ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE SERVICIOS DE RED CON HERREMIENTAS DEVOPS

CREACIÓN DE DMZ EN EQUIPOS ESPECIALIZADOS DE RED MEDIANTE DEVOPS

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

LUIS ALFONSO PADILLA CAMALLE

DIRECTOR: FERNANDO VINICIO BECERRA CAMACHO

DMQ, ABRIL 2023

CERTIFICACIONES

Yo, Luis Alfonso Padilla Camalle 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.

Luis Alfonso Padilla Camalle

luis.padilla@epn.edu.ec

luis.padilla.cam@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Luis Alfonso Padilla Camalle, bajo mi supervisión.

ING. Fernando Vinicio Becerra 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.

LUIS ALFONSO PADILLA CAMALLE

DEDICATORIA

El presente trabajo de titulación lo dedicó principalmente a mis queridos padres, Rosa Camalle y Luis Padilla Hidalgo, por ser los inspiradores y darme las fuerzas necesarias para cumplir esta meta, por el gran apoyo durante el proceso de la carrera, sus consejos, valores, amor y sacrificios realizados de estos años. Estoy feliz y orgulloso de ser su hijo, son los mejores padres.

A mis queridos hermanos, Manuel Padilla y Mario Padilla, por enseñarme a vivir, trabajar, ser responsable y pensar antes de actuar. También por el apoyo de un lugar donde quedarme para continuar estudiando, los consejos y estar junto a mi lado durante todo este proceso académico.

A mis profesores de la carrera que durante estos años me brindaron sus conocimientos, consejos, sus vivencias, anécdotas y experiencias.

Luis A. Padilla C.

AGRADECIMIENTO

En primer lugar, agradezco a mis padres por todo el apoyo brindado durante toda mi vida. Por sus consejos, enseñanzas y conocimientos que me han servido para convertirme en la persona que soy ahora.

Agradezco a mis hermanos por la paciencia, apoyo incondicional y la ayuda brindada durante todo este proceso para llegar a cumplir una meta más de mi vida.

Agradezco a mi querido sobrino, Jhonatan Salazar, por ser la base inspiradora para seguir estudiando y demostrarle que con esfuerzo y paciencia se puede cumplir nuestras metas.

En espacial al Ing. Fernando Becerra por la oportunidad de realizar este trabajo de titulación. También le agradezco por motivación, compresión y dedicación puesta en el proceso de desarrollo del proyecto, acompañado de recomendaciones, sonrisas y ayuda en la creación del presente documento.

Agradezco a mis amigos por los consejos y ayuda brindada en momento críticos de la carrera.

Luis A. Padilla C.

ÍNDICE DE CONTENIDOS

RESUMEN

Hoy en día la automatización de procesos continúa desarrollándose, con la finalidad de implementar nuevos servicios en el menor tiempo posible, solventando problemas del error humano y un mejoramiento continuo de los sistemas. La automatización de redes está involucrada en estos desarrollos, donde Ansible es utilizado como una herramienta compatible en la automatización de dispositivos *networking*. Ansible logra generar tareas de configuración para los equipos de red, a través de un servidor principal o nodo controlador, que incluye Python, Ansible y SSH.

La comunicación establecida con el protocolo SSH entre los dispositivos *networking* y el servidor permite ejecutar un archivo de formato YAML denominado como *playbook*. El *playbook* contiene las tareas de configuración que serán implementadas en los equipos comunicados con SSH.

El primer capítulo presenta una breve descripción del componente desarrollado, objetivos, alcance y marco teórico, cuyos puntos permiten el desarrollo de este proyecto de titulación.

El segundo capítulo presenta la metodología utilizada en el desarrollo del proyecto, donde se detalla los procedimientos aplicados en cada objetivo, hasta obtener un algoritmo que use Ansible para la configuración de una DMZ en un dispositivo Cisco ASA.

El tercer capítulo muestra los resultados alcanzados, en el que se describe el desarrollo de los objetivos planteados, las configuraciones implementadas y las pruebas de funcionamiento del algoritmo creado con Ansible.

El cuarto capítulo indica las conclusiones obtenidas al desarrollar este proyecto de titulación.

Finalmente, el quito capítulo expone las recomendaciones sobre las complicaciones y soluciones que se presentaron durante el desarrollo del proyecto.

PALABRAS CLAVE: Ansible, DevOps, SSH, Cisco ASA, DMZ y ACL

ABSTRACT

Today, process automation is continuing to develop, with the purpose to implement new services in the shortest possible time, solving human error problems and continuous improvement of systems. Network automation is involved in these developments, where Ansible is used as a compatible tool of networking devices management. Ansible manages to generate configuration tasks for network equipment, through a main server or controller node, which includes Python, Ansible and SSH.

The communication established with the SSH protocol between the networking devices and the server allows executing a file in YAML format called playbook. The playbook has the configuration tasks that will be implemented in the equipment communicated with SSH.

The first chapter presents a brief description of the developed components, objectives, scope and theoretical framework, whose points allow the development of this titling project.

The second chapter presents the methodology used in the project development, where the procedures applied in each objective, until I get an algorithm that uses Ansible form setting up a DMZ on a Cisco ASA device.

The third chapter shows the results achieved, in which it's described the development of the proposed objectives, implemented configurations and the performance tests of algorithm created by Ansible.

The fourth chapter indicates the conclusions obtained when developing this titling project.

Finally, the fifth chapter presents the recommendations about the complications and solutions that were presented during the project development.

KEYWORDS: Ansible, DevOps, SSH, Cisco ASA, DMZ y ACL

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La configuración de los dispositivos de red resulta un proceso repetitivamente trabajoso, desperdiciando tiempo y recursos, incluso las configuraciones realizadas pueden ser erróneas por parte de personas que están comenzando en este campo. Por este motivo la automatización de estos procesos permitirá solventar estos inconvenientes e implementar una red funcional con sus respectivas características.

El presente trabajo de integración curricular está enfocado en crear un algoritmo con Ansible que permita configurar una DMZ en equipos de red Cisco ASA. El algoritmo permite automatizar este proceso repetitivo con los componentes necesarios para su funcionamiento, evitando errores humanos y gran esfuerzo. A su vez se garantiza eficiencia y eficacia al momento de utilizar Ansible en la configuración de equipos de *networking*.

1.1 Objetivo general

Implementar servicios de red mediante herramientas de DevOps

1.2 Objetivos específicos.

- Analizar DevOps y sus características.
- Diseñar el algoritmo para poder implementar DMZ.
- Implementar una DMZ mediante herramientas de DevOps.
- Verificar el funcionamiento del algoritmo.

1.3 Alcance

El presente proyecto permite a los estudiantes el manejo de las herramientas de DevOps y aplicarlo a los equipos de *networking*, para ello se va a necesitar virtualizar los elementos de red y poder simularlos ya que no se tiene equipos físicos. En este proyecto de titulación se van a implementar servicios de red como son: VRF, *multicast*, políticas de seguridad y DMZ y finalmente servicios web y DNS mediante herramientas de DevOps para automatizar su despliegue. Además, este proyecto es macro el cual contiene un total de cuatro proyectos de titulación los cuales tienen su propia línea de despliegue.

1.4 Marco Teórico

DevOps

El término DevOps proveniente de *Development and Operations,* está enfocado a la idea de producir productos de manera rápida, incluyendo un soporte frente a cambios en el mercado. En DevOps se incluye la automatización de procesos y levantamiento de servicios [1] [2] [3].

Algunas de las ventajas de DevOps se exponen a continuación:

- **Metodología ágil**: Con la unión de la organización y priorización de objetivos se alcanza resultados rápidos y generación de nuevas aplicaciones [4]. A esto se agrega la reducción de costos mejorando la productividad por parte de los colaboradores y empleados.
- **Colaboración y responsabilidad**: Existe una unión entre los distintos departamentos y equipos con el fin de alcanzar los objetivos de la empresa. Un ejemplo es la unión de los departamentos de desarrollo y operaciones [2].
- **Procesos automatizados**: Generación de producción continua y rápida de aplicaciones o productos [3].
- **El cliente como foco principal**: El principal objetivo de la empresa es elaborar un producto para satisfacer las necesidades del cliente [3]. Con DevOps se puede solventar dicha necesidad para una mejora en la entrega de productos, ya que se demora menos.

Equipos de Cisco ASA

En una red de empresas, organizaciones, universidades o similares existen ataques eventuales por parte de los denominados *hackers*, que buscan robar información u ocasionar pérdidas para las entidades mencionadas. Frente a este problema se considera el uso de cortafuegos (*firewalls*) o dispositivos de seguridad, donde se prioriza la protección y seguridad de la red. Estos equipos revisan, aceptan o deniegan el tráfico de acuerdo a las configuraciones de seguridad establecidas [5].

Los equipos Cisco ASA se centran en la seguridad de la red, caracterizándose por tener una variedad de configuración para brindar seguridad. En las funciones de los equipos ASA se permite la creación de *firewalls*, segmentación de la red en pequeñas zonas y VPN (Virtual Private Network). Por otro lado, no es recomendable el uso de

routers para crear firewalls porque el equipo se vuelve lento y no son diseñados para ese fin [6].

Una aplicación principal del dispositivo Cisco ASA es la creación de una DMZ (*Demilitarized Zone*), permitiendo dividir a la red en pequeñas zonas y asignando a cada una de ellas distintos niveles de seguridad [7]. Un claro ejemplo se muestra en la Figura 1.1 que contiene la formación de tres zonas denominadas como: *outside*, DMZ e *inside*.

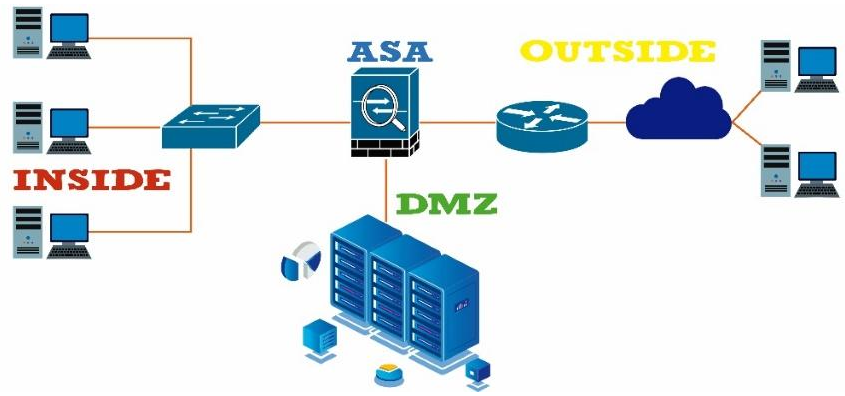

Figura 1.1 Topología DMZ

En un equipo Cisco ASA la zona *outside* es una parte de la red por donde ingresa el Internet, permitiendo a las otras zonas de la red el acceso a la extranet. Por otra parte, la zona DMZ es otra parte de la red donde se ubican los servidores, ya sean públicos o privados. La zona DMZ permite tanto a los usuarios del Internet como a los de la empresa, el acceso a los servidores con ciertas restricciones de seguridad. A diferencia de las anteriores zonas, la zona *inside* se conoce como una red privada donde una entidad tiene su red con los usuarios [8] [9].

En un dispositivo Cisco ASA se asigna un nivel de seguridad a cada zona, este nivel es un parámetro configurado en la interfaz e indica que tan confiable o insegura es la red. El nivel de seguridad puede tomar un valor entre 0 a 100, donde el valor de 0 se considera a una red insegura, por otra parte, mientras el valor sea distinto de 0 se tendrá una cierta confiabilidad, siendo el valor 100 el máximo nivel de seguridad [9] [10].

La zona privada es configurada con el máximo nivel de seguridad, por lo tanto, los usuarios de otras zonas no pueden acceder a la red privada, pero la zona *inside* puede acceder a otras zonas, todo dependiendo de las configuraciones establecidas en las zonas existentes [8].

Al tener varias zonas divididas con el dispositivo Cisco ASA se consigue una seguridad controlada. La seguridad es complementada con la configuración de NAT (*Network Address Translation*), ACL (*Access Control List*), VPN (*Virtual Private Network*), entre otros, permitiendo al Cisco ASA la revisión y control de tráfico circulante entre zonas. A su vez, su aplicación permite a las empresas o entidades evitar posibles ataques externos [10].

Conceptos fundamentales de Ansible

Ansible es una de las herramientas de automatización la cual tiene la idea de DevOps (*Development and Operations*). Ansible tiene una mayor aplicación en la configuración y actualización para la administración de servidores, equipos de red, servicios en la nube, entre otros. Para el funcionamiento de Ansible es necesario el manejo del protocolo SSH con el fin de realizar conexiones seguras entre los equipos. Ansible maneja el lenguaje Python en relación a la automatización que permiten la creación de libros de jugadas (*playbooks*) el cual está en formato YAML [11] [12]. En la Figura 1.2 se puede observar el manejo básico de Ansible.

Figura 1.2 Escenario de Ansible

Ansible es una aplicación sencilla y flexible desde el punto de vista del usuario, además la herramienta tiene un funcionamiento efectivo para automatizar procesos y los usuarios no necesitan aprender un nuevo lenguaje de programación ya que solo utiliza etiquetas [13]. Ansible en mucho de los casos tiene que utilizar módulos de su repositorio, por ejemplo, entre equipos Cisco: *routers* y ASAs se utilizan diferentes módulos, para el primer ejemplo se necesita cisco.ios, mientras el segundo el cisco.asa [14].

La automatización de equipos en Ansible se realiza mediante los *playbooks*, un archivo con formato YAML, dentro del cual se puede escribir un lenguaje de declaración de datos, ofreciendo una buena capacidad y comprensión de escritura. Es

decir que con los *playbooks* se puede tener una gran cantidad de configuraciones en pocas líneas de código, las cuales son fáciles de comprender para el usuario [13] [15].

En esencia, con Ansible se puede automatizar las redes cumpliendo el objetivo principal de simplificar las tareas involucradas en configurar, administrar y operar las topologías de red, servicios y la conectividad entre ellos [16] [17]. Ansible es apropiado para administrar diferentes ambientes, desde pequeñas configuraciones con una menor cantidad de procesos hasta entornos empresariales con miles de procesos [16]. En la Figura 1.3 se puede observar un ejemplo en donde el servidor de Ansible se comunicaría con los dispositivos de red para su automatización y administración.

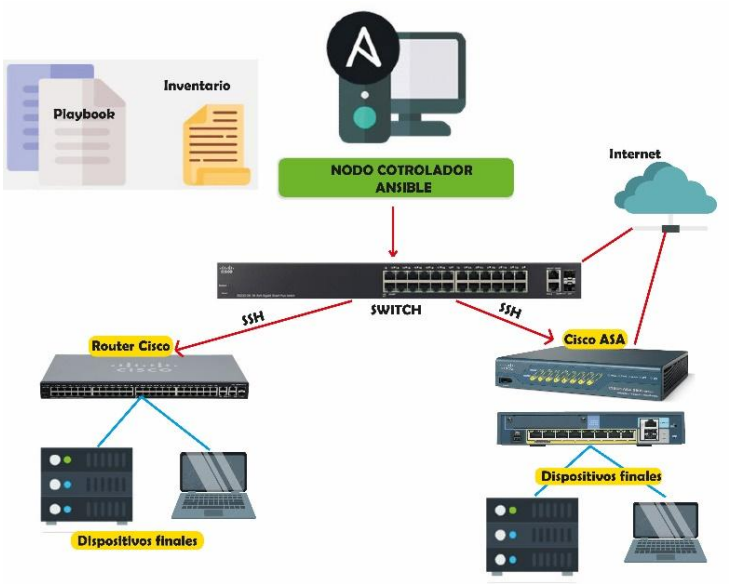

Figura 1.3 Automatización de redes

Programas para simular Redes

Para crear una red físicamente con todos los dispositivos necesarios para crear una DMZ contrae gastos económicos para aquellas personas que buscan realizar pruebas de funcionamiento o incluir nuevas tecnologías. Una solución que solventa este problema es el manejo de un *software* dedicado a simular equipos de red con todas las características. Entre los existentes se destacan Packet Tracer, Eve-ng, PNETLab y GNS3. Cada uno de los *softwares* antes mencionados ofrece la capacidad de crear ambientes de pruebas complejos, mostrando ventajas de rendimiento, operación, simulación, agregación de herramientas adicionales, etc. Incluso algunas son usadas por distintas organizaciones para certificaciones, como es el caso de Cisco [18] [19]. En la Tabla 1.1 se presentan las características de cada *software*.

Tabla 1.1 Diferencias de simuladores y emuladores de redes [18] [19] [20]

Programas	**Packet Tracer**	**Eve-ng**	**PNETLab**	**GNS3**
Instalación	Fácil, solo el programa.	Necesita instalar y configurar un servidor	Necesita instalar y configurar un servidor	Necesita instalar un servidor y la aplicación
Compatibilidad de proveedores	Solo equipos Cisco	Varios	Varios	Varios
Adquisición de imágenes	Ninguna	Algunas son de pago	Algunas son de pago	Algunas son de pago
Imágenes	No	Sí	Sí	Sí
Conexión	Terminal incluido	Con HTML5 o putty	Terminal o HTML5	Maneja putty u otros tipos
Ambiente real configurable	No, por parte de servidores Linux	Sí	Sí	Sí
Consumo de recursos	No en exceso	Sí	Sí	Sí

Cada *software* puede ser instalado en los distintos tipos de sistemas operativos establecidos como Linux, Mac y Windows. Por otra parte, la creación e instalación de los servidores requeridos para algunas aplicaciones como GNS3, EVE-ng y PNetlab se hace uso de programas de virtualización, los más usados son VirtualBox y VMware [21].

