ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE SERVICIOS DE NETWORKING MEDIANTE DEVOPS

IMPLEMENTACIÓN DE UN SERVIDOR SNMP V3 CON DEVOPS PARA LA ADMINISTRACIÓN DE EQUIPOS DE NETWORKING

TRABAJO DE INTEGRACIÓN CURRICULAR

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

MATEO DAVID CASTRO CAZAR

mateo.castro@epn.edu.ec

DIRECTOR: FERNANDO VINICIO BECERRA CAMACHO

fernando.becerra@epn.edu.ec

DMQ (Distrito Metropolitano de Quito) febrero 2024

CERTIFICACIONES

Yo, MATEO DAVID CASTRO CAZAR 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.

MATEO DAVID CASTRO CAZAR

mateo.castro@epn.edu.ec

mateo.d.c.c@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por MATEO DAVID CASTRO CAZAR, bajo mi supervisión.

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.

MATEO DAVID CASTRO CAZAR

ÍNDICE DE CONTENIDOS

RESUMEN

El objetivo del presente proyecto es implementar el protocolo SNMPv3 (Simple Management Protocol Versión 3) utilizando DevOps, donde el servidor SNMP tiene el objetivo de monitorear todas y cada una de las maquinas host, routers y switches qué forman parte de una red específica, para este propósito el software que será utilizado para realizar el monitoreo de la red es Nagios Core, y el encargado de hacer la automatización de los diferentes pasos que serán tomados para la instalación e implementación de Nagios Core es Ansible DevOps.

En el primer capítulo se describe cual es el objetivo general del proyecto y abarca de igual forma los objetivos específicos a ser cumplidos en los capítulos posteriores, además, se exponen definiciones detalladas de los elementos que conforman el proyecto de integración curricular.

En el segundo capítulo se explica de manera breve la metodología que aplica, describe de forma general todos los objetivos específicos mencionados en el capítulo posterior con información adicional sobre los mismos.

En el tercer capítulo se exponen los resultados, donde, de manera detallada y especifica se exponen cada uno de los objetivos con las respectivas soluciones a los objetivos planteados en el primer capítulo y en el segundo capítulo.

Para finalizar en el cuarto capítulo abarcan las conclusiones donde se da un análisis de los resultados obtenidos. Evaluando el impacto de la solución en los ámbitos disciplinar-investigativo, social o laboral-profesional.

PALABRAS CLAVE: Protocolo SNMP, Nagios Core, Ansible, DevOps.

ABSTRACT

The objective of this project is to implement the SNMPv3 protocol (Simple Management Protocol Version 3) using DevOps, where the SNMP server has the objective of monitoring each and every one of the host machines, routers and switches that are part of a specific network, for this purpose the software that will be used to monitor the network is Nagios Core, and the one in charge of automating the different steps that will be taken for the installation and implementation of Nagios Core is Ansible DevOps.

The first chapter describes the general objective of the project and also covers the specific objectives to be met in the following chapters, as well as detailed definitions of the elements that make up the curriculum integration project.

The second chapter briefly explains the methodology applied, describes in general terms all the specific objectives mentioned in the following chapter with additional information about them.

The third chapter presents the results, where, in a detailed and specific manner, each one of the objectives is presented with the respective solutions to the objectives stated in the first chapter and in the second chapter.

Finally, the fourth chapter includes the conclusions where an analysis of the results obtained is given. Evaluating the impact of the solution in the disciplinary-research, social or labor-professional fields.

KEYWORDS: *SNMP protocol, Nagios Core, Ansible, DevOps*

1 DESCRIPCIÓN

El despliegue de un servidor SNMPv3 con DevOps puede ser una clave fundamental en las redes de la actualidad, una red cuenta con un gran número de dispositivos enlazados, los cuales están ofreciendo o recibiendo un servicio específico, por lo cual, necesita de un servidor que monitoree su estado actual, con este objetivo el servidor SNMP es el encargado de realizar esta tarea.

En la actualidad existen muchas opciones en el campo de los servidores de monitoreo, donde, todos cuentan con sus ventajas propias, en este caso el software que se utiliza para el monitoreo de la red donde están enlazados los diferentes dispositivos dentro de una red es Nagios Core.

Nagios Core ofrece el servicio de monitoreo de manera gratuita o de paga, y su despliegue puede conllevar algunos pasos necesarios para la instalación con el objetivo de cumplir la tarea asignada de manera correcta y eficaz, además, también necesita realizar la configuración del servicio de monitoreo para cada una de las maquinas presentes en la red.

Lo que se busca es la completa automatización tanto del despliegue del servidor que cuenta con Nagios Core como de los servicios que el mismo brinda a los usuarios que lo conforman, por lo que Ansible DevOps se encarga del proceso de automatización, ya que, es una herramienta de gestión DevOps encargada de configuraciones automáticas.

Ansible cuenta con características específicas para cumplir el proceso de automatización estas deben ser configuradas de manera adecuada, tales como, los inventarios donde se almacenan las direcciones IP de los equipos, *playbooks* los cuales son los scripts de Ansible donde se ejecutan los pasos configurados por el administrador de red para el despliegue de un servicio.

Ese es el rol que cumple Ansible, la automatización, para lograr el despliegue correcto de Nagios Core y los clientes que forman parte de la red monitoreada de igual forma los servicios que mantienen los equipos de la red son monitoreados con la ayuda del servidor SNMP.

1.1 Objetivo general

Implementar un servidor SNMPv3 con la utilización de herramientas DevOps con el objetivo de cumplir tareas de automatización para la administración de equipos de networking de manera totalmente automatizada para comprobar el estado actual de los diferentes dispositivos dentro de la red que será monitoreada en un ambiente virtualizado.

1.2 Objetivos específicos

1. Analizar DevOps que soluciona cada servicio de networking.

2. Diseñar la solución para cada servicio de networking mediante herramientas de DevOps.

3. Implementar las soluciones mediante herramientas de DevOps para el despliegue de los servicios de networking.

4. Verificar el funcionamiento de cada servicio de networking implementado mediante DevOps.

1.3 Alcance

El presente proyecto pretende la utilización de DevOps y despliegue de soluciones para host o equipos de networking, con el objetivo de llevar a cabo este propósito es necesario contar con las herramientas disponibles de DevOps que cuenten con las capacidades de automatizar el despliegue de los servicios propuestos y además se puedan realizar pruebas de las soluciones.

1.4 Marco teórico

Nagios Core

Nagios Core es un software creado con las capacidades de realizar el monitoreo de red a cada uno de los diferentes elementos que forman parte de una red usualmente, como, maquinas host, dispositivos de red (Routers, switches), el monitoreo dentro de cada dispositivo de la red puede variar dependiendo el deseo del administrador de red.

Nagios Core ofrece una cantidad de monitoreos distintos como monitorear la carga del dispositivo, monitorear los servicios que el dispositivo posee (HTTP, HTTPS, SSH) para comprobar si los servicios se encuentran levantados o no, entre otros [8].

Protocolo SNMP

El protocolo SNMP esta principalmente enfocado al intercambio de información de administración dado en dispositivos de red y un servidor de monitoreo, permite la obtención de datos del dispositivo en cuestión y con las capacidades de realizar modificaciones en la configuración, el mismo organiza los datos del dispositivo en una estructura de árbol que lo denomina como “MIB” (*Management information base*).

Utiliza el protocolo UDP como protocolo de transporte, existen varias versiones del protocolo, el actual es el protocolo versión 3, esta mejora la seguridad del protocolo en comparación con las antiguas versiones, mejorando de tal forma su eficiencia, la arquitectura que mantiene es la de cliente/servidor, con el puerto 161 por defecto [9].

En la última versión del protocolo se enfoca principalmente en ofrecer los servicios de autenticación, confidencialidad e integridad, utiliza marcas de tiempo, cifrado, listas ACL, HASH y el concepto de flujos, todo esto lo emplea para evitar posibles alteraciones sea forzadas o inintencionales en mensajes recibidos, también para ataques de repetición (Reenvió de mensajes), puede evitar el *sniffing* entre otros servicios [9].

VirtualBox

VirtualBox es un software de virtualización, hipervisor de tipo 2 utilizado principalmente para la virtualización de sistemas operativos sobre la PC principal, dando como resultados las máquinas virtuales, este soporta múltiples host y varios sistemas operativos, posee un peso más bajo en comparación con otros softwares de virtualización tales como VMware.

Es menos restrictivo en el tema de las licencias, capaz de ejecutar maquinas del software VMware [10], tiene archivos de configuración XML, software de código abierto, pero también cuenta con su versión de paga, mejor opción al momento de necesitar ejecutar varias máquinas virtuales sobre computadores de uso personal, el mismo no exige una cantidad de recursos muy alta de memoria para usarlo con todas sus funciones [10].

GNS3

GNS3 es una red grafica multiplataforma ejecutable en Windows, OS X y Linux, el propósito en el cual GNS3 se encuentra enfocado principalmente es en el diseño de redes virtuales

con sus respectivas pruebas de funcionamiento en la PC donde se encuentre instalado el software.

Capaz de simular redes con Cisco IOS, Juniper, MikroTik, Arista y Vyatta, comúnmente este software es utilizado y aplicado por estudiantes para realizar pruebas en diferentes equipos de *networking,* por ejemplo, con Cisco IOS se practica el enrutamiento y conmutación, entre otros.

GNS3 permite simular ambientes de las redes de la actualidad, no es necesario la utilización de hardware físico para los testeos que se necesiten para comprobar el funcionamiento de una red, esto se puede realizar todo en un ambiente completo de virtualización, permite la modificación y creación de la red a disposición del administrador de la red [11].

2 METODOLOGÍA

- En primera instancia se realizará una investigación sobre el manejo de DevOps, Ansible, SNMPv3 y equipos de *networking.*

- Después de haber realizado la investigación sobre los temas se realiza el diseño dinámico de la configuración de SNMP v3 en equipos de *networking* con Ansible.

- Se implementa el algoritmo el cual incluye el servidor SNMP para que se pueda conectar con los equipos de *networking* y SNMPv3 en equipos de *networking* con Ansible

- Finalmente se realiza pruebas del algoritmo diseñado mediante una herramienta de DevOps y se verifica su buen funcionamiento en al menos 3 dispositivos.

3 RESULTADOS

El funcionamiento del sistema consiste principalmente en la automatización de la instalación de un servidor SNMP con DevOps, por lo cual, la automatización es realizada por Ansible, mientras el software encargado del servidor SNMP es Nagios Core en un ambiente virtualizado que en este caso se lo realiza en GNS3.

Los respectivos clientes de monitoreo que serán utilizados para realizar pruebas de funcionamiento del software, cuentan con el sistema operativo de código libre Linux

Ubuntu, además, un router de la marca Cisco y un switch que son configurados e implementados dentro del software GNS3.

3.1 Analizar la herramienta de DevOps que soluciona cada componente del proyecto de titulación

DevOps se caracteriza por su eficacia, flexible implementación y aprovisionamiento de procesos de negocio, ya que integra la entrega, el despliegue y operaciones, todo de forma automatizada.

DevOps cuenta con herramientas obligatorias en la automatización donde existen varias fases y cada una de sus fases cuenta con sus respectivas herramientas, sus 3 fases son *build, deployment* y *operations*.

En la fase *build* las herramientas son cruciales para el ambiente de DevOps, con ágil y fluido despliegue [5], se han convertido herramientas para la administración del software desplegado y el ciclo de vida del servicio [6][7]. En la fase de *deployment* es el despliegue total del servicio, mientras que la fase de *operations* es el servicio puesto en marcha.

Existen varias herramientas de gestión que cumplen con el proceso de configuración y orquestación, en este caso se tomaran en cuenta tres herramientas de gestión, la primera es Chef o Progress Chef, de segunda herramienta es Jenkins y por último seria Ansible.

Chef

Chef es una herramienta de gestión DevOps capaz de realizar la configuración, aprovisionamiento y gestión de la infraestructura, pero ocupa más recursos del sistema en comparación a Ansible, por lo que puede provocar ralentizaciones en el sistema, su funcionamiento es adecuado con sistemas de mayor escala y cumple bien este servicio.

Tomando en cuenta esto la herramienta que se eligió para el despliegue del servidor de monitoreo fue Ansible, ya que, cumple con los requerimientos necesarios para el despliegue del mismo y sus características permitirán la implementación de manera eficaz y automatizada.

Jenkins

Jenkins es una herramienta de gestión DevOps, de código abierto, con el objetivo principal de acelerar el proceso de entrega y desarrollo de software con ayuda de la automatización,

puede ser utilizado en las diferentes etapas que conllevan a la implementación de un servicio, como la compilación, el testeo, la documentación o el despliegue [12].

Ansible

Ansible es más simple en comparación con otras herramientas de gestión DevOps disponibles en el mercado, realiza los trabajos de configuración, aprovisionamiento y gestión.

Ofrece un servicio más sencillo, capaz de ejecutarse en segundo plano sin afectar el rendimiento del sistema, su principal uso se centra principalmente en realizar cambios en un sistema y configuraciones en nuevas máquinas, esto permite una escalabilidad más eficaz de dispositivos por lo que la velocidad de despliegue aumenta considerablemente.

Una vez revisado las diferentes herramientas de gestión DevOps la herramienta que se utilizará para la implementación del servidor SNMPv3 es Ansible.

Ansible es una herramienta de gestión de la configuración, simple, flexible, ofrece la poderosa capacidad de automatizar las tareas comunes de la infraestructura y de funcionar tanto para la gestión como para el despliegue de nuevas aplicaciones de varios niveles en una o en varias máquinas simultáneamente sin la necesidad de depender de otra herramienta para lograr este propósito.

Ansible utiliza conexiones SSH para conectarse a cada uno de los servidores remotos para empezar utilizar sus respectivos scripts creados, en Ansible un *script* es conocido como *playbook,* el código es enviado sobre la conexión SSH y ejecuta el *script*, por lo cual, no es necesario que ansible se encuentre instalado en el servidor remoto [1][2].

Ansible ofrece una serie de beneficios por la cual se ha convertido en la opción de muchos en la gestión, configuración y despliegue de servicios, tales como, sintaxis de fácil lectura, ninguna instalación en los servidores o host remotos, administrar desde uno hasta miles de nodos [2].

Playbook

El *playbook* es el término que Ansible utiliza para un *script* de gestión de configuración, esto permite que una serie de tareas específicas configuradas se ejecuten a la vez y cumplan esta serie de pasos en diferentes maquinas o servidores remotos que se encuentren enlazados al *playbook*.

