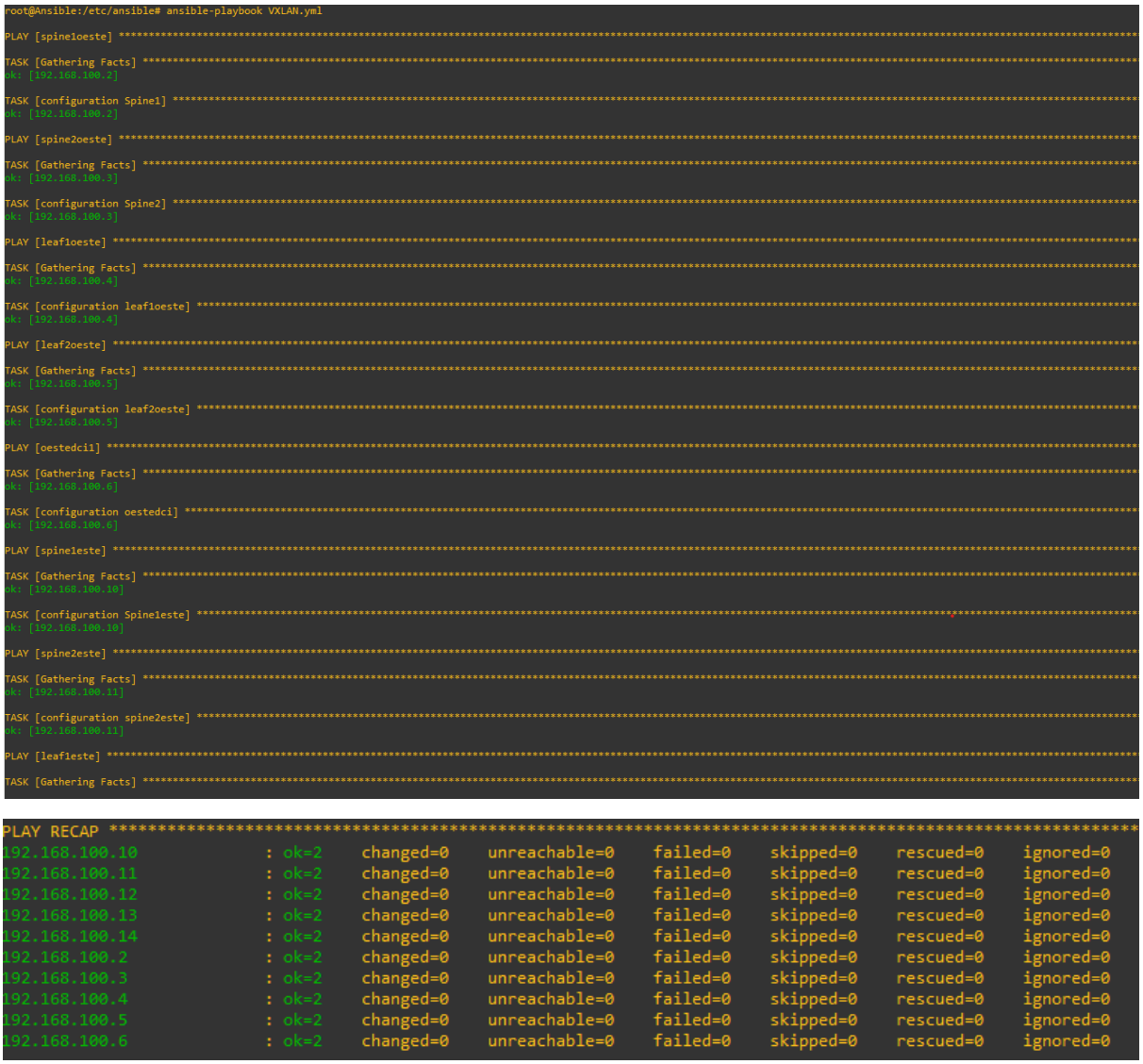



*Fuente.* Elaboración propia

- Ejecutamos el playbook y verificamos que se despliegue la configuración hacia los nodos exitosamente.