Un aspecto importante para crear redes en estos programas y simularlos es la memoria RAM con la velocidad de procesamiento de una máquina. Dependiendo de los programas a usar, se consume una cierta cantidad de memoria RAM y

almacenamiento mientras se está ejecutándose. Entre los *softwares* mencionados se destaca GNS3 por tener un menor consumo, ya que permite asociar no solo imágenes sino también servidores, máquinas virtuales, contenedores configurados, entre otros [22] . En la Figura 1.4 se puede visualizar el software GNS3 en donde se crean topologías reales.

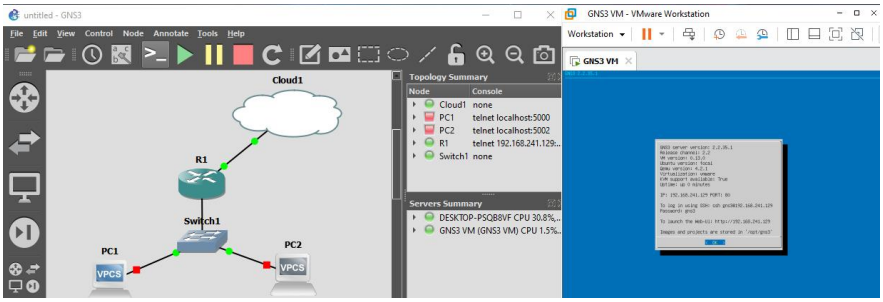

Figura 1.4 GNS3

2 METODOLOGÍA

En el presente trabajo de titulación se realizó una investigación de tipo documental, de modo que comienza por la búsqueda de contenidos sobre el funcionamiento, complementos, características, recursos necesarios y beneficios de la herramienta Ansible para desplegar una DMZ de manera automatizada en equipos especializados de seguridad Cisco.

A la investigación se suma temas importantes para configurar los equipos de seguridad Cisco ASA mediante Ansible. Se menciona el levantamiento de SSH tanto en el servidor Ansible y el dispositivo ASA, junto con la configuración del DMZ. Gracias a la investigación previa de las herramientas de Ansible y Cisco ASA se establecen configuraciones para las interfaces, NAT, DHCP y ACLs, reduciendo los errores de configuración. La metodología aplicada durante este proceso se denota en lo siguiente:

Se instaló los complementos necesarios para simular la red en GNS3, en donde se usó imágenes de ASA, router Cisco, Linux Ubuntu 16.04 para montar un pequeño servidor y un contenedor Linux Ubuntu 20.04 que contiene Ansible junto con openSSH. Con los elementos se armó una topología de red en GNS3 que permita montar una DMZ. Se estableció la configuración del router y la nube para que la topología tenga acceso a Internet. Por parte del ASA se configuró una interfaz junto con la habilitación de SSH, con restricciones de conexión, usuario y contraseña. En los servidores se monta un sistema de correo junto con DNS, mientras que en el servidor de Ansible se instaló los paquetes, incluido módulos que faciliten el trabajo de Ansible y SSH.

Antes de crear el *playbook*, se añadió a las configuraciones de SSH, las claves del equipo ASA. Por otro lado, se añade al inventario de Ansible los parámetros de conexión del ASA, por ende, para demostrar que el inventario funciona, se realizó una prueba de *ping pong.* Tras ello, se crea un modelo de *playbook* que configure un DMZ añadiendo interfaces, DHCP, NAT y creación de ACLs.

Finalmente se realizó las respectivas pruebas del funcionamiento del *playbook*, a su vez se verificó que el dispositivo ASA se haya configurado tras haber ejecutado el *playbook*. Además, se incluye pruebas para comprobar que la DMZ funcione correctamente, para ello, se usan las máquinas virtuales que verifican el funcionamiento y comportamiento del DMZ. Adicional, al *playbook* se añade opciones para que el usuario ingrese la información más importante del DMZ.

3 RESULTADOS

En este apartado se exponen aspectos relevantes para el levantamiento y administración de una DMZ a través de Ansible. Primeramente, se presenta un resumen de la investigación acerca de las características de DevOps, manejo de Ansible y DMZ. Luego, se construye una topología en GNS3, donde se pueda implementar una DMZ, en él se incluye máquinas virtuales que son utilizados para el nodo controlador de Ansible, máquinas clientes y servidores. Finalmente, se específican las conexiones SSH junto con la creación del *playbook* para levantar el DMZ, ayudando a comprender de mejor manera la intervención de Ansible en la automatización de dispositivos Cisco ASA en la red.

3.1 Análisis de DevOps y sus características

DevOps

De acuerdo a lo incluido en el marco teórico, la idea de DevOps consiste en la producción de nuevos servicios y productos en el menor tiempo posible, basándose en la colaboración entre diferentes departamentos. Además, a la idea se añade las características de automatización y un continuo mejoramiento en el proceso del levantamiento de los servicios o productos.

Ansible

Con respecto al marco teórico, se presenta brevemente las características de Ansible:

- La herramienta es libre, fácil y sencilla de usar.
- Su aplicación está relacionada al levantamiento de servicios en diferentes dispositivos.
- Requiere de tres componentes importantes: SSH, el *playbook* y Python.
- Ejecuta archivos en formato YAML.
- Mayormente compatible con sistemas Linux.
- Cuenta con un repositorio con diversos ejemplos de aplicación.
- Utiliza módulos preestablecidos para uso sencillo.
- Sigue la topología cliente-servidor.
- Instalación sencilla.

La instalación de Ansible en los sistemas Linux de la distribución Ubuntu se realiza mediante el comando ***apt install ansible*** y el único archivo de configuración de ansible está ubicado en el directorio /etc/ansible/ con el nombre de ***ansible.cfg***. También en la misma ubicación se encuentra el inventario por defecto de Ansible nombrado como ***hosts***. A partir de estos archivos y un *playbook* se puede usar Ansible en un dispositivo administrable.

DMZ

Una Zona Desmilitarizada abarca diferentes funciones entre las zonas creadas. La creación de una DMZ en un dispositivo Cisco ASA requiere de las siguientes configuraciones básicas:

- Interfaces para las distintas zonas.
- Protocolo DHCP para las zonas en caso de requerirla.
- NAT estático y dinámico.
- Rutas estáticas.
- Políticas de clase.
- Mapas de clase.
- Reglas indicadas por ACL.

La configuración de una DMZ depende de las necesidades presentes en la red previstas por un administrador, por lo que en el presente proyecto se denota las configuraciones básicas para un escenario específico.

3.2 Diseño del algoritmo para poder implementar DMZ

Las configuraciones básicas de una DMZ son la base para la creación del algoritmo. Por lo tanto, en esta sección se expone los comandos propios del dispositivo Cisco ASA que configuran una DMZ.

Configuración de interfaz

En un dispositivo Cisco ASA se requiere de los siguientes atributos para configurar una interfaz:

- **Dirección IP:** La asignación de una dirección IP puede configurarse como DHCP o dirección estática. La opción DHCP se establece cuando la interfaz obtiene una dirección de manera dinámica. En cambio, en la opción de dirección estática el usuario especifica la dirección IP. Para establecer una dirección estática en la interfaz se utiliza ***ip address direccion_IP mascara_de_Red***, por otra parte, para establecer DHCP se utiliza ***ip address dhcp***.
- **Nivel de seguridad:** Indica la confiabilidad de la red. Dentro de los equipos ASA el nivel de seguridad se asigna mediante ***security-level número (0 – 100)***.
- **Nombre de la interfaz:** Es un nombre asignado a las interfaces necesario en las configuraciones de DHCP, NAT, ACL, etc. El nombre de la interfaz elimina la necesidad de escribir la toda la interfaz, sea este *Ethernet*, *Fast Ethernet* y *Giga Ethernet* con su respectiva numeración (0/0, 0/1, 0/2, etc.). Para asignar un nombre a la interfaz se usa ***nameif nombre***.
- **Estado de la interfaz:** El parámetro tiene dos opciones para establecer sí la interfaz está habilitada o no. El comando para habilitar la interfaz es ***no shutdown***, por lo contrario, para deshabilitarlo es con ***shutdown***.

Configuración de DHCP

El dispositivo Cisco ASA permite establecer configuraciones adicionales como DHCP para asignar direcciones IP de manera dinámica o estático. Para su habilitación se requiere de los siguientes atributos:

- **Rango de direcciones IP**: Indica una cierta cantidad de direcciones que puede manejar DHCP para asignar direcciones IP en la red. El comando para su configuración es ***dhcpd address rango_IP nombre_zona***.

- **Direcciones de DNS**: Distribuye en la red las direcciones de los servidores de nombres de dominio. En el equipo Cisco ASA se configura como: ***dhcpd dns dirección_DNS1 dirección_DNS2 interface inside***.
- **Habilitación de DHCP**: El protocolo DHCP es habilitado mediante el comando ***dhcpd enable nombre_zona***, donde se especifica la zona en donde se va a distribuir las direcciones IP.

Configuración de una ruta estática

Un dispositivo Cisco ASA puede conectarse al Internet mediante una ruta estática, por lo cual es necesario su configuración en una DMZ. El comando ***route nombre_zona 0.0.0.0 0.0.0.0 Dirección_IP_Proveedor*** permite establecer la ruta estática en el ASA, siendo necesario contar con la dirección IP del proveedor para su funcionamiento. La dirección IP debe ser la interfaz del equipo vecino al cual está conectado el equipo ASA. El equipo vecino es el aparato que permite la salida al Internet, puede ser un dispositivo de red, un router o un switch.

Configuraciones de políticas y mapas de clase

Las configuraciones de políticas y mapas permiten asignar un comportamiento entre distintas zonas. Los comportamientos que se logren tras configurar las políticas y mapas son acerca del acceso permitido o no entre zonas, por ejemplo, una zona X podrá acceder a equipos de otra zona Y, pero la zona Y no tendrá permisos para acceder a los equipos de la zona X. Para establecer este tipo de comportamientos se utilizan los siguientes aspectos:

- **Mapa de clase:** Posibilita la identificación del tipo de tráfico sobre el cual se realiza una inspección. El parámetro se configura mediante ***class-map nombre-clase***.
- **Política de mapa:** Trata del tipo de operación a realizar con el tráfico, sea de tipo inspección, descarte o aprobación. Su comando es ***policy-map nombre-clase***.
- **Clase:** Dentro de la política de mapa se establece una clase con el comando ***class nombre-clase***. La clase indica el tipo de tráfico en el que se aplicara una operación. En una DMZ se puede establecer la operación de inspección mediante el comando ***inspect tráfico (icmp dns, http, snmp, etc.)***.
- **Política de servicio:** Indica a que interfaz o zona se aplicará el mapa y política. Su modo de configuración es con el comando ***service-policy nombre-clase interface nombre-zona***.

Configuración de NAT

Dentro de una zona X de una DMZ se encuentran los servidores, aquellos que pueden ser públicos o privados. Tras este detalle se menciona el uso de NAT para que los usuarios del exterior puedan acceder a los servidores mediante una IP pública. Dentro de un equipo Cisco ASA se configura NAT mediante los siguientes aspectos:

- **Objeto de red:** Objeto que contiene los atributos del NAT, se crea mediante el comando ***object network nombre_objeto***.
- ***Host o subred***: En los atributos del objeto de red se puede especificar un *host* o la red al que se va a aplicar el NAT dinámico o estático. Al establecer un *host* se utiliza ***host IP_host*** y para denotar una red es mediante ***subnet direccionIP_de_red mascara_de_red***.
- **NAT:** Se incluye dentro del objeto de red, especificando en que zona se va a aplicar el NAT junto con su tipo: estático y dinámico. Para el caso de un NAT dinámico se hace uso del comando ***nat (zona1,zona2) dynamic interface***, por otro lado, para establecer un NAT estático se usa ***nat (zona1,zona2) static IP_publica***.

Configuración ACLs

La configuración de las Listas de Control de Acceso dentro de un dispositivo Cisco ASA pueden ser de tipo estándar o extendida. Un ACL requiere de los siguientes atributos:

- Nombre o número de ACL
- Tipo (estándar y extendida)
- Protocolo (ejemplo.: IP, TCP, UDP, ICMP, etc.)
- Direcciones de origen y destino
- Puertos de origen y Destino

Por ejemplo, un ACL que impida consultas mediante el protocolo ICMP puede tener la estructura: ***access-list nombre_ACL extended deny icmp any any***.

3.3 Implementación de una DMZ mediante herramientas de DevOps

Para crear una DMZ mediante Ansible, se necesita de una topología en donde pueda ser implementada, siendo su aplicación en el campo de seguridad. Entonces, para poder tener una topología de red lo más real posible, se instala el *software* GNS3 con

su respectivo servidor, creado en el programa de virtualización de Sistemas Operativos VMware. Además, dentro de GNS3 se instalan imágenes respectivas del Cisco ASA, router Cisco, máquinas Linux de la distribución Ubuntu y la imagen ligera de *tiny core* para simular a los clientes.

Instalación de GNS3

Los archivos necesarios para la instalación de GNS3 se encuentran en la página oficial. Se descargan los archivos tomando en consideración el instalador para Windows 10 y la imagen del servidor de GNS3 compatible con VMware. Al instalar el servidor en VMWare se conservan las características que solicita la VM (*Virtual Machine*) como recursos de 40 (GB) de almacenamiento, memoria RAM de 4 (GB), una interfaz de red NAT y otra interfaz de red interna. Con los recursos instalados se asocia el programa de GNS3 con el servidor mediante el nombre de la máquina virtual, tal como se muestra en la Figura 3.1.

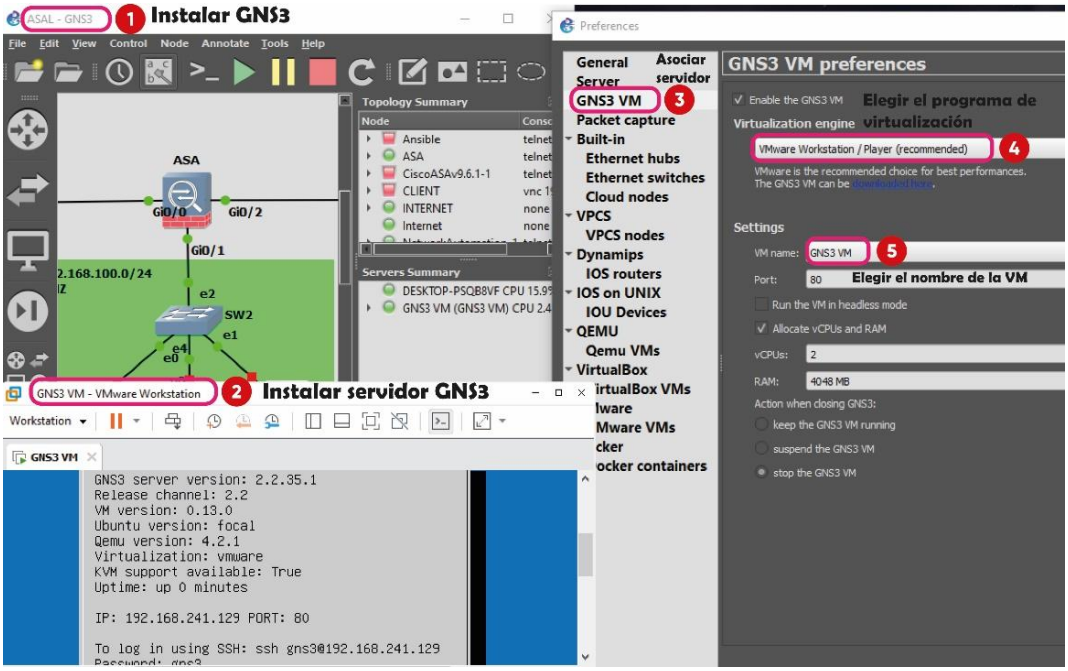

Figura 3.1 Instalación de GNS3

Instalación de imágenes en GNS3

Las imágenes instaladas para la implementación son: router Cisco, Cisco ASA, Tiny Core de Linux, Ubuntu Server 16.04 y el contenedor *Network Automation*. La instalación de cada imagen se realiza de manera similar. El router, el ASA, Tiny Core y el Ubuntu Server 16.04 son instalados a través de la ventana de preferencia de GNS3, mientras que el contenedor *Network Automation* se instalan con los archivos del

repositorio de GNS3. Un ejemplo de la instalación de imágenes se muestra en la Figura 3.2.

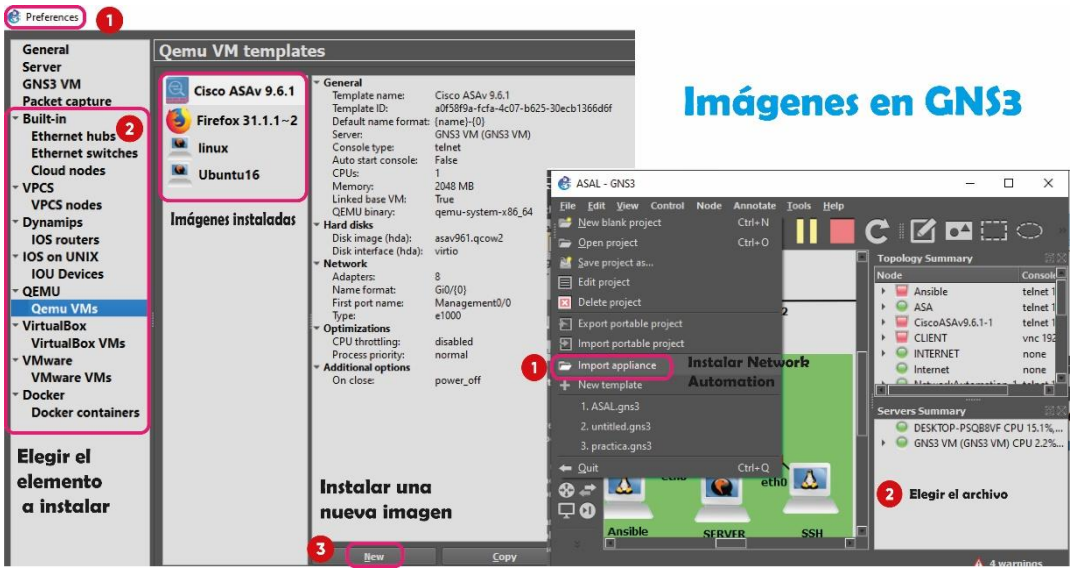

Figura 3.2 Imágenes en GNS3

Adicional, solamente los archivos en formato QCOW2 pertenecientes a imágenes de servidores o equipos de red pueden ser instaladas en el software de GNS3.

