ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE SERVIDORES CON HERRAMIENTAS DE DEVOPS

CREACIÓN DE SERVIDOR DE LOGS CON SOFTWARE LIBRE CON ANSIBLE

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

ALCARRAZ CASTRO RONY MAURICIO

DIRECTOR: FERNANDO VINICIO BECERRA CAMACHO

DMQ, AGOSTO 2022

CERTIFICACIONES

Yo, RONY MAURICIO ALCARRAZ CASTRO 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.

RONY MAURICIO ALCARRAZ CASTRO

Rony.alcarraz@epn.edu.ec

Rony07alcarraz@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por RONY MAURICIO ALCARRAZ CASTRO, bajo mi supervisión.

FERNANDO VINICIO BECERRA CAMACHO

DIRECTOR

Fernando.becerra@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmo que el trabajo de integración curricular aquí descrito, así como el producto resultante 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.

RONY MAURICIO ALCARRAZ CASTRO

DEDICATORIA

Mi tesis se la dedico con mucho cariño a mis padres por sus sacrificios, por sus esfuerzos, por su apoyo y por creer en mí siempre. Pese a los obstáculos que hemos tenido como familia siempre los cinco hemos logrado salir adelante y por eso este logro no solo es mío, sino que es nuestro ya que sin ustedes no sería posible.

En memoria a mi tío José Alcarraz, ya que fue una persona muy importante en mi vida. Mas que un tío, fuiste un segundo padre que se preocupaba por mí. Tu anhelo siempre fue que estudiara y que saliera adelante para convertirme en una persona profesional, tú me encaminaste desde pequeño a ser una persona trabajadora, responsable, respetuosa, deportista y sobre todo a sonreír a la vida.

No te importo tener una enfermedad para disfrutar plenamente de la vida y tampoco te limitaste a ser feliz. Gracias por el apoyo incondicional que les distes a mis padres. Ahora, sé que estas en un lugar de paz y cuidándonos desde el cielo a mí y a toda la familia.

Sé que estarás muy orgulloso de este logro.

AGRADECIMIENTO

Ante todo, agradezco a mi padre celestial YAHWEH por la salud y la vida de mis padres y la mía. Él ha permitido que mis padres guíen mi camino en el trayecto de esta nueva meta.

A mis padres, por ser el pilar fundamental para cumplir mis metas. Sus experiencias, sus dificultades y sus problemas me han permitido comprender que el estudio es la mejor herencia que me pueden dejar. Mediante sus consejos, sus esfuerzos, su amor, su comprensión y su educación he logrado convertirme en una persona de bien.

Al Ing. Fernando Becerra por el apoyo que me brindo en la realización del presente trabajo y también por las enseñanzas que me compartió en el transcurso de la carrera.

A mis primeros maestros que forjaron mi habito de estudio entre ellos el Lic. Rodrigo Velazco.

A mis amigos por la ayuda que me han brindado tanto en la vida estudiantil como en la vida personal. Por ser parte de buenas, malas e inolvidables experiencias de las fiestas a las que salíamos a divertirnos, porque sonreír es uno de los objetivos de la vida

ÍNDICE DE CONTENIDOS

RESUMEN

El presente proyecto de titulación tiene como propósito crear un de servidor de logs en *software* libre con ansible, para ello se realizará un análisis de Ansible y de al menos 3 plataformas, que utilicen este servicio en el área de TI para el despliegue del servidor de logs. Luego, se seleccionará la mejor plataforma para su implementación.

Con Ansible implementado en las máquinas virtuales, se creará y configurará nuevas instancias de almacenamiento para el despliegue del *playbook* que contendrán el proceso de implementación del servidor de logs, al igual que el indexador de datos.

A continuación, se detalla la estructura del presente trabajo, el cual posee 5 capítulos que se dividen de la siguiente manera:

El capítulo uno contiene una descripción del proyecto, objetivo general, objetivos específicos, el alcance del proyecto y el marco teórico.

El capítulo dos contempla la metodología aplicada para el desarrollo del proyecto de titulación.

El capítulo tres consiste en los resultados, aquí se presenta el análisis de Ansible, algunas plataformas que almacenan los logs y la selección de una plataforma para ser implementada. Una vez seleccionado el servidor, se presenta su implementación de forma manual y después de forma automatizada, considerando que este proceso se repite para el indexador.

Finalmente, en el capítulo cuatro y cinco se encuentran las conclusiones y recomendaciones respectivamente, las cuales fueron obtenidas durante el desarrollo del presente trabajo de titulación.

PALABRAS CLAVE: Ansible, Splunk, *playbook*, servidor de logs, indexador, SLSE, ISUF y automatización.

ABSTRACT

The current degree project, aims to create a free software log server with ansible, for this an analysis of Ansible and at least 3 platforms that use this service in the IT area for the deployment of log servers. Then, the best platform for its implementation will be selected.

With Ansible deployed on the virtual machines, new storage instances will be created and configured for the playbook deployment that will contain the log server deployment process, as well as the data indexer.

Next, the structure of the present work is detailed, which has 5 chapters that are divided in the following way:

Chapter one contains a description of the project, general objective, specific objectives, the scope of the project and the theoretical framework.

Chapter two contains the methodology applied for the development of the degree project.

Chapter three consists of the results, here we present the analysis of Ansible, some platforms and the selection of a platform to be implemented. Once the server is selected, its implementation is presented manually and then in an automated way, considering that this process is repeated for the indexer.

Finally in chapter four and five are the conclusions and recommendations respectively, which were obtained during the development of this degree work.

KEYWORDS: *Ansible, Splunk, playbook, log server, indexer, SLSE, ISUF and automation.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el presente componente se creó un servidor de logs en *software* libre con Ansible, para ello, en primera instancia se presenta un análisis sobre Ansible (herramienta de automatización de DevOps), por lo que se detalla su funcionamiento, su estructura y las dependencias para la elaboración de los *playbooks*. Después, se presenta una investigación sobre los siguientes servidores de logs: Splunk, Graylog y Papertrail, donde se establece la definición, características y dependencias para su funcionamiento.

Mediante una tabla comparativa, se analizan las características de las plataformas mencionadas y, en consecuencia, se procede a la selección de Splunk como la plataforma para el levantamiento del servidor, considerando las ventajas de escalabilidad, eficiencia, rapidez de instalación y funcionamiento que tiene ante los otros servidores.

Después, se procede a diseñar los *playbooks* para el Servidor de Logs Splunk Enterprise (SLSE) y el Indexador Splunk Universal Forwarder (ISUF), en donde se inicia con la configuración de la máquina orquestadora y se revisa los requisitos de los nodos clientes (herramientas o paquetes adicionales). Después de establecer la conexión entre servidor y cliente, se ejecuta una prueba de funcionamiento (ping-pong), y es entonces que se procede con la implementación de estos de forma manual.

El proceso de implementación de forma manual es importante debido a que se necesita comprobar la operatividad del SLSE y del ISUF previo al proceso de automatización (creación de los *playbooks*). Al referirse que la implementación se realizará de forma manual, quiere decir que el proceso se lo realizará desde la CLI (*Command Line Interface*).

En base a los comandos del proceso manual se implementa dos scripts que son dependencias de los *playbooks* para la automatización del SLSE y del ISUF. Posteriormente se ejecuta pruebas de funcionamiento, para lo cual se debe crear dos máquinas virtuales únicamente con la conexión Ansible y acceder a la máquina orquestadora para compilar los *playbooks*.

Finalmente, se verifica que el SLSE y el ISUF se encuentran operando, para ello, primero se ingresa al navegador y se accede a la dirección del servicio con el puerto de escucha correspondiente del SLSE, después de verificar la operatividad, se debe

ingresar al menú y verificar la conexión de los indexadores que tienen las máquinas clientes.

1.1 Objetivo general

Implementar servidores con herramientas de DevOps

1.2 Objetivos específicos

- Analizar el Ansible y sus características.
- Diseñar el *playbook* para implementar un servidor de logs.
- Implementar un *playbook* para el servidor de logs.
- Verificar el funcionamiento del *playbook*.

1.3 Alcance

El presente proyecto de titulación tiene como alcance el análisis de Ansible, sus características, funcionamiento y la estructura de un *playbook*. Conjuntamente, se realizará una investigación sobre las plataformas de servidores de logs que tienen mayor demanda en la actualidad, además se describirán las dependencias de cada una de ellas.

En base a las mejores características de las plataformas de servidores de logs, se procederá a la selección de una de las opciones, para posteriormente, implementarla sobre una máquina virtual.

Consecutivamente, se configurará la herramienta de orquestación para evitar inconvenientes con el proceso de automatización. Una vez hecha la configuración, es necesario realizar pruebas de conexión con los nodos clientes, para ello se realizará la prueba de ping-pong. A continuación, se instalará el servidor de logs mediante un proceso de implementación manual, para descartar problemas en la elaboración de los *playbooks*.

Finalmente, los comandos ejecutados de forma manual, se transcriben a dos scripts (uno para implementar el servidor y otro para el indexador), para automatizar el proceso mediante Ansible. Una vez realizada la instalación, se comprobará el funcionamiento con ayuda de un nodo cliente y se verificará la indexación de datos en la plataforma del servidor.

1.4 Marco Teórico

Automatización

La automatización permite agilizar la infraestructura tecnológica con la finalidad de aumentar la productividad de los servicios del área de TI. La automatización proporciona información de la escalabilidad de una empresa, reduce errores y mejora la seguridad. Además, permite que los trabajadores del área de TI se enfoquen en el trabajo estratégico, mas no del administrativo [1].

La primera solución a considerar para iniciar este proceso es cuando se realiza la migración a nuevos equipos físicos, sin embargo, existen algunos problemas asociados con los costos de implementación y el tiempo de desarrollo en el proceso de migración [2]. Ante estos problemas mencionados se puede optar por la virtualización, que es una estrategia óptima para el despliegue de servicios de TI que tienen limitaciones en los recursos de hardware.

Mediante la automatización se reduce el tiempo de implementación de aplicaciones o *software* que maneja el área de TI dentro de una empresa. La automatización tiene como finalidad la creación de instrucciones y procesos repetibles con el cual se facilita las labores diarias a los trabajadores internos y se mejora la prestación de servicios a los clientes [1].

Virtualización

Bob Muglia quien fue especialista en investigación y desarrollo de *software* de Microsoft Corporation. Bob Muglia señaló: “La Virtualización es una estrategia para desplegar los recursos del ordenador en diferentes capas aisladas - hardware, *software*, datos, red, almacenamiento unas de las otras [3]”. Es decir que la virtualización es una metodología que va permitir dividir de manera eficiente los recursos que brinda un ordenador.

La virtualización permite centralizar diferentes tareas en una sola máquina y ejecutarlas al mismo tiempo [3]. Incluso otro beneficio es el ahorro de espacio físico, energía eléctrica y recursos computacionales en el caso de virtualización de servidores.

Los elementos que integran el funcionamiento de la virtualización son la máquina virtual y el hipervisor. El hipervisor es la parte del *software* que abastece las características solicitadas como el almacenamiento del disco, la tarjeta de red y CPU al momento de la creación de una máquina virtual, es decir es el responsable de administrar los recursos de la máquina física. En cambio, la máquina virtual es la emulación de una plataforma virtual que se crea a partir del *software* [3].

Ansible

Ansible es conocida como una herramienta de gestión de la configuración, pero en realidad se puede utilizar para diferentes tipos de escenarios tales como [4]:

- Aprovisionamiento: abastece los recursos computacionales que requieren los nodos clientes registrados en el inventario de Ansible.
- Despliegue de aplicaciones: posee la herramienta Ansible Tower con la cual se puede seguir el proceso de implementación de una aplicación.
- Seguridad y cumplimientos: con ayuda de los *playbooks* de Ansible se puede crear reglas de firewall y políticas de seguridad.
- Orquestación: es el responsable de la automatización en los despliegues de diferentes aplicaciones o servicios.
- Gestión de la configuración: entrega información del *software* y *hardware* de la empresa que administra Ansible.

Ansible trabaja con el modelo de inserción (Push Model), es decir la máquina orquestadora abastece directamente los recursos computacionales hacia los nodos clientes, sin la necesidad de alertas y de esta manera se evita la instalación de aplicaciones adicionales en los nodos administrados [5]. Es por ello, que el presente proyecto solo necesita la configuración de la máquina orquestadora, mas no de las máquinas clientes, pues únicamente se debe cumplir algunos requisitos.

Servidor de logs

El término log hace referencia a un registro de sucesos que ocurren durante un lapso de tiempo en un dispositivo en particular (nodo cliente), para ello se necesita de una máquina que almacene los registros (Servidor) [6]. Estos registros permiten analizar el funcionamiento del sistema y aplicaciones que maneja el cliente. Los datos de registro son la información extraída de un mensaje de registro para decirle por qué se generó el mensaje de registro [7].

En particular, el sistema Unix tiene mensajes de cierre e inicio de sesión de usuario, los cortafuegos tienen mensajes de aceptación y denegación de ACL (*Access Control List*), los sistemas de almacenamiento en disco generan mensajes de registro cuando ocurran fallas o, en algunos casos, cuando el sistema perciba una falla inminente [7].

En base a la información recolectada el administrador puede encontrar posibles soluciones a problemas presentes y posteriores, pero previo a ello se debe entender el

significado de cada campo del registro de logs. En la Tabla 1.1, se presenta los significados de los diferentes niveles de logs que puede contener un registro.