**Figura 52**

*Resultado del despliegue de configuración Playbook Ansible*







*Fuente.* Elaboración propia

## CONCLUSIONES

- Se ha concluido que es posible emular una infraestructura spine and leaf utilizando la herramienta de virtualización GNS3, con esto antes de implementar en un ambiente de producción testearemos que funcione correctamente en un ambiente virtualizado
- Se ha concluido que con el protocolo de VXLAN suplimos la necesidad del agotamiento de VLANS de 4049 a más de 16 millones esto dará la apertura a los proveedores de servicios de alojar un mayor número de inquilinos
- Una vez realizadas las pruebas de concepto se llegó a la conclusión de que VXLAN permite una gran escalabilidad además con las mejoras se minimiza él envió incensario de tráfico broadcast logrando una mejor conexión en la red
- Con la prueba de concepto se pudo establecer la conectividad entre los equipos finales separados remotamente empleando la misma VXLAN, esto nos permite utilizar direccionamiento privado para interconectar cliente y optimizar el uso de direccionamiento publico IPV4
- Se concluye que VXLAN permite simplificar la red, haciendo innecesario el protocolo Spanning Tree, el trunking y las stretching VLANS, las máquinas virtuales puedes desplazarse entre hosts en diferentes VLANS sin necesidad de afectarlas cambiando sus direcciones IPs
- El protocolo EVPN se utiliza como el plano de control de superposición y proporciona la conectividad virtual entre los dominios de capa 2 y capa 3 a través de la red
- Con la prueba de concepto validamos que la latencia entre los dispositivos de red es más estable, cuando existe la caída de un nodo spine o leaf no tiene un impacto tan grande en el rendimiento de la estructura general

- Con la evolución de la tecnología y el beneficio que otorga el multi-tenant el requisito en la red cambio por completo antes muchas empresas disponían de un solo DC ahora disponen de múltiples DCs en diferentes ubicaciones para lograr una alta disponibilidad y la continuidad del negocio

- A medida que la empresa se expande por todo el mundo surgían problemas debido a la distancia físicas entre los centros de datos, la red requería soportar la movilidad de aplicaciones de un data center a otro para la continuidad del negocio con el protocolo EVPN se reduce la carga en el plano de datos y de control mejora la implementación de funciones del aprendizaje de mac con esto mejora la eficiencia operativa, a la vez de otorgar la alta disponibilidad

- Con la adopción de la nube ha hecho que los proveedores de servicios de Dcs reconsideren su arquitectura de red para suplir esa limitación de ancho de banda y latencia las redes requieren admitir múltiples rutas para que pueda optimizarse el flujo de tráfico en la red, el ancho de banda y latencia, para esto se requiere la agregación más rápida de los enlaces disponibles mediante el uso de una capa subyacente basada en IP de capa 3 junto a la superposición VXLAN-EVPN

- Con Cumulus Linux no te ata a una marca o a un proveedor de servicio, con su software libre no solo te permite obtener un menor costo de implementación en la red y un mayor ROI

- Cumulus Linux dispone de mejores herramientas de automatización, son compatible con todos los protocolos de comunicación de la red, te permite personalizar y escalar la red sin problemas

- El open networking como Ansible es que es una herramienta de automatización de la red es intuitiva y fácil de aprender, ya no es necesario ingresar de equipo en equipo si

tienes una red con más de 100 o 1000 dispositivos desde Ansible puedes enviar la configuración o cambios a los nodos requeridos con sus ficheros YAML a través de los llamados Playbooks

- Con Ansible se logra una eficiencia en recursos y tareas evitando que los ingenieros malgasten su tiempo muchas veces realizando un doble trabajo, a la vez ansible utiliza SSH para la conexión a los nodos el cual minimiza los riesgos de seguridad

## RECOMENDACIONES

- Es importante ir transformándose digitalmente y a medida que la tecnología crece necesitamos adecuarnos y preparar un centro de datos moderno que pueda suplir las necesidades del negocio, implementando nuevos protocolos que nos permitirá optimizar la red como lo es VXLAN/EVPN

- Generalmente se ocupa demasiado OPEX en smartnets garantías para los equipos de marcas tradicionales, el open box con su SO de Cumulus Linux no solo mejorará la red, sino que reducirá el OPEX y a la larga genera un mayor retorno de inversión

- Al utilizar Ansible para optimizar la red es necesario como buena práctica tomar backups de la configuración de los equipos, ya que el cambio si es masivo puede generar algún problema si no está bien el Playbook y provocar perdidas de servicio.