Topología para implementar una DMZ

Con los recursos previamente instalados se procede a construir la topología que muestre el escenario perfecto para la implementación de una DMZ mediante Ansible. La topología creada para la implementación del proyecto se muestra en la Figura 3.3. Además, en la topología se añade una nube NAT acompañado de un router que permiten el acceso al Internet, los cuales ayudan a descargar los paquetes de Ansible, el módulo cisco.asa y herramientas adicionales para configurar servidores, que después se utilizarán en la comprobación del DMZ.

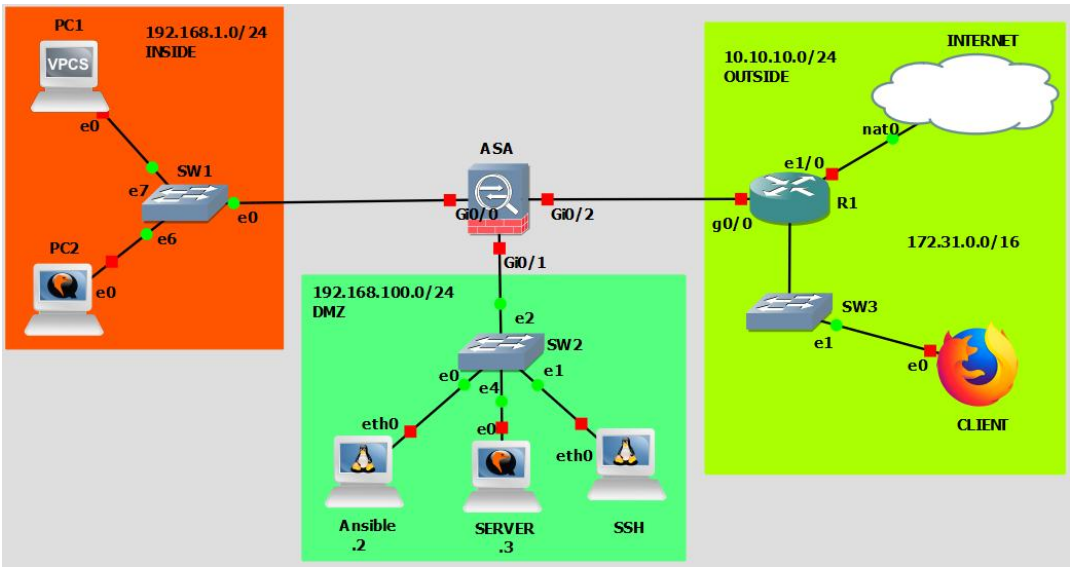

Figura 3.3 Topología DMZ en GNS3

Configuración del router cisco

En el dispositivo router Cisco se configura las interfaces *Giga Ethernet* 0/0, *Ethernet* 1/1 y *Ethernet* 1/0 respectivamente con las direcciones IP 10.10.10.1/24, 172.31.0.1/16 y DHCP. En la última interfaz se habilita DHCP para obtener una dirección IP de la nube NAT. Los comandos usados para habilitar la dirección IP fueron ***ip address 10.10.10.1 255.255.255.0*** e ***ip address dhcp***.

Para configurar el router como el punto que permite la salida hacia el Internet, se establece un NAT dinámico mediante el uso de dos reglas ACL. El primer ACL se establece con el comando ***access-list 100 permit ip any any*** y se relaciona con la interfaz Ethernet 1/0 a través de ***ip nat inside source list 100 Ethernet1/0 overload***. Finalmente, en la interfaz Giga0/0 se añade el atributo ***ip nat inside***, mientras que en la interfaz Ethernet1/0 se añade ***ip nat outside***. Al establecer estas configuraciones se permite a los dispositivos internos de la red la conexión a Internet.

Para el caso de la otra red 172.31.0.0/16 se establece otro ACL con la numeración 101 siendo los comandos: ***access-list 101 permit ip any any*** e ***ip nat inside source list 101 Ethernet1/0 overload***. Para terminar, se incluye solamente el comando ***ip nat inside*** en la interfaz Ethernet1/1, porque la interfaz Ethernet1/0 ya cuenta con el atributo ***ip nat outside***. En la Figura 3.4 se puede visualizar el archivo ***running-config*** que indica las configuraciones realizadas en el dispositivo router Cisco.

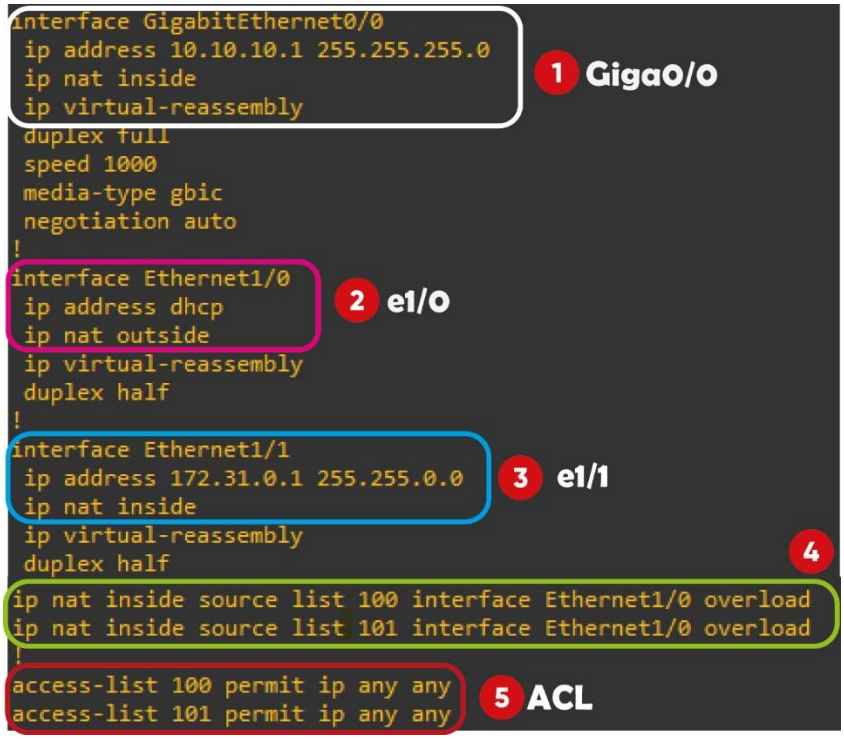

Figura 3.4 Router Cisco

Instalación Ansible y SSH

Al usar una máquina Linux para el nodo administrador de Ansible, se aprecia que la máquina no tiene instalado los paquetes de SSH y Ansible. La máquina se conecta momentáneamente a la nube NAT con acceso al Internet para instalar las herramientas. Para el caso de SSH se instala mediante el comando ***sudo apt install openssh-server***, de ahí se puede ingresar a su repositorio y configurarlo. Para el caso de Ansible de igual forma se usa ***sudo apt install Ansible***, incluso se puede especificar la versión a instalar. De acuerdo a las características de Ansible, también se requiere de Python, aunque dentro de Linux ya viene instalado. Los directorios correspondientes a estos paquetes se encuentran en la dirección /etc/ssh/ y /etc/ansible/ en donde se ubican los archivos de configuración para su funcionamiento.

En la colección de Ansible existe una gran cantidad de módulos útiles para administrar distintos equipos. Para el caso de los equipos ASA se usa el módulo cisco.asa, pero por constantes cambios en las nuevas versiones de este módulo, se opta por la versión 3.0.0, con la finalidad de tener menos problemas al momento de usar el módulo y administrar el ASA. El comando para instalar la versión 3.0.0 del módulo es ***ansible-galaxy collection install cisco.asa:3.0.0***. En la Figura 3.5 se muestra la instalación de SSH, Ansible y el módulo cisco.asa.

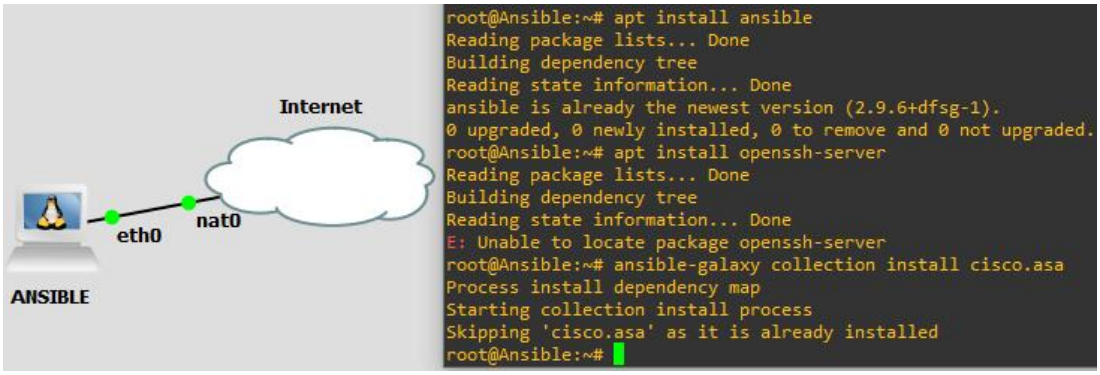

Figura 3.5 Instalación de Ansible

Dentro de GNS3 se puede hacer uso de contenedores, por lo que se tiene un contenedor especial denominado como *Network Automation* encargado de automatizar la red. El contenedor es una distribución de Linux Ubuntu versión 20.04 que ya cuenta con las herramientas de Ansible y SSH, por lo tanto, en la topología del proyecto se usa el contenedor *Network Automation* para el nodo administrador. Cabe mencionar que dentro del contenedor se necesita instalar solamente el módulo cisco.asa, por ende, se requiere la conexión con la nube NAT.

Asignación de dirección IP estática

El nodo administrador de Ansible es configurado con una dirección IP estática necesaria para evitar problemas de direcciones en la red, incluso se utiliza en la configuración del ASA para limitar la conexión de SSH. La máquina de Ansible se coloca en la red DMZ como se muestra en la topología de la Figura 3.3.

Para establecer la dirección estática en el nodo controlador se ingresa al directorio /etc/networks/, dentro del mismo se edita un archivo denominado como ***interfaces***. Los datos establecidos dentro del servidor Ansible son: una dirección IP estática 192.168.100.2/24, Gateway 192.168.100.1 y dos servidores DNS 8.8.8.8 192.168.100.3. En la Figura 3.6 se muestra la asignación mencionada.

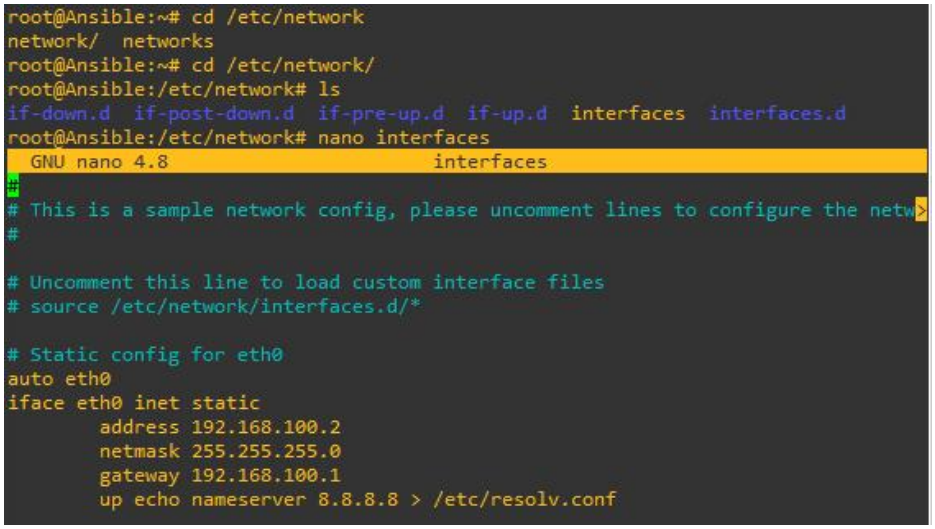

Figura 3.6 Asignación de dirección IP estática

Para aplicar los cambios de la nueva dirección IP se reinicia el servicio de red o simplemente reiniciar la máquina Ansible. Para comprobar los cambios se puede usar del comando ***ifconfig*** o ***show ip –a***, todo depende de la versión del sistema operativo para admitir estos comandos, caso contrario se requiere de un comando diferente que muestre la dirección de IP del equipo. En la Figura 3.7 se muestra la nueva dirección.

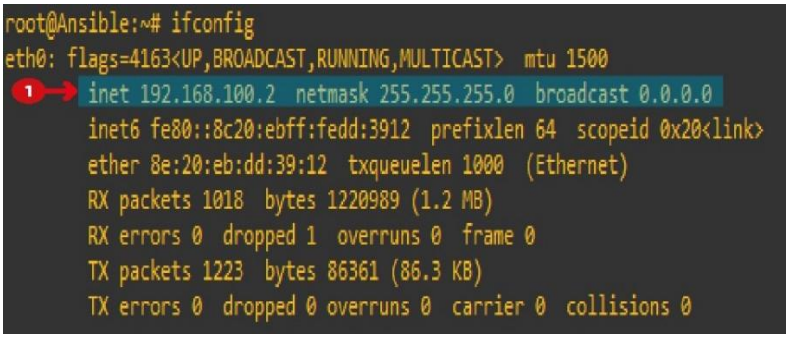

Figura 3.7 IP estática

Configuración de SSH

SSH es necesario para el uso de Ansible, ya que permite una conexión segura con algún dispositivo, en este caso el nodo Ansible junto al Cisco ASA. Para establecer SSH en equipos ASA es necesario configurar ciertos parámetros de seguridad que permiten limitar el acceso de otros dispositivos que no tienen permitido la conexión SSH.

De acuerdo a la topología de la Figura 3.3 donde se encuentra el Cisco ASA se configura la interfaz Gigabit Ethernet 0/1 para establecer la conexión SSH con el servidor Ansible localizado en la zona DMZ. Dentro de la interfaz se incluye una

dirección IP estática 192.169.100.1/24, un nivel de seguridad del 50% y el nombre del interfaz denotado como DMZ. En la Figura 3.8 se puede apreciar los primeros pasos de la configuración SSH.

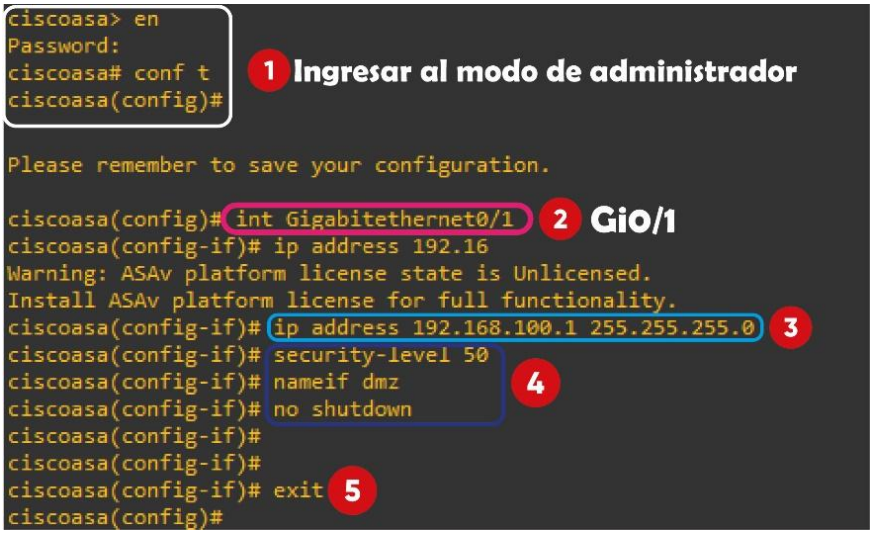

Figura 3.8 SSH en equipos ASA parte 1

Además, para limitar el acceso al Cisco ASA se establece una contraseña de administración *enable*, un usuario con su respectiva contraseña, la autentificación local del nuevo usuario con SSH, creación de nuevas llaves con RSA y especificación del *host.* Por ende, solamente el host 192.168.100.2 perteneciente al nodo Ansible tiene permisos de conexión al equipo ASA mediante SSH. Las configuraciones mencionadas se pueden observar en la Figura 3.9.

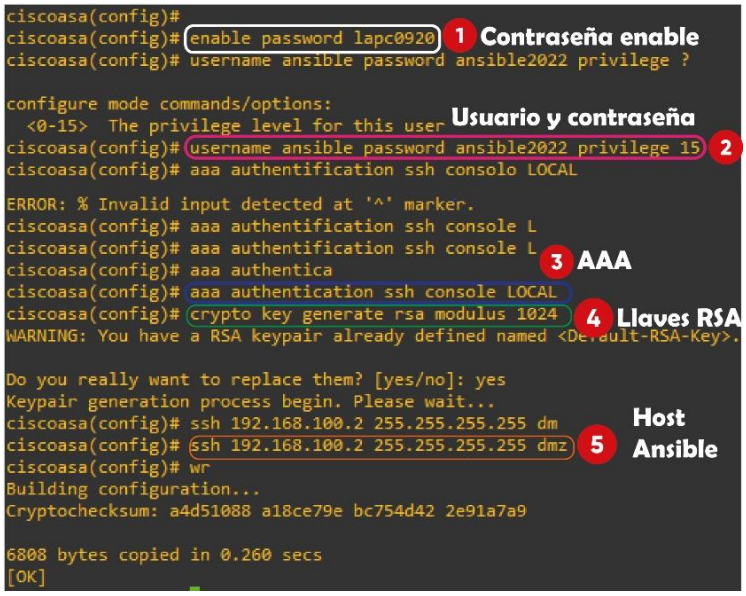

Figura 3.9 SSH en equipos ASA parte 2

Por otra parte, en el servidor de Ansible se edita el archivo ssh_config localizado en el directorio /etc/ssh/. Dentro del archivo ssh_config se descomenta una línea necesaria para el cifrado de las conexiones SSH y se agrega una línea nueva para el intercambio de llaves SSH. Este proceso se hace para que Ansible pueda usar SSH con el equipo ASA y no se genere un error en relación al método de intercambio de llaves. Las líneas del algoritmo de cifrado e intercambio de llaves se muestran en la Figura 3.10.

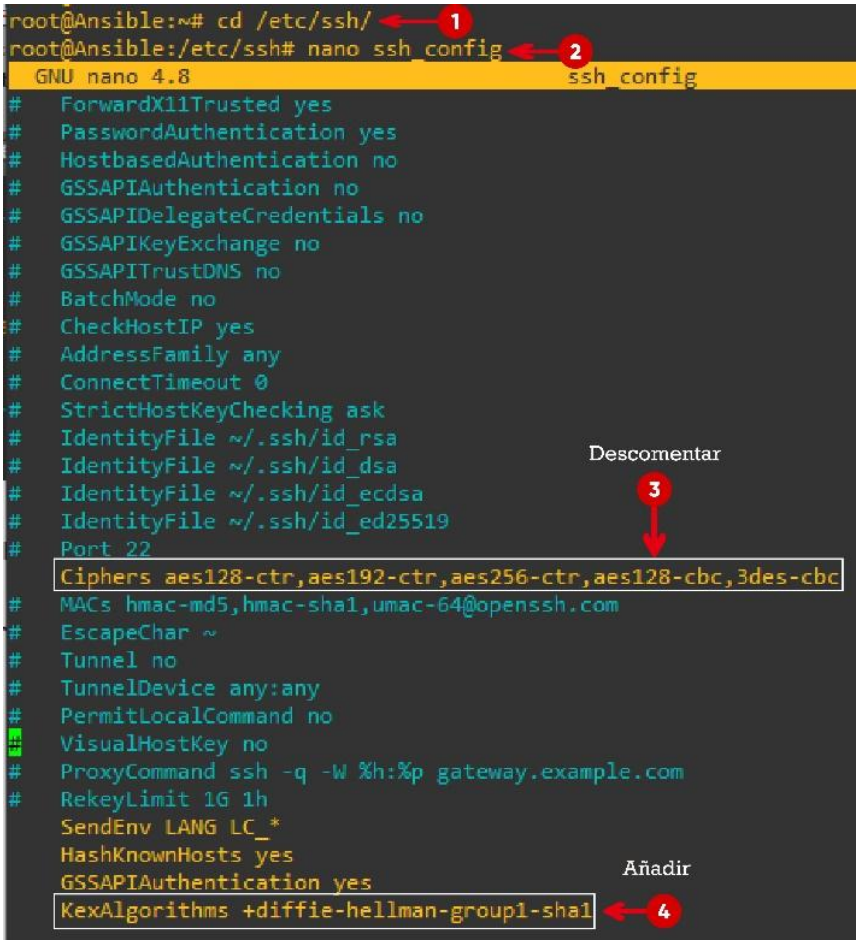

Figura 3.10 Configuración SSH en el nodo Ansible

A continuación, se crean las nuevas llaves SSH en el nodo Ansible con el comando ***ssh-keygen***. También se establece la conexión SSH con el ASA para copiar las claves con ***ssh-copy-id usuario@IP_del_dispositivo***. Con este proceso se facilita la conexión SSH cuando lo usa Ansible. La configuración se observa en la Figura 3.11.

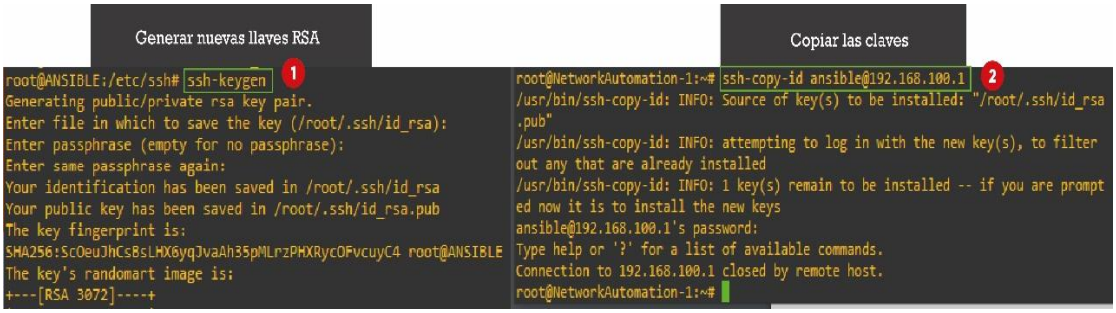

Figura 3.11 Claves SSH entre ASA y el nodo Ansible

Inventario Ansible

En Ansible existen dos archivos, el primero contiene la configuración de Ansible y el segundo la información de los *hosts* para establecer la conexión. El segundo archivo por defecto se encuentra con el nombre de hosts, pero se lo conoce como el inventario de Ansible. De igual forma Ansible por defecto toma datos de este inventario para realizar las conexiones, aunque puede configurarse para apuntar a otro directorio.

Para evitar problemas se usa el directorio por defecto de Ansible, por ende, se agrega la información del dispositivo ASA dentro de Ansible. La información a agregar del dispositivo ASA en el repositorio es:

- Nombre con su dirección IP: ***ASAL ansible_host = 192.168.100.1***.
- Modulo cisco.asa: ***ansible_network_os = cisco.asa.asa***.
- Usuario y contraseña SSH: ***ansible_user = ansible*** y ***ansible_password = ansible2022***.
- Aceptación al ingreso de administración *enable*: ***ansible_become = true***.
- Contraseña *enable*: ***ansible_become_pass = lapc0920***.
- Variables de conexión: ***ansible_connection=ansible.netcommon.network_cli*** y ***ansible_become_method=ansible.netcommon.enable***.
- Lenguaje de interpretación Python: ***ansible_python_interpreter = /usr/bin/Python3***.

Los datos registrados en el inventario dependen del dispositivo a administrar con Ansible, en el caso de los equipos ASA se requieren de estos datos para establecer una conexión correcta. En la Figura 3.12 se muestran los datos establecidos del ASA.

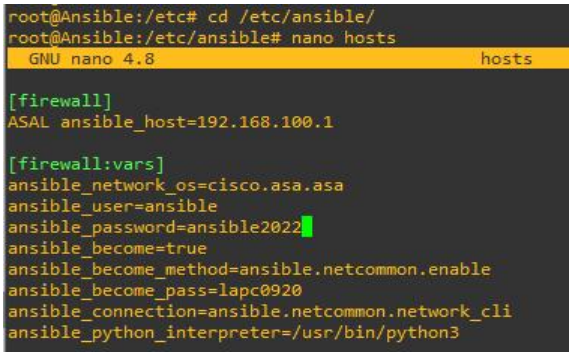

Figura 3.12 Inventario Ansible para equipos ASA

Para comprobar que las configuraciones realizadas junto con el inventario Ansible están correctamente implementadas se realiza una prueba ***ping pong*** tal como se observa en la Figura 3.13. Esta prueba trata de un ***ping*** realizado a través de Ansible e indica que los datos del inventario son correctos en caso de ser positivo, en caso de no serlo la prueba muestra en mensaje de error al establecer el ping. Con el comando ***ansible (nombre asignado en el inventario) –m ping*** se puede hacer la prueba. Este tipo de prueba es indispensable para comenzar a construir el *playbook*.

Figura 3.13 Prueba *ping pong*

Creación del *playbook*

Para la creación del *playbook* se considera ciertos parámetros para administrar el dispositivo ASA. Entre ellos se encuentra la configuración de interfaces faltantes, habilitación de DHCP para la red privada, reglas entre la red DMZ e inside, salida al Internet, configuración NAT y creación de una lista de control acceso para el tráfico permitido. A estas especificaciones y acorde a lo topología creada se crea el *playbook* completo que se muestra en el ANEXO III.

El *playbook* se divide en tres partes referentes a: parámetros de conexión, variables o archivos a usar y las tareas. En los parámetros de conexión para los dispositivos Cisco ASA es necesario incluir el medio de conexión CLI, el nombre del host de acuerdo al inventario y atributos para el acceso como administrador enable del Cisco ASA. Al no incluir los elementos mencionados en el *playbook* se presenta errores de conexión por los permisos insuficientes.

En el apartado de variables se incluye otros libros de jugadas o variables alojados en otros archivos que pueden ser incluidos o no en el *playbook*, incluso para tener un procedimiento dinámico puede agregarse atributos para solicitar el acceso de datos por parte del usuario. La siguiente parte del *playbook* es el apartado de tareas donde se incluye una cantidad de tareas que configuran un host, pero la plantilla es diferente para cada host administrado por Ansible y también depende del módulo para los distintos hosts. En la Figura 3.14 se muestra las partes del *playbook* su funcionamiento en los equipos ASA.

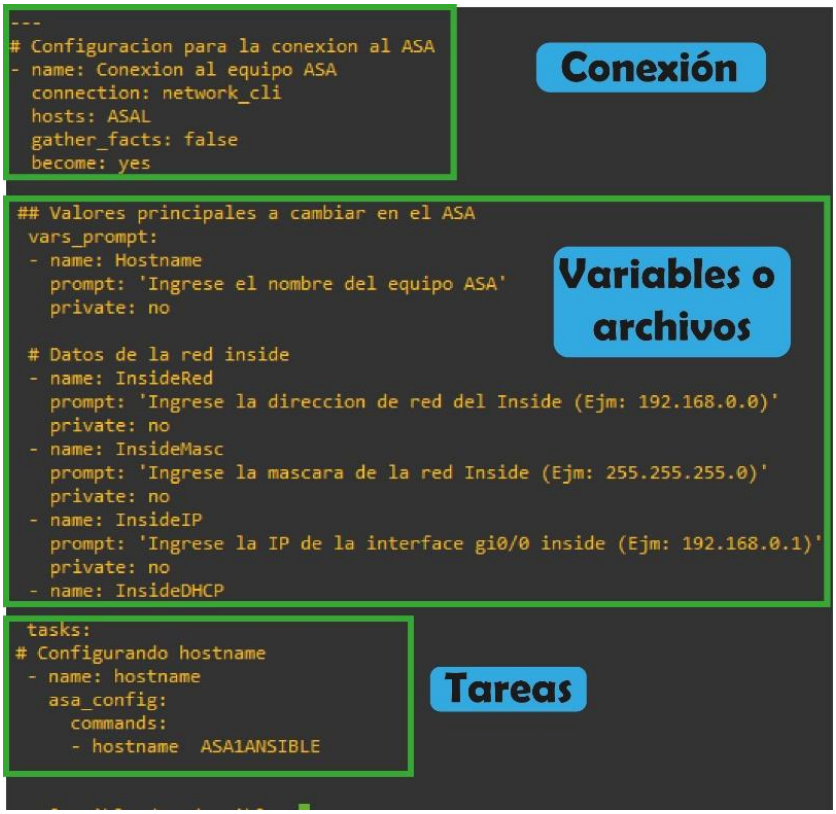

Figura 3.14 Partes del *playbook*

En el *playbook* se denota la utilización del módulo cisco.asa. El módulo tiene limitaciones al momento de configurar los equipos ASA. Un ejemplo comparativo es al momento de configurar una interfaz con los módulos cisco.ios y cisco.asa. Utilizando cisco.ios en el *playbook* se incluye el módulo especifico de IOS, la dirección IP, máscara de red en números enteros y la opción para habilitar interfaz. En cambio, el módulo cisco.asa requiere de los comandos normales para habilitar la interfaz como sí se estuviera manejando el equipo. Pese a esta diferencia el módulo es capaz de administrar el dispositivo Cisco ASA, siempre que los comandos sean correctamente utilizados en el *playbook*.

Una breve prueba del módulo Cisco ASA es cambiar el nombre del dispositivo. Para lograrlo, se crean dos *playbooks* diferenciándose en un sentido directo y dinámico. El primer *playbook* asigna el nombre directamente tras ejecutarse, en cambio en el segundo se incluye una variable que solicita al usuario un nombre para el dispositivo ASA. Como se mencionó en el apartado anterior para hacer uso del módulo ASA solamente se necesita el módulo y el comando para cambiar el nombre del equipo, siendo este: ***hostname nuevo_nombre***. Las pruebas se presentan en la Figura 3.15, en donde se ejecuta el *playbook* mediante el comando ***ansible-playbook nombre.yml***.

Figura 3.15 Primara prueba de Ansible en ASA

Siguiendo el ejemplo de prueba se crean 2 *playbooks* con distintas tareas que permitan configurar las características de una DMZ. El primer *playbook* es la base para construir el segundo playbook, ya que contiene la plantilla de tareas. El segundo *playbook* tiene adicional una sección de variables para que el usuario ingrese la información del ASA cuando ejecute el *playbook*. En la Tabla 3.1 se muestra las variables utilizadas para el segundo playbook con su respectiva descripción.

Tabla 3.1 Variables del segundo *playbook*

Variables	**Descripción**
Hostname	Nombre del dispositivo ASA
InsideIP	Dirección IP de interfaz Gi0/0
InsideRed	Dirección de red de la zona inside
InsideMasc	Máscara de la red inside
InsideDHCP	Rango de direcciones IP para DHCP
OutsideIP	Dirección IP de interfaz Gi0/2
OutsideMasc	Máscara de la red outside
IPvecino	Direccion IP del proveedor de Internet
dmzRED	Dirección de red de la zona DMZ
dmzMasc	Máscara de la red DMz
IPprivadaServer	Dirección IP privada del servidor de correo
IPpublicServer	Dirección IP pública del servidor de correo

La plantilla para pedir el ingreso de información para la configuracion del dispositivo se realiza mediante el parámetro ***vars_prompt***, dentro de este parámetro de pueden establecer los atributos *name*, *prompt* y *private*. El atributo *name* permite establecer el nombre de la variable. El atributo *prompt* ayuda a ingresar el mensaje que se mostrara al usuario cuando ejecute el *playbook* y a su vez permite el ingreso de la información solicitada en el mensaje. Finalmente, el atributo *private* se encarga de agregar un método de cifrado a la información que ingrese el usuario. En la Figura 3.16 se muestra la platilla de la sección variables del segundo *playbook*.

```
## Valores principales a cambiar en el ASA
 vars_prompt:
 - name: Hostname
   prompt: 'Ingrese el nombre del equipo ASA'
   private: no

 # Datos de la red inside
 - name: InsideRed
   prompt: 'Ingrese la direccion de red del Inside (Ejm: 192.168.0.0)'
   private: no
 - name: InsideMasc
   prompt: 'Ingrese la mascara de la red Inside (Ejm: 255.255.255.0)'
   private: no
 - name: InsideIP
   prompt: 'Ingrese la IP de la interface gi0/0 inside (Ejm: 192.168.0.1)'
   private: no
 - name: InsideDHCP
   prompt: 'Ingrese rango de IPs para el DHCP inside  (Ejm: 192.168.0.10-192.168.0.20)'
   private: no

  # Datos de la red Outside
 - name: OutsideIP
   prompt: 'Ingrese una IP perteneciente a la red Outside (Ejm: 20.20.20.6)'
   private: no
 - name: OutsideMasc
   prompt: 'Ingrese la mascara de la red Outside (Ejm: 255.255.0.0)'
   private: no
 - name: IPvecino
   prompt: 'Ingrese la direccion IP del Router Vecino (Ejm: 20.20.20.1)'
   private: no

  # Datos de la red DMZ
 - name: dmzRED
   prompt: 'Ingrese direccion de red de la red DMZ (Ejm: 192.168.200.0)'
   private: no
 - name: dmzMasc
   prompt: 'Ingrese mascara de la red DMZ (Ejm: 255.255.255.0)'
   private: no

  #Datos para el NAT
 - name: IPprivadaServer
   prompt: 'Ingrese IP privada del servidor'
   private: no
 - name: IPpublicServer
   prompt: 'Ingrese IP publica del servidor'
   private: no
```

Figura 3.16 Plantilla de Variables

A continuación, se procede con la creación de tareas para configurar las interfaces faltantes del dispositivo ASA: la Giga Ethernet 0/0 y Giga Ethernet 0/2 que corresponden respectivamente a la red *inside* y *outside*.

En relación con los atributos para configurar interfaces de una DMZ, se establece dos tareas dentro del *playbook*. La primera tarea se encarga de configurar la interfaz Giga Ethernet 0/0, estableciendo: una dirección de 192.168.1.1/24, nivel de seguridad 100, nombre ***inside*** y la interfaz habilitada. La segunda tarea configura la interfaz Giga Ethernet 0/2, indicando: una dirección estática de 10.10.10.2/24, un nivel de 0 y nombre ***outside*** con la interfaz habilitada. Al utilizar el módulo cisco.asa se incluye los parámetros *lines* y *parents*, los cuales indican el lugar del archivo ***running-config*** en donde se registran las nuevas configuraciones. En la Figura 3.17 se muestran las tareas de configuración de las interfaces.