Su objetivo principal es que todos los pasos necesarios para el despliegue de un servicio o configuraciones posteriores se encuentren automatizadas y sin errores dentro del *playbook,* para evitar carga del administrador de la red y que todo se encuentre de manera automatizada y simplificada para el ser humano, los *playbooks* cuentan con sintaxis YAML, generalmente facilita al usuario a leerlo y escribirlo más fácilmente [1].

Inventarios en Ansible

Un inventario en Ansible es usado para categorizar los hosts administrados por Ansible, pueden ser categorizados por secciones, estas secciones dependen de la función que mantenga el host dentro de la red, esto simplifica al administrador de la red tener acceso a la información del campo que cumple cada uno de los usuarios, la información desplegada indica la sección en donde se encuentra y la dirección IP.

El inventario funciona a que todos los hosts dentro del inventario puedan recibir tareas automatizadas de manera simultánea. De esta forma, permite seleccionar a un grupo de hosts sin la necesidad de seleccionar uno de uno a uno agilizando el servicio que se ofrecerá y facilitando al administrador de la red su trabajo [3][1][4].

3.2 Diseñar la solución para cada servicio de networking mediante herramientas de DevOps

Se busca automatizar el despliegue del servicio de Nagios Core, por lo cual, el primer paso a realizar seria la creación de la topología de red esta topología se encuentra en el software para redes virtuales GNS3, luego, toca añadir las máquinas virtuales, un router de la marca Cisco y un switch para el respectivo monitoreo en los mismos.

Una máquina virtual está encargada de la automatización de la red, la cual es la máquina de Ansible, se tiene que configurar los requerimientos que necesita Ansible para poder realizar la automatización los cuales son los *playbooks* y los diferentes inventarios para lograr la automatización del despliegue del servidor SNMP.

La otra máquina es la del servidor SNMP que utiliza el software Nagios Core, esta máquina es la encargada de cumplir las tareas de monitoreo, lo realiza al cliente, al servidor HTTP, al router de la marca Cisco modelo c7200 y al switch, donde tanto el host Linux, el servidor, el router Cisco y el switch necesitan configurarse con direcciones IP para recibir los servicios del servidor SNMP de manera adecuada.

3.3 Implementar las soluciones mediante herramientas de DevOps para el despliegue de los servicios de networking

Para empezar, es necesario realizar la creación de máquinas virtuales, por lo tanto, es necesario descargar Linux Ubuntu con la versión deseada en este caso Ubuntu 20.04.6, esto se debe instalar a 4 máquinas virtuales distintas o realizar copias de la máquina virtual para no repetir el mismo proceso 3 veces adicionales.

Para la instalación de las máquinas virtuales en este caso se utiliza el software de virtualización VirtualBox, donde el primer paso es añadir una nueva máquina virtual que en este caso el Ubuntu que este utiliza es Ubuntu 20.04.6, en la Figura *3.1* se puede observar la creación de la máquina virtual.

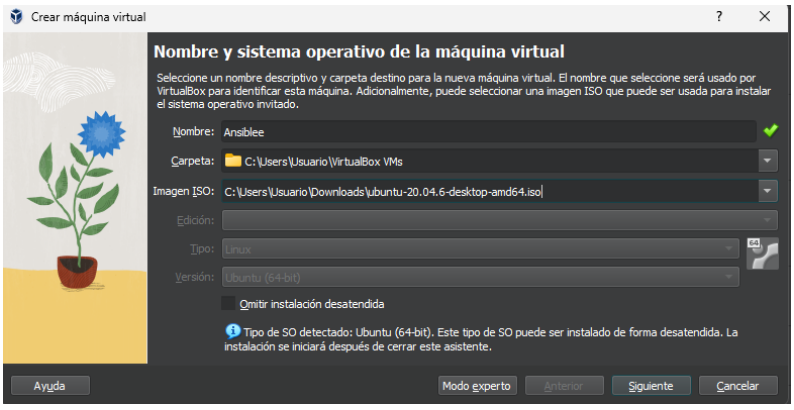


Figura 3.1 Instalación de máquina virtual.

A continuación, se añade el usuario y contraseña con el cual se accede a la máquina virtual que usa el sistema operativo Linux Ubuntu, en la Figura *3.2* se comprueba la creación del usuario y la contraseña.

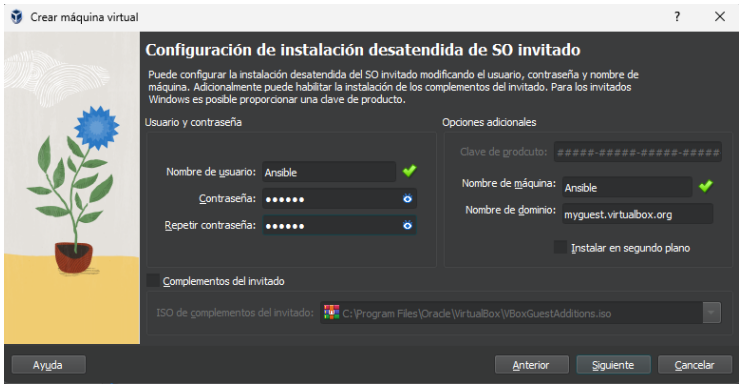


Figura 3.2 Creación de usuario y contraseña

Una vez finalizado se procede a darle a la máquina virtual su hardware correspondiente, en este caso se da una memoria base de 2560Mb con un solo procesador. En la Figura 3.3 se observa los recursos de memoria base y procesadores brindadas a la máquina virtual.

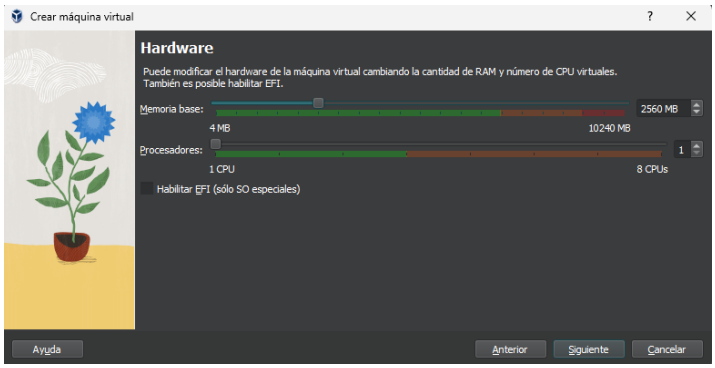


Figura 3.3 Hardware máquina virtual

Después, se procede con el disco duro virtual, lo recomendable es darle una cantidad adecuada para evitar problemas en el futuro, por lo cual 25Gb puede ser una buena opción. En la Figura 3.4 se observa el proceso mencionado.

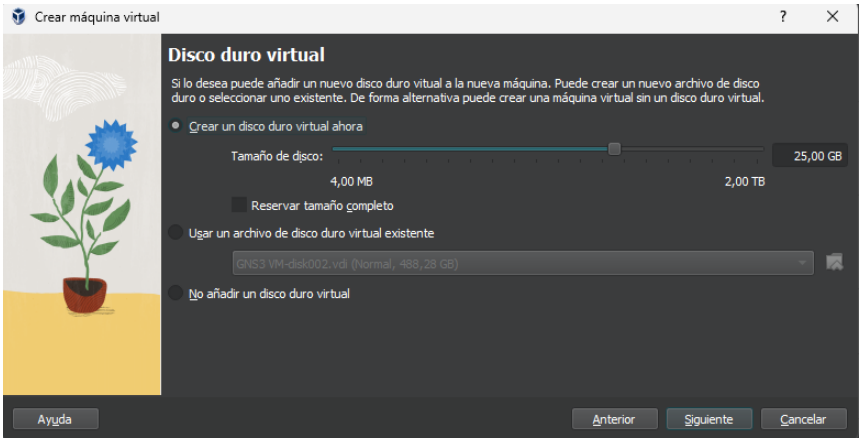


Figura 3.4 Disco duro virtual

Terminado esto se abre la máquina virtual y se empieza con su instalación, donde lo que queda ahora de hacer es esperar que la máquina virtual se instale correctamente. En la Figura 3.5 se observa el proceso de instalación.

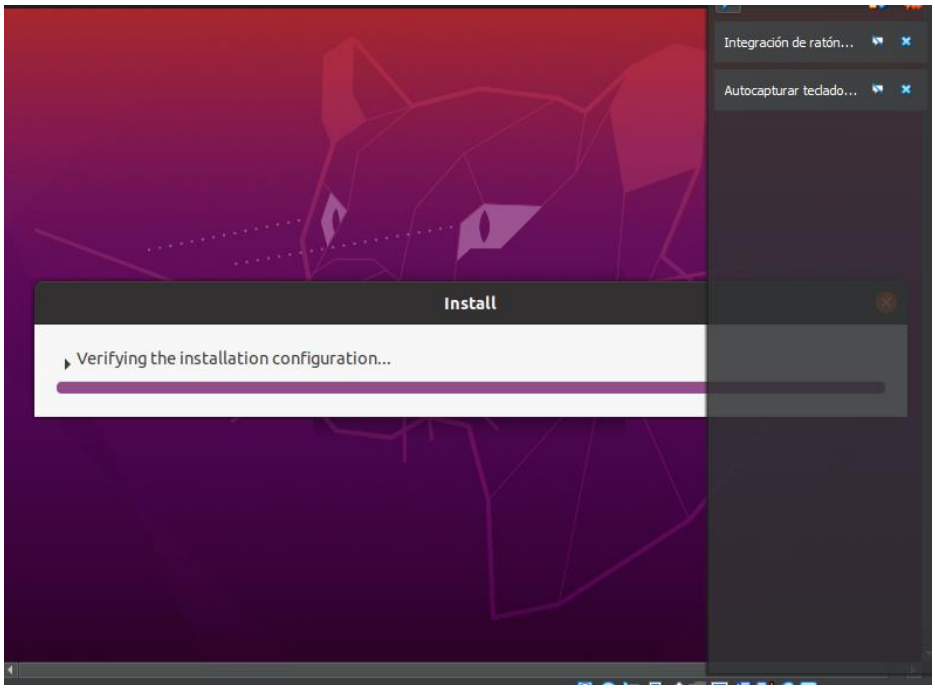


Figura 3.5 Instalación máquina virtual Linux

Al momento de realizarse la instalación se deben seguir una serie de pasos para la correcta instalación, si es que ocurrió algún problema con el nombre de usuario y contraseña configurados anteriormente, se lo puede realizar otra vez en una siguiente pantalla. En la Figura 3.6 se observa la pantalla mencionada.

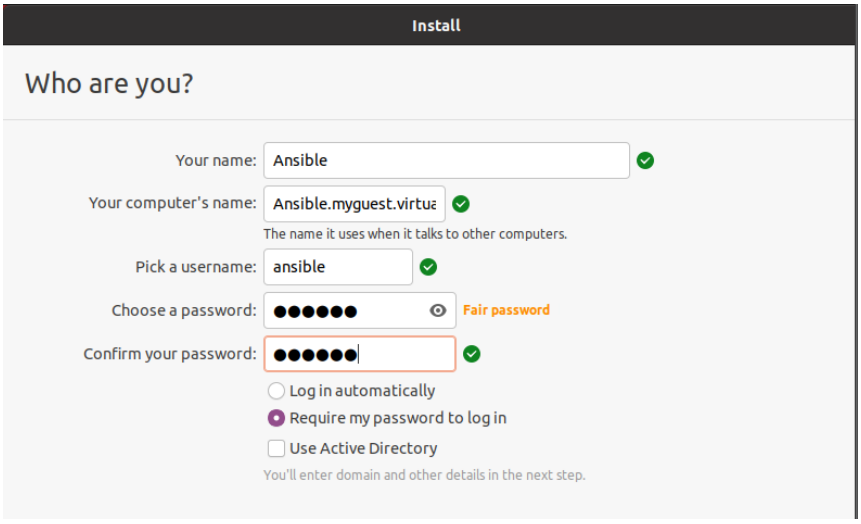


Figura 3.6 Correcciones nombre de usuario y contraseña

Una vez se encuentre realizada la instalación de la máquina Ubuntu Linux, se realiza la clonación de la maquina 3 veces, con esto ya se tendría el número necesario de máquinas virtuales la cuales son 4.

Luego, se procede con la creación de la red virtual, esta se realiza en el diseñador de redes virtuales GNS3, por lo que se debe realizar la instalación previa de GNS3 para utilizar sus funciones, para ello, se debe acceder a su sitio web https://gns3.com/software/download y descargar dependiendo el sistema operativo que se tenga. En la Figura 3.7 se observa el sitio web mencionado.

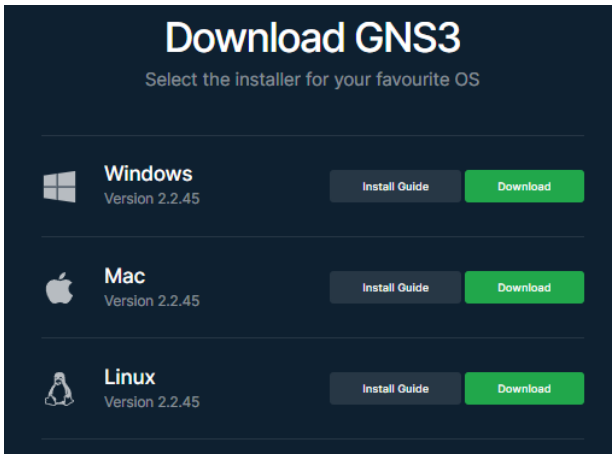


Figura 3.7 Descargar GNS3

Si se van a utilizar máquinas virtuales dentro de GNS3 es necesario que se descargue de igual forma la versión GNS3 VM, la cual se encuentra en esa misma pantalla, y se descarga dependiendo el software de virtualización. En la Figura 3.8 se observa donde se accede

al sitio y las versiones disponibles para descargar, en este caso se descarga la versión de VirtualBox.

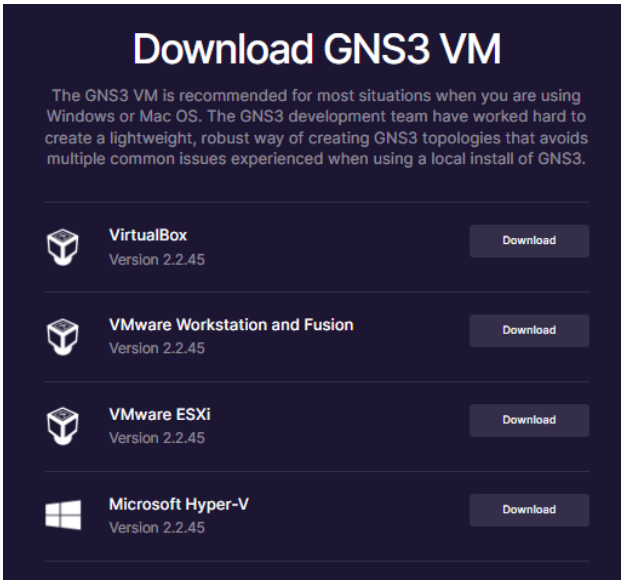


Figura 3.8 Descargar GNS3 VM

Una vez descargados ambos, se procede con la respectiva instalación, el GNS3 VM debe ser instalado dentro de VirtualBox, para la instalación dentro de VirtualBox, dentro del software lo primero una vez descomprimido el archivo de descarga del GNS3 VM es acceder a las siguientes opciones, en la Figura 3.9 se observa el proceso mencionado sobre la instalación de GNS3 VM.

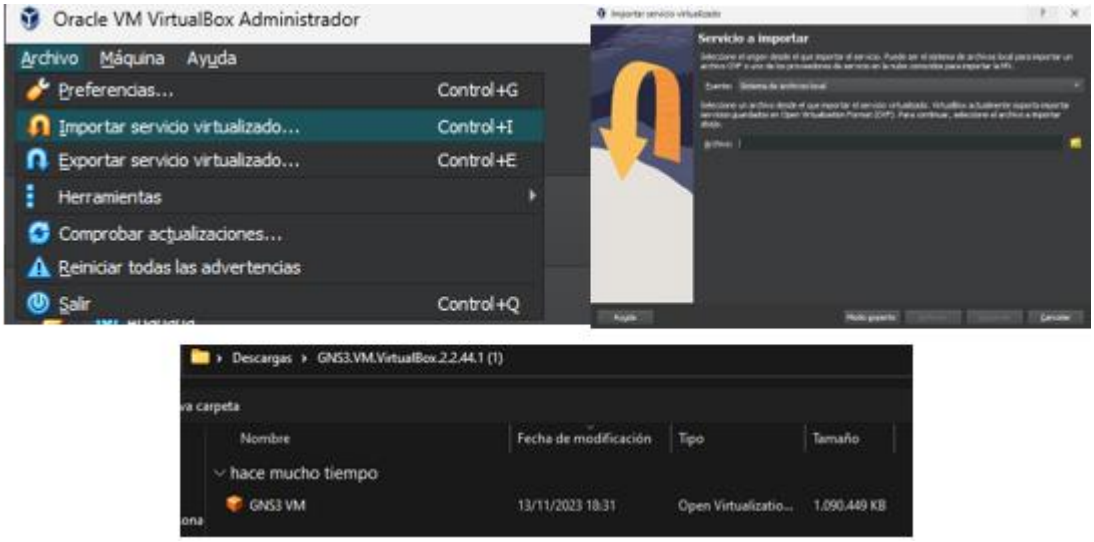


Figura 3.9 importación de servicio virtualizado

Para la instalación del software GNS3 simplemente se debe seguir los pasos correspondientes brindados por el instalador.

Ahora, se debe activar la virtualización dentro de GNS3 para el manejo de máquinas virtuales, para ello, se debe acceder a GNS3 con la configuración del GNS3 VM realizada anteriormente, ahora se accede a la pantalla de "*Edit"* y luego en *"Preferences",* donde se desplegará la siguiente pantalla, en la Figura 3.10 se observa la pantalla de *"Preferences"*.

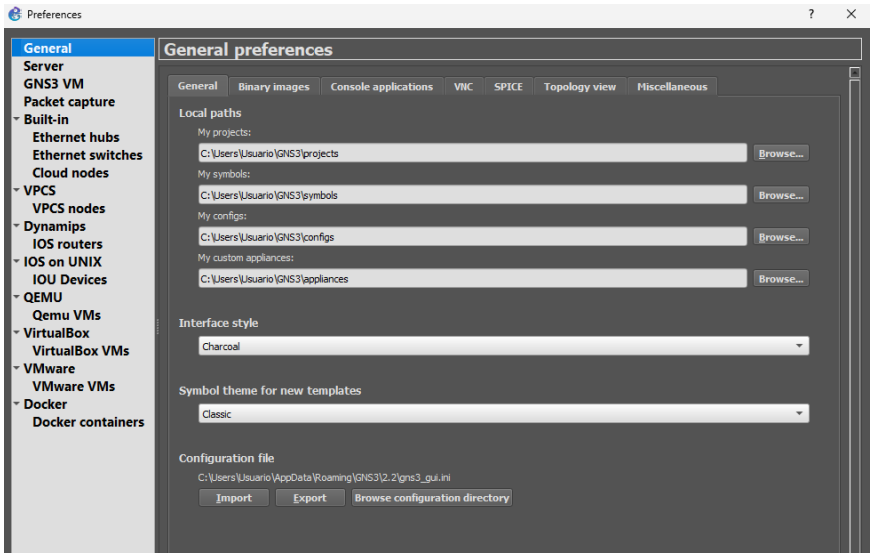


Figura 3.10 Preferencias en GNS3

Después, se accede donde dice "GNS3 VM" y se lo configura de la siguiente forma. En la Figura 3.11 se visualiza la configuración realizada.

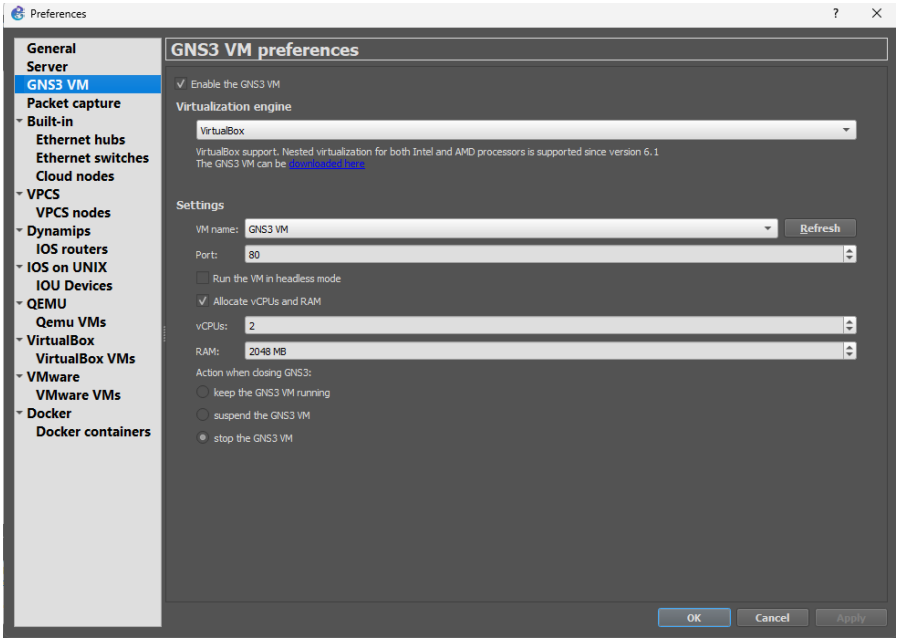


Figura 3.11 Configuración GNS3 VM

Una vez completado lo anterior, finalmente solo queda añadir las máquinas virtuales que se encuentran configuradas en VirtualBox, para esto, dentro del menú "*Preferences",* toca ir a la pestaña de "*VirtualBox VMs",* donde se deben añadir uno por uno las máquinas virtuales necesarios para el proyecto. En la Figura 3.12 se observa la pestaña de *"VirtualBox VMs"*.

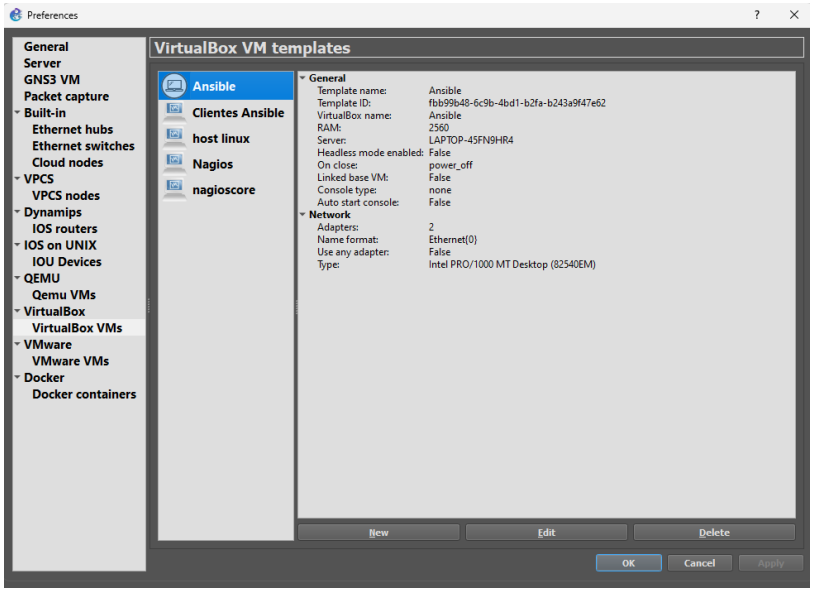


Figura 3.12 Configuración máquinas de VirtualBox en GNS3

Simplemente se presiona en la parte donde dice *"New",* lo primero es elegir el tipo de servidor con el que se ejecutara la máquina virtual de VirtualBox, se debe tomar la opción *"Run the VirtualBox VM on my local computer"*. En la Figura 3.13 se visualiza la opción tomada.

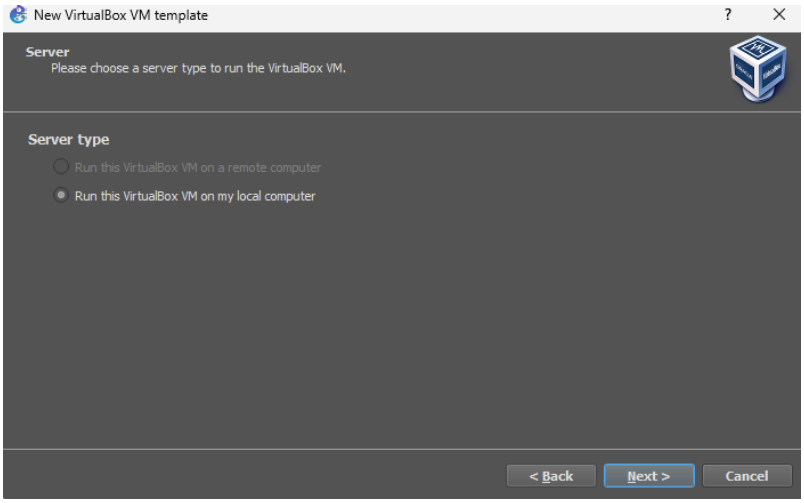


Figura 3.13 Tipo de servidor para la máquina virtual de VirtualBox

Lo siguiente es escoger la máquina virtual que se encuentra en VirtualBox y que se llevará al diseñador de redes virtuales GNS3, y con eso la máquina virtual ya se encontraría dentro de GNS3. En la Figura 3.14 se visualiza la elección de la máquina virtual.

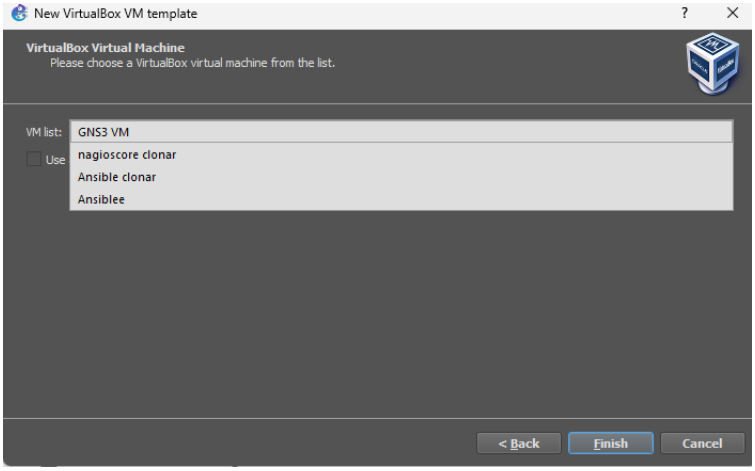


Figura 3.14 Elección máquina virtual de VirtualBox

Presiona "*Finish" y* se aplica los cambios mediante el botón de "*Apply"* para que se añada la máquina virtual, luego se presiona en la máquina virtual añadida, luego en "Edit", donde se debe modificar el número de adaptadores con el cual cuenta, debe de ser el mismo número de adaptadores tanto en GNS3 como en el software de virtualización VirtualBox. En la Figura 3.15 se realiza los cambios realizados en los adaptadores de red.

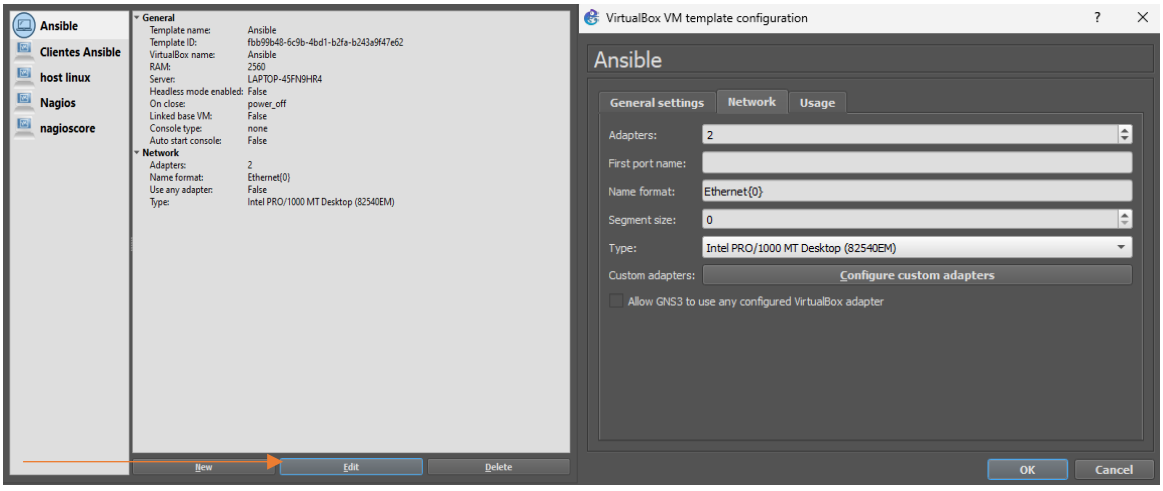


Figura 3.15 Modificar los adaptadores de Red

Siempre al modificar cualquier cosa en esta pestaña es necesario presionar el botón de "*Apply"*, y con esto las máquinas virtuales de VirtualBox ya están configuradas de manera correcta dentro de GNS3.

Después, el siguiente paso es añadir los router Cisco, para añadir router Cisco a GNS3, es necesario descargar el *appliance* del router cisco que se desee, el router utilizado en este caso se encuentra en la siguiente dirección: https://www.gns3.com/gns3/appliance/download?url=https%3A%2F%2Fraw.githubusercontent.com%2FGNS3%2Fgns3-registry%2Fmaster%2Fappliances%2Fcisco-7200.gns3a.

Una vez descargado, ahora es necesario dirigirse a la pantalla de GNS3, y en la pestaña de *"File"* y luego en la opción *"Import appliance".* En la Figura 3.16 se indica la opción de *"Import appliance"*.

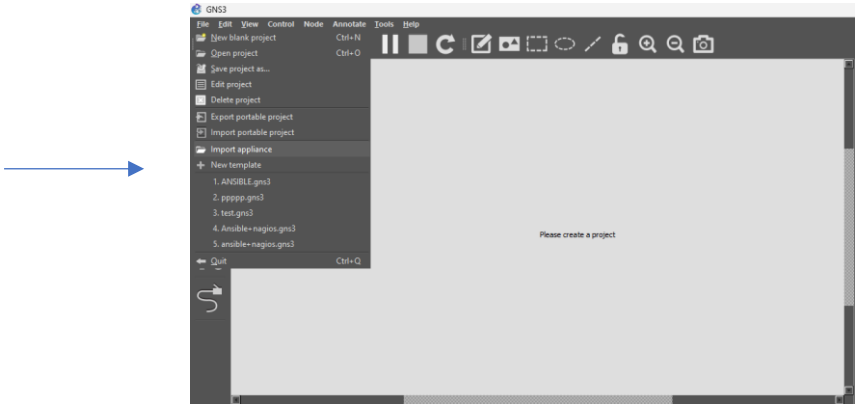


Figura 3.16 Importar appliance

Y se escoge el archivo descargado, luego se procede a configurar esta *appliance,* donde se deben seguir los siguientes pasos. En la Figura 3.17 se visualiza la configuración del *appliance*.

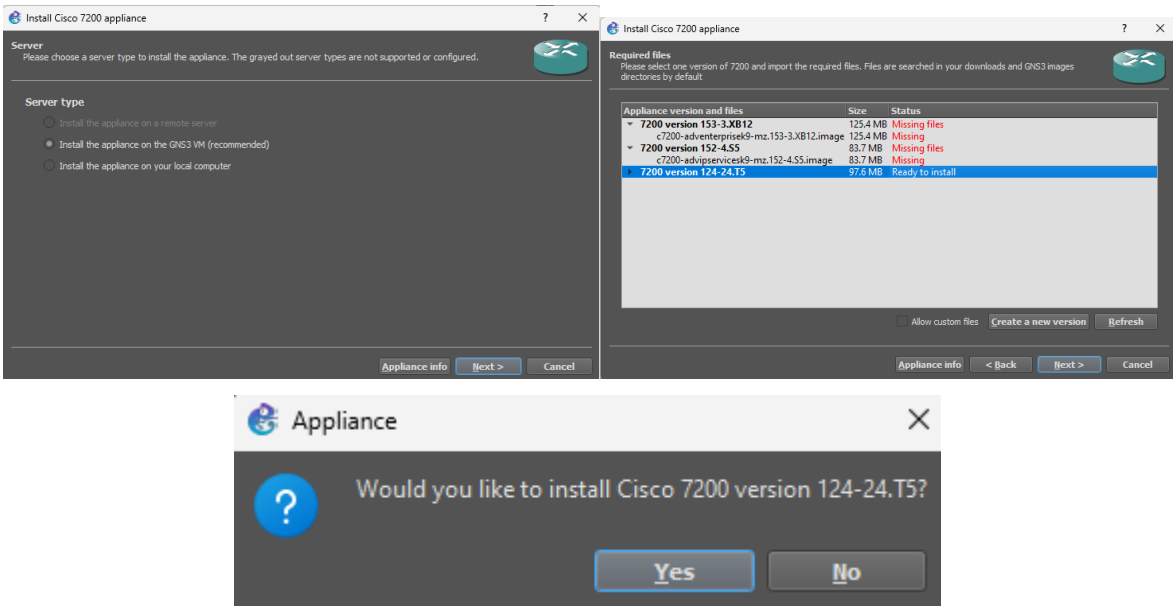


Figura 3.17 Instalación appliance

Ahora, simplemente se escoge la topología que se desee, cuenta con 4 máquinas virtuales, un router Cisco y un switch, en la Figura 3.18 se observa la topología.

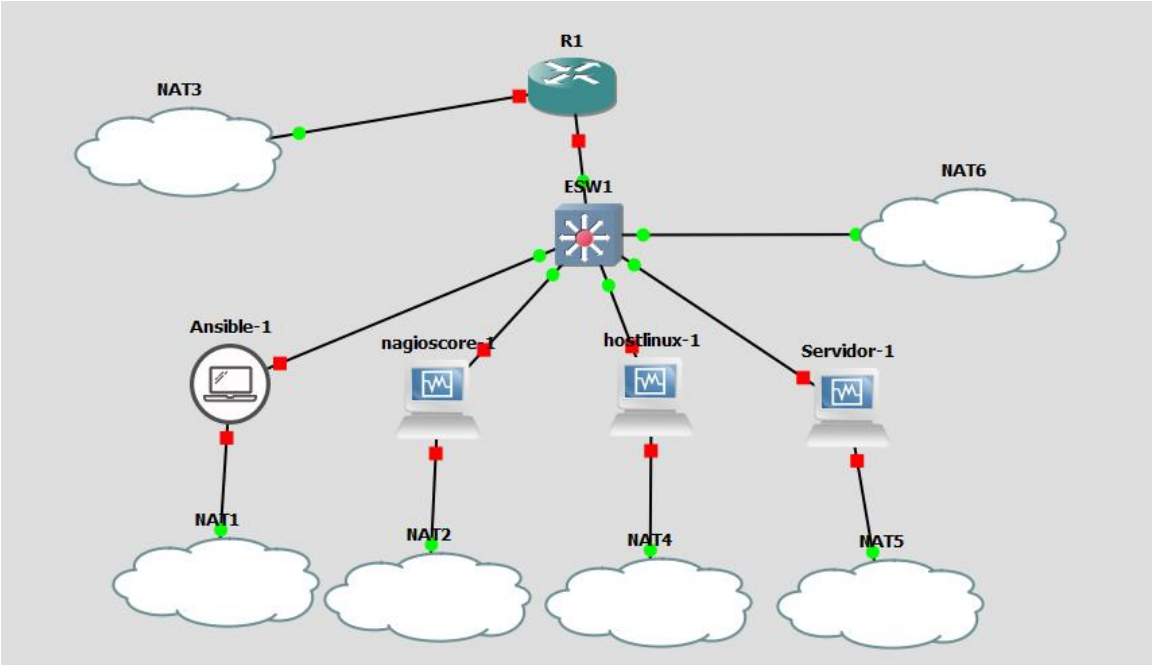


Figura 3.18 Topología en GNS3

El software que se utiliza para el monitoreo es Nagios Core, el cual es instalado en una maquina con sistema operativo Linux al igual que sus clientes, Ansible se conecta a este servidor mediante comunicaciones SSH, por lo cual, lo primero que se debe realizar es la instalación de Ansible en la máquina virtual, con el comando *"pip install ansible"*

Una vez usado el comando para la instalación de Ansible, se recomienda crear un directorio para tener todo más organizado donde se encuentran los inventarios, *playbooks* esto se lo realiza de manera sencilla con el comando, *"mkdir <nombre del directorio>"*

Después de creado el directorio se procede con la instalación del servicio ssh tanto en la maquina con Ansible como en la máquina del servidor SNMP, por lo cual se utiliza el comando *"sudo apt-get install openssh-server"*

Pero, en la máquina donde se encuentra el servidor SNMP, se debe realizar otra configuración adicional, ya que es necesario que la maquina Ansible pueda acceder a la máquina del servidor como usuario administrador *(root),* por esto se debe realizar una configuración con la ayuda del comando *"sudo nano /etc/ssh/sshd_config"* y modificar la siguiente línea, en la Figura 3.19 se observa la línea a modificar.

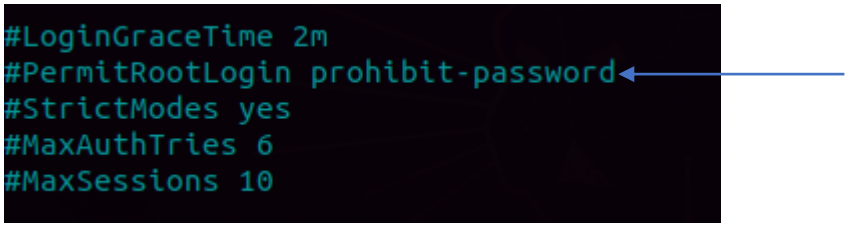


Figura 3.19 Archivo de configuración de SSH en el servidor SNMP

Y debe ser modificada a esto, en la Figura 3.20 se visualiza el cambio realizado

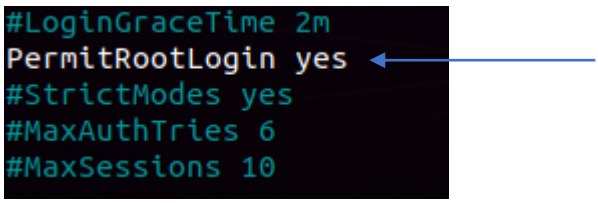


Figura 3.20 Archivo de configuración de SSH modificado

Una vez finalizado este cambio se resetea el servicio SSH para que aplique los respectivos cambios en el archivo de configuración con el comando *"sudo systemctl restart ssh"*

Ahora, se procede a crear llaves de SSH públicas y privadas, esto para que la maquina con Ansible pueda acceder como *root* a la máquina del servidor SNMP sin necesidad de indicar la contraseña nuevamente, para esto se utiliza el comando "*ssh-keygen*" para la creación de las llaves, esto debe realizarse en la máquina que cuenta con Ansible.

Se procede a presionar "*enter"* repetidas veces hasta finalizar el proceso de creación de las llaves, ahora se debe compartir la llave privada de Ansible con el servidor SNMP para el acceso, el comando *"ssh-copy-id root@<Direccion IP del servidor>"* se aplica en la máquina de Ansible.

En esta parte se debe aceptar con un *"yes"* y colocar la contraseña del servidor SNMP para que la llave sea copiada con éxito y se pueda acceder mediante SSH sin necesidad de colocar la contraseña.

Una vez realizado lo anterior se procede con la creación del inventario, para esto simplemente vamos a la máquina de ansible para la creación del archivo con extensión .ini, se accede al directorio creado anteriormente y se procede con la creación del inventario esto se lo realiza con el comando *"nano <archivo con extension .ini>"*

En este archivo se añade la dirección IP del servidor SNMP, en la **Figura 3.21** se observa el nombre del archivo creado con la respectiva dirección IP del servidor SNMP.

Figura 3.21 Inventario Ansible

Después de crear el inventario se procede con la creación del *playbook* con el cual se automatiza el despliegue de Nagios Core, para ello, se debe crear el *playbook* correspondiente, estos archivos cuentan con una extensión .yml, y se lo crea usando el comando *"nano <Archivo con extensión .yml>"*

Una vez creado al archivo se procede con la programación del *playbook*, el *playbook* en cuestión solamente es para activar el servicio de Nagios Core, para los archivos de configuración se necesitan otros *playbooks.* El playbook para la instalación de Nagios Core se encuentra al final del documento [Ver ANEXO III].

En el primer *playbook,* se encuentran varias tareas creadas justamente para el despliegue e implementación, en las primeras líneas del *playbook* se define el propósito del *playbook,* el inventario a cuál está vinculado y las *vars* que serán utilizadas dentro del playbook. En la Figura 3.22 se observa lo mencionado anteriormente.

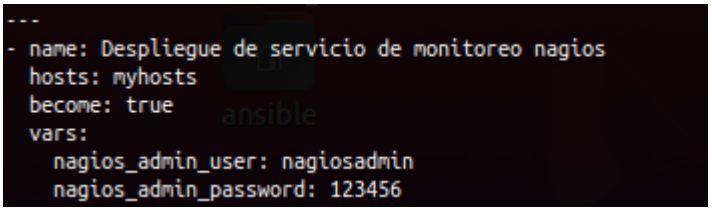


Figura 3.22 Definición inicio del playbook

Una vez definido, se procede a dar las tareas a cumplir por el *playbook*, la primera tarea definida es para actualizar el cache, el propósito de esta línea es para actualizar los repositorios para conocer si existen versiones de paquetes que pueden ser instaladas o actualizadas. En la Figura 3.23 se observa la línea de texto necesaria para la actualización del cache.

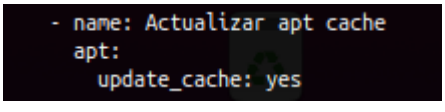


Figura 3.23 Tarea cache

El siguiente paso es actualizar los paquetes a las versiones que pueden ser actualizadas, para ello se crea una tarea donde mediante la ayuda del *upgrade* se logra este propósito. En la Figura 3.24 se observa la tarea mencionada para realizar el *upgrade*.

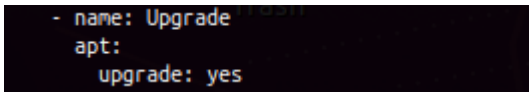


Figura 3.24 Tarea Upgrade

Luego se procede a descargar e instalar los paquetes necesarios para el despliegue del servidor de monitoreo.

Esto se logra con la ayuda de *apt* seguido de *"name: "{{ item }}""* con *"state: present"* para descargar la última versión del paquete especificado, con la ayuda de "*with_items:"*, se especifica cada uno de los paquetes para el despliegue de Nagios Core. En la Figura 3.25 se visualiza la creación de la tarea para la descarga e instalación de los paquetes necesarios para el funcionamiento del servidor SNMP.

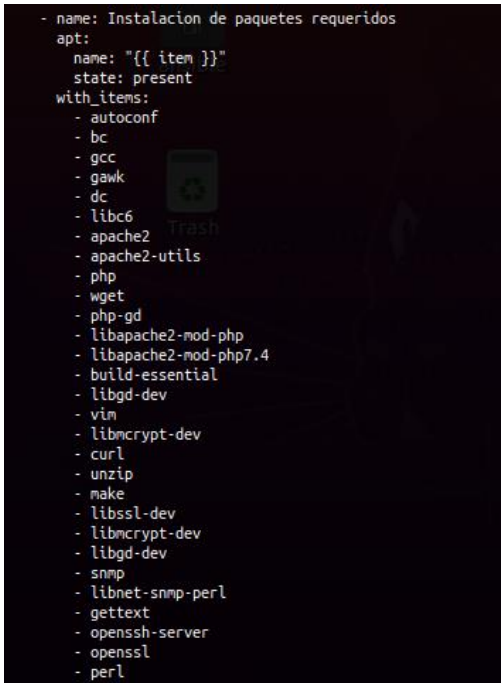


Figura 3.25 Instalación paquetes necesarios

Una vez instalado los paquetes, se procede a crear un usuario nagios y se lo añade a un grupo, esto con la ayuda de la línea de texto "*shell",* que funciona de tal forma como si las líneas de texto colocadas sean como si estuvieran siendo escritas en la terminal Linux. En la Figura 3.26 se observa la tarea creada para la creación del usuario nagios y grupo.

```
- name: Crear usuarios nagios y grupo
  shell: |
    useradd -m -s /bin/bash nagios
    groupadd nagcmd
    usermod -aG nagcmd nagios
    usermod -aG nagcmd www-data
```

Figura 3.26 Creación usuario grupo Nagios

Luego con la ayuda de igual forma de la línea de texto *"shell"*, se procede a configurar el firewall para admitir los puertos de ssh y http y se procede al reinicio del firewall para que efectué los cambios. En la Figura 3.27 se visualiza la tarea para permitir puestos ssh y http.

```
- name: Permitir puertos de ssh y http
  shell: |
    ufw allow ssh
    ufw allow http
    ufw reload
```

Figura 3.27 Configuración firewall

Después, se procede a la descarga de Nagios Core y de los Plugins que ocupa Nagios Core, con la función "get_url:" permite que con la ayuda de la siguiente línea de texto "url:" y con la dirección del archivo se procede a la descarga, la dirección para la descarga de Nagios Core es la siguiente: https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.5.0/nagios-4.5.0.tar.gz , y con la ayuda de la línea *"dest:"* se especifica la ruta a donde irá el archivo después de ser descargado.

Y para la descarga de Nagios Plugins es la siguiente dirección: https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.8/nagios-plugins-2.4.8.tar.gz

En la Figura 3.28 se observa la tarea para la descarga de ambos archivos .tar necesarios para la implementación del servidor.

```
- name: Descargar nagios core
  get_url:
    url: https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.5.0/nagios-4.5.0.tar.gz
    dest: /tmp/nagios-4.5.0.tar.gz
    validate_certs: no

- name: Descargar nagios plugins
  get_url:
    url: https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.8/nagios-plugins-2.4.8.tar.gz
    dest: /tmp/nagios-plugins-2.4.8.tar.gz
    validate_certs: no
```

Figura 3.28 Descargar Nagios Core y Nagios Plugins

Ahora se descomprime los respectivos archivos descargados, con la ayuda de la función “unarchive:”, con “src:” es para especificar la ruta y el archivo que se descomprimirá, el “dest:” de igual forma para definir la ruta donde será ruta de destino, “remote_src: yes” para especificar que en el servidor remoto se hará la descompresión. En la Figura 3.29 se visualiza la tarea para la descomprensión de los archivos.

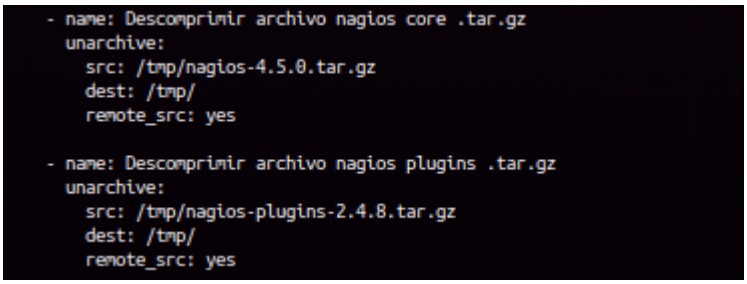


Figura 3.29 Descomprensión de archivos

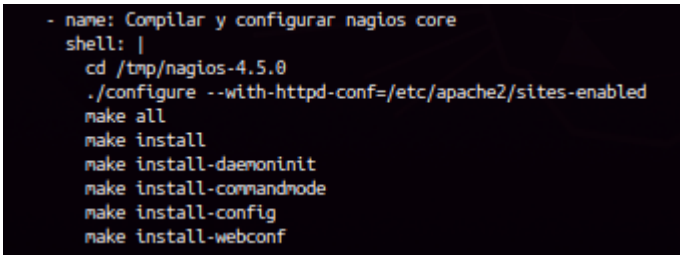
Ahora mediante la función *“shell",* se procede a acceder al archivo descomprimido para realizar la compilación y configuración pertinentes, con la línea de texto *“make install”,* esto instala los archivos binarios, CGIs y archivos html.

Con “*make install-daemoninit”* se instalan los archivos Daemon y también los configura para su inicio.

Con “*make install-commandmode”* lo instala y configura el archivo externo de comando.

Con *“make install-config”* funciona para instalar archivos de configuración, estos son requeridos para habilitar el inicio de Nagios.

Y por último “*make install-webconf”,* este último funciona para configuración de los archivos del servidor web y además realiza las configuraciones pertinentes de apache. En la Figura 3.30 se observa la tarea para la configuración y compilación mencionadas.



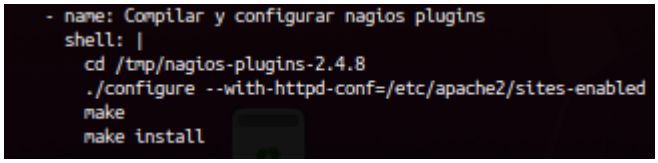


Figura 3.30 Configuración y compilación Nagios Core y sus plugins

Ahora se procede a la creación del usuario administrador de Nagios Core, con la ayuda de la función *"shell"*, se utiliza el comando htpasswd para la creación de un usuario y contraseña para acceder a la interfaz web de Nagios, en la Figura 3.31 se visualiza la tarea para la creación del usuario administrador de Nagios Core, en esta línea de texto se usan las *vars* creadas anteriormente, en la Figura 3.2231 se observan las vars mencionadas.

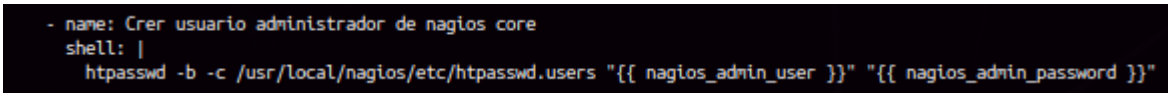


Figura 3.31 Creación usuario y contraseña interfaz web Nagios

Lo siguiente es añadir una línea de texto en el archivo de configuración "commands.cfg" para habilitar los agentes NRPE en el servidor SNMP, "line:" es añadir líneas de texto al archivo de configuración, y la línea de texto que dice "insertafter:" funciona para especificar después de que línea será añadido al texto, y "EOF" significa *End Of File* para especificar que la línea de texto sea añadida al final del archivo de configuración. En la Figura 3.32 se visualiza la tarea para añadir el comando de utilización de los agentes NRPE.

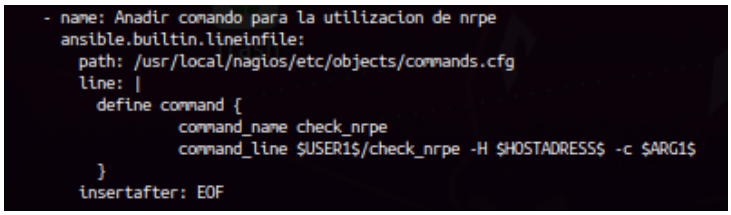


Figura 3.32 Comando para la utilización de los agentes NRPE

Después, se procede a iniciar, habilitar el servicio de Nagios y apache, se reiniciará el servicio apache con el comando *"systemctl",* donde "*enable"* es para habilitar el servicio, *"restart"* para el reinicio y *"start"* para empezar el servicio. En la Figura 3.33 se visualiza la tarea para el inicio, habilitación del servicio de Nagios Core y de igual de forma el inicio, la habilitación, y el reinicio del servicio de Apache.

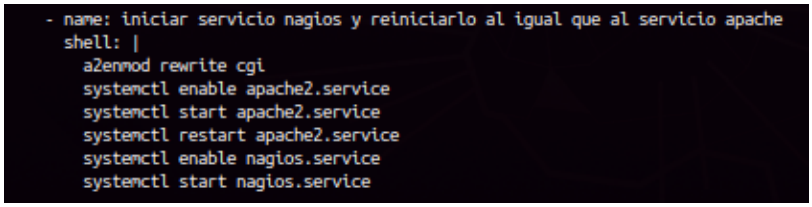


Figura 3.33 Inicio reinicio de apache y Nagios

Ahora se reinicia la maquina objetivo con la función *"reboot",* gracias a esto se puede reiniciar la máquina de manera exitosa. En la Figura 3.34 se observa la tarea del reinicio de la máquina virtual.

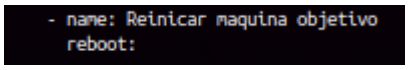


Figura 3.34 Reboo*t*

Ya para finalizar simplemente se dan permisos en un script de Nagios para realizar cambios en la interfaz web del servidor Nagios, con la ayuda del comando "*chmod"* que funciona para dar permisos de escritura, lectura y ejecución. En la Figura 3.35 se observa la tarea para los permisos.

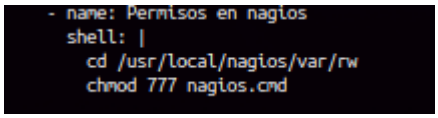


Figura 3.35 Permisos Nagios

Y con esto ya estaría finalizada la implementación de Nagios Core.

A continuación de creado el *playbook* para la instalación de Nagios Core, se procede con la creación del archivo de configuración del cliente Linux para el monitoreo de Nagios Core, este archivo se crea en la maquina con Ansible mediante el comando "*nano"* y debe tener una extensión .cfg, para ello, el archivo de configuración se encuentra de igual forma al final del documento [Ver ANEXO IV].

En el archivo de configuración se define el host a ser monitoreado con la ayuda de "define host" el uso *"use"* y se lo define como "linux-server" para que conozca que se trata de una maquina con el sistema operativo Linux, el nombre del host *"host_name"*, su alias, y su respectiva dirección IP *"address"*. En la Figura 3.36 se visualiza la definición del host cliente.

```
define host {
        use             linux-server
        host_name       cliente
        alias           cliente
        address         192.168.92.146
}
```

Figura 3.36 Definir host Linux.

Ahora se van a definir los diferentes servicios monitoreados al cliente.

Esto se realiza con la ayuda del *"check_command",* para que utilice los diferentes *checks* correspondientes *"check_ping"* para realizar el ping entre el servidor SNMP y la máquina cliente.

El *"check_local_procs"* para el monitoreo del número total de procesos.

El "*check_local_load"* para el monitoreo de la carga actual del dispositivo.

Y por último el "*check_ssh"* para verificar el servicio *ssh* se encuentra levantado y con eso estaría completado el archivo de configuración del cliente Linux. En la Figura 3.37 se visualiza definidos los servicios que serán monitoreados al cliente Linux.

```
define service {
  use                     generic-service
  host_name               cliente
  service_description     PING
  check_command           check_ping!100.0,20%!500.0,60%
}

define service {

    use                   generic-service
    host_name             cliente
    service_description   Total Processes
    check_command         check_local_procs!250!400!RSZDT
}


define service {

    use                   generic-service
    host_name             cliente
    service_description   Current Load
    check_command         check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}
define service {

    use                   generic-service
    host_name             cliente
    service_description   SSH
    check_command         check_ssh
    notifications_enabled 0
}
```

Figura 3.37 Definición de servicios Host Linux

Una vez finalizado con la definición de los servicios del host Linux, se procede creación del archivo de configuración del servidor HTTP, de igual forma el archivo de configuración se encuentra al final del documento [Ver ANEXO V].

Lo primero a realizar es la definición del host, se define el host a ser monitoreado otra vez con la ayuda de *"define host"* el uso *"use"* donde se le configura como *"linux-server"* de igual forma que al cliente ya que las dos son máquinas cuentan con el sistema operativo Linux, el nombre del host *"host_name"*, su alias, y su respectiva dirección IP *"address"*. En la Figura 3.38 se observa la definición del host servidor.

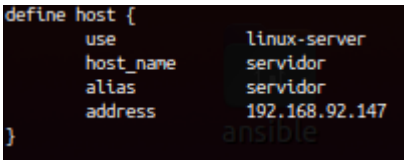


Figura 3.38 Definición host servidor HTTP.

Después, se procede definir los servicios que son monitoreados a el servidor, de igual forma con la ayuda del *"check_command",* para que utilice los diferentes *checks* correspondientes.

El *"check_ping"* para realizar el ping correspondiente entre el servidor HTTP y el servidor SNMP.

El *"check_local_procs"* que permite el monitoreo del número total de procesos.

El "*check_local_load"* que permite que se realice el monitoreo de la carga actual del dispositivo.

El "*check_ssh"* para verificar si el servicio ssh se encuentra levantado.

Y por último el servicio que será monitoreado es el HTTP, esto con la ayuda de *"check_http"* con eso estaría completado el archivo de configuración del servidor HTTP. En la Figura 3.39 se visualiza el número total de servicios que se monitorea en el servidor HTTP.

Figura 3.39 Definición de servicios del Servidor HTTP

Seguido de esto ahora se procede con la creación de otro archivo de configuración .cfg, esta vez con el objetivo de crear un archivo de configuración para el monitoreo de los routers Cisco, el archivo de configuración se encuentra al final del archivo de igual forma [Ver ANEXO VI].

Para comenzar, se debe definir el host con “define host” el uso *“use”* que en este caso el uso se lo define como “generic-switch” para que se conozca que se realizará el monitoreo en un router o un switch, el nombre del host *“host_name”*, su alias, y su respectiva dirección IP *“address”*. En la Figura 3.40 se visualiza la definición del router cisco.

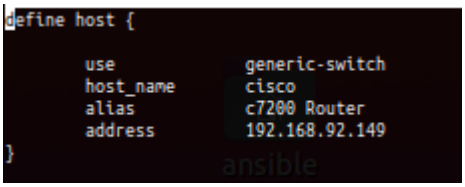


Figura 3.40 Definición del Router Cisco

A continuación, se procede a la definición de los servicios a monitorear del router Cisco, donde nuevamente se utiliza *“check_command”,* para que utilice los diferentes *checks*.

El *“check_ping”* que funciona para comprobar si existe conexión entre el servidor de monitoreo y el router cisco.

El “check_snmp” para el monitoreo correspondiente del tiempo que se encuentra levantado el router Cisco. En la Figura 3.41 se observa los servicios que se monitoreara en el router.

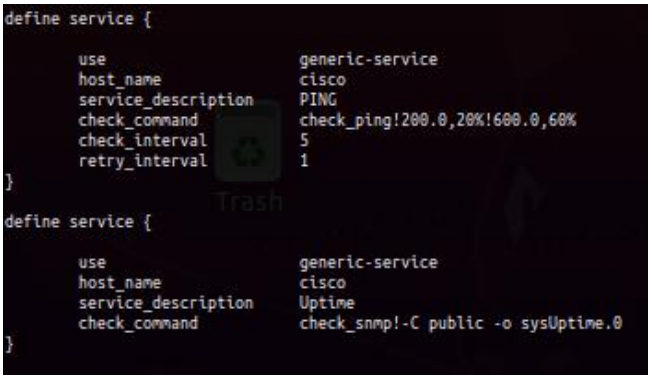


Figura 3.41 Definición de los servicios de router Cisco

Después, se procede con la creación de otro archivo de configuración .cfg, esta vez con el objetivo de crear un archivo de configuración para el monitoreo del switch, el archivo de configuración se encuentra al final del archivo de igual forma [Ver ANEXO VII].

Primero, se debe definir el host con “define host” el uso *“use”* que en este caso el uso se lo define como “generic-switch” para que se conozca que se realizará el monitoreo en un router o un switch, el nombre del host *“host_name”*, su alias, y su respectiva dirección IP *“address”*. En la Figura 3.42 se visualiza la definición del switch.

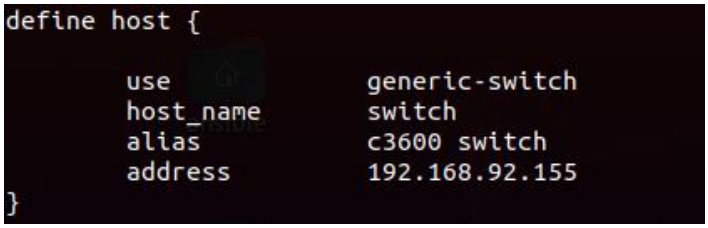


Figura 3.42 Definición Switch

Se procede a la definición de los servicios a monitorear del switch, donde nuevamente se utiliza *“check_command”,* para que utilice los diferentes *checks*.

El *“check_ping”* que funciona para comprobar si existe conexión entre el servidor de monitoreo y switch.

El “check_snmp” para el monitoreo correspondiente del tiempo que se encuentra levantado el router Cisco. En la Figura 3.43 se observa los servicios que se monitoreara en el switch.

```
define service {

        use                     generic-service
        host_name               switch
        service_description     PING
        check_command           check_ping!200.0,20%!600.0,60%
        check_interval          5
        retry_interval          1
}

define service {

        use                     generic-service
        host_name               switch
        service_description     Uptime
        check_command           check_snmp!-C public -o sysUptime.0
}
```

Figura 3.43 Definición de los servicios del switch.

Después, para enviar los diferentes archivos .cfg creados a la máquina del servidor Nagios Core se utilizan *playbook* adicionales diferentes al de implementación Nagios Core. Estos *playbooks* funcionan para habilitar el monitoreo en la máquina cliente, el servidor HTTP y el router Cisco.

El primer playbook para la habilitación de los dispositivos a ser monitoreados es para la máquina cliente, de igual forma se encuentra al final del documento [Ver ANEXO VIII].

Al principio del *playbook* se encuentra la definición del nombre del *playbook*, y el inventario al cual está vinculado el *playbook*. En la Figura 3.424 se visualiza la definición del *playbook* para la habilitación del cliente.

```
---
- name: configuraciones para cliente linux
  hosts: myhosts
  become: true
```

Figura 3.424 Definición playbook clientes Linux

Lo primero que se va a realizar es la transferencia del archivo al servidor SNMP.

La función *"template"* permite la transferencia de archivos, con *"src:"* se especifica el archivo que será enviado, con *"dest:"* se especifica la ruta a donde será enviado el archivo de configuración de los clientes Linux. En la Figura 3.435 se visualiza la tarea para la transferencia del archivo de configuración del cliente hacia el servidor SNMP.

```
  - name: Archivo para cliente linux
    template:
      src: hostlinux.cfg
      dest: /usr/local/nagios/etc/objects/hostlinux.cfg
```

Figura 3.435 Transferencia archivo al servidor SNMP

La siguiente parte del playbook es para añadir una línea de texto al archivo de configuración de Nagios que se encuentra en el servidor SNMP, la función de esta línea de texto es para que el servidor de monitoreo tome en cuenta el archivo enviado.

Para esto se utiliza la función *"lineinfile"* para añadir la línea de texto deseada.

Lo siguiente es especificar la ruta del archivo con *"path:"*.

Después se añade la línea de texto que se va a adicionar, con "line:".

Para finalizar con la ayuda de "insertafter:" se decide en que parte del archivo se desea añadir la línea de texto, en este caso es con *"EOF"* que significa *"End Of File"* para añadir la línea al final del archivo. En la Figura 3.446 se visualiza la creación de la tarea para añadir la línea de texto para que el archivo de configuración del cliente sea tomado en cuenta por Nagios Core.

```
- name: linea para permitir el archivo de configuracion del cliente linux
  lineinfile:
    path: /usr/local/nagios/etc/nagios.cfg
    line: 'cfg_file=/usr/local/nagios/etc/objects/hostlinux.cfg'
    insertafter: EOF
```

Figura 3.446 Línea habilitar archivo de configuración host Linux

Ya para finalizar se procede al reinicio del servicio de Nagios, con la ayuda del *"systemd:"* que funciona para especificar el servicio que será reiniciado, con la función *"name:"* es para especificar el servicio, y con el *"state"* para especificar que se hará con el servicio, en este caso con *"restarted"* es para reiniciarlo. En la Figura 3.457 se observa la tarea para el reinicio del servicio de Nagios.

```
- name: Reinicar servidor nagios
  systemd:
    name: nagios
    state: restarted
```

Figura 3.457 Reinicio servidor Nagios en playbook hostlinux

Luego, se crea un *playbook* de igual forma para la habilitación del monitoreo del servidor HTTP, este funciona de igual forma para habilitar el archivo de configuración y enviar el archivo de configuración del servidor HTTP al servidor donde se encuentra instalado el software Nagios Core [Ver ANEXO IX].

Lo primeras líneas del *playbook* son simplemente para especificar el nombre, el inventario al cual está vinculado. En la Figura 3.468 se visualiza la definición del *playbook* para la habilitación del servidor HTTP.

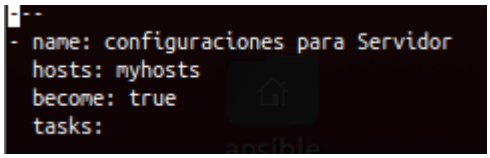


Figura 3.468 definición playbook servidor

Igual que en el caso del cliente Linux utiliza de igual forma las mismas funciones *"template:", "src:"* y *"dest"*, que simplemente es para la transferencia del archivo de configuración del servidor. En la Figura 3.479 se observa la tarea para la transferencia del archivo de configuración del servidor HTTP hacia el servidor SNMP.

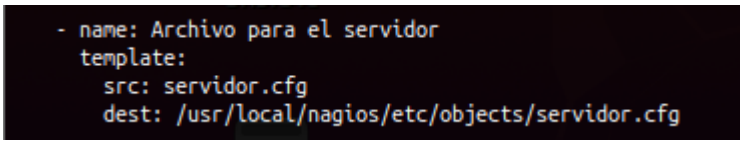


Figura 3.479 Transferencia archivo de configuración del servidor

Luego sigue la línea que de igual forma que en el *playbook* anterior funciona para habilitar el archivo de configuración para el monitoreo del servidor, con ayuda de *"lineinfile:", "path:" "line:" "insertafter:"*, solo que ahora varia la línea de *"line:"* ya que ahora el archivo de configuración para ser tomado en cuenta ahora es el archivo de configuración del servidor HTTP. En la Figura 3.50 se observa la creación de la tarea para añadir la línea de texto.

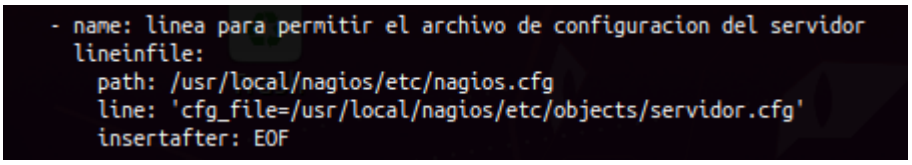


Figura 3.50 Línea habilitar archivo de configuración servidor

Ahora se procede al reinicio del servicio de Nagios, de igual forma que en el *playbook* anterior se utiliza *"systemd:" "name:"* y *"state"* para el reinicio del servidor Nagios. En la Figura 3.51 se observa la tarea para el reinicio del servicio de Nagios.

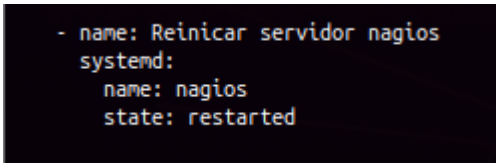


Figura 3.51 Reinicio servidor Nagios en playbook del servidor

Luego, se crea un playbook para la habilitación de monitoreo del router Cisco, de igual forma se encuentra al final del documento [Ver ANEXO X].

El comienzo del *playbook* es simplemente para especificar el nombre, el inventario al cual está vinculado. En la Figura 3.52 se visualiza la definición del *playbook* para la habilitación del router Cisco.

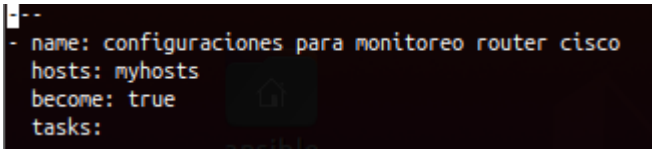


Figura 3.52 Definición playbook router

De igual forma que en los casos anteriores se utilizan las funciones *"template:", "src:"* y *"dest"*, que simplemente es para la transferencia del archivo de configuración del router Cisco, En la Figura 3.53 se observa la tarea para la transferencia del archivo de configuración del router Cisco hacia el servidor SNMP.

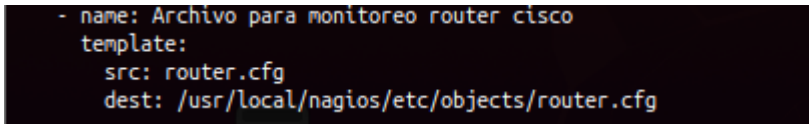


Figura 3.53 Transferencia archivo de configuración del Router

Después, al igual que en los *playbooks* anteriores, ahora se debe habilitar el archivo de configuración para el monitoreo del router, con ayuda de *"lineinfile:", "path:" "line:" "insertafter:"*.

En este caso varia la línea de *"line:"* ya que ahora el archivo de configuración para ser tomado en cuenta ahora es el archivo de configuración del router. En la Figura 3.54 se observa la creación de la tarea para añadir la línea de texto para que el archivo de configuración del router Cisco sea tomado en cuenta por Nagios Core.

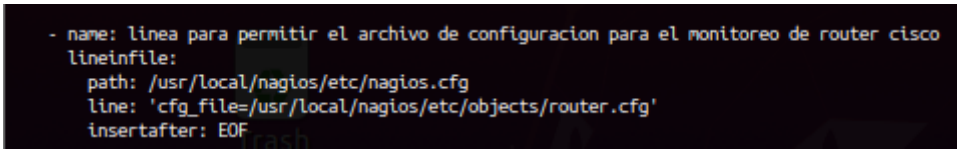


Figura 3.54 Línea para habilitar archivo de configuración del router

Finalmente, se procede a reiniciar el servicio de Nagios, con las funciones utilizadas en los *playbooks* anteriores. En la Figura 3.485 se observa el reinicio del servicio de Nagios.

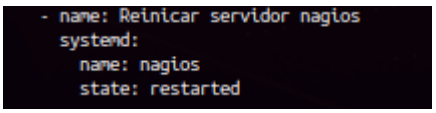


Figura 3.485 Reinicio servicio Nagios en playbook router

Luego, se crea un playbook para la habilitación de monitoreo del switch, de igual forma se encuentra al final del documento [Ver ANEXO XI].

El comienzo del *playbook* es simplemente para especificar el nombre, el inventario al cual está vinculado. En la Figura 3.56 se visualiza la definición del *playbook* para la habilitación del switch.

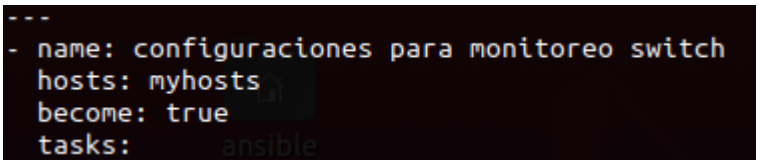


Figura 3.56 Definición playbook switch

De igual forma que en los casos anteriores se utilizan las funciones *"template:", "src:"* y *"dest"*, que simplemente es para la transferencia del archivo de configuración del switch, En la Figura 3.57 se observa la tarea para la transferencia del archivo de configuración del switch hacia el servidor SNMP.

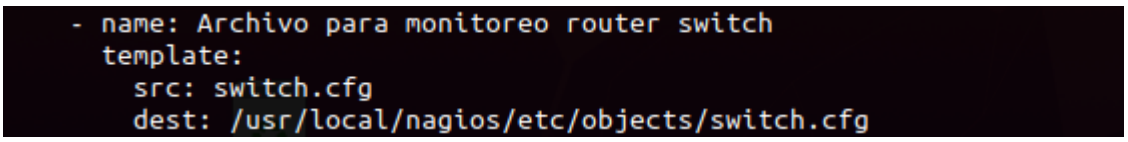


Figura 3.57 Línea para habilitar archivo de configuración del switch

Al igual que en los *playbooks* anteriores, se habilita el archivo de configuración para el monitoreo del switch, con *"lineinfile:", "path:" "line:" "insertafter:"*, en este caso varia la línea de *"line:"* ya que ahora el archivo de configuración para ser tomado en cuenta ahora es el archivo de configuración del switch. En la Figura 3.58 se observa la creación de la tarea

para añadir la línea de texto para que el archivo de configuración del switch sea tomado en cuenta por Nagios Core.

```
- name: linea para permitir el archivo de configuracion para el monitoreo del switch
  lineinfile:
    path: /usr/local/nagios/etc/nagios.cfg
    line: 'cfg_file=/usr/local/nagios/etc/objects/switch.cfg'
    insertafter: EOF
```

Figura 3.58 Línea para habilitar archivo de configuración del switch

Finalmente, se procede a reiniciar el servicio de Nagios, con las funciones utilizadas en los *playbooks* anteriores. En la Figura 3.59 se observa el reinicio del servicio de Nagios.

```
- name: Reinicar servidor nagios
  systemd:
    name: nagios
    state: restarted
```

Figura 3.59 Reinicio servicio Nagios en playbook switch

Y con todo esto ya se encuentra configurados los 4 clientes que serán monitoreados y configurado todo dentro del servidor con el software Nagios Core.

En el software Nagios Core, los parámetros monitoreados son los siguientes:

Current load: Es para verificar la carga actual del dispositivo de red monitoreado en el dispositivo monitoreado.

Current Users: Para comprobar la cantidad de usuarios conectados en ese momento en el dispositivo monitoreado.

HTTP: Para comprobar si el servicio de HTTP se encuentra levantado en el dispositivo monitoreado.

SSH: Para comprobar si el servicio de HTTP se encuentra levantado en el dispositivo monitoreado.

Swap usage: Monitorear para comprobar si se está utilizando en el dispositivo monitoreado.

Root partition: Para comprobar la partición del disco en el dispositivo monitoreado.

Total processes: Para comprobar el número total de procesos en el dispositivo monitoreado

3.4 Verificar el funcionamiento de cada servicio de networking implementado mediante DevOps.

Finalmente, se procede con la comprobación del funcionamiento de los *playbooks*.

Para ello, se accede la maquina donde se encuentra instalado Nagios Core para comprobar si el servicio se encuentra levantado y funcionando correctamente, lo primero será comprobar si el software Nagios Core se encuentra funcionando para ello simplemente se accede a la dirección IP del servidor en el navegador de la siguiente forma *"http://<Direccion IP del servidor>/nagios/"*

y se accede con las credenciales configuradas en el *playbook*. En la figura Figura 3.60 se visualiza la interfaz web de Nagios Core.

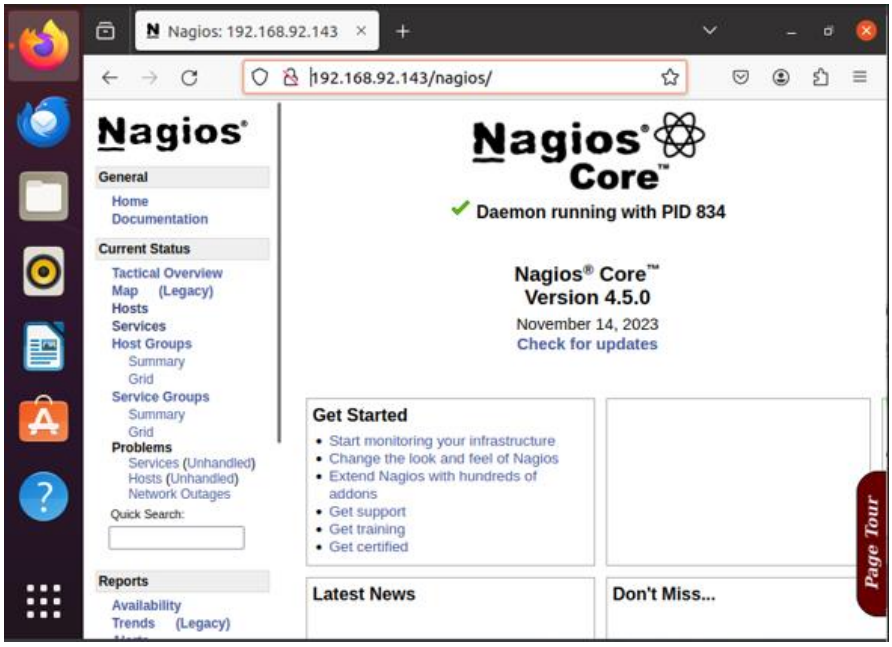


Figura 3.60 Interfaz Nagios Core

Una vez comprobado que Nagios Core está funcionando se procede a comprobar si está monitoreando el host Linux, el servidor HTTP, el router y el switch, los 4 hosts deben estar presente como *"UP"*, esto se logra accediendo a la sección de "Hosts" en la interfaz web de Nagios Core. En la Figura 3.61 se observa los hosts mencionados.

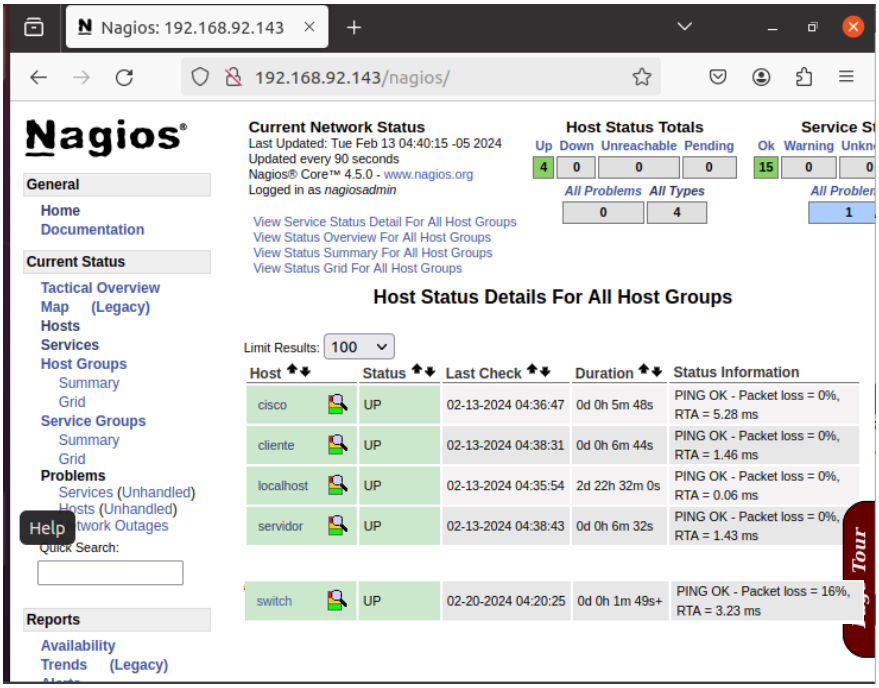


Figura 3.61 Hosts monitoreados en la interfaz web de Nagios

Una vez comprobado que los hosts están siendo monitoreados adecuadamente se procede a comprobar los servicios, para ello se accede al host seleccionándolo y se comprueba que todo está en orden, en primer lugar, se revisa el router. En la Figura 3.62 se observa la comprobación del funcionamiento de monitoreo del router cisco.

Figura 3.62 Comprobación Router Cisco

Luego, se comprueba el estado de monitoreo del switch, para ello ahora se accede a switch dentro de la pestaña “Hosts”, para comprobar que todo se encuentra OK. En la figura 3.63 se comprueba el estado del switch

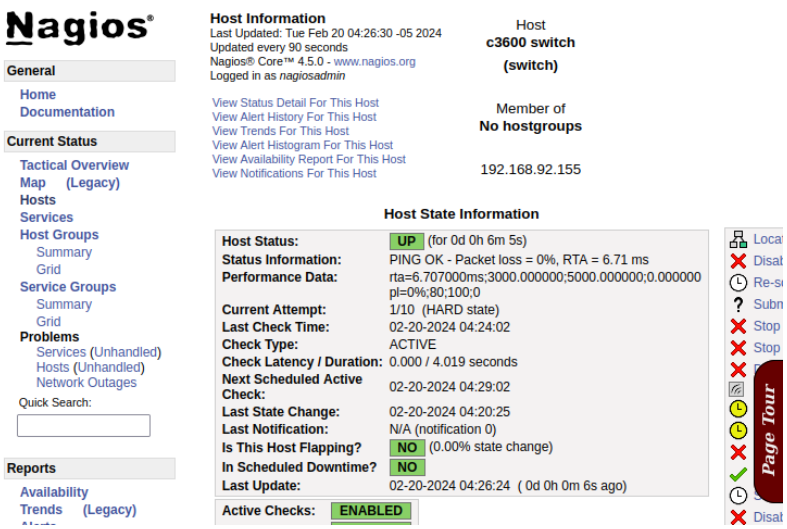


Figura 3.63 Comprobación switch

Una vez comprobado que el router y el switch están levantados de manera correcta, se procede a comprobar los servicios monitoreados al cliente Linux, para ello se accede a la pestaña de servicios. En la Figura 3.64 se observa la pestaña de servicios dentro de la interfaz web de Nagios Core.

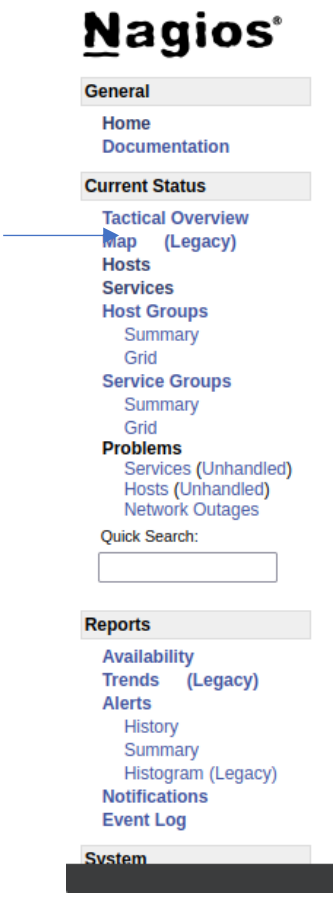


Figura 3.64 Pestaña de servicios en interfaz web de Nagios Core.

Luego, se procede a comprobar los diferentes servicios que fueron especificados para el monitoreo, se empieza con los servicios del cliente Linux, donde los servicios fueron definidos en los archivos de configuración, ahora simplemente se comprueba si los servicios están siendo monitoreados. En la Figura 3.65 se observan los servicios monitoreados al cliente Linux.

cliente	Current Load	OK	02-13-2024 05:08:13	0d 0h 32m 56s	1/3	OK - load average: 0.39, 0.30, 0.33
	PING	OK	02-13-2024 05:09:49	0d 0h 31m 20s	1/3	PING OK - Packet loss = 0%, RTA = 1.79 ms
	SSH	OK	02-13-2024 05:01:25	0d 0h 29m 44s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.9 (protocol 2.0)
	Total Processes	OK	02-13-2024 05:05:24	0d 0h 35m 45s	1/3	PROCS OK: 47 processes with STATE = RSZDT

Figura 3.65 Monitoreo de los servicios del cliente Linux

Una vez comprobado que los servicios del cliente Linux se encuentran siendo monitoreados de manera adecuada, se procede a comprobar el *localhost*. En la Figura 3.66 se monitorea los servicios al localhost.

localhost	Current Load	OK	02-13-2024 05:12:00	0d 0h 35m 42s	1/4	OK - load average: 1.35, 0.76, 0.49
	Current Users	OK	02-13-2024 05:09:30	2d 23h 2m 53s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	02-13-2024 05:10:13	2d 23h 2m 29s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0,013 second response time
	PING	OK	02-13-2024 05:09:13	2d 23h 2m 51s	1/4	PING OK - Packet loss = 0%, RTA = 0.07 ms
	Root Partition	OK	02-13-2024 05:10:48	2d 23h 2m 14s	1/4	DISK OK - free space: / 12728 MiB (54,74% inode=85%):
	SSH	OK	02-13-2024 05:12:25	0d 0h 35m 18s	1/4	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.11 (protocol 2.0)
	Swap Usage	OK	02-13-2024 05:09:41	0d 0h 38m 1s	1/4	SWAP OK - 100% free (1162 MB out of 1162 MB)
	Total Processes	OK	02-13-2024 05:09:12	0d 0h 38m 30s	1/4	PROCS OK: 47 processes with STATE = RSZDT

Figura 3.66 Monitoreo de los servicios del localhost

Después de comprobar que el monitoreo correspondiente en el localhost está funcionando de manera correcta, se procede a verificar que el servidor, recordando que los servicios fueron anteriormente configurados en el archivo de configuración del servidor HTTP. En la Figura 3.67 se comprueba los servicios monitoreados al servidor.

servidor	Current Load	OK	02-13-2024 05:12:14	0d 0h 32m 0s	1/3	OK - load average: 1.21, 0.75, 0.49
	HTTP	OK	02-13-2024 05:06:12	0d 0h 38m 2s	1/3	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0,006 second response time
	PING	OK	02-13-2024 05:07:49	0d 0h 36m 25s	1/3	PING OK - Packet loss = 0%, RTA = 1.82 ms
	SSH	OK	02-13-2024 05:09:25	0d 0h 34m 49s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.9 (protocol 2.0)
	Total Processes	OK	02-13-2024 05:11:01	0d 0h 33m 13s	1/3	PROCS OK: 47 processes with STAT = RSZDT

Figura 3.67 Monitoreo de los servicios del servidor HTTP

Y con esto finaliza el presente proyecto de titulación, las pruebas de funcionamiento se encuentran grabadas en video para comprobar el funcionamiento correcto en **ANEXO II** Código QR de la implementación y pruebas del funcionamiento.

4 CONCLUSIONES

- Automatizar el despliegue de servidores de cualquier tipo (HTTP, FTP, SSH, entre otros) con DevOps permite agilizar de gran manera el proceso, ya que debido a sus tareas de automatización el trabajo manual del administrador de la red es reducido sustancialmente logrando que no existan muchas fallas humanas o velocidades ineficientes todo esto porque el levantamiento ya es totalmente automatizado.
- Utilizar herramientas de gestión DevOps, tales como, Ansible, Chief o Jenkins, cumplen con todos los requerimientos necesarios para el despliegue de servidores de cualquier tipo en una red, logrando que se puedan implementar de manera más rápida y eficaz debido a sus características propias de automatización del despliegue del servicio.
- Contar con un servidor SNMP permite que se realice el monitoreo de todas las máquinas que constan en una red, necesita una escalabilidad muy veloz, con herramientas DevOps se logra este propósito, debido a que se puede configurar para que la red pueda escalar de manera automatizada eliminando el proceso manual para cada nuevo host que se adicione a la red.
- Implementar un servidor SNMP mediante el uso de herramientas DevOps es una solución simple y eficaz en estos tiempos, y Ansible es capaz de satisfacer cada necesidad de configuración, aprovisionamiento y despliegue debido a sus características propias que la convierten en una herramienta simple y fácil de utilizar.
- Ofrecer servicios automatizados logra un rápido despliegue e Implementación de servicios en la actualidad, debido a que en épocas anteriores todo se realizaba de manera manual, provocando que exista una necesidad total de manejo del humano, lo que provocaba en algunos casos retrasos ineficientes, gracias a la automatización ya no sucede de la misma forma.
- Analizados los resultados obtenidos se observa la rapidez que ofrece la utilización de DevOps en comparación con realizar todo de manera manual, este mismo puede lograr la implementación de un servidor SNMPv3 con todos sus requerimientos y permitiendo el funcionamiento correcto del mismo.

5 RECOMENDACIONES

- Debido a que Ansible varía sus *playbooks* dependiendo del sistema operativo con el cual trabaja se recomienda que se estudie las diferentes sintaxis qué existen entre un sistema operativo y otro, con esto se evitaría errores al crear los *playbooks* de Ansible.

- Existen muchas herramientas DevOps, cada una ofrece sus propios beneficios y sus desventajas, y lo recomendable es utilizar la herramienta que más se ajuste al caso donde será utilizada.

- Nagios Core es un software de monitoreo con una gran cantidad de servicios que no fueron explorados por completo en este proyecto de titulación, por lo que es recomendable investigar más sobre Nagios Core para conocer todas las funciones, facilidades y ventajas que el mismo ofrece para darle un uso más adecuado y eficaz en futuros proyectos.

6 REFERENCIAS

[1] Hochstein, L., & Moser, R. (2017). Ansible: Up and Running: Automating configuration management and deployment the easy way. " O'Reilly Media, Inc.".

[2] Shah, G. (2015). Ansible Playbook Essentials. Packt Publishing Ltd.

[3] Witt, A., & Westling, S. (2023). Ansible in different cloud environments.

[4] McAllister, J. (2017). Implementing DevOps with Ansible 2. Packt Publishing Ltd.

[5] Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. Ieee Software, 33(3), 94-100.

[6] Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016, May). What is DevOps? A systematic mapping study on definitions and practices. In Proceedings of the scientific workshop proceedings of XP2016 (pp. 1-11).

[7] Ryder, T. (2016). Nagios core administration cookbook. Packt Publishing Ltd.

[8] Mongkolluksamee, S., Pongpaibool, P., & Issariyapat, C. (2010, May). Strengths and limitations of Nagios as a network monitoring solution. In Proceedings of the 7th International Joint Conference on Computer Science and Software Engineering (JCSSE 2010). Bangkok, Thailand (pp. 96-101).

[9] Lorge, F., Ricci, S., Iglesias, A., Meloni, M., & Torres, P. Protocolo SNMP.

[10] Li, P. (2010). Selecting and using virtualization solutions: our experiences with VMware and VirtualBox. Journal of Computing Sciences in Colleges, 25(3), 11-17.

[11]Neumann, J. C. (2015). The book of GNS3: build virtual network labs using Cisco, Juniper, and more. No Starch Press.

[12]Sentrio. (16 de Septiembre de 16). Introducción a Jenkins: ¿qué es, para qué sirve y cómo funciona? (Sentrio) Recuperado el 28 de Enero de 2024, de https://sentrio.io/blog/que-es-jenkins/

7 ANEXOS

ANEXO I: CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 20 de Febrero de 2024

De mi consideración:

Yo, FERNANDO VINIVIO BECERRA CAMACHO, en calidad de Directora del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN SERVIDOR SNMP V3 CON DEVOPS PARA LA ADMINISTRACIÓN DE EQUIPOS DE NETWORKING asociado a la IMPLEMENTACIÓN DE SERVICIOS DE NETWORKING MEDIANTE DEVOPS elaborado por el estudiante MATEO DAVID CASTRO CAZAR de la carrera en Tecnología Superior en Redes y Telecomunicaciones, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 9%.

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

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

LINK

https://epnecuador-my.sharepoint.com/:f:/g/personal/fernando_becerrac_epn_edu_ec/Er6a43QpcUFJur6ujtzHgKABRXXqaGg1rMcrE0gm2oc98w?e=4bQH7K

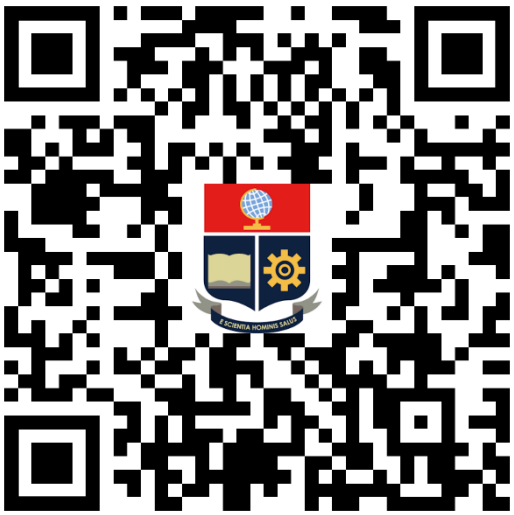
Atentamente,

Fernando Vinicio Becerra Camcho

Docente

Escuela de Formación de Tecnólogos

ANEXO II: ENLACE VIDEO DEL FUNCIONAMIENTO



ANEXO II Código QR de la implementación y pruebas del funcionamiento.

ANEXO III: PLAYBOOK DESPLIEGUE E IMPLEMENTACIÓN DE NAGIOS CORE

```
---
- name: Despliegue de servicio de monitoreo Nagios
  hosts: myhosts
  become: true
  vars:
    nagios_admin_user: nagiosadmin
    nagios_admin_password: 123456
  tasks:
    - name: Actualizar apt cache
      apt:
        update_cache: yes

    - name: Upgrade
      apt:
        upgrade: yes

    - name: Instalacion de paquetes requeridos
      apt:
        name: "{{ item }}"
        state: present
      with_items:
        - autoconf
        - bc
```

- gcc
- gawk
- dc
- libc6
- apache2
- apache2-utils
- php
- wget
- php-gd
- libapache2-mod-php
- libapache2-mod-php7.4
- build-essential
- libgd-dev
- vim
- libmcrypt-dev
- curl
- unzip
- make
- libssl-dev
- libgd-dev
- snmp
- libnet-snmp-perl
- gettext
- openssh-server
- openssl

```
      - perl

  - name: Crear usuario Nagios y grupo
    shell: |
      useradd -m -s /bin/bash nagios
      groupadd nagcmd
      usermod -aG nagcmd nagios
      usermod -aG nagcmd www-data

  - name: Permitir puertos de ssh y http
    shell: |
      ufw allow ssh
      ufw allow http
      ufw reload

  - name: Descargar Nagios Core
    get_url:
      url: https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.5.0/nagios-4.5.0.tar.gz
      dest: /tmp/nagios-4.5.0.tar.gz
      validate_certs: no

  - name: Descargar Nagios plugins
    get_url:
      url: https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.8/nagios-plugins-2.4.8.tar.gz
      dest: /tmp/nagios-plugins-2.4.8.tar.gz
```

```
    validate_certs: no

- name: Descomprimir archivo Nagios Core .tar.gz
  unarchive:
    src: /tmp/nagios-4.5.0.tar.gz
    dest: /tmp/
    remote_src: yes

- name: Descomprimir archivo Nagios plugins .tar.gz
  unarchive:
    src: /tmp/nagios-plugins-2.4.8.tar.gz
    dest: /tmp/
    remote_src: yes

- name: Compilar y configurar Nagios Core
  shell: |
    cd /tmp/nagios-4.5.0
    ./configure --with-httpd-conf=/etc/apache2/sites-enabled
    make all
    make install
    make install-daemoninit
    make install-commandmode
    make install-config
    make install-webconf
```

```
- name: Crear usuario administrador de Nagios Core
  shell: |
    htpasswd -b -c /usr/local/nagios/etc/htpasswd.users "{{ nagios_admin_user }}" "{{ nagios_admin_password}}"

- name: Compilar y configurar Nagios plugins
  shell: |
    cd /tmp/nagios-plugins-2.4.8
    ./configure --with-httpd-conf=/etc/apache2/sites-enabled
    make
    make install

- name: Anadir comando para la utilizacion de los agentes NRPE
  ansible.builtin.lineinfile:
    path: /usr/local/nagios/etc/objects/commands.cfg
    line: |
      define command {
            command_name check_nrpe
            command_line $USER1$/check_nrpe -H $HOSTADRESS$ -c $ARG1$
      }
    insertafter: EOF

- name: iniciar servicio Nagios y reiniciarlo al igual que al servicio apache
  shell: |
    a2enmod rewrite cgi
    systemctl enable apache2.service
```

```
    systemctl start apache2.service
    systemctl restart apache2.service
    systemctl enable nagios.service
    systemctl start nagios.service

- name: Reiniciar maquina objetivo
  reboot:

- name: Permisos en Nagios
  shell: |
    cd /usr/local/nagios/var/rw
    chmod 777 nagios.cmd
```

ANEXO IV: ARCHIVO DE CONFIGURACIÓN CLIENTE LINUX

```
define host {

        use             linux-server
        host_name       cliente
        alias           cliente
        address         192.168.92.146
}

define service {

        use                     generic-service
        host_name               cliente
        service_description     PING
        check_command           check_ping!100.0,20%!500.0,60%
}

define service {

        use                     generic-service
        host_name               cliente
```

```
        service_description     Total Processes
        check_command           check_local_procs!250!400!RSZDT
}

define service {

        use                     generic-service
        host_bane               cliente
        service_description     Current Load
        check_command           check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

define service {

        use                     generic-service
        host_name               cliente
        service_description     SSH
        check_command           check_ssh
        notifications_enabled   0
}
```

ANEXO V: ARCHIVO DE CONFIGURACIÓN SERVIDOR HTTP

```
define host {

        use             linux-server
        host_name       servidor
        alias           servidor
        address         192.168.92.147
}

define service {

        use                     generic-service
        host_name               servidor
        service_description     PING
        check_command           check_ping!100.0,20%!500.0,60%
}

define service {

        use                     generic-service
        host_name               servidor
```

```
        service_description     Total Processes
        check_command           check_local_procs!250!400!RSZDT
}

define service {

        use                     generic-service
        host_name               servidor
        service_description     Current Load
        check_command           check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

define service {

        use                     generic-service
        host_name               servidor
        service_description     SSH
        check_command           check_ssh
        notifications_enabled   0
}

define service {
```

```
        use                     generic-service
        host_name               servidor
        service_description     HTTP
        check_command                   check_http
        notifications_enabled   0
}
```

ANEXO VI: ARCHIVO DE CONFIGURACIÓN ROUTER CISCO

```
define host {

        use                     generic-switch
        host_name               cisco
        alias                   c7200 Router
        address                         192.168.92.149
}

define service {

        use                     generic-service
        host_name               cisco
        service_description     PING
        check_command                   check_ping!200.0,20%!600.0,60%
        check_interval          5
        retry_interval          1
}

define service {

        use                     generic-service
        host_name               cisco
        service_description     Uptime
        check_command                   check_snmp!-C public -o sysUptime.0
}
```

ANEXO VII: ARCHIVO DE CONFIGURACIÓN DEL SWITCH

```
define host {

        use             generic-switch
        host_name       switch
        alias           c3600 switch
        address                 192.168.92.155
}

define service {

        use                     generic-service
        host_name               switch
        service_description     PING
        check_command                   check_ping!200.0,20%!600.0,60%
        check_interval          5
        retry_interval          1
}

define service {

        use                     generic-service
        host_name               switch
        service_description     Uptime
        check_command                   check_snmp!-C public -o sysUptime.0
}
```

ANEXO VIII: PLAYBOOK CONFIGURACIONES CLIENTE LINUX

```
---
- name: Configuraciones para cliente linux
  hosts: myhosts
  become: true
  tasks:

    - name: Archivo para cliente linux
      template:
        src: hostlinux.cfg
        dest: /usr/local/nagios/etc/objects/hostlinux.cfg

    - name: Linea para permitir el archivo de configuracion del cliente linux
      lineinfile:
        path: /usr/local/nagios/etc/nagios.cfg
        line: 'cfg_file=/usr/local/nagios/etc/objects/hostlinux.cfg'
        insertafter: EOF

    - name: Reiniciar servidor Nagios
      systemd:
        name: nagios
        state: restarted
```

ANEXO IX: PLAYBOOK CONFIGURACIONES SERVIDOR HTTP

```
---

- name: Configuraciones para Servidor

  hosts: myhosts

  become: true

  tasks:



    - name: Archivo para el servidor

      template:

        src: servidor.cfg

        dest: /usr/local/nagios/etc/objects/servidor.cfg



    - name: Linea para permitir el archivo de configuracion del servidor

      lineinfile:

        path: /usr/local/nagios/etc/nagios.cfg

        line: 'cfg_file=/usr/local/nagios/etc/objects/servidor.cfg'

        insertafter: EOF



    - name: Reinicar servidor nagios

      systemd:

        name: nagios
```

state: restarted

ANEXO X: PLAYBOOK CONFIGURACIONES ROUTER CISCO

```
---

- name: Configuraciones para monitoreo router cisco

  hosts: myhosts

  become: true

  tasks:



    - name: Archivo para monitoreo router cisco

      template:

        src: router.cfg

        dest: /usr/local/nagios/etc/objects/router.cfg



    - name: Linea para permitir el archivo de configuracion para el monitoreo de router cisco

      lineinfile:

        path: /usr/local/nagios/etc/nagios.cfg

        line: 'cfg_file=/usr/local/nagios/etc/objects/router.cfg'

        insertafter: EOF



    - name: Reiniciar servidor Nagios

      systemd:

        name: nagios
```

state: restarted

ANEXO XI: PLAYBOOK PARA CONFIGURACIONES SWITCH

```
---
- name: configuraciones para monitoreo switch
  hosts: myhosts
  become: true
  tasks:

    - name: Archivo para monitoreo router switch
      template:
        src: switch.cfg
        dest: /usr/local/nagios/etc/objects/switch.cfg

    - name: linea para permitir el archivo de configuracion para el monitoreo del switch
      lineinfile:
        path: /usr/local/nagios/etc/nagios.cfg
        line: 'cfg_file=/usr/local/nagios/etc/objects/switch.cfg'
        insertafter: EOF

    - name: Reinicar servidor nagios
      systemd:
        name: nagios
        state: restarted
```