- Antes de realizar una configuración es importante ejecutar primero en un ambiente controlado como lo es GNS3 que nos permite emular nuestra infraestructura, si todo se ejecuta de manera correcta podemos implementarlo en nuestra infraestructura física y pasarlo a producción.

- Documentar como se encuentra la red es un trabajo que se debe realizar en todas las empresas si un ingeniero conoce toda su infraestructura y la documenta va a ser mucho más sencillo realizar cambios sea a nivel físico como lógico.

- Es importante seguir capacitándose día a día ya que las tecnologías avanzan de una manera exponencial y por esto tenemos que estar preparados para enfrentar las demandas de la transformación digital

## REFERENCIAS BIBLIGRÁFICAS


Buresh, B., Eline, D., Jansen, D., Gmitter, J., Ostermiller, J., Moreno, J., . . . Kondalam, S. (23 de Abril de 2016). Obtenido de A Modern, Open and Scalable Fabric VXLAN EVPN: https://www.cisco.com/c/dam/en/us/td/docs/switches/datacenter/nexus9000/sw/vxlan_evpn/VXLAN_EVPN.pdf

Chatani Chatani, A. (8 de Junio de 2020). *Redes Definidas por Software en Centros de Datos.* Obtenido de https://riull.ull.es/xmlui/bitstream/handle/915/19776/Redes%20definidas%20por%20software%20en%20centros%20de%20datos.pdf?sequence=1

Ch, G. D. S., Naranjo, E. F., & Marrone, L. (2018, November). SDN-Ready WAN networks: Segment Routing in MPLS-Based Environments. In 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 173-178). IEEE.

Cisco. (2006). *CCNA* . Obtenido de Diseño de Red en 3 capas: https://ccnadesdecero.es/diseno-jerarquico-de-redes/

Cisco. (2019). *The Anatomy of Net DevOps.* Obtenido de What is NetDevOps: https://blogs.cisco.com/developer/anatomy of netdevops

Cisco. (s.f.). *Cisco Nexus 700 Series Switches Build a next-generation network*. Recuperado el 18 de noviembre, 2022. Obtenido de https://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-

DELLEMEC. (2017). *The Economic Advantages of Open Networking and Customer Choice.* Obtenido de Dell's market changing Open Networking initiative: https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/the-economic-advantages-of-open-networking.pdf

GNS3. (2021). Obtenido de Getting Started with GNS3. Recuperado el 4 de noviembre, 2022 de: https://docs.gns3.com/docs/

Hewlett Packard Enterprise. (2018). *HPE FlexFabric 5710 Switch Series VXLAN Configuration.* Obtenido de VXLAN packet format: https://techhub.hpe.com/eginfolib/networking/docs/switches/5710/5200-5004_VXLAN_cg/content/517705090.htm

HPE Aruba Networking. (2022). *What is Spine-Leaf Architecture* . Obtenido de Spine-Leaf Architecture Explained: https://www.arubanetworks.com/faq/what-is-spine-leaf-architecture/

HUAWEI. (9 de Diciembre de 2020). *S12700 V200R011C10 Configuration Guide - VXLAN*. Obtenido de VXLAN Network Architecture: https://info.support.huawei.com/info-finder/encyclopedia/en/EVPN.html

HUAWEI. (19 de Octubre de 2021). *IP Encyclopedia*. Obtenido de What is EVPN: https://info.support.huawei.com/info-finder/encyclopedia/en/EVPN.html