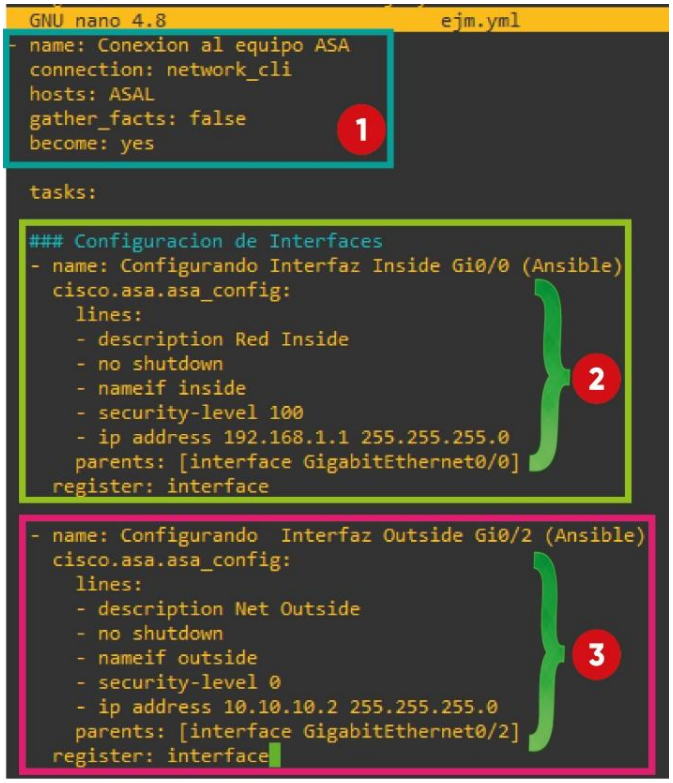

Figura 3.17 *Playbook* configuración de interfaces

En el *playbook* anterior indica parámetros de los interfaces ya configurados, mientras que en el segundo se utilizan variables para configurar las interfaces de manera dinámica, solicitando al usuario ingresar los datos al momento de ejecutar el *playbook.* Las variables usadas para la interfaz Giga Ethernet 0/0 son ***{{InsideIP}}*** y ***{{InsideMasc}}***, en cambio, para la interfaz Giga Ethernet 0/2 son ***{{OutsideIP}}*** y ***{{OutsideMasc}}.*** En la Figura 3.18 se pude apreciar el segundo *playbook*.

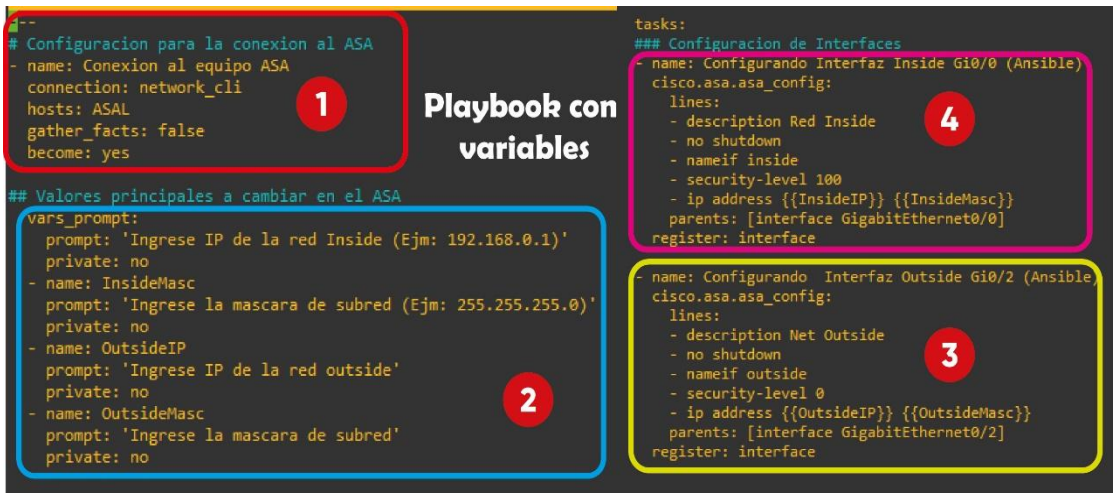

Figura 3.18 *Playbook* configuración de interfaces dinámica

En la siguiente sección del *playbook* se asigna una tarea para la configuracion de DHCP en la zona de red *inside*, en donde se utiliza el módulo cisco.asa denotado como ***asa_config***, al usar este módulo específico se puede usar el atributo ***commands*** que permite ejecutar comandos en el dispositivo Cisco ASA. Los comandos ubicados en el *playbook* para configurar DHCP en la zona *inside son:*

- ***dhcpd address 192.168.1.30-192.168.1.60***
- ***dhcpd dns 8.8.8.8 192.168.100.3 interface inside***
- ***dhcpd enable inside***

En la Figura 3.19 se muestra la sección para la configuración de DHCP mediante Ansible y contiene la misma información anteriormente descrita.

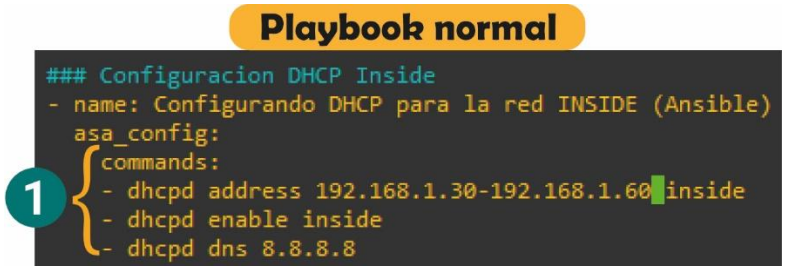

Figura 3.19 DHCP con Ansible

El segundo *playbook* contiene la misma información de la Figura 3.19, pero se diferencia por incluir una variable denominada como ***{{InsideDHCP***}}, tal como se muestra en Figura 3.20. La variable permite el ingreso del rango de direcciones IP a ser distribuidas con DHCP en la zona inside en caso de que un usuario necesite establecer su propio rango.

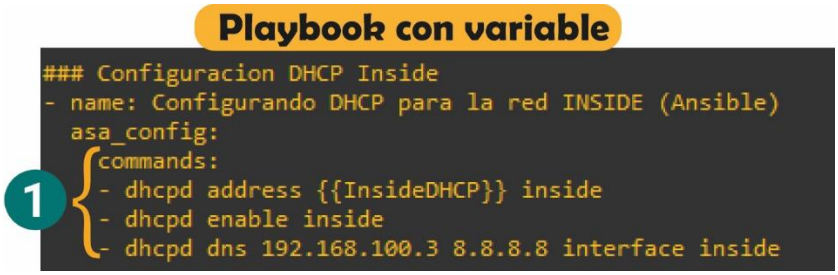

Figura 3.20 DHCP configuración dinámica

A continuación, se procede con la creación de una nueva sección utilizada en la configuracion de una ruta estática mediante Ansible. En el *playbook* se utiliza el atributo ***commands*** del módulo cisco.asa y el comando ***route outside 0.0.0.0 0.0.0.0 10.10.10.1***. En la Figura 3.21 se muestra la sección del *playbook* para establecer una ruta estática de las dos formas, primero sin variables y segundo con variables.

Además, la dirección estática es establecida en congruencia con la topología diseñada. En el caso del segundo *playbook* se utiliza la variable ***{{IPvecino}}*** para el ingreso de datos.

Figura 3.21 Ruta estática con Ansible

Después, en el *playbook* se crea tres tareas dedicadas a la creación de políticas y mapas de clase. En la primera tarea utilizando el atributo *commands* del módulo cisco.asa se establecen los siguientes comandos:

- Mapa de clase: ***class-map INSIDE-DMZ***
- Política de mapa: ***policy-map INSIDE-DMZ***
- Clase: ***class INSIDE-DMZ***
- Política de servicio: ***service-policy INSIDE-DMZ interface inside***

Dentro de la segunda tarea se establece una inspección de tráfico por defecto del dispositivo Cisco ASA, siendo ***match default-inspection-traffic***, para lograrlo se usa el atributo *lines* y *parents*, indicando que la configuracion de inspección se ubique en ***class-map INSIDE-DMZ***. Finalmente, en la tercera tarea se indica el tipo de tráfico a inspeccionar siendo los más comunes ICMP, DNS, HTTP, entre otros. Los protocolos a inspeccionar son colocados mediante el atributo *parents* en policy-map INSIDE-DMZ, class INSIDE-DMZ. En la Figura 3.22 se puede observar las tres tareas mencionadas junto con los parámetros usados en Ansible.

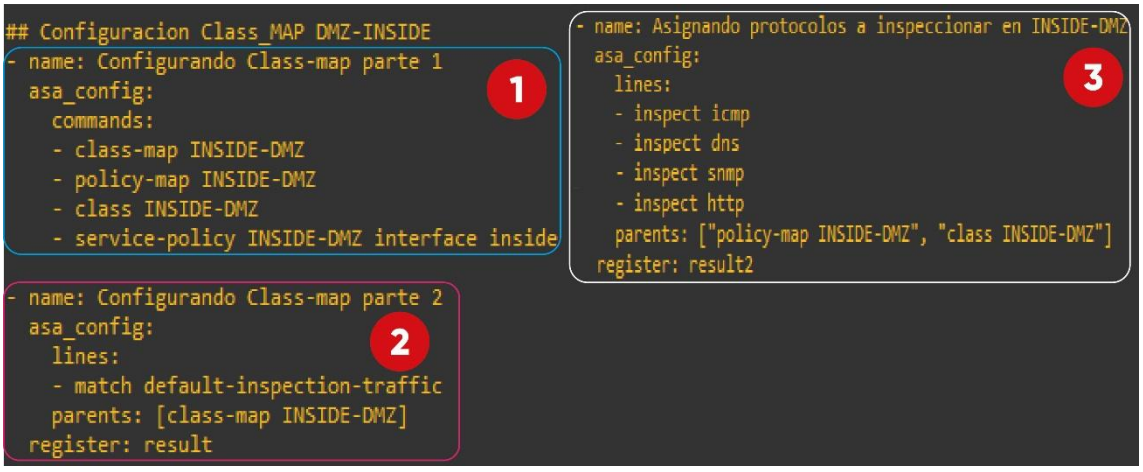

Figura 3.22 Configuración *Inside* – DMZ

Las 4 tareas siguientes son utilizadas para configurar NAT en el dispositivo Cisco ASA. En la primera tarea se crean tres objetos para el servidor Linux, la red *inside* y la red DMZ. Los objetos de red se crean con la siguiente denominación: ***object network SERVER***, ***object network DMZ-INTERNET*** y ***object network LAN-INTERNET***. También se utiliza el atributo *commands* para que los comandos se ejecuten en el dispositivo ASA mediante Ansible. La primera tarea de NAT se muestra en la Figura 3.23.

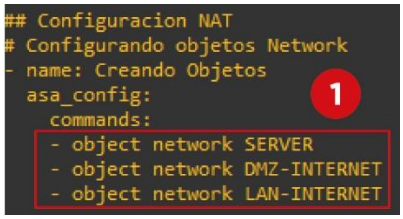

Figura 3.23 Objetos de red

La segunda tarea establece un NAT estático para el servidor de correo con distribución Linux Ubuntu 16.04. En el *playbook* se utiliza *lines* denominando los comandos ***host 192.168.100.3 y nat (dmz,outside) static 10.10.10.20***, que serán ubicados en ***el object network SERVER*** mediante *parents*. Usando la misma estructura se crea otra plantilla con dos variables que solicitan las direcciones IP del servidor. La variable usada para la IP privada es ***{{IPprivadaServer}}*** y para la IP pública ***{{IPpublicServer}}***. Esta sección del *playbook* de las dos plantillas se muestran en la Figura 3.24.

Figura 3.24 NAT estático con Ansible

En la tercera tarea se establece un NAT dinámico para la zona DMZ, permitiendo el acceso al Internet de cualquier host ubicado en esta zona. En la plantilla de esta sección se especifica la red DMZ con ***subnet 192.168.100.0 255.255.255.0*** y para el NAT mediante ***nat (dmz,outside) dynamic interface***. Finalmente, la cuarta tarea tiene la misma función que la tercera tarea, pero pertenece a un NAT dinámico de la zona *inside,* siendo sus comandos: ***subnet 192.168.1.0 255.255.255.0*** y ***nat (inside,outside) dynamic interface***. En la tarea 3 y 4 corresponde respectivamente para los objetos de ***DMZ-INTERNET*** y ***LAN-INTERNET***. Las tareas mencionadas se

encuentran en la Figura 3.25, en donde se adjunta la plantilla con variables para el ingreso de las direcciones IP públicas, privadas y de red, necesarios para estableces el NAT.

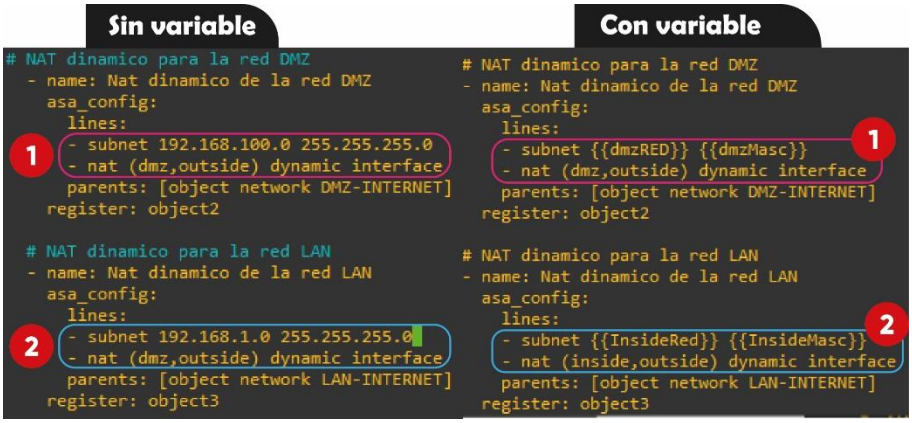

Figura 3.25 NAT dinámico con Ansible

La configuración de ACLs con Ansible puede ser mediante la utilización de comandos normales o con la característica del módulo cisco.asa. En el módulo cisco.asa existe una plantilla para completar los datos de un ACL de manera sencilla. Entonces se procede con la creación de una plantilla de 8 ACL que permitan consultas al servidor de correo, guiándose de los siguientes comandos:

- ***access-list acls extended deny icmp any any***
- ***access-list acls extended permit tcp any host 192.168.100.3 eq www***
- ***access-list acls extended permit udp any host 192.168.100.3 eq www***
- ***access-list acls extended permit tcp any host 192.168.100.3 eq domain***
- ***access-list acls extended permit udp any host 192.168.100.3 eq domain***
- ***access-list acls extended permit tcp any host 192.168.100.3 eq smtp***
- ***access-list acls extended permit tcp any host 192.168.100.3 eq pop3***
- ***access-list acls extended deny ip any any***

Para crear una sección en el *playbook* que configure una lista de control de acceso se considera como ejemplo el segundo comando mencionado: creación de un ACL para permitir el tráfico HTTP. En el ACL se incluye el nombre **acls**, tipo ***extended***, permitir el tráfico HTTP, protocolo **TCP** y dirección de destino con el protocolo **HTTP** o **www**. El resultado obtenido una vez se ejecute el *playbook* es*:* ***access-list acls extended permit tcp any host 192.168.100.3 eq www***. Siguiendo la misma plantilla se crean

otros ACLs que nieguen el tráfico ICMP e IP, a su vez permitan SMTP, POP y *domain*. La plantilla de los ACLs se muestra en la Figura 3.26.

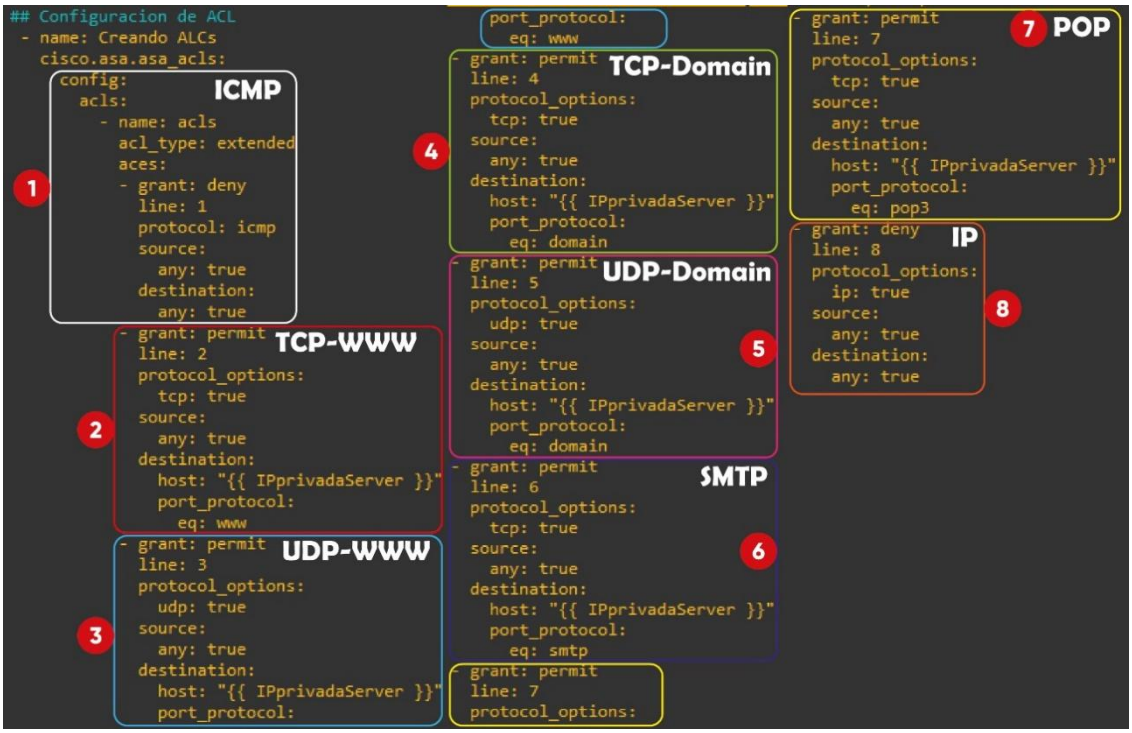

Figura 3.26 ACL con Ansible

Para automatizar la creación de ACLs se reutiliza la variable de la dirección IP privada del servidor de correo, denominada como ***{{IPprivadaServer}}***. La plantilla se muestra en la Figura 3.26. Además, para que la lista de ACL funcione, debe ser activada mediante el comando **access-group nombre_del_ACL in interface outside**. En el *playbook* se crea una nueva tarea que permita el funcionamiento del ACL para la zona *outside* como se muestra en la Figura 3.27.

```
## Habilitando el ACL
- name: Habilitando el ACLs
  asa_config:
    commands:
    - access-group acls in interface outside
```

Figura 3.27 Activar ACL en la zona *outside*

Finalmente, en el *playbook* se crea la última tarea para guardar las configuraciones realizadas en el dispositivo ASA, a través del comando ***write***. La Figura 3.28 muestra la parte final del algoritmo creado.

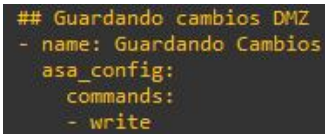

Figura 3.28 Función guardar

Creación del servidor de Correo

El servidor de correo utilizado para pruebas se crea en la máquina de Ubuntu Server 16.04, elegida por ser compatible con el paquete ***SquirrelMail***. La creación del servidor se resume en los siguientes pasos:

1. Para habilitar el sitio web del correo se instala apache2 (***sudo apt install apache2***).
2. Para habilitar el servicio de DNS se instala bind9 (***sudo apt install bind9***)
3. Para el servicio de correo se instala: mailutils (***sudo apt install mailutils***), postfix (***sudo apt install postfix***), pop (***sudo apt install courier-pop***), imap (***sudo apt install courier-imap***) y *SquirrelMail* (***sudo apt install squirrelmail***).
4. Configurar *SquirrelMail* indicando que se utilizara *courier* y el dominio lapc.com para las direcciones de correo.
5. Crear dos usuarios con dirección: ***cisco1@lapc.com*** y ***cisco2@lapc.com***. También, establecer la respectiva contraseña para ambos usuarios.
6. Habilitar los usuarios mediante el envío de un correo a partir del comando ***mail cisco1@lapc.com***.
7. Configurar *bind* ubicado en el directorio ***/etc/bind/***. Se crean dos archivos en el que se incluye datos del servidor para establecer su dominio: dirección IP privada 192.168.100.3, IP pública 10.10.10.20 y dominio del sitio web ***correolp.com***. En la Figura 3.29 se muestra las plantillas de configuración.
8. Reiniciar los servicios mediante el reinicio del equipo o el comando ***service nombre_del_servicio restart***.
9. Comprobar el funcionamiento del servidor mediante un navegador estilizado las direcciones IP y el dominio tal como se muestra en la Figura 3.30.

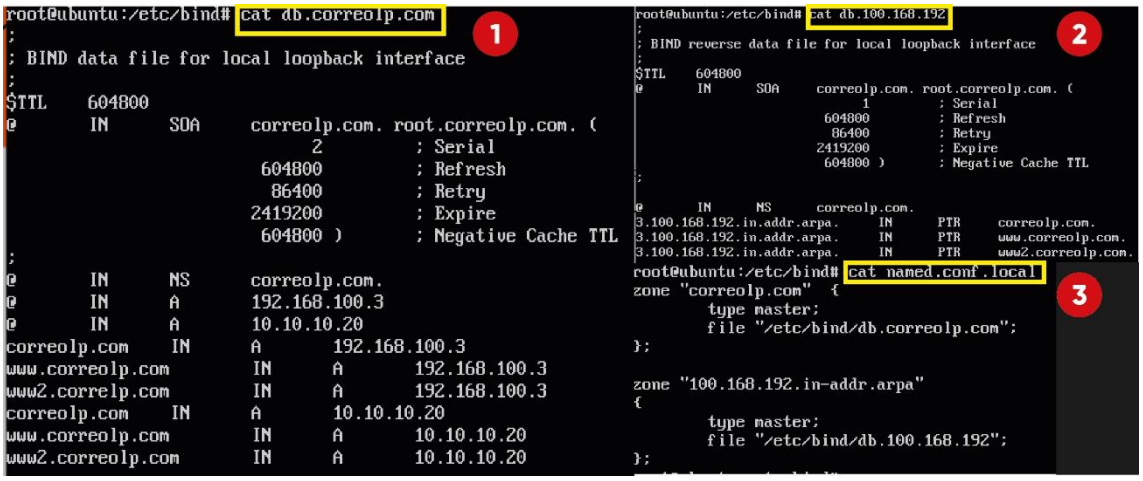

Figura 3.29 Archivos de *bind9*

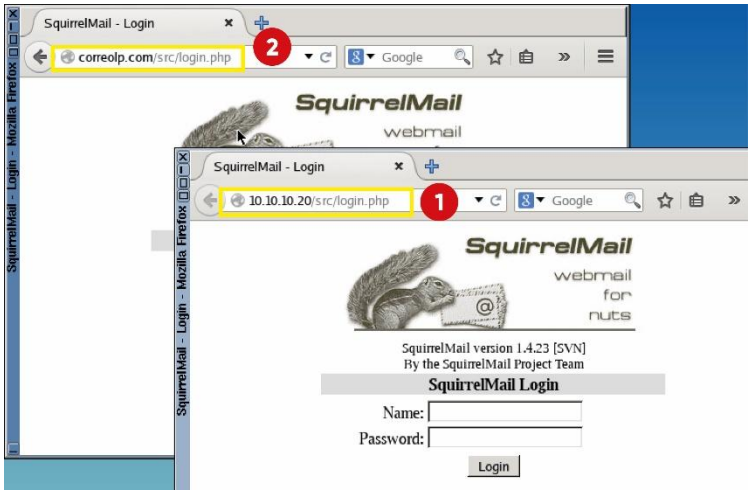

Figura 3.30 Conexión al servidor de correo

3.4 Verificación del funcionamiento del algoritmo

Al tener el *playbook* completo se realizan diferentes pruebas de funcionamiento que, al momento de ejecutarlo, indiquen que el dispositivo Cisco ASA se haya configurado correctamente utilizando Ansible.

Ejecución del *playbook* completo

Al ejecutar el *playbook* con variables en el nodo controlador se puede apreciar que se solicita datos al usuario para luego ir cumpliendo las tareas sin errores. En la Figura 3.31 y Figura 3.32 se puede apreciar la ejecución del *playbook*, en donde se muestra a cada tarea con su respectiva descripción de la configuración de las interfaces, NAT, DHCP, políticas y ACLs.

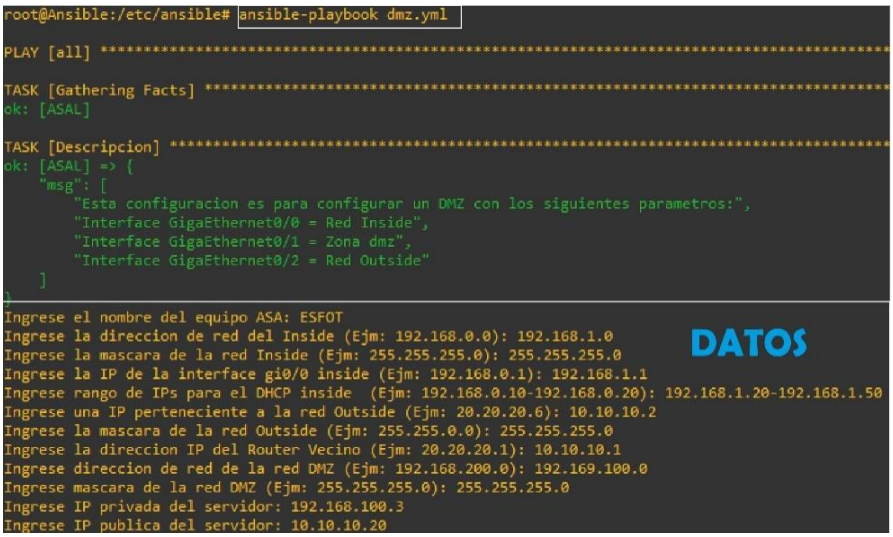

Figura 3.31 Ejecución del *playbook* parte 1

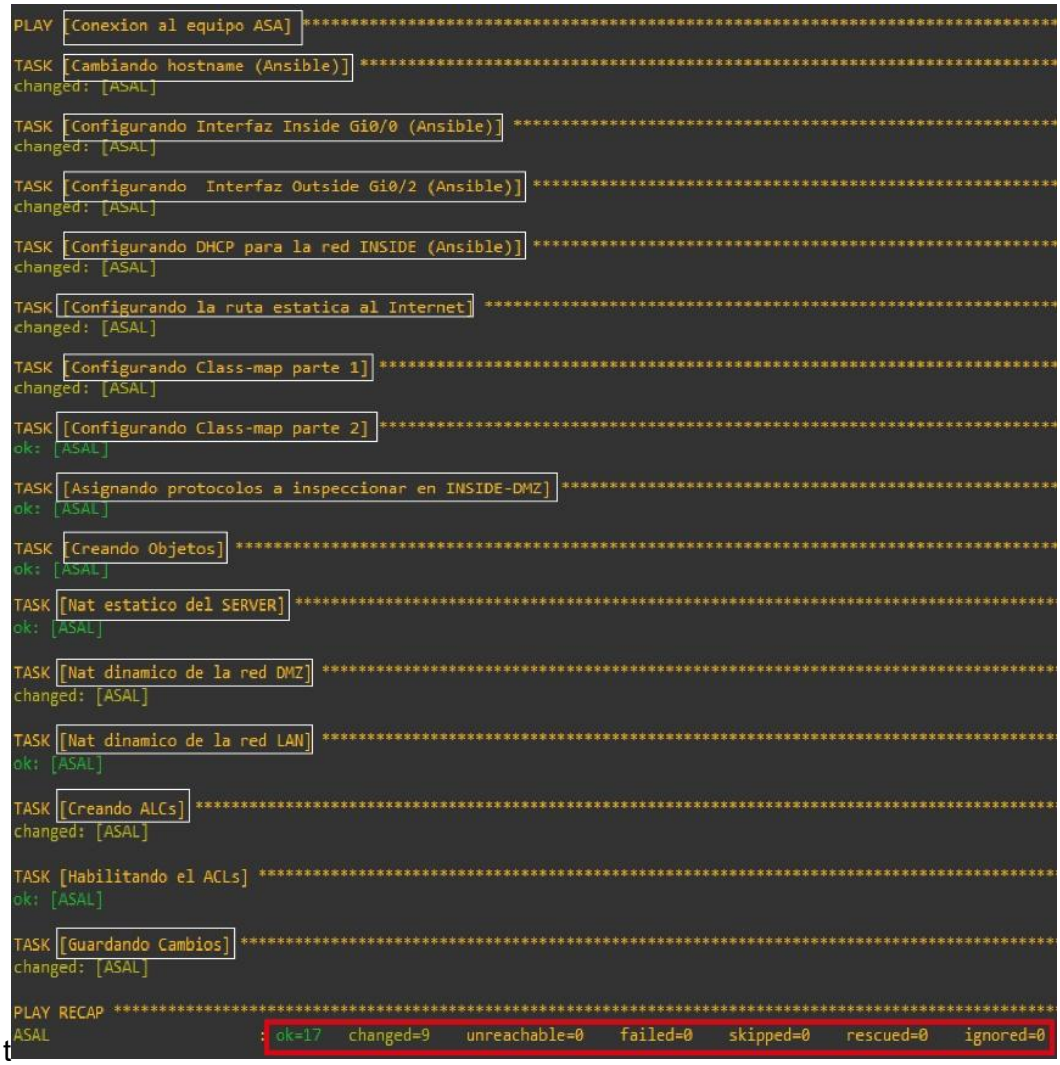

Figura 3.32 Ejecución del *playbook* parte 2

El resultado tras ejecutar el *playbook* completo se puede observar en el archivo de configuración de *running-config* del dispositivo Cisco ASA. Para poder visualizar las configuraciones realizadas se usa el comando ***show running-config***, Las distintas configuraciones se encuentran dispersas en el archivo de configuración del Cisco ASA, por tal razón en la Figura 3.33 se detalla solamente las configuraciones producidas con Ansible.

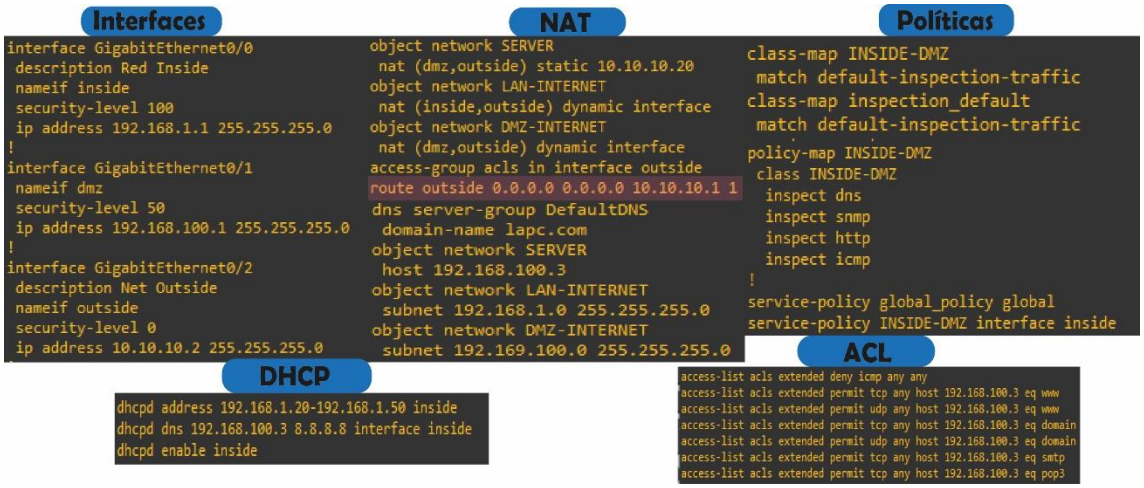

Figura 3.33 Archivo *running-config*

Comprobación entre la Zona *Inside* – DMZ

En la primera prueba se realiza dos peticiones ICMP, el primer ping se realiza desde un servidor de la zona DMZ hacia un servidor de la zona *inside*. Se obtiene una respuesta de host no encontrado o inalcanzable tal como se muestra en la Figura 3.34. Por otro lado, en el segundo ping es realizado desde un host de la zona *inside* un host la zona DMZ, como resultado se obtiene un ping correcto recibiendo respuesta del servidor como se muestra en la Figura 3.34. Con estos resultados obtenidos se verifica el correcto funcionamiento de la configuración del equipo ASA.

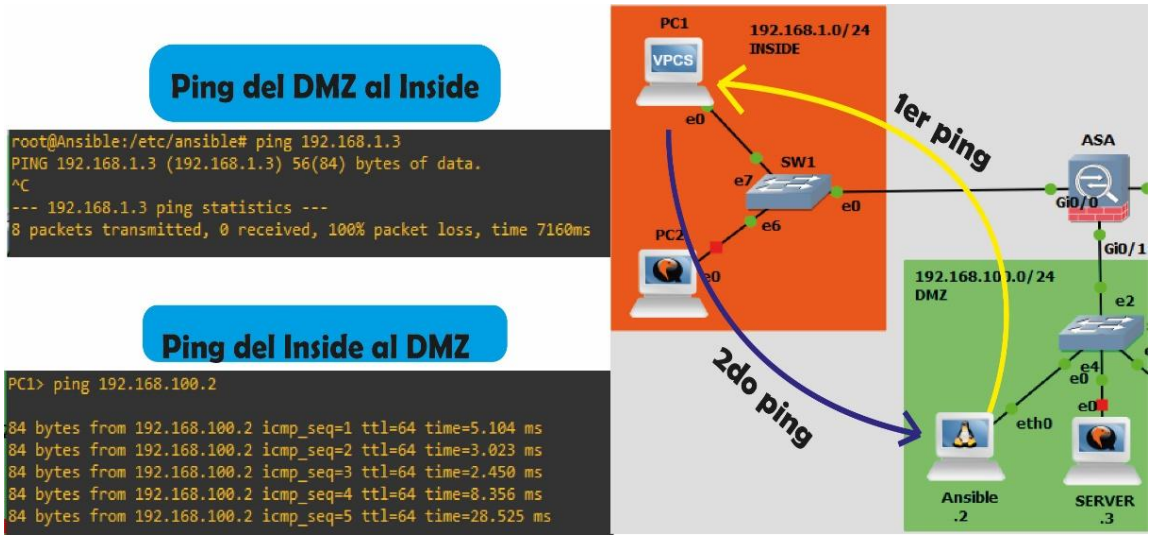

Figura 3.34 Prueba 1

Comprobación del NAT dinámico

La segunda prueba consiste en comprobar el NAT dinámico establecidos para las zonas DMZ e *inside*. Similar a la anterior prueba se realiza un ICMP a un sitio web indiferente desde un host de la zona inside y luego del DMZ. En la Figura 3.35 se muestra dos casos que indican cuando el NAT está bien o mal configurado. El primer caso se obtiene una respuesta del sitio web del Internet, a diferencia del segundo caso que se obtiene una respuesta errónea de host no accesible. Por tal razón, la configuración es correcta cuando se tenga respuesta del Internet.

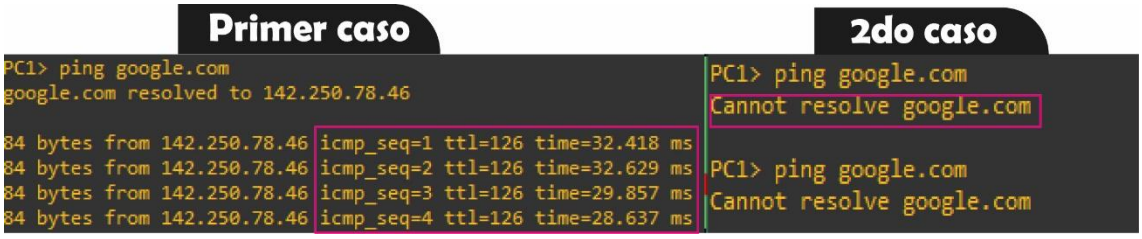

Figura 3.35 Prueba 2

Comprobación del NAT estático

En la tercera prueba se usa un host idealizado encontrado en el Internet. De la misma manera se utiliza el protocolo ICMP para comprobar la conexión al servidor alojado en la zona DMZ, en este caso es un servidor de correo con dirección IP privada 192.168.100.3 y IP pública 10.10.10.20. Además, es necesario que la lista de control de acceso no este activada, ya que interfiere con los resultados indicados a continuación.

En caso de tener activos el ACL, las reglas que se deben parar momentáneamente son:

- ***access-list acls extended deny icmp any any***
- ***access-list acls extended deny ip any any***

Para eliminar la regla al comienzo del comando se agrega **no**, en caso de desactivar las reglas se usa el comando ***no access-group acls in interface outside***. Esto necesariamente para realizar la prueba de funcionamiento del nat estático.

En la Figura 3.36 se presenta los dos casos usando la petición ICMP con la dirección IP privada del servidor y la pública. En el primer caso el servidor no da respuesta tras usar su dirección IP privada, a diferencia del segundo caso cuando se utiliza la dirección pública. Obteniendo estos resultados se indica que la configuración establecida es correcta, caso contrario al obtener resultados diferentes se entiende que la configuración está mal o que está activo el ACL.

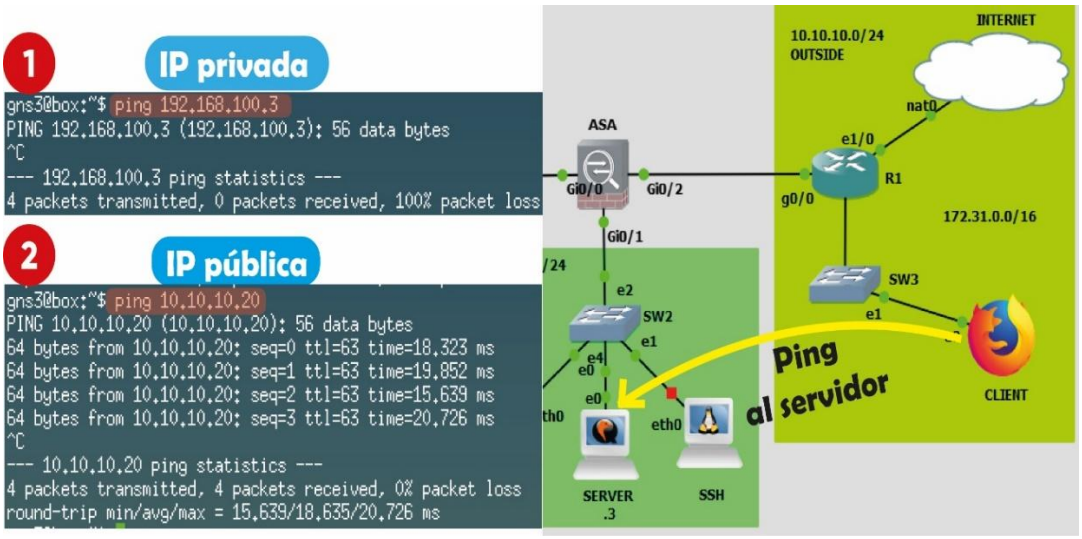

Figura 3.36 Prueba 3

Otra opción para evitar el uso del terminal y comprobar el funcionamiento del NAT estático del servidor es mediante el uso de un navegador, de tal manera que se muestre acceso al sitio web o servicio ofrecido por el servidor al utilizar la dirección IP pública.

Comprobación del ACL

En la última prueba se utiliza un navegador para comprobar el funcionamiento del servidor de correo. En la prueba se comprueba los siguientes aspectos:

- No se debe tener respuesta del servidor tras usar el protocolo ICMP con las direcciones IP privada y pública del servidor de correo.
- En el navegador se debe tener acceso a la interfaz de correo mediante la dirección IP pública del servidor.
- Acceso a la interfaz de correo utilizando una dirección de dominio.
- Envió de mensajes por parte del aplicativo de correo y comprobación del recibimiento del mismo.

De acuerdo a los aspectos mencionados se muestra en la Figura 3.37 y Figura 3.38 las pruebas del complimiento de los aspectos mencionados.

Figura 3.37 Prueba 4 parte 1

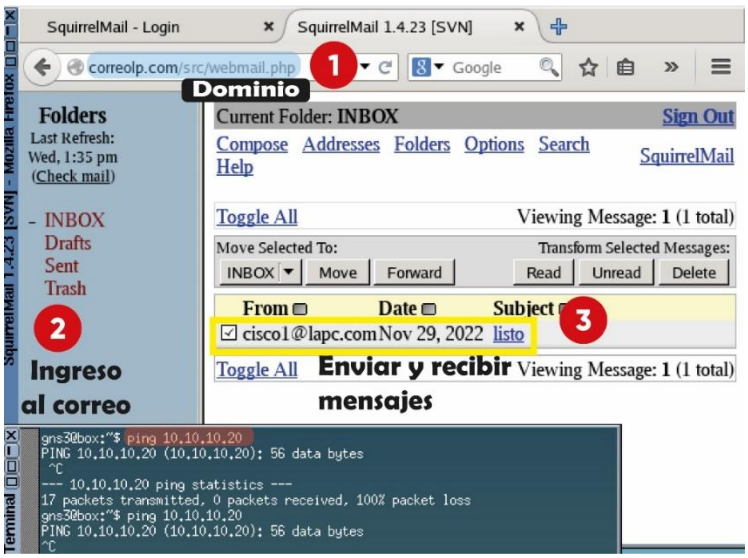

Figura 3.38 Prueba 4 parte 2

Al tener diferentes resultados a los mencionado se revisa las configuraciones ingresadas. Por ejemplo, el servidor DNS está configurado para traducir el dominio a la dirección IP 10.10.10.20, por lo tanto, al configurar NAT en el dispositivo ASA con una dirección IP pública distinta no se podrá acceder al correo por medio del dominio, solamente por dirección IP. De igual forma dependerá del ACL implementado, porque sí no está permitido el puerto para el uso de DNS no se podrá acceder de ninguna manera al servidor con un dominio. Al considerar estas posibles causas se puede corregir el error en caso de no ser por errores del *playbook* ejecutado.

4 CONCLUSIONES

- La investigación previa de las características y manejo de Ansible permitió reconocer los recursos necesarios para montar en entorno real en GNS3 que muestre la utilidad de Ansible al usarlo en un dispositivo Cisco ASA.
- Gracias al *software* de simulación de redes GNS3 se logró crear una red real que permita usar Ansible para la creación de una DMZ. En la red se incluye máquinas virtuales, el dispositivo ASA, switch, router, conexión a Internet, entre otros. Cada dispositivo con su propio software interno. Incluso se permitió la comprobación del funcionamiento del DMZ tras ser configurado con Ansible.
- SSH es el protocolo principal en la comunicación entre el Cisco ASA y el nodo Ansible, ya que permitió la conexión entre los equipos. Al usar Ansible se necesitó de otros parámetros de configuración SSH para evitar problemas de compatibilidad con el equipo ASA o errores al momento de intercambiar claves SSH. El algoritmo **diffie-hellman-group1-sh1** fue el que permitió corregir el error tras ser agregado en el archivo de configuración SSH perteneciente al nodo Ansible.
- El dispositivo Cisco ASA posee varias características de configuración que permitieron el levantamiento de un DMZ con características de NAT, DHCP, ACL, zonas *inside*, DMZ, *outside*, y distintos comportamientos de las zonas para mayor seguridad. Además, el equipo funcionó correctamente con Ansible al configurar SSH solamente habilitado para el host administrador.
- Al trabajar con Ansible se observó que la configuración de un DMZ especificada en el *playbook* es sencilla y fácil de comprender. Con la inclusión de distintas tareas se tuvo mayor entendimiento de como levantar un DMZ. Además, tras incluir variables, el proceso se vuelve dinámico ya que permitió al

usuario el ingreso de los datos Manualmente sin tener que escribir los comandos.

- Utilizando el módulo cisco.asa se consiguió un mejor manejo de Ansible para crear una plantilla propia dedicada al levantamiento de un DMZ. El módulo fue utilizado para las distintas tareas creadas para añadir protocolos NAT, ACL, DCHP y configuraciones de interfaces con sus respectivos atributos. Las diferentes tareas fueron ordenadas de tal manera que refleje los pasos primordiales para el levantamiento de un DMZ.
- El módulo cisco.asa de la nueva versión tuvo deficiencias al intentar aplicar los diferentes parámetros de configuración. Este caso fue una complicación durante el proceso de creación de la plantilla, ya que módulo cisco.asa tenía nuevas funciones o que fueron retiradas. Además, el módulo presentaba pocos parámetros para establecer las configuraciones del DMZ a diferencia de otros módulos como el cisco.ios. Para evitar este inconveniente se procedió con el uso de una versión anterior. Además, la mayor parte de la plantilla incluye comandos propios para configurar el dispositivo cisco ASA.
- Al ejecutar la plantilla en el nodo Ansible se observó que en pocos segundos las tareas se iban ejecutando sin ningún problema para el caso del *playbook* sin variables. Los mismo ocurría cuando el *playbook* manejaba variables, pero antes de completar las tareas se solicitaba datos al usuario. Para ambos casos se logró configurar el dispositivo cisco ASA en el menor tiempo posible, comprobado tras visualizar el archivo de configuración del aparato ASA.
- Los hosts creados en la topología de GNS3 permitieron comprobar la funcionalidad del equipo ASA tras ser configurado con Ansible. Las diferentes pruebas se llevaron a cabo en los hosts de las tres zonas que permitieron arreglar el *playbook* en caso de mostrar errores de configuración. El resultado de las distintas pruebas permitió comprobar que el *playbook* funcione correctamente y en caso de presentar problemas conocer las posibles causas.
- Ansible mostró un rendimiento excelente para configurar los dispositivos de red Cisco ASA. Los resultados muestran a Ansible como una herramienta de uso sencillo y comprensible para el usuario que tiene pocos conocimientos del tema. Incluso se tiene una mejor perspectiva de la configuración de equipos junto con su procedimiento reflejado en las tareas.
- Ansible muestra errores solamente cuando el inventario tiene los datos del dispositivo son erróneos, el *playbook* incluye algún parámetro que no es compatible con el módulo o el SSH está mal configurado. Tomando en

consideración estos errores habituales se puede dar soluciones a los usuarios que no conocen del error.

5 RECOMENDACIONES

- La seguridad es un parámetro esencial para los dispositivos *networking*. Por eso al momento de habilitar el protocolo SSH es necesario limitar el acceso al dispositivo Cisco ASA, indicando solamente la dirección IP de los hosts o de la red.
- Comprobar que la versión del dispositivo Cisco ASA sea compatible con Ansible. Para más información de compatibilidades puede encontrarse en el repositorio de Ansible.
- Para poder usar Ansible con los dispositivos Cisco ASA hay que configurar los archivos SSH del repositorio **/etc/ssh/config_ssh**, agregando algoritmos al archivo para el intercambio de llaves compatibles con los dispositivos de red.
- Agregar en el inventario de Ansible los parámetros necesarios para que pueda establecerse la conexión en los equipos administrados. Los parámetros difieren de acuerdo al tipo de equipo. Para el caso de Cisco ASA se requiere de los parámetros indicados en el documento. Para verificar que se tenga acceso con Ansible y los datos del inventario estén correctos se puede realizar la prueba **Ping Pong**.
- Tomar en consideración la versión del módulo a descargar de la colección de Ansible, ya que pueden tener nuevas funcionalidades o se hayan eliminado. Para el caso del módulo cisco.asa la versión actualizada no incluye funciones para establecer la lista de control de acceso ingresando parámetros, solamente con comandos. Pero en su versión anterior todavía cuenta con la funcionalidad de ACL.
- Los errores comunes al momento de crear un *playbook* se presentan en el formato, uso de módulos no compatibles con el dispositivo y parámetros mal

utilizados. Por eso es necesario revisar el *playbook* conjuntamente con su repositorio web y ejemplos para evitar estos problemas.

- Al usar un *playbook* con variables se debe considerar que dichas variables no deben repetirse. En cambio, cuando el usuario ingrese datos cuando se haya ejecutado el *playbook*, se considera el ingreso correcto de los datos, ya que en su ejecución las tareas pueden presentar de errores.

6 REFERENCIAS BIBLIOGRÁFICAS

[1] Anónimo, «RedHat,» 10 Mayo 2022. [En línea]. Available: https://www.redhat.com/es/topics/devops. [Último acceso: 18 Diciembre 2022].

[2] G. M. Jiménez , Análisis de incidentes en el ámbito TIC con DevOps, Catalunya: Universidad Politécnica de Catalunya, 2016.

[3] J. Angulo, «Autentia,» 17 Agosto 2018. [En línea]. Available: https://www.autentia.com/2018/08/17/entendiendo-devops-en-5-minutos/. [Último acceso: 18 Diciembre 2022].

[4] P. Parada, «IeBS,» 9 Diciembre 2022. [En línea]. Available: https://www.iebschool.com/blog/devops-ventajas-beneficios-agile-scrum/. [Último acceso: 18 Diciembre 2022].

[5] Anónimo, «Cloudflare,» 2022 Abril 15. [En línea]. Available: https://www.cloudflare.com/es-es/learning/network-layer/network-security/. [Último acceso: 2022 Diciembre 18].

[6] O. Santos y J. Frahim, Cisco ASA: All-in-One Firewall, IPS, Anti-X, and VPN Adaptive Security Appliance, Segunda ed., EE.UU.: CiscoPress, 2010.

[7] L. Faelin, «Hackmd,» 29 Julio 2022. [En línea]. Available: https://hackmd.io/@Faelin/cisco-ios-reference-asa-firewall-devices. [Último acceso: 18 Diciembre 2022].

[8] M. «CCNA Academy,» 7 Diciembre 2022. [En línea]. Available: https://www.ccnaacademy.com/2022/12/what-is-asa-firewall-security-zones-how.html. [Último acceso: 19 Diciembre 2022].

[9] Anínimo, «W0lff4ng,» 14 Mayo 2020. [En línea]. Available: https://www.w0lff4ng.org/configuracion-basica-cisco-asa/#:~:text=El%20security%20level%20indica%20que,valor%20entre%200%20y%20100.. [Último acceso: 19 Diciembre 2022].

[10] Cisco, «cisco.com,» 1 Marzo 2022. [En línea]. Available: https://www.cisco.com/c/en/us/td/docs/security/asa/asa917/configuration/general/asa-917-general-config.pdf. [Último acceso: 17 Diciembre 2022].

[11] J. Geerling, Ansible for DevOps: Server and Configuration Management for Humans, 1 ed., M. Newman y K. Geerling, Edits., Leanpub, 2020.

[12] J. Edelman, S. S. Lowe y M. Oswalt, Network Programmability and Automation, 1 ed., V. Wilson y C. Allen, Edits., EE.UU.: O'Reilly media, 2018.

[13] J. e. Endelman, Network Automation with Ansible, primera ed., EE. UU.: O'Reilly Media, 2016.

[14] W. Irtaza, IT Infrastructure Automation Using Ansible, 1ra ed., India: BPB Publication, 2022.

[15] B. Ghimire, «GEEKFLARE,» 11 Enero 2021. [En línea]. Available: https://geekflare.com/es/yaml-introduction/. [Último acceso: 17 Diciembre 2022].

[16] J. Wijaya, «Network Automation using Ansible for Cisco Routers Basic Configuration,» Bandung Institute of Technology, Bandung, 2018.

[17] F. Mohd , A. Khairunnisa , H. Iman y R. Rafiza , *Network Automation using Ansible for EIGRP Network,* Malaysia: JCRINN, 2021.

[18] L. Agapidis, «Networks Training,» 9 Noviembre 2022. [En línea]. Available: https://www.networkstraining.com/gns3-vs-eve-ng-vs-cisco-packet-tracer/. [Último acceso: 19 Diciembre 2022].

[19] J. Jiménez, «Redes Zone,» 11 Marzo 2022. [En línea]. Available: https://www.redeszone.net/noticias/power/calefaccion-suelo-radiante-electrico-

agua-cual-mejor/. [Último acceso: 19 Diciembre 2022].

[20] «Pnetlab,» [En línea]. Available: https://pnetlab.com/pages/documentation?slug=hardware-requirements. [Último acceso: 20 Diciembre 2022].

[21] k. Sims, «Let Me Tech You,» 4 Enero 2022. [En línea]. Available: https://letmetechyou.com/eve-ng-vs-gns3-complete-2022-review/. [Último acceso: 19 Diciembre 2022].

[22] G. González , «Uvadoc,» 2022. [En línea]. Available: https://uvadoc.uva.es/bitstream/handle/10324/57310/TFG-G5812.pdf?sequence=1&isAllowed=y. [Último acceso: 19 Diciembre 2022].

7 ANEXOS

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 3 de marzo de 2023

De mi consideración:

Yo, FERNANDO VINICIO BECERRA CAMACHO, en calidad de Director del Trabajo de Integración Curricular titulado CREACIÓN DE DMZ EN EQUIPOS ESPECIALIZADOS DE RED MEDIANTE DEVOPS elaborado por el estudiante LUIS ALFONSO PADILLA CAMALLEde la carrera en TECNÓLOGÍA 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 14%.

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.

https://epnecuador-my.sharepoint.com/:b:/g/personal/fernando_becerrac_epn_edu_ec/EWJ2wDxSOQVCnFvHoSDMqN0ByFMpNHhjXtZBEc4E61h-GQ?e=wMNlO5

Atentamente,

Fernando Vinicio Becerra Camacho

Docente

Escuela de Formación de Tecnólogos

ANEXO II: ENLACE DEL VIDEO DE LA IMPLEMENTACIÓN DEL ALGORITMO Y PRUEBAS DE VERIFICACIÓN

El video demostrativo se encuentra en el enlace *https://youtu.be/l52Gj-52GEM* o puede tener acceso mediante el código QR.

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

ANEXO III: *PLAYBOOK* DE ANSIBLE COMPLETO

Playbook 1: Comprende las configuraciones normales de una DMZ, pero sin utilizar variables.

```
---
# Configuración para la conexión al ASA
- name: Conexion al equipo ASA
  connection: network_cli
  hosts: ASAL
  gather_facts: false
  become: yes
  tasks:
  ### Cambiar el Hostname del Equipo ASA
  - name: Cambiando hostname (Ansible)
    cisco.asa.asa_config:
      commands:
      - hostname ESFOT (ESCUELA DE FORMACIÓN DE TECNÓLOGOS)
  ### Configuración de Interfaces
  - name: Configurando Interfaz Inside Gi0/0 (Ansible)
    cisco.asa.asa_config:
      lines:
      - description Red Inside
      - no shutdown
      - nameif inside
      - security-level 100
      - ip address 192.168.100.1.1 255.255.255.0
      parents: [interface GigabitEthernet0/0]
    register: interface
```

```
- name: Configurando  Interfaz Outside Gi0/2 (Ansible)
  cisco.asa.asa_config:
    lines:
    - description Net Outside
    - no shutdown
    - nameif outside
    - security-level 0
    - ip address 10.10.10.2 255.255.255.0
    parents: [interface GigabitEthernet0/2]
  register: interface
### Configuración DHCP Inside
- name: Configurando DHCP para la red INSIDE (Ansible)
  asa_config:
    commands:
    - dhcpd address 192.168.1.30-192.168.1.60 inside
    - dhcpd enable inside
    - dhcpd dns 192.168.100.3 8.8.8.8 interface inside
## Ruta estática al Internet
- name: Configurando la ruta estática al Internet
  asa_config:
    commands:
    - route outside 0.0.0.0 0.0.0.0 10.10.10.1
## Configuración Class_MAP DMZ-INSIDE
- name: Configurando Class-map parte 1
  asa_config:
    commands:
    - class-map INSIDE-DMZ
```

```
    - policy-map INSIDE-DMZ
    - class INSIDE-DMZ
    - service-policy INSIDE-DMZ interface inside
- name: Configurando Class-map parte 2
  asa_config:
    lines:
    - match default-inspection-traffic
    parents: [class-map INSIDE-DMZ]
  register: result
- name: Asignando protocolos a inspeccionar en INSIDE-DMZ
  asa_config:
    lines:
    - inspect icmp
    - inspect dns
    - inspect snmp
    - inspect http
    parents: ["policy-map INSIDE-DMZ", "class INSIDE-DMZ"]
  register: result2
## Configuración NAT
# Configurando objetos Network
- name: Creando Objetos
  asa_config:
    commands:
    - object network SERVER
    - object network DMZ-INTERNET
    - object network LAN-INTERNET
# Nat estático del servidor
```

```
 - name: Nat estático del SERVER
   asa_config:
     lines:
     - host 192.168.100.3
     - nat (dmz,outside) static 10.10.10.20
     parents: [object network SERVER]
   register: object1
 # NAT dinámico para la red DMZ
 - name: Nat dinámico de la red DMZ
   asa_config:
     lines:
     - subnet 192.168.100.0 255.255.255.0
     - nat (dmz,outside) dynamic interface
     parents: [object network DMZ-INTERNET]
   register: object2
 # NAT dinámico para la red LAN
 - name: Nat dinámico de la red LAN
   asa_config:
     lines:
     - subnet 192.168.1.0 255.255.255.0
     - nat (inside,outside) dynamic interface
     parents: [object network LAN-INTERNET]
   register: object3
## Configuración de ACL
 - name: Creando ALCs
   cisco.asa.asa_acls:
     config:
```

```
acls:
  - name: acls
    acl_type: extended
    aces:
    - grant: deny
      line: 1
      protocol: icmp
      source:
        any: true
      destination:
        any: true
    - grant: permit
      line: 2
      protocol_options:
        tcp: true
      source:
        any: true
      destination:
        host: 192.168.100.3
        port_protocol:
          eq: www
    - grant: permit
      line: 3
      protocol_options:
        udp: true
      source:
        any: true
```

```
  destination:
    host: 192.168.100.3
    port_protocol:
      eq: www
- grant: permit
  line: 4
  protocol_options:
    tcp: true
  source:
    any: true
  destination:
    host: 192.168.100.3
    port_protocol:
      eq: domain
- grant: permit
  line: 5
  protocol_options:
    udp: true
  source:
    any: true
  destination:
    host: 192.168.100.3
    port_protocol:
      eq: domain
- grant: permit
  line: 6
  protocol_options:
```

```
      tcp: true
    source:
      any: true
    destination:
      host: 192.168.100.3
      port_protocol:
        eq: smtp
  - grant: permit
    line: 7
    protocol_options:
      tcp: true
    source:
      any: true
    destination:
      host: 192.168.100.3
      port_protocol:
        eq: pop3
  - grant: deny
    line: 8
    protocol_options:
      ip: true
    source:
      any: true
    destination:
      any: true
## Habilitando el ACL
- name: Habilitando el ACLs
```

```
    asa_config:
      commands:
      - access-group acls in interface outside
  ## Guardando cambios DMZ
  - name: Guardando Cambios
    asa_config:
      commands:
      - write
```

Playbook 2: Comprende las configuraciones de una DMZ incluyendo variables para un proceso dinámico.

```
---
  ## Configuración DMZ y ACLS (Access Control List) en equipos ASA con Ansible
  # Nombre: Luis Padilla
  # Carrera: Tecnología en redes y telecomunicaciones
# Descripción del contenido del playbook
- hosts: all
  tasks:
  - name: Descripción
    debug:
      msg:
      - 'Esta configuración es para configurar un DMZ con los siguientes paramet      ros:'
      - 'Interface GigaEthernet0/0 = Red Inside'
      - 'Interface GigaEthernet0/1 = Zona dmz'
      - 'Interface GigaEthernet0/2 = Red Outside'
# Configuracion para la conexión al ASA
- name: Conexion al equipo ASA
  connection: network_cli
```

```
 hosts: ASAL
 gather_facts: false
 become: yes
## Valores principales a cambiar en el ASA
 vars_prompt:
 - name: Hostname
   prompt: 'Ingrese el nombre del equipo ASA'
   private: no
 # Datos de la red inside
 - name: InsideRed
   prompt: 'Ingrese la direccion de red del Inside (Ejm: 192.168.0.0)'
   private: no
 - name: InsideMasc
   prompt: 'Ingrese la máscara de la red Inside (Ejm: 255.255.255.0)'
   private: no
 - name: InsideIP
   prompt: 'Ingrese la IP de la interface gi0/0 inside (Ejm: 192.168.0.1)'
   private: no
 - name: InsideDHCP
   prompt: 'Ingrese rango de IP para el DHCP inside  (Ejm: 192.168.0.10-192.16
8.0.20)'
   private: no
  # Datos de la red Outside
 - name: OutsideIP
   prompt: 'Ingrese una IP perteneciente a la red Outside (Ejm: 20.20.20.6)'
   private: no
 - name: OutsideMasc
```

```
  prompt: 'Ingrese la máscara de la red Outside (Ejm: 255.255.0.0)'
  private: no
- name: IPvecino
  prompt: 'Ingrese la direccion IP del Router Vecino (Ejm: 20.20.20.1)'
  private: no
 # Datos de la red DMZ
- name: dmzRED
  prompt: 'Ingrese direccion de red de la red DMZ (Ejm: 192.168.200.0)'
  private: no
- name: dmzMasc
  prompt: 'Ingrese mascara de la red DMZ (Ejm: 255.255.255.0)'
  private: no
 #Datos para el NAT
- name: IPprivadaServer
  prompt: 'Ingrese IP privada del servidor'
  private: no
- name: IPpublicServer
  prompt: 'Ingrese IP pública del servidor'
  private: no
tasks:
### Cambiar el Hostname del Equipo ASA
- name: Cambiando hostname (Ansible)
  cisco.asa.asa_config:
    commands:
    - hostname "{{Hostname}}"


### Configuracion de Interfaces
```

```
- name: Configurando Interfaz Inside Gi0/0 (Ansible)
  cisco.asa.asa_config:
    lines:
    - description Red Inside
    - no shutdown
    - nameif inside
    - security-level 100
    - ip address {{InsideIP}} {{InsideMasc}}
    parents: [interface GigabitEthernet0/0]
  register: interface
- name: Configurando  Interfaz Outside Gi0/2 (Ansible)
  cisco.asa.asa_config:
    lines:
    - description Net Outside
    - no shutdown
    - nameif outside
    - security-level 0
    - ip address {{OutsideIP}} {{OutsideMasc}}
    parents: [interface GigabitEthernet0/2]
  register: interface
### Configuracion DHCP Inside
- name: Configurando DHCP para la red INSIDE (Ansible)
  asa_config:
    commands:
    - dhcpd address {{InsideDHCP}} inside
    - dhcpd enable inside
    - dhcpd dns 192.168.100.3 8.8.8.8 interface inside
```

```
## Ruta estática al Internet
- name: Configurando la ruta estática al Internet
  asa_config:
    commands:
    - route outside 0.0.0.0 0.0.0.0 {{IPvecino}}
## Configuracion Class_MAP DMZ-INSIDE
- name: Configurando Class-map parte 1
  asa_config:
    commands:
    - class-map INSIDE-DMZ
    - policy-map INSIDE-DMZ
    - class INSIDE-DMZ
    - service-policy INSIDE-DMZ interface inside
- name: Configurando Class-map parte 2
  asa_config:
    lines:
    - match default-inspection-traffic
    parents: [class-map INSIDE-DMZ]
  register: result
- name: Asignando protocolos a inspeccionar en INSIDE-DMZ
  asa_config:
    lines:
    - inspect icmp
    - inspect dns
    - inspect snmp
    - inspect http
    parents: ["policy-map INSIDE-DMZ", "class INSIDE-DMZ"]
```

```
  register: result2
## Configuracion NAT
# Configurando objetos Network
- name: Creando Objetos
  asa_config:
    commands:
    - object network SERVER
    - object network DMZ-INTERNET
    - object network LAN-INTERNET
# Nat estático del servidor
- name: Nat estático del SERVER
  asa_config:
    lines:
    - host {{IPprivadaServer}}
    - nat (dmz,outside) static {{IPpublicServer}}
    parents: [object network SERVER]
  register: object1
# NAT dinámico para la red DMZ
- name: Nat dinámico de la red DMZ
  asa_config:
    lines:
    - subnet {{dmzRED}} {{dmzMasc}}
    - nat (dmz,outside) dynamic interface
    parents: [object network DMZ-INTERNET]
  register: object2
# NAT dinámico para la red LAN
- name: Nat dinámico de la red LAN
```

```
  asa_config:
    lines:
    - subnet {{InsideRed}} {{InsideMasc}}
    - nat (inside,outside) dynamic interface
    parents: [object network LAN-INTERNET]
  register: object3
## Configuracion de ACL
 - name: Creando ALCs
  cisco.asa.asa_acls:
    config:
      acls:
        - name: acls
          acl_type: extended
          aces:
          - grant: deny
            line: 1
            protocol: icmp
            source:
              any: true
            destination:
              any: true
          - grant: permit
            line: 2
            protocol_options:
              tcp: true
            source:
              any: true
```

```
  destination:
    host: "{{ IPprivadaServer }}"
    port_protocol:
      eq: www
- grant: permit
  line: 3
  protocol_options:
    udp: true
  source:
    any: true
  destination:
    host: "{{ IPprivadaServer }}"
    port_protocol:
      eq: www
- grant: permit
  line: 4
  protocol_options:
    tcp: true
  source:
    any: true
  destination:
    host: "{{ IPprivadaServer }}"
    port_protocol:
      eq: domain
- grant: permit
  line: 5
  protocol_options:
```

```
    udp: true
  source:
    any: true
  destination:
    host: "{{ IPprivadaServer }}"
    port_protocol:
      eq: domain
- grant: permit
  line: 6
  protocol_options:
    tcp: true
  source:
    any: true
  destination:
    host: "{{ IPprivadaServer }}"
    port_protocol:
      eq: smtp
- grant: permit
  line: 7
  protocol_options:
    tcp: true
  source:
    any: true
  destination:
    host: "{{ IPprivadaServer }}"
    port_protocol:
      eq: pop3
```

```
        - grant: deny
          line: 8
          protocol_options:
            ip: true
          source:
            any: true
          destination:
            any: true
  ## Habilitando el ACL
  - name: Habilitando el ACLs
    asa_config:
      commands:
      - access-group acls in interface outside
  ## Guardando cambios DMZ
  - name: Guardando Cambios
    asa_config:
      commands:
      - write
```

ANEXO IV: Archivo *running-config* del dispositivo ASA

```
ESFOT# show running
: Saved
```

:

: Serial Number: 9AKBLBF9P7K

: Hardware: ASAv, 2048 MB RAM, CPU Pentium II 2600 MHz

:

ASA Version 9.6(1)5

!

hostname ESFOT

domain-name lapc.com

enable password iying0ocgx4ZIzLu encrypted

xlate per-session deny tcp any4 any4

xlate per-session deny tcp any4 any6

xlate per-session deny tcp any6 any4

xlate per-session deny tcp any6 any6

xlate per-session deny udp any4 any4 eq domain

xlate per-session deny udp any4 any6 eq domain

xlate per-session deny udp any6 any4 eq domain

xlate per-session deny udp any6 any6 eq domain

names

!

interface GigabitEthernet0/0

 description Red Inside

 nameif inside

 security-level 100

 ip address 192.168.1.1 255.255.255.0

!

interface GigabitEthernet0/1

```
 nameif dmz
 security-level 50
 ip address 192.168.100.1 255.255.255.0
!
interface GigabitEthernet0/2
 description Net Outside
 nameif outside
 security-level 0
 ip address 10.10.10.2 255.255.255.0
!
interface GigabitEthernet0/3
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/4
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet0/5
 shutdown
 no nameif
 no security-level
 no ip address
```

<--- More --->

Warning: ASAv platform license state is Unlicensed.

Install ASAv platform license for full functionality.

!

interface GigabitEthernet0/6

 shutdown

 no nameif

 no security-level

 no ip address

!

interface Management0/0

 management-only

 shutdown

 no nameif

 no security-level

 no ip address

!

ftp mode passive

dns server-group DefaultDNS

 domain-name lapc.com

object network SERVER

 host 192.168.100.3

object network LAN-INTERNET

 subnet 192.168.1.0 255.255.255.0

object network DMZ-INTERNET

 subnet 192.169.100.0 255.255.255.0

access-list acls extended deny icmp any any

```
access-list acls extended permit tcp any host 192.168.100.3 eq www
access-list acls extended permit udp any host 192.168.100.3 eq www
access-list acls extended permit tcp any host 192.168.100.3 eq domain
access-list acls extended permit udp any host 192.168.100.3 eq domain
access-list acls extended permit tcp any host 192.168.100.3 eq smtp
access-list acls extended permit tcp any host 192.168.100.3 eq pop3
access-list acls extended deny ip any any
pager lines 23
mtu inside 1500
mtu dmz 1500
mtu outside 1500
no failover
no monitor-interface service-module
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
no arp permit-nonconnected
!
object network SERVER
 nat (dmz,outside) static 10.10.10.20
object network LAN-INTERNET
 nat (inside,outside) dynamic interface
object network DMZ-INTERNET
 nat (dmz,outside) dynamic interface
access-group acls in interface outside
route outside 0.0.0.0 0.0.0.0 10.10.10.1 1
timeout xlate 3:00:00
```

```
timeout pat-xlate 0:00:30
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 sctp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
user-identity default-domain LOCAL
aaa authentication ssh console LOCAL
no snmp-server location
no snmp-server contact
crypto ipsec security-association pmtu-aging infinite
crypto ca trustpoint _SmartCallHome_ServerCA
 no validation-usage
 crl configure
crypto ca trustpool policy
 auto-import
crypto ca certificate chain _SmartCallHome_ServerCA
 quit
telnet timeout 5
ssh stricthostkeycheck
ssh 192.168.100.2 255.255.255.255 dmz
ssh timeout 5
ssh version 2
ssh key-exchange group dh-group1-sha1
console timeout 0
dhcpd dns 8.8.8.8
```

```
!
dhcpd address 192.168.1.30-192.168.1.60 inside
dhcpd dns 192.168.100.3 8.8.8.8 interface inside
dhcpd enable inside
!
dynamic-access-policy-record DfltAccessPolicy
username ansible password /56MnqWh29myV1yG encrypted privilege 15
!
class-map INSIDE-DMZ
 match default-inspection-traffic
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns migrated_dns_map_1
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map global_policy
```