Tabla 1.1 Niveles de logs,[8].

Niveles de logs	Significado
Off	Registro desactivado.
Fatal	Tarea fallida, el servidor o aplicación no funciona.
Serious	Tarea fallida, el servidor o aplicación está funcionando. También indica un error irrecuperable.
Warning	Error potencial y pérdida de recursos.
Audit	Evento significativo que afecta al servidor.
Information	Información general de la tarea
Config	Configuración de la tarea
Detalil	Información general de la subtarea
Fine	Información de seguimiento general
Thinner	Información detallada del seguimiento de entrada y salida de valores.
Finest	Información de seguimiento con los detalles necesarios para eliminar problemas
Everybody	Se registran todos los sucesos.

2 METODOLOGÍA

Al inicio se investigó información acerca de DevOps, además se detallaron sus características, funciones y métodos para la monitorización de datos; con esto se obtuvo conocimiento de la herramienta Ansible para la automatización del servidor de logs. Además, se investigaron varias plataformas de servidores de logs para entender cómo funcionan.

Al tener varias plataformas de servidores de logs que utilizan Ansible como herramienta de orquestación, se realizó una comparación de las plataformas. De esta manera, se seleccionó una para implementar el servidor en una máquina virtual. Además, se utilizó un *software* en otra máquina virtual que permitirá indexar los datos y con ello verificar el funcionamiento del servidor.

Una vez determinado el servidor de logs, en primer lugar, se procedió a la implementación de la herramienta orquestadora Ansible, para lo cual se utilizó un hipervisor (VMware) que permitió al usuario ejecutar máquinas virtuales. Para el funcionamiento de Ansible, se empleó dos máquinas virtuales que cumplieron el rol de orquestador y cliente. En segundo lugar, se procedió a la implementación del servidor y del indexador de forma manual y, posteriormente en base al proceso manual, se crearon

y se configuraron los directorios de Ansible para la automatización del servidor y del indexador.

Para las pruebas de funcionamiento de la automatización del servidor y del indexador, se inició con la ejecución del *playbook* que contiene el despliegue del servidor de logs. Después, se ejecutó el otro *playbook* que contiene el despliegue de los indexadores para nodos clientes. Finalmente, se pudo verificar la operatividad del servidor, así como la conectividad entre servidor-cliente.

3 RESULTADOS

En primer lugar, se presenta el análisis de Ansible y sus características, una vez comprendido el funcionamiento de Ansible, se investigan las plataformas de servidores de logs (Splunk, Graylog y Papertrail), indicando las características y dependencias que necesitan para su funcionamiento. Además, se realiza una comparación de las tres plataformas y se selecciona una de ellas para su implementación.

Luego se procede a configurar la máquina orquestadora y verificar los requisitos de la máquina cliente, después de esto, se realiza una prueba básica de funcionamiento (prueba de ping-pong). Por otro lado, se implementa el SLSE y el ISUF de forma manual.

Después, se procede a desarrollar los *playbooks* para levantar el SLSE y el ISUF de manera automatizada. Finalmente, se verifica el funcionamiento de los *playbooks* implementados y se visualiza la conectividad entre el servidor y los nodos clientes.

3.1 Análisis de Ansible y sus características.

DevOps

El termino DevOps fue empleado por primera vez por Yhens Wasna y Patrick Debois en la charla sobre “Infraestructura Ágil”, en la Conferencia Agile 2008 en Toronto. DevOps es un acrónimo compuesto por dos palabras provenientes del inglés, la primera es “*Development*” que significa desarrollo y “*Operations*” que significa operaciones [9].

DevOps nace por la necesidad de acelerar los procesos de *software* existentes en una organización, debido a que el objetivo se orientaba en mejorar los procesos del personal de desarrollo y operaciones para ofrecer una nueva experiencia al cliente. Entre los beneficios que se pueden obtener son un mayor control en los cambios del mercado y alcanzar los objetivos de la organización [9].

Esta tecnología se centra en el cambio de los procesos que existe en el área de TI, mediante la colaboración de los equipos de desarrollo y operaciones, usando herramientas de automatización para obtener sistemas eficientes, como se muestra en la **Figura** 3.1. Los métodos más usados en DevOps que permiten optimizar el desarrollo del *software* y además mejorarlo son: Scrum, Agile y Kanban. Estos métodos permiten dividir los procesos para monitorizar cada uno de ellos y determinar avances o retrasos de cada tarea [9].

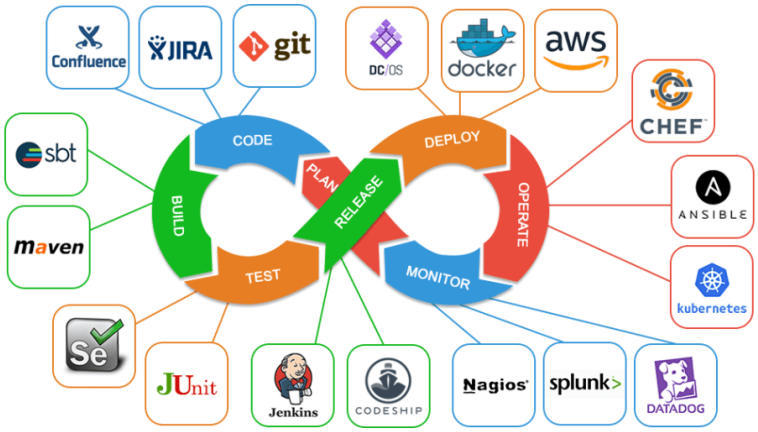

Figura 3.1 Herramientas de Automatización de DevOps [10].

Scrum es un marco de trabajo el cual está basado en una metodología Agile para el desarrollo de *software* de un proyecto. Una metodología Agile es un proceso de cambio que abarca una nueva forma de trabajo, es decir busca reaccionar rápidamente sobre el cambio y ser flexible con el cambio, pero sin sacrificar la estabilidad del proyecto [9].

Agile es una metodología iterativa e incremental que consiste en la división de un proyecto de *software* en pequeñas partes, lo que permite concluir el proyecto en poco tiempo y de manera eficiente[9]. Se enfoca en la interacción de los individuos, mas no en las herramientas y procesos, es decir trabaja en colaboración con los clientes y acepta el cambio [9].

Kanban es una metodología que se apoya de un marco visual para gestionar un proyecto a medida que avanza, es decir identifica falencias en los procesos para encontrar soluciones, optimizar el flujo de trabajo y limitar el progreso del trabajo [9].

Debido a que el objetivo de DevOps es reducir el tiempo de desarrollo de las aplicaciones y apresurar la creación de nuevas funciones basándose en las exigencias de los usuarios, se debe cumplir con las siguientes fases [9]:

- Planificación

- Codificación
- Compilación
- Pruebas
- Liberación
- Despliegue
- Funcionamiento
- Monitoreo

Ansible

Ansible es una tecnología de aprovisionamiento de sistemas de DevOps que permite acondicionar la infraestructura, administrar las aplicaciones y organizar los procesos del área de TI. Gracias a la automatización de procesos que brinda Ansible, se puede instalar de forma rápida un sistema de *software* a un grupo de usuarios, optimizar la seguridad en base a la información recopilada de los usuarios y actualizar los sistemas del área de TI [11].

Esta herramienta permite a las organizaciones adaptarse a los nuevos modelos de negocio que se está desarrollando en base a la tecnología. Sin embargo, este proceso implica que el equipo de TI deba tener conocimiento de Ansible, claramente el beneficio se reflejará en la rapidez de aprovisionamiento y menor trabajo para el área de TI [11]. Es más eficiente ejecutar un *playbook*, que contenga una operación específica para un grupo de usuarios, que conectarse máquina a máquina e ir realizando la tarea establecida.

Para el funcionamiento de Ansible se necesita de dos nodos o máquinas, el primero se lo denomina orquestador o nodo administrador y el segundo nodo cliente. El nodo administrador es el encargado de insertar a los nodos clientes algunos programas conocidos como módulos, estos módulos van a permitir la automatización en los nodos clientes [11]. Se debe tener en cuenta que la conexión de Ansible es entre usuarios Ansible, es decir, tanto en el nodo administrador como en el nodo cliente se debe tener un usuario Ansible.

Ansible es una herramienta sencilla de instalar, sin embargo, existen requisitos tanto para el nodo administrador como para los nodos clientes. A continuación, se presenta los requisitos de Ansible [12].

Requisitos para el nodo administrador:

- La máquina debe tener Python, ya sea una versión actualizada o antigua.

- Si el nodo de control necesita de ajustes adicionales, se debe configurar la documentación del plugin.
- Considerar que un nodo de control tiene mayor eficiencia en la nube.

Requisitos para el nodo cliente:

- La máquina debe tener Python, pero es suficiente si tiene una versión antigua.
- Realiza una conexión SSH para enviar los módulos SFTP (Protocolo de transferencia segura de archivos), los cuales permiten la automatización.
- Si el protocolo SFTP no está disponible se debe modificar el archivo ansible.cfg y cambiar al protocolo SCP (Protocolo de copia segura).

Después de establecer la conexión por SSH, Ansible utiliza archivos con la extensión YML, debido a que se puede ejecutar un lenguaje de programación de alto nivel (YAML). Es decir que cualquier persona puede programar sin necesidad de un conocimiento profundo de programación [13].

Los archivos YML son denominados *playbooks*, que permiten organizar y establecer procesos para el área de TI. Sin embargo, Ansible tiene una sección de módulos la cual permite ejecutar comandos desde el terminal del nodo administrador [13]. La diferencia entre utilizar un *playbook* y los módulos de Ansible, es la automatización, ya que los módulos solo representan tareas específicas en un orden establecido. En cambio, el *playbook* detecta que el estado del sistema no coincide con las órdenes del *playbook* y procede abastecer los recursos faltantes [13].

A continuación, en la **Tabla** 3.1 se presenta algunos módulos importantes para el proceso de automatización mediante *playbooks*:

Tabla 3.1 Módulos de Ansible, [14].

Módulos	**Función**
Gestión de paquetes	Permite instalar cualquier paquete en un sistema.
Servicio	Permite iniciar, detener y recargar paquetes instalados.
Copia	Permite copiar un archivo de la máquina local o remota a una ubicación en la máquina remota.
Depuración	Permite visualizar errores durante la ejecución y es útil para eliminar variables o expresiones sin detener el *playbook*.
Archivo	Permite administrar el archivo y sus propiedades.
Lineinfile	Permite gestionar líneas en un archivo de texto.
Git	Permite administrar las comprobaciones de github de los repositorios para ejecutar archivos o *software*.
Cli-*command*	Permite remitir configuraciones basadas en texto a equipos de red a través de la CLI.

Archivo	Permite crear un archivo comprimido de uno o más archivos. El sistema asume que la fuente de compresión existe en el destino.
Comando	Permite tomar la denominación del comando seguido de una lista de instrucciones separados por espacios.

El uso de estos módulos directamente desde Ansible es apropiado para operaciones sencillas, sin embargo, no son adecuados para escenarios complejos de administración configuración y orquestación. Para estos casos es necesario utilizar los *playbooks* los cuales permitirán cambiar las tareas administrativas largas y complejas en rutinas fácilmente repetibles con resultados predecibles y exitosos [15].

Debido a que los *playbooks* son archivos de texto escritos en formato YAML necesitan cumplir algunos parámetros como los que se enlista a continuación [15]:

- Empezar con tres guiones (---).
- Sangría adecuada utilizando espacios y no tabuladores.

Además, entre su estructura se debe definir algunos conceptos importantes como los siguientes [15]:

- *hosts*: se define las direcciones IP o el grupo que contiene a los nodos administrados.
- *tasks*: se define las tareas a realizar, en esta sección se utiliza los módulos de Ansible.
- *become*: se define si las tareas se las ejecutara en modo root.

Para ejecutar los *playbooks* se debe utilizar el comando ansible-*playbook* en la máquina orquestadora acompañado del nombre del archivo (ansible-*playbook* ejemplo.yml). Posteriormente, si las tareas tuvieron éxito los resultados a obtener serán visualizados con un mensaje de “OK”, caso contrario se visualizarán los errores que está impidiendo ejecutar la tarea en específica.

Plataformas de Servidores de Logs

Splunk

Splunk es un *software* que busca innovar los procesos de TI, adaptase a los avances tecnológicos y optimizar la seguridad de los datos que manejan las empresas. Además, busca cambiar el concepto de datos, es decir, no solo quiere que sean un registro de lo que sucede, sino que se conviertan en acciones [16].

Entre los productos que brinda Splunk está el servidor denominado Splunk Enterprise, esta es una plataforma que muestra la información más importante de los datos monitorizados. Simplifica la recopilación, gestión y búsqueda de cantidades masivas de datos generados por máquinas, servidores, aplicaciones y dispositivos móviles. Se utiliza ampliamente en el análisis web, gestión de aplicaciones, cumplimiento de tareas y seguridad [16].

Esta plataforma permite retener, registrar y comparar datos provenientes de diversos clientes en tiempo real. Dichas funcionalidades, ayudan a reconocer patrones de datos, diagnosticar problemas potenciales y generar soluciones. Los datos se almacenan por medio de un *syslog* o algunos de sus derivados denominados *forwarder* [17].

Splunk tiene una interfaz web intuitiva donde el personal del área del TI puede revisar los logs o el panel de control denominado *dashboards*, para generar informes, gráficos y alertas [17].

Graylog

Graylog es una herramienta de análisis que permite centralizar el análisis de eventos, es decir, desde un único lugar, se puede observar el estado de todos los clientes conectados; en consecuencia, no es necesario instalar ficheros de logs en cada uno de ellos [18].

El sistema de Graylog se compone al menos de las siguientes dependencias como Graylog Server, MongoDB y Elasticsearch, como en la **Figura** 3.2. Cada uno de estos componentes es necesario y no puede sustituirse por ninguna otra tecnología [18]. La implementación de este servidor puede estar configurada de tres maneras como mínima, multimodo simple y multimodo complejo [19].

En cualquiera de las configuraciones se puede realizar una búsqueda personalizada en base a consultas estructuradas, es decir, mediante un lenguaje de programación estandarizado se administran las bases de datos y se verifica la operatividad del sistema [19].

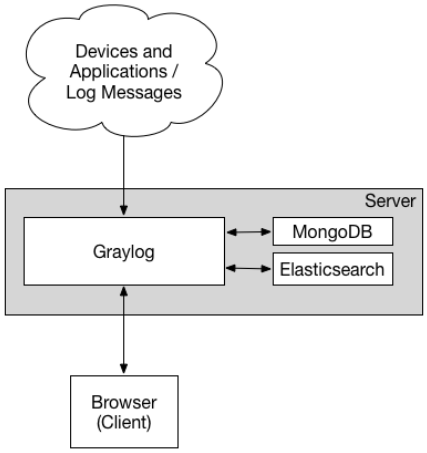

Figura 3.2 Estructura de Graylog [19].

Papertrail

Papertrail es una herramienta que administra los registros de diferentes máquinas y se implementa únicamente en la nube. Está enfocada en una resolución rápida de problemas de infraestructura y aplicaciones de una organización. Papertrail consolida los registros en un solo lugar, para realizar un seguimiento de las actividades en tiempo real y encontrar soluciones a problemas existentes [20].

Es un servicio basado principalmente en el estándar *syslog framework*. A cada cliente se le asigna un puerto, el cual es utilizado para enviar datos de registro del sistema. Esta no es la forma más segura de registro, ya que por defecto la conexión no es protegido con SSL/TLS y todos los datos se envían a través de Internet sin cifrar. Una falla adicional en la seguridad es que los dispositivos no están autorizados antes de poder registrar datos [21].

Si se comete un error tipográfico en un archivo de configuración, es muy posible que un usuario de PaperTrail esté enviando sus datos de registro a otro cliente. PaperTrail permite iniciar sesión a través de syslog para sistemas basados en UNIX, Linux y Mac OS X. Funciona como un sistema servidor de registro y puede registrar cualquier dato que se pueda enviar de forma nativa al registro del sistema, como firewalls, conmutadores u otro equipo de red [21].

Selección del Servidor de logs

Una vez analizado el funcionamiento y las dependencias de cada uno de los servidores anteriormente descritos, se concluye que la mejor plataforma para levantar el servidor es Splunk, debido a:

- Mayor rapidez en la configuración e implementación del servidor.
- No tiene dependencias en el despliegue del servidor.
- No se necesita *software* adicional en las máquinas clientes para analizar los logs, pero si se desea se puede indexar información de máquinas remotas con ayuda de un *software* indexador.

En la **Tabla** 3.2 se presenta a detalle las características analizadas para la selección del servidor.

Tabla 3.2 Comparación de plataformas de servidores, [22].

Funciones	Splunk Enterprise	Graylog	Papertrail
Archivado y retención	SI	SI	SI
Auditoria	SI	SI	SI
Seguimiento	SI	SI	NO
Visualización de datos	SI	SI	SI
Registro de eventos	NO	SI	SI
Recopilación de registros	SI	SI	SI
Gestión de remediación	SI	SI	NO
Registros del servidor	SI	SI	SI
Alertas de umbral	SI	SI	SI
Aplicación de escritorio Mac	SI	SI	NO
Aplicación de escritorio Windows	SI	SI	NO
Aplicación de escritorio Linux	NO	SI	NO
Aplicación local Linux	SI	SI	NO
Aplicación local Windows	SI	SI	NO
Nube	SI	SI	SI
Dependencias	NO	SI	NO
Dificulta de implementación alta	NO	SI	NO

3.2 Diseño del *playbook* para la implementación un servidor de logs

En este proceso de diseño, se debe considerar las dependencias del *playbook*, es decir las herramientas que necesita el *playbook* para su funcionamiento. Para ello se instaló 2 máquinas virtuales con el sistema operativo Ubuntu 20 utilizando el hipervisor VMware. La primera máquina virtual es destinada a cumplir el rol de orquestador y la segunda máquina virtual a cumplir el rol de cliente. Considerando que solo la máquina orquestadora debe tener instalado el *software* Ansible.

Configuración de la máquina orquestadora

En la máquina orquestadora se debe aprovisionar los elementos necesarios para el funcionamiento de Ansible, los cuales se presentan a continuación:

- En la **Figura** 3.3, se presenta el comando con el cual se ejecuta la actualización de los paquetes.

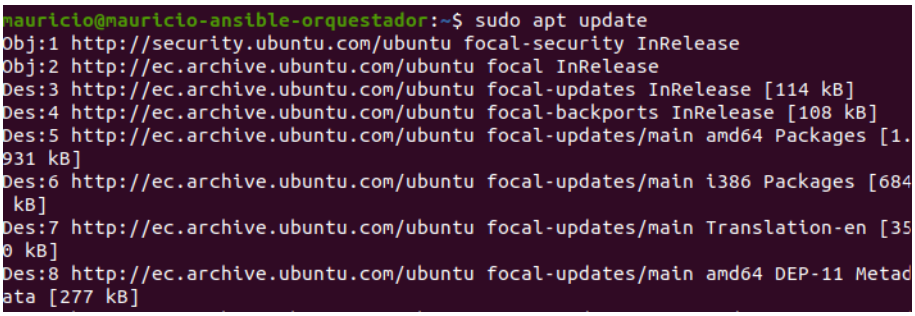

Figura 3.3 Actualización de paquetes del orquestador.

- En la **Figura** 3.4, se observa el comando que se emplea para la instalación de Python en su versión mínima.

Figura 3.4 Instalación de Python en el orquestador.

Después de aprovisionar a la máquina orquestadora, se procede a optimizar la gestión de los repositorios desde donde se instala el *software* Ansible, por lo que se debe instalar el paquete *software-properties-common* con el comando observado en la **Figura** 3.5**.**

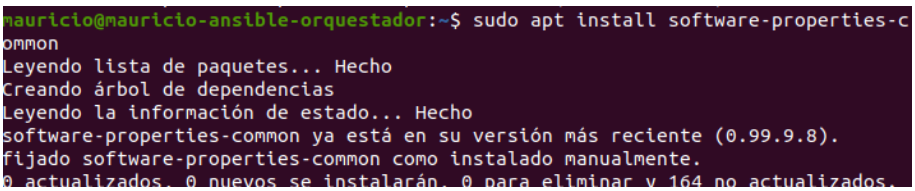

Figura 3.5 Optimización de repositorios para Ansible.

Posteriormente, se debe incluir los archivos de paquetes personales de Ansible (PPA) en la lista del sistema de la máquina orquestadora, como se presenta en la **Figura** 3.6.

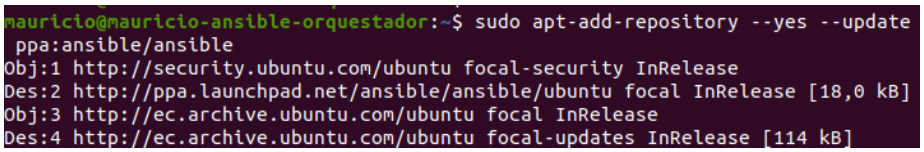

Figura 3.6 PPA de Ansible.

A continuación, en la **Figura** 3.7 se presenta el comando utilizado para la instalación del *software* Ansible.

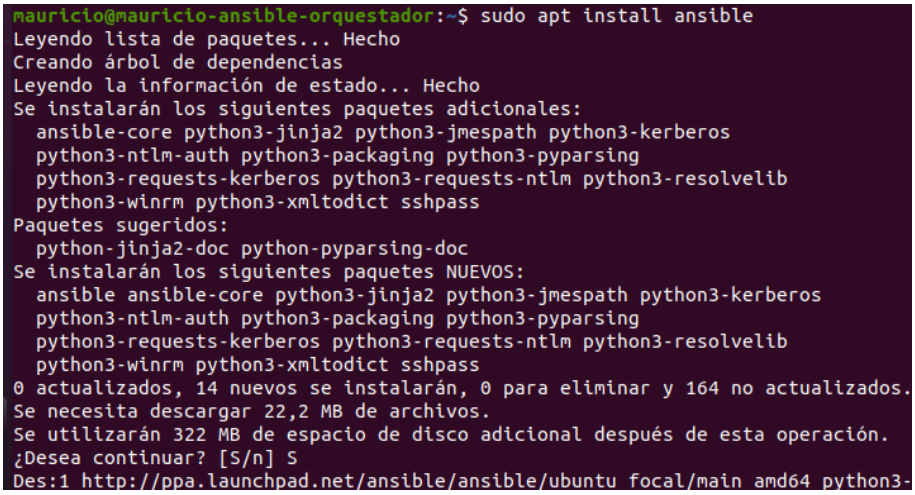

Figura 3.7 Instalación de Ansible.

Por seguridad y para optimizar el funcionamiento de Ansible, es necesario crear un usuario denominado “Ansible” en la máquina orquestadora, como se visualiza en la **Figura** 3.8**.**

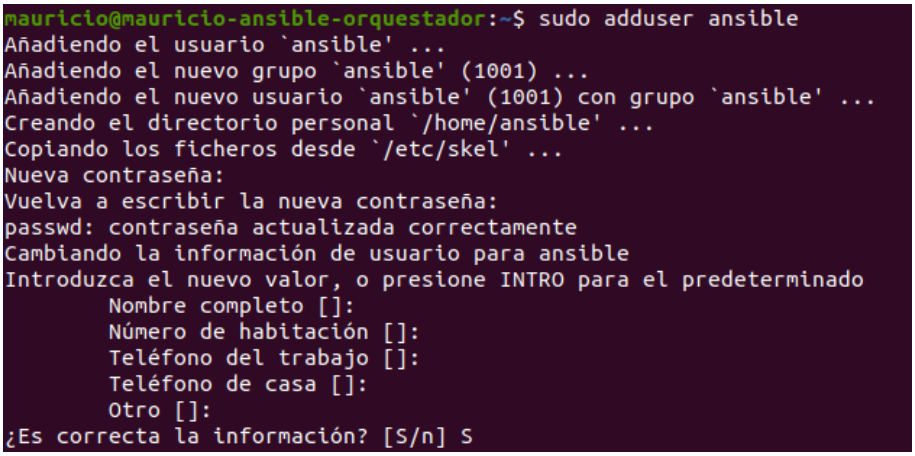

Figura 3.8 Creación de usuario Ansible en el orquestador.

Para registrar las direcciones de los clientes que administra Ansible, primero se debe ingresar al directorio en donde se encuentra Ansible y editar el archivo de hosts, como se presenta en la **Figura** 3.9.

```
mauricio@mauricio-ansible-orquestador:~$ cd /etc/ansible/
mauricio@mauricio-ansible-orquestador:/etc/ansible$ ls
ansible.cfg  hosts  roles
mauricio@mauricio-ansible-orquestador:/etc/ansible$ sudo gedit hosts
```

Figura 3.9 Ingreso y edición del directorio hosts.

En la **Figura** 3.10 se observa el registro de un grupo denominado *clientSplunk* el cual contiene la dirección IP de la máquina cliente. Es importante considerar la sintaxis del registro, ya que es fundamental al momento de emplear *playbooks*.

```
1 [clientSplunk]
2 192.168.242.142
```

Figura 3.10 Registro de la máquina cliente.

Para que el usuario Ansible pueda tener privilegios de root, se debe instalar el paquete sudo con el comando que se observa en la **Figura** 3.11.

```
mauricio@mauricio-ansible-orquestador:~$ sudo apt install sudo
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
sudo ya está en su versión más reciente (1.8.31-1ubuntu1.2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 164 no actualizados.
mauricio@mauricio-ansible-orquestador:~$
```

Figura 3.11 Instalación del paquete sudo.

Después, con el comando “sudo visudo”, se procede añadir el usuario Ansible, en donde se establece la regla de no solicitar contraseña, como se presenta en la **Figura** 3.12.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
ansible ALL=(ALL:ALL) NOPASSWD: ALL
```

Figura 3.12 Anulación de contraseña para el usuario Ansible.

Luego, para establecer la conexión SSH entre orquestador y cliente, se debe generar una clave SSH en el orquestador, pero en el usuario Ansible, como se visualiza en la **Figura** 3.13.

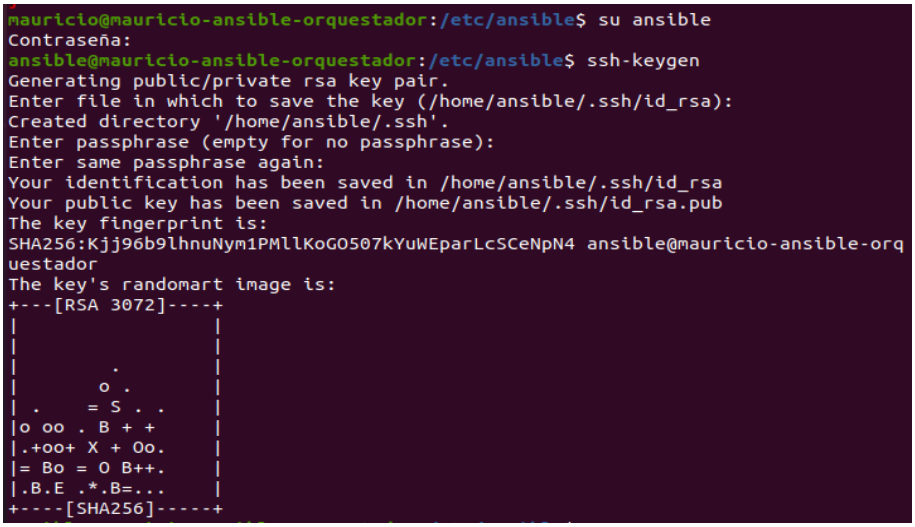

Figura 3.13 Generación de clave SSH.

Finalmente, mediante el comando mostrado en la **Figura** 3.14, se procede a compartir la clave SSH desde el orquestador al nodo cliente. Hay que considerar que la clave SSH debe ser la misma para cualquier nodo cliente.

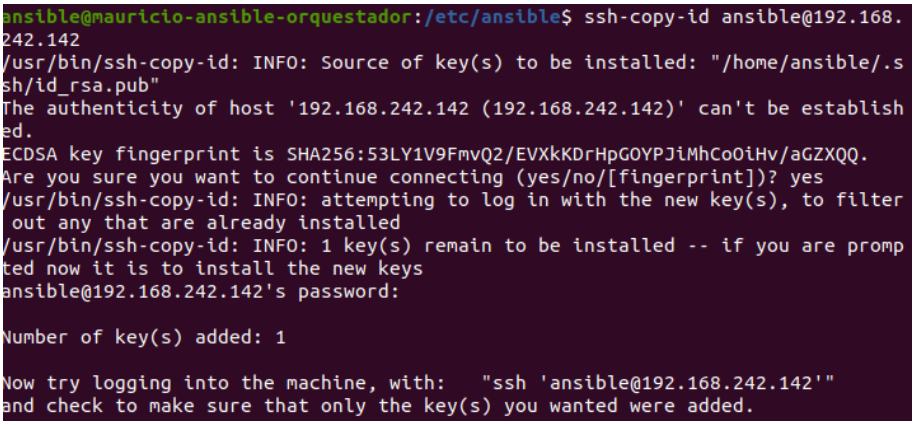

Figura 3.14 Compartir clave SSH.

Requisitos de la máquina cliente

En la máquina cliente se debe revisar que contenga los paquetes necesarios para la conexión SSH que utiliza Ansible, los cuales se presentan a continuación:

- Si la máquina no tiene actualizado los repositorios, en la **Figura** 3.15 se presenta el comando con el cual se debe ejecutar la actualización de los paquetes.

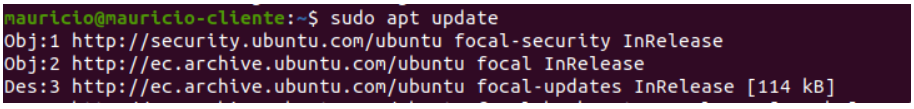

Figura 3.15 Actualización de paquetes del cliente.

- Si la máquina no tiene el paquete de Python, en la **Figura** 3.16 se observa el comando que se debe emplear para la instalación de Python ya sea en su versión mínima o la más actualizada.

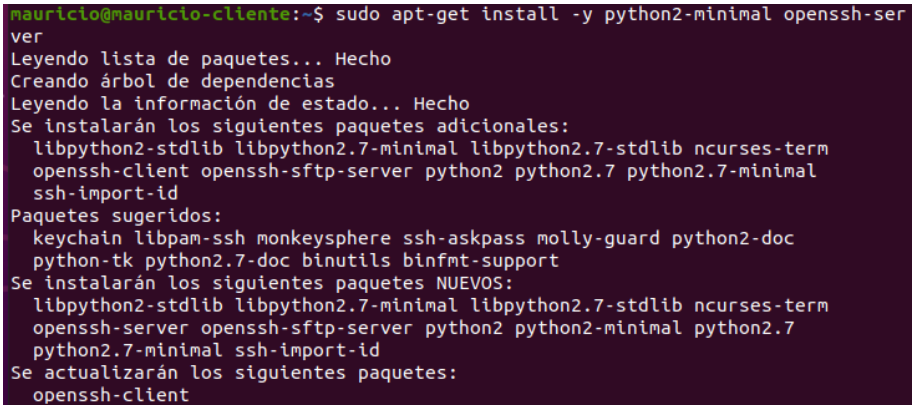

Figura 3.16 Instalación de Python en el cliente.

Por seguridad y para optimizar la conexión de Ansible, se creó un usuario denominado "Ansible" en la máquina cliente, como se presenta en la **Figura** 3.17.

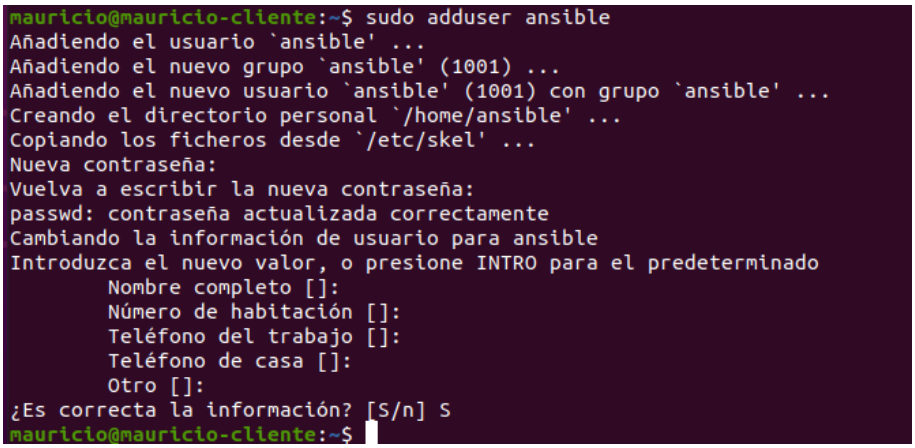

Figura 3.17 Creación de usuario Ansible en el cliente.

3.3 Implementación de un *playbook* para el servidor de logs.

Previo a la implementación del *playbook* se debe verificar la comunicación entre orquestador y cliente para asegurar que es posible la transferencia de datos.

Prueba de conexión entre orquestador y cliente

En esta instancia se debe comprobar la conexión entre orquestador y cliente, para lo cual, se realizó la prueba de ping-pong. Esta prueba consiste en ejecutar un ping a los clientes y si los clientes pueden responder a esta solicitud de ping, la respuesta es

reflejada como pong. En la **Figura** 3.18 se presenta la prueba de funcionamiento de Ansible, considerando que la prueba de ping-pong se realiza desde la máquina orquestadora y con el usuario Ansible.

```
ansible@mauricio-ansible-orquestador:/etc/ansible$ ansible all -m ping
192.168.242.142 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Figura 3.18 Prueba de ping-pong.

Para comprobar la operatividad del SLSE, primero se instaló de forma manual, es decir, el proceso se lo realizó desde la CLI. Luego, se elaboró un script que es dependencia del *playbook* para el proceso de automatización. De igual forma, este procedimiento se lo realizó para el ISUF.

Creación del SLSE de forma manual

A continuación, se describe los pasos para implementar el SLSE:

- En primer lugar, se debe ingresar a la página oficial de Splunk (**Figura** 3.19) y posteriormente crear una cuenta para tener acceso a los productos de Splunk.

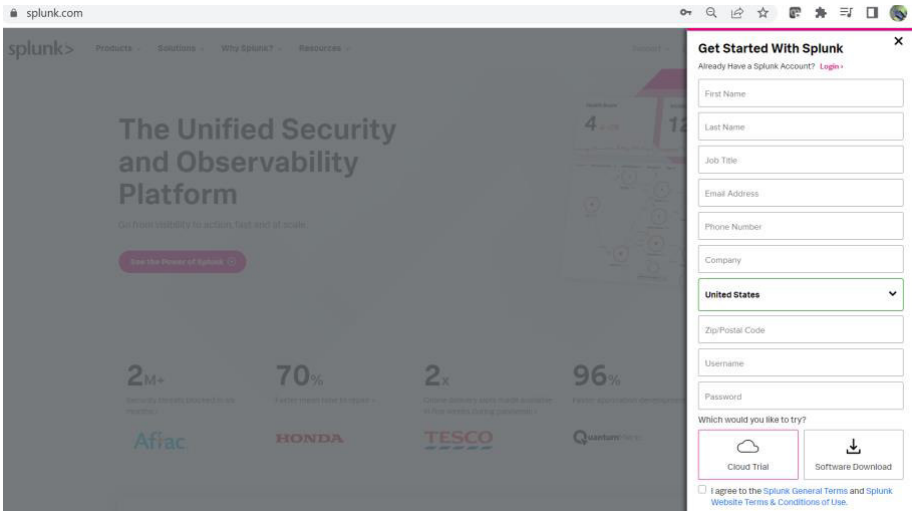

Figura 3.19 Creación de cuenta en Splunk.

- Después de crear una cuenta en Splunk, se procede a descargar el *software* para Linux en cualquiera de las extensiones mostradas en la **Figura** 3.20. En este caso se seleccionó la extensión .deb.

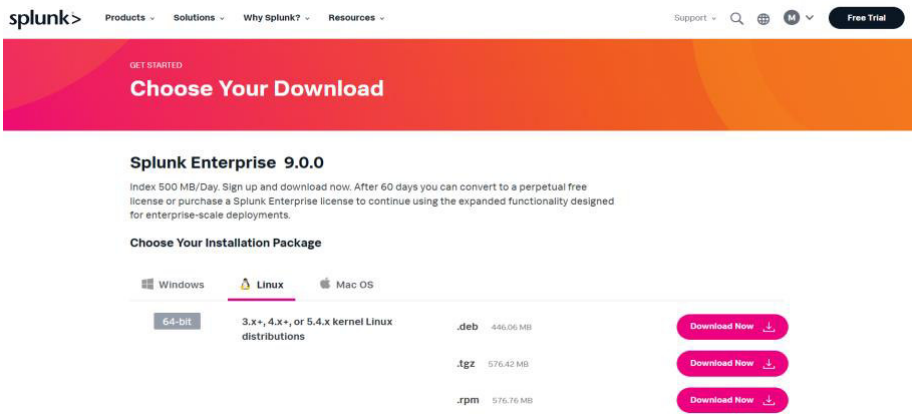

Figura 3.20 Descarga de Splunk Enterprise 9.0.0.

- Splunk también permite realizar la descarga del *software* desde la CLI, para lo cual, se debe copiar el comando que nos proporciona y pegarlo en la CLI del ordenador. En la **Figura** 3.21 se muestra el método de CLI que proporciona Splunk.

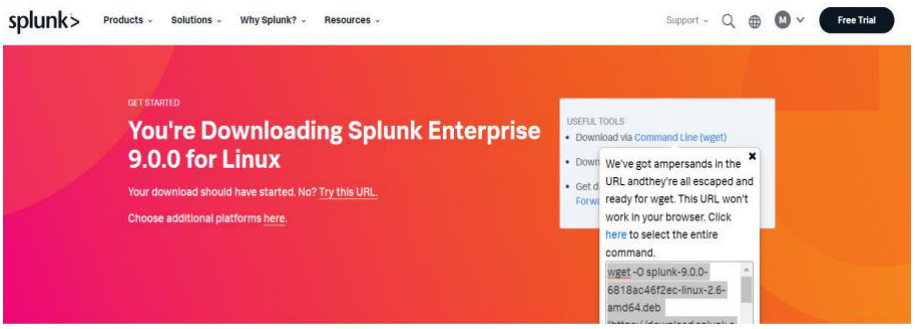

Figura 3.21 Método de descarga por CLI.

- Ya que el proceso de implementación del servidor es manual, se seleccionó el segundo método para descargar el *software*. Para ello, se especificó que el *software* se descargue en el directorio de Descargas mediante el comando presentado en la **Figura** 3.22.

```
mauricio@mauricio-server-splunk:~$ wget -O Descargas/splunk-9.0.0-6818ac46f2ec-
linux-2.6-amd64.deb "https://download.splunk.com/products/splunk/releases/9.0.0
/linux/splunk-9.0.0-6818ac46f2ec-linux-2.6-amd64.deb"
--2022-06-22 13:10:42--  https://download.splunk.com/products/splunk/releases/9
.0.0/linux/splunk-9.0.0-6818ac46f2ec-linux-2.6-amd64.deb
Resolviendo download.splunk.com (download.splunk.com)... 18.67.0.112, 18.67.0.6
1, 18.67.0.77, ...
Conectando con download.splunk.com (download.splunk.com)[18.67.0.112]:443... co
nectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 467724794 (446M) [binary/octet-stream]
Guardando como: “Descargas/splunk-9.0.0-6818ac46f2ec-linux-2.6-amd64.deb”

Descargas/splunk-9. 100%[==================>] 446,06M   625KB/s    en 15m 59s

2022-06-22 13:26:42 (476 KB/s) - “Descargas/splunk-9.0.0-6818ac46f2ec-linux-2.6
-amd64.deb” guardado [467724794/467724794]
```

Figura 3.22 Descarga de Splunk Enterprise 9.0.0 mediante CLI.

- Una vez descargado el *software*, se debe descomprimir el archivo y para ello se utiliza el comando dpkg de la **Figura** 3.23. Este comando permite descomprimir archivos de la extensión .deb.

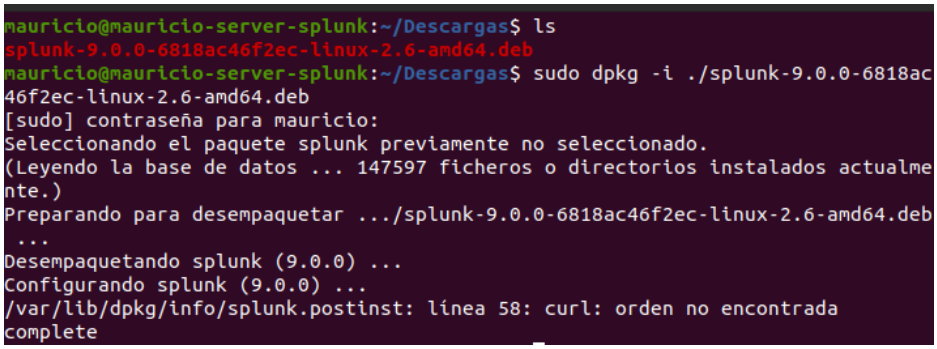

Figura 3.23 Comando para descomprimir archivos .deb.

- Una vez descomprimido el archivo, este se instala por defecto en el directorio /opt. Posteriormente, se inicia el servicio de Splunk Enterprise con ayuda del comando presentado en la **Figura** 3.24.

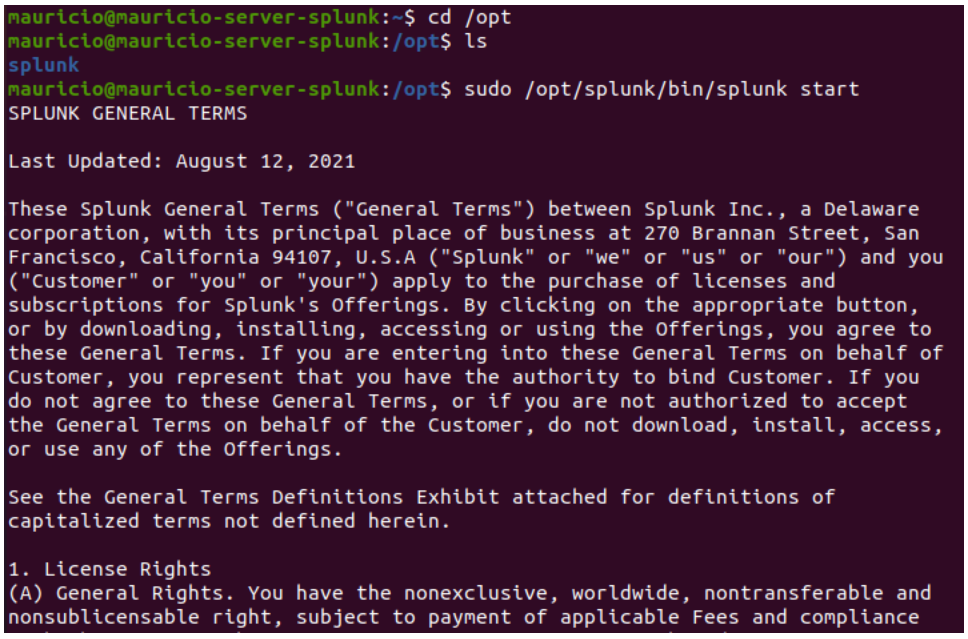

Figura 3.24 Inicio del servicio Splunk Enterprise.

- Durante el inicio del servicio se debe aceptar los términos y condiciones de la licencia de Splunk, tal como se muestra en la **Figura** 3.25. Mas adelante se solicitará que se registre el usuario y contraseña de administrador del SLSE que está próximo a levantarse, como se presenta en la **Figura** 3.26.

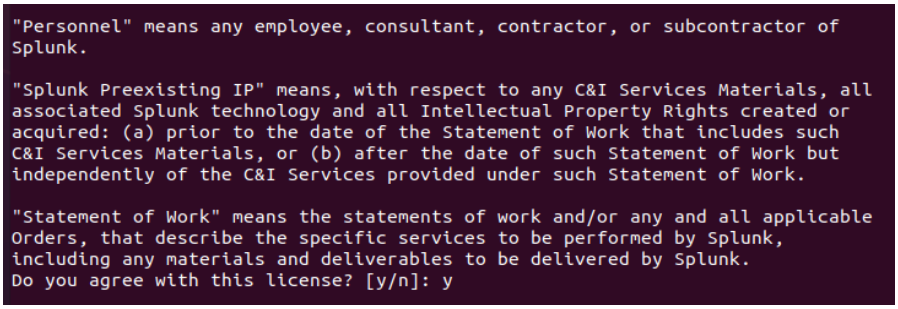

Figura 3.25 Aceptación de términos y condiciones.

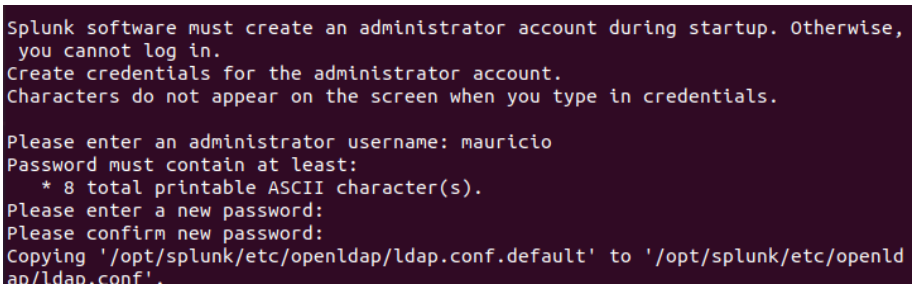

Figura 3.26 Registro de usuario y contraseña.

- Seguidamente se visualizará un mensaje en donde indica la interfaz web con la dirección y el número de puerto (**Figura** 3.27), por el cual se accede al SLSE.

Figura 3.27 Dirección web para el acceso al SLSE.

- Para verificar el funcionamiento, se debe ingresar al navegador y colocar la dirección IP del ordenador acompañado del número de puerto, como se presenta en la **Figura** 3.28.

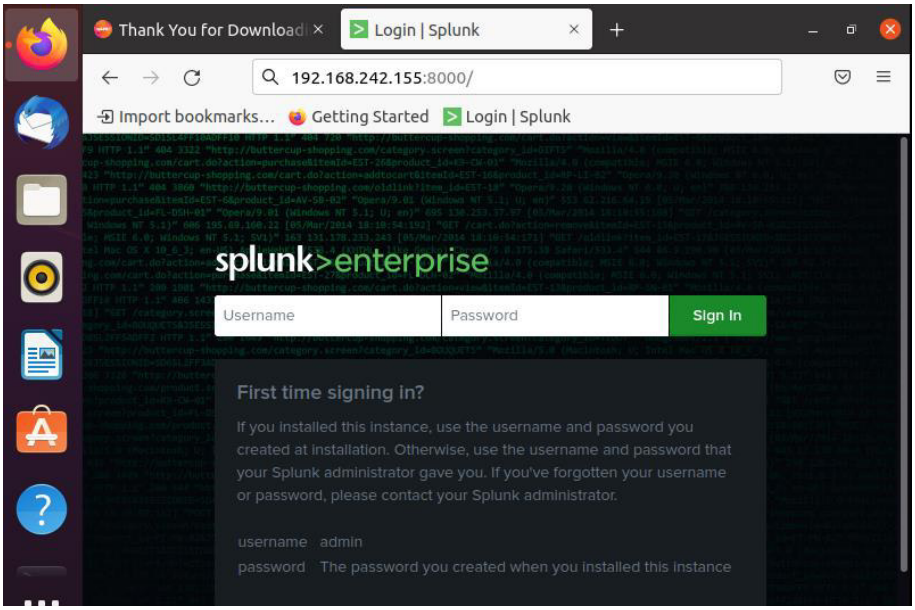

Figura 3.28 Verificación de funcionamiento del SLSE.

- Finalmente, se coloca el usuario y contraseña para acceder al SLSE, al iniciar sesión se despliega la interfaz de la página principal del servidor mostrada en la **Figura** 3.29.

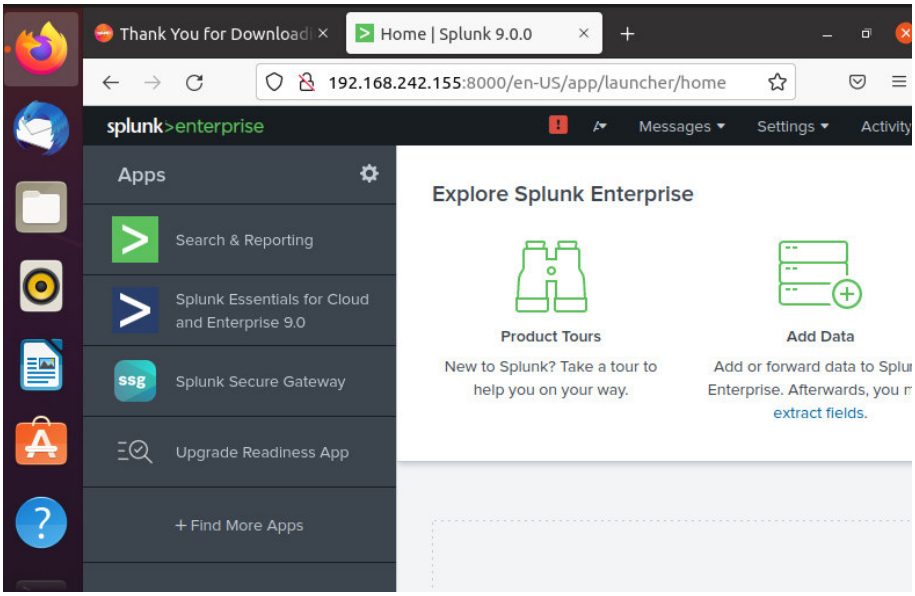

Figura 3.29 Interfaz principal del servidor.

Creación del *playbook* para el SLSE de forma automatizada

Para automatizar los procesos de TI se estructuró un script que mediante el *playbook* pueda levantar el servidor. Es por ello, que el primer paso fue crear un script **(Figura** 3.30) con todos los comandos que se ejecutó de forma manual en la creación del SLSE.

A continuación, se presenta la estructura del script el cual se encuentra conformado por 5 secciones:

- En la primera sección se establecen las variables que emplea el SLSE, por lo que en la línea 4 se define la variable de la URL desde donde se procederá a descargar el *software* y en la línea 5 se establece el directorio en donde se extraerá el instalador del servidor.
- En la segunda sección se ejecuta la descarga del *software*, para lo cual, en la línea 16 se hace uso del comando wget, este permite descargar archivos desde la web.
- En la tercera sección se ejecuta la extracción del instalador, para ello se ejecuta el comando dpkg el cual se visualiza en la línea 21.
- En la cuarta sección se establece las credenciales para acceder al servidor, con el fin de crear un archivo denominado user-seed.conf, como se presenta en la línea 26.
- Finalmente, en la última sección se da inicio al servicio del SLSE, por lo que es necesario ejecutar el comando de la línea 34.

```
1 #!/bin/bash
2
3 ### Seccion de variables
4 WGET_URL= "https://download.splunk.com/products/splunk/releases/9.0.0/linux/splunk-9.0.0-6818ac46f2ec-linux-2.6-amd64.deb"
5 INSTALL_DIR='/opt' #En este directorio no puede existir otro software de Splunk
6
7 ### Fin de la seccion de variables
8
9 if [ -d "/opt/splunk" ]
10 then
11         echo "El host ya tiene instalado el servidor Splunk Enterprise, esto está destinado solo para nuevas instalaciones."; exit
   1;
12 fi
13
14 # Seccion de descarga del software Splunk Enterprise
15 echo "Descargando Splunk Enterprise"
16 wget -O /opt/splunk-9.0.0-6818ac46f2ec-linux-2.6-amd64.deb "https://download.splunk.com/products/splunk/releases/9.0.0/linux/-
   splunk-9.0.0-6818ac46f2ec-linux-2.6-amd64.deb"
17
18 # Seccion de extraccion del instalador de Splunk Enterprise
19 echo "Extrayendo instalador de Splunk Enterprise"
20 cd /opt
21 sudo dpkg -i /opt/splunk-9.0.0-6818ac46f2ec-linux-2.6-amd64.deb
22 echo "El instalador de Splunk Enterprise fue exitoso"
23
24 # Seccion de registro de credenciales
25 echo "Configuracion de acceso al Servidor"
26 cat <<EOF >/opt/splunk/etc/system/local/user-seed.conf
27 [user_info]
28 USERNAME=mauricio
29 PASSWORD=mauricio
30 EOF
31
32 # Seccion de inicio del servicio de Splunk Enterprise
33 echo "Iniciando servicio de Splunk Enterprise"
34 /opt/splunk/bin/splunk start --no-prompt --accept-license --answer-yes
35
36 # Hecho!
37 echo "Instalacion completada"
```

Figura 3.30 Script para la instalación de SLSE.

Una vez creado el script, se procedió a ingresar a la máquina orquestadora Ansible, en donde se creó un directorio denominado “Scripts” con el comando que se presenta en la **Figura** 3.31.

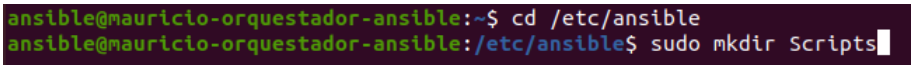

Figura 3.31 Creación del directorio para los scripts.

Después, se debe crear el script para el SLSE con el comando mostrado en la **Figura** 3.32. La denominación del script puede ser aleatoria ya que no tiene influencia, sin embargo, la extensión debe ser .sh para que el *playbook* lo reconozca como un script.

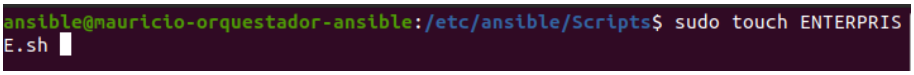

Figura 3.32 Creación del script para el SLSE.

Posteriormente, se procede a copiar el código del script que se creó en la **Figura** 3.30 y con ayuda del comando de la **Figura** 3.33 se edita el script "ENTERPRISE.sh".

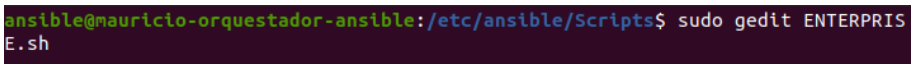

Figura 3.33 Comando para editar el script del SLSE.

A continuación, se debe crear un directorio denominado "*Playbooks*" dentro del orquestador Ansible, para ello se utilizó el comando presentado en la **Figura** 3.34**.**

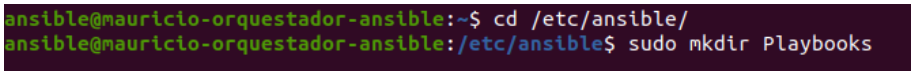

Figura 3.34 Creación del directorio para los *playbooks*.

Dentro del directorio de "*Playbooks*" se debe crear un archivo con la extensión. yml, ya que los *playbooks* manejan esta extensión. El *playbook* que se creó se denominó "SplunkEnterprise.yml" con ayuda del comando de la **Figura** 3.35.

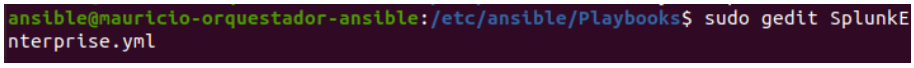

Figura 3.35 Creación del *playbook* para el SLSE.

Dentro del *playbook* se tiene la estructura mostrada en la **Figura** 3.36, en la línea 2 se establece los host clientes a los que se ejecutara el *playbook*, en este caso se tiene un grupo denominado "serverSplunk". La línea 3 hace referencia a un módulo de configuración, en donde se puede determinar si existe conexión con los hosts remotos. La línea 4 determina si las tareas se ejecutan en el usuario root. La línea 5 establece las tareas del *playbook*, en este caso se tiene 2 tareas en específicas, las cuales son transferir el script y ejecutar el script en las máquinas remotas.

La primera tarea está compuesta por la línea 7 que establece el nombre de la tarea del *playbook*. Seguidamente en la línea 8 se hace uso del módulo *copy*, el cual permite copiar un archivo desde la máquina local o remota a una ubicación en la máquina remota. En la línea 9 se establece la ubicación del script desde donde se va transferir y en la línea 10 el destino o directorio a donde se va a transferir el script. Por último, en la línea 11 se asigna permisos de lectura, grabación y ejecución para el directorio en donde se alojará el script.

La segunda tarea está compuesta por la línea 13 que establece el nombre de la tarea del *playbook*. Seguidamente en la línea 14 se hace uso del módulo *command*, el cual permite ejecutar el script que fue transferido a la máquina remota desde la CLI.

Figura 3.36 Estructura del *playbook* para el SLSE.

Una vez levantado el SLSE, se puede realizar las pruebas de funcionamiento, para lo cual, existe 2 maneras de indexar los datos al servidor. La primera forma es utilizar los datos del ordenador en el cual está alojado el servidor, un ejemplo es cargar la carpeta de logs en donde se encuentra los errores de diferentes procesos.

La segunda forma es indexar los datos de diferentes ordenadores, sin embargo, es necesario utilizar el *software* ISUF. A continuación, se presenta el proceso de forma manual y automatizada para la implementación del ISUF.

Creación del ISUF de forma manual

La implementación de este *software* se realizó sobre una máquina virtual con el sistema operativo Ubuntu 20. Para la instalación se debe ingresar a la página oficial de Splunk y seleccionar el método de descarga por CLI y después ejecutar la línea de comandos que brinda Splunk, como se presenta en la **Figura** 3.37.

Figura 3.37 Descarga del *software* ISUF.

Una vez finalizada la descarga, se debe ingresar al directorio en el que se encuentra el *software* y extraer el archivo con el comando que se presenta en la **Figura** 3.38, pues sirve para descomprimir archivos con la extensión .tgz.

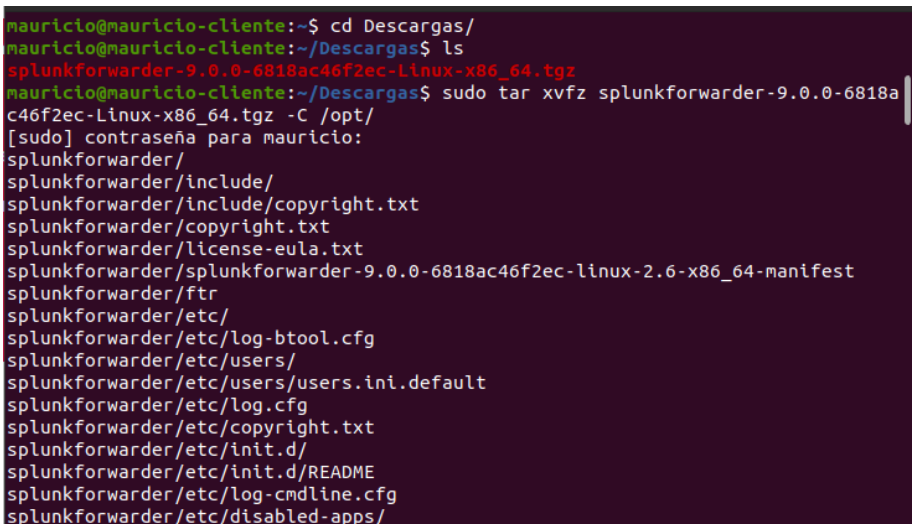

Figura 3.38 Comando para descomprimir archivos .tgz.

Después de extraer el archivo, es necesario habilitar el inicio de arranque del ISUF con el comando mostrado en la **Figura** 3.39. Seguidamente, se debe aceptar los términos y condiciones de la licencia del ISUF.

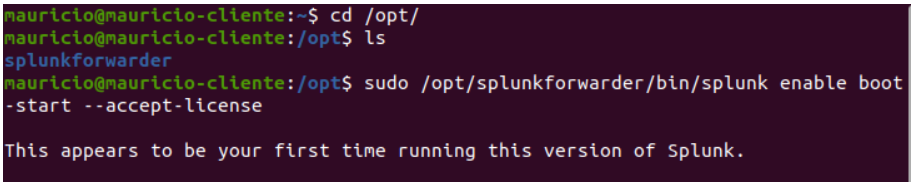

Figura 3.39 Habilitación del ISUF.

Posteriormente, con el comando presentado en la **Figura** 3.40 se añade la dirección y el puerto del SLSE por el cual va recibir los datos desde el ISUF.

```
mauricio@mauricio-cliente:/opt$ sudo /opt/splunkforwarder/bin/splunk add forward
-server 192.168.242.155:9997
Added forwarding to: 192.168.242.155:9997.
mauricio@mauricio-cliente:/opt$
```

Figura 3.40 Agregación de la dirección IP del SLSE.

Una vez añadido la dirección del SLSE, con ayuda del comando de la **Figura** 3.41 se puede visualizar el estado de la conexión entre el servidor y la máquina cliente.

```
mauricio@mauricio-cliente:/opt$ sudo /opt/splunkforwarder/bin/splunk list forwar
d-server
Active forwards:
        None
Configured but inactive forwards:
        192.168.242.155:9997
mauricio@mauricio-cliente:/opt$
```

Figura 3.41 Estado de conexión.

En el caso de que la conexión se encuentre inactiva, se puede iniciar el servicio del ISUF mediante el comando que se muestra en la **Figura** 3.42.

```
mauricio@mauricio-cliente:/opt$ sudo /opt/splunkforwarder/bin/splunk start

Splunk> CSI: Logfiles.

Checking prerequisites...
        Checking mgmt port [8089]: open
                Creating: /opt/splunkforwarder/var/run/splunk/appserver/i18n
                Creating: /opt/splunkforwarder/var/run/splunk/appserver/module
s/static/css
                Creating: /opt/splunkforwarder/var/run/splunk/upload
                Creating: /opt/splunkforwarder/var/run/splunk/search_telemetry
                Creating: /opt/splunkforwarder/var/spool/splunk
                Creating: /opt/splunkforwarder/var/spool/dirmoncache
                Creating: /opt/splunkforwarder/var/lib/splunk/authDb
                Creating: /opt/splunkforwarder/var/lib/splunk/hashDb
New certs have been generated in '/opt/splunkforwarder/etc/auth'.
        Checking conf files for problems...
                Invalid key in stanza [webhook] in /opt/splunkforwarder/etc/sy
stem/default/alert_actions.conf, line 229: enable_allowlist (value: false).
                Your indexes and inputs configurations are not internally cons
istent. For more information, run 'splunk btool check --debug'
        Done
        Checking default conf files for edits...
        Validating installed files against hashes from '/opt/splunkforwarder/s
plunkforwarder-9.0.0-6818ac46f2ec-linux-2.6-x86_64-manifest'
        All installed files intact.
```

Figura 3.42 Activación de la conexión.

Para corroborar que el estado del servidor se activó, se debe volver a ejecutar el comando de la **Figura** 3.41 y verificar el cambio como se presenta en la **Figura** 3.43. Debido a que anteriormente se inició el servicio, el sistema solicitará un usuario y contraseña de administrador para verificar el estado de la conexión.

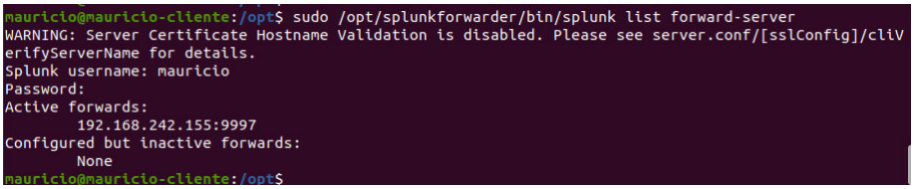

Figura 3.43 Verificación del estado de conexión.

Una vez activado el ISUF, se procede a agregar una nueva entrada de datos al SLSE para enviar información desde el directorio deseado con el comando de la **Figura** 3.44.

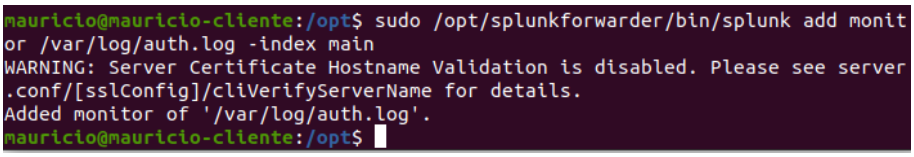

Figura 3.44 Creación de una nueva entrada para el envío de información.

Para comprobar la indexación de datos desde la máquina cliente al servidor, se necesita un puerto por el cual reciba la información. Es por ello, que se debe agregar un puerto de escucha. A continuación, se presenta el proceso para activar el puerto en el SLSE y verificar el funcionamiento.

Primero, se debe ingresar en el menú de *Settings* y seleccionar la opción *Forwarding and Receiving*, luego se debe ingresar el puerto “9997” en la opción *Configure receiving*, como se presenta en la **Figura** 3.45.

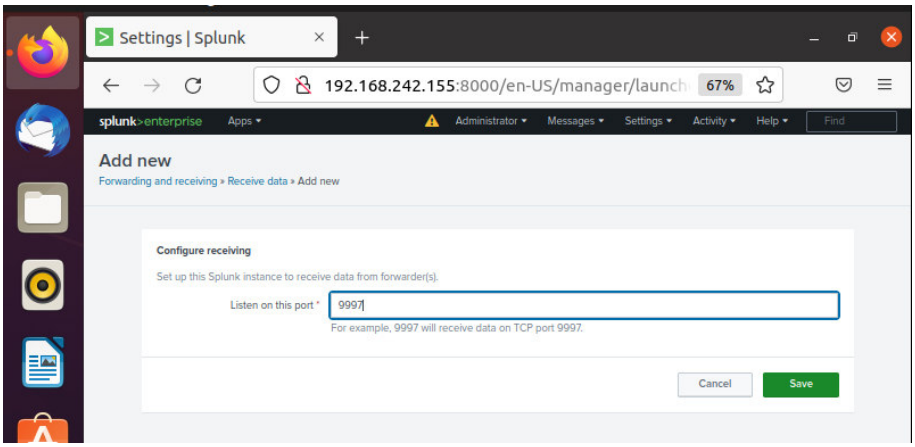

Figura 3.45 Configuración del puerto de escucha del SLSE.

Para visualizar si se estableció la conexión, es necesario dirigirse al menú *Search & Reporting*, seleccionar la opción *Data Summary* y se desplegará en la pantalla el nombre de la máquina cliente que se conectó al SLSE, como se muestra en la **Figura** 3.46.

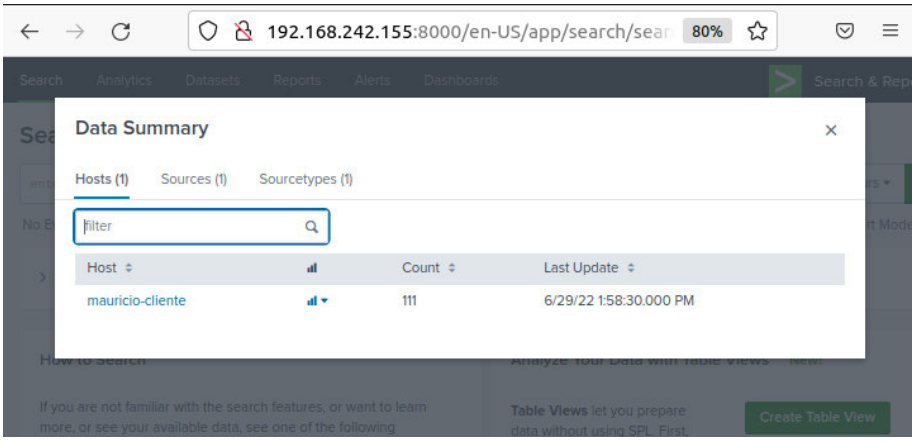

Figura 3.46 Conexión cliente-servidor de forma manual.

Creación del *playbook* para el ISUF de forma automatizada

Para automatizar los procesos de TI, se estructuró un script que mediante el *playbook* pueda indexar los datos desde la máquina cliente al servidor. Es por ello que se creó un script (**Figura** 3.47) con todos los comandos ejecutados de forma manual en la creación del ISUF. A continuación, se presenta la estructura del script el cual se encuentra conformado por 7 secciones:

- En la primera sección se establece las variables que emplea el indexador, por lo que en la línea 4 se define la variable de la URL desde donde se procederá a descargar el *software* en la versión 9.0.0 y en la línea 5 se elige el directorio en donde se extraerá. Además, tiene algunas variables para la conexión entre el ISUF y el SLSE, como se puede visualizar en las líneas 6,7 y 8.
- En la segunda sección se ejecuta la descarga del *software* mediante el comando wget, que se encuentra en la línea 19.
- En la tercera sección se extrae el ISUF, para ello se ejecuta el comando tar -zxf, el cual se visualiza en la línea 24.
- En la cuarta sección mediante el comando de la línea 29 se añade la dirección IP del SLSE al cual se va enviar los datos.
- En la quinta sección se establece las credenciales para ISUF, para ello se crea un archivo denominado user-seed.conf, como se presenta en la línea 33.
- En la sexta sección con el comando de la línea 41 se da inicio al servicio de indexación de datos.
- Finalmente, en la última sección con ayuda del comando de la línea 45 se habilita el servicio de arranque del indexador.

```
#!/bin/bash

### Seccion de variables
WGET_URL= "https://download.splunk.com/products/universalforwarder/releases/9.0.0/linux/splunkforwarder-9.0.0-6818ac46f2ec-Linux-x86_64.tgz"
INSTALL_DIR='/opt'   #Directorio de instalacion, considerar que no puede existir 2 o mas softwares de Splunk en este directorio
DEPLOYMENT_SERVER=192.168.242.156 #Es la direccion del Servidor Splunk Enterprise
DEPLOYMENT_PORT=9997 #Puerto por donde recibe los datos el Servidor
SPLUNK_TEMPADMINPASS=mauricio #Autenticacion

### Fin de la seccion de variables

if [ -d "/opt/splunkforwarder" ]
then
        echo "El host ya tiene instalado el Indexador Splunk Universal Forwarder, esto esta destinado solo para nuevas instalaciones"; exit 1;
fi

# Seccion de descarga del software Splunk Universal Forwarder
echo "Descargando Splunk Universal Forwarder"
wget -O /opt/splunkforwarder-9.0.0-6818ac46f2ec-Linux-x86_64.tgz "https://download.splunk.com/products/universalforwarder/releases/9.0.0/linux/-
splunkforwarder-9.0.0-6818ac46f2ec-Linux-x86_64.tgz"

# Seccion de extraccion del Indexador Splunk Universal Forwarder
echo "Extrayendo Splunk Universal Forwarder"
cd /opt
tar -zxf /opt/splunkforwarder-9.0.0-6818ac46f2ec-Linux-x86_64.tgz
echo "El Indexador Splunk Universal Forwarder fue exitoso"

# Seccion de configuracion del Servidor al cual se va indexar la informacion
echo "Configurando parametros del servidor"
/opt/splunkforwarder/bin/splunk add forward-server 192.168.242.156:9997 --no-prompt --accept-license --answer-yes

# Seccion de registro de credenciales
echo "Configurando credenciales del Indexador Splunk Universal Forwarder"
cat <<EOF >/opt/splunkforwarder/etc/system/local/user-seed.conf
[user_info]
USERNAME=mauricio
PASSWORD=mauricio
EOF

# Seccion de inicio del Indexador Splunk Universal Forwarder
echo "Iniciando servicio del Indexador Splunk Universal Forwarder"
/opt/splunkforwarder/bin/splunk start --no-prompt --accept-license --answer-yes

# Seccion de habilitacion del servicio
echo "Habilitando el inicio de arraque"
/opt/splunkforwarder/bin/splunk enable boot-start -systemd-managed 0

# Hecho!
echo "Instalacion completada!"
```

Figura 3.47 Script para la instalación del ISUF.

En el directorio “Scripts” que se creó mediante los comandos de la **Figura** 3.31 se debe añadir un nuevo script para el ISUF, como se presenta en la **Figura** 3.48.

```
ansible@mauricio-orquestador-ansible:/etc/ansible/Scripts$ sudo touch INDEXADOR.sh
```

Figura 3.48 Creación del script para el ISUF.

Posteriormente se procede a copiar el código del script que se creó de la **Figura** 3.47 con ayuda del comando que se presenta en la **Figura** 3.49.

```
ansible@mauricio-orquestador-ansible:/etc/ansible/Scripts$ sudo gedit INDEXADOR.sh
```

Figura 3.49 Comando para editar el script del ISUF.

Luego se debe ingresar al directorio “*Playbooks*” que se creó con el comando de la **Figura** 3.34 y crear un nuevo *playbook* para el ISUF, el cual se puede visualizar en la **Figura** 3.50.

```
ansible@mauricio-orquestador-ansible:/etc/ansible/Playbooks$ sudo gedit SplunkForwarder.yml
```

Figura 3.50 Creación del *playbook* para el ISUF.

La estructura del *playbook* del ISUF es idéntica al *playbook* del SLSE, con la diferencia de que se debe cambiar los nombres de los scripts y el grupo al cual se va a ejecutar el

playbook, además se añadió la tarea de indexar información al servidor, tal como se muestra desde la línea 16 en la **Figura** 3.51.

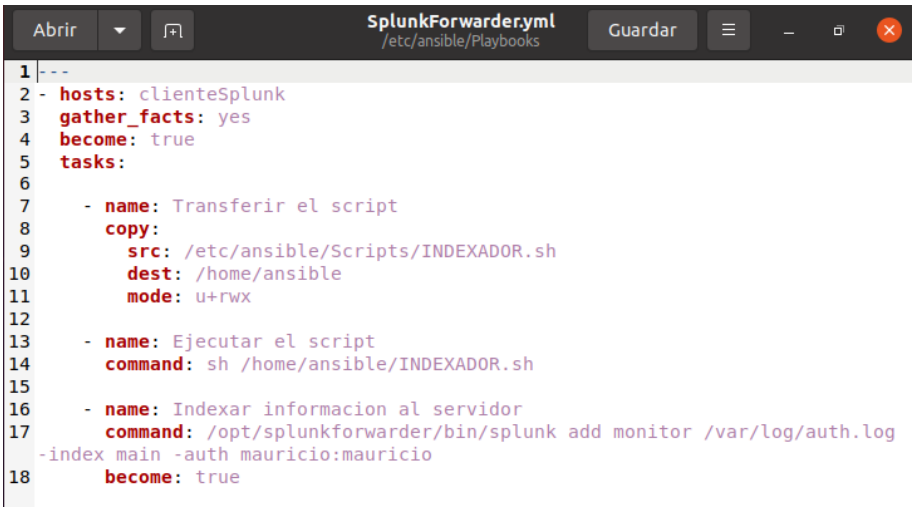

Figura 3.51 Estructura del *playbook* para el ISUF.

3.4 Verificación de funcionamiento del *playbook*.

Funcionamiento del *playbook* para el SLSE

Después de estructurar y crear el *playbook* dentro de Ansible se procedió a verificar la funcionalidad del *playbook*. Para ello, fue necesario crear una máquina virtual solo con la conexión de Ansible y ejecutar el *playbook* con el comando de la **Figura** 3.52, además aquí se muestran los resultados de las tareas que arroja Ansible al ejecutar el *playbook*.

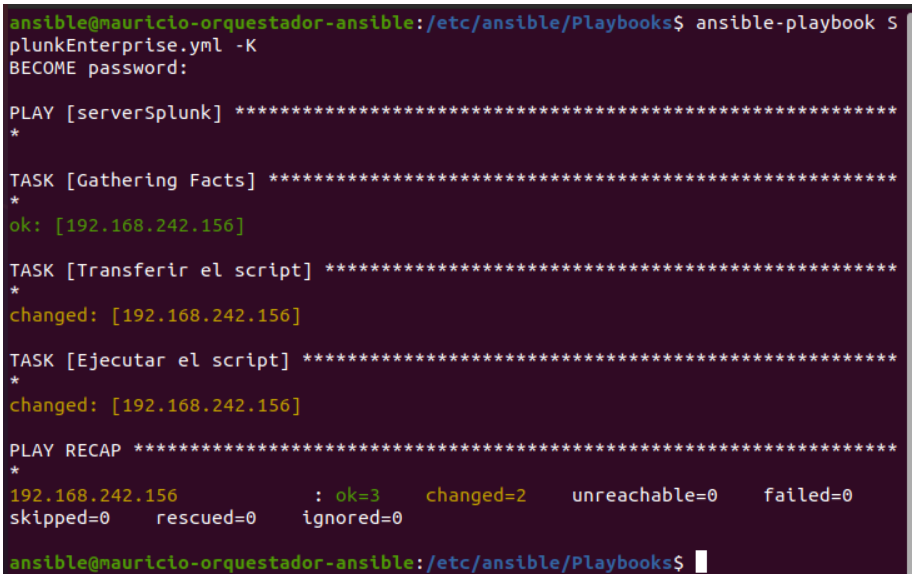

Figura 3.52 Resultados de la ejecución del *playbook* para el SLSE.

Una vez ejecutado el *playbook*, se procede a verificar si el servicio se encuentra levantado y para ello se debe ingresar al navegador y escribir la IP acompañado del número de puerto, tal como se muestra en la **Figura** 3.53.

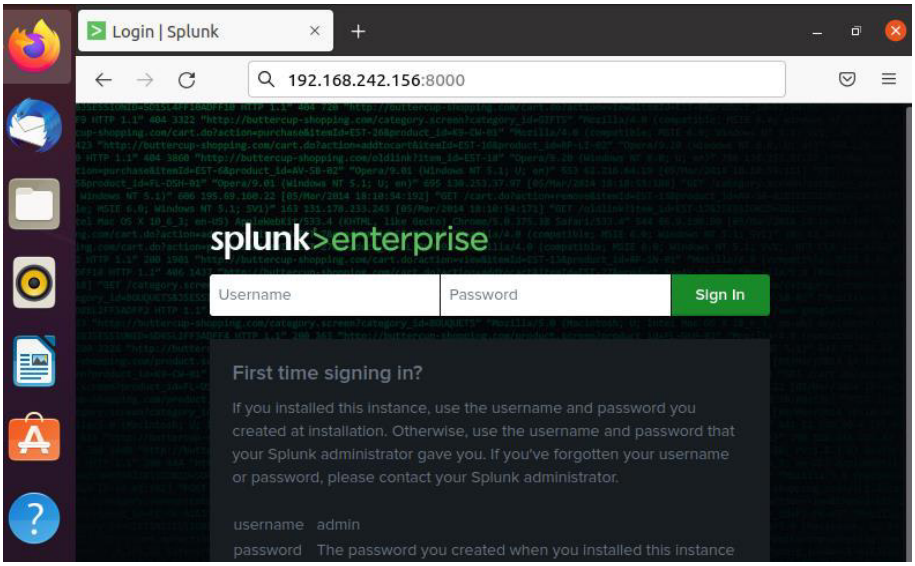

Figura 3.53 Verificación de funcionamiento del SLSE de forma automatizada.

Finalmente, para ingresar al SLSE se debe colocar las credenciales registradas en el script y después se desplegará la pantalla principal, como se puede visualizar en la **Figura** 3.53.

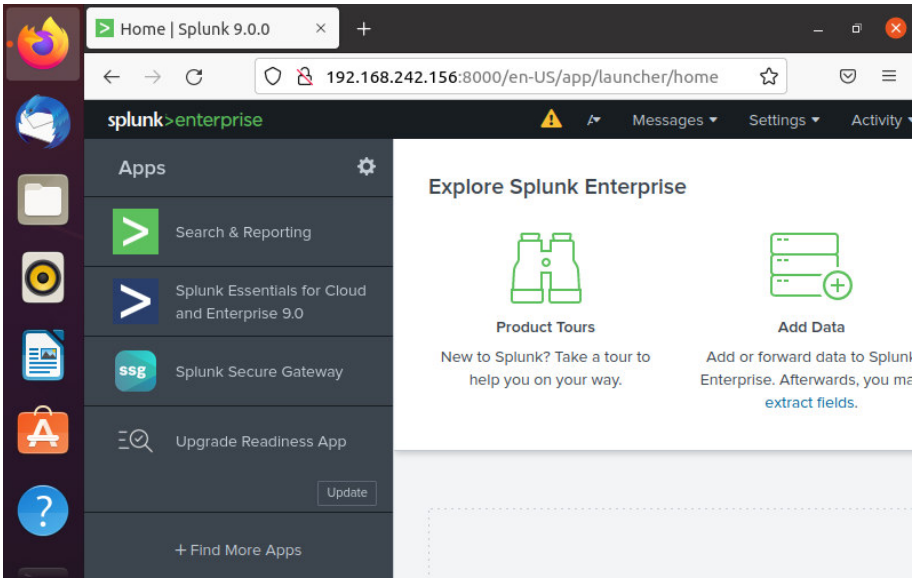

Figura 3.54 Interfaz principal del servidor de forma automatizada.

No olvidar añadir el puerto de escucha (**Figura** 3.45).

Funcionamiento del *playbook* para el ISUF

Para verificar la funcionalidad del *playbook*, es necesario crear una máquina virtual con la conexión de Ansible y ejecutar el *playbook* con el comando de la **Figura** 3.55, además aquí se muestran los resultados de las tareas que arroja Ansible al ejecutar el *playbook* del ISUF.

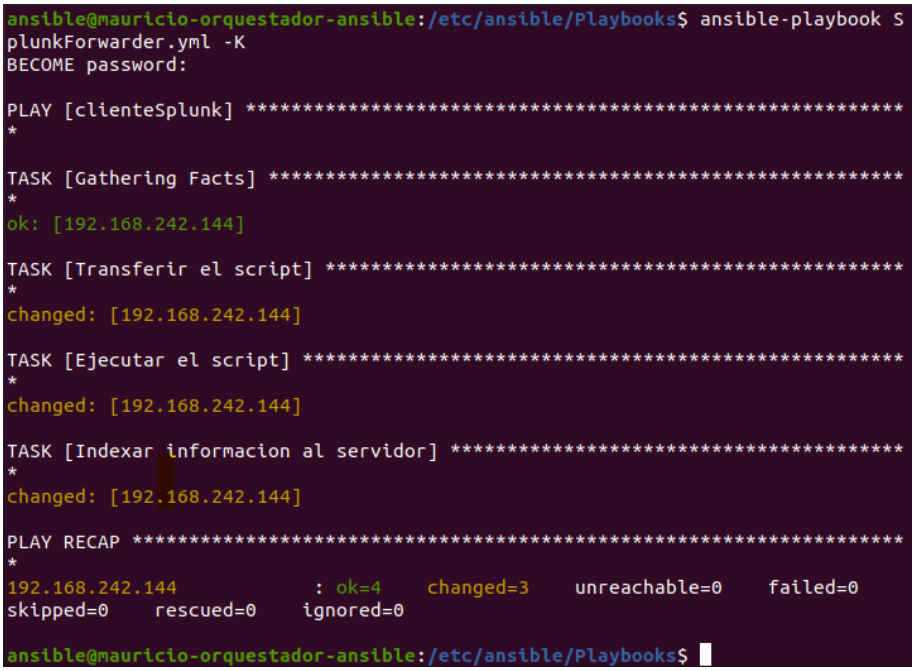

Figura 3.55 Resultados de la ejecución del *playbook* para el ISUF.

Una vez ejecutado el *playbook*, se procede a verificar si la Indexación de datos es correcta, para ello se debe ingresar al servidor y ver si se añadió la máquina cliente, como se puede evidenciar en la **Figura** 3.56.

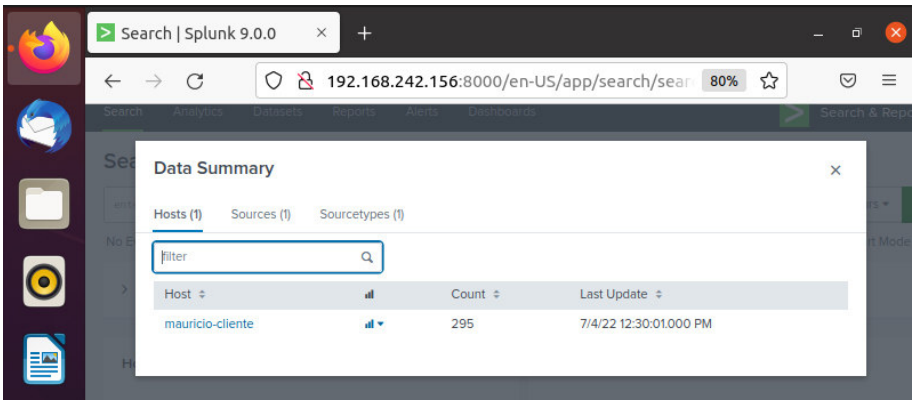

Figura 3.56 Conexión cliente-servidor de forma automatizada.

4 CONCLUSIONES

- El estudio del manejo de DevOps permitió conocer al *software* de Ansible, las plataformas de servidores de logs (Splunk, Graylog y Papertrail) y el proceso de automatización en el área de TI.
- Splunk es una herramienta óptima y multifuncional para las actividades del área de TI, además la interfaz del servidor Splunk Enterprise fue fácil de manipular al momento de verificar la indexación de datos de los nodos clientes.
- A partir del proceso manual de implementación del servidor, se pudo diseñar un script con todos los pasos y mediante el *playbook* automatizar el levantamiento del servidor y del indexador.
- La prueba de ping-pong permitió verificar la conexión entre el orquestador y los nodos clientes previo al despliegue del servidor Splunk Enterprise, de manera que, se evitaron posibles fallas de envío y recepción de datos al momento de ejecutar el *playbook*.
- Mediante los *playbooks* se automatizó la implementación del SLSE y del ISUF, de igual forma se comprobó la conectividad entre el servidor y los nodos clientes.
- Con el uso del módulo *copy* de Ansible se pudo obtener mayor facilidad para el diseño del *playbook* del servidor de logs, debido a que permitió copiar el script desde la máquina orquestadora hacia las máquinas administradas y después con el módulo *command* compilar el script desde la CLI de los nodos clientes.
- Debido a que las direcciones IP son diferentes en cada nodo cliente, se ejecuta el comando de la **Figura** 3.14 en cada máquina, con ello se comparte la clave SSH y establece la conexión Ansible.
- En el proceso manual la dirección IP de la máquina virtual fue el de la **Figura** 3.46, mientras que en el proceso automatizado fue el de la **Figura** 3.56, con ello se corrobora que la implementación de cada proceso no fue fraudulenta.
- En la **Figura** 3.36 y **Figura** 3.51 se presenta las estructuras de los *playbooks* para la automatización del SLSE e ISUF, también en la **Figura** 3.30 y **Figura** 3.47 se muestra los scripts empleados para la elaboración de los *playbooks*. A través de estos componentes se obtuvo una correcta operatividad del proyecto.

5 RECOMENDACIONES

- En caso de realizar una nueva implementación ya sea del SLSE o del ISUF y los resultados de las tareas del *playbook* no se ejecuten correctamente, se debe

ingresar a la página oficial de Splunk y verificar las actualizaciones del *software* y licencias, para posteriormente, modificar el script y reemplazar las líneas de código de descarga.

- Un problema que puede presentar Ansible son sus actualizaciones, debido a que los módulos y la sintaxis del código del *playbook* pueden cambiar, es por necesario ingresar a la página oficial de Ansible y corroborar posibles cambios.
- Ansible no necesita de *software* adicional en los nodos clientes, sin embargo, es necesario verificar que Python y openssh-server se encuentren instalados en cualquier distribución de Linux, ya que normalmente estos paquetes vienen instalados por defecto.
- Dentro del *playbook* se debe dar los permisos de lectura, escritura y ejecución para los usuarios de las máquinas remotas, ya que es imprescindible para compilar el script desde la CLI de los nodos clientes.
- Para la conexión entre el orquestador y los nodos clientes es indispensable la generación de una llave SSH en el orquestador y copiarla mediante el código de la **Figura** 3.14**.**
- En el SLSE se debe definir el puerto de escucha por el que se va a recibir los datos de los nodos clientes mediante el ISUF que comúnmente es el puerto 9997.
- Para añadir una nueva entrada de datos al SLSE se debe agregar en el *playbook* del ISUF la dirección de cualquier directorio (directorio con información de una.

6 REFERENCIAS

[1] R. A. Castillo Solis, “Renovación de la infraestructura tecnológica VMware vSphere para la migración de servicios en el área de TI de una empresa de Seguridad.,” Universidad Tecnológica del Perú , Lima , 2021.

[2] M. Ayuso, “Implementación de máquinas virtuales para la instalación de aplicaciones que se ejecutan bajo diferentes sistemas operativos,” in *I Jornadas Nacionales sobre Aplicación de las Tecnologías de la Información y Comunicaciones Avanzadas (ATICA 2009)* , vol. 1, Alcalá, 2009.

[3] Y. Fernández Romero and K. García Pombo, “Virtualización,” Habana , Sep. 2011.

[4] Red Hat Ansible, “Complete IT automation,” 2020.

[5] L. Enciso and C. E. Morales Iñiguez, “Ansible una estrategia de administración y configuración automatizada sobre GNS3 con OSPF para redes empresariales medianas,” Loja, Jul. 2021.

[6] J. C. Gómez Navarro, “Servidor de Logs Centralizado,” Universidad Politécnica de Valencia, Valencia, 2014.

[7] A. Chuvakin, K. Schmidt, and C. Phillips, *Logging and Log Management*. Waltham, 2013.

[8] IBM, “Log level values,” *WebSphere Application Server Network Deployment*, Jun. 19, 2021.

[9] A. M. Redondo Felipe and F. de J. Núñez Cárdenas, “DevOps: un vistazo rápido DevOps: a quick look,” 2022. [Online]. Available: https://repository.uaeh.edu.mx/revistas/index.php/huejutla/issue/archive

[10] F. Jiménez, “DevOps: La Evolución de IT para la entrega continua,” Aug. 27, 2018.

[11] Red Hat, “Ansible Basics,” Nov. 02, 2020.

[12] Red Hat, “Installation Guide,” Jul. 12, 2022.

[13] Shashank Hegde, “10 módulos de Ansible que debes conocer,” Sep. 11, 2019.

[14] S. Hegde, “10 módulos de Ansible que debes conocer,” Sep. 11, 2019.

[15] Red Hat Ansible, “Workshop - Escribir su primer Playbook,” 2022.

[16] splunk, “splunk,” 2022.

[17] O. Gonzáles, “Generación de ciberinteligencia con Splunk,” Universidad Politécnica de Valencia , Valencia, 2021.

[18] A. Ruiz, “Centralización y análisis de eventos de seguridad con Graylog,” 2019.

[19] Graylog, “Architecture,” 2018.

[20] S. Luz, “GNU LinuxSoftware Papertrail será tu gestor de logs favorito con soporte Cloud y un potente buscador,” Jun. 17, 2016.

[21] A. Shaout and R. Banksto, “Enterprise it logging in the cloud: analysis and approaches,” Mar. 2014.

[22] Capterra, “Comparing 3 Log Management Software Products,” Mar. 19, 2018.

7 ANEXOS

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 22 de agosto de 2022

De mi consideración:

Yo, FERNANDO BECERRA, en calidad de Director del Trabajo de Integración Curricular titulado CREACIÓN DE SERVIDOR DE LOGS CON *SOFTWARE* LIBRE CON ANSIBLE elaborado por el estudiante RONY ALCARRAZ de 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 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.

https://epnecuador-my.sharepoint.com/:b:/g/personal/fernando_becerrac_epn_edu_ec/EUaEMYhW6mVEv289ubIwiQ4BJfB6hPjOI0G9usUhQOuSLg?e=RBwtEe

Atentamente,

FERNANDO BECERRA

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces

Código QR de la implementación y pruebas de funcionamiento