Krattiger, L., Kapadia, S., & Jansen, D. (Abril de 2017). *Building Data Centers with VXLAN BGP EVPN.* Obtenido de A Cisco NX-OS Pespective: https://dl.hellodigi.ir/dl.hellodigi.ir/dl/book/Building%20Data%20Centers%20with%20VXLAN%20BGP%20EVPN%20A%20Cisco%20NX-OS%20Perspective.pdf

Naranjo, E. F., & Ch, G. D. S. (2017, October). Underlay and overlay networks: The approach to solve addressing and segmentation problems in the new networking era: VXLAN encapsulation with Cisco and open source networks. In 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM) (pp. 1-6). IEEE.

NVIDIA Networking. (2022). *NVIDIA Cumulus Linux*. Obtenido de Cómo funciona Cumulus

Linux : https://www.nvidia.com/es-la/networking/ethernet-switching/cumulus-linux/

Red Hat. (21 de Junio de 2022). *Ansible: conceptos básicos.* Obtenido de Introducción Ansible: https://www.redhat.com/es/topics/automation/learning-ansible-tutorial

Salazar Chacón, G. (2019). El Networking, IT y la Transformación Digital [Diapositivas de PowerPoint]. Facultad de Ingeniería, Pontificai Universidad Católica del Ecuador

Salazar-Chacón, G., Naranjo, E., & Marrone, L. (2020). Open networking programmability for VXLAN Data Centre infrastructures: Ansible and Cumulus Linux feasibility study. Revista Iberica de Sistemas e Tecnologias de Informacao, (E32), 469-482.

Salazar-Chacón, G. D. (2020, October). VXLAN-IPSec Dual-Overlay as a Security Technique in Virtualized Datacenter Environments. In 2020 IEEE ANDESCON (pp. 1-6). IEEE.

Salazar-Chacón, G. D., & Marrone, L. (2020, November). OpenSDN Southbound Traffic Characterization: Proof-of-Concept Virtualized SDN-Infrastructure. In 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 0282-0287). IEEE.

Salazar-Chacón, G. D., & García, A. R. R. (2021, April). Segment-Routing Analysis: Proof-of-Concept Emulation in IPv4 and IPv6 Service Provider Infrastructures. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-7). IEEE.

Salazar Chacón, G. D. (2021). Hybrid Networking SDN y SD-WAN: Interoperabilidad de arquitecturas de redes tradicionales y redes definidas por software en la era de la digitalización (Doctoral dissertation, Universidad Nacional de La Plata).

Salazar Chacón, G. (2022, a). Las Redes WAN y su Impacto Empresarial [Diapositivas de PowerPoint]. Facultad de Ingeniería, Pontificai Universidad Católica del Ecuador

Salazar Chacón, G. (2022, b). Paradigma Underlay - Overlay [Diapositivas de PowerPoint]. Facultad de Ingeniería, Pontificai Universidad Católica del Ecuador

Scazzariello, M., Ariemma, L., Di Battista, G., & Patrignani, M. (2021). Megalos: A Scalable Architecture for the Virtualization of Network Scenarios. *Future Internet*, 7-8. Obtenido de VXLAN Packet Encapsulation.: https://www.researchgate.net/figure/VXLAN-Packet-Encapsulation_fig1_354232739

TechLibrary. (8 de Enero de 2023). *Guía del usuario del BGP*. Obtenido de Descripción general del BGP: https://www.juniper.net/documentation/mx/es/software/junos/bgp/topics/topic-map/bgp-overview.html

TechTarget Networking. (Junio de 2022). *VLAN (virtual LAN)*. Obtenido de What is a VLAN (virtual LAN): https://www.techtarget.com/searchnetworking/definition/virtual-LAN

Tigre Cortes, J. J., & Mejía, M. (Noviembre de 2012). *Universidad Tecnológica Israel* . Obtenido de Desarrollar el Diseño de Redes virtuales Locales (VLAN) para aislar el tráfico de broadcast: http://repositorio.uisrael.edu.ec/bitstream/47000/339/1/UISRAEL-EC-SIS-378.242-3.